

8

# INFORMATIKA

8

INFORMATIKA

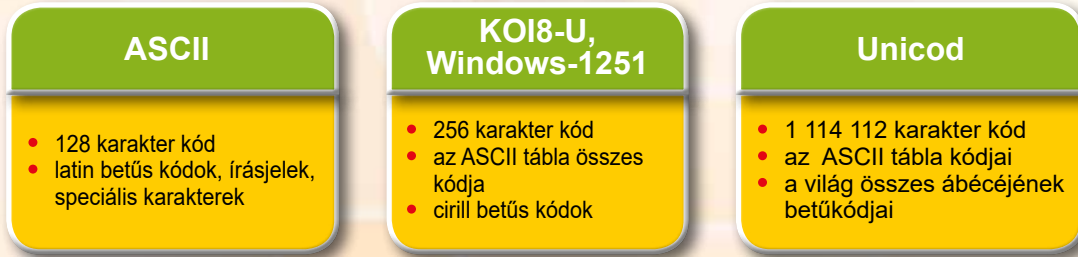


2021

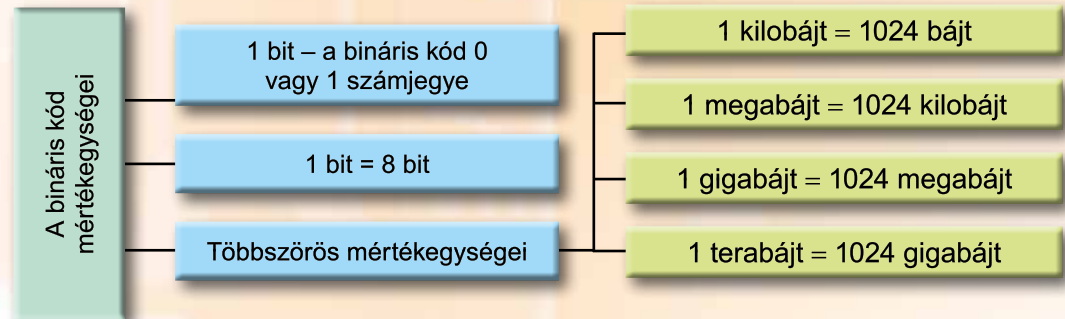
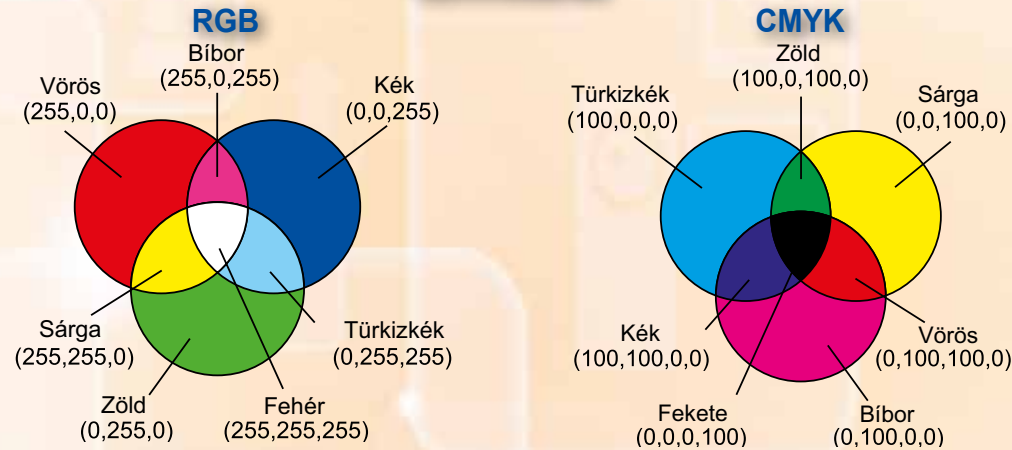


# Az 1. fejezet összefoglalása

## Kódtáblázat

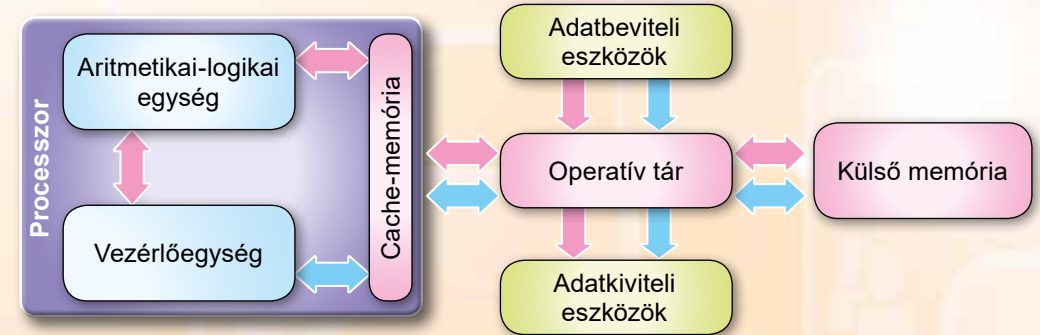


## Színmodellek

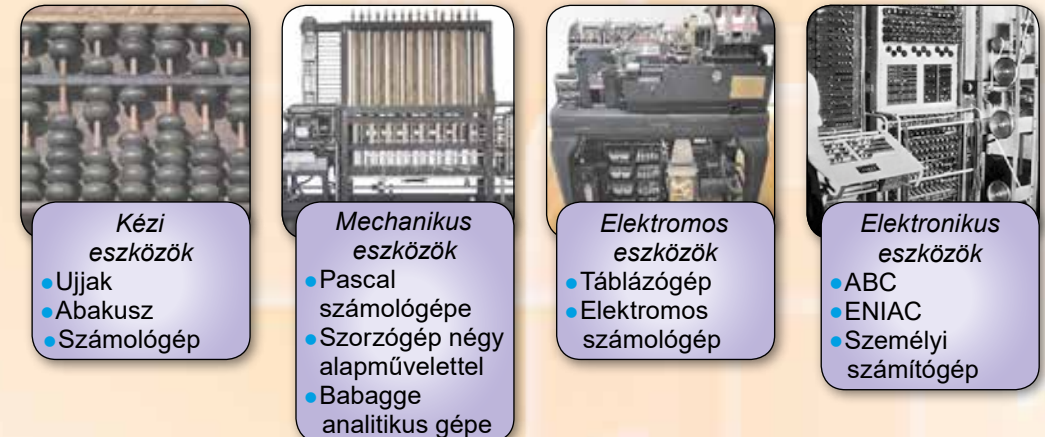


# A 2. fejezet összefoglalása

## Információs folyamatok megvalósításának modellje alapvető számítógépes eszközök segítségével

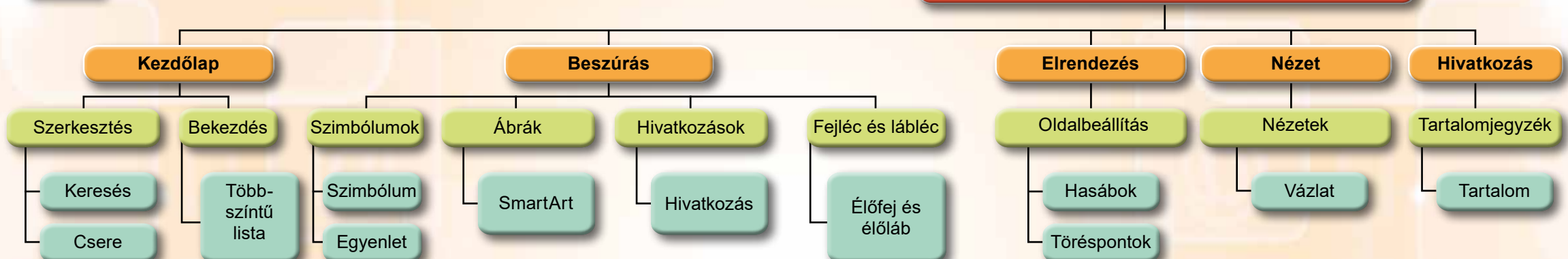


## Az információs folyamatok megvalósításának eszközei



# A 3. fejezet összefoglalása

## A szövegszerkesztők alapvető műveletei



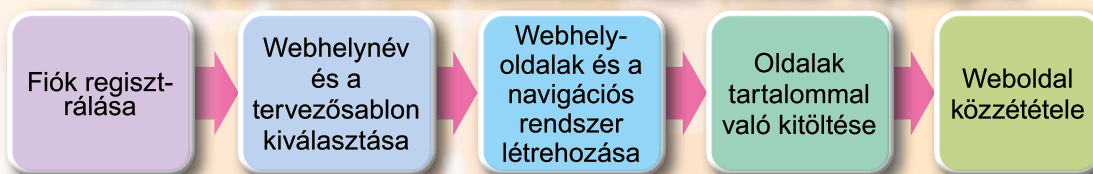


## A 4. fejezet összefoglalása

### HTML TAG-ek

| TAG                | Rendeltetése                         | Példa   |
|--------------------|--------------------------------------|---|
| <b>h1, ..., h6</b> | Cím létrehozása                      | <h1> HTML nyelv </h1>                                   |
| <b>p</b>           | Bekezdés létrehozása                 | <p>A HTML tag-jei – a dokumentum formázó utasításai</p> |
| <b>br</b>          | Sortörés                             | <br>  |
| <b>a</b>           | Hiperhivatkozás beszurása            | <a href="second-page.html">A második oldalhoz</a>       |
| <b>img</b>         | Kép beszurása fajlból                |                       |
| <b>hr</b>          | Vizszintes elvlasztó vonal beszurása | <hr>  |

### Webhely készítése online weboldaltervező rendszer segítségével



## Az 5. fejezet összefoglalása

### Alkalmazások

Lejátszók (playerek)



Megtekintés, meghallgatás

Grabberek, rekorderek



Audio- és videoadatok rögzítése

Konverterek



Fájlformátum átalakítók

Szerkesztők



Audio- és videoadatok felhasználói szintű kezelése

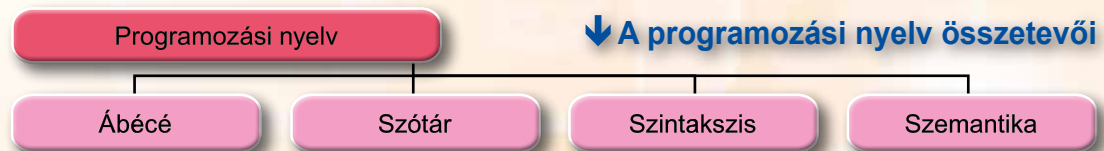
Stúdióprogramok



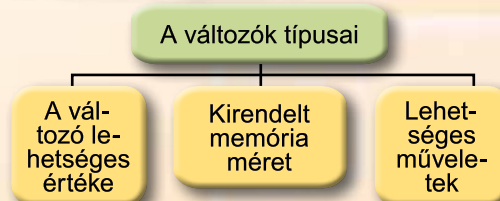
Hang- és videoadatok professzionális szerkesztése

### Műveletek

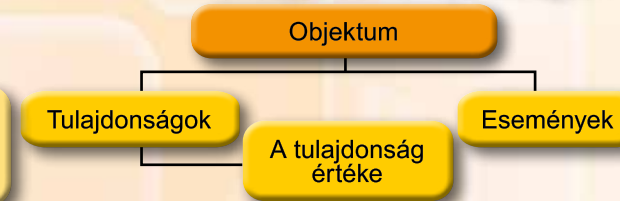
## A 6. fejezet összefoglalása



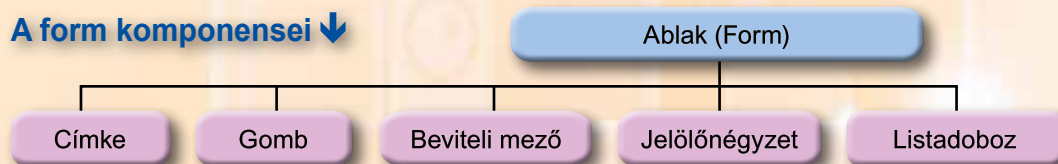
### A változók típusai ↓



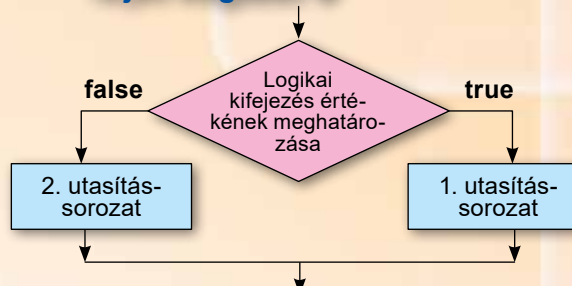
### ↓ Objektum tulajdonságok



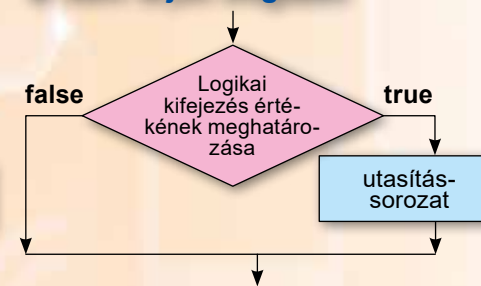
### A form komponensei ↓



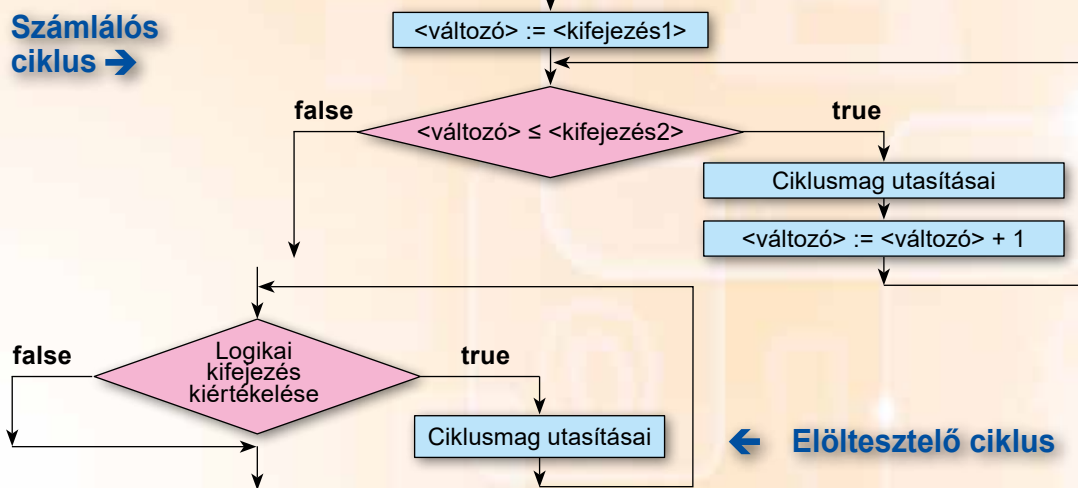
### Teljes elágazás ↓



### ↓ Nem teljes elágazás



### Számlálós ciklus →



### ← Előtesztelő ciklus

# INFORMATIKA

Tankönyv  
a magyar oktatási nyelvű általános  
középfokú tanintézetek 8. osztálya számára



*Ajánlotta Ukrajna Oktatási és Tudományos Minisztériuma*

Львів  
Видавництво „Світ”  
2021



УДК 004(075.3)  
I-74

Перекладено за виданням:

**Інформатика** : підруч. для 8-го кл. закл. заг. серед. освіти /  
Йосиф Ривкінд [та ін.]. — Київ : Генеза, 2021

Авторський колектив :

**Йосиф Ривкінд, Тетяна Лисенко, Людмила Чернікова, Віктор Шакоцько**

*Рекомендовано Міністерством освіти і науки України*  
(наказ Міністерства освіти і науки України від від 22.02.2021 № 243)

**Видано за рахунок державних коштів. Продаж заборонено**

I-74 **Інформатика** : підруч. для 8 кл. з навч. угор. мов. закл. заг.  
серед. осв. / Йосиф Ривкінд [та ін.] ; пер. Ф. Г. Біро. – Львів : Світ,  
2021. – 256 с. : іл.

ISBN 978-966-914-352-5

УДК 004(075.3)

*Навчальне видання*

**РИВКІНД Йосиф Якович, ЛИСЕНКО Тетяна Іванівна,  
ЧЕРНІКОВА Людмила Антонівна та ін.**

**ІНФОРМАТИКА**

**Підручник для 8 класу з навчанням угорською мовою  
закладів загальної середньої освіти**

*Рекомендовано Міністерством освіти і науки України*

**Видано за рахунок державних коштів. Продаж заборонено**

Підручник відповідає Державним санітарним нормам і правилам  
«Гігієнічні вимоги до друкованої продукції для дітей»

Переклад з української мови

Перекладач *Біро Ференц Гейзович*

Угорською мовою

Редактор *Христина Зикань*

Обкладинка *Олександра Павленка*

Художнє оформлення, комп'ютерна обробка ілюстрацій

*Василя Марущинця*

Формат 70×100/16. Ум. друк. арк. 20,8. Обл.-вид. арк. 19,56.

Тираж 2036 пр. Зам. № 21-300

Державне підприємство „Всеукраїнське спеціалізоване видавництво „Світ”  
79008 м. Львів, вул. Галицька, 21

Свідоцтво суб'єкта видавничої справи серія ДК № 4826 від 31.12.2014  
www.svit.gov.ua, e-mail: office@svit.gov.ua

Друк ПрАТ «Білоцерківська книжкова фабрика»

09100, Київська обл., м. Біла Церква, вул. Леся Курбаса, буд. 4

Свідоцтво суб'єкта видавничої справи серія ДК № 5454 від 14.08.2017

© Ривкінд Й. Я., Лисенко Т. І., Чернікова Л. А.,  
Шакоцько В. В., 2021

© Видавництво «Генеза», оригінал-макет, 2021

© Біро Ф. Г., переклад угорською мовою, 2021

ISBN 978-966-914-352-5 (угор.)


ISBN 978-966-11-1179-9 (укр.)



## KEDVES NYOLCADIKOSOK!


Ebben a tanévben folytatjuk az igen fontos és érdekes tantárgy, az **informatika** tanulmányozását.


Az előző osztályokban megismerkedtetek az operációs rendszer objektumaival és műveleteket végeztetek ezekkel, információs modelleket hoztatok létre, grafikai, szöveges, multimédiás és számadatokat dolgoztatok fel, megismerkedtetek az elektronikus levelezéssel és az internetes kereséssel, megtanultatok csapatban dolgozni az interneten, algoritmusokat állítottatok össze különböző végrehajtók számára.

A nyolcadik osztályban megismerkedtek majd az adatok kódolásával, áttekintitek a számítógép hardverét és szoftverét, folytatjátok a szöveges, valamint multimédiás adatok feldolgozását, megtanultok saját honlapokat létrehozni és közzétenni az interneten. Megtanultok videofájlokat készíteni, hangokat feldolgozni, programokat fejleszteni és végrehajtani egy programozási rendszerben. Ebben segít majd nektek ez a tankönyv.


A könyvet fejezetekre osztottuk. Minden fejezet pontokból, és azok alpontjaiból áll. Minden pont ismétlő kérdésekkel kezdődik. Ezeket megválaszolva könnyebb lesz az új tananyag elsajátítása. A kérdéseket -vel jelöljük.


Olvasátok a tankönyvet figyelmesen! Minden pont végén az **Összefoglalás**  rubrikában megtaláljátok annak rövid összefoglalását. A gyakorlati készségek begyakorlásához a **Használjuk a számítógépet!**  rubrikán belül található gyakorló feladatok és azok részletes kidolgozása található. Ezt


a rubrikát a <https://cutt.ly/8hedSjp> linkre kattintva, vagy a  QR-kód segítségével találhatjátok meg.

Az egyes pontok **Felelj a kérdésekre!** című résszel zárulnak . Ajánljuk, hogy a pont elolvasása után válaszoljatok ezekre a kérdésekre. A kérdések melletti jel azt mutatja, hogy az adott kérdés megválaszolása:

- – alapszintű;
- – középszintű;
- \* – vagy emelt szintű ismereteknek felel-e meg.






Ugyanígy meg vannak jelölve a kérdéseket követő gyakorlati feladatok a **Végezd el a feladatokat!**  rubrikában. A házi feladatnak ajánlott fel-

adatokat -al, a párokban vagy kiscsoportokban elvégzendő feladatokat

-al jelöltük.





A törzsanyagon kívül a könyvben a következő kiegészítéseket találjátok:

-  – *Ha többet szeretnél tudni*
-  – *Tudtad-e, hogy...?*
-  – *Érdekes történelmi tények*
-  – *Linux-használóknak*
-  – *Videószerkesztő-használóknak*
- **Magyarázó szótár** (zöld háttérrel kiemelve)

A 8. osztályban folytatjátok a *Szövegfeldolgozás* témakörét a **Microsoft Office 365** szövegszerkesztő alkalmazásban, ugyanezen anyag leírása a **Microsoft Word 2010**-hez a tankönyv elektronikus mellékletében található.

Az *algoritmusok* és a *programok* fejezetben kétféle jelölésre számíthattok:

-  – *A Pascalt használóknak;*
-  – *A Pythont használóknak.*

A tankönyv végén **Szótár** található, mely tartalmazza az alapfogalmak meghatározásait.

A feladatokhoz és a gyakorlati munkához tartozó fájlokat az interneten, a <http://sites.google.com/pu.org.ua/allinf> oldalon, vagy a QR-kód segítségével találjátok meg.

***Sok sikert kívánunk a legérdekesebb  
és legmodernebb tudomány,  
az INFORMATIKA tanulmányozásához!***



*A szerzők*



# 1. fejezet

## Adatok kódolása

*Ebben a fejezetben megismerkedtek:*

- az adatfeldolgozással mint információs folyamattal
- az üzenetek kódolásával és dekódolásával
- a szövegek és képek kódolásával
- a kódtáblázattal
- a színmodellel
- bináris kódolással
- a bináris kód mértékegységeivel

### 1.1. Az adatfeldolgozás mint információs folyamat. Üzenetek kódolása és dekódolása



1. Milyen információs folyamatokat ismertek? Mi az üzenetfeldolgozás folyamatának lényege?
2. Milyenek lehetnek az üzenetek megadásának módjai?
3. Mik az adatok? Hozzatok fel példákat!

#### *Az adatfeldolgozás mint információs folyamat*

A modern társadalomban nagyon sok szakma kapcsolódik a szavak, számok, grafikai ábrázolások formájában előállított adatok feldolgozásához. A könyvelők az elvégzett munkákat jellemző információk alapján végzik el a fizetések kiszámítását. Az orvosok a vizsgálati eredményekre támaszkodva határozzák meg a diagnózist. A meteorológusok a levegő mozgásáról összegyűjtött adatok alapján készítik el az előrejelzéseiket. A csillagászok a különböző kozmikus testek mozgáspályáját, azok elhelyezkedése alapján számítják ki. A művész a könyv tartalma alapján készíti el annak illusztrációit (1.1. ábra). A tanulók is adatokat dolgoznak fel, amikor feladatokat oldanak meg, fogalmazásokat írnak, grafikonokat rajzolnak stb.



1.1. ábra. Adatok feldolgozása különböző szakmákban



Ahogy már tudjátok, az **adatok feldolgozása** az a folyamat, amelynek során új adatokat hozunk létre a meglévők alapján.

Amikor például egy matematikafeladatot oldunk meg, akkor a feladat feltételéből ismert kiinduló adatokból új adatokat – a feladat eredménye – állítunk elő. A futóverseny eredményének meghatározása azt jelenti, hogy a versenyzők neveit az általuk elért időeredmények szerint növekvő sorrendbe rendezzük. A táblázatkezelőben egy diagram elkészítésekor a meglévő adatokat új, grafikai formában ábrázoljuk.

Az adatokat különféle eszközök – például számítógépek – segítségével is feldolgozhatjuk. Ez esetben az adatokat olyan formában kell előállítani, amilyenben a számítógép képes ezeket feldolgozni, valamint meg kell határozni azokat a szabályokat, amelyek alapján a feldolgozást végre kell hajtani.

### Üzenetek kódolása

Az üzenet tárolása, továbbítása, feldolgozása során gyakran szükség van arra, hogy megváltoztassuk az üzenet reprezentációjának (megjelenítésének) módját.

- A szóbeli üzeneteket papírra rögzítik, ekkor az emberi hangot betűkre cserélik. Ez leginkább az üzenet tárolása céljából történik.
- A telefonbeszélgetés során a szóbeli üzeneteket elektromos jelekre cserélik. Ennek célja az, hogy az üzenetet nagy távolságra lehessen továbbítani.
- A matematikaórán a számokat jelölő szavakat számjegyekké, a műveleteket jelölő szavakat műveleti jelekké alakítjuk. Ebben a formában könnyebb az információ feldolgozása.
- Az üzenet tartalmának védelme céljából rejtjelezhetik azt, amelynek során a betűket más betűkre, számokra vagy jelekre cserélik.

**Rejtjel, kód** (lat. *codex* – könyv szóból) – megegyezéssel jelölésrendszer, amelyet az üzenetek tárolása, továbbítása és feldolgozása céljából hoztak létre.

Amikor tehát az üzenet tartalmát alkotó jeleket más jelekre cserélik, **kódolják az üzenetet**.

A kódolás az adatfeldolgozás egyik példája. Ilyenkor megváltozik a megjelenítési mód, de az üzenet tartalma nem változik.

A kódolást gyakran alkalmazzák a mindennapokban is, hogy az üzeneteket ne csak az ember, de a gép is képes legyen feldolgozni. Az árukat például leggyakrabban vonalkóddal kódolják (1.2. ábra). A vonalkód számjegyei az áru nevét, származási helyét, gyártóját kódolják. Ezt a számkódot az ember általi információfeldolgozás és rendszerezett adattárolás céljából hozták létre. A számjegyeket a vonalak vastagsága és az azok közötti távolság kódolja. Ezt a grafikai kódot olvassa be a vonalkódozó, és dolgozza fel a későbbiek során a számítógép.



1.2. ábra.

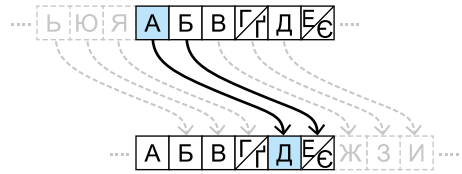
Egy árucikk vonalkódja

Az üzenetek kódolása céljából nemcsak az alkalmazott karaktereket kell ismerni, hanem a kód előállításának módját is.

A történelemből jól ismert az a kódolási mód, amelyet Gaius Julius Caesar (i. e. 100–44) alkalmazott a



tábornokaival folytatott levelezésében. A rejtjelezés lényege az, hogy minden egyes betűt az ábécében egy tőle meghatározott távolságra lévő betűvel kell helyettesíteni (1.3. ábra). A *Hello* szóból 4 betűnyi eltolással ily módon a *Khoos* szót kapjuk.



1.3. ábra. A 4 pozícióval eltolt Caesar-kód előállítás

### Az üzenetek dekódolása

Ismerve azt a szabályt, amely szerint az üzenetet kódolták, dekódolhatjuk. A **dekódolás** az a folyamat, amelynek elvégzése után a kódból visszafejtjük az eredeti üzenetet.

Dekódolást végzünk például akkor, ha hangosan felolvassunk egy szöveget, zenét játszunk kottából, megfejtünk egy titkosírást vagy a vonalkód alapján azonosítunk egy árucikket stb.

Ha egy üzenetet 4 pozíciós jobbra tolt Caesar-kóddal kódoltak, akkor a dekódolás során a betűket 4 hellyel balra kell eltolni. A *mő rdtsü* üzenet dekódolva tehát *jó napotra* változik.

Az információ kódolása és dekódolása információfeldolgozási feladatnak tekinthető.



#### Használjuk a számítógépet!



Végezzétek el a gyakorlati feladatot a <https://cutt.ly/dhdeH0F> linkről, vagy a QR-kód segítségével!



#### Összefoglalás

**Az adatfeldolgozás** – új adatok előállítása a meglévők alapján.

Az üzenetek **kódolása** – az üzenetet reprezentáló egyik szimbólumsorozat cseréje egy másik szimbólumsorozatra.

Az üzenetek kódolására azok tárolása, továbbítása, átalakítása és védelme céljából van szükség. Ilyenkor megváltozik a megjelenítés mód, de az üzenet tartalma nem változik.

Az üzenetek kódolásakor meg kell adni az új kód előállításánál alkalmazandó szimbólumokat és az új kód képzésének szabályait.

Az üzenet **dekódolása** alatt az eredeti üzenet visszaállítását értjük a kódolt üzenet alapján.



#### Felelj a kérdésekre!

- 1°. Mit értünk információfeldolgozás alatt?
- 2°. Milyen új adatokat állíthatunk elő a következő adatok alapján:
  - a) az üzletben levő árucikkek mennyisége és ára;
  - b) a tanulók magassága;
  - c) két település távolsága és a közöttük haladó vonat sebessége?



- 3°. Mit értünk az *üzenetek kódolása* alatt?
- 4°. Milyen célból végezhetünk kódolást?
- 5°. Milyen célból kódolják a zenét hangjegyekkel; a közlekedés szabályait közlekedési táblákkal?
- 6°. Milyen üzeneteket kódolhatunk színekkel, grafikai jelekkel?
- 7°. Mit szükséges meghatározni a kódolás megkezdése előtt?
- 8°. Hogyan kódolják a *Tengeri csata* játékban a hajók helyzetét? Miért?
- 9°. Mit értünk *dekódolás* alatt?



## Végezd el a feladatokat!

1°. Hozz fel példákat üzenetek kódolására, amelyeket a zene, fizika, kémia, földrajz stb. tanulmányozása során alkalmaztatok!

2°. Kódold 4 betű eltolásával keletkező Caesar-kóddal az alábbi üzeneteket:

- a) *informatika*;      b) *iskolánk*;      c) *üzenetek kódolása és dekódolása!*



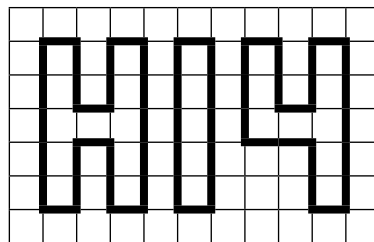
3°. Az üzenetek 3 betű eltolásos Caesar-kóddal lettek kódolva. Végezd el a dekódolást:

- a) *úbgšügóegu*;                      b) *mp kfr*;  
c) *ógúngbfrfqüü c ücöy!*

4°. Egy ceruzarajz a következőképpen van kódolva: a nyíl a ceruza mozgásának irányát mutatja, a szám pedig az elmozdulás mértékét kockákban. Hozd létre a képet!

- a)  $\uparrow 5 \rightarrow 3 \downarrow 5 \leftarrow 1 \uparrow 4 \leftarrow 1 \downarrow 4 \leftarrow 1$ ;  
b)  $\leftarrow 3 \downarrow 5 \rightarrow 3 \uparrow 1 \leftarrow 2 \uparrow 1 \rightarrow 2 \uparrow 1 \leftarrow 2 \uparrow 1 \rightarrow 2 \uparrow 1$ .

5°. Kódold az 1.4. ábrán látható képet az előző feladatban megismert módszer szerint: a nyíl azt az irányt jelöli, amelyben a ceruza a rácsvonal mentén mozog, a szám pedig a kockák számát mutatja!



1.4. ábra. Az 5. feladathoz

6°. Egészítsd ki a megismert kódolási módszert oly módon, hogy meg lehessen adni a ceruza mozgását rajzolással és anélkül is!

7°. Az egyik legrégebbi rejtjelezési forma az i. e. III. századból származó Polübiosz-négyzet vagy Polübiosz-tábla. Itt minden betűt (vagy betűpárt) egy négyzetrácsban helyezünk el. Kódoláskor minden betűt egy oszlop és egy sor jelöl, amelyek metszésében a betű áll. Az ábra az ukrán ábécé egy lehetséges kódolását mutatja.

|   | 1 | 2 | 3 | 4   | 5   | 6 |
|---|---|---|---|-----|-----|---|
| 1 | А | Б | В | Г/Ґ | Д   | Е |
| 2 | Є | Ж | З | И/Й | І/Ї | К |
| 3 | Л | М | Н | О   | П   | Р |
| 4 | С | Т | У | Ф   | Х   | Ц |
| 5 | Ч | Ш | Щ | Ь   | Ю   | Я |

1.5. ábra. Polübiosz-tábla



Dekódold az alábbi üzeneteket:

- a) 53 43 13 52 21 52 42;
- b) 62 43 51 34 31 11 33 33 65!

8\*. Készítsd el a magyar ábécé kódolására alkalmas Polübiosz-négyzetet, és kódold az alábbi szövegeket:

- a) *informatika*;
- b) *üzenet!*

9\*. Gondolj ki egy kódolási módszert, majd ennek segítségével kódold a következő mondatokat:

- a) *Ma szép nap van*;
- b) *Szeretem a focit!*



10\*. Keress információkat a következő kódolási módszerekről:

- a) Morze-ábécé;
- b) szemafor-nyelv;
- c) Braille-írás;
- d) Selam vagy napkeleti virágnyelv!

## 1.2. Szövegek és képek kódolása



1. Mit értünk az *üzenetek kódolása* alatt? Mi célból végzik?
2. Mondj példákat szöveges üzenetek kódolására!
3. A felépítési módjuk szerint milyen számítógépes grafikákat ismertek? Jellemezzétek őket!

### A szöveg karaktereinek kódolása

Ahhoz, hogy szöveges adatokat számítógéppel dolgozhassunk fel, számok segítségével kell azt kódolnunk.

A szöveg kódolása során **kódtáblázatokat** használnak, amelyekben minden egyes alkalmazandó karakterhez egy szám van rendelve. Az USA-ban 1963-ban kidolgoztak egy ilyen kódtáblázatot, amelyet telexgépekhez fejlesztettek ki. Ez később szabvánnyá vált a számítástechnikában és az **ASCII (American Standard Code for Information Interchange)** – amerikai szabványos információcsere kód – nevet kapta.

Az **ASCII** kódtábla az angol abécé betűit, számjegyeket, elválasztójeleket, szerkesztési és formázási karaktereket tartalmaz, amelyek a 0 és a 127 közötti számokat feleltetnek meg (1.1. táblázat, 10. old).

**Teletype** (a latin *tele* – távoli és az angol *type* – nyomtatás szavakból) – elektromechanikus írógép, amelyek vezetékeken továbbított információt nyomtattak.

Az **ASCII** kódtábla karaktereit az 1. melléklet tartalmazza, amelyet a <https://cutt.ly/NkBXn8o> vagy a [uk.wikipedia.org/wiki/ASCII](http://uk.wikipedia.org/wiki/ASCII) weboldalakon találhattok meg.

Az **ASCII**-kód csak az angol betűket tartalmazza. Más nyelvek karaktereinek kódolásához ettől eltérő kódra volt szükség. A **KOI8-U (KOI – код обміну інформацією (információcsere kódja))** és a **Windows-1251** az **ASCII** karaktereken kívül még a cirill betűket is tartalmazza. Ezeknek a karaktereknek a 128 és a 255 közötti kódok felelnek meg.

## Az ASCII kódtábla kódcsoportjai

| Intervallum                                      | Kódcsoport                    | Példa | Leírás   |
|--|-------------------------------|-------|--|
| 0-tól 31-ig, 127                                 | Speciális karakterek          | 10    | Új sor jel                                       |
|  |                               | 13    | Sor elejére ugrás                                |
|  |                               | 27    | Az <b>Esc</b> billentyű lenyomásával egyenértékű |
| 32-től 64-ig,<br>91-től 96-ig,<br>123-tól 126-ig | Számjegyek és elválasztójelek | 32    | Szókőz   |
|  |                               | 48    | A 0 számjegy                                     |
|  |                               | 123   | {  |
| 65-től 90-ig,                                    | Nagybetűk                     | 65    | A  |
|  |                               | 66    | B  |
|  |                               | 90    | Z  |
| 97-től 122-ig                                    | Kisbetűk                      | 97    | a  |
|  |                               | 98    | b  |
|  |                               | 122   | z  |

A **Windows** operációs rendszerben a **Windows-1251** kódolás a szabványos. Itt az ukrán „a” karakter kódja 224, az „i” kódja 179, a „r” kódja pedig 180. A 128 és 255 közé eső kódoknak megfelelő karaktereket megtekinthetitek a 2. mellékletben a <https://cutt.ly/NkBXn8o> vagy [uk.wikipedia.org/wiki/Windows-1251](http://uk.wikipedia.org/wiki/Windows-1251) weboldalakon.

**Cirill ábécé** – a szláv nyelvek ábécéje. Ide tartozik az ukrán, orosz, belorusz, bolgár és más nyelvek. Az angol és egyes nyugat-európai nyelv **latin ábécét** használ.

A 0 és 255 közötti kódok elégségesek arra, hogy az angol, a cirill és még néhány más nyelv betűit kódoljuk, de ezeken kívül még rengeteg karakter létezik (görög, arab, kínai stb.). Hogy ezeket a karaktereket is le-  
hessen kódolni, kifejlesztették a **UNICODE**

(**Uniform Character Encoding** – *univerzális karakterkészlet*) szabványt.

A **UNICODE** táblázatban a kódok 17-től 65 536-ig terjedhetnek, ezek segítségével 1 114 112 karakter kódolható, amely kielégíti valamennyi nyelv igényeit. Ahogy minden kódrendszerben, a **UNICODE**-ban is a 0 és 127 közé eső kódok megegyeznek az **ASCII**-kóddal. A kód további részében megtalálhatjuk a különféle nyelvek karaktereit. Például az ukrán ábécé „a” betűjének a kódja 1072, az „i” kódja 1110, a „r” kódja pedig 1169 és így tovább.

Különböző kódtáblák szerkezetének összehasonlítása az 1.2. táblázatban található.



## Különböző kódtáblák szerkezetének összehasonlítása

| Kódtábla     | Karakterek száma | 0 és 127 közötti kódok   | 128 és 255 közötti kódok                      | 255-öt meghaladó kódok |
|--------------|------------------|--|---|------------------------|
| ASCII        | 128              | Angol betűk, számjegyek és elválasztójelek, speciális karakterek | Nincs   | Nincs                  |
| KOI8-U       | 256              | Ahogy az ASCII-ben   | Cirill betűk, más nyelvek különleges betűi    | Nincs                  |
| Windows-1251 | 256              | Ahogy az ASCII-ben   | Cirill betűk, más nyelvek különleges betűi    | Nincs                  |
| UNICODE      | 1 114 112        | Ahogy az ASCII-ben   | Különböző nyelvek betűi, speciális karakterek |                        |

**Ha többet szeretnél tudni**

Az **Excel** táblázatok olyan funkciókkal rendelkeznek, amelyek lehetővé teszik egy adott szimbólum kódjának meghatározását, valamint a megadott kód alapján beazonosítja a megfelelő szimbólumot a **Windows-1251** és a **Unicode** kódtáblázatokban:

- **CODE** – meghatározza a szimbólum kódját a **Windows-1251** kódtáblázatban;
- **UNICODE** – meghatározza a szimbólum kódját a **UNICODE** kódtáblázatban;
- **CHAR** – meghatározza a szimbólum számkódját a **Windows-1251** kódtáblázatban;
- **UNICHAR** – meghatározza a szimbólum számkódját a **UNICODE** kódtáblázatban.

**Ha többet szeretnél tudni**

A **UNICODE** táblázat nemcsak betűket és számjegyeket tartalmaz, hanem egyéb szimbólumokat is: például bejegyzett védjegy jelet (registered trademark), pénznemek jeleit, transzkripció jeleket, ideogramokat is. Az ukrán pénznem, a hrivnya ₴ jelének

kódja például a 8 372, az emberke ideogrammé 🧑 4036002432, a nő ideogrammé 🧑 4036002433 és így tovább (ezen szimbólumok kódja nem egyezik meg a sor-számukkal).

A speciális karaktereket a **UNICODE**-tábla megfelelő számértékeinek ismeretében könnyen beilleszthetjük a dokumentumainkba. E célból tartsuk lenyomva az **Alt** billentyűt, és írjuk be a számbillentyűzetet. Ily módon beilleszthetjük például a hóembert ❄️ (kódja 9 731), a virág alakú szívet 💖 (kódja 10 087), a hangjegyet 🎵 (kódja 9 834), a sakk huszárt ♗ (kódja 9 800), valamint egyéb jeleket. A karaktereket és kódjaikat megtalálhatjuk a **UNICODE**-tábla ([unicode-table.com](http://unicode-table.com)) oldalon. Ha azonban a kiválasztott jelet a számítógépünkön telepített betűtípusok nem tartalmazzák, az nem lesz látható a képernyőnkön.

**Ideogram** (ókori görög ἰδέα – ötlet, γράμμα – írott jel, levél) – olyan írott jel, amely a betűvel ellentétben nem egy adott nyelv hangzását, hanem egy egész szó, fogalom, ötlet közvetítését jelenti.



## Képek kódolása

A színek kódolása fontos a számítógépes grafikában. Különböző szabályok léteznek a színek kódolására, ezek a **színmodellek**.

**Színmodell** alatt a számszerűsített jellemzőket értjük, amelyeket a spektrum színeinek előállítására használnak.

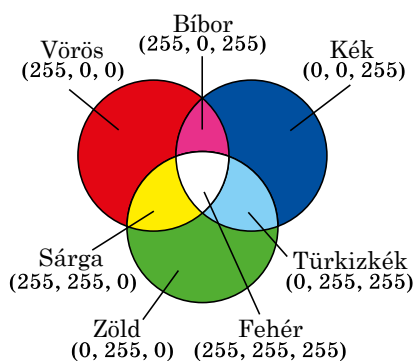
**Spektrum** (a latin *spectrum* – lelki kép, tünemény szóból) – színek, a prizma-val fölbontott fehér fény színsora.

A két leggyakoribb színmodell az **RGB** és a **CMYK**. Ez azért van így, mert ezek a modellek a monitor és a nyomtató színelőállítási módjait írják le.

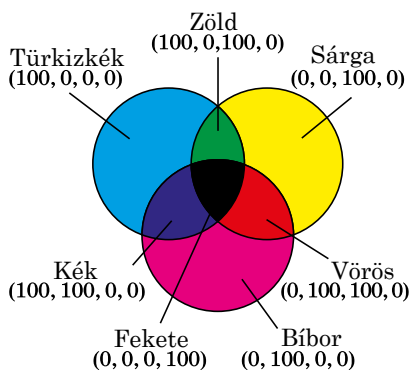
Az **RGB** modell alapszínei (1.6. ábra) a **piros (Red)**, a **zöld (Green)** és a **kék (Blue)**.

A modellben minden egyes komponens intenzitását egy 0 és 255 közötti érték jelöli. Az **RGB** modell segítségével  $256^3 \approx 16,7$  millió szín kódolható.

A **CMYK** modell négy alapszínt tartalmaz: **türkizkék (Cyan)**, a **bíbor (Magenta)**, a **sárga (Yellow)** és a **fekete (Black)** (1.7. ábra). A **CMYK** színeinek arányát az adott szín előállításában százalékban kell megadni (0 és 100 közötti egész számok).



1.6. ábra. Az RGB színmodell



1.7. ábra. A CMYK színmodell

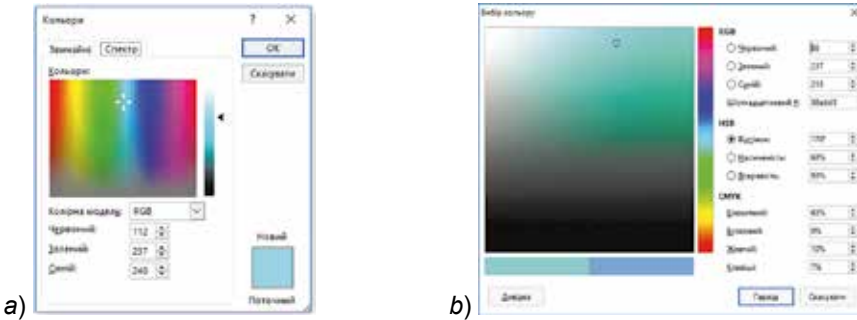
Az 1.3 táblázatban különböző színek kódolása található.

1.3. táblázat

## Példák színek kódolására RGB és CMYK színmodellekben

| Szín      | RGB modell |     |     | CMYK modell |    |    |     |
|-----------|------------|-----|-----|-------------|----|----|-----|
|           | R          | G   | B   | C           | M  | Y  | K   |
| Fehér     | 255        | 255 | 255 | 0           | 0  | 0  | 0   |
| Szürke    | 122        | 122 | 122 | 0           | 0  | 0  | 50  |
| Fekete    | 0          | 0   | 0   | 0           | 0  | 0  | 100 |
| Smaragd   | 80         | 200 | 120 | 60          | 0  | 40 | 22  |
| Rózsaszín | 255        | 128 | 192 | 0           | 25 | 20 | 0   |

Az előbbieken kívül léteznek más színmodellek is, például a **HSB**, melyben minden színt három alapvető összetevővel: **árnyalat (Hue)**, **telítettség (Saturation)** és **fényerő (Brightness)** jellemezzük.



1.8. ábra. A szín kódjának meghatározása a **Word** és az **Excel** (a) használatokor és a **LibreOffice Draw** alkalmazásban (b)

A különböző alkalmazások használatokor lehetőség van a színek kódok megjelenítésére. Például a **Word** és az **Excel** alkalmazásokban az objektumok kitöltésére szolgáló szín kiválasztásakor a színek kódját az **RGB** vagy **HSB** modellben (1.8. a. ábra) és a **LibreOffice Draw**-ban – az **RGB**, **HSB** és **CMYK** modellekben adjuk meg (1.8. b. ábra).

A grafikus adatok kódolása másképp történik, mindig a grafika típusától függően.

Mint már tudjátok, a raszteres képek apró képpontokból, pixelekből állnak. A raszteres kép kódolásához meghatározzuk a benne lévő pixelek számát és az egyes pixelek színét számkészlet kódolja.

A vektorgrafikus kép mértani alakzatokból (grafikai alapelemekből) – szakaszokból, sokszögekből, görbékből, ellipszisekből – épül fel. Minden mértani alakzat rendelkezik koordinátákkal, matematikai képlet segítségével megadható formákkal, a körvonal és a kitöltés értékeivel, amelyek számokkal kódolhatók. A vektorgrafikus kép kódolása a képet alkotó grafikai alapelemek tulajdonságai értékeinek a megadása.

Például egy kör szerkesztéséhez elegendő kódolni a központja helyét, a kör sugarát, a körvonal vastagságát, stílusát és színét, a kitöltés színét.

**Használjuk a számítógépet!**



Végezzék el a gyakorló feladatokat a <https://cutt.ly/phdeK0J> link, vagy a QR-kód segítségével!

**Összefoglalás**

Szöveges üzeneteket úgy dolgozhatunk fel számítógép segítségével, hogy azt előzőleg számok segítségével kódoljuk. E célból például az **ASCII**, **KOI8-U**, **Windows-1251**, **Unicode** kódtáblázatokat használjuk.

Az **ASCII**-vel 128, a **KOI8-U** és **Windows-1251** segítségével 256, a **Unicode**-dal 1 114 112 különböző karaktert kódolhatunk. Valamennyi kódtábla első 128 karaktere megegyezik, ami megfelel az **ASCII**-nek.

**Színmodell** alatt a számszerűsített jellemzőket értjük, amelyeket a spektrum színeinek előállítására használnak.

Amikor a képet a képernyőn szeretnénk megjeleníteni, akkor a leggyakrabban az **RGB** színmodellt alkalmazzuk, ha pedig nyomtatni szeretnénk, akkor a **CMYK** modellt.

A raszteres kép kódolásához meghatározzuk a benne lévő pixelek számát, az egyes pixelek színét számkészlet kódolja.

A vektorgrafikus kép kódolását a képet alkotó grafikai alapelemek tulajdonságai értékeinek a megadása eredményezi.



### Felelj a kérdésekre!

- 1°. Hogyan végezzük a karakterek kódolását számítógépes szövegfeldolgozás során?
- 2°. Milyen karaktertáblákat ismerünk?
- 3°. Milyen karaktereket kódolhatunk **ASCII**, **KOI8-U**, **Windows-1251** és **Unicode** táblák segítségével?
- 4\*. Mi a közös, és mi a különbség az **ASCII**, **KOI8-U**, **Windows-1251** és **Unicode** táblákban?
- 5°. Mi a színmodell?
- 6°. Milyen színmodelleket ismersz? Jellemezd őket!
- 7°. Mi a raszteres (pixeles) kódolás?
- 8°. Mi a vektorgrafikus kép kódolása? Hozz fel példákat!



### Végezd el a feladatokat!



- 1°. Az alábbi üzenetet **ASCII** tábla segítségével kódoltuk. Végezd el a dekódolást (használd az 1. mellékletet vagy a <https://cutt.ly/NkBXn8o> linket):  
a) 83 117 110;                      b) 104 111 109 101!
- 2°. Kódold az előbbi feladatban dekódolt üzenetet **Windows-1251** tábla segítségével (használd a 2. mellékletet vagy a <https://cutt.ly/NkBXn8o> linket):  
a) *Informatika*;                      b) a nevedet!
- 3\*. Az alábbi közlemény egyes szavai a **Unicode** tábla szerint vannak kódolva. Keresd meg a **Unicode** táblában a megfelelő karaktereket és egészítsd ki a szöveget! A szimbólumok kereséséhez használhatod a **Unicode® Character Table** ([unicode-table.com](http://unicode-table.com)) weboldalt:  
*Ma reggel az idő 9925, a meteorológusok 9928-at ígérnek. 127979-be indulás-kor ne felejtsd magaddal vinni 9730-at!*
- 4\*. Határozd meg, milyen karaktereket lehet beilleszteni az **Alt + kód** billentyű-kombinációkkal a szövegedbe, ha a kód értéke 1234, 3754, 8986, 9708, 9415, 10001, 10239!
- 5°. Nyisd meg az **1.fejezet\1.2. pont\1.2.5.xlsx feladat** fájlt! Az **RGB** színmodell színekódjait a táblázat cellái tárolják. Fesd ki a cellákat a kódoknak megfelelő színnel! Állítsd be a színértékeket a kitöltési színösszeállítás ablak **Spektrum** lapján!
- 6\*. A cellák különböző színű színezésével hozz létre egy képet az **1. munkalapon**! Kódold a képet úgy, hogy a kapott kép színes celláinak színekódjait az **RGB** színmodell szerint helyezed el a **2. munkalap** megfelelő celláiban! Mentsd el a feladatot a saját mappádba **feladat 1.2.6.xlsx** néven!

## 1.3. Bináris kódolás



1. Mit értünk az üzenetek kódolása és dekódolása alatt?
2. Milyen kódtáblákat használunk szövegek kódolása során?
3. Mit jelentenek a *kilo*, *mega*, *giga* prefixumok?



## Bináris kód. Bit és Byte

Ahogy arról már beszéltünk, adatok kódolásakor különböző jelrendszereket használhatunk. A nyelv hangjainak kódolása betűk segítségével történik. Az ukrán nyelvben 33, az angolban 26 betűt alkalmaznak. A számok írásakor 10 számjegyre van szükségünk. A gyalogátkelőn a forgalom szabályozását három szín segítségével oldják meg. A morzeábécé csak kétféle jelet használ, a *pontot* és a *vonást*.

A Samuel Morse (1791–1872) amerikai művész által 1844-ben a távirón történő adatátvitel céljából megalkotott, később a tiszteletére elnevezett ábécében a betűket pontokból és vonalakkból álló sorozatokkal kódolták (1.9. ábra).



1.9. ábra. Samuel Morse és a telegráf

Néhány magyar betű morzekódját az 1.4. táblázat tartalmazza.

1.4. táblázat

### Néhány magyar betű morzekódja

| Betű | Morzekód |
|------|----------|
| a    | · –      |
| d    | – · ·    |
| p    | · – – ·  |
| r    | · – ·    |
| t    | –        |

A *pad* szó morzekódja tehát · – – · – – · · lesz.

Amikor az üzenetet két jel segítségével kódoljuk, **bináris kódolás**ról beszélünk. Az ilyen kódolás eredményeként előálló kódot **bináris**nak nevezzük.

A számítógépekben is bináris kódolást használunk. A számítógépekben valamennyi információt **0** és **1** számjegyek sorozataként kell előállítani.

A 0 és 1 számjegyeket a bináris kódban biteknek (az angol *binary digit* – *bináris szám*) nevezzük.



Az 1.4. táblázatból láthatjuk, hogy a  $t$  betű kódolásához egyetlen jel szükséges, egy vonal, az  $a$  betű kódolásához kettő, a  $p$  kódolásához pedig négy. A távírókban az egyes betűket hosszabb szünetekkel választották el egymástól. A számítógépekben ez a módszer nem igazán használható jól. Kényelmesebb minden betű kódolásához ugyanannyi jelet használni.

Ha egy üzenet kódolására egyetlen bitet (amelynek értéke 0 és 1 lehet) használunk, akkor a következő üzeneteket áll módunkban kódolni:

- egy állítás *igaz* (1) vagy *hamis* (0);
- egy ember neve *férfi* (1) vagy *nő* (0);
- egy kapcsoló *bekapcsolt* (1) vagy *kikapcsolt* (0) állapotban van stb.

Két bitből  $2^2 = 4$  kódot készíthetünk (00, 01, 10, 11). Ezekkel kódolhatjuk például az égtájakat: 00 – észak, 01 – kelet, 10 – dél, 11 – nyugat.

Ha három bitünk van, akkor a kódok száma  $8 (2^3)$  (000, 001, 010, 011, 100, 101, 110, 111), ezek egy sakktabla sorának vagy oszlopának kódolására lehetnek alkalmasak.

4 bitből 16 ( $2^4$ ), ötből pedig 32 ( $2^5$ ) kód állítható elő és így tovább.

Nyolc bitből  $2^8 = 256$  kód állítható elő, ami elég az angol és az ukrán (magyar stb.) betűk, továbbá a számjegyek és néhány különleges karakter kódolásához. Pontosan ennyi kódot tartalmaz a **Windows-1251** tábla.

A nyolc bitet tartalmazó sorozatot **bájtnak** (régebben *byte*-nak írtuk) nevezzük.

$$1 \text{ bájtn} = 8 \text{ bit.}$$

## A bináris kód hossza

Ha egy üzenet minden karakterét 8 biten kódoljuk, akkor az egy karakter kódolásához 8 bitet, tehát 1 bájtot használunk.

Egy **üzenet bináris kódjának hossza** az üzenet kódolásához szükséges bájtok száma. Az *informatika érdekes!* üzenet például a szóközökkel és felkiáltójellel együtt 23 karaktert tartalmaz. Ha minden karaktert egy bájttal kódolunk, akkor a hossza 23 bájtn lesz.

Azt már tudjuk, hogy a **Windows-1251** táblázatban minden karakternek egy 0 és 255 közötti szám felel meg. Ezeket a számokat egy bájton kódolhatjuk. Az 1.5. táblázat ilyen kódolásokat mutat be.

1.5. táblázat

Bináris kódok

| Karakter | Szám kód | Bináris kód |
|----------|----------|-------------|
| !        | 33       | 00100001    |
| @        | 64       | 01000000    |
| W        | 103      | 01100111    |
| ю        | 254      | 11111110    |
| я        | 255      | 11111111    |

A **Windows-1251** táblában tehát minden karaktert egy bájton ábrázolunk.

Ha tehát egy szöveg bináris kódját szeretnénk előállítani, minden karakter kódját ki kell cserélni annak bináris megfelelőjére. Ily módon kódolja a karaktereket a



**Jegyzetömb**, amely a mentéskor **ANSI** (*American National Standards Institute*) kódolást alkalmaz. Az ily módon mentett fájlban a bájtok száma megegyezik a betűk számával. Fontos megjegyezni, hogy az **Enter** billentyű leütése két bájtnyi vezérlő kódot eredményez, az egyik a sor elejére ugrás, a másik pedig az új sor jele.

Szövegek kódolásakor a **Unicode** kódtáblázatban a különböző karakterek bináris kódja eltér – 1 byte, 2 byte, 4 byte.

A számítógéppel feldolgozott grafikai, audio- és videoinformációt szintén binárisan kell kódolni.

Az üzenet bináris kódjának a hosszát **adatmennyiségnek** is nevezik.

### A bináris kód hosszának többszörös mértékegységei

A bináris kód hosszának kifejezésére a *kilo*, *mega*, *giga* és *tera* prefixumok segítségével többszörös mértékegységeket alkottak. Ezeket a prefixumokat megtalálhatjuk a SI mértékegységrendszerben. Ugyanitt megtalálhatjuk a megfelelő szorzószámokat is. Történelmileg úgy alakult, hogy ezek a prefixumok az informatikában kicsit mást jelölnek, mint a fizikában. Hatodik osztályban megtanultátok, hogy:

$$1 \text{ kbájt (kilobájt)} = 2^{10} \text{ bájt} = 1024 \text{ bájt}$$

$$1 \text{ Mbájt (megabájt)} = 2^{10} \text{ kbájt} = 2^{20} \text{ bájt} = 1\,048\,576 \text{ bájt};$$

$$1 \text{ Gbájt (gigabájt)} = 2^{10} \text{ Mbájt} = 2^{20} \text{ kbájt} = 2^{30} \text{ bájt}$$

$$1 \text{ Tbájt (terabájt)} = 2^{10} \text{ Gbájt} = 2^{20} \text{ Mbájt} = 2^{30} \text{ kbájt} = 2^{40} \text{ bájt}.$$

Ez azért alakult így, mert a számítógépekben bináris kódokat használunk, amelyben kényelmesebb a 2 hatványaival számolni, mint a 10 hatványaival. Mivel pedig  $2^{10} = 1024$  elég közel van az 1000-hez, ezért a  $2^{10}$  bájtot nevezték el **kilobájtnak**. Hasonlóképpen a  $2^{10}$  kilobájtból lett az 1 **megabájt** és így tovább.

A bináris kódhossz egységeket a fájlméretek, az adathordozók kapacitásának, a hálózatokon keresztül továbbított adatmennyiségek feltüntetésében, valamint az adatsebességek mérésében használják.



#### Ha többet szeretnél tudni

1999-ben a Nemzetközi Elektrotechnikai Bizottság új szabályt vezetett be a bináris prefixumok megnevezésére. E szabály szerint 1 kB = 1000 byte és az 1024 byte új nevet kapott – 1 **kibibájt** (1.6. táblázat).

1.6. táblázat

#### A prefixumok (előtagok) és jelentésük, a Nemzetközi Elektrotechnikai Bizottság által jóváhagyva

| Byte-ok száma       |           |           |                    |           |          |
|---------------------|-----------|-----------|--------------------|-----------|----------|
| Prefixumok a SI-ben |           |           | Bináris prefixumok |           |          |
| Név                 | Rövidítés | Érték     | Név                | Rövidítés | Érték    |
| kilobájt            | kB        | $10^3$    | kibibájt           | KiB       | $2^{10}$ |
| megabájt            | MB        | $10^6$    | mebibájt           | MiB       | $2^{20}$ |
| gigabájt            | GB        | $10^9$    | gibibájt           | GiB       | $2^{30}$ |
| terabájt            | TB        | $10^{12}$ | tebibájt           | TiB       | $2^{40}$ |

A *kilo*, *mega*, *giga*, *tera* és más prefixumok hagyományos értékére való áttérés a bináris kód hosszának mérésekor meglehetősen lassú.

A tankönyv további részében  $1 \text{ kbyte} = 1024 \text{ byte}$ .

### Átváltás a bináris kód hosszának mértékegységei között

A bináris kód hosszát bitekben, bájtokban és azok többszöröseiben fejezhetjük ki. Lássunk néhány feladatot, amelyben megvizsgáljuk, hogyan történik az átváltás ezek között a mértékegységek között!

**1. feladat.** A bináris kód hossza 4,5 Mbájt. Fejezd ki ezt bájtokban! Hány karaktert kódolhatunk a **Windows-1251** kódolásban?

**Megoldás.** Az átalakítás során előbb alakítsuk át a kód hosszát kilobájtokra. Mivel  $1 \text{ Mbájt} = 1024 \text{ kbájt}$ ,  $4,5 \text{ Mbájt} \cdot 1024 = 4608 \text{ kbájt}$ .

Figyelembe véve, hogy  $1 \text{ kbájt} = 1024 \text{ bájt}$ :

$4608 \text{ kbájt} \cdot 1024 = 4\,718\,592 \text{ bájt}$ .

**Felelet:**  $4,5 \text{ Mbájt} = 4\,718\,592 \text{ bájt}$ . Mivel egy karakter kódjának a hossza egy bájt a **Windows-1251** táblában, így  $4\,718\,592$  karakter kódolható.

**2. feladat.** A bináris kód hossza  $4\,194\,304\,000$  bit. Fejezd ki a kódot egész számmal a lehető legnagyobb mértékegységben!

**Megoldás.** Fejezzük ki a kód hosszát bájtokban, figyelembe véve, hogy  $1 \text{ bájt} = 8 \text{ bit}$ :

$4\,194\,304\,000 \text{ bit} : 8 = 524\,288\,000 \text{ bájt}$ .

Az eredményünk meghaladja az 1024-et, amivel a bájtokat kilobájtokká alakíthatjuk, ezért áttérhetünk kilobájtokra:

$524\,288\,000 \text{ bájt} : 1024 = 512\,000 \text{ kbájt}$ .

Az előzőhöz hasonló gondolatmenet mentén áttérhetünk megabájtokra:

$512\,000 \text{ kbájt} : 1024 = 500 \text{ Mbájt}$ .

Mivel a kapott szám kevesebb 1024-nél, magasabb mértékegységre már nem térhetünk át oly módon, hogy az eredmény egész szám legyen.

**Felelet:**  $4\,194\,304\,000 \text{ bit} = 500 \text{ Mbájt}$ .

**3. feladat.** Mérd fel az informatika tankönyv kódjának közelítő mértékét! Az illusztrációk kódjának hosszát hagyd figyelmen kívül!

**Megoldás.** A könyv szövegét **Windows-1251**-ben kódoltuk, ahol egy karakter kódjának hossza 1 bájt.

A szöveg egy sora átlagosan 60 karaktert tartalmaz, vagyis egy sor bináris kódja 60 bájt. Oldalanként átlagosan 50 sort találunk, tehát egy oldalnyi szöveg  $50 \cdot 60 = 3000$  bájt. A könyv nagyjából 250 oldalt tartalmaz, az ehhez tartozó kód hossza nagyjából  $3000 \cdot 250 = 750\,000$  bájt. Váltssuk ezt át egy nagyobb mértékegységre:

$750\,000 \text{ bájt} : 1024 \approx 732,4 \text{ kbájt}$ .

**Felelet:** a könyv szövegének kódja  $732,4 \text{ kbájt}$ .

**Használjuk a számítógépet!**

Végezzék el a gyakorló feladatot a <https://cutt.ly/whdeZob> link, vagy a QR-kód segítségével!

**Összefoglalás**

A kódolásnak azt a módját, amikor csak kétféle jelet használunk, **binárisnak** nevezük. Az így keletkezett kód a **bináris kód**.

A bináris kód 0 vagy 1 jegye az 1 **bit**.

A 8 bitből álló kódsorozat az 1 **bájt**:

$$1 \text{ bájt} = 8 \text{ bit.}$$

**Az üzenet kódjának hossza** alatt a kódja bájtokban kifejezett hosszát értjük.

A **Windows-1251** kódtáblában minden karakternek 1 bájtnyi kód felel meg.

1 kbájt (kilobájt) =  $2^{10}$  bájt = 1024 bájt;

1 Mbájt (megabájt) =  $2^{10}$  kbájt =  $2^{20}$  bájt = 1 048 576 bájt;

1 Gbájt (gigabájt) =  $2^{10}$  Mbájt =  $2^{20}$  kbájt =  $2^{30}$  bájt;

1 Tbájt (terabájt) =  $2^{10}$  Gbájt =  $2^{20}$  Mbájt =  $2^{30}$  kbájt =  $2^{40}$  bájt.

**Felelj a kérdésekre!**

- 1°. Mit értünk bináris kódolás alatt?
- 2°. Mi a bit?
- 3°. Mi a bájt?
- 4°. Mit értünk az üzenet kódjának hossza alatt?
- 5°. Milyen hosszú egy karakter kódja a **Windows-1251** táblában?
- 6°. Mekkora lesz a **Jegyzetömbben** létrehozott szöveg bináris kódja, ha az **ANSI** kódban mentettük? És ha **UTF-8** kódban?
- 7\*. Mennyiben fog különbözni annak a **Jegyzetömbben** létrehozott fájlnek a tartalma, amely a nevedet tartalmazza attól a fájlától, amely a neved betűit soronkénti tördelésben tartalmazza?
- 8°. Mi az 1 kbájt, 1 Mbájt, 1 Gbájt és 1 Tbájt?
- 9°. Milyen matematikai műveletet kell végrehajtani a következő átalakítások során:
  - a) biteket bájtokká alakítunk;
  - b) bájtokat kilobájtokká alakítunk;
  - c) megabájtokat kilobájtokká alakítunk?

**Végezd el a feladatokat!**

- 1°. Hozz fel példát a mindennapi életben alkalmazott bináris kódolásra!
- 2°. Kódold a morzeábécé segítségével a *kikötő*, *társaság*, *sütemény* szavakat! Milyen hosszúak lesznek a keletkező kódok? (*Megjegyzés*: Használd a morzetáblázatot!)



3°. Határozd meg a következő üzenetek bináris kódjának hosszát abból a feltételzésből kiindulva, hogy **Windows-1251** kódlappal történt a kódolás:

a) a *Windows-1251* kódtábla;

b) a morzeábécé két szimbólumot használ – a pontot és a vonalat!



4°. Határozd meg az alábbi mondat kódjának hosszát, ha azt **Jegyzettömbben** állítottuk elő! Vizsgáld meg a különböző kódolásokat!

*Az ember barátok nélkül olyan, mint a fa gyökerek nélkül.*

5°. Töltsd ki az üres helyeket:

a) 5 bájt = \_\_\_\_\_ bit;

c) 10 kbájt = \_\_\_\_\_ bit;

b) 4096 bájt = \_\_\_\_\_ kbájt;

d) 3 Mbájt = \_\_\_\_\_ bájt!



6°. Töltsd ki az üres helyeket:

a) 2 bájt = \_\_\_\_\_ bit;

c) 2 kbájt = \_\_\_\_\_ bit;

b) 8192 bájt = \_\_\_\_\_ kbájt;

d) 100 Mbájt = \_\_\_\_\_ kbájt!

7\*. A szöveg egy sora 60 karaktert tartalmaz, egy oldalon 40 sor fér el. Hány oldalnyi lehet a szöveg, ha a kód hossza nem haladhatja meg a 8 Mbájtot feltételezve, hogy **Windows-1251** kódtáblát használtunk?

8\*. A helyi hálózat adatátviteli sebessége 100 megabit másodpercenként. Mennyi időbe telik, amíg ez a hálózat videofájlt küld, ha bináris kódja 4 GB hosszú?

## 1. SZÁMÚ GYAKORLATI MUNKA

### Feladatok megoldása, a szöveges kód hosszának meghatározása

**Figyelem!** A számítógép használata során tartsd be a balesetvédelmi szabályokat és az egészségügyi előírásokat!

1. Határozd meg a szöveg bináris kódjának a hosszát, ha **Windows-1251** kódolást alkalmaztunk, a szöveg 32 oldalas, oldalanként 48 sornyi szöveget tartalmaz, egy sorban átlagosan 56 karakter helyezkedik el. A kód hosszát a legnagyobb, még egész számokkal kifejezhető mértékegységben határozd meg!

2. A **Jegyzettömb** segítségével határozd meg, mennyi az 1. számú gyakorlati munka üzenet kódjának hossza, ha:

a) **Windows-1250** (ANSI kódolás);

b) **Unicode (UTF-8-as kódolás)**!

A fájlokat **gyakorlati 1\_Windows.txt** és **gyakorlati 1\_unicode.txt** néven mentsd el!

3. Töltsd ki a táblázatkezelőben a következő táblázatot!

| Bit     | Bájt    | kbájt | Mbájt |
|---------|---------|-------|-------|
| 614 400 |         |       |       |
|         | 524 288 |       |       |
|         |         | 256   |       |
|         |         |       | 4     |

A fájlt mentsd el **gyakorlati 1.xlsx** néven!

## 2. fejezet

# A számítógép hardvere

*Ebben a fejezetben megismerkedtek:*

- a személyi számítógéppel
- a számítógép hardverelemeinek rendeltetésével és főbb jellemzőivel:
  - a processzorral
  - a belső és külső memóriával
  - adatbeviteli és -kiviteli eszközökkel
  - a multimédiás eszközökkel
- a számítástechnikai eszközök fejlődésével
- korszerű számítógépek típusaival és azok alkalmazásával

### 2.1. A személyi számítógép felépítése

1. Mik a számítógép főbb részei? Mi ezek rendeltetése?
2. A számítógép melyik eleme végzi az adatok feldolgozását? Hol található meg ezt az elemet a számítógépben?
3. Nevezd meg néhány információs folyamatot! A számítógép melyik eleme végzi az adatok tárolását?

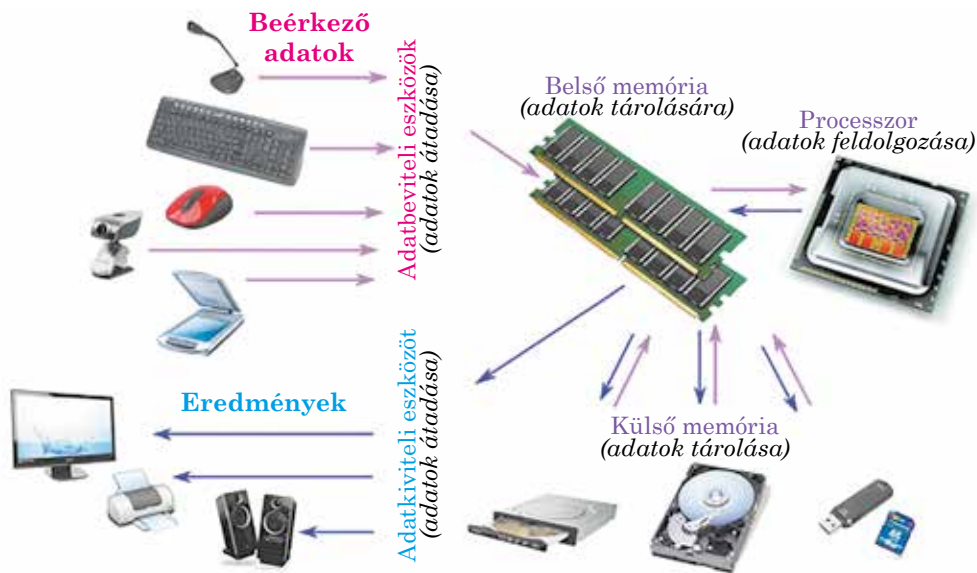
#### *A számítógép és az információs folyamatok*

Az ötödik osztályban már megismerkedtünk a számítógép főbb részeivel. Most ennél részletesebben megismerkedünk a számítógépben lezajló információs folyamatokkal, és a számítógép részeinek ebben játszott szerepével (2.1. ábra). Az adatok a **beviteli eszközökről** (billentyűzet, egér, mikrofon, szkennel stb.) vagy a **külső memóriából** (merevlemez, optikai tároló, pendrive, memóriakártya stb.) kerülnek a **belső memóriába**.

A memóriából az adatok a **processzorba** kerülnek, innen a feldolgozás után visszajutnak a **belső memóriába**, majd a **külső memóriába** mentődnek, vagy a kiviteli eszközökön (monitor, nyomtató, hangfalak stb.) kerülnek megjelenítésre.

Mindezen műveletek egy-egy információs folyamattal kapcsolhatók össze, amelyeket a megfelelő szoftverek vezérelnek.

**Hardver** – a számítógép kézzel fogható részei.



2.1. ábra. A számítógépben zajló információs folyamatok vázlata

### A processzor és annak rendeltetése

Azt már tudjátok, hogy a számítógépben a processzor (2.2. ábra) végzi az adatok feldolgozását. Ez az eszköz biztosítja a programok futását a számítógépben. A processzor, mint minden algoritmus-végrehajtó, saját utasításkészlettel rendelkezik.

A processzor a számítógép egyik legfontosabb része. Tartalmazza a programok értelmezését és végrehajtását lehetővé tevő **vezérlőegységet**, valamint a matematikai műveletek elvégzéséhez elengedhetetlen **aritmetikai-logikai egységet**. A mai processzorok már saját belső memóriával – **cache-memória** – is rendelkeznek. Ennek rendeltetésére később még visszatérünk.



2.2. ábra. 8 magvas AMD Ryzen 7 processzor

A processzor legfőbb jellemzője a **műveleti sebesség**, amely az **órajel frekvenciájától**, a **magok számától**, a **sínszélességtől** és a **cache-memória méretétől** függ (2.1. táblázat). Ezen tulajdonságok értékei határozzák meg a processzor **adattfeldolgozási sebességét**.

2.1. táblázat

#### Az asztali gép és a táblagép processzorainak néhány jellemzője

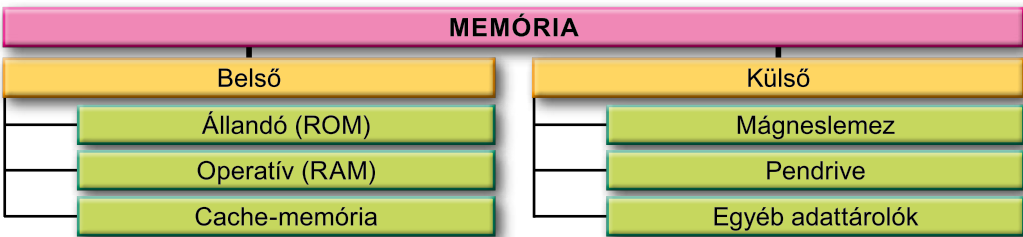
| Tulajdonság | Mit jellemez?                   | Jellemző érték               |          |
|-------------|---------------------------------|------------------------------|----------|
|             |                                 | Asztali gép                  | Táblagép |
| Modell      | A gyártó által adott megnevezés | AMD Ryzen Threadripper 3990X | KIRIN970 |



| Tulajdonság         | Mit jellemez?  | Jellemző érték |          |
|---------------------|--|----------------|----------|
|                     |  | Asztali gép    | Táblagép |
| Órajel frekvenciája | A processzor részeit összehangoló vezérlőjelek frekvenciája    | 4,3 GHz        | 2,36 GHz |
| Magok száma         | Az egy eszközbe integrált azonos szerkezetű processzorok száma | 64             | 8        |
| Sínszélesség        | A processzor által egyidejűleg feldolgozott bináris kód hossza | 64 bit         | 64 bit   |

### A számítógép memóriája. Belső memória

A memória rendeltetése az adatok tárolása. A memóriát **belső** és **külső memóriára** oszthatjuk (2.3. ábra). A memória legfőbb jellemzője a kapacitás, amit bájtokban, kilobájtokban, megabájtokban, terabájtokban fejezünk ki.



2.3. ábra. A memória típusai

A belső memória biztosítja a számítógép működését. A processzorhoz hasonlóan az alaplapon helyezkedik el (2.4. ábra).

Az **operatív tár** (RAM) a modern számítógép elengedhetetlen része, itt tároljuk a processzor által feldolgozandó adatokat.

A processzor csak az operatív tárból képes utasításokat és adatokat fogadni. A processzor által feldolgozott adatok ugyanide íródnak vissza. Ahhoz tehát, hogy a processzor feldolgozhassa az adatokat, azokat át kell adni (be kell tölteni) az operatív tárba az adatbeviteli eszközökről vagy a külső memóriából.

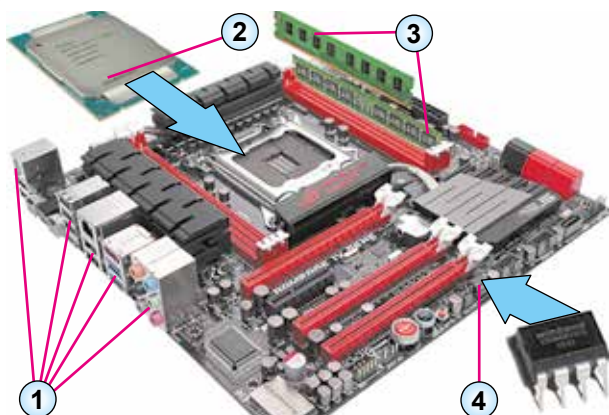
A számítógép kikapcsolásakor az operatív tár tartalma elvész.

Az operatív tár modulokba szervezett integrált áramkörökből áll, amelyeket az alaplapon speciális illesztőhelyeire építünk be (2.4. 3. ábra). Egy számítógép általában több ilyen modult tartalmaz.

**Alaplap** (ang. *motherboard* vagy *mainboard*) – a modern személyi számítógép elengedhetetlen része, biztosítja az adatátvitelt a számítógép részei között.

**Csatlakozó** (ang. *slot*) – rés, nyílás.  
**Illesztőhely** (ang. *socket*) – mélyedés, csatlakozó aljzat.





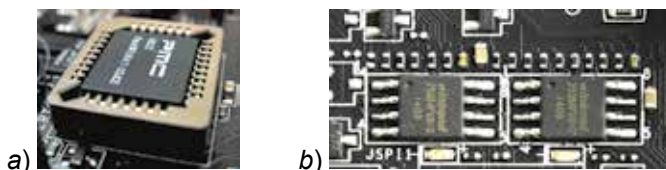
1. Adatbeviteli és -kiviteli eszközök csatlakozói
2. Processzor
3. RAM
4. ROM

2.4. ábra. A számítógép alaplapja

Ahogy már említettük, a mai számítógépeknek része a **cache-memória** is (cache – angolul *rejtekhelyet, titkos raktárat* jelent). Rendeltetése, hogy meggyorsítsa az adatátvitelt a RAM és a processzor között. A cache-nek három szintjét különböztetik meg. Az első szintű cache kapacitása a legtöbb processzornál 128 kb-át. A második és harmadik szintű cache kapacitása az egyes processzortípusok esetében jelentősen eltérhet egymástól. 2020-ban az asztali gépek esetében a második szintű cache kapacitása 1-8 Mb-át, a harmadik szintűé pedig 2-20 Mb-át. A cache kapacitásának növelése általában a processzor sebességének növekedésével jár.

Ahhoz, hogy a számítógépet működésbe hozzassuk, szükség van **állandó memóriára** (ROM) is. A számítógép bekapcsolása után az indításhoz elengedhetetlen programok betöltődnek a ROM-ból a RAM-ba, majd a processzor végrehajtja azokat.

Az állandó memória egy vagy néhány integrált áramkörből, IC-ből áll, amelyeket az alaplap speciális csatlakozóiba (2.5. a. ábra) vagy közvetlenül az alaplapra (2.5. b. ábra) illesztünk.



2.5. ábra. RAM IC az alaplapokon

## Külső tárolóeszközök

A belső tárolóktól (ez alól a ROM kivétel) eltérően a külső tárolóeszközök rendeltetése az adatok tartós tárolása. A belső memóriáktól nemcsak rendeltetésükben különböznek, hanem egyes tulajdonságaik értékeiben is:

- megnövelt adattároló (a személyi számítógépeknél 25 TB-ig);
- a számítógép kikapcsolása után a külső tárolók tartalma nem vész el (energiafüggetlen);
- lehetőség van különböző külső adattárolók hozzákapcsolására a számítógéphez, a számítógép kikapcsolása nélkül.

A merevlemezek és az elektronikus áramkörök a külső adattárolás fő adathordozói. Korábban a számítógépekhez hajlékonylemezeket, optikai lemezeket és mágneses szalagokat használtak adathordozóként.

A külső tárolóeszközök különbözőek lehetnek. A **merevlemez** egy vagy néhány mágneses anyaggal bevont fémlmezből áll (2.6. ábra). A lemezre egy elektromágneses írófej segítségével írónak fel az adatok, amely felmágnesezi a lemez egyes részeit. Ugyanez az eszköz olvassa le az adatokat a lemezről.



1. Lemezfelület
2. Elektromágneses író- és olvasófej

2.6. ábra. A merevlemez felépítése

A 2.2. táblázatban a mai merevlemezek legfontosabb paramétereit és azok jellemző értékeit foglaltuk össze.

2.2. táblázat

**A merevlemez jellemzői**

| Tulajdonság  | Jellemző érték        |                       |
|--------------|-----------------------|-----------------------|
|              | Asztali               | Notebookok            |
| Kapacitás    | 1-16 TB               | 0,5-5 TB              |
| Átmérő       | 3,5 inch              | 2,5 inch              |
| Fordulatszám | 5400-15 000 ford/perc | 5400-10 000 ford/perc |

Az elektronikus chipen lévő külső memóriát flash memóriának is nevezik.

Három fő típusát különböztetjük meg:

- pendrive;
- memóriakártya;
- szilárdtest, vagy SSD meghajtó.

A pendrive-okat elsősorban a különböző számítógépek közötti adatcserére használják. **USB-csatlakozókkal** csatlakoztathatók a számítógéphez. A legtöbb pendrive kapacitása 4-128 GB, bár vannak olyan eszközök, amelyek sokkal nagyobb kapacitással rendelkeznek, akár 2 TB-al.

A különböző **memóriakártyákat** (2.7. ábra) többnyire okostelefonok, fotó- és videokamerák memóriaeszközeként használják. A memóriakártya típusától függően a megfelelő csatlakoztatókat használják ezek beolvasásához. Napjainkban a legelterjedtebbek az okos-



2.7. ábra. Különböző típusú flash-memóriák

telefonokban használt **microSD** kártyák. A memóriakártya kapacitása 4 és 512 GB között mozog.

**A szilárdtest vagy SSD-meghajtóknak** (2.7. ábra) ugyanaz a szerepük, mint a merevlemezeknek. Leginkább a hordozható (laptopok, táblagépek) számítógépek fő külső

**SSD** (ang. **Solid-State Drive**) – szilárdtest (félvezető) alapú meghajtó.

memóriaeszközeiként használják. A merevlemezekhez képest az SSD-k nagyobb adatátviteli sebességgel, ugyanakkor ma-

gasabb költséggel rendelkeznek ugyanazon kapacitás mellett. A maximális kapacitás szempontjából az SSD-k továbbra is alacsonyabbak (legfeljebb 8 TB) a merevlemez-meghajtóknál.



2.8. ábra. SSD-meghajtó



2.9. ábra. A külső memória csatlakoztatása

A számítógép sebességének növelése érdekében az adatcsere sebességének növelésével számos laptop két külső memóriaeszközzel van felszerelve – egy „gyorsabb” SSD-vel (a számítógép által gyakran használt programok, például az operációs rendszer betöltésére), és jelentősen „lassúbb” merevlemez-meghajtóval, ami biztosítja a szükséges memória kapacitását.

A külső merevlemez-meghajtók vagy SSD-k felhasználhatók adatok biztonsági mentésére, nagy mennyiségű bizalmas adat tárolására, nagy mennyiségű adat átadására számítógépes hálózatok használata nélkül és egyéb célokra is. Ellentétben a belső eszközökkel, amelyeket a bővíthelyeken keresztül csatlakoztatnak az alaplaphoz és a mobil eszközök rendszeregységébe helyezik, a külső eszközöket USB-csatlakozók segítségével csatlakoztatják a számítógéphez (2.9. ábra).



2.10. ábra.  
Az IBM 350 Disk File behelyezése a repülőgép csomagterébe



### Tudtad-e, hogy...?

Az egyik első merevlemez-meghajtót az IBM fejlesztette ki és 1956-ban került értékesítésre. Ezt az eszközt **IBM 350 Disk File**-nek nevezték, mindössze 5 MB kapacitású volt, 50 fémlmezből állt, amelyet mágneses anyagréteg borított. Az átmérője 24 inch (60,96 cm) volt, a készülék teljes súlya meghaladta az 1 tonnát (2.10. ábra).



**Tudtad-e, hogy...?**

A kiemelkedő tudós, **Ljubomir Romankiv** (szül. 1931 04. 17., Zsovka városában, Lemberg megyében) jelentős mértékben hozzájárult a modern mágneses-lemezmeghajtók technológiájának létrehozásához és fejlesztéséhez. Az ukrán származású amerikai tudós az IBM vezető munkatársa volt. Kidolgozta a mágneslemezek felületéről történő adatok írására és olvasására szolgáló olvasó- és írófejek létrehozásának technológiáját, amely jelentősen csökkentette a merevlemezek gyártásának költségeit. Ez a technológia lehetővé tette olyan merevlemezek gyártását, amelyek elérnek a számítógép rendszeregységében. Ljubomir Romankiv több mint 60 szabadalmaztatott találmány társszerzője.



2.11. ábra.  
Ljubomir Romankiv



**Ha többet szeretnél tudni**

Az optikai adattárolókban az adatok rögzítése és olvasása lézerek segítségével történik. Ezek a lemezek kapacitásukban és újraírhatóságukban különböznek egymástól. A 2.3. táblázatban az optikai lemezek típusait és azok tulajdonságait foglaltuk össze.

2.3. táblázat

**Az optikai lemezek típusai**

| Rendeltetése   |                |                 | Kapacitása |
|----------------|----------------|-----------------|------------|
| Csak olvasható | Egyszer írható | Többször írható |            |
| CD-ROM         | CD-R           | CD-RW           | 640-800 MB |
| DVD-ROM        | DVD-R, DVD+R   | DVD-R, DVD+R    | 9,4 GB     |
| BD-ROM         | BD-R           | BD-RE           | 100 GB     |

Az **optikai adattárolókkal** való munka során az adatok írása és olvasása lézersugárral történik.



**Használjuk a számítógépet!**



Végezzétek el a gyakorló feladatokat a <https://cutt.ly/5hdeX36> link, vagy a QR-kód segítségével!



**Összefoglalás**

A számítógép legfontosabb részei közé tartoznak azok az eszközök, amelyek az **adatbevitelt** és **-kivitelt**, **adatfeldolgozást** és **tárolást** végzik.

A **processzor** az az eszköz, amely az adatok feldolgozását végzi. Tartalmaz egy **vezérlőegységet**, amely biztosítja az utasítások értelmezését és végrehajtását, valamint egy **aritmetikai-logikai egységet** a számítások elvégzése céljából. A processzor legfontosabb tulajdonsága a **műveleti sebesség**, amely az **órajel frekvenciájától**, a **magok számától**, a **sinszélességtől** és a **cache kapacitásától** függ (2.1. táblázat). Ezen tulajdonságok értékei határozzák meg a processzor **adatfeldolgozási sebességét**.

Az adatok tárolása a memória feladata, amit **belső** és **külső memóriára** osztunk.

A **belső adattárolók** (operatív tár, RAM, cache) az alaplapon helyezkedik el. Ezekből a memóriákból az adatok a számítógép kikapcsolásakor törlődnek, az állandó memóriákból viszont nem.

A **külső adattárolók** (memóriák) közé soroljuk a merevlemez, az optikai lemezekeket, a flash-memóriát. Rendeltetésük az adatok tartós tárolása. Ezekben a tárolókban az adatok a számítógép kikapcsolása után is megmaradnak.

A memória legfontosabb tulajdonsága a kapacitás, amelyet kbájtokban, Mbájtokban, Gbájtokban és Tbájtokban stb. fejezhetünk ki.



### Felelj a kérdésekre!

- 1°. Foglald össze a számítógépben zajló információs folyamatokat a 2.1. ábra alapján!
- 2°. Mi a processzor rendeltetése? Milyen részei vannak a processzornak?
- 3°. Milyen memóriákat találhatunk egy modern számítógépben?
- 4°. Nevezd meg a processzor legfontosabb paramétereit és azok jellemző értékeit!
- 5°. Ismertesd a memóriák típusait! Mi a memória legfontosabb jellemzője? Miben méri a memória kapacitását?
- 6°. Mi a RAM rendeltetése? Hol helyezkedik el a RAM?
- 7°. Mi az állandó memória (ROM) rendeltetése? Miben különbözik ez a RAM-tól?
- 8°. Mi a cache rendeltetése?
- 9°. Nevezd meg, milyen eszközök tartoznak a külső memóriához! Sorold fel ezek legfontosabb paramétereit és azok jellemző értékeit!
- 10\*. Mi a közös bennük, és miben különböznek egymástól a külső memóriák típusai?



### Végezd el a feladatokat!

- 1°. Keresd meg az interneten (például a **hotline.ua** oldalon) a modern számítógépek jellemzőit, válassz ki három olyan processzortípust, amely az utóbbi időben került forgalomba Ukrajnában, majd töltsd ki a táblázatot!

| Tulajdonság       | Értékek |  |  |
|-------------------|---------|--|--|
| Modell            |         |  |  |
| Órajel frekvencia |         |  |  |
| Magok száma       |         |  |  |
| Sínszélesség      |         |  |  |



- 2\*. Keresd meg az interneten (például a **hotline.ua** oldalon) a modern számítógépek jellemzőit, majd határozd meg, hogy maximálisan mekkora RAM-ot ajánlanak a notebookokhoz! Töltsd ki a táblázatot!

| Tulajdonság | Értékek |  |  |
|-------------|---------|--|--|
| Modell      |         |  |  |
| Gyártó      |         |  |  |



| Tulajdonság | Értékek |  |  |
|-------------|---------|--|--|
| Kapacitás   |         |  |  |
| Ár          |         |  |  |

- 3\*. Derítsd ki, hogy az otthoni számítógépednek milyen a processzora és a memóriája! Tölts ki egy táblázatot a **Használjuk a számítógépet!** rubrikában használtakhoz hasonlóan (<https://cutt.ly/5hdeX36>)!
- 4\*. Az internet segítségével keresd meg a környezeted számítástechnikai szakszerveleteinek katalógusait, és határozd meg azoknak a processzoroknak a paramétereit, amelyeket ezekben forgalmaznak a 2.1. táblázatnak megfelelően!
- 5\*. Az internet és a szakszerveletek katalógusai alapján hasonlítsd össze az azonos magszámú és órajelű processzorok másodsztívu cache-memóriáinak kapacitáseit!
- 6\*. Készíts rövid ismertetőt a ROM-ok cseréjének lehetőségeiről!

## 2.2. Adatbeviteli eszközök

1. Milyen eszközök kapcsolhatók egy számítógéphez? Hogyan lehet őket csoportosítani?
2. Milyen beviteli eszközök kapcsolhatók egy számítógéphez? Milyen típusú adatok bevitelére szolgálnak ezek?
3. Milyen műveletek végezhetőek el az egérrel?

### A beviteli eszközök típusai

Ahogy azt már tudjátok, a számítógépnek vannak az adatok bevitelére, illetve kivitelére szolgáló eszközei.

A beviteli eszközöket a bevitt adatok típusa szerint csoportosíthatjuk. Ennek alapján a beviteli adatok szolgálhatnak:

- szöveges adatok bevitelére (**billentyűzet**);
- grafikai adatok bevitelére (**szkenner, fényképezőgép, digitális rajztábla**);
- audioadatok bevitelére (**mikrofon, digitális hangrögzítő**);
- videoadatok bevitelére (**videokamera, webkamera, TV-tuner**).

A beviteli eszközök egy másik csoportja különféle programok vezérlésére szolgál. Ezek közé tartozik az **egér**, a **touchpad**, az **elektronikus multimédiás tábla**, az **érintőképernyő**, a **joystick**, a **gamepad** (ang. *gamepad* – játékvezérlő) (2.12. ábra), a **keypad** (ang. *keypad* – billentyűzetvezérlő) (2.13. ábra), a **kormány**, a **pedálok**, a **dancepad** (ang. *dancepad* – tánctér) (2.14. ábra) stb.



2.12. ábra. Gamepad



2.13. ábra. Keypad



2.14. ábra. Dancepad

A tudományos célra alkalmazott számítógépekben különleges beviteli eszközöket is találhatunk, például hőérzékelőket, légnedvesség-mérőket, különböző szennyeződések szenzorait, sebességmérőket stb.

### Billentyűzet

A billentyűzet rendeltetése szöveges adatok és utasítások bevétele. A billentyűzetek csoportosíthatóak néhány szempont szerint:

- rendeltetés szerint: *szabványos* (asztali gépekhez), *kompakt* (kisebb méretű mobil eszközökhöz), *multimédiás* (a multimédiás tartalommal végzett műveletek irányműködéséhez közvetlenül a billentyűzetről), *játék-vezérlőbillentyűzet* (a számítógépes játékokhoz);
- a csatlakozó típusa szerint: *vezetékes* és *vezeték nélküli* (rádió, bluetooth, wifi).

**A vetített billentyűzet** (2.15. ábra) hatékony adatbeviteli eszköz a mobil eszközök számára. Egyesíti az adatok szabványos méretű billentyűzetről történő bevitelének kényelmét és a mobilitást. Képe bármilyen sík felületre vetíthető és maga a kivetítő kicsi. Egyébként az ilyen billentyűzettel végzett munka nem különbözik a szokásos billentyűzettel végzett munkától.



2.15. ábra. Vetített billentyűzet



### Érdekes történelmi tények

**Norwich** (Nagy Britannia) városának Princes nevű utcáján található egy kőből kifagrott billentyűzet. A helyi lakosok és a látogatók azon törnek a fejüket, hogy hogyan is jelenhetett meg a járdán a billentyűzet, mely több száz éves. Földönkívüli civilizáció jele, vagy a művészeti egyetem hallgatóinak a tréfája?



2.16. ábra. Billentyűzet Norwich utcáján

### A programok vezérlésének eszközei

Az egérrel parancsokat és utasításokat lehet továbbítani a felhasználótól a számítógépes alkalmazásokhoz.

Az egérrel még számos más műveletet is el lehet végezni. A legfontosabb ezek közül a kattintás. Az egéren található görgő segítségével végrehajtható egy bizonyos menüparancs, kiválasztható egy objektum, nagyítható vagy kicsinyíthető a kijelző, megnyitható egy helyi menü, vagy éppen objektumot illeszthetünk be.

Az egereket is több szempont szerint csoportosíthatjuk:

- rendeltetésük szerint lehetnek: *asztali gép*hez, *notebookhoz* vagy *játékgép*hez készült egerek;
- a csatlakozó típusa szerint: *vezetékes (USB, PS/2)* és *vezeték nélküli* (rádió, **bluetooth** wifi stb.);
- az érzékelőjük típusa szerint: *optikai*, *lézeres* és *giroszkópos*;
- a gombok száma szerint: *2*, *3-5*, *6-9*, *10* és *több gombos*;
- kialakítás szerint: *szimmetrikus*, *balkezes*, *szimmetrikus ergonomikus*.



2.17. ábra. Egér

A 2.17. ábrán egy vezetékes lézeres ergonomikus egeret láthatunk, amelyen 10 gombot és egy görgőt találhatunk.

Gyakorlatilag valamennyi notebook alapfelszerelése a **touchpad** (*touch* – hozzáérni, *pad* – lapocska, párna, terület), de ennek az eszköznek is van asztali géphez kapcsolható változata (2.18. ábra). Ha az ujjunkat húzogatjuk a lapkán, az mozgásba hozza az egérmutatót. Ha egyszer megnyomjuk a lapkát, az a bal gombbal történt egyszeri kattintással egyenértékű, ha kétszer, akkor a dupla kattintással. A touchpad gombjai az egér gombjainak funkcióival bírnak.



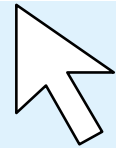
2.18. ábra. Külső touchpad



### Ha többet szeretnél tudni

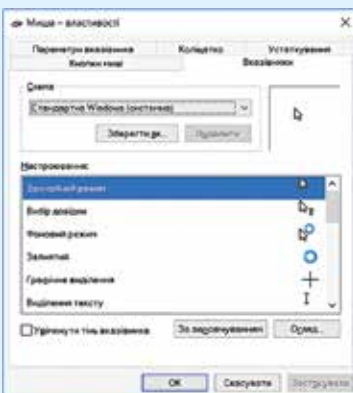
#### Az egérmutató megjelenésének megváltoztatása

A képernyőkön az egérmutató kinézete különböző helyzetekben különböző lehet. A legtöbb esetben nyíl alakú (2.19 ábra). Ugyanakkor lehetőség van az egérmutató megjelenésének módosítására. Ehhez nyissuk meg a **Start** ⇒ **Gépház** ⇒ **Eszközök** ⇒ **Egér** ⇒ **Egér további beállításai** menüt! Nyissuk meg az **Egér tulajdonságai** ablak ⇒ **Mutatók** menüt (2.20. ábra)! Itt több lehetőségünk van az egérmutató kinézetének megváltoztatására:

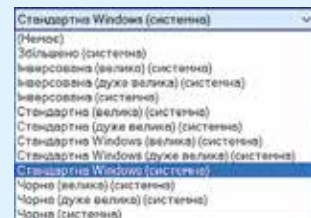


2.19. ábra. Egérmutató

- az egérmutató megjelenésének megváltoztatása a **Sémában**. Ehhez:
  1. Nyissuk meg a **Séma** menüt (2.21. ábra)!



2.20. ábra. Egér tulajdonságai ablak



2.21. ábra. A Séma menü listája

2. Válasszuk ki a megfelelő egérmutatót!
  3. Kattintsunk az **Alkalmaz** gombra!
  4. Zárjuk be az ablakot;
- minden módhoz válasszunk különböző képeket a **Testreszabás** almenü listájából:
    1. Válasszunk egy tetszőleges módot a listából, például a **Szövegkijelölést!**
    2. Válasszuk a **Tallózás** gombot!
    3. A **Tallózás** ablak **Cursor** mappájából válasszuk ki a megfelelő egérmutatót, például a **beam\_im** nevűt!
    4. A kiválasztott mutató nagyított képét a **Villámnézet** mezőben tudjuk megtekinteni.
    5. Válasszuk a **Megnyitás** gombot!
    6. Ismételjük meg az 1-5 pontokat más módokra alkalmazva!
    7. Válasszuk az **Alkalmaz** gombot!
    8. Zárjuk be az ablakot!

Az operációs rendszer istallálása közben a **Cursor** mappába helyezett egérmutatók képeinek száma meglehetősen korlátozott. Ugyanakkor a felhasználó, az interneten számtalan egérmutatóként használható képek gyűjteményét találhat, például, ha az *egér kurzor* kulcsszóval keres. Az egérmutatóként használható képek gyűjteményét – ez a **cur** vagy **ani** kiterjesztéssel rendelkező fájlok kollektója. Ezeket a fájlokat a **Cursor** mappába érdemes másolni, és ha alkalmazni szeretnénk, akkor a fentebbi pontokat kell követnünk.

**Érintőképernyőket** leginkább táblagépekben, okostelefonokban, tájékoztató rendszerekben alkalmazunk. Az adatbevitelt az ujjainkkal vagy speciális tollal végezzük, amelyekkel a képernyő megfelelő részeihez kell hozzáérnünk. A képernyő érzékeli az érintést, és továbbítja a megfelelő információkat a számítógéphez. A legfontosabb tulajdonságuk a képernyőméret (3 és 70 inch között), valamint az érzékelés módja.

A **multimédiás digitális táblát** különféle prezentációk bemutatására használják, többnyire oktatási intézményekben. Ezek különböző méretben (70-79, 80-89, 90 feletti inch) elérhetők, továbbá az érzékelés módjában is különböznek.

### Grafikus adatbeviteli eszközök

A **szkenner**eket (ang. *scanner* – vizsgáló eszköz) grafikus adatok számítógépre történő bevitelére használják. A fejegység fémcsöve alulról megvilágítja a beolvasandó anyagot. A fényérzékelők elemzik a visszavert fény intenzitását és színét, és ezeket az adatokat digitális képpé alakítják.



2.22. ábra.  
Kézi szkenner

Szkennelhetünk szöveges dokumentumot is, de eredményül a szöveg grafikus képét kapjuk. Ez a kép speciális programokkal – **optikai karakterfelismerő rendszerek (OCR – Optical Character Recognition – optikai karakterfelismerés)** – szöveggé alakítható.

Megkülönböztetünk: *kézi szkenner*eket, *asztali szkenner*eket és *szkenner kamerákat*.

A **kézi szkenner**ek (2.22. ábra) kisméretűek és kényelmesek kisebb méretű képek beolvasásához. Szkennelés közben a beolvasandó anyag mozdulatlan és maga a szkenner mozog. Az ilyen típusú szkenner



széles körben használják az üzletekben árukra nyomtatott vonalkódok, vonatokon QR-kódok valamint úti okmányok beolvasására.

A közelmúltban elterjedtek a kézi roll szkennerek, amelyeket különféle hordozható számítógépekhez csatlakoztatva használnak jó minőségű szkennelt képek készítéséhez. Ezeknek a szkennereknek a mérete kicsi és USB csatlakozással vagy önálló akkumulátorral működnek. Általában Wi-Fi kapcsolaton keresztül csatlakoztathatók a hálózathoz. Az egyik ilyen szkennert a 2.23. ábra mutatja.

Az **asztali szkennerek** közül a legelterjedtebbek a **síkgyas szkennerek** (2.24. ábra, 2.4. táblázat). A képet a tárgytartó üvegre kell helyezni, az olvasó egység alatta halad végig. Nagyszámú képpel vagy szöveggel lévő oldalak beolvasásához automatikus lapadagolóval és beépített memóriával rendelkező szkennereket alkalmaznak.

2.4. táblázat

**A síkgyas szkennerek főbb tulajdonságai**

| <b>Tulajdon-ság</b> | <b>Jellemzői</b>  | <b>Mértékegység</b>   | <b>A modern szkennekben való értékek, például az Epson V550 Photoban</b> |
|---------------------|---|---|--|
| Formátum            | A beolvasandó anyag munkaterületének mérete             | mm (méret)  | 216 mm × 297 mm (A4-es formátum)   |
| Felbontás           | Egy hüvelynyi kép képpontjai felismerésének a képessége | Képpontok hüvelyenként – <b>dpi</b> (ang. <i>dot per inch</i> ) | 6400 × 9600  |
| Színek száma        | A szkennert által felismerhető színek száma             | Egységek  | $2,8 \cdot 10^{14}$ (48 bit/pixel)                                       |

A síkgyas szkennerek mellett vannak asztali szkennerek, amelyekben a szkennelési anyagot egy mozdulatlan beolvasó szkenneli be (2.25. ábra).



2.23. ábra.

Kézi roll szkennert



2.24. ábra.

Asztali szkennert



2.25. ábra.

Asztali roll szkennert



2.26. ábra.

Kamera szkennert

A **kamera szkennert** vezeték nélküli vagy dokumentum szkennertnek is nevezik. Jellemző tulajdonsága az, hogy a beolvasandó anyagot, akár csak a fényképezőgépek-nél, egyszerre kapjuk. A beolvasandó anyag felett van a beolvasó kamera (2.26. ábra). Ha megváltoztatjuk a dokumentum és a szkennelést végző kamera közötti távolságot, akkor különböző méretű képeket kaphatunk az objektumokról ahelyett, hogy a művelet csak A4 formátumra korlátozódna. Ezen szkennerek közül több típusban található egy lámpa, ami megvilágítja a beolvasandó anyagot. Az ilyen szkennereket gyakran használják a könyvtárakban könyvek, folyóiratok, szöveges dokumentumok, fényképek és egyéb képek elektronikus másolatának elkészítéséhez.



Az érintőképernyőhöz sokban hasonlít a **digitalizáló tábla**, amelyet rajzolásra, grafikai információk bevitelére használnak. A felhasználó egy speciális tollal hozza létre az ábrát. Előfordul, hogy maga a kép nem jelenik meg a táblán, csak a számítógép képernyőjén. Más esetekben a kép a tábla felületén jelenik meg (2.27. ábra).



2.27. ábra.  
Digitalizáló tábla

A digitalizáló táblák a munkafelületük mérete, felbontóképességük (2000 és 4000 dpi – képpont inchenként – között) szerint különböznek egymástól. Sokféleségük abban is megnyilvánul, milyen nyomásszintet tudnak megkülönböztetni az érintőceruzák (512-től 2048-ig).

### Egyéb adatbeviteli eszközök

Videó vagy hangfájlok beviteléhez fotó és videokamerákat, webkamerákat, mikrofonokat és egyéb multimédiás eszközöket használnak. Itt jegyezzük meg, hogy a kamerák legfontosabb paramétere a felbontóképesség, ami meghatározza a kép minőségét.

**Videokamerák** esetében megkülönböztetjük az **SD** (*Standard Definition*, 720 × 576 képpont), a **HD** (*High Definition*, 1280 × 720 képpont), a **Full HD** (*Full High Definition* 1920 × 1080 képpont) és **Ultra HD** (*Ultra High Definition*, 3840 × 2160 képpont) felbontóképességet.

A **webkamerák** esetében a felbontást ugyanaz határozza meg, mint más digitális fényképezőgépekben, a hüvelykben lévő pontok számának értéke. Egy további jellemző a maximális képkockaszám, amelyet a kamera egy másodperc alatt képes rögzíteni. Ennek a tulajdonságnak az értéke általában 15 és 60 képkocka/másodperc között mozog.

A **fényképezőgépek** esetében a képminőséget az objektív minősége és a képpontok száma határozza meg (10 és 45 megapixel között).

A **mikrofon** olyan eszköz, amely a hangrezgéseket elektromos jellé alakítja. A számítógépekben az elektromos rezgések digitalizálódnak – olyan formátummá alakulnak, amelyet más számítógépes eszközök könnyedén feldolgoznak.

A mikrofonokat audio (hang) adatok számítógépre történő bevitelére használják. Több csoportra osztjuk:

- működési elv (*dinamikus, kristálmikrofon, szalagmikrofon, kondenzátormikrofon* stb.);
- csatlakozás szerint *beépített* (notebookokban, videokamerákban) és *külső* (külön csatlakozik a számítógéphez USB-csatlakozóval), (2.28. ábra);
- hangerő érzékenység szerint.



2.28. ábra.  
Külső mikrofon

A számítástechnikai eszközök fejlődésének köszönhetően a mikrofonok használata széles körben elterjedt a számítógépes kommunikációkban, audio- és videokonferenciáknál, szövegek hangalapú bevitelénél stb.

Az adatbeviteli eszközök folyamatosan változnak, javítják a dizájnjukat és minőségüket, új típusú készülékek jelennek meg. Az eszközök vásárlása előtt érdemes elolvasni a szakértők és a felhasználók véleményét azok használatáról. Az ilyen anyagokat rendszeresen közzéteszik az interneten.





### Összefoglalás

Az **adatbeviteli eszközökhöz** tartozik a szövegbevitelre szolgáló **billentyűzet**, a grafikai adatok bevitelére alkalmazható **szkenner**, **fényképező**, **digitalizáló tábla**, a hang digitalizálására szolgáló **mikrofon** és a videók digitalizálását lehetővé tévő **videokamera**, **webkamera**, **Tv-tuner**, valamint a számítógép vezérléséhez szükséges **egér**, **touchpad**, **elektronikus tábla**, **érintőképernyő**, **joystick**, **keypad**, **gamepad**, **kormány**, **pedál**, **dancepad**.

Az eszközök kiválasztásakor figyelembe kell vennünk azok rendeltetését, a fő tulajdonságát és az árát.



### Felelj a kérdésekre!

- 1<sup>o</sup>. Milyen beviteli eszközöket ismerünk?
- 2<sup>o</sup>. Milyen szempontok szerint csoportosíthatjuk a billentyűzeteket? Hozz fel példákat!
- 3<sup>o</sup>. Milyen típusú egerekkel van felszerelve a számítógépes termetek? Nevezd meg az egerek paramétereit!
- 4\*. Véleményed szerint milyen paraméterekkel kell rendelkezni egy otthoni használatra szánt egérnek?
- 5<sup>o</sup>. Mely eszköz segítségével tudunk grafikus képeket bevinni a számítógépbe?
- 6<sup>o</sup>. Mire használják a szkennert? Milyen típusú szkennereket ismersz?
- 7\*. Milyen típusú szkennerekkel van felszerelve iskolátok számítógépes terme? Milyen típusu szkennert javasoltok a számítógépterembe?
- 8<sup>o</sup>. Milyen célt szolgálnak a multimédiás felszereléshez tartozó eszközök?
- 9\*. Véleményed szerint, miben különbözik a videokamera a webkamerától? Mi bennük a közös?



### Végezd el a feladatokat!

- 1<sup>o</sup>. A tankönyv alapján készítsd el a beviteli eszközök csoportosítását!
- 2\*. A tankönyv adatai alapján készíts egy diás prezentációt, amely bemutatja a beviteli eszközök osztályozását! Mentsd el a munkádat **2.2.2. feladat** néven a mappádba!
- 3<sup>o</sup>. Internetes források (például a **hotline.ua**) alapján keresd ki három olyan egér jellemzőit, amely a közelmúltban került forgalomba! Töltsd ki a táblázatot!



| Tulajdonság           | Érték |  |  |
|-----------------------|-------|--|--|
| Modell                |       |  |  |
| Csatlakozó            |       |  |  |
| Mozgásérzékelő típusa |       |  |  |
| Gombok száma          |       |  |  |
| Kialakítás            |       |  |  |
| Rendeltetés           |       |  |  |



- 4\*. Internetes források (például a **hotline.ua**) alapján keresd ki három olyan asztali roll szkennert jellemzőit, amelyeket Ukrajnában forgalmaznak! A minta alapján hozz létre egy táblázatot a szövegszerkesztők egyikében, majd töltsd ki a táblázatot!

| Tulajdonság         | Érték |  |  |
|---------------------|-------|--|--|
| Modell, gyártó      |       |  |  |
| Csatlakozó          |       |  |  |
| Szkennelési terület |       |  |  |
| Felbontás           |       |  |  |
| Előszkennelési idő  |       |  |  |

- 5\*. Internetes adatok vagy a szaküzletek katalógusai alapján kínálj két kamera szkennert modellt az iskolai könyvtár számára elektronikus könyvtár létrehozása céljából! Indokold meg javaslataidat a megfelelő szövegfájlban!

- 6\*. A rajzolni szerető barátodnak javasolj 2-3 digitalizáló tábla modellt! Indokold a javaslatokat a megfelelő szövegfájlban!



- 7\*. Készíts beszámolót a különféle típusú mikrofonok használatáról multimédiás anyagok létrehozásához!

### 2.3. Kiviteli eszközök



1. Milyen eszközök kapcsolhatók egy számítógéphez? Hogyan lehet őket csoportosítani?
2. Milyen kiviteli eszközök kapcsolhatók egy számítógéphez? Milyen típusú adatok kivitelére szolgálnak ezek?
3. Milyen típusú számítógépes eszközök vannak az informatika-szaktantermetekben? Mire szolgálnak ezek?

A kiviteli eszközöket a beviteli eszközökhöz hasonlóan feloszthatjuk a feldolgozott adattípusok szerint. Az információk megjelenítésére szolgálnak:

- a szöveges és grafikus adatokat a **monitor, nyomtató, plotter (rajzgép)** jeleníti meg;
- a hangadatokat a **fülhallgató** és a **hangfalak** jelenítik meg;
- a videoadatokat a **projektorok**.

Vannak eszközök, amelyek több különböző adattípus megjelenítésére alkalmasak, például a monitor, a multimédiás projektor, megjelenítő fal.

#### Monitor

A **monitor** a személyi számítógép legfontosabb megjelenítő eszköze. Működési elvük alapján a következő kijelzőket különböztethetjük meg:

- **folydékkristályos (LCD – Liquid Crystal Display)**;
- **plazma**;
- **OLED (Organic Light Emitting Diode – szerves fénykibocsátó dióda, LED)**,
- **elektronikus tinta (e-ink)**.

A legelterjedtebb manapság a folydékkristályos technológia.

Az **LCD** monitorok lehetnek:

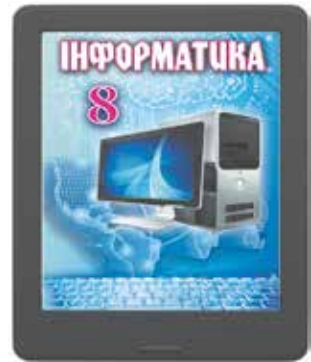
- a működési elv alapján (*TN + film* a *PLS*, az *IPS* és a *VA* technológiák);
- a képátló mérete alapján (19 és 37 inch közötti);
- a felbontás alapján (1280 × 1024-tól 5120 × 2880-ig pixel);
- a reakcióidő alapján – ennyi idő alatt változik meg a képpont színe (1 és 8 ms közötti);
- a képernyő oldalhosszainak arányai alapján (16 : 9, 16 : 10, 21 : 9, 5 : 4).

Az LCD-monitorok alapvető jellemzője az a fényforrás, amellyel megvilágítják a képernyő külső felületét, ahol a kép keletkezik. A modern LCD-monitorok túlnyomó többsége **LED technológiát** (LED fénykibocsátó dióda) használ a képernyő megvilágítására.

A **plazma monitorokat** főleg nagy demonstrációs képernyők létrehozására használják, mivel alacsony felbontásúak, de a keletkezett kép fényereje nagy. A közelmúltban ezt a technológiát felváltotta a folyadékkristályos technológia.

Az **OLED** technológiát több mint 15 éve használják kiváló minőségű monitorok létrehozására, a magas kontraszt és a kép fényereje valamint kicsi (legfeljebb 1 cm) vastagságuk miatt. Ellenben ezeknek a monitoroknak a magas költsége, a fényerő jelentős csökkenése 3-5 éves működés után, jelentősen csökkenti piaci értéküket.

Az **e-tinta** (elektronikus tinta) olyan megoldást jelent, amely a papírra nyomtatott tintát imitálja digitális eszközökön. Emiatt a legtöbb ilyen eszköz csak fekete-fehérben működik. Hosszan működik (több hét) utántöltés nélkül, a képernyő villogásmentes, így a képernyőről történő olvasás sokkal kényelmesebb. Ezért az e technológiával létrehozott monitorokat elsősorban az e-könyvekben használják (2.29. ábra) – táblagépekben, amelyeket szövegek olvasására terveztek. Az **E Ink** 2016 óta dob piacra színes monitorokat, de a hagyományos számítógépeken való széles körű használatukat akadályozza a nagyon alacsony képernyőfrissítési frekvencia, az alacsony számú (4096) reprodukálható szín és a további háttérvilágítás szükségessége.



2.29. ábra. E-tintán alapuló elektronikus könyv



2.30. ábra. Videókártya

## Videókártya

Mielőtt az adatok megjelenhetnének a képernyőn, azokat a számítógép egy speciális eszköze, a **videókártya** dolgozza fel. A videókártyák lehetnek *alaplagra integráltak* vagy *külsők*. Az alaplapra integrált videókártya egy specializált alaplapba beépített chip. Ezt a megoldást olyan számítógépeknél alkalmazzák, amelyeken nem dolgoznak fel nagy mennyiségű videoinformációt. Amennyiben nagy mennyiségű videót kell feldolgozni, egy külső, az alaplap egyik bővítőhelyébe illesztett videókártyát célszerű alkalmazni (2.30. ábra).

**Integrált** (lat. *integer* – egész) – elválaszthatatlanul összekapcsolva, egész.

**Integráció** (lat. *integratio* – utánpótlás) – egyes részek egyesítése.

**Chipset** (ang. *chip* – mikroséma, *set* – készlet) – egy chipkészlet, amely adatcserét biztosít a számítógépes eszközök között.

A videokártya egy speciális processzort és kiegészítő RAM-ot (**videó-RAM** is) tartalmaz. A videokártyákat különböző gyártók állítják elő, például **AMD Radeon RX 5500 XT**, **NVIDIA GeForce RTX 2080**, különbözhetnek a videomemória kapacitásában (2 és 24 Gb-át között), az alaplapi csatlakozójukban (**PCI Express 3.0**, **PCI Express 4.0**) és a monitorhoz való csatlakozásban (**HDMI**, **D-Sub**, **DisplayPort**, **DVI-D**).

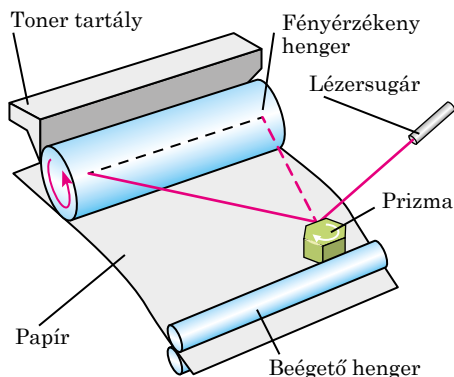
### Nyomatók

Az információ papír vagy fényképlemez alapú megjelenítésére **nyomatókat** és **plottereket** használnak.

A nyomtatók főbb tulajdonságai:

- a felület mérete, amelyre nyomtatnak – **A4**, **A3**;
- színek használata – **színes**, **fekete-fehér**;
- felbontás – **600 x 1200-tól 5760 x 1440 dpi-ig** és nagyobbak;
- nyomtatási sebesség **1 és 80 oldal/perc** között;
- csatlakozó típusa – **vezetékes**, **vezeték nélküli**.

A nyomtatók között lehetnek **mátrixnyomatók**, **lézernyomatók**, **hőnyomatók**, **tintasugaras nyomtatók**, **szublimációs nyomtatók** és **3D nyomtatók**. A leggyakoribbak azok a nyomtatók, amelyek tintasugaras és lézeres technológiát alkalmaznak, bár a közelmúltban a szublimációs és a 3D nyomtatók egyre népszerűbbek.



2.31. ábra. Lézernyomató

**A lézernyomató.** A lézernyomató egyik fő alkotóeleme a fényérzékeny anyaggal bevont henger. (2.31. ábra). A lézeregység vékony fénysugarat generál, amely a henger töltését megváltoztatja a beesési ponton. A festékpór az elektromos töltés miatt rátapad a henger felületére. A festék mennyisége a henger töltöttségének arányától függ az adott pontban. A papír áthalad a forgó henger alatt és közben rátapad a festék.

Ezután a papíron lévő festéket egy beégető egység (ún. *ráégető*) magas hőfokon, közel 180 °C-on fixálja a papíron. Ezt követően sem a nedvesség, sem a közvetlen napfény nem árt neki.

Színes nyomtatáshoz színes tonereket használunk.

**A tintasugaras nyomtató.** Az ilyen típusú nyomtatók képkalkotási technológiája azon alapul, hogy nagyon kicsi, különböző

**Toner** (ang. *toner* – árnyalatot adó) – porszerű vagy krémszerű festék, amely megváltoztatja azon felület színét, amelyikre rávitték.

**Ráégető** (ang. *fuse* – égető) – olvasztó, sütő.

színű tintacseppek kerülnek papírra vagy más anyagra. A cseppek előállításának módjától függően megkülönböztetünk:

- **piezoelektromos technológiát** – a cseppet egy olyan lemez préseli ki, amely elektromos áram hatására deformálódik (például **Epson** nyomtatók);
- **hőbuborék technológiát** – a gyors felmelegedés hatására a tinta helyenként felforr és a keletkezett gőzbuborék egy-egy tintacseppet lő a papírlapra (például **Hewlett Packard**, **Canon**).



A tintasugaras nyomtatók egyik hátránya, hogy a közvetlen napfénynek kitett tintás nyomatok egy idő után kifakulhatnak, valamint kevésbé vízálló a tinta.

A **szublimációs nyomtatással** élénk, színes képek nyomtathatók különféle felületekre: papírra, kartonra, valamint szövetre, üvegre, fajanszra és egyebekre. A szublimáció (latin *sublimiter* – felfelé, egyenesen) az anyag szilárd állapotból gázállapotba történő átmenetének fizikai jelensége, megkerülve a folyékony állapotot. A szublimációs nyomtatásnak számos lehetősége van:

**Poliészter** (görög πολύ – sok és a német *Essig-Aether* – ecetsav-éter szóból) – olyan anyag, amelyet többek között szintetikus szövetek létrehozására használnak.

- **közvetlen „nedves” nyomtatás** speciális (szublimációs) festékekkel poliésztert tartalmazó felületre, vagy olyan felületre, amelyre korábban poliészter anyagréteget helyeztek;
- **közvetlen „száraz” nyomtatás** különböző színű fóliák felhasználásával (2.32. ábra), amelyből a melegítés során a szilárd tinta elpárolog és átkerül arra a felületre, amelyen a minta kialakul;
- **közvetett nyomtatás**, amelynek első lépése a nyomtatás tintasugaras nyomtatóval és szublimációs festékekkel, egy speciális szublimációs papírra. A második szakaszban a papíron létrehozott képet a tárgy felületére viszik fel, amelyre a képet nyomtatni kell, és 100 °C feletti hőmérsékletre melegítik. A festék a papírról elpárolog és képeket képez csészéken, pólókon, tányérokon stb. Ebben az esetben poliészter szöveteket vagy felületeket is használnak előzőleg felhordott poliészter anyagréteggel.



2.32. ábra. Szublimációs nyomtató és a hozzá tartozó színes fóliák készlete



2.33. ábra. Szublimációs nyomtató

A közvetlen „nedves” nyomtatást általában az ipari nyomdaipari vállalatok nagy képkiadásaihoz, a közvetlen „száraz” nyomtatást kiváló minőségű fényképek nyomtatásához használják (2.33. ábra), közvetett nyomtatást pedig ajándéktárgyak készítéséhez használják, akár otthoni célra is.

A szublimációs nyomtatás legfőbb előnye a kiváló képminőség, valamint a nedveséghez és a fényekhez való magas ellenálló képesség.

### Tudtad-e, hogy...?

A híres angol tudós és feltaláló, Charles Babbage javasolta elsőként a számítások eredményeinek szilárd felületre történő felvitelét. Első számítógépének megtervezése során Babbage nyomtatót csatlakoztatott hozzá, amely rézlemezekre préselte rá a számítások eredményét. Az így kapott szám sokáig tárolható volt és lehetetlenné tette az emberi figyelmetlenség miatt bekövetkező hibákat.

**A plotterek** (rajzgépek) olyan speciális berendezések, amelyeket akkor alkalmazunk, ha grafikus formájú kimenő adatra van szükségünk. Rajzok, vázlatok, poszterek és egyéb nagy képek nyomtatására szolgál, általában 297 mm × 420 mm-nél nagyobbak (A3 formátum).

A modern plotterek kialakításának alapjául szolgáló elvek nem sokban különböznek a nyomtatók alapelveitől. A leggyakoribbak a tintasugarat alkalmazó plotterek.

A **3D nyomtató** digitális modellekből háromdimenziós tárgyak alkotására képes eszköz (2.34. ábra). A 3D-s nyomtatókat azon anyagok típusa szerint különböztetik meg, amiből "felépül" az adott tárgy.

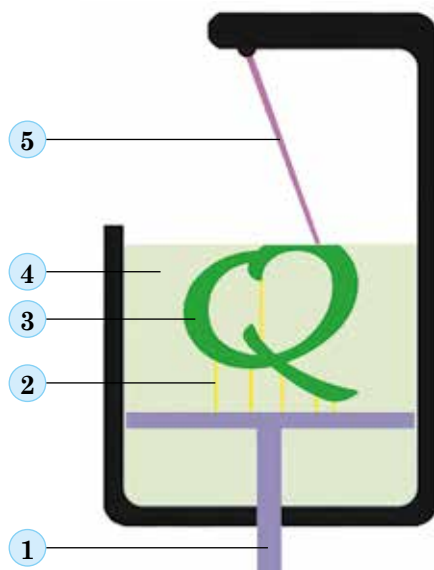
A következő technológiát használják:

- speciális gyanta megkötése fény (lézersugárzás vagy LED) hatására – **SLA** (ang. *Stereolithography apparatus* – sztereolitográfiai berendezések), vagy sztereolitográfiai technológia (2.34. ábra);
- speciális porszemcsék összeolvasztása és szilárdulása lézersugár hatására – **SLS** technológia (ang. *Selective Laser Sintering* – szelektív lézer szinterezés);
- megolvadt műanyagok megszilárdulása – **FDM** technológia (ang. *Fused Deposition Modeling* – fuzionált lerakódás modellezése).

Az **FDM** technológia a legolcsóbb. Ennek köszönhetően használata széles körben elterjedt mind az otthonunkban, mind a tanintézményekben. Az **SLA** és az **SLS** technológiák jelentős előnnyel rendelkeznek a gyártási tárgyak pontosságában, és széles körben használják őket az ipari termelésben.

A 3D nyomtatókat csoportosíthatjuk az általuk készíthető objektumok maximális mérete (hosszúság, szélesség, magasság) szerint is.

A három fő technológia ötfokú skálán történő összehasonlítását a 2.5. táblázat tartalmazza.



1. Tárgyalkotási felület
2. A tárgyalkotás segédanyagai, melyeknek nincs ellenállásuk
3. Nyomtató tárgy (Q betű)
4. Sugár hatására megszilárduló anyag
5. Ultraibolya sugár

2.34. ábra. Az SLA 3D nyomtató működési elve

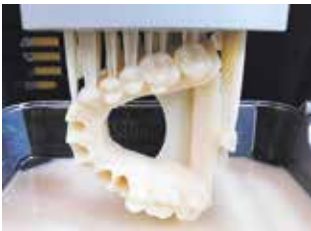




A 3D-nyomatási technológiák jellemzőinek összehasonlítása

|   | Megnevezés                                   |                                    |                                    |
|---|--|------------------------------------|------------------------------------|
|   | FDM  | SLA                                | SLS                                |
| Felbontás                                       | ●●○○○  | ●●●●●                              | ●●●●○                              |
| Gyártási pontosság                              | ●●●●○  | ●●●●●                              | ●●●●●                              |
| Felületi érdesség                               | ●●○○○  | ●●●●●                              | ●●●●○                              |
| Összetett termékek létrehozásának képessége     | ●●●○○  | ●●●●○                              | ●●●●●                              |
| A használat egyszerűsége                        | ●●●●●  | ●●●●●                              | ●●●●○                              |
| Asztali nyomtatók nyomtatási tárgyainak méretei | 200 mm ×<br>× 200 mm ×<br>× 300 mm           | 300 mm ×<br>× 335 mm ×<br>× 200 mm | 165 mm ×<br>× 165 mm ×<br>× 320 mm |
| A helyiségre vonatkozó követelmény              | Légkondicionálás vagy speciális szellőztetés | Elegendő egy irodahelyiség         | Ajánlott külön műhelybe helyezni   |

A 3D nyomtatókat széles körben alkalmazzák az emberi tevékenység különböző területein (2.35-2.38. ábra). Különösen aktívan használják őket az orvostudományban.



2.35. ábra. Fogprotézis készítése 3D nyomtatóval



2.36. ábra. Mellkasprotézis készítése 3D nyomtatóval



2.37. ábra. 3D nyomtatóval elkészített épületmodell



2.38. ábra. Egy autó 3D-s hengerblokk modellje



2.39. ábra. 3D nyomtatási technológiák videoanyagainak QR-kódjai

3D nyomtatási technológiákat bemutató videóanyagokat a következő linkeken lát-hatjátok: <https://cutt.ly/ag1ZJpO> és <https://cutt.ly/Kg1ZZup>, vagy használhatjátok a 2.39-es ábra QR-kódját.



2.40. ábra. Az Eiffel-torony modelljének megalkotása 3D tollal

A **3D tollak** új adatkimeneti eszközök a térben történő „rajzoláshoz”. Általában a 3D nyomtatók FDM technológiáját használják, vagyis képeket készítenek egy olvadt műanyag vékony (0,5-1,5 mm) sugárjának kinyomásával, amely azonnal megkeményedik és háromdimenziós tárgyat képez (2.40. ábra). Vannak olyan tollak, amelyekhez az ultraibolya sugárzás hatására megkeményedő gyantákon alapuló többszínű tintákat használnak. Az ilyen tollak sokáig működhetnek anélkül, hogy újratöltenék.

A 3D tollak munkavégzéséről szóló videóanyagok a következő linkeken tekinthetők meg: <https://cutt.ly/Xg1ZCJ1> vagy <https://cutt.ly/6g1ZBJ2>, vagy

<https://cutt.ly/qg1Z1eo> vagy a 2.41-es ábra QR-kódjával.



2.41. ábra. A 3D tollak munkavégzéséről szóló videóanyagok QR-kódja

Az eszközök tulajdonságainak értékei folyamatosan változnak, fejlődnek különböző alakban is és minőségben is. Az eszközök vásárlása előtt érdemes elolvasni a szakértők és a felhasználók véleményét azok használatáról. Az ilyen anyagokat rendszeresen közzéteszik az interneten.

### Használjuk a számítógépet!



Végezzétek el a gyakorló feladatokat a <https://cutt.ly/AhdeMSZ> link, vagy a QR-kód segítségével!

### Összefoglalás

A **kiviteli eszközökhöz** tartoznak a szöveges és grafikai adatokat megjelenítő **monitor**, **nyomtató**, **plotter**, a hang lejátszásához szükséges **hangfalak** és **fülhallgatók**, a videólejátszáshoz pedig a **multimédiás projektor** és **videofal**.

A monitorok a képképzés módja szerint lehetnek:  **folyadékkristályos monitorok** vagy **LCD monitorok**,  **plazma**, **OLED**, **elektronikus tinta – e-ink** és így tovább. A legtöbb modern monitor folyadékkristályos technológiát alkalmaz.

Mielőtt az adatok megjelenhetnének a képernyőn, azokat a számítógép egy speciális eszköze, a **videokártya** dolgozza fel. A videokártya egy speciális processzort és kiegészítő RAM-ot (**videó-RAM** is) tartalmaz.

A **nyomtatókat** és a **plottereket** az adatok papírra, filmre vagy más anyagokra történő nyomtatására használják.

A működési elv alapján a nyomtatók lehetnek **mátrixnyomtatók**, **lézernyomtatók**, **hőnyomtatók**, **tintasugaras nyomtatók**, **szublimációs nyomtatók** és **3D-nyomtatók**. A leggyakoribbak azok a nyomtatók, amelyek tintasugaras és lézeres technológiát használnak.

Az elmúlt években a háromdimenziós tárgyak létrehozásának technológiája nagyot fejlődött a **3D-s nyomtatók** és **3D-s tollak** megjelenésével.



**Felelj a kérdésekre!**

- 1<sup>o</sup>. Milyen kiviteli eszközöket ismerünk?
- 2<sup>o</sup>. Milyen szempontok szerint csoportosíthatjuk a monitorokat? Hozz fel példákat a manapság használt monitorok típusaira!
- 3<sup>o</sup>. Milyen típusú nyomtatóval van felszerelve az informatikatermetek? Nevezd meg a nyomtató paramétereit!
- 4\*. Véleményed szerint milyen paraméterekkel kell rendelkeznie egy otthoni használatra szánt videokártyának?
- 5<sup>o</sup>. Miben különbözik a lézernyomtató a tintasugaras nyomtatótól? Melyikkel lehet szebb képet létrehozni?
- 6<sup>o</sup>. Milyen nyomtatókkal készítek az ajándéktárgyakat kis mennyiségben?
- 7<sup>o</sup>. Melyek azok a kimeneti eszközök, amelyek a számítógépes multimédia részét képezik?
- 8\*. Mit gondolsz, miben különbözik a 3D nyomtató a 3D tolltól? Mi bennük a közös?
- 9\*. Véleményed szerint milyen paraméterekkel kell rendelkeznie egy otthoni használatra szánt nyomtatónak?



**Végezd el a feladatokat!**

- 1<sup>o</sup>. A tankönyv alapján készítsd el a kiviteli eszközök csoportosítását!
- 2\*. A tankönyv adatai alapján készíts egy diát, amely bemutatja a beviteli eszközök osztályozását! Mentsd el a munkádat **2.3.2. feladat** néven a mappádba!
- 3<sup>o</sup>. Internetes források (például a **hotline.ua**) alapján keresd ki három olyan monitor jellemzőit, amelyek a közelmúltban kerültek forgalomba Ukrajnában! Töltsd ki a táblázatot!

| Tulajdonság      | Érték |  |  |
|------------------|-------|--|--|
| Modell           |       |  |  |
| Gyártó           |       |  |  |
| Képtároló, inch  |       |  |  |
| Típus            |       |  |  |
| Felbontás, pixel |       |  |  |
| Válaszidő, mp    |       |  |  |

- 4<sup>o</sup>. A tankönyv adatai alapján készítsd el a nyomtatók csoportosítását valamely általad ismert alkalmazás segítségével! Mentsd a dokumentumot **2.3.4. feladat** néven a mappádba!



- 5<sup>o</sup>. Határozd meg az otthoni számítógépek paramétereit, és töltsd ki a táblázatot a **Használjuk a számítógépet!** rubrikában szereplő táblázatnak megfelelően (<https://cutt.ly/AhdeMSZ>)!

## 2. fejezet



6\*. Internetes adatok vagy a szaküzletek katalógusai alapján határozd meg három különböző gyártótól származó, Ukrajnában forgalmazott nyomtató (tintasugaras és lézeres) paramétereit! Töltsd ki a táblázatot!

| Tulajdonság                     | Érték |  |  |
|---------------------------------|-------|--|--|
| Modell                          |       |  |  |
| Gyártó                          |       |  |  |
| Működési elv                    |       |  |  |
| Felbontás, pixel                |       |  |  |
| Nyomtatási sebesség, oldal/perc |       |  |  |



7\*. Készíts rövid prezentációt a tintasugaras nyomtatóról! Mutasd be a felépítését, működését és alkalmazását!



8\*. Készíts előadást (prezentációt) a 3D nyomtatók és a 3D tollak használatáról! Fordíts különös figyelmet a 3D nyomtatók ipari használatára!

### 2.4. A számítástechnikai és számítógépes eszközök története



1. A számítógépnek milyen típusait ismeritek? Miben különbözik ezek rendeltetése?
2. Milyen alkalmazásai vannak a számítógépeknek az emberek különféle tevékenységi területein?
3. Nevezetek meg olyan ukrán tudósokat, akik jelentősen hozzájárultak a számítástechnika fejlődéséhez!

#### Az információs folyamatok megvalósításának szakaszai

Az emberiség fejlődése szorosan kapcsolódik az üzenetek továbbításának, feldolgozásának és tárolásának fejlődéséhez. Ez a fejlődés több szakaszra bontható (2.6. táblázat).

2.6. táblázat

#### Az információs folyamatok megvalósításának fejlődése

| A szakasz                  | Történelmi kor                                | Az adatok továbbításának, feldolgozásának, tárolásának eszközei   |
|----------------------------|---|---|
| Kézi eszközök kora         | Az ókortól a XV. század közepéig              | <i>Adathordozók</i> – agyagtáblák, papirusztekercsek, rovábotok, képek.<br><i>Adatok továbbítása</i> – jelzőtüzek, dobok, küldöncök, postagalambok, lovasfutárok.<br><i>Adatok feldolgozása</i> – emberi agy és ujjak, kézi számolóeszközök (abakusz, szorobán stb.). |
| A mechanikus eszközök kora | A XV. század közepétől a XIX. század közepéig | <i>Adathordozók</i> – könyvek, újságok, folyóiratok, fotólemezek.<br><i>Adatok továbbítása</i> – postaszolgálat.<br><i>Adatok feldolgozása</i> – nyomdagépek, fényképezők, aritmóméterek, zenegépek, gramfonok, kézi számológépek.                                    |

| A szakasz                     | Történelmi kor                               | Az adatok továbbításának, feldolgozásának, tárolásának eszközei   |
|-------------------------------|--|---|
| Az elektromos eszközök kora   | A XIX. sz. közepétől a XX. sz. 40-es éveikig | <i>Adathordozók</i> – lemezek, filmek, mágnesszalagok.<br><i>Adatok továbbítása</i> – telefon, távíró, rádió.<br><i>Adatok feldolgozása</i> – fonográf, gramofon, elektromos írógépek, elektromos aritmométerek, magnók, nyomdagépek. |
| Az elektronikus eszközök kora | A XX. sz. 40-es éveitől napjainkig           | <i>Adathordozók</i> – mágneses és optikai lemezek, mikrocsipek.<br><i>Adatok továbbítása</i> – televíziós műsorszórás, számítógépes hálózatok, mobilszolgáltatás.<br><i>Adatok feldolgozása</i> – zsebszámológépek, számítógépek      |

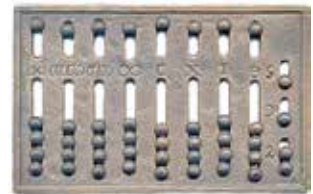
### Üzenetfeldolgozás eszközeinek fejlődése

Vizsgáljuk meg részletesebben az üzenetfeldolgozás eszközeinek, jelesül a számolás eszközeinek fejlődését. A számolás első eszközei az emberi ujjak voltak (2.42. ábra). Erről tanúskodnak például a római számok (I, V, X). A régi szláv számlálásban is ujjakat használtak, ezt bizonyítja az öt (*pjaty*) szó, ami egy kézfejet jelentett.



2.42. ábra. Számolás ujjakon

Jelentős lépés volt a számolás fejlődésében az abakusz feltalálása az i. e. V. században az ókori Görögországban. Az ötletet a görögök a babiloniaktól vették át. A 2.43. ábrán az abakusz egy későbbi változatát, a római abakuszt láthatjuk. Az abakusz tehát egy függőleges bevágásokkal ellátott deszka, amelynek vágataiba apró tárgyakat, jellemzően köveket helyeztek el. Ebből származik a készülék neve – **számológép** (lat. *calculus* – kavics), és a **számolás** (ang. *calculate* – számolni). Hasonló eszközöket más országokban, Kínában (**szuálpány**), Japánban (**szorobán**), Oroszországban (**szcsoti** – számológép) (2.44. ábra). Ezek az eszközök a XX. század végéig használatban voltak.



2.43. ábra.  
Római abakusz



2.44. ábra. Számológép

Ezekkel az egyszerű és széles körben elterjedt eszközökkel egyidejűleg az ókori Görögországban és Rómában különböző fogaskerekekből összeállított számolóeszközöket is használtak, melyeket egy elsüllyedt ókori hajón talált eszközök tanúsítják. Ezt az eszközt **antiküthérai szerkezetnek** nevezték el, mivel a roncsot Antiküthéra mellett



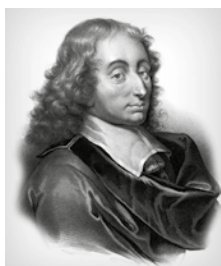
2.45. ábra. Antiküthérai mechanizmus:  
a megtalált maradványok és a számítógépes modell

találták a szivacshalászok 1902-ben. A tudósoknak csak a XX. század második felében sikerült megfejteni a szerkezeti elemek rendeltetését és restaurálni azt (2.45. ábra). Az i. e. 150 körül létrehozott eszköz naptárként is funkcionált, alkalmas volt az égitestek – Nap, Hold, Merkúr, Vénusz, Mars, Jupiter, Szaturnusz – helyzetének meghatározására.

Sajnos az antiküthérai mechanizmus létrehozásához elengedhetetlen tudás hosszú időre feledésbe merült. A hasonló eszközök létrehozására irányuló próbálkozások csak a XVII. század végén éledtek újra, mivel az ipar, a kereskedelem és a haditechnika fejlődése ekkor ezt egyre erősebben igényelte.

1642-ben **Blaise Pascal** (1623–1662) (2.46. ábra) francia matematikus, fizikus, mérnök a királyi tanács elé terjesztette egy mechanikus számolóeszköz tervét. A későbbiek során több tíz ilyen eszközt (2.47. ábra) készített. Az eszközöket összeadásra és kivonásra használták.

A későbbiekben több matematikus és mérnök fejlesztette tovább Pascal ötleteit. **Gottfried Leibnitz** (1646–1716) (2.48. ábra) német tudós hozta létre az első olyan számológépet, amellyel mind a 4 alapműveletet el lehetett végezni (2.49. ábra).



2.46. ábra.  
Blaise Pascal



2.47. ábra.  
Pascal számológépe



2.48. ábra.  
Gottfried Leibnitz



2.49. ábra.  
Leibnitz  
számológépe

Ezeket az eszközöket idővel tovább tökéletesítették és a XX. század közepéig használták is, amíg fel nem váltották őket az elektronikus kalkulátorok.

A számológépek tökéletesítése során a tudósok arra törekedtek, hogy a gép képes legyen automatikusan számításokat végezni egy előre kidolgozott program alapján (2.50. ábra). Az ilyen gép létrehozását először **Charles Babbage** (1792–1871) angol matematikus és tervező (2.51. ábra) tűzte ki céljául. **Analiticalengine**-nek (**analitikus gép**) nevezte el az eszközt.

A tervek szerint Babbage eszköze a következő részekből állt:

1. **Raktár** – mai szóhasználattal **memória** a számok tárolására.





2.50. ábra.  
Müller számológépe



2.51. ábra.  
Charles Babbage

2. **Malom** – a matematikai műveletek végrehajtására, mai nyelven **aritmetikai egység**.

- 3. A matematikai műveletek sorrendjét (**vezérlőegység**).
- 4. A kezdőértékeket beállító egység.
- 5. Kiviteli egység.

**Ada Lovelace** (1815–1852) (2.52. ábra), George Byron angol költő lánya, Babbage társa volt az analitikai gép tervezésében és kivitelezésében. Ő dolgozta ki a programtervezés lépéseit. Ezért Ada Lovelace-t tekintik az első programozónak, tiszteletére nevezték el **Ada**-nak az egyik programozási nyelvet.



2.52. ábra.  
Ada Lovelace

A XIX. sz. végéig a számológépek kézi meghajtású mechanikus gépek voltak (2.28. ábra). **Herman Hollerith** (1860–1929) amerikai tudós csak a XIX. század végén fejlesztette ki az első elektromos árammal hajtott eszközt, a **tabulátort** (2.53. ábra). A gépet a népszámlálás adatainak feldolgozásához használták. Az emberek adatait nem papírra írták, hanem egy speciális, lyukakkal ellátott kártyára (**lyukkártya**) vitték fel (2.54. ábra).

**Perforáció** (lat. *perforate* – perforáció) – lyukasztás, lyukak készlete



2.53. ábra.  
Hollerith tabulátora

| PERSONENSTATISTIK |           |   |            |   |       |   |         |   |      | HAUSHALTUNGSSTATISTIK |       |   |         |   |      |   |       |   |         | WOHNUNGSSTATISTIK |      |   |       |   |  |  |  |  |  |
|-------------------|-----------|---|------------|---|-------|---|---------|---|------|-----------------------|-------|---|---------|---|------|---|-------|---|---------|-------------------|------|---|-------|---|--|--|--|--|--|
| Zählort           | Geb. Jahr |   | Arbeitsort |   | Eink. |   | Wohnung |   | Miet |                       | Kond. |   | Wohnung |   | Miet |   | Kond. |   | Wohnung |                   | Miet |   | Kond. |   |  |  |  |  |  |
| O. A. Gem.        | M         | F | M          | F | M     | F | M       | F | M    | F                     | M     | F | M       | F | M    | F | M     | F | M       | F                 | M    | F | M     | F |  |  |  |  |  |
| 0                 | 0         | 0 | 0          | 0 | 0     | 0 | 0       | 0 | 0    | 0                     | 0     | 0 | 0       | 0 | 0    | 0 | 0     | 0 | 0       | 0                 | 0    | 0 | 0     | 0 |  |  |  |  |  |
| 1                 | 1         | 1 | 1          | 1 | 1     | 1 | 1       | 1 | 1    | 1                     | 1     | 1 | 1       | 1 | 1    | 1 | 1     | 1 | 1       | 1                 | 1    | 1 | 1     | 1 |  |  |  |  |  |
| 2                 | 2         | 2 | 2          | 2 | 2     | 2 | 2       | 2 | 2    | 2                     | 2     | 2 | 2       | 2 | 2    | 2 | 2     | 2 | 2       | 2                 | 2    | 2 | 2     | 2 |  |  |  |  |  |
| 3                 | 3         | 3 | 3          | 3 | 3     | 3 | 3       | 3 | 3    | 3                     | 3     | 3 | 3       | 3 | 3    | 3 | 3     | 3 | 3       | 3                 | 3    | 3 | 3     | 3 |  |  |  |  |  |
| 4                 | 4         | 4 | 4          | 4 | 4     | 4 | 4       | 4 | 4    | 4                     | 4     | 4 | 4       | 4 | 4    | 4 | 4     | 4 | 4       | 4                 | 4    | 4 | 4     | 4 |  |  |  |  |  |
| 5                 | 5         | 5 | 5          | 5 | 5     | 5 | 5       | 5 | 5    | 5                     | 5     | 5 | 5       | 5 | 5    | 5 | 5     | 5 | 5       | 5                 | 5    | 5 | 5     | 5 |  |  |  |  |  |
| 6                 | 6         | 6 | 6          | 6 | 6     | 6 | 6       | 6 | 6    | 6                     | 6     | 6 | 6       | 6 | 6    | 6 | 6     | 6 | 6       | 6                 | 6    | 6 | 6     | 6 |  |  |  |  |  |
| 7                 | 7         | 7 | 7          | 7 | 7     | 7 | 7       | 7 | 7    | 7                     | 7     | 7 | 7       | 7 | 7    | 7 | 7     | 7 | 7       | 7                 | 7    | 7 | 7     | 7 |  |  |  |  |  |
| 8                 | 8         | 8 | 8          | 8 | 8     | 8 | 8       | 8 | 8    | 8                     | 8     | 8 | 8       | 8 | 8    | 8 | 8     | 8 | 8       | 8                 | 8    | 8 | 8     | 8 |  |  |  |  |  |
| 9                 | 9         | 9 | 9          | 9 | 9     | 9 | 9       | 9 | 9    | 9                     | 9     | 9 | 9       | 9 | 9    | 9 | 9     | 9 | 9       | 9                 | 9    | 9 | 9     | 9 |  |  |  |  |  |

2.54. ábra. Lyukkártya, amit az 1910-es németországi népszámláláson használtak

Hollerith 1896-ban alapította a tabulátorok gyártására szakosodott vállalatát, amelyből néhány tulajdonosváltás és átszervezés után 1924. február 14-én megalakult az **IBM Corporation**. Ma ez a vállalat a mainframe számítógépek egyik legnagyobb gyártója.

A XX. század közepéig sokat javítottak a mechanikus és elektromechanikus gépek szerkezetén, gyorsítottak a műveletek elvégzésén.



2.55. ábra.  
Konrad Zuse

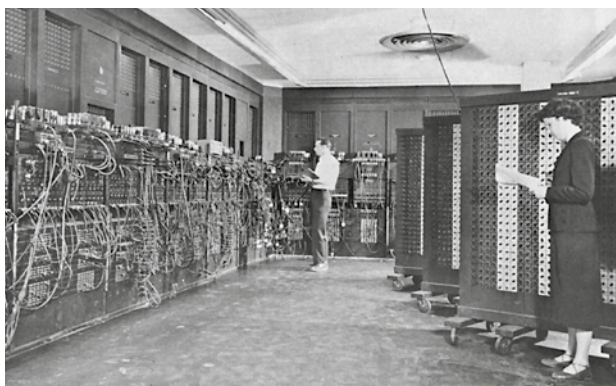
A XX. század 30-as–40-es éveiben a tudósok új típusú, elektromos eszközöket gyártó számítógépeken dolgoztak. 1941-ben Németországban **Konrad Zuse** (1910–1995) létrehozta az elektromágneses reléken működő **Z3**-at. Ez a számítógép már kettes számrendszert használt. Ugyanúgy Zuse fejlesztette ki 1950-ben az elektroncsöveket alkalmazó **Z4**-et.

Ugyancsak 1941-ben az USA-ban **John Vincent Atanasoff** (1903–1995) és **Clifford Barry** (1918–1963) létrehozták az első kettes számrendszert alkalmazó elektroncsöves gépet, az **ABC**-t (**Atanasoff-Berry Computer**). Valamivel később (1939–1944), ugyancsak az USA-ban **Howard Aiken** (1900–1973) az IBM-nel együttműködve létrehozta az automatikus vezérlésű elektromechanikus gépet, a **Mark-1**-et.

Brit tudósok 1943–1944-ben a legnagyobb titokban a német rádiótávíratok rejtjelezésének megfejtésére létrehozták a **Colossust**. Ez abban különbözött a többi számítógéptől, hogy a vezérlőprogramját a memóriájában tárolta.

A fejlesztést egy angol tudóscsoport végezte **M. Newman** (1897–1984) és **T. Flowers** mérnök (1905–1998) vezetésével, figyelembe véve a megoldás matematikai megközelítését, amelyet **Alan Turing** (1912–1954) javasolt.

1943 és 1946 között az USA-ban **John Mauchly** (1907–1980) és **Presper Eckert** (1919–1995) megalkották az **ENIAC**-ot (**Electronic Numerical Integrator and Calculator**), a 18 000 elektroncsövből álló 30 t súlyú számítógépet, amely ötezer műveletet volt képes elvégezni másodpercenként (2.56. ábra). A projekt befejezése után a fejlesztők azonnal hozzáláttak az **EDVAC** (**Electronic Discrete Variable Automatic Computer**) katonai célú számítógép létrehozásához.



2.56. ábra. ENIAC számítógép



A munkába bekapcsolódott **Neumann János** (1903–1957) magyar származású matematikus, aki a munka kapcsán megírta a *First Draft of a Report on the Edvac* című munkáját, amelyben lefektette az univerzális számítógép működésének elveit. Bár a munkában Eckert és Mauchly is részt vettek, ezeket az elveket a világ *Neumann-elvekként* ismerte meg.

A számítógépek a nagy mennyiségű és viszonylag olcsó **minigépek** megjelenésével kezdtek elterjedni, melyeket később személyi számítógépeknek neveztek el. Kezdetben ezeket alkatrészekből álló készletekként forgalmazták, amelyeket a felhasználónak kellett összeszerelni és a tévékészülékeket használták monitorként. Az első személyi számítógépeket a következő cégek készítették:

- **MITS (Micro Instrumentation and Telemetry Systems)** által gyártott **Altair 8800** (1975, 2.57. ábra);
- **Apple Computer Company** által gyártott **Apple I** (1976, 2.58. ábra);
- **IBM** által készített **IBM PC** (1981, 2.59. ábra).



2.57. ábra. Altair 8800



2.58. ábra. Apple I



2.59. ábra IBM PC

A személyi számítógépek elterjedése nagymértékben kibővítette a számítógépek segítségével megoldható feladatokat.

A számítástechnika fejlődésének történetét részletesebben az időrendi táblázatban vizsgálhatjuk (3-as melléklet, <https://cutt.ly/NkBXn8o>).

### A számítástechnika fejlődése Ukrajnában

1951-ben Kijevben üzembe helyezték a **Szerhij Olekszijovics Lebegyev** (1902–1973) vezetésével létrehozott első olyan számítógépet – **MEOM** (kis elektronikus számítógép) – amely a programot a memóriájában tárolta. Ez volt az első ilyen számítógép az akkori Szovjetunióban. Paraméterei a következők voltak:

- az elektroncsövek száma 6000 körül;
- kettes számrendszert használt;
- sínszélesség – 16 számjegy és egy előjelbit;
- memória – 31 szám és 63 utasítás;
- műveleti sebesség – 3000 művelet percenként.

A számítógép első programját **Katerina Lohvinyivna Juscsenko** (1919–2001) írta (2.60. ábra).

Kijevben a számítógépek fejlesztését a kibernetikai kutatóintézetben folytatták **Viktor Mihajlovics Hluskov** (1923–1982) vezetésével. Itt hozták létre:

- a **Kijev** számítógépet (1959, fejlesztők B. V. Hnedenko, L. M. Dasevszkij, K. L. Juscsenko);



2.60. ábra.  
K. L. Juscsenko

## 2. fejezet

- a **Dnyipro** számítógép-sorozat (1961, a fejlesztés vezetője B. M. Malinovszkij);
- a mérnöki számítások elvégzésére szánt **Prominy** (1963) számítógépet;
- a mérnöki számítások elvégzésére tervezett **MIR** sorozatot a 60-as években;
- számos katonai célú számítógépet.

**Pionír** (ang. *pioneer* – első telepes, kedvezményező, kutató) – olyan ember, aki egy ismeretlen tevékenységi területen elsőként tapossa ki az utat.



2.61. ábra.

A számítástechnika pionírja érem

Az IEEE (Computer Society) nemzetközi egyesület V. M. Hluskovot 1996-ban *A számítástechnika pionírja* éremmel (2.61. ábra) tüntette ki.

A számítástechnika rakétatechnikai alkalmazásainak fejlesztésében Ukrajna több városának fejlesztői vettek részt. A szeverodonecki *Impulz* tervezőiroda munkatársai fejlesztették ki az M6000–M7000 automatikus termelésvezérlési rendszereket, az *Iszkra* billentyűzetes számítógépeket, a ballisztikus rakéták számítógépes vezérlőrendszereit. A Szovjetunióban elsőként Sztitlovodszkban gyártották a kijevi *Krisztal*

tervezőiroda által fejlesztett **Elektronika** zsebszámológépeket (2.62. ábra).

A 70-90-es években a harkovi *Hartron* által fejlesztett rakétavezérlési rendszereket a kijevi rádiógyárban állították elő.

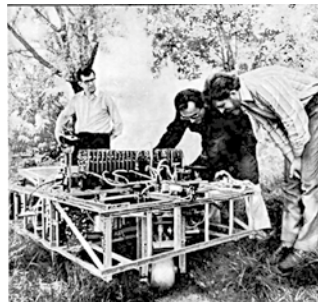
A 70-80-as években a kijevi Rádiótechnikai Intézetben fejlesztették és a *Burevisznik* gyárban állították elő a *Karat* hajó- és tengeralattjáró vezérlési rendszert.

1975-ben az Ukrán TA kibernetikai intézetében M. M. Amoszov vezetésével fejlesztették ki az első szállító robotot, a **TAIR**-t (2.63. ábra), amely képes volt a természeti akadályok kikerülésére.

Ukrán tudósok és mérnökök a 60-80-as években egy sor számítógépet terveztek különböző rakétarendszerek irányítására. A világ legnagyobb ballisztikus rakétájának, a dnyipropetrovszki déli gépgyárban előállított **P-36M2 Szatanának** a harkivi *Elektropriklad* és a kijevi rádiógyár munkatársai készítették a fellövést és repülést vezérlő számítógépeket (2.64. ábra).



2.62. ábra.  
Zsebszámológép



2.63. ábra. A tervezők a TAIR robot beállításán dolgoznak



2.64. ábra.  
A P-36M2 fellövése





**Használjuk a számítógépet!**



Végezzék el a gyakorló feladatokat a <https://cutt.ly/Uhde1FA> link, vagy a QR-kód segítségével!



**Összefoglalás**

Az első számolóeszközöket több ezer évvel ezelőtt hozta létre az emberiség. A számítástechnika fejlesztésében jelentős szerepet játszottak B. Pascal, G. Leibnitz, Ch. Babbage, A. Lovelace, H. Hollerith. Az első elektronikus számítógépeket K. Zuse (**Z4**), J. Atanasoff és C. Barry (**ABC**), H. Aiken (**Mark-1**), **Colossus** angol tudóscsoport, J. Mauchly és P. Eckert (**ENIAC**) valamint a Sz. Lebegyev vezette szovjet tudóscsoport (**MEOM**) hozták létre a XX. század 40-50-es éveiben.

A Szovjetunióban Ukrajnában hozták létre az első számítógépeket. A számítástechnika gazdasági, vezetési és katonai célú hasznosításába Ukrajnában több tíz tervező és kivitelező vállalat volt bevonva. Vezető szerepe az Ukrajna TA V. Hluskov vezette kibernetikai kutatóintézetének volt.



**Felelj a kérdésekre!**

- 1°. Nevezd meg a számítástechnika fejlődésének korszakait!
- 2°. Foglald össze, hogy változtak az adatok feldolgozásának eszközei a különböző korokban!
- 3°. Ismertesd, mi újat jelentett Pascal munkássága az ember által alkotott számolóeszközök fejlődésében!
- 4°. Miben különbözik Leibnitz számológépe a Pascalétól?
- 5°. Milyen részei voltak a Babbage által tervezett univerzális számítógépnek?
- 6°. Ismertesd a számítástechnika területén a XX. század 40-50-es évei során végbement fejlődést! Milyen tudósok játszottak kiemelkedő szerepet ebben?
- 7°. Nevezd meg az első személyi számítógépeket! Kik és mikor hozták létre ezeket?
- 8°. Mi az ukrán tudósok szerepe a számítástechnika fejlődésében? Mutasd be a számítástechnika fejlődését Ukrajnában!



**Végezd el a feladatokat!**



- 1°. Készítsd el a számítógépek fejlődésének időrendi diagramját a huszadik század közepétől kezdve a század végéig! Ehhez használd az egyik ismert alkalmazást!
- 2°. Az internet vagy nyomtatott sajtótermékek alapján keress anyagokat az első számítógépekről és töltsd ki a táblázatot!

| Tulajdonság                     | Érték |          |     |
|---------------------------------|-------|----------|-----|
|                                 | Z4    | Colossus | ABC |
| Fejlesztők                      |       |          |     |
| Elektroncsövek száma            |       |          |     |
| Kettes számrendszer alkalmazása |       |          |     |
| A program tárolása a memóriában |       |          |     |



3\*. Vessd össze az első elektroncsöves számítógépek, az amerikai **ENIAC** és az ukrán **MEOM** tulajdonságait! Az összehasonlítás eredményét táblázat formájában készítsd el egy ismert alkalmazásban!



4\*. Készíts rövid beszámolót arról, milyen fejlesztéseket hajtottak végre az ukrán tudósok az atomjégtörők vezérlése céljából!

5\*. Készíts beszámolót az első Ukrajnában létrehozott számítógépek programozásának sajátosságairól! Használd ehhez K. L. Juscsenko emlékiratait!

## 2.5. A MODERN SZÁMÍTÓGÉPEK TÍPUSAI ÉS ALKALMAZÁSA



1. A számítógépnek milyen típusait ismeritek? Miben különbözik ezek rendeltetése?
2. Milyen alkalmazásai vannak a számítógépeknek az emberek különféle tevékenységi területein?
3. Nevezetek meg olyan ukrán tudósokat, akik jelentősen hozzájárultak a számítástechnika fejlődéséhez!
4. Mik azok az okos eszközök?

### A modern számítógépek típusai

Már tudjátok, hogy manapság a leginkább elterjedt az úgynevezett **személyi számítógép**, amelyek között **asztali gépeket** és **mobil számítógépeket** különböztetünk meg. Az asztali gépek között megkülönböztetjük az **irodai**, az **otthoni** és a **játékra tervezett** gépeket. A mobil gépek közé soroljuk a **notebookokat**, **netbookokat**, **táblagépeket**, **okostelefonokat**.

A nagy számítási igényű összetett feladatokat nagy teljesítményű **szuperszámítógépek** segítségével oldják meg. Ugyanakkor a szuperszámítógépek több ezer különböző feladatot tudnak végrehajtani és a felhasználók százezreit szolgálják ki szerte a világon.

2020 közepére a világ legerősebb számítógépe a **Fugaku** számítógép volt (2.65. ábra), amelyet a **Fujitsu** hozott létre a RIKEN Számítástechnikai Központ (Kobe, Japán) számára ugyanabban az évben. Ez a központ nagy teljesítményű számításokat kutat, éghajlati modellezéseket végez, valamint kutatja a mesterséges intelligencia, a biológia, a katasztrófaelmélet és a részecskefizika területét.



2.65. ábra. **Fugaku** számítógép

A **Fugaku** számítógép Fujitsu SoC A64FX processzorokat használ, mindegyik processzor 48 magos és 3,2 GHz-es.

A processzormagok száma a számítógép összes processzorában körülbelül 7,3 millió. Ez a számítógép másodpercenként  $4 \cdot 10^{17}$  műveleti sebességgel képes adtműveleteket végrehajtani.

**Szervereknek** speciális számítógépeket használnak. Az ilyen számítógépek nagyobb teljesítményűek és több adatot tárolhatnak, mint a személyi számítógépek. A rendeltetésétől és a felhasználók számától függően, akiknek az ilyen szerverek szolgáltatásokat nyújtanak, egyetlen dedikált szerver processzort (például iskolai e-mail





szervert) vagy több száz vagy ezer ilyen processzort (például banki vagy internetszolgáltató szervert) használhatnak.

A legnépszerűbb személyi számítógépek között vannak olyan kézi számítógépek, amelyek egyesítik a mobiltelefon és a számítógép funkcióit, ezek az **okostelefonok** (2.66. ábra). Ukrajna lakossága által használt okostelefonok száma az utóbbi időben évente csaknem 20%-kal nő. Napjainkban az okostelfont nem csak kommunikációs eszközként használják, hanem az internetes szolgáltatások használatára is.



2.66. ábra.  
Okostelefon

Érdekes tendenciát mutat az okostelefonok használata Ukrajna azon falvaiban és városaiban, ahol nincs vagy nagyon gyenge az internetkapcsolat. A **Kyivstar** mobilszolgáltató felmérése szerint 2020-ban a falusiaknál naponta átlagosan 3-4-szer nagyobb az internetes adatforgalom, mint a nagyvárosok lakóinál. Például Losakova Huta faluban (Csernyihiv megye) minden okostelefon-tulajdonos átlagosan naponta 41 380 MB mobilinternetet használt fel, míg Szkadovszkban (Herszon megye) – 16 620 MB, Dnyiproban pedig csak 9 360 MB.

A [wearesocial.com](http://wearesocial.com) oldal adatai szerint 2020 elején a felhasználók több mint 53%-a használt okostelefont az internetezéshez. Átlagosan egy felhasználó egy nap 7 órát használja az internetet, ez az ébren töltött időnk 40%-a. Az okostelefonok és az internet hirtelen és gyökeresen megváltoztatják az emberek életét. Az okostelefonok fő tulajdonságait és azok közelítő értékeit 2020 közepére a 2.7. táblázat mutatja.

2.7. táblázat

**Az okostelefonok fő tulajdonságainak értéke**

| <i>Tulajdonság</i>         | <i>Érték</i>                            |
|----------------------------|---|
| Operatív memória (RAM), GB | 4–12                                    |
| Beépített memória, GB      | 64–512                                  |
| A processzor típusa        | Exynos 9611; Kirin 810; Apple A13       |
| Kijelző méret, inch        | 4,6–6,8                                 |
| Hátlapi kamera, Mpixel     | 16–64                                   |
| Adatátvitel                | 4G; 5G                                  |
| Operációs rendszer         | Android; iOS                            |
| Gyártó                     | Samsung; Xiaomi; Apple; HUAWEI; OnePlus |

**Számítógépek használata**

Manapság a számítógépet széles körben alkalmazzák a tudományban, az ipari termelésben, a hivatalokban, a bankrendszerben, az oktatásban, a gyógyításban, a közlekedésben, a kommunikációban, a mezőgazdaságban, a szociális ellátó rendszerben és a gazdaság más területein.

A tudományban a számítógépeket többek között számítógépes kísérletek elvégzésére használják. A tudományos kísérletek egy része meglehetősen sok anyagi, műszaki, energetikai problémát vet fel. A számítógépes modellezés sajátosságait már a 7. osztályban megtanulták.

Fontos szerepet játszik a számítógép a termelésben. A különböző termékek modellezése és számítógépes tervezése csökkenti a tervezés idejét, javítja a minőséget és a termelés hatékonyságát, csökkenti az árakat. A számítógép alkalmazása előtt egy új autó tervezésétől a gyártásig 5-6 év telt el, ami manapság 1 év alá csökkent.

A szolgáltatások területén a számítógépeket különféle adatok – szövegek, táblázatok, adatbázisok, képek, fotók, multimédiás adatok tárolására és feldolgozására használják. Nehéz lenne ma már olyan munkahelyet találni, ahol ne használnának számítógépeket. A könyvtárak és kórházak kartotékjait adatbázisok váltották ki, amelyek kezelése sokkal kényelmesebb a hagyományos kartotékoknál. A könyvelő a számítógépe segítségével néhány perc alatt elvégzi azt a munkát, amire régebben órákra vagy napokra volt szükség. A bankár anélkül követheti a tőzsdei árfolyamokat, hogy elmozdulna az asztala mellől és néhány egérgattintással intézi az átutalásokat.



2.67. ábra.

Az állam okostelefonban

Az okostelefonok egyre nagyobb szerepet játszanak az emberi élet minden területének számítógépesítésében. Vásárlásra az interneten, otthoni „inteligens” eszközök kezelésére, adatok cseréjére használják.

Például Ukrajnában 2019-ben elindították *Az állam okostelefonban* nevű programot (2.67. ábra).

E program keretében lehetőség van arra, hogy a felhasználók a kormányzati és pénzügyi intézményekben, valamint különböző irodákban személyes jelenlét nélkül tudnak:

- bankszámlát nyitni;
- készpénz nélkül fizetni az állami és önkormányzati intézmények különféle szolgáltatásaiért;
- digitális aláírás segítségével jelentéseket küldeni az állami szerveknek;
- elektronikus formában megkapni a különféle gépek és berendezések garancia-kártyáit;
- hozzáférni a különféle nyilvántartásokhoz:
  - nemzetközi szállítmányozási engedélyek;
  - földhivatalok;
  - erdészet;
  - gyógyszerek és az azokra vonatkozó igazolások érvényessége stb.

Az „okos ház” vezérléséhez nincs szükség további számítógépekre, elegendő egy okostelefon és egy eszköz, amely az összes „inteligens” eszközt egyetlen egységbe egyesíti. Egy ilyen eszköz a **hub** (ang. *hub* – tevékenységi központ). A 2.68. ábra az okos ház eszközeinek csatlakoztatásának egyik lehetőségét mutatja be. Az okostelefon segítségével a háztulajdonos megteheti, hogy:

- adatokat kapjon a házon belüli biztonsági jellemzőkről – nincs-e tűz (füstérzékelő, 2.68. 7. ábra); vízáttörés (vízérzékelő, 2.68. 6. ábra), gázszivárgás (gázérzékelő, 2.68. 4. ábra), behatoltak-e idegenek a házba (mozgásérzékelő, 2.68. 1. ábra, és webkamera, 2.68. 11. ábra) stb.;
- a háztartási gépeket vezeték nélküli kapcsolók segítségével vezérli – robotpor szívó (2.68. 13. ábra), légtisztító (2.68. 12. ábra), légkondicionáló vagy más eszköz, vezeték nélküli kapcsoló segítségével (2.68. 5. ábra);



- |                                       |   |
|---------------------------------------|---|
| 1. Mozgásérzékelő                     | 9. Vezeték nélküli világításszabályzó               |
| 2. Padlófűtés                         | 10. Okos lámpák                                     |
| 3. Hőérzékelő                         | 11. Webkamera                                       |
| 4. Gázérzékelő                        | 12. Légtisztító                                     |
| 5. Vezeték nélküli kapcsoló           | 13. Robotporszívó                                   |
| 6. Vízérzékelő                        | 14. Okostelefon a megfelelő vezérlési alkalmazással |
| 7. Füstérzékelő                       |   |
| 8. A ház Hub-ja (okos otthon központ) |   |

2.68. ábra. Az okos ház berendezéseinek vezérlőrendszere

- vezeték nélküli kapcsolóval vezéri a ház világítását (2.68. 9. ábra) és okos lámpákkal (2.68. 10. ábra).

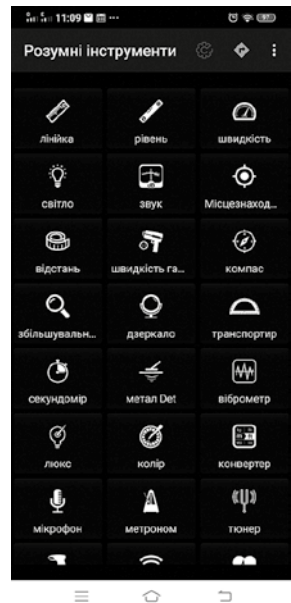
Ezen eszközök ma már megvásárolhatóak és használhatóak otthon.

Anélkül, hogy elhagynánk otthonunkat, munkába vagy hazafelé tartanánk, rendelhetünk ételt, árut, buszjegyet, foglalhatunk szállodát, turisztikai utakat, fizethetünk a közüzemi szolgáltatásokért, időpontot egyeztetethetünk orvosunkkal.

### Okostelefonok használata az oktatásban







Már tudjátok, hogyan lehet segítségetekre a számítógép a tanulásban, a karantén ideje alatt aktívan használtátok az internet különböző oktatási tananyagait. Bizonyos okostelefonos alkalmazásokat használva azonban bárhol és bármikor tanulhatsz.

Lehetőség van különböző fizikai és matematikai kutatások elvégzésére otthon vagy a természetben. Például az **Android** operációs rendszer egyik alkalmazása a **Smart Tools**. A következő eszközökkel rendelkezik (2.69. ábra):



2.69. ábra. Inteligens eszközök kezelését vezérlő alkalmazás

## 2. fejezet

- **skála**  – kisméretű tárgyak mérése az okostelefon képernyőjén keresztül;
- **vízmérték**  – a vízszintes és függőleges síkok pontos elhelyezésének megállapítása;
- **távolság**  – a tárgyak közötti távolság beállításához;
- **iránytű**  – a Föld égtájainak meghatározásához;
- **fénymérő**  – a megvilágítás szintjének mérésére;
- **fém-detektor**  – fémek keresésére.

Az operációs rendszerek egyes verzióiban az „intelligens” eszközök alapértelmezés szerint telepítve vannak. Ha ez nincs telepítve, akkor lehetőség van az ingyenes alkalmazások letöltéséhez az *Áruház*on keresztül.

Egy másik érdekes tanulásra alkalmas eszköz egy olyan alkalmazás, amely egy adott, általában síkon árázolt objektumot egyesít, további virtuális elemekkel, háromdimenziós képpel vagy háromdimenziós animációval (2.70. ábra).



2.70. ábra. Egy mamut animálása kiterjesztett valóságban

Bizonyos alkalmazások használatával (például JigSpace, Google Lens, Atom Visualizer, Civilisations AR, Futurio App, Skyscrapers AR) különböző szögekből tekinthetünk meg történelmi épületet, szobrot, megnézhetjük, hogyan működik az emberi test, valamint ellátogathatunk különböző múzeumok kiállításaira stb. A lényeg az, hogy sehova se kell mennünk, csak okostelefonnal és megfelelő alkalmazással kell rendelkezünk. Magunk is készíthetünk virtuális anyagokat.

### A számítógéphasználat fő területei

A számítástechnika főbb alkalmazási területeit így foglalhatjuk össze:

- nagy mennyiségű és nagy pontosságú számítások elvégzése;
- számítógépes modellezés és számítógépes kísérletek;
- automatizált vezérlőrendszerek működtetése;
- nagy mennyiségű adat tárolása és feldolgozása;
- gyors információcsere;
- ipari, háztartási és katonai eszközök irányítása;



- a tanulás támogatása;
- diákok, szakemberek távoktatásának megszervezése, a mozgáskorlátozott emberek tanulásának támogatása.

## A számítógép igény szerinti konfigurálása

A számítógép konfigurációját rendeltetésének megfelelően kell kialakítani. A következő paraméterekre kell figyelmet fordítani:

- **a processzor számítási teljesítménye**, amit az órajel frekvenciája, a magok száma, a második és harmadik szintű cache mérete határoz meg. Minél nagyobbak ezek az értékek, annál nagyobb a processzor teljesítménye;
- **a RAM kapacitása** – minél nagyobb, annál nagyobb a gép teljesítőképessége;
- **a merevlemez kapacitása** – attól függ, mekkora adatmennyiséget kell a felhasználónak feldolgozni, audio- és videofeldolgozáshoz lényegesen több szükséges, mint szövegszerkesztéshez;
- **külső videokártya** – erősen befolyásolja a videók feldolgozásának sebességét. A videó RAM mérete és a grafikus processzor teljesítménye a döntő fontosságú két tényező;
- **a monitor képminősége**, ami a méretétől, a felbontásától és a válaszeitől függ.

A számítógép kiválasztására nyilván egyéb tulajdonságok is befolyással vannak, például SSD meghajtó szükségessége, az alaplap sajátosságai, a hálózati tápellátás nélküli üzemidő (mobil eszközöknél), csatlakoztathatóság különböző hálózatokhoz. A konfiguráció kialakítására természetesen annak az ára is lényegesen kihat.

Általában a számítógépek, valamint más termékek kiválasztásakor az ár és a funkcionalitás (minőség) közötti optimális arányról beszélnek. Nincs értelme a játékokra használt számítógépet irodai munkára (szöveges dokumentumok előkészítésére, üzleti grafikák feldolgozására, táblázatokkal való munkára) használni. Egy ilyen nagy teljesítményű számítógép erőforrásait még 10%-ban sem fogjuk tudni felhasználni, a költségek pedig akár tízszer magasabbak lehetnek. Az otthoni számítógépeket az irododai munkák mellett multimédiás tartalmak (videók és zene, fényképek) feldolgozására is használják, valamint aktívan használják az interneten folytatott kommunikációhoz, a távoktatáshoz és egyebekhez is.

Az internetáruházak nagy része lehetővé teszi a számítógép összeállításának kialakítását intelligens szűrők alkalmazása által. A **Hotline** (<https://hotline.ua/computer/nas-tonnye-kompyutery>) rendeltetéstől függően a következő csoportokat különbözteti meg: *kezdő számítógép, munkagép, munkaállomás, optimális alapkonfiguráció, univerzális optimális konfiguráció, haladó optimális konfiguráció, nagy teljesítményű játékgép*. A **Rozetka** (<http://rozetka.com.ua/computers-notebooks>) három csoportot különböztet meg: *kezdő, munka és tanulás, játék*.

A rendszeregységek árai 3,5 ezertől 400 ezer hrivnya között változik (2020-as évi adatok). A 2.8. táblázatban néhány, a fenti csoportosításnak megfelelő gép paramétereit foglaltuk össze.

## Különböző rendeltetésű számítógépek paraméterei

| Tulajdonság            | Érték                          |                         |                            |
|------------------------|--------------------------------|-------------------------|----------------------------|
|                        | Irodai (tanulásra és munkára)  | Otthoni                 | Játék                      |
| Típus                  | Asztali                        | Asztali                 | Asztali                    |
| Processzor             | AMD A6-9500                    | Intel Core i5-7400      | Intel Core i9-9900K        |
| Órajel frekvencia, GHz | 3,5                            | 3                       | 3,6                        |
| RAM, Gb                | 8                              | 16                      | 64                         |
| Merevlemez, Gb         | 1000                           | 2000                    | 4000                       |
| Videókártya            | Integrált AMD Radeon R5 Series | NVIDIA GeForce GTX 1050 | NVIDIA GeForce RTX 2080 Ti |
| VideoRam, Gb           | Nincs                          | 2                       | 11                         |
| Hangkártya             | Integrált                      | Integrált               | Integrált                  |
| Monitor                | LG 19M45A                      | ASUS VA24EHE            | Samsung Curved C49HG90DMI  |
| Képtároló, inch        | 21,5                           | 23,8                    | 49                         |
| Felbontás              | 1920 x 1080                    | 1920 x 1080             | 3840 x 1080                |
| Válaszidő, mp          | 5                              | 5                       | 1                          |



### Használjuk a számítógépet!



Végezzétek el a gyakorló feladatokat a <https://cutt.ly/Mhde2rC> link vagy a QR-kód segítségével!



### Összefoglalás

A számítógépeket rendeltetésük és számítási teljesítményük alapján **szuperszámítógépekre** és **személyi számítógépekre** oszthatjuk. Az utóbbiak között megkülönböztetünk **asztali** és **mobil** számítógépeket. Az asztali gépek között vannak **irodai**, **otthoni** és **játékgépek**. A mobil eszközök közé a **notebookok**, **netbookok**, **táblagépek** és **okostelefonok** tartoznak.

Az okostelefonok egyre nagyobb szerepet játszanak az emberi élet minden területének számítógépesítésében, a legtöbb online vásárláshoz, az „inteligens” otthoni eszközök kezeléséhez, a hálózatokon történő adatcseréhez stb. Az okostelefonokon lévő alkalmazások segítségével különféle fizikai és matematikai kutatásokat végezhetünk, hozzáférhetünk az internet oktatási anyagaihoz.

Attól függően választjuk ki az összetevőit (konfigurációját), hogy a számítógépet mire fogjuk használni. Ugyanakkor különös figyelmet kell fordítanunk a következő tulajdonságok értékeire: a processzor teljesítményére, a RAM-ra, a merevlemez-meghajtó





kapacitására (okostelefonok esetében – a külső memória mennyiségére), külön video adapter jelenlétére, a videokártyára, a monitor által megjelenített kép minőségére.







### Felelj a kérdésekre!

- 1<sup>o</sup>. Mit nevezünk személyi számítógépeknek? Milyenek lehetnek ezek?
- 2<sup>o</sup>. Rendeltetésük szerint hogyan csoportosíthatjuk a személyi számítógépeket? Mit tudsz a számítógépek egyéb tulajdonságok szerinti besorolásáról?
- 3<sup>o</sup>. Mi a különbség a különféle típusú hordozható számítógépek között?
- 4<sup>o</sup>. Mi az a szuperszámítógép? Milyen célokra használják őket? Melyek ezen számítógépek tulajdonságainak fő értékei?
- 5<sup>o</sup>. Mi az okostelefon? Miért kapta ezt a nevet, és milyen célokra használják?
- 6<sup>o</sup>. A világon egyre több felhasználó miért használ okostelefont az internetezésre? Szerinted ez a tendencia a jövőben is folytatódik?
- 7<sup>o</sup>. Miért van annyi különbség a különböző számítógépek tulajdonságaiban? Mondj példákat a különböző típusú számítógépek osztályozására!
- 8<sup>o</sup>. Milyen példákat ismersz az okostelefonok alkalmazására? Használtál már okostelefont erre a célra?
- 9<sup>o</sup>. Hozz fel példákat arra, hogy hol alkalmazzák a számítógépeket!
- 10<sup>o</sup>. Mi befolyásolja a számítógép kiválasztását különböző alkalmazási területek esetében?



### Végezd el a feladatokat!

- 1<sup>o</sup>. Készítsd el a számítógépek osztályozásának vázlatát egy tetszőleges, általad ismert alkalmazás segítségével!
-  2<sup>o</sup>. Válaszd ki az otthoni médiaközpontként használni kívánt számítógépet! Indokold meg a választásod!
- 3<sup>o</sup>. Készíts gyakorlati munkát a 7. osztályosok számára például *A fizikai mennyiségek mérése* témakörben! A feladatot okostelefonod segítségével oldd meg!
-  4<sup>o</sup>. Készíts beszámolót a kiterjesztett valóság alkalmazásának lehetőségeiről a 8. osztályos tantárgyak ismeretanyagának keretében!
-  5<sup>o</sup>. Készíts válogatást okostelefonok kiterjesztett valóság alkalmazásaiból, amelyek oktatási célokra használhatók! Az eredményeket prezentáció vagy video podcast formájában mutasd be!
-  6<sup>o</sup>. Tudj meg többet a ma létező legerősebb szuperszámítógépekről! Készíts egy táblázatot az öt legerősebb szuperszámítógép értékeivel! Írd le azt is, hol használják a legerősebb szuperszámítógépet!
- 7<sup>o</sup>. Hozz létre egy táblázatot a szövegszerkesztőben a 2.8. táblázatnak megfelelően! Az online áruházakból származó adatok felhasználásával töltsd ki a táblázatot, a különböző célokra használt személyi számítógépek tulajdonságainak aktuális értékeivel! Elemezd, mely tulajdonságértékek változtak jelentősen! Miért történhetett ez?

## 2. SZÁMÚ GYAKORLATI MUNKA

## Számítógép-konfiguráció kialakítása igény szerint

**Figyelem!** A számítógép használata során tartsd be a balesetvédelmi szabályokat és az egészségügyi előírásokat!

- Állíts össze számítógépes konfigurációt a következő szakmákban dolgozók számára:
  - egy írónak, aki a dolgozószobájában írja a műveit, végzi az elektronikus levelezését, és anyagokat keres az interneten;
  - egy diáknak, aki a feladatait készíti a számítógépen, létrehozza és karbantartja a saját fonotékáját, oktatófilmeket hoz létre és szerkeszti azokat;
  - egy internetes kiadvány tudósítójának, aki a világ legkülönbözőbb helyéről rendszeresen küldi tudósításait!
- Az előbbi feladat alapján a szövegszerkesztőben készíts el, majd töltsd ki a következő táblázatot!

| Tulajdonság               | Érték             |                    |                        |
|---------------------------|-------------------|--------------------|------------------------|
|                           | író számítógépére | diák számítógépére | tudósító számítógépére |
| Típus                     |                   |                    |                        |
| Processzor                |                   |                    |                        |
| Órajel frekvencia, GHz    |                   |                    |                        |
| RAM-kapacitás, Gb         |                   |                    |                        |
| Merevlemez kapacitása, Gb |                   |                    |                        |
| Monitor                   |                   |                    |                        |
| Képtároló, inch           |                   |                    |                        |
| Felbontás                 |                   |                    |                        |
| Választidő, mp            |                   |                    |                        |

- Ugyanebben a fájlban indokold meg a választásaidat!
- A kapott fájlt ments el a saját mappádba **praktikai 2.docx** néven, majd a fájlt küldd el elektronikus levélben a tanárodnak!

# 3. fejezet

## Szövegfeldolgozás

*Ebben a fejezetben megismerkedtek:*

- mind azzal, amit a szövegszerkesztők használatáról tudni kell
- a szöveges dokumentumok formátumainak sokféleségével
- azzal hogyan illeszthetünk a szövegbe olyan karaktereket, amelyeket nem találunk a billentyűzetben
- azzal hogyan lehet szöveget és egyéb elemet keresni és cserélni
- azzal hogyan kell egyszerre több dokumentummal dolgozni
- azzal hogyan használhatjuk a számozást, felsorolást, többhasábos tördelést, fejléctet, lábléctet
- azzal hogyan alkalmazhatjuk a stílusokat
- azzal hogyan ellenőrizzük a dokumentum szerkezetét és hogyan kell létrehozni tartalomjegyzéket
- azzal hogyan illesztünk hivatkozásokat a szöveges dokumentumokba
- azzal hogyan kell dokumentumokat megosztani
- azzal hogyan illesztünk a szövegbe táblázatot, képletet, képeket

### 3.1. Szövegfeldolgozó rendszerek. Szövegtörédek keresése és cseréje



1. Mi a *szövegszerkesztő* rendeltetése? Mire szolgál?
2. Milyen objektumok vannak a szövegszerkesztőben? Milyen tulajdonságokkal rendelkeznek?
3. Milyen módon jelölhetünk ki szövegrészeket a szövegszerkesztőben?

#### *A szövegfeldolgozók rendeltetése és főbb jellemzői*

Az 5. osztályos informatikai tananyagból emlékezhetek rá, hogyan kell szöveget létrehozni és szerkeszteni **Word 2010**-ben: megismerkedtünk a beírás, szerkesztés, formázás fortélyáival, a karakterek és a bekezdések tulajdonságaival, grafikai elemek és táblázatok beillesztésével, valamint a szöveges dokumentum kinyomtatásával.

A **Word** szövegszerkesztő mellett számos különféle alkalmazás létezik, amelyeket szöveg létrehozására és feldolgozására terveztek, ezeket **szövegfeldolgozó rendszereknek** hívják. Mindegyik általános célú alkalmazás szoftverhez tartozik. Az alkalmazás megválasztása a felhasználó igényeitől függ.

Ezekkel az alkalmazásokkal létrehozhatjuk, elmenthetjük a szöveges dokumentumokat és könnyedén módosíthatjuk a korábban elkészítettet. Átrendezhetjük a szövegrészeket, átmásolhatjuk azokat a dokumentum különböző részeire, beilleszthetjük

más dokumentumok részeit a szövegbe, és szükség esetén törölhetjük a végrehajtott módosításokat. A legtöbb szövegszerkesztő automatizálja a helyesírási és szintaktikai hibák keresését. Ezeknek az alkalmazásoknak a használata nagyban megkönnyíti a szöveg szerkesztését, képletek, ábrák, táblázatok stb. beillesztését. Egy szöveges dokumentum egyszeri létrehozásával bármikor tetszőleges számú példányt kaphatunk mind elektronikus, mind pedig papír formában.

Legtöbbjük támogatja a **WYSIWYG** (ang. *What You See Is What You Get* – amit látsz, azt kapsz) technológiát, amely lehetővé teszi, hogy a dokumentumot a képernyő ugyanabban a formában jelenítse meg, ahogyan kinyomtatják.

Szövegfeldolgozó rendszereket használnak az emberi tevékenység minden területén különféle dokumentumok, nyomtatott kiadványok, füzetek, plakátok, könyvek szerkesztésénél.

Ezekben a programokban a feldolgozás fő tárgya egy **szöveges dokumentum**, amely különböző objektumokat tartalmazhat: karaktereket, szavakat, mondatokat, bekezdéseket, oldalakat, ábrákat, táblázatokat, diagramokat stb. A szöveges dokumentum minden objektumának megvannak a maga tulajdonságai (3.1. táblázat).

3.1. táblázat

### A szöveges dokumentum objektumai



| Objektum | Az objektum tulajdonsága   |
|----------|--|
| Karakter | Betűtípus, méret, szín, aláhúzás, módosítás, eltolás, karakterek közötti távolság stb.           |
| Bekezdés | Behúzás a bal és a jobb margó szélétől, az első sor behúzása, igazítás, sorköz, bekezdésköz stb. |
| Oldal    | Margók, tájolás, méret, számozás, fejléc, lábléc stb.  |
| Kép      | Típus, méret, színek, elhelyezés, szöveg körbefuttatása stb.                                     |
| Táblázat | Sorok és oszlopok száma, cellaméret stb.   |





A szöveg külalakjának a megváltoztatását szövegformázásnak nevezzük.

A leggyakoribb szöveges fájlformátumok: a **TXT, RTF, DOCX, PDF, ODT** és egyéb. Ezeknek a szövegformátumoknak a jellemzőit a 3.2. táblázat tartalmazza.

3.2. táblázat

### A szöveges dokumentumok formátumainak jellemzői

| Kiterjesztés | Ikon  | Jellemzők  |
|--------------|---|--|
| TXT          |  | A fájlban csak formázatlan, bekezdésekre tördelt szöveg tárolható. Különböző szövegszerkesztők dolgozhatják fel, amelyek különböző operációs rendszerekkel működhetnek.  |
| DOCX         |  | Formázott szöveg mellett gyakorlatilag bármilyen média formátumot tartalmazhat. A többi szöveges formátumhoz képest tömörebb tárolást (akár 70%-osat) tesz lehetővé. A 2007-es verziótól kezdve a <b>Word</b> szövegszerkesztők alapértelmezett fájlformátuma. |

| Kiterjesztés | Ikon  | Jellemzők   |
|--------------|---|---|
| RTF          |  | A fájl szöveget, képeket, beillesztett objektumokat és azok tulajdonságainak értékeit tárolja. Különböző operációs rendszerekben működő szövegszerkesztők által támogatott. |
| PDF          |  | Nem szerkeszthető, megtekintésére <b>Adobe Reader</b> vagy a böngészőket használják. A fájl a formázott szöveget és az összes beillesztett objektumot tárolja.              |
| HTML         |  | Szöveges dokumentumok weblap formátumban történő mentése. Az ikon képe attól függ, hogy melyik az operációs rendszer alapértelmezett böngészője.                            |
| ODT          |  | Az <b>Open Writer</b> szövegszerkesztő által létrehozott szövegfájlok szabványos formátuma.   |

A 8. osztályban folytatjuk a szövegszerkesztők tanulmányozását a **Microsoft Word** alkalmazás példáján. A tankönyv anyaga leírja az **Office 365**-nek a működését. Ez a leírás alkalmazható a **Word** minden verziójára a 2013-astól kezdve. A tankönyv elektronikus melléklete a **Microsoft Word 2010**-es verziójához tartalmaz anyagokat.

A **Word** szövegszerkesztővel készített szöveges dokumentumok alapértelmezett fájlformátuma a **DOCX**, de a program más szöveges dokumentumformátumokat is támogat. Szöveges dokumentum nem szabványos formátumban történő mentéséhez nyissuk meg a **Fájl típusa** listát a **Dokumentum mentése** ablakban, és válasszuk ki a kívánt fájlformátumot!

### Speciális karakterek elhelyezése a szövegben

Már megtanultátok, hogyan kell szöveget létrehozni és szerkeszteni, megismertek a beírás, szerkesztés, formázás fortélyaival, a karakterek és a bekezdések tulajdonságaival. A szövegszerkesztőben azonban rengeteg egyéb művelet is végezhető. A szövegszerkesztőben a szövegben olyan szimbólumokat is elhelyezhetünk, amelyek nincsenek a billentyűzeten.

A **Word szövegszerkesztő**ben más nyelvek betűit, matematikai jeleket, egyezményes jeleket is beilleszthetünk. Ilyenek például a © ≤ β ∑ € ∞ ≈ ☺ ♪ §. Erre szükségetek lesz, amikor matematika-, fizika-, kémiaórára készültök. Csakúgy, mint a **Wordbe**, ezeket a karaktereket **PowerPoint** alkalmazásban létrehozott prezentációkba is be lehet illeszteni.

Az ilyen jeleket a következő algoritmus szerint illesszük be:

1. Helyezzük a kurzort a szövegben a beszúrás helyére!
2. Kattintsunk a **Beszúrás** fülön a **Szimbólum** csoport **Szimbólum**

gombjára!

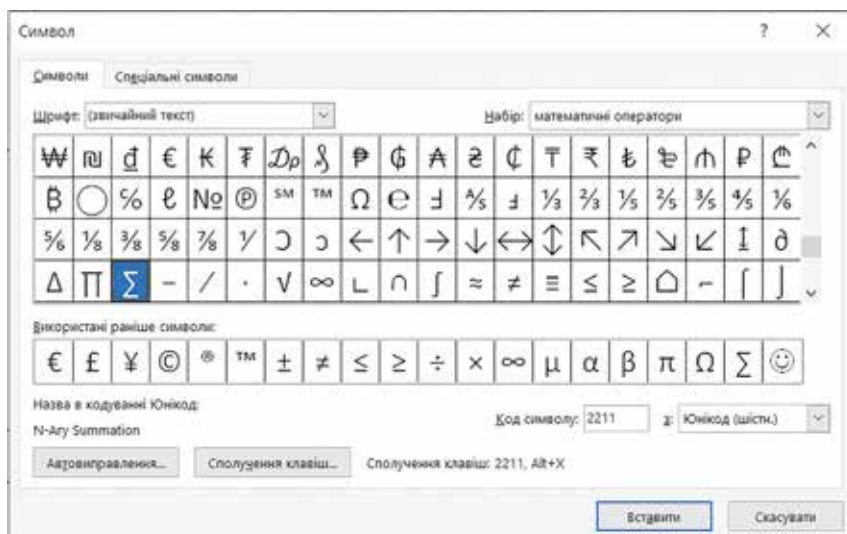
3. Jelöljük ki a megnyíló párbeszédablakban (3.1. ábra) a megfelelő szimbólumot!



3.1. ábra.  
A szimbólumok listája

Amennyiben a szükséges szimbólumot nem találjuk a listán:

1. Kattintsunk a **További szimbólumok** utasításra!
2. Tekintsük meg a megnyíló párbeszédablakban (3.2. ábra) a **Szimbólumok** és a **Különleges karakterek** fülek tartalmát! Szükség esetén kiválaszthatjuk a **Betűtípust**, ez megváltoztatja az elérhető szimbólumok listáját.
3. Válasszuk ki a szimbólumot!
4. Kattintsunk a **Beszúrás** gombra!



3.2. ábra. A Szimbólum ablak

Ha egy szimbólum beillesztésére gyakran van szükség, a **Szimbólum** ablakban külön **Billentyűkombinációt** rendelhetünk hozzá (a **Szimbólum** ablak **Billentyűparancs** gombjának lenyomása után ki kell tölteni a párbeszédablak mezőit). Ezután a billentyűparancs segítségével gyorsan beilleszthető a szimbólum.

A **Word**ben rajzzjellegű szimbólumokat is beilleszthetünk a szövegbe, például ilyeneket:



Ezeket a szimbólumokat az **MS Outlook**, **Webdings**, **Wingdings** stb. betűtípusokban találjuk. A billentyűzet megnyomásakor betűk helyett kis ikonok kerülnek a szövegbe.

A speciális szimbólumok tartalmazzák azokat a formázási jeleket, amelyeket már az 5. osztályban használtál. A **formázási jelek** közé azokat a jeleket soroljuk, amelyeket a felhasználó visz be a beírás során, de nyomtatásban nem jelennek meg. Ha azonban bekapcsoljuk ezek megjelenítését, akkor speciális jelekként megjelennek a képernyőn (3.3. táblázat). Ha megjelenítjük a formázási jeleket is (**Kezdőlap** menü, **Bekezdések** fül, **Minden látszik** gomb ¶¶), könnyebben megérthetjük a dokumentum szerkezetét. Ha látjuk a **Bekezdés vége** ¶ jeleket, pontosan tudjuk, hol végződik a bekezdés és hol kezdődik a következő.

A szöveg szerkesztése és a karakterek formázása a már régebben megismert módzerekkel történik.



## Formázási jelek

| Szimbólum          | Billentyű     | Jel | Szimbólum           | Billentyű          | Jel                       |
|--------------------|---------------|-----|---------------------|--------------------|---------------------------|
| Szóköz             | Szóköz        | ·   | Nem törhető szóköz  | Ctrl+Shift+sóköz   | °                         |
| Bekezdés           | Enter         | ¶   | Nem törhető kötőjel | Ctrl+Shift+kötőjel | –                         |
| Tabulátor          | Tab           | →   | Sortörés            | Shift+Enter        | ↵                         |
| Feltételes kötőjel | Ctrl+kötő-jel | ¬   | Oldaltörés          | Ctrl+Enter         | .....Розрив сторінки..... |



## Ha többet szeretnél tudni

Vannak olyan szimbólumsorozatok, amelyeket a bevétel során a **Word** automatikusan egy másik szimbólumra cserél. Ezekre láthatunk néhány példát a 3.4. táblázatban.

3.4. táblázat

## Speciális szimbólumok bevétele


| Szimbólumsorozat        | Helyettesítő jel | Szimbólumsorozat | Helyettesítő jel |
|-------------------------|------------------|------------------|------------------|
| Szóköz, kötőjel, szóköz | – (gondolatjel)  | Pont, pont, pont | ...              |
| (c)                     | ©                | -->              | →                |
| (r)                     | ®                | <--              | ←                |
| (tm)                    | ™                | <==              | ↩                |
| :)                      | ☺                | <=>              | ↔                |
| :                       | ☹                | ==>              | ➔                |

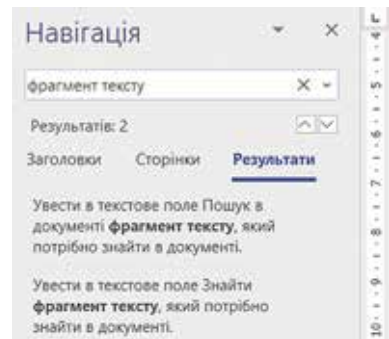
A szimbólumsorozatok automatikus cseréje a **Word**ben alapértelmezett. A felhasználó a **Szimbólum** párbeszédablak **Automatikus javítás** gombjával saját elemeket adhat ehhez a listához.

## Szöveg keresése és elemek cseréje

A **Word** szövegszerkesztő automatikusan megkereshet a dokumentumban egy szövegrészt (karaktert, szót, kifejezést). Például ellenőrzi, hogy előfordul-e egy adott szó vagy kifejezés a szövegben.


Ehhez:

- Válaszd a **Keresés**  **Пошук** gombot **Kezdőlap** menü **Szerkesztés** fülében, amely megnyitja a keresőmező **Navigációs** paneljét (3.3. ábra)!
- Írj be a **Keresés a dokumentumban** szövegmezőbe egy szövegrészletet, amelyet megszeretnél találni!



3.3. ábra. A keresés eredményeinek **Navigációs** panelje

Ennek eredményeként a **Navigációs** panel megmutatja egy adott szövegrészlet előfordulásának számát a szövegben, a szükséges szövegrészt tartalmazó mondatokat, és megjelennek a talált szövegrészek közötti mozgáshoz szükséges gombok

 A szövegben az összes talált szövegrész színessel lesz kiemelve.

Speciális keresésre is lehetőség van a szövegszerkesztőben.

Ehhez:

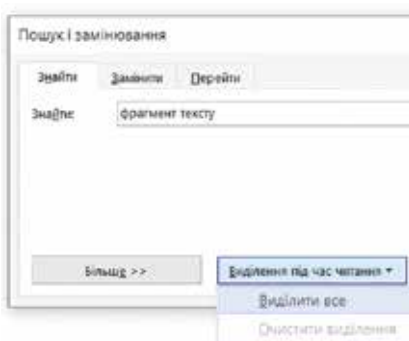
1. Válaszd a **Kezdőlap** menü **Szerkesztés** fülének **Irányított keresés** gombját,

ami a **Keresés és Cseré**

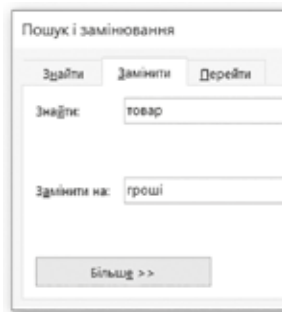


ablakot nyitja meg (3.4. ábra)!

2. Írd be a **Keresett szöveg** mezőbe azt a szövegrészt, amit meg szeretnél találni!
3. Szükség szerint hajtsd végre az egyik műveletet:
  - kattints a **Következő** gombra ahhoz, hogy a részletet tartalmazó minden bejegyzést egymás után megkereshesd;
  - végezd el a következő műveleteket: **Kiemelés olvasáshoz** ⇒ **Az összes kiemelése**, a megadott szövegrészlet összes előfordulásának kijelölése!



3.4. ábra. A **Keresés** ablak




3.5. ábra. A **Cseré** ablak

Ezenkívül az **Egyebek** gombra kattintva megadhatjuk a keresési paramétereket: keresési irány (lefelé, felfelé, mindenütt), kis- és nagybetűk megkülönböztetése, szövegrészlet vagy teljes szó stb.

A **Word** szövegszerkesztőben nemcsak szövegrészeket kereshetünk, hanem különféle speciális karaktereket is (bekezdésjel, tabulátor, szakasz vége, oldaltörés stb.), valamint egy adott típusú szövegre (betűtípus, szín, nyelv stb.). Egy ilyen keresés végrehajtásához a párbeszédpanelen válasszuk a **Speciális** vagy **Formátum** gombokat, és válasszuk ki a kívánt feltételt a megadott listából.

Ne feledd, hogy a keresendő szöveg beírásakor gondosan figyelj a beírás helyességére, mert a keresés alapja a karakterek teljes egybeesése. A keresési művelet nem vonatkozik a dokumentum azon szövegrészeire, amelyeket a beágyazott objektumok (rajzok, diagramok stb.) tartalmaznak.

Ha a dokumentumban szükség van a szöveg egy részének kicserélésére, akkor a következő algoritmust hajtsuk végre:

1. Válasszuk a **Kezdőlap** menü **Szerkesztés** fülének **Cseré**  gombját!
2. A **Keresett szöveg** szövegmezőjébe (3.5. ábra) írjuk be a keresendő szót (például *áru*)!
3. A **Cseré erre** szövegmezőbe írjuk be azt a szót, amire le szeretnénk cserélni (például *pénz*)!

Ahhoz, hogy a csere a dokumentum egészében automatikusan menjen végbe, válasszuk **Az összes cseréje** gombot és a választott cseréhez kattintsunk a **Következő** gombra, szükség esetén pedig a **Csere** lehetőségre!


A szövegszerkesztők ezen szolgáltatása sok esetben hasznos, például egy vezetőnév vagy intézmény nevének automatikus cseréjére a teljes dokumentumban; ismételt szóközpök cseréje egy szóközpökre az internetről mentett szövegek szerkesztésekor; több bekezdés egyesítése a **Bekezdés vége ¶** jel lecserélésével **Szóközpökre**.

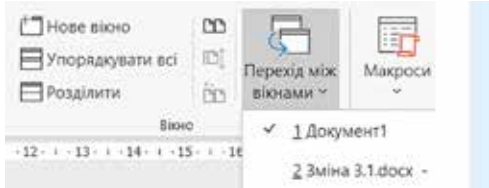


### Ha többet szeretnél tudni

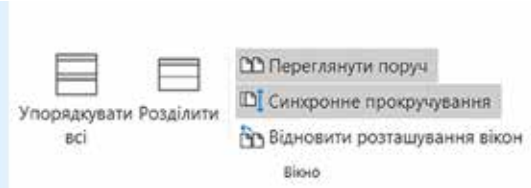
#### A dokumentumok ablakaival végzett munka jellemzői

Több megnyitott ablak kezelését a **Word** szövegszerkesztőben a **Nézet** menü **Ablak** fülben lehetséges (3.6. ábra). Az ablakok között a következő algoritmus szerint válthatunk:

1. Válasszuk az **Ablakváltás** lehetőséget  a **Nézet** menü **Ablak** fülben!
2. A megnyitott ablakok között válasszuk a számunkra szükségeset (az aktuális ablakot, amit épp használunk, √ jelzi)!



3.6. ábra. Az **Ablakváltás** menüje



3.7. ábra. Az **Ablak** fül lehetőségei

A dokumentum két szakaszát egyidejűleg megjeleníthetjük az **Ablak** fül **Felosztás** gombjával. Ezzel a módszerrel figyelhetjük az egyik szakaszt és szerkeszthetjük a másikat.

Ahhoz, hogy visszakерüljünk az egy ablakos szerkesztőmódba, kattintsunk az **Ablak** fül **Felosztás megszüntetése** gombra, vagy kattintsunk kétszer a dokumentumokat elválasztó **Méretezőre**.

Amikor két különböző dokumentumot szeretnénk összehasonlítani, akkor az oda-vissza váltás helyett egymás mellé helyezhetjük a dokumentumokat. Ehhez a **Nézet** menü **Ablak** fülének **Párhuzamos nézet** lehetőségét válasszuk. Ezen műveletek elvégzése után a monitor félbe lesz osztva, az egyik felében lesz az egyik dokumentum, a másikban pedig a másik. A dokumentumok görgetése egyidejűleg (szinkronban) fog történni. Ezzel a módszerrel könnyen összehasonlíthatjuk a két dokumentumot soronként, vagy megkereshetjük a különbségeket.

Ahhoz, hogy visszavonjuk a két dokumentum együttes görgetését, kapcsoljuk ki a **Párhuzamos görgetést** az **Ablak** fülben (3.7. ábra)! A párhuzamos nézet megszüntetéséhez ismételten nyomjuk meg a **Párhuzamos nézet** gombot.



### Használjuk a számítógépet!



Végezzétek el a gyakorlati feladatot a <https://cutt.ly/vhde9vY> linkről, vagy a QR-kód segítségével!



### Összefoglalás

Sokféle alkalmazás létezik szövegek létrehozására és feldolgozására, ezeket **szövegszerkesztő rendszereknek** nevezzük.

Ezekkel az alkalmazásokkal szöveges dokumentumokat hozhatunk létre, elmenthetjük azokat, szerkeszthetjük és formázhatjuk az objektumaikat, automatizált helyesírás-ellenőrzést végezhetünk és kinyomtathatjuk azokat. Ezeknek az alkalmazásoknak a használata nagyban megkönnyíti a szöveg szerkesztését és formázást, képletek, ábrák, táblázatok stb. beillesztését.

A **szöveges dokumentum** szöveges elemei: a karakter, szó, mondat, bekezdés, oldal. Ezenkívül a dokumentum tartalmazhat még különböző objektumokat: képeket, ábrákat, táblázatokat, diagrammokat stb. A szöveges dokumentum minden objektumának megvannak a maga tulajdonságai.

A **Word** szövegszerkesztő alapértelmezett fájlformátuma a **DOCX**. Ezenkívül a **Word** más formátumokkal mentett dokumentumokkal is működik, például **RTF, TXT, DOC, PDF, HTML, ODT** és így tovább.

A **Word** szövegszerkesztőben lehetőségünk van olyan szimbólumok elhelyezésére a szövegben, amelyek nincsenek a billentyűzeten. Ezeket a szimbólumokat **Beszúrás, Szimbólum** eszközzel illeszthetjük a szövegbe.

A **Word** szövegszerkesztőben automatikusan megkereshetünk a dokumentumban egy szövegrészt (karaktert, szót, kifejezést) és kicserélhetjük a szövegrészeket egy másikra. Ehhez válasszuk a **Keresés/Csere** gombot a **Kezdőlap** menü **Szerkesztés** fülében, amely megnyitja a keresőmező **Navigációs** paneljét, ahova beírhatjuk a keresendő szövegrészt. Ezután a **Egyebek** gombra kattintva megadhatjuk a keresési paramétereket: keresési irány (lefelé, felfelé, mindenütt), kis- és nagybetűk megkülönböztetése, szórészlet vagy teljes szó, stb.



A **Word** szövegszerkesztő lehetővé teszi a felhasználó számára, hogy egyszerre több megnyitott dokumentummal dolgozzon. Ez lehetővé teszi a felhasználó számára a dokumentumok megtekintését, azok tartalmának összehasonlítását, az egyik dokumentum szövegrészeinek beszúrását a másikba, egyesítését és így tovább. A **Word**-ben megnyitott ablakok kezeléséhez vannak speciális eszközök, amelyek az **Ablak fül Nézet** menüben találhatóak.




### Felelj a kérdésekre!

- 1<sup>o</sup>. Mit nevezünk szövegszerkesztő rendszereknek? Melyek a főbb jellemzőik?
- 2<sup>o</sup>. Milyen formátumban menthető el egy szöveges dokumentumfájl? Melyik dokumentumformátum a szabványos? Mik a szöveges dokumentumok különböző formátumainak a jellemzői?
- 3<sup>o</sup>. Milyen speciális karaktereket lehet beilleszteni egy szöveges dokumentumba? Hogyan kell őket beilleszteni? Mikor válhat hasznodra?
- 4<sup>o</sup>. Milyen esetekben használják a formázási jelek megjelenítésének módját? Hogyan lehet bekapcsolni?
- 5<sup>o</sup>. Hogyan történik a dokumentumokban adott szövegrészek keresése? Mondj példákat ezen lehetőség használatára!
- 6<sup>o</sup>. Hogyan cserélhető ki automatikusan a dokumentum egy szövegrésze egy másikkal? Mondj példákat ezen lehetőség használatára!
- 7<sup>o</sup>. Hogyan jeleníthető meg egyszerre a képernyőn több megnyitott dokumentumablak? Hogyan kell ezt kivitelezni? Mikor célszerű a használatuk?
- 8<sup>o</sup>. Kérdezd meg a szüleidet, hogyan használják a szövegszerkesztő rendszereket a munkájuk során! Majd beszéljétek meg ezt társaitokkal!
- 9<sup>o</sup>. Szerinted mi ösztönözte a szövegszerkesztő technológiák fejlődését, és mit értünk „papír nélküli technológiák” alatt?



**Végezd el a feladatokat!**

- 1°. A szövegszerkesztőben létrehozott elektronikus dokumentum 560 karaktert tartalmaz. Mivel egyenlő bináris kódjának hossza a **KOI-8**-as kódtáblázatban? A számításokat a füzetedben végezd el!
- 2°. A számítógép merevlemezén keress meg egy **TXT** kiterjesztésű dokumentumot és nyisd azt meg a **Word** szövegszerkesztőben! A füzetedbe írd be a végrehajtások sorrendjét!
-  3°. Hozz létre egy új szöveges dokumentumot és írd bele a megadott szöveget! A dokumentumot **feladat 3.1.3.docx** néven mentsd el a saját mappádba!

A háromszög szögeinek az összege  $\alpha + \beta + \gamma = 180^\circ$ . A kör területe  $\frac{1}{4}\pi d^2$ -el egyenlő.

A Halak csillagjegy alatt születtem ☿. Én ☺ 🎵 hallgatni, 📖 olvasni, a 🖨️ dolgozni, 🌱 termesztetni. Viszont nagyon ☺ repülni a ➔.

- 4°. Nyiss meg egy szöveges dokumentumot (például a **3 fejezet\3.1rész\minta.docx** fájlt)! Keresd meg és jelöld ki az összes **a** betűt a szövegben! Cseréld le **@** -ra! MentSD el **feladat 3.1.4.docx** néven a saját mappádba!



- 5°. Nyiss meg egy szöveges dokumentumot (például a **3 fejezet\3.1rész\minta.docx** fájlt)! Cseréld le az összes **Számítógép**, **Számítógépnek**... szavakat PC-re! MentSD el **feladat 3.1.5.docx** néven a saját mappádba! Figyeld meg, hogy az osztálytársaid melyik módszert választják a cserére! Mondj véleményt arról, melyik módszer a helyénvalóbb!



- 6°. A **Súgó** használatával találd meg az összes nagy **M** betűt a tanárotok által megadott szövegben (például a **3Fejezet\3.1rész\minta.docx**-ban)! MentSD el **feladat 3.1.6.docx** néven a saját mappádba!



- 7°. Magyarázd el, hogyan lehet a szöveg összes bekezdését egyszerre egyesíteni az automatikus cserével! Végezd el a feladatot a tanárod által megadott dokumentumban (például a **3Fejezet\3.1rész\minta.docx**-ban)! MentSD el **feladat 3.1.7.docx** néven a saját mappádba!

- 8°. A **Word** szövegszerkesztőben nyiss meg három szöveges fájlt (például a **3Fejezet\3.1rész** mappából a **minta.docx**, **minta1.docx**, **minta2.docx** fájlokat)! Használd az ablakok közötti váltásra vagy a **Tálcát**, vagy az **Ablakváltás** lehetőséget! Hozz létre egy új szöveges dokumentumot és másold bele a három megnyitott dokumentum címét! MentSD el **feladat 3.1.8.docx** néven a saját mappádba!



- 9°. Alkossatok három fős csoportokat az osztályban! Keressetek adatokat három olyan íróeszközzel, amelyet az emberek különböző korszakokban használtak! Készítsetek 5 diából álló prezentációt, majd mutassátok be!



## 3.2. Többszintű listák, szakaszok, hasábok, fej- és láblécek beillesztése a szövegbe



1. Milyen típusú listákat lehet készíteni a szövegszerkesztőben? Hogyan lehet létrehozni a listákat?
2. Milyenek lehetnek a szöveges dokumentumok megjelenítési módjai? Mire szolgálnak?
3. Mire szolgálnak a vízszintes vonalozón a behúzóok?



### Többszintű listák létrehozása

A **listák** lényegében speciálisan formázott bekezdések. Listákkal általában objektumok vagy teendők felsorolását formázzuk meg. A **Word**ben háromféle listát alkalmazhatunk: *felsorolás*, *számozott*, *többszintű*.




A felsorolás és számozott listával már az 5. osztályban megismerkedtünk, ezért részletesebben inkább a többszintű listával fogunk megismerkedni, melynek egy példáját a 3.8. ábra tartalmazza.

| Évszakok:   | Évszakok:  | Évszakok:  |
|---|--|--|
| 1. Tél <ul style="list-style-type: none"> <li>• December</li> <li>• Január</li> <li>• Február</li> </ul>  | 1. Tél <ul style="list-style-type: none"> <li>a) December</li> <li>b) Január</li> <li>c) Február</li> </ul>  | 1. Tél <ul style="list-style-type: none"> <li>1.1. December</li> <li>1.2. Január</li> <li>1.3. Február</li> </ul>  |
| 2. Tavasz <ul style="list-style-type: none"> <li>• Március</li> <li>• Április</li> <li>• Május</li> </ul> | 2. Tavasz <ul style="list-style-type: none"> <li>a) Március</li> <li>b) Április</li> <li>c) Május</li> </ul> | 2. Tavasz <ul style="list-style-type: none"> <li>2.1. Március</li> <li>2.2. Április</li> <li>2.3. Május</li> </ul> |
| 3. Nyár   | 3. Nyár  | 3. Nyár  |
| 4. Ősz  | 4. Ősz   | 4. Ősz   |

3.8. ábra. Többszintű listák példája

Többszintű a lista, amikor a listák egyes elemei alá új listák vannak rendelve. A szövegszerkesztőben maximum 9 szintű lista hozható létre.

Többszintű lista elkészítéséhez:

1. A kurzort a leendő lista elejére állítjuk.
2. Megnyitjuk a **Kezdőlap**, **Bekezdés** fül **Többszintű lista**  lehetőséget (3.9. ábra).
3. Válasszuk ki a megfelelő listanézetet a **Listatár**-ból! Kiválasztása után az új lista első száma (jelölője) megjelenik a szövegben.
4. A lista új elemei begépelése után válasszuk a **Kezdőlap**, **Bekezdés** fül **Behúzás csökkentése**  (a bekezdés távolságának csökkentése) vagy **Behúzás növelése**  (bekezdés távolságának növelése) lehetőséget és ezzel alakítjuk a lista szintjeit! Ugyanezt megtehetjük a **Tab** vagy a **Shift+Tab** billentyűkombinációval is.

A bekezdés csökkentését vagy növelését a margótól bármikor megtehetjük, a létrehozásukkor is és a szerkesztésük során is. A többszintű listák szerkesztése és formázása hasonló az egyszintű listákhoz.



3.9. ábra. A Többszintű lista megnyitott ablaka

## Szakaszok létrehozása egy szöveges dokumentumban

A szöveges dokumentum általában egy szakaszból áll, minden oldal beállítása (oldalméret, margók, orientáció) ugyanolyan. Hosszú dokumentumok esetében azonban előfordulhat, hogy egyes oldalak paramétereit eltérőek legyenek. Például egy olyan dokumentumba, amelyben az oldalak tájolása álló, el kell helyezni egy olyan sok oszlopból álló táblázatot, amelyhez a fekvő tájolás a megfelelő. Ezt a problémát szakasztörések elhelyezésével orvosolhatjuk.

A szöveges dokumentum szakaszokra bontásához be kell illeszteni egy **szakasztörést** abban a pontban, ahol egy másik szakasz kezdődik. Ha több szakasznak kell lennie a dokumentumban, akkor mindegyik szakaszhoz egy új szakasztörés kell. Szakasmegszakítások hiányában a szöveges dokumentum egy szakaszt alkot.

Többféle szakasztörés létezik:

- *következő oldal* – az új szakasz a következő oldalon kezdődik;
- *folymatos* – új szakasz kezdése az oldalon belül;
- *páros oldal* – új szakasz kezdése a következő páros oldalon;
- *páratlan oldal* – új szakasz kezdése a következő páratlan oldalon.

Szakasztörések létrehozását ajánlatos a **Nyomatási elrendezés** és a bekapcsolt **Minden látszik** módba elvégezni. Az **Elrendezés** ⇒ **Oldalbeállítás** ⇒ **Töréspontok**

 parancs végrehajtása után a **Szakasztörések** listából válasszuk ki a nekünk megfelelőt! A töréspont helyén megjelenik egy láthatatlan szakasztörés szimbólum, például

EZ ——— Розрив розділу (з наступної сторінки) ———.

Ha el kell távolítani a dokumentumból a szakasztörést, törölnünk kell a **szakasztörés** rejtett szimbólumot a szövegből. Ekkor a szakasztörést követő szakasz egyesül az előzővel, emiatt formátuma is automatikusan megváltozik.

## Hasábok szerkesztése

A **Word** szövegszerkesztő lehetővé teszi egy dokumentum szövegének több oszlopba (hasábra) helyezését, mint ahogy az újságokban és folyóiratokban történik (3.10. ábra).

Віртуальна реальність — високорозвинута форма комп'ютерного моделювання, яка дозволяє користувачеві зануритись у штучний світ і безпосередньо діяти в ньому за допомогою спеціальних сенсорних пристроїв, які пов'язують його рухи з аудіовізуальними ефектами. При цьому зорові, слухові, дотикові і моторні відчуття користувача замінюються їх імітацією, що її генерує комп'ютер. Характерними

ознаками віртуальної реальності є: моделювання в реальному масштабі часу; імітація оточення з високим ступенем реалізму; можливість діяти на оточення і мати при цьому зворотний зв'язок.

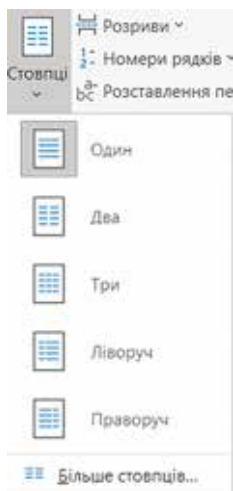
Штучний простір, створений комп'ютерами, який має всі ознаки реальності як такої, що піддається проникненню і трансформації ззовні. При цьому у віртуальній реальності можливі комунікації не лише з іншими людьми, але і з

віртуальними, штучними персонажами.

Термін «віртуальність» уперше з'явився в XVII сторіччі, в розробках класичної механіки, як позначення математичного експерименту, обмеженого об'єктивною реальністю, зокрема, накладеними зовнішніми обмеженнями і зв'язками. А власне формулювання «віртуальної реальності» трапляється вже в XIX сторіччі.

Уривок з Вікіпедії

### 3.10. ábra. Három hasábra tördelt szöveg

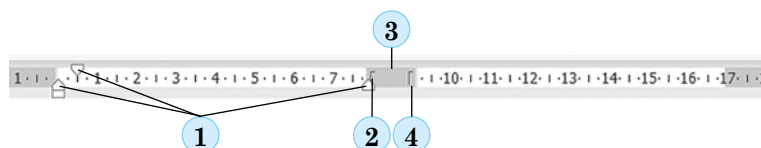


3.11. *ábra.*  
A **Hasábok**  
legördülő menü

A szöveget az **Elrendezés** ⇒ **Oldalbeállítás** ⇒ **Hasábok** parancs végrehajtása után tördelhetjük több hasábra. A legördülő menüből (3.11. ábra) kell kiválasztani a hasábok számát és elrendezését. A szöveg automatikusan a kiválasztott hasábokba lesz tördelve.

Ha vissza szeretnénk térni az egyhasábos tördeléshez, jelöljük ki a szöveget, majd a legördülő menüben válasszuk ki az egyhasábos tördelést. A művelet után az első sor előtt és az utolsó sor után megmarad a **szakasztörés rejtett** formázókarakter, amit ajánlatos eltávolítani.

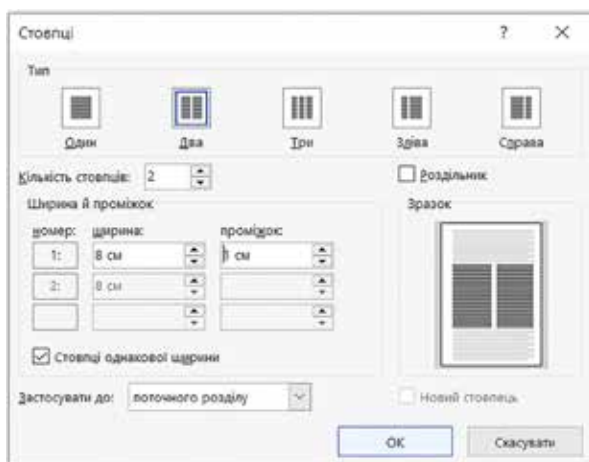
Ha a kurzor egy több hasábra tördelt szövegben van, akkor a vízszintes vonalzon a bekezdésbehúzásokon és tabulátorjeleken kívül **hasábelválasztó** jeleket is találunk (3.12. ábra). Ezek mozgásával megváltoztathatjuk a hasábszélességeket és a hasábok távolságát.



1. Bekezdésbehúzások
2. Első hasáb jobb széle
3. Hasábelválasztó
4. Második hasáb bal széle

3.12. *ábra.* Markerek a vízszintes vonalzon

A hasábok formázása a **Hasábok** párbeszédablakban (3.13. ábra) is elvégezhető, amit az **Elrendezés** ⇒ **Oldalbeállítás** ⇒ **Hasábok** ⇒ **További hasábok** paranccsal nyithatunk meg. Itt is beállíthatjuk a hasábok számát és a távolságot, elválasztóvonalat (függőleges vonal) helyezhetünk el a hasábok között. Ha elkészültünk a beállításokkal, nyomjuk meg az **OK** gombot!



3.13. *ábra.* **Hasábok** párbeszédablak

## Fejlécek létrehozása szöveges dokumentumban


Hosszú dokumentumok esetében gyakran alkalmaznak fejléceket (a francia **colon** – oszlop és a latin **titul** – cím összevonásából), ezek a dokumentum oldalainak margóin elhelyezkedő „szolgálati” információkat tartalmazó mezők.

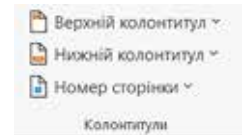
A **Wordben felső (fejléc), alsó (lábléc)** és **oldalsó** fejléceket használhatunk. Ezek a fejlécek leginkább a dokumentum és a szerző nevét, az oldalszámot, a fejezet címét, grafikai elemeket tartalmaznak.

Az első oldal fejléce különbözhet a többitől, ugyanígy különbözhet a páros és páratlan oldalak fejlécei, vagy az egyes fejezetek fejlécei.

A **Wordben** beilleszthetünk sablon alapján készült fejléceket, de saját fejléceket is létrehozhatunk, amit aztán el is menthetünk a sablonok közé. A fejléceket csak **Nyomtatási elrendezés** és **Olvasás** nézetben láthatjuk. A fejléc szerkesztése közben a szöveg szerkesztése nem lehetséges.

Ha kész sablonból szeretnénk elhelyezni fejléceket, a következőképpen járunk el:


1. Nyissuk meg a **Beszúrás** lapot!
2. Válasszuk ki az **Élőfej és élőláb** csoportból az **Élőfej** vagy az **Élőláb** elemet (3.14. ábra)!
3. Válasszunk a felajánlott sablonok közül!
4. Írjuk be a sablonba a megfelelő szöveget! A **Tervezés** ideiglenes eszköztár segítségével végezzük el a finomításokat (3.15. ábra)!
5. Zárjuk be a fejléceket az **Élőfej és élőláb eszközök** ⇒ **Tervezés** ⇒ **Bezárás** ⇒ **Élőfej és élőláb bezárás**  gombbal!




3.14. ábra. Az **Élőfej és élőláb** csoport



3.15. ábra. Az élőfej sablonja és a **Tervezés** ideiglenes eszköztár

A szaggatott vonallal határolt mezőbe írjuk a fej- vagy lábléc tartalmát. A fejléc beillesztése után a **Szalagon** elérhetővé válik a **Tervezés** ideiglenes eszköztár. Segítségével különböző elemeket – **Oldalszám** , **Dátum és idő**, **Képek** stb. – helyezzünk el a fejlécen. A fejléc helyét az oldalon a **Pozíció** lap elemeinek segítségével szabályozhatjuk.

Az élőfej és élőláb között a **Navigáció** lap **Váltás az élőfej és élőláb között**  eszközzel válthatunk. Ezt az eszközt is a **Tervezés** eszköztárban találjuk.

A létrehozott fejléceket alkalmazhatjuk a teljes dokumentumra, a páros vagy a páratlan oldalakra vagy csak az első oldalra. A fejléc alkalmazási területét valamint egyes tulajdonságainak beállításait a **Tervezés** lap **Beállítások** csoportjában található eszközökkel végezzük.

Amennyiben a dokumentumban különböző fejléceket szeretnénk alkalmazni, osszuk a dokumentumot szakaszokra, ekkor szakaszonként különböző fejléceket hozhatunk létre!

A fejléceket természetesen szerkeszthetjük és formázhatjuk is. Ezt a **Beszűrés** ⇒ **Élőfej és élőláb** ⇒ **Élőfej (élőláb)** ⇒ **Élőfej (élőláb) szerkesztése** eszközzel kezdeményezzük! A **Beszűrés** ⇒ **Élőfej és élőláb** ⇒ **Élőfej (élőláb)** ⇒ **Élőfej (élőláb) eltávolítása** eszközzel törölhetjük a fejléceket.



### Ha többet szeretnél tudni

#### A tabulátor használata

Már tudjátok, hogy a **Tab** gomb lenyomásával a kurzor egy megadott távolságra ugrik. Ez az alapértelmezett érték 1,25 cm. A **Tab** gomb lenyomásakor ® rejtett formázási jel keletkezik.

A **Tab** billentyűt használva a szövegrészeket különböző távolságokra helyezhetjük el a sorokban, a tabulátort úgy helyezzük el a vonalzón, hogy a megfelelő helyre kattintunk. A **Tab** billentyű lenyomása következtében a szövegrészek között keletkezett üres teret kitölthetjük szimbólumokkal (pont, vonás, egyenes stb.). Például:

|          |         |           |            |
|----------|---------|-----------|------------|
| 1) Tél   | Tavaszi | Nyár      | Ősz        |
| December | Március | Június    | Szeptember |
| Január   | Április | Július    | Október    |
| Február  | Május   | Augusztus | November   |

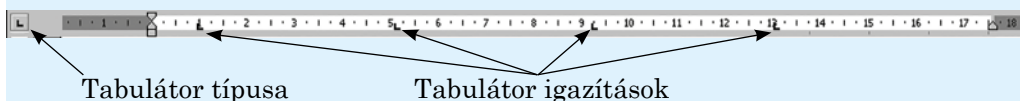
2) A **KFT** igazgatója \_\_\_\_\_ I. V. Szidorenko

- 3) 1. fejezet ..... 1. old.  
 2. fejezet ..... 8. old.  
 3. fejezet ..... 15. old.

A **Word** szövegszerkesztőben többféle tabulátor közül választhatunk, amelyet a vízszintes vonalzótól balra található **Tabulátor** jel mutat. Ha rákattintunk, akkor megváltozik a tabulátor típusa:

|  |                    |  |                     |
|--|--------------------|--|---------------------|
|  | – balra igazítás;  |  | – középre igazítás; |
|  | – jobbra igazítás; |  | – decimális.        |
|  | – tabulátor vonal; |  |                     |

Ezután a vonalzón megjelöljük a tabulátor pozícióit, a megfelelő tabulátortípust választva. Ennek következtében a vonalzón megjelennek a tabulátorpozíciók (3.16. ábra).



3.16. ábra. Vízszintes vonalzó a tabulátor jelekkel

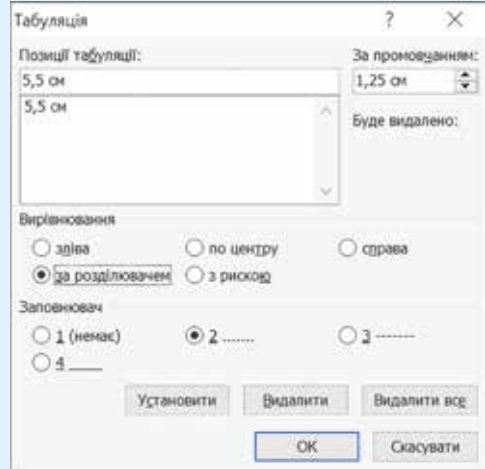


Ahhoz, hogy módosítani tudjuk a tabulátor értékeit, meg kell nyitnunk a **Tabulátorok** párbeszédablakot, kétszer kattintva a vonalzón lévő tabulátorjelre (3.17. ábra). A megjelenő ablakban a következő értékeket állíthatjuk be:

- a bal margótól számított távolságot cm-ben;
- igazítást a tabulátor helyzetéhez képest;
- kitöltést, ami a tabulátor használatakor keletkezett üres teret tölti ki.

A tabulátor új értékeinek megadása után kattintsunk a **Felvétel** gombra és a tabulátor értékei módosulnak. A tabulátorok helyzetét és tulajdonságait a **Tabulátor** párbeszédablakban szabályozhatjuk, amit a **Kezdőlap** menü **Bekezdés** fül **Tabulátorok** eszközzel is megnyithatunk.

Ha el akarjuk távolítani a tabulátort, húzzuk le a vonalzóról!



3.17. ábra. Tabulátor párbeszédablak



### Használjuk a számítógépet!






Végezzétek el a gyakorlati feladatot a <https://cutt.ly/uhde8j2> linkről, vagy a QR-kód segítségével!



### Összefoglalás

A bekezdéseket felsorolássá vagy sorszámozott listává alakíthatjuk, így tehetjük jobban áttekinthetővé a szövegeket. Ezt úgy tehetjük meg, hogy kijelöljük a megfelelő szövegrészt, majd a **Kezdőlap** ⇒ **Bekezdés** ⇒ **Többszintű lista** gombra kattintunk. Ha szükséges, a gombok legördülő menüjéből kiválaszthatjuk a megfelelőt.

A lista új elemei begépelése után, válasszuk a **Kezdőlap Bekezdés** fül **Behúzás csökkentése**  (a bekezdés távolságának csökkentése) vagy **Behúzás növelése**  (bekezdés távolságának növelése) lehetőséget és ezzel alakítsuk a lista szintjeit!

Ahhoz, hogy egy szövegrészt szakaszokra bontsunk, szakasztöréseket kell elhelyezni rajtuk. Szakasztörések létrehozását ajánlatos a **Nyomatási elrendezés** és a bekapcsolt **Minden látszik** módban elvégezni. Az **Elrendezés** ⇒ **Oldalbeállítás** ⇒ **Töréspontok**  parancs végrehajtása után a **Szakasztörések** listából válasszuk ki a nekünk megfelelőt. A töréspont helyén megjelenik egy láthatatlan szakasztörés szimbólum.

A szöveget az **Elrendezés** ⇒ **Oldalbeállítás** ⇒ **Hasábok** parancs végrehajtása után tördelhetjük több hasábra. A legördülő menüből kell kiválasztani a hasábok számát és elrendezését. A szöveg automatikusan a kiválasztott hasábokba lesz tördelve.

Hosszú dokumentumok esetében gyakran alkalmaznak fejléceket, ezek a dokumentum oldalainak margóin elhelyezkedő „szolgálati” információkat tartalmazó mezők.

A **Word**ben *felső (fejléc)*, *alsó (lábléc)* és *oldalsó* fejléceket használhatunk. Ezek a fejlécek leginkább a dokumentum és a szerző nevét, az oldalszámot, a fejezet címét, grafikai elemeket tartalmaznak.

Az első oldal fejléce különbözhet a többitől, ugyanígy különbözhet a páros és páratlan oldalak fejléce, vagy az egyes fejezetek fejlécei.

A fejléceket csak **Nyomatási elrendezés** és **Olvasás** nézetben láthatjuk. A fejléc szerkesztése közben a szöveg szerkesztése nem lehetséges. Fejléceket a **Beszúrás** lap **Élőfej- és élőláb** csoportjainak eszközeivel helyezhetünk el a dokumentumban.



### Felelj a kérdésekre!

- 1°. Mit nevezünk többszintű listának? Hogyan hozható létre?
- 2°. Hogyan változtatjuk meg a felsorolási jeleket?
- 3°. Hogy tördelhetjük a szövegeket hasábokba? A hasábok milyen tulajdonságait változtathatjuk meg?
- 4°. Mik a szakaszok? Mire szolgálnak?
- 5°. Hogyan állíthatjuk be a hasábok szélességét, a köztük lévő behúzásokat, számukat?
- 6°. Mik a fejlécek? Milyenek lehetnek?
- 7°. Mire szolgálnak a fejlécek? Milyen objektumokat helyezhetünk el ezekben?
- 8\*. Hogy állíthatunk be a dokumentum oldalaihoz különböző fejléceket?
- 9°. Mi az a tabulátor? Milyen körülmények között használják?
- 10°. Mit gondolsz, a 3.2. fejezetben bemutatott szövegek objektumai hogyan járulnak hozzá a szöveg jobb megjelenítéséhez és felépítéséhez?



### Végezd el a feladatokat!

- 1°. Hozz létre egy szöveges dokumentumot a minta alapján! A listának a következő értékeit állítsd be: a fő pontok bekezdésének távolsága a margótól 2 cm, az alpontok távolsága pedig 3 cm legyen! Mentse el a dokumentumot **feladat 3.2.1.docx** néven a saját mappádba! Hasonlítsd össze a kapott munkát a mintával, majd értékeld a látottakat!



- 2\*. Hozz létre egy új dokumentumot és készíts többszintű listát az alábbi szöveg felhasználásával! Mentse el a dokumentumot **feladat 3.2.2.docx** néven a saját mappádba!

#### 1. Reggeli

- 1.1 Tea/kávé
- 1.2 Vajás kenyér

#### 2. Ebéd

- 2.1 Borscs leves
- 2.2 Sült krumpli
- 2.3 Káposztasaláta
- 2.4 Üdítő

#### 3. Vacsora

- 3.1 Aludttej
- 3.2 Zsömle

A színeket a következő csoportokba sorolhatjuk: **meleg** (piros, narancssárga, sárga) és **hideg** (világoskék, sötétkék, lila), valamint megkülönböztetünk **könnyű** és **nehéz színeket**.

*Könnyű színek* – átlátszó, hideg, világos, az éghez és a térhez hasonló.

*Nehéz színek* – sötét, telített, sűrű; ide tartozik a fekete, a sötétszürke, lila és sötétvörös.

- 3°. Nyisd meg a tanárod által megadott fájlt (például a **3fejezet\3.2részlminta.docx**-et)! A szöveges dokumentum minden pontját formázd különálló szakaszként, ami új lapon kezdődik! A szakaszokat formázd az alábbiak szerint:

- páratlan szakaszok: *álló* tájolás, 2 cm margó, oldalszám *felül* középen;
- páros szakaszok: *fekvő* tájolás, margók 1 cm, oldalszám *balra lent!*

Mentsd a dokumentumot **feladat 3.2.3.docx** néven a mappádba! Melyik feladat volt nehezebb számodra? Hogyan oldottad meg a nehézségeket?



4\*. Nyisd meg a tanárod által meghatározott fájlt (például a **3.fejezet\ 3.2.rész\ minta 3.2.4.docx**-et)! A második bekezdést tördeld két egyforma szélességű oszlopba, melyek között nincs elválasztó, a harmadikat pedig három különböző szélességű, vonallal elválasztott hasábbal! Mentsd el a fájlt **feladat 3.2.4.docx** néven a mappádba!

5\*. Nyisd meg a tanárod által megadott fájlt (például a **3.fejezet\ 3.2.rész\ minta 3.2.5.docx**-et)! Hozz létre élőfejet, illeszd bele a mai dátumot *középre igazítva*, valamint élőlábat, ebbe illeszd bele a *nevedet balra igazítva!* Mentsd a dokumentumot **feladat 3.2.5.docx** néven a mappádba!



6\*. Nyisd meg a tanárod által megadott fájlt (például a **3.fejezet\ 3.2.rész\ minta 3.2.5.docx**-et)! Végezd el a következő formázásokat: az élőfej a páratlan oldalon a dokumentum nevét tartalmazza, élőláb a páros oldalakon a *dátumot és időt!* Mentsd a dokumentumot **feladat 3.2.5.docx** néven a mappádba!

### 3. SZÁMÚ GYAKORLATI MUNKA

#### Különböző objektumokat tartalmazó szöveges dokumentum létrehozása

**Figyelem!** A számítógép használata során tartsd be a balesetvédelmi szabályokat és az egészségügyi előírásokat!

1. Nyissátok meg A *balesetvédelmi szabályokat és az egészségügyi előírásokat tartalmazó minta 3gym.docx* szöveges dokumentumot, ami a **3.fejezet\3gyakorlati** mappában található!
2. Készítsetek többszintű listát a következő beállítások szerint: a kék színű szövegrész – első szintű címek, a lila szövegrész – második szintű címek! A számozást a következőképpen állítsátok be <sup>1.</sup>\_\_\_\_\_!  
1)\_\_\_\_\_!
3. A szöveg minden **V** betűjét cseréljétek le **B** betűre!
4. A szövegben szereplő összes vesszőt cseréld le a **VESSZŐ** szóra, az összes pontot pedig a **PONT** szóra!
5. A zöld színnel jelölt szövegrészt tördeld két egyforma szélességű 7 cm széles oszlopba!
6. Hozz létre élőfejet a dokumentumba, illeszd bele a *mai dátumot*, valamint élőlábat, ebbe illeszd bele a *nevedet!*
7. Mentsd a dokumentumot **3sz gyakorlati munka.docx** néven a mappádba!

### 3.3. STÍLUSOK ALKALMAZÁSA A DOKUMENTUMBAN. A DOKUMENTUM SZERKEZETÉNEK KIALAKÍTÁSA



1. Milyen tulajdonságait ismerjük a szöveg oldalainak? Hogy állíthatjuk ezeket be?
2. Milyen dokumentumnézetek léteznek **Word**ben? Hogyan váltunk ezek között?
3. Hogyan van strukturálva az informatika tankönyv? Milyen objektumokat találhatunk az oldalain?

## Stílusok a szöveges dokumentumokban

Miután létrehoztunk egy szöveges dokumentumot, megfelelően meg kell formázunk, be kell állítani a kívánt tulajdonságértékeket a szöveghez, táblázatokhoz, ábrákhoz stb. Ha a dokumentum hosszú és sok különböző objektumot tartalmaz, az objektumok formázása sok időt igényel. Ha ráadásul sok, ugyanúgy formázott objektumot tartalmaz, külön erőfeszítés a formázások megtartása.

A dokumentumok formázásának megkönnyítése és meggyorsítása céljából célszerű *stílusokat* alkalmazni. A **Stílus** bizonyos objektum tulajdonságainak megadott értékeit tartalmazza. Például a behúzások beállításával megadhatjuk a bekezdések sorainak bal és jobb oldali margótól való távolságát, a sorok közötti távolságot stb.; a táblázat stílusánál pedig beállítható a szegélyvonalak színe és stílusa, a táblázat cellájának kitöltési színe stb.

A **Word**ben alapértelmezetten elérhető a *stílustár*, amelyben az objektumok tulajdonságai harmonikusan vannak megválasztva, a mai dizájn kívánalmainak megfelelően. Ilyen stílusok a szöveges dokumentum különböző objektumai – szöveg, táblázat, grafikai objektumok – számára léteznek (3.5. táblázat).

3.5. táblázat

### Word stílusok

| Objektum | Stílus neve                      | Minta | Tulajdonságok   |
|----------|----------------------------------|-------|---|
| Szöveg   | Címsor4                          |       | Betűtípus: <i>Cambria</i><br>Méret: 12<br>Szín: <i>fekete</i><br>Stílus: <i>félkövér, dőlt</i><br>Sorköz: <i>1,5 sor</i><br>Első sor behúzása: <i>nincs</i><br>Igazítás: <i>sorkizárt</i><br>Térköz előtte: <i>3 pt</i> |
| Táblázat | Világos árnyékolás, 3 jelölőszín |       | Vonal színe: <i>zöld</i><br>Kitöltés: <i>első sor zöld</i><br>Vonalstílus: <i> folytonos</i><br>Vastagság: <i>0,5 pt</i>  |
| Alakzat  | Enyhe effektus, jelölőszín 6     |       | Szín: <i>világosbarna</i><br>Átlátszóság: 0<br>Vonal: <i>folytonos</i><br>Vastagság: <i>1 pt</i><br>Vonalszín: <i>sötétbarna</i><br>Árnyék: <i> jobbra lent</i>   |

Stílusok alkalmazásával egyetlen művelettel több objektum tulajdonságát változtathatjuk meg. Ha például egy **Címsor4** stílusú bekezdést (3.5. tábl.) kézzel formázunk meg, az nyolc művelet elvégzését jelenti, míg a **Címsor4** stílus alkalmazásával egyetlen művelettel érhetjük el ugyanazt: a stílus kiválasztásával.

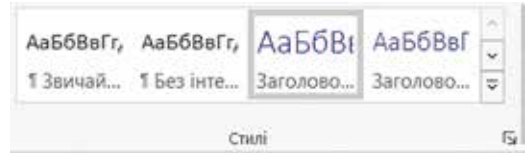
A stílusok gyűjteményét saját stílusokkal egészíthetjük ki, amelyeket a már létező stílusok alapján alakíthatunk ki. A dokumentum mentésekor automatikusan mentésre kerülnek az alkalmazott stílusok, hogy az újbóli megnyitáskor a dokumentum külalakja ne változzon.

### A stílustár stílusainak alkalmazása

A dokumentum stíluskészletét a **Kezdőlap, Stílusok** csoportjában (3.18. ábra) találjuk.

A stíluskészlet minden eleme úgy került kialakításra, hogy annak külalakja megfeleljen a választott stílusnak, a nevek melletti jelölés pedig azt mutatja, milyen objektumra érvényesíthető az adott stílus: bekezdésekre – ¶, vagy mindkettőre – ¶a. A címsorstílusokat a dokumentum strukturális egységeinek (fejezetek, pontok, alpontok) formázásakor használjuk.

A egyéb objektumok (táblázatok, képek, ábrák, grafikonok stb.) stílusainak kiválasztására és szerkesztésére használjuk az ideiglenes **Tervezés, Elrendezés, Formátum** menüket (3.19. ábra).



3.18. ábra. A dokumentum stíluskészlete



3.19. ábra. Táblázatstílusok és képstílusok

A stílustár tetején mindig a legutóbb alkalmazott stílusokat találjuk. A és a gombokkal lapozhatjuk a további stílusok listáját, az **Egyebek** gombra kattintva pedig megtekinthetjük a teljes stílustárát.

A stílus alkalmazásához ki kell választani az objektumot, valamint a kívánt stílusválasztatot is a listából. Felhívjuk figyelmeteket arra, hogy amikor egy stílusválasztat fölé viszíték az egeret, a kiválasztott formátum dinamikus előnézete láthatóvá válik, ha alkalmazni szeretnétek, akkor rá kell kattintanotok.

A stílus helyi menüjének segítségével a stílusokat átnevezhetjük és törölhetjük. Ezeket a műveleteket nem végezhetjük el a címsorstílusokon.

Ha el kívánunk tekinteni a stílusok alkalmazásától és vissza szeretnénk térni az alapértelmezett beállításokhoz, kattintsunk a **Formázás törlése** gombra.

### Témák alkalmazása a dokumentumokban

Stílusokkal nemcsak egyes objektumokat formázhatunk meg, de a teljes dokumentumot is.

A **Téma** a dokumentum egészére alkalmazott stílus. A témáknak egyedi neve van, és stíluselőírásokat tartalmaznak a dokumentumban alkalmazott objektumokra (szimbólumok színe, mérete, betűtípusa, bekezdések igazítása, sorköz, ábrák formázása stb.).



3.20. ábra. Beépített témák

A **Word** beépített témáit az **Elrendezés** menü **Témák** csoportjában található **Témák** legördülő menüs gombbal érhetjük el (3.20. ábra).

Témát választhatunk a dokumentum létrehozásakor, de a szerkesztése során is. A témák áttekintése során a dokumentum dinamikusan formázódik az aktuális témának megfelelően. A téma kiválasztása értelemszerűen kihat a szövegekre, táblázatokra, grafikai elemekre alkalmazott stílusokra is. Minden elem az egységes stílusnak megfelelően lesz formázva. *Fontos megjegyezni*, hogy a téma megváltoztatása csak a stílusokkal megformázott elemek külalakjára van kihatással.

### A dokumentumok stilisztikai tervezésének szabályai

A különféle típusú hivatalos dokumentumok (megrendelések, tanúsítványok, jelentések, levelek stb.) elkészítésekor be kell tartani a tervezésükre, felépítésükre és tartalmukra vonatkozó egységes követelményeket és szabályokat, amelyeket állami szabvány szerint készítettek. Ezeknek a szabványoknak a betartása biztosítja:

- a dokumentumok gyors és magas színvonalú elkészítését;
- a dokumentum tartalmának megértését annak, aki dolgozni fog vele;
- a dokumentumok gyors keresésének megszervezését;
- üzleti papírok számítógépes feldolgozását.

Ukrajnában a dokumentumok stilisztikai tervezésének több alapvető normáit alkalmazják, amelyek folyamatosan frissülnek. Például a ДСТУ 4163-2003 *A szervezeti és adminisztratív dokumentációk egységes rendszere. A dokumentumok stilisztikai tervezésének követelményei* (<https://cutt.ly/shevOsc>) oldalon. E szabvány szerint:

- a hivatalos dokumentumok lapmérete A4-es (210 cm × 297 cm) és A5-ös (210 cm × 148 cm);
- bizonyos típusú dokumentumok lapmérete A3-as (297 cm × 420 cm) és A6-os (105 cm × 148 cm);
- a margótól való távolság: balról 30 mm, jobbról 10 mm, alulról és felülről 20 mm;
- a fő szöveg betűtípusa – *Times New Roman*, betűméret – 12-14 mm, sorköz – 1-1,5, bekezdésméret 1,25 cm;
- a logó képe a lap bal oldalán, a szervezet nevével azonos szinten helyezkedik el.

Ezenkívül a szabvány meghatározza a listák, táblázatok, dátumok, számozás és egyéb objektumok elkészítésének és formázásának szabályait.

A dokumentumok elkészítésére vonatkozó állami követelmények mellett vannak bizonyos felsőfokú intézmények által elfogadott előírások, szakdolgozatok és tudományos előadások elkészítésére vonatkozó követelmények, az iskolai tankönyvek, illetve minden más írott nyomtatványra vonatkozó formai és tartalmi követelményrendszer kidolgozásra került. Ezeket a szabályokat osztályügyi rendeletekben teszik közzé, és azokat előzetesen a dokumentumok készítőinek tudomására hozzák. Ezek kissé eltérhetnek a hagyományosaktól és figyelembe veszik a kiadvány nyomtatási jellemzőit is. Az alábbiakban az Iskolai Kisakadémia (MAH) elkészítésére vonatkozó szabályokat olvashatjuk (<https://cutt.ly/FkvLwBG>):



- a dokumentum mérete legfeljebb 30 nyomtatott oldal, lapméret A4-es, a szövegszerkesztő fájlformátuma *Microsoft Word*, egy oldalon legfeljebb 40±2 sor lehet;
- betűtípus – *Times New Roman*, betűméret 14-es, 1,5 sorköz, sorkizárt;
- bekezdés – *általános*;
- margó beállítások: felső – 20 mm; alsó és baloldali – nem kevesebb mint 20 mm, jobboldali – 10 mm;
- a címsorokat kettős sorköz választja el a felső és alsó szövegrésztől; a fejezetek címei nagybetűvel írottak az oldal közepére igazítva; az általános szöveg címsorait bekezdésből, kis betűkkel írjuk (az első kivételével). A címsorok végére nem teszünk pontot;
- oldalszámozás a jobb felső sarokba kerüljön, az első oldalra ne kerüljön számozás.

Egyes szervezetek kialakíthatják saját vállalati stílusukat dokumentumaik megtervezésére, mint például a betűtípus, méret, színek, bekezdések, a logó elhelyezésének és megformázására vonatkozó szabályok. A dokumentum sablonokat ezeknek a követelményeknek megfelelően fejlesztjük ki. Például egy cég vállalati stílusát a 3.21. ábra mutatja be.

Ha a dokumentumot magáncélra készítik, akkor is kívánatos, egy egységes stílus követése. Ezért be kell tartani a dokumentumok stilisztikai tervezésének szabályait:

- a dokumentum fő szövegét kívánatos egy stílusban megtervezni, kiemelni a címeket és egyes szemantikai töredékeket;
- a dokumentumban a különböző színek és betűtípusok száma nem haladhatja meg a hármat;
- a karakterek méretét, a sor- és bekezdésközt úgy kell megválasztani, hogy a szöveg könnyedén olvasható legyen anélkül, hogy megerőltetné a szemet;
- a színvilágnak meg kell felelnie a dokumentum céljának – az üdvözlőkártya fényes, telített színekkel díszíthető, egy baráti levél pedig nyugodtabb tónusú legyen;
- kívánatos azonos típusú adatokat táblázatokban bemutatni;
- a dokumentum grafikus képeinek (rajzok, diagramok, ábrák) ki kell egészíteniük a szöveg tartalmát, meg kell magyarázniuk annak egyes pontjait.



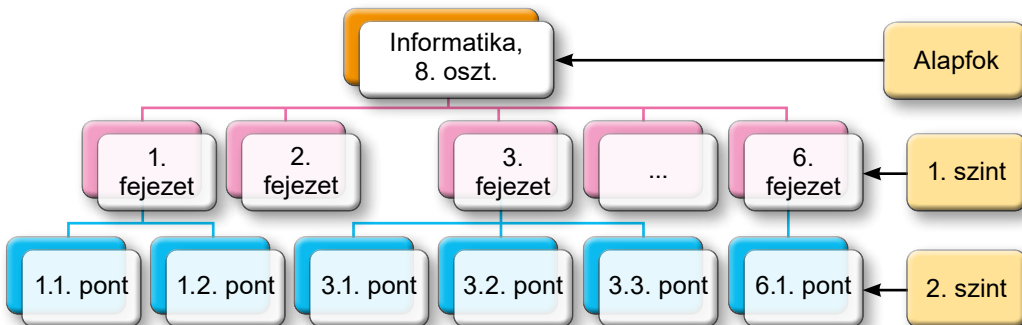
3.21. ábra. Egy cég vállalati stílusa

## A dokumentum szerkezete

A hosszú, sok oldalt tartalmazó dokumentumokat (referátumok, évfolyammunkák, brosúrák, könyvek stb.) átlátható szerkezeti elemekre – részekre, fejezetekre, paragrafusokra, pontokra – kell felosztani, ily módon a dokumentum egy hierarchikus szerkezetet kap. Ilyesféle struktúrákkal már találkozhatok az irodalomórákon, amikor a fogalmazásokhoz vázlatot készítettetek.

A **dokumentum szerkezete** alatt tehát a dokumentum részeiből álló hierarchikus szerkezetet értjük.


Ha megtekintjük ennek a tankönyvnek a felépítését (3.22. ábra), akkor láthatjuk, hogy annak első szintjén a fejezetek, a másodikon a pontok, a harmadikon az alpontok találhatóak.



3.22. ábra. A dokumentum szerkezete

### A dokumentum szerkezetének kialakítása és annak alkalmazása

A dokumentum szerkezetét a címsorstílusok segítségével alakítjuk ki. A legmagasabb szintű strukturális elemhez értelemszerűen a **Címsor 1** stílust rendeljük, a másodikhoz pedig **Címsor 2**-t és így tovább. Ezt a következőképpen végezzük: jelöljük ki a dokumentum egy strukturális elemét (fejezetcím, pont, alpont neve stb.), majd választjuk ki a **Stílusok** csoportból a megfelelő szintű címsort!

Az így kialakított dokumentum szerkezetét **Vázlat** nézetben tekinthetjük meg, amit **Nézet** menü **Nézetek** csoportjának eszközeivel, vagy a **Vázlat**  gombjára kattintva nyitunk meg.

Ez a dokumentumnézet eltér a megszokott nyomtatási nézettől (3.23. ábra).

Програма ЮНЕСКО "Інформація для всіх"

#### 1. Преамбула.

- Інформація і знання є величезним загальнолюдським завоюванням. Вони необхідні для просування освіти, науки, культури і комунікацій, для збагачення культурної різноманітності і відвертості управління. Мандат ЮНЕСКО "сприяти вільному руху ідей словом і справою" ясно вказує на роль, яку організація покликана грати в тому, щоб зробити інформацію і знання доступною для всіх, з кінцевою метою подолання розриву між інформаційно-багатішими і інформаційно-біднішими.
- Для досягнення цієї мети, відповідно до свого конституційного мандата ЮНЕСКО повинна "підтримувати, зберігати, збільшувати, і поширювати знання", гарантуючи і забезпечуючи "збереження і захист" зафіксованих матеріалів і ініціювати "способи міжнародної співпраці, розроблені на те, щоб дати народам всіх країн доступ до опублікованих матеріалів, незалежно від мови і способів їх публікації".
- Міжнародна інтелектуальна співпраця є надзвичайно важливою в наведеному світі, який все більш об'єднується розвитком інформаційних і комунікаційних технологій (ІКТ) і збільшеною конвергенцією їх застосувань. Наслідки цього розвитку, особливо в освіті, науці, культурі і комунікаціях, самими різними способами, прямо або побіжно, впливають на всі країни-члени ЮНЕСКО, групи населення і окремих людей.
- Технологічна конвергенція сама, що посилюється, по собі не сприяє ні глобальному обміну моральними, культурними і суспільними цінностями, ні розвитку рівноправного доступу до інформації для всіх.

#### Програма ЮНЕСКО "Інформація для всіх"

##### 1. Преамбула.

- 1. Інформація і знання є величезним ...
- 2. Для досягнення цієї мети, відповідно до свого ...
- 3. Міжнародна інтелектуальна співпраця є ...
- 4. Технологічна конвергенція сама, що посилюється, ...
- 5. Нові способи доступу, обробки і захисту інформації ...
- 6. У цьому контексті ЮНЕСКО з своїм мандатом - ...

##### 2. Мандат.




- 1. Програма "Інформація для всіх" повинна зіграти ...
- 2. Програма повинна сприяти зменшенню розриву між ...
- 3. В рамках програми повинні робитися прями ...
- 4. Пріоритетним напрямом Програми повинна бути ...

##### 3. Цілі програми.

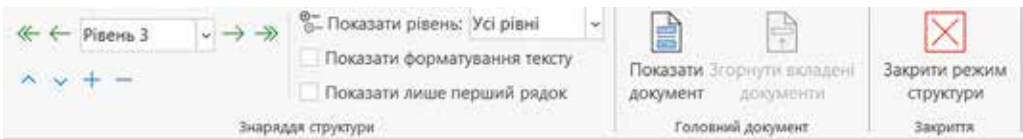
- 1. У дусі Міжнародної Декларації Прав Людини, яка ...
- 2. Як одна з основних програм ЮНЕСКО програма ...
- 3. Зокрема програма "Інформація для всіх" націлена ...
- 4. Заохочення і розширення доступу до інформації за ...
- 5. Розвиток міжнародної рефлексії і дискусій по ...
- 6. Сприяння тренінгу, безперервній освіті і навчання ...

3.23. ábra. Dokumentum Nyomtatási nézetben és Vázlat nézetben


**Vázlat** nézetben a szerkezeti elemek *összecsukva* maradnak, a képernyőn csak a szerkezeti elemeket láthatjuk. Hogy a szerkezet jól áttekinthető legyen, az azonos szintű elemekhez azonos mértékű behúzás tartozik. Az egyes szintek mellett a következő jeleket láthatjuk:

-  – amennyiben a szinthez tartoznak alacsonyabb szintek és szöveg;
-  – amennyiben a szinthez nem tartozik alárendelt szöveg;
-  – amennyiben a szöveg a legalacsonyabb szinthez tartozik (az úgynevezett fő szöveg (törzsszöveg), amelyre nem alkalmazták a *címsor* stílust).

**Vázlat** nézetben a **Menüszalagon** megjelenik a **Vázlat** ideiglenes eszköztár (3.24. ábra). Ennek vezérlőelemeit használhatjuk a dokumentum szintjének kiválasztására, az elem szintjének megváltoztatására, új elemek beillesztésére és régiek törlésére.



3.24. ábra. **Vázlat** eszköztár



A **Vázlat** nézet legnagyobb előnye az, hogy a dokumentum tetszőleges szintjének beállítását úgy végezhetjük, hogy abban az alacsonyabb szintek nem zavarnak, mivel el vannak rejtve. Ezt úgy érjük el, hogy a **Nézet** menü **Vázlat** eszköztárban a **Vázlateszközök** között kiválasztjuk, melyik szintet szeretnénk megtekinteni. Ha a dokumentum egy szerkezeti elemének szeretnénk látni az alárendelt szintjeit, kattintsunk a megfelelő címsor előtti  jelre. A művelet ismétlése újra elrejtí a szöveget.








Ha egy alárendelt szövegnek túl nagy a betűmérete, esetleg túlságosan hosszú, emiatt zavarja a vázlat áttekinthetőségét, beállíthatjuk, hogy annak szövegét formázástól mentesen tekinthessünk meg (kapcsoljuk ki a **Szövegformázás megtekintése** jelölőt), vagy kérhetjük, hogy csak a szöveg első sora jelenjen meg (**Csak az első sor megjelenítése** jelölő).

A **Vázlat** nézetben kényelmes a dokumentum felépítésének szerkesztése a szövegdarabok szintjének és az elhelyezésük sorrendjének megváltoztatásával, a **Vázlateszközök** eszközcsoport vezérlőivel (3.6. táblázat).




3.6. táblázat

**Vázlateszközök eszközcsoport vezérlőelemei a Vázlat nézetben**

| Vezérlőelem  | Rendeltetése   |
|--|--|
| <b>Előléptetés Címsor 1 stílussá</b>  | Az elem előléptetése a vázlat legmagasabb szintjére. A szöveghez automatikusan hozzárendelődik a <i>Címsor 1</i> stílus. |
| <b>Lefokozás szövegtörzszé</b>        | Az elem lefokozása a vázlat legalacsonyabb szintjére. A szöveghez automatikusan hozzárendelődik a <i>Normál</i> stílus.  |

| Vezérlőelem   | Rendeltetése   |
|---|--|
| Előléptetés  | A kijelölt elem előléptetése vagy lefokozása magasabb, illetve alacsonyabb szintre |
| Lefokozás    |  |
| Feljebb      | Az elem elhelyezése a vázlatba felfelé vagy lefelé                                 |
| Lejjebb      |  |
| Kibontás     | A szöveg megjelenítésére vagy elrejtésére  |
| Elrejtés     |  |
|  Рівень 3    | A vázlatban megjelenítendő szintek kiválasztása                                    |

*Érdemes megjegyezni, hogy a cím kijelöléséhez vagy mozgatásához a belső szöveg kijelölése és áthelyezése társul.*

Ha a dokumentum egy szerkezeti elemének szeretnénk látni az alárendelt szintjeit, akkor használjuk a ,  és a  gombokat. Függőleges vagy vízszintes vonalak jelennek meg a képernyőn, jelezve a szöveg új helyét.

A Vázlat kinyomtatható. Papíron ugyanúgy fog kinézni, mint a képernyőn. A bekezdésformázások figyelmen kívül lesznek hagyva, míg a karakterformázásokat figyelembe veszik.



### Használjuk a számítógépet!

Végezzétek el a gyakorlati feladatot a <https://cutt.ly/jhde4U7> linkről, vagy a QR-kód segítségével!

### Összefoglalás


Az objektum formázásának megkönnyítése céljából stílusokat alkalmazhatunk. A **Stílus** az adott objektum paraméter értékeinek névvel ellátott gyűjteménye. A **Wordben** a különböző objektumokhoz **stílusgyűjtemények** állnak rendelkezésünkre. Ezeket a stílusokat a dizájn követelményeinek alapján dolgozták ki, és harmonikusan egyesítik az objektumok tulajdonságainak értékeit. Stílusok a szöveges dokumentum különböző objektumaihoz – szöveg, táblázat, grafikai objektumok stb. – vannak kifejlesztve. A szöveges elemekhez rendelhető stílusokat a **Kezdőlap menü Stílustárában** találjuk. A egyéb objektumok (táblázatok, képek, ábrák, grafikonok stb.) stílusainak kiválasztására és szerkesztésére az ideiglenes **Tervezés, Elrendezés, Formátum** menüket használják.

A **Téma** a dokumentum egészére alkalmazott stílus. A témáknak egyedi neve van, és stíluselőírásokat tartalmaznak a dokumentumban alkalmazott objektumokra.

A **Word** beépített témáit az **Elrendezése** menü **Témák** csoportjában található **Témák** legördülő menügombbal érhetjük el.

A különböző típusú szöveges dokumentumok elkészítésekor be kell tartani azok kialakítására, felépítésére és tartalmára vonatkozó egységes követelményeket és szabályokat, amelyeket állami szabványok, vállalati követelmények, az üzleti dokumentáció szabályai határoznak meg.

A **dokumentum szerkezete** alatt a dokumentum összetevői által alkotott hierarchikus vázlatot értjük. Ilyet leggyakrabban a hosszú, több fejezetből, paragrafusból álló dokumentumban hozunk létre.

A dokumentum struktúráját a címsorstílusok – **Címsor 1, Címsor 2, ..., Címsor 9** – segítségével hozzuk létre. Az ilyen módon strukturált dokumentum szerkezetét **Vázlat** nézetben tekinthetjük meg. Ebbe a nézetbe a **Vázlat** gomb vagy a **Nézet** menü **Nézetek** csoportjának **Vázlat**  eszköze segítségével léphetünk át. A dokumentum **Vázlat** nézetben történő megtekintésének előnye az, hogy testreszabható a dokumentum bármely szintjének megjelenítése, miközben elrejtjük azokat a szinteket, amelyek az alsó hierarchiát foglalják el.

**Vázlat** nézetben kényelmes a dokumentum felépítésének szerkesztése a szövegrészek szintjének és elhelyezésük sorrendjének megváltoztatásával, a **Vázlateszközök eszközcsoport vezérlőelemei** vezérlőivel.



### Felelj a kérdésekre!

- 1°. Mik a stílusok? A **Word** milyen objektumaihoz alkalmazhatunk stílusokat?
- 2°. A stílusok az objektumok mely tulajdonságaira vannak hatással? Mire használják ezeket?
- 3°. Hogyan alkalmazhatóak a stílusok a szöveges dokumentum különböző objektumaira?
- 4°. Melyek a dokumentumok stilisztikai tervezésének általános szabályai?
- 5°. Mi a dokumentum témája? A dokumentum milyen tulajdonságait jellemzik?
- 6°. Mire szolgál a dokumentum **Vázlat** nézet? Hogyan lehet bekapcsolni?
- 7°. Mi a dokumentum szerkezete? Hozzatok fel példákat.
- 8°. Mik a címsorstílusok? Hányféle címsor áll rendelkezésünkre a **Word**ben?
- 9°. Milyen változtatásokat lehet elvégezni a **Vázlat** nézetben?





### Végezd el a feladatokat!

- 1°. Nyisd meg a tanárod által megadott fájlt (például a **3fejezet\ 3.3.rész\ minta.3.3.1 docx**-et)! Formázd a dokumentumot az alábbi stílusokkal:
  - címek – *Könyv címe* stílus;
  - szöveg – *Erős kiemelés*;
  - táblázat – *Táblázat (rácsos) 1 világos, 2 jelölőszín*;
  - kép – *Fémszerű ovális*!



- 2°. Nyisd meg a tanárod által megadott fájlt (például a **3fejezet\ 3.3.rész\ minta.3.3.2 docx**-et)! Ez a dokumentum stílusok segítségével van formázva. Határozd meg, milyen értékeket rendeltek az objektumok tulajdonságaihoz az alkalmazott stílusokban! Az eredményeket írd be a füzetedbe!

| Objektum sorszáma | Objektum stílusa | Stílus | Tulajdonság értékei |
|-------------------|------------------|--------|---------------------|
| 1                 |                  |        |                     |

- 3°. Nyisd meg a tanárod által megadott fájlt (például a **3fejezet\3.3.rész\minta.3.3.3.docx**-et)! Határozd meg, hogy a dokumentum esetében melyik téma van használva! Változtasd meg a dokumentum témáját *Fazettára*, majd pedig – *Integrálra*! Figyelmesen vizsgáld meg, hogy a dokumentumobjektumok tulajdonságainak mely értékei változnak, amikor módosítottad a dokumentum témáját! Írd le a következtetést a füzetbe! Melyik téma tetszett a legjobban? Miért?
- 4\*. Keresd meg az interneten a ДСТУ 4163-2003 állami szabvány szövegét, és ismerkedj meg a hivatalos dokumentumokban használt táblázatok és ábrák formázásának szabályaival! Jegyzeteld be őket a füzetedbe! Kérdezd meg szüledet, milyen követelményeket támasztanak munkahelyükön a dokumentumok formázásával kapcsolatban!
- 5°. Nyisd meg a tanárod által megadott fájlt (például a **3fejezet\3.3.rész\minta.3.3.5.docx**-et)! Formázd meg a dokumentumok azon stilisztikai követelmények szerint, melyek fentebb találhatóak! Mentsd a dokumentumot **feladat 3.3.5.docx** néven a mappádba!
-  6\*. Alkossatok három fős csoportokat! Dolgozzatok ki saját stílust *A világ csodái* nevű civil szervezet számára! Írjátok be az alapvető követelményeket a füzetetekbe! Formázzátok meg a tanár által megadott fájlt (például a **3fejezet\3.3.rész\minta.3.3.6.docx**-et) az általatok kidolgozott követelmények felhasználásával! Mentsétek a dokumentumot **feladat 3.3.6.docx** néven a mappátokba! Mutassátok be a munkátokat! Melyik munka tetszett a legjobban? Miért?
-  7°. Nyisd meg a tanárod által megadott fájlt (például a **3fejezet\3.5.rész\minta.3.3.7.docx**-et)! Formázd meg a dokumentumot a következő stílusok segítségével:
- első szintű címek – *Címsor 1*
  - második szintű címek – *Címsor 2*
  - harmadik szintű címek – *Címsor 3*
  - egyéb szöveg – *Normál!*
- Tekintsd meg a dokumentum struktúráját! Mentsd a dokumentumot **feladat 3.3.7.docx** néven a mappádba!
- 8\*. Nyisd meg a tanárod által megadott fájlt (például a **3fejezet\3.3.rész\minta.3.3.8.docx**-et)! Tekintsd meg **Vázlat** nézetben! Rendezd a dokumentum fejezeteit számuk szerint sorba! Szükség szerint állítsd be a szövegrészek szintjeit! Mentsd a dokumentumot **feladat 3.3.8.docx** néven a mappádba!

### 3.4. Tartalomjegyzék létrehozása. Hivatkozások. Dokumentumokkal való csapatmunka



1. Mik a stílusok? Mire használjuk őket?
2. Mi a hiperhivatkozás? Mire használjuk? Hogyan készíthetünk hiperhivatkozást a prezentációkban?
3. Hogyan történik a dokumentumokkal való csapatmunka a **Google** felhőkörnyezetben? Milyen hozzáférési jogok adhatók a felhasználóknak?

#### *Automatikus tartalomjegyzék létrehozása*

Az előző részekben megismerkedtetek a szöveges dokumentumok feldolgozásának néhány lehetőségével – a dokumentum felépítésének megtekintésével, szakaszok létrehozásával, stilisztikai tervezéssel stb. Ezenkívül a **Word** szövegszerkesztőben



vannak más eszközök is a hosszú dokumentumok létrehozásának és feldolgozásának leegyszerűsítésére.

Megismerkedünk a tartalomjegyzék automatikus létrehozásával.

A **tartalomjegyzék**et olyan táblázat formájában alakították ki, amely tartalmazza a dokumentum szerkezeti egységeit és azokat az oldalszámokat, ahol ezek az egységek kezdődnek (3.25. ábra).

**Зміст**

|   |   |
|---|---|
| Програма ЮНЕСКО «Інформація для всіх» .....   | 1 |
| 1. Преамбула .....  | 1 |
| 2. Мандат .....   | 2 |
| 3. Цілі програми .....  | 2 |
| 4. Основні розділи програми .....   | 3 |
| Розділ 1. Розвиток інформаційної політики на міжнародному, регіональному і національному рівнях ..... | 3 |
| Розділ 2. Розвиток людських ресурсів, навиків і умінь у століття інформації .....                     | 3 |
| Розділ 3. Посилення ролі інституцій в забезпеченні доступу до інформації .....                        | 3 |
| Розділ 4. Розвиток інструментів, способів і систем інформаційного менеджменту .....                   | 4 |
| Розділ 5. Інформаційні технології для освіти, науки, культури і комунікацій .....                     | 4 |
| 5. Принципи здійснення програми .....   | 4 |
| 1. Партнерство і співпраця .....  | 4 |
| 2. Оцінки .....   | 5 |
| 3. Принципи здійснення проекту .....  | 5 |

3.25. ábra. Dokumentum tartalomjegyzéke (minta)

A tartalomjegyzékét általában a szöveg elejére vagy végére helyezük el.

Minden tartalomjegyzék hiperhivatkozás is egyben a dokumentum megfelelő címére.

Ha a hosszú dokumentum valamennyi strukturális egységét a megfelelő címsor-stílussal formáztuk, a dokumentum tartalomjegyzéke automatikusan létrehozható. Ehhez:

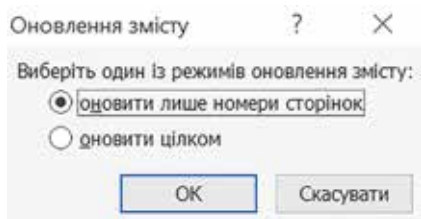
1. Helyezzük a kurzort a dokumentum azon pontjára, ahová a tartalomjegyzékét be szeretnénk illeszteni!
2. Hajtsuk végre a **Hivatkozás** => **Tartalomjegyzék** => **Tartalom** parancssort!
3. Válasszunk a beépített stílusok közül (3.26. ábra)!

A tartalomjegyzék segítségével könnyedén navigálhatunk a dokumentumban, hamar megtalálhatjuk a fejezeteket, pontokat. E célból elegendő lenyomott **Ctrl** billentyű mellett a tartalomjegyzék megfelelő elemére kattintani. Figyeljétek meg, hogy a művelet elvégzése során megváltozik az egérmutató alakja !

A **Wordben** elkészített dokumentum tartalomjegyzéke a szövegszerkesztő szokásos eszközeivel szerkeszthető és formázható.



3.26. ábra. Tartalomjegyzék – stílusok



3.27. ábra. Tartalomjegyzék frissítése ablak

elemek helye megváltozott a dokumentum szerkesztése vagy formázása miatt). A dokumentumban található tartalomjegyzék törléséhez jelöljük ki a tartalmat és nyomjuk meg a **Delete** gombot!

Amennyiben a szerkesztés során megváltoztattuk a dokumentum szövegét vagy szerkezetét, a tartalomjegyzéket frissíteni kell. E célból hajtsuk végre a **Hivatkozások** ⇒ **Tartalomjegyzék** ⇒ **Frissítés műveletet** . Ezután a **Tartalomjegyzék frissítése** ablakban (3.27. ábra) a következőket választhatjuk: teljes jegyzék frissítése (ha változtatásokat hajtottunk végre a dokumentum szerkezeti elemeinek címsorain), vagy csak az oldalszámok frissítése (ha a szerkezeti

### Hivatkozások készítése a szöveges dokumentumokban

A dokumentum tartalomjegyzéke a szöveges dokumentum kézenfekvő objektuma, de a dokumentumon belül csak annak szerkezeti elemeire nyújt információt. Egy szöveges dokumentumba **hiperhivatkozások** használatával a dokumentum más objektumára is ugorhatunk. Az ilyen hivatkozásokkal már megismerkedtünk, amikor weblapokkal dolgoztunk vagy amikor prezentációkat készítettünk.

A dokumentumok különféle hivatkozásokat – olyan speciális bejegyzéseket, amelyek a dokumentum egyéb helyeire mutatnak – tartalmaznak. A hivatkozások a dokumentum egyes részeinek vagy objektumainak logikai összekapcsolására szolgálnak. A szövegben egy táblázatra, egy rajzra, egy diagramra, forrásdokumentumra mutató hivatkozást helyezhetünk el, ezeket *belső* hivatkozásoknak nevezzük. A hivatkozások lehetnek *külső* is, amikor egy másik honlagra, **Google Drive**-ra vagy a hálózatokon lévő fájlra mutató hivatkozást helyezünk el a dokumentumban. Ha a hivatkozásra kattintunk, automatikusan átlépünk a hivatkozott objektumra.

Például egy fizika projektben végrehajtott csoportos munka beszámolójában nem szükséges minden anyagot egy dokumentumba zsúfolni, hanem elegendő a csoporttagok **Google Drive**-ra feltöltött anyagok hiperhivatkozásait bemásolni a dokumentumba. Ezáltal a beszámoló átláthatóbb és kompaktabb lesz, valamint időt takarítunk meg az elkészítésével.

A szövegben lévő hiperhivatkozásokat össze kell kapcsolni a dokumentum valamely objektumával, ez lehet egy szó, egy szövegdarab, kép stb. A hiperhivatkozás követéséhez ki kell választania ezt az objektumot a kurzorral, miközben a **Ctrl** billentyűt lenyomva tartjuk. A hiperhivatkozás szövege egy szöveges dokumentumban, valamint a weboldalakon és a prezentációkban más színű karakterekkel van kiemelve és aláhúzva.

A belső hiperhivatkozások létrehozása előtt meg kell formázni a dokumentum összes objektumát, amelyre navigálni szeretnénk a **Címsor** stílusok különböző típusaival, vagy a megfelelő **könyvjelzők** létrehozásával.



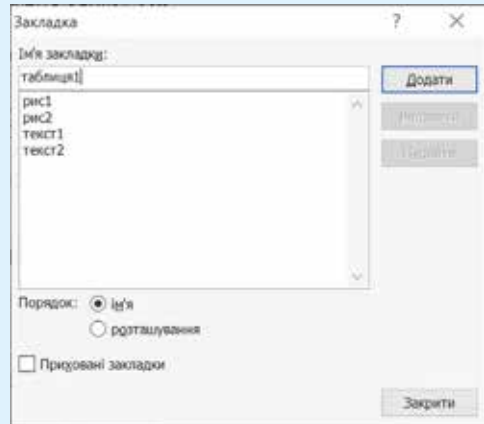
#### Ha többet szeretnél tudni

A **könyvjelzővel** olyan helyet jelölünk a dokumentumban, amelyet könnyen meg szeretnénk találni. Adhatunk neki egyedi nevet, így könnyen azonosíthatók. A könyvjelző nevének betűvel kell kezdődnie. Tartalmazhatnak betűket, számokat, különböző szimbólumokat, de szóközt nem. Könyvjelzőt a szöveges dokumentum bármely objektumához (szó, szövegrészlet, kép táblázat) készíthetünk.

A könyvjelzők felvételének folyamata:

1. Jelöljétek ki azt a helyet a dokumentumban, ahova könyvjelzőt szeretnétek beszúrni!
2. Kattintsatok a **Beszúrás** menü **Hivatkozások** fül **Könyvjelzők** gombra!
3. A **Könyvjelző neve** mezőbe írjátok be egy nevet (3.28. ábra)!
4. Kattintsatok a **Hozzáadás** gombra!

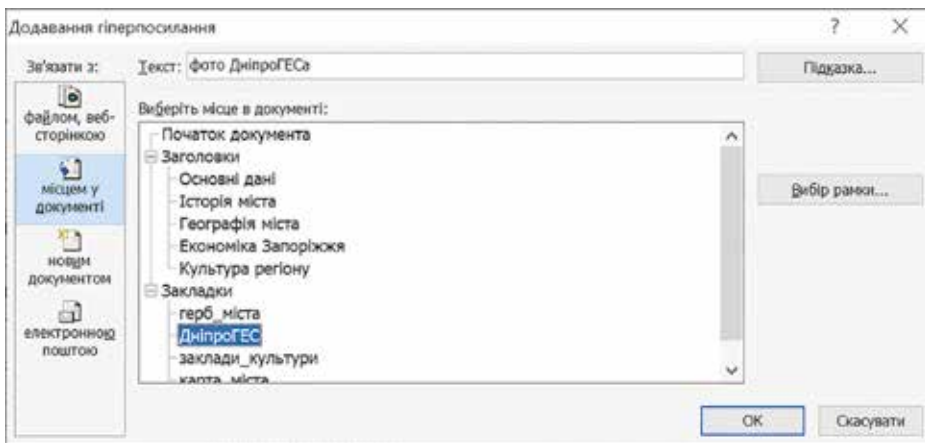
Ha törölni szeretnétek egy könyvjelzőt, elegendő a törölni kívánt könyvjelző nevére kattintani, majd pedig a **Törlés** gombra.



3.28. ábra. A könyvjelző ablak

Belső hiperhivatkozás létrehozásához szöveges dokumentumban, hajtsuk végre az alábbi lépéseket:

1. Formázzuk meg a dokumentumnak azt az objektumát, amelyre hivatkozunk majd a **Címsor** stílusok különböző típusaival, vagy a megfelelő **könyvjelzők** létrehozásával!
2. Jelöljük ki a dokumentum azon objektumát, amelyhez kötjük a hivatkozást (szövegrészt, képet stb.)!
3. Végezzük el a **Beszúrás** ⇒ **Hivatkozások** ⇒ **Hivatkozás** műveletet!
4. A **Hivatkozás beszúrása** ablakban válasszuk ki a **Hivatkozás erre:** listából **A dokumentum adott pontja** lehetőséget (3.29. ábra)!

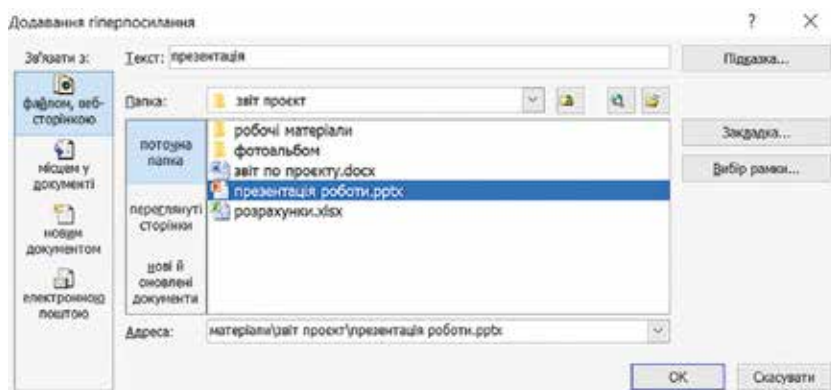


3.29. ábra. Belső hivatkozás készítésére szolgáló **Hivatkozás beszúrása** ablak

5. A **Válasszon egy dokumentumon belüli pontot** mezőből válasszuk ki azt a **Címsort** vagy **Könyvjelzőt**, amelyre ugrani fogunk!
6. Nyomjuk meg az **OK** gombot!

Külső hiperhivatkozás létrehozásához a helyi hálózaton vagy a számítógép adathordozóján lévő fájlhoz hajtsuk végre az alábbi lépéseket:

1. Jelöljük ki a dokumentum azon objektumát, amelyhez kötjük a hivatkozást (szövegrészt, képet stb.)!
2. A **Hivatkozás beszúrása** ablakban válasszuk ki a **Hivatkozás erre:** listából a **Létező fájl**, vagy **weblap** lehetőséget!
3. Nyissuk meg a megfelelő mappát, majd válasszuk ki a megfelelő fájlt (3.30. ábra)!
4. Nyomjuk meg az **OK** gombot!

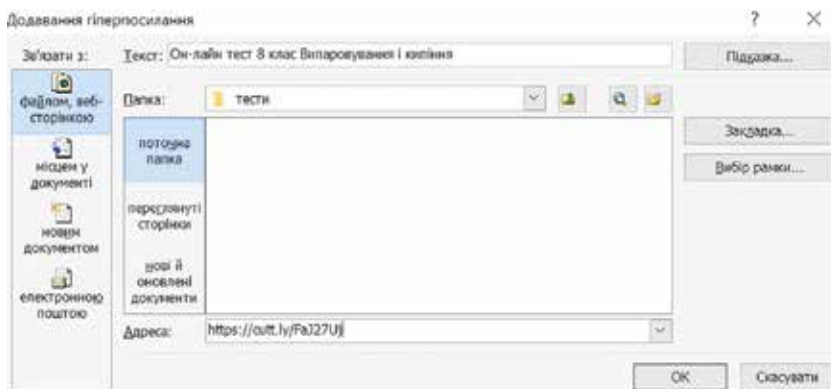


3.30. ábra. Külső hivatkozás készítésére szolgáló **Hivatkozás beszúrása** ablak

Az ilyen hiperhivatkozás kiválasztása után megnyílik az a párbeszéd ablak, amelyet az ilyen típusú fájlok kezelésére terveztek és megnyílik benne a kiválasztott fájl. Például, ha a hiperhivatkozás létrehozása során egy prezentációs fájlt választottunk ki, akkor a hiperhivatkozás kiválasztása után megnyílik a **Microsoft PowerPoint** alkalmazás, és a kiválasztott dokumentum betöltődik abba.

Hiperhivatkozás weblapra történő beillesztéséhez egy szöveges dokumentumban hajtjuk végre az alábbi lépéseket:

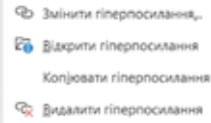
1. Jelöljük ki a dokumentum azon objektumát, amelyhez kötjük a hivatkozást (szövegrészt, képet stb.)!
2. A **Hivatkozás beszúrása** ablakban válasszuk ki a **Hivatkozás erre:** listából a **Létező fájl**, vagy **weblap** lehetőséget!
3. Írjuk be vagy másoljuk be a kívánt weboldal címét a **Cím** mezőbe (3.31. ábra)!
4. Nyomjuk meg az **OK** gombot!



3.31. ábra. Hiperhivatkozás weblapra történő beillesztéséhez szolgáló **Hivatkozás beszúrása** ablak

Miután kiválasztottunk egy ilyen hivatkozást a szóveges dokumentumban, megnyílik egy böngészőablak, amely megjeleníti azt a weboldalt, amelynek címét a **Cím** mezőbe írjuk be.

A beillesztett hiperhivatkozások módosíthatók, másolhatók vagy törölhetők a hivatkozás helyi menü parancsai segítségével (3.32. ábra).



3.32. ábra. A hivatkozás helyi menüjének parancsai

**Кözös munka dokumentumokkal**

A modern szóvegszerkesztő eszközei képesek arra, hogy több felhasználó által megszervezzék a dokumentumokkal való közös munkát. Ilyen eszközökre van szükség ahhoz, hogy a különböző felhasználók, a csoportmunka résztvevői együttműködhessenek a különféle dokumentumok elkészítésében, ugyanakkor nyomon kövessék a társszerzők által a dokumentumban végrehajtott változásokat.

Már tudjátok, hogyan kell dolgozni a dokumentumokkal a **Google** felhőszolgáltatásainak használatával, ahol a különböző felhasználók megfelelő hozzáféréssel együttesen hozhatnak létre dokumentumokat, tekinthetik meg és szerkeszthetik azokat, kommentálhatják és változtatásokat végezhetnek, megvitathatják a felmerülő kérdéseket. Így a társszerzők különböző helyeken tartózkodhatnak, mégis ezt a munkát egyidejűleg vagy számukra megfelelő időben végezhetik.

A **Microsoft Office 365** szoftvercsomagjában található **Word** szóvegszerkesztő legújabb verziójában, amely analóg a **Google** felhőszolgáltatásával, megtalálhatóak azok az eszközök, amelyek képesek a dokumentumokkal való együttműködés (csapatmunka) megszervezéséhez.

A számítógépre telepített **Microsoft Word** szóvegszerkesztőnek szintén vannak eszközei az együttműködésre. A dokumentum azonban nem kerül egyidejűleg feldolgozásra, és a különböző felhasználók felváltva szerkesztik a dokumentumot, offline módban a számítógépükön, valamint a módosított dokumentumot elküldik más társszerzőknek véleményezésre és további szerkesztésre.

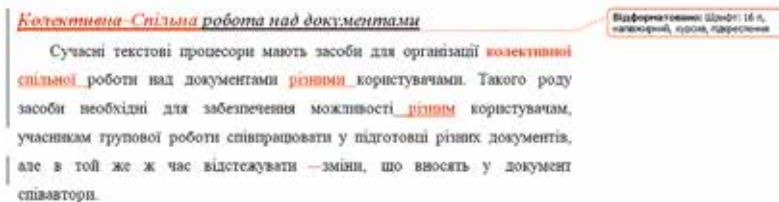
A szóveges dokumentumok ilyen feldolgozása a **Wordben** a **Véleményezés** nézetben történik. Ezt a nézetet a következőképpen kapcsolhatjuk be: **Véleményezés** →

**Nyomon követés** (3.33. ábra) **Változások követése** gomb.



3.33. ábra. A **Véleményezés** menü lehetőségei

Ha a nézet engedélyezése után szerkesztjük és formázzuk a dokumentumot, akkor minden változás megjelenik a dokumentumban. A dokumentumszerkesztési műveletek megjelennek a dokumentum szövegében, és a jobb mezőben – a formázási műveletek végrehajtnak.





3.34. ábra. **Szövegrész** kinézete a **Véleményezés** nézetben

Például, ha véleményezési nézetben törölünk egy szót, vagy kicseréljük egy másikra, akkor az nem törlődik a szövegből, hanem az át lesz húzva, és a beszúrt szöveg más színnel lesz kiemelve (3.34. ábra).

A különböző felhasználók által végrehajtott módosítások a dokumentumban különböző színben jelennek meg. A szakértői véleményezésű dokumentumok mentésekor az összes módosítás mentésre kerül.

Egy ilyen dokumentum megnyitása után a dokumentum szerzője (vagy egy másik felhasználó) láthatja a változásokat, elolvashatja és eldöntheti, hogy elfogadja vagy elutasítja ezeket a módosításokat. Elutasítás esetén az elvégzett változtatások eltűnnek, elfogadás esetén pedig véglegesen mentődnek.

Egyetlen javítás feldolgozásához ki kell jelölni a javítandó részt és a **Véleményezés** menü **Változások** fülben az **Elfogadás**  vagy pedig az **Elvetés**  gombok közül kell választani. Az előző vagy a következő javításhoz való visszatéréshez válasszuk a **Vissza** vagy a **Következő** gombot!

Ha a dokumentum szerzője elolvasta a módosításokat és egyetért velük, akkor megteheti, hogy az **Elfogadás** parancsok közül kiválasztja a **Minden módosítás elfogadása** lehetőséget vagy pedig, ha nem ért egyet a módosításokkal, akkor az **Elvetés** parancsok közül a **Minden módosítás elvetése** lehetőséget.

Az összes javítás feldolgozása után egy üzenet jelenik meg a képernyőn: *Nincsenek megjegyzések vagy követett változások.* Ezt követően a véleményezési nézet kikapcsolható a **Változások követése** gomb újbóli kiválasztásával, és az új szerkesztések már nem követhetők.



### Ha többet szeretnél tudni

A véleményezendő dokumentumra a dokumentum szerzője korlátozhatja a hozzáférési jogokat a dokumentum közös szerkesztése során, éspedig: megtilthatja a dokumentum stílusok által történő szerkesztését, beállíthatja, hogy a dokumentum csak olvasható legyen, meghatározhatja, hogy mely felhasználók jogosultak a szerkesztésre, jelszavakat tehet a módosításokra. Minden ilyen korlátozás beállítása elvégezhető a **Véleményezés** menü **Védelem** fülének **Szerkesztés korlátozása** lehetőséggel.



### Használjuk a számítógépet!



Végezzétek el a gyakorlati feladatot a <https://cutt.ly/Uhde5qq> linkről, vagy a QR-kód segítségével!



### Összefoglalás

A **tartalomjegyzék** egy olyan lista, amely tartalmazza a dokumentum szerkezeti egységeit és azokat az oldalszámokat, ahol ezek az egységek kezdődnek. A tartalomjegyzéket általában a szöveg elején vagy végén helyezzük el. Minden tartalomjegyzék hiperhivatkozás is egyben a dokumentum megfelelő címére.

Ha a dokumentum valamennyi strukturális egységét a megfelelő címsorstílussal formáztuk, a dokumentum tartalomjegyzéke automatikusan létrehozható: **Hivatkozás** ⇒ **Tartalomjegyzék** ⇒ **Tartalom**. Amennyiben a szerkesztés során megváltoztattuk a dokumentum szövegét vagy szerkezetét, a tartalomjegyzéket frissíteni kell. E célból hajtunk végre a **Hivatkozások** ⇒ **Tartalomjegyzék** ⇒ **Frissítés** műveletet!



A szöveges dokumentumba **hiperhivatkozások** használatával a dokumentum bármely objektumára ugorhatunk. Hiperhivatkozások létrehozhatók egyetlen dokumentumon belül (*belső hiperhivatkozás*) vagy külső forrásokhoz kötve (*külső hiperhivatkozás*). A szövegben lévő hiperhivatkozásokat össze kell kapcsolni a dokumentum valamely objektumával, ez lehet egy szó, egy szövegdarab, kép stb. A hiperhivatkozás szövege más színű karakterekkel van kiemelve és aláhúzva.

A belső hiperhivatkozások létrehozása előtt meg kell formázni a dokumentum összes objektumát, amelyre navigálni szeretnénk a **Címsor** stílusok különböző típusaival, vagy a megfelelő **könyvjelzők** létrehozásával. Hiperhivatkozásokat a **Hivatkozás beszúrása** párbeszédablakban készíthetünk, amelyet a **Beszúrás** ⇒ **Hivatkozások** ⇒ **Hivatkozás** művelet végrehajtásával nyithatunk meg.

A **Microsoft Word** szövegszerkesztő a véleményezési nézetet használja a dokumentumokkal való együttműködés megszervezésére, amelyet a **Véleményezés** ⇒ **Nyomon követés** ⇒ **Változások követése** gombbal érhetünk el. Ha a nézet engedélyezése után szerkesztjük és formázzuk a dokumentumot, akkor minden változás megjelenik a dokumentumban. A szakértői véleményezésű dokumentumok mentésekor az összes módosítás mentésre kerül.

Egy ilyen dokumentum megnyitása után a dokumentum láthatja a változásokat, elolvashatja és eldöntheti, hogy elfogadja vagy elutasítja ezeket a módosításokat. A szerkesztések elvégzését a **Véleményezési** menü vezérlőivel tehetjük. A módosítások feldolgozása után ki kell kapcsolni a véleményezési nézetet és a dokumentumban végrehajtott új változtatásokat már nem lehet ilyenkor nyomon követni.



### Felelj a kérdésekre!





- 1<sup>o</sup>. Mik a hivatkozások egy dokumentumban? Mire használhatjuk?
- 2<sup>o</sup>. Mi a tartalomjegyzék? Mi a rendeltetése?
- 3<sup>o</sup>. Hogy hozunk létre **Wordben** tartalomjegyzéket?
- 4<sup>o</sup>. Mi a hiperhivatkozás egy szöveges dokumentumban? Mire használják őket?
- 5<sup>o</sup>. Egy szöveges dokumentumban milyen típusú hiperhivatkozásokat hozhatunk létre?
- 6\*. Mi a közös és mi a különbség a belső és külső hiperhivatkozások létrehozásakor a dokumentumban?
- 7<sup>o</sup>. Mi az a könyvjelző? Mire használják?
- 8<sup>o</sup>. Mi az a dokumentum véleményezési nézete? Mire használják?
- 9<sup>o</sup>. Hogyan tekintheti meg a felhasználó a dokumentum javításait? Miként jelennek meg a dokumentumban?
- 10<sup>o</sup>. Milyen műveletek hajthatók végre a véleményezési nézetben?
- 11\*. Hogyan lehet megszervezni a dokumentummal való együttműködést (csapatmunkát) egy személyi számítógépre telepített **Word** szövegszerkesztő segítségével?



### Végezd el a feladatokat!



- 1<sup>o</sup>. Nyisd meg a tanárod által megadott fájlt (például a **3fejezet\ 3.4.rész\min-ta.3.4.1 docx**-et)! Formázd meg a dokumentum szerkezeti elemeit a megfelelő szintű címsorokkal! Illeszd be a dokumentum elejére a tartalomjegyzéket! Mentse a dokumentumot **feladat 3.4.1.docx** néven a mappádba!
- 2<sup>o</sup>. Nyisd meg a tanárod által megadott fájlt (például a **3fejezet\ 3.4.rész\min-ta.3.4.2 docx**-et)! Illeszd be a dokumentum végére a tartalomjegyzéket! Mentse a dokumentumot **feladat 3.4.2.docx** néven a mappádba!

- 3°. Nyisd meg a tanárod által megadott fájlt (például a **3fejezet\3.4.rész\minta.3.4.3.docx**-et)! Írd bele a dokumentumba az *Ukrajna 7 csodája* projekt nyerteseinek nevét, valamint ezekről a csodákról szóló Wikipedia-cikkek hivatkozásait! Mentse a dokumentumot **feladat 3.4.3.docx** néven a mappádba!
-  4\*. Hozz létre egy szöveges dokumentumot, amely tartalmazza a barátaid vezetőnevét, keresztnévét és születési dátumát! Készíts mindegyikük nevéhez hivatkozást, amely a közösségi hálózatokon található oldalukra mutat! Mentse a dokumentumot **feladat 3.4.4.docx** néven a mappádba!
-  5°. Alkossatok párokat! Nyissátok meg a tanárotok által megadott fájlt (például a **3fejezet\3.4.rész\minta.3.4.5.docx**-et)! A dokumentumba szereplő minden egyes eszköz nevéhez készítsetek hiperhivatkozásokat, amelyek a **3fejezet\3.4.rész\3.4.5kép.docx** fájlban szereplő képekre utalnak! Mentsetek a dokumentumot **feladat 3.4.5.docx** néven a mappátokba! Magyarazzátok el, hogy előnyös volt-e az, hogy párban dolgoztatok! Voltak-e félreértések a közös munka során?
-  6\*. Hozz létre egy táblázatot a szöveges dokumentumban, amely tartalmazza régiótok öt történelmi emlékművének leírását, megjelölve a nevét és a helyét! Adj hozzá az objektum nevéhez az interneten megtalálható képekre utaló hiperhivatkozást, mentse saját **Google Drive**-odba! Adj hozzá hiperhivatkozást az emlékmű **Google Térkép**éhez! Mentse a dokumentumot **feladat 3.4.6.docx** néven a mappátokba!
- 7°. Nyisd meg a tanárotok által megadott fájlt (például a **3fejezet\3.4.rész\minta.3.4.7.docx**-et)! Válaszolj a szövegben az internet veszélyeivel és kockázataival kapcsolatos kérdésekre! Formázd meg a fájlt saját belátásod szerint! Tekintsd meg a dokumentumot véleményezési nézetben! Fogadd el az összes elvégzett javítást! Mentse a dokumentumot **feladat 3.4.7.docx** néven a mappádba!
-  8\*. Hozz létre egy szöveges dokumentumot **feladat 3.4.8.docx** néven és hozz létre benne öt szabályt az interneten való biztonságos viselkedésről! Küldd el véleményezésre elektronikusan barátodnak/barátnődnek! A barátnődötől/barátodtól visszakapott dokumentumot véleményezés nézetben tekintsd meg! Mentse el a dokumentumot a mappádban **feladat 3.4.8.docx** néven!

#### 4. SZÁMÚ GYAKORLATI MUNKA

##### Stílusok. A dokumentum felépítése. Automatikus tartalomjegyzék készítése. Hiperhivatkozás

**Figyelem!** A számítógép használata során tartsd be a balesetvédelmi szabályokat és az egészségügyi előírásokat!

1. Nyisd meg a tanár által megadott fájlt (például a **3fejezet\3.4.rész\minta gym4.docx**-et)!
2. A szövegben szereplő számozásokhoz állítsd be az *1. szintet*, a felsorolásjelekkel ellátottakhoz pedig a *2. szintet*!
3. A **Vázlat** nézetbe rendezd a dokumentum szerkezeti elemeit a pontoknak megfelelő sorrendbe!
4. Stílusok alkalmazásával formázd a táblázatokat és a rajzokat!
5. Formázd meg a szöveges dokumentum szerkezeti részeinek nevét a megfelelő szintű címsor stílusokkal!
6. A dokumentum elejére készítsd el annak tartalomjegyzékét!

7. Minden képhez csatolj hiperhivatkozást, mely a megfelelő internetes forrásra utal majd!
8. Mentsd a dokumentumot a mappádba **gyakorlati munka 4\_ Vezetéknév.docx** néven!
9. Helyezd a dokumentumot a tanár által megadott, megosztott mappába!
10. A megosztott mappából nyisd meg a tőled jobbra lévő társad 4. számú gyakorlati munkáját, majd hajtsd végre a következő javításokat: töröld a dokumentum tartalomjegyzékét, és változtasd meg a táblázat és az egyik ábra tervezési stílusát!
11. Mentsd el a fájlt a javításokkal **gyakorlati munka 4\_ Vezetéknév\_javított.docx** néven a megosztott mappába!
12. Nyisd meg a saját 4. számú gyakorlati munkádat a szomszédod szerkesztése utáni változatban!
13. Ellenőrizd a javításokat, utasítsd el a tartalomjegyzék törlését, és fogadd el az objektum formázásának változását!
14. Mentsd a dokumentumot a mappádba **gyakorlati munka 4\_ Vezetéknév\_végleges.docx** néven!
15. Zárd be a **Word** alkalmazást!

### 3.5. Speciális grafikai objektumok létrehozása a szövegszerkesztőkben



1. Hogy írhatunk a szövegbe olyan matematikai jeleket, mint például  $x_3$ ,  $x^3$ ,  $\frac{3}{8}$ ,  $\alpha + \beta$ ?
2. Milyen grafikai objektumokat helyezhetünk el a szövegben? Hogyan?
3. Milyen szerkesztési és formázási műveleteket végezhetünk el a **Word** grafikai objektumain? Milyen eszközökkel tehetjük ezt meg?

#### Képletek készítése a szöveges dokumentumokban

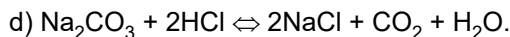
Ha egy referátumot írunk, vagy egy diákonferencián tartandó előadásunkhoz készülünk, szükség lehet arra, hogy a szövegbe képleteket, egyenleteket, tételek bizonyításait illesszük be, amihez speciális matematikai, fizikai, kémiai jelek szükségesek.

Az egyszerű matematikai kifejezések lineáris alakúak, vagyis az elemeit egy sorba írhatjuk. Az ilyen kifejezés nem tartalmaz gyökjeleket, törtet és egyéb nemlineáris struktúrákat. Például:

$$a) c_2 m_2 (t_2 - \Theta) = c_1 m_1 (\Theta - t_1);$$

$$b) ax^2 + bx + c \geq 0;$$

$$c) S = v_0 t + at^2/2;$$



Az ilyen kifejezések beírása során szükség lehet speciális szimbólumokra, amelyekkel a **Beszűrés**  $\Rightarrow$  **Szimbólumok**  $\Rightarrow$  **Szimbólum**  $\Omega$   $\Rightarrow$  **Egyéb szimbólumok** eszközzel illeszthetünk be. Ezenkívül használhatjuk még a **Kezdőlap**  $\Rightarrow$  **Betűtípus** csoportjának **Alsó index**  $x_1$  és **Felső index**  $x^2$  eszközét.

Előfordulnak azonban nem lineáris többszintes struktúrák is, például:

$$a) \frac{h_1}{h_2} = \frac{\rho_1}{\rho_2};$$

$$b) E_k = \frac{mv^2}{2};$$

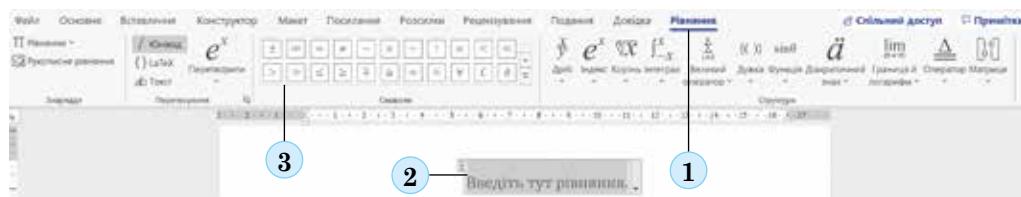
$$c) \eta = \frac{A_1}{A_2} \cdot 100 \%;$$

$$d) \frac{3x+7}{4} - \frac{x-3}{2} = \frac{5x+2}{8};$$

$$e) y = \begin{cases} -3x^3 - 2, & \text{ha } x < 0, \\ \frac{7x}{4} - 2, & \text{ha } x \geq 0; \end{cases} \quad f) \frac{5a^4 - 20b^2}{2a^2 + b^2} : \frac{30(a-4b)^2}{9a^4 - b^4}.$$

Ilyen képleteket **Egyenletszerkesztő** segítségével készíthetünk, amely tartalmazza az ilyen struktúrák beillesztéséhez szükséges sablonokat. A benne lévő képletek külön szimbólumokból és struktúrákból épülnek fel, a megfelelő sablonok felhasználásával.

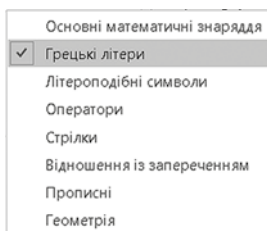
Az **Egyenletszerkesztőt** a **Beszúrás**  $\Rightarrow$  **Szimbólumok**  $\Rightarrow$  **Egyenlet**  $\Pi$  parancssal indíthatjuk. Indítása után a **Menüszalagon** megnyílik az **Egyenleteszközök** ideiglenes eszköztár (3.35. 1. ábra), a dokumentumba pedig bekerül egy speciális beviteli mező, ahová a képletet beilleszthetjük (3.35. 2. ábra).



1. Az **Egyenleteszközök** lap
2. Képlet beírásának helye
3. Szimbólumok beillesztésére szolgáló gombok

3.35. ábra. Az **Egyenleteszközök** lap

A képlet beírása során a betűket, műveleti jeleket a billentyűzettel, a speciális jeleket a **Szimbólum beillesztése** eszközzel írjuk az egyenletbe (3.35. 3. ábra).



3.36. ábra.

Szimbólumcsoportok



3.37. ábra.

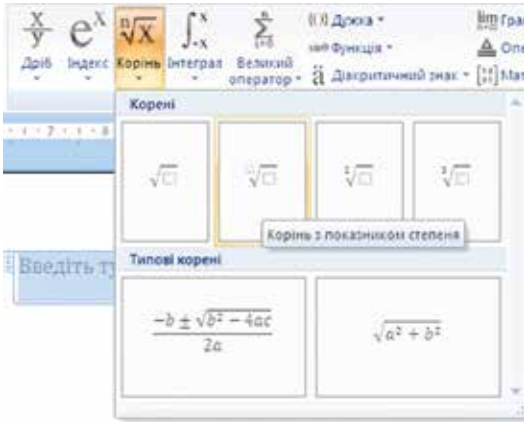
Szimbólumok listája

A beilleszthető szimbólumok nyolc csoportot alkotnak (3.36. ábra), amelyeket blokkokra bontottak (3.37. ábra). A **Menüszalag** csak azokat a szimbólumokat tartalmazza, amelyeket legutóbb használtunk. Ha másik szimbólumcsoportra van szükségünk, akkor a  $\square$  jel segítségével nyissuk meg a **Szimbólumcsoportok** listáját. A teljes listát akkor láthatjuk, ha az aktuális csoport címére kattintunk.

A listák áttekintése és a kívánt szimbólum megtalálása után kattintsunk rá, és az bekerül a kurzor aktuális helyére.

A szimbólumokon kívül rendelkezésünkre állnak különböző matematikai struktúrák (törtek, függvények, zárójelek, gyökök stb.) sablonjai is. Ezeket az **Egyenleteknél** találjuk a **Struktúrák** csoportban.

A struktúrák típusonként csoportokba vannak foglalva, amelyeket a megfelelő legördülő menükre (3.38. ábra) kattintva tekinthetünk meg. Ha a megfelelő sablonra kattintunk, az bekerül a szerkesztett kifejezésbe.



3.38. ábra. A **Struktúrák** csoport **Gyökök** sablonja



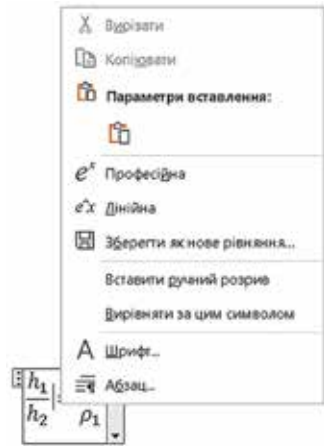
3.39. ábra. Tört sablonja

A sablont természetesen fel kell tölteni adatokkal. A beillesztés helyei szaggatott vonallal vannak kijelölve (3.39. ábra). A mezőkbe nemcsak számokat és betűket illeszthetünk, hanem sablonokat is.

A képleten belül az egérmutatóval vagy a kurzormozgató billentyűkkel mozoghatunk. A képletszerkesztőből akkor lépünk ki, ha a képlet területén kívülre kattintunk.

A képlet szerkesztése, elemek beillesztése, törlése, áthelyezése, megváltoztatása a **Wordben** megszokott módokon történik. Ezt a képlet helyi menüjének (3.40. ábra) eszközeivel tehetjük meg.

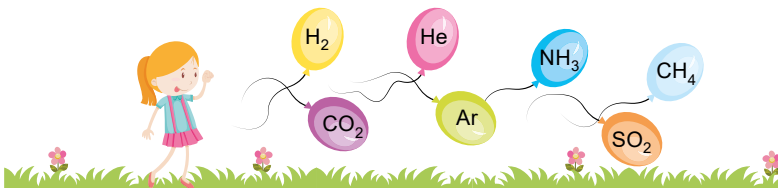
A megszerkesztett képletet a felhasználó hozzáadhatja a beépített képletekhez és később szükség szerint használhatja.



3.40. ábra.  
A képlet helyi menüje


### Feliratok a szöveges dokumentumokban

A szövegben különböző feliratokat is elhelyezhetünk, ha például a szöveget hozzáadjuk egy alakzathoz, például egy képhez a **Szövegdoboz** segítségével (3.41. ábra).

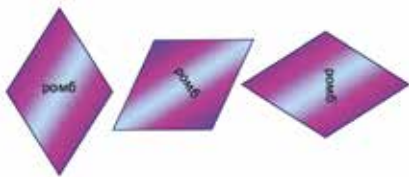


3.41. ábra. Feliratok egy képen

Többféle módon lehet feliratokat elhelyezni egy szöveges dokumentumban: elhelyezhetjük egy alakzat közepébe, vagy a **Wordnek** speciális objektumait használjuk erre a célra, **Szövegdobozt** vagy a **WordArtot**.

Ha szöveges feliratot grafikai objektumként szeretnénk elhelyezni, használjuk a **Beszúrás** menü **Ábrák** csoportjának **Szövegdoboz**  eszközt.

Az alakzatok többségének területén elhelyezhetünk szöveget is, amely megmagyarázza ennek az ábrának a jelentését. Erre az alakzat helyi menüjének **Szöveg hozzáadása** menüpontjának segítségével nyílik lehetőség.



3.42. ábra. Szöveg elhelyezése alakzatokban



3.43. ábra. Szövegyírányok az alakzatokban

*Бажаємо успіху!*


3.44. ábra. WordArt objektum

kötött módon történnek.

A szövegben **WordArt** objektumokat (3.44. ábra) is elhelyezhetünk, ezek a szöveg művészi formázását teszik lehetővé. Ezek az objektumok felhasználhatók szövegcímek, üdvözlőkártyák, plakátok, rajzok és diagrammok létrehozásához.

A **WordArt** objektumoknak kétféle – **szöveges** (betűtípus, méret, szín, stílus) és **grafikai** (méret, elhelyezkedés, kitöltés, vonalszín, árnyék stb.) – tulajdonságai vannak.

A **WordArt** beillesztése a következőképpen történik:

1. Válasszuk a **Beszúrás** ⇒ **Szöveg** ⇒ **WordArt**  gombját!
2. Válasszunk stílust!
3. Írjuk be a szöveget a **WordArt** szövegmezőjébe!
4. Kattintsunk a mezőn kívülre!

Az objektum a szövegkurzor helyére kerül majd. Az objektum formázása és szerkesztése a **Formátum** ideiglenes eszköztár segítségével történik.

### SmartArt objektumok létrehozása a szöveges dokumentumokban

A **SmartArt** objektumok (ang. *smart* – okos, *art* – művészet) a **Microsoft Office** grafikus objektumai, amelyeket arra terveztek, hogy strukturált adatokat jelenítsenek meg egy szöveges dokumentumban különböző sémák formájában. Ezek használata kifejezőbbé és világosabbá teszi a dokumentumot.



A **Word** szövegszerkesztő beépített **SmartArt** objektumok gyűjteményével rendelkezik, amelyeket több kategóriába sorolnak (3.7. táblázat).

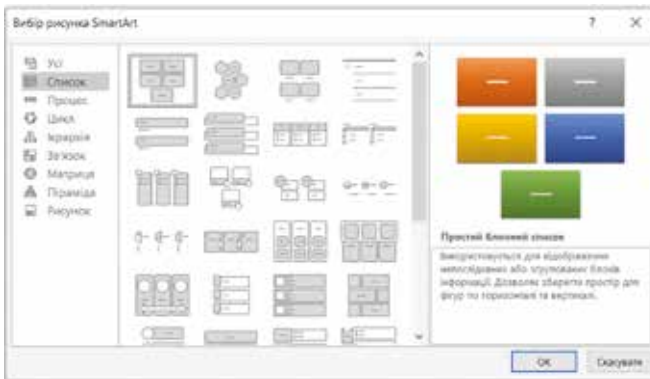
3.7. táblázat

**Különbözö kategóriájú SmartArt objektumok**

|   |  |  |  |
|---|--|--|--|
|   |  |  |  |
| <p><b>Lista</b><br/>nem meghatározott sorrend szerinti adat</p>                     | <p><b>Ciklus</b><br/>folytonos folyamat megjelenítése</p>                              | <p><b>Kapcsolatok</b><br/>kapcsolatok szemléltetése</p>  | <p><b>Piramis</b><br/>aránykapcsolatok szemléltetése úgy, hogy a legnagyobb összetevő legfelül vagy legalul jelenjen meg</p> |
|   |  |  |  |
| <p><b>Folyamatok</b><br/>lépések megjelenítése egy folyamatban való ütemtervben</p> | <p><b>Hierarchia</b><br/>hierarchikus viszonyokat jelenít meg az objektumok között</p> | <p><b>Mátrix</b><br/>annak szemléltetése, hogy az egyes részek miként viszonyulnak az egészhez</p> |  |

**SmartArt** objektum létrehozásához kövessük az alábbi lépéseket:

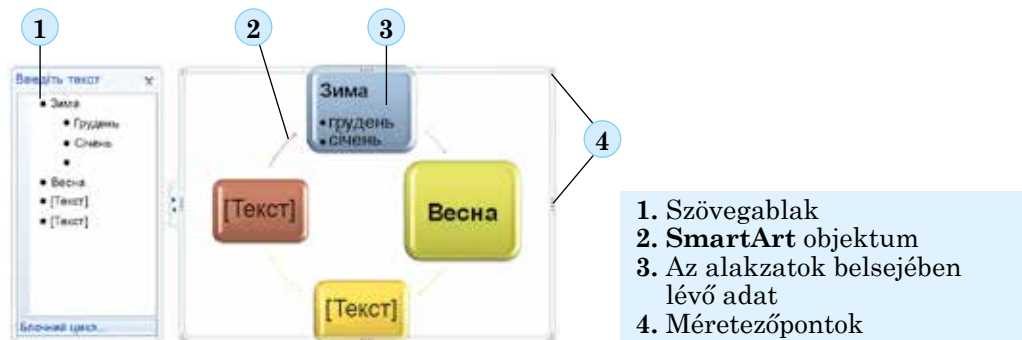
1. Válasszuk ki a dokumentumban azt a helyet, ahová az objektum beillesztésre kerül!
2. Végezzük el a **Beszúrás** ⇒ **Ábrák** ⇒ **SmartArt** műveletet, amely megnyitja a **SmartArt** listát (3.45. ábra)!
3. A baloldali listából válasszuk ki a nekünk megfelelő **SmartArt** típust!
4. Válasszuk ki a megfelelő elrendezési típust a **SmartArt** ablak közepéből!
5. Nyomjuk meg az **OK** gombot!




3.45. ábra. A **SmartArt** ablak listája

Az elrendezés kiválasztásakor figyelembe kell venni az adatok megjelenítésének módját, az elrendezések célját, amelyet az ablak jobb oldalán írnak le. Figyelembe kell venni a szöveg méretét és a típusok elemeinek számát is – nagy számuk negatívan befolyásolja megjelenítésüket és a megértésüket.

A **SmartArt** objektum beillesztése után a szövegben megjelenik annak makettje, amelyet méretezőpontokkal ellátott keret vesz körül (3.46. ábra).



3.46. ábra. **SmartArt** objektum a szövegben




A kívánt szöveget közvetlenül az alakzatokba írjuk, ha az egérmutatóval az alakzatba kattintunk. Ez a beillesztett objektumtól balra található szövegterületen is elvégezhető – a bevitt adatok automatikusan megjelennek a megfelelő ábrán. A szöveges terület elrejthető (bezárva ennek a területnek az ablakát), és a bal szélen található gomb kiválasztásával megjeleníthető .





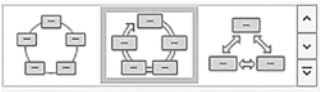






Az elrendezéstől függően a szövegterületen lévő minden listaelem a **SmartArt** objektumon külön ábraként, vagy az ábrán belüli listaelemként jelenik meg (3.45. ábra). Listaelem hozzáadása vagy törlése esetén ugyanezeket a műveleteket kell elvégezni az alakzaton is.

A **SmartArt** objektumok szerkesztése és formázása a két ideiglenes fül, a **Tervezés** és a **Formátum** vezérlőinek használatával történik, amelyek a menüszalagon megjelennek a **SmartArt** objektum kiválasztása után (3.8. táblázat).

3.8. táblázat

#### A **Tervezés** és **Formátum** ideiglenes menü vezérlőelemeinek rendeltetése

| Vezérlőelemek  | Rendeltetés   |
|--|---|
| <b>Ábra létrehozása</b> csoport a <b>Tervezés</b> fülben   |   |
| <b>Alakzat hozzáadása</b>   | További, azonos szintű alakzat hozzáadása a kijelölt alakhoz. A gombok listájában kiválaszthatjuk az alakzathoz hozzáadásának egyéb lehetőségeit. |
| <b>Listajel hozzáadása</b>  | További alacsonyabb szintű alakzat hozzáadása a kijelölt alakzathoz   |
| <b>Jobbról balra</b>        | A <b>SmartArt</b> ábra (balról jobbra, illetve jobbról balra tartó) irányának megfordítása  |



| Vezérlőelemek  | Rendeltetés   |
|--|---|
| <b>Elrendezés</b>           | A kijelölt alakzat ágelrendezésének módosítása                            |
| <b>Előléptetés</b>          | A kijelölt listajel vagy figura feljebb helyezése a hierarchiában         |
| <b>Lefokozás</b>            | A kijelölt listajel vagy figura lejjebb helyezése a hierarchiában         |
| <b>Szövegablak</b>          | Másik objektumelrendezés kiválasztása                                     |
| <b>Elrendezések csoport a Tervezés fülben</b>  |   |
|                             | Másik objektumelrendezés kiválasztása                                     |
| <b>SmartArt stílusok a Tervezés fülben</b>   |   |
| <b>Színváltoztatás</b>      | Színvariáció módosítása   |
|                             | <b>SmartArt</b> stílusok, az objektum stílusának kiválasztása             |
| <b>Visszaállítás csoport a Tervezés fülben</b>   |   |
| <b>Ábra visszaállítása</b>  | <b>SmartArt</b> ábrákon eddig végrehajtott formázási módosítások elvetése |
| <b>Alakzatok csoport a Formátum fülben</b>   |   |
| <b>Alakzatváltás</b>       | A rajz alakjának módosítása teljes formázás megőrzésével                  |
| <b>Nagyobb</b>            | A kijelölt alakzat nagyítása  |
| <b>Kisebb</b>             | A kijelölt alakzat kicsinyítése   |

### Használjuk a számítógépet!




Végezzék el a gyakorlati feladatot a <https://cutt.ly/phde6TZ> linkről, vagy a QR-kód segítségével!


### Összefoglalás


Egyszerű képletek beírása során szükség lehet a speciális szimbólumok beillesztésére. Ezt a **Beszúrás** ⇒ **Szimbólumok** ⇒ **Szimbólum** ⇒ **Egyéb szimbólumok** paranccsal tehetjük meg. Továbbá használhatjuk még a **Kezdőlap** ⇒ **Betűtípus** csoportjának **Alsó index**  és **Felső index**  eszközét.


Az összetett, többszintű képletek beillesztése a **Képletszerkesztő** segítségével történik. A beillesztett képletek szimbólumokból és speciális sablonokból kialakított struktúrákból állnak, amelyeket ki kell tölteni tartalommal.

A szimbólumokon kívül rendelkezésünkre állnak különböző matematikai struktúrák: törték, függvények, zárójelek, gyökök stb. sablonjai is. Ezeket az **Egyenleteknél** találjuk a **Struktúrák** csoportban. A struktúrák típusonként csoportokba vannak foglalva, amelyeket a megfelelő legördülő menükre kattintva tekinthetünk meg. Ha a megfelelő sablonra kattintunk, az bekerül a szerkesztett kifejezésbe.

A szövegben különböző feliratokat is elhelyezhetünk, ha például a szöveget hozzáadjuk egy alakzathoz, például egy képhez a **Szövegdoboz** segítségével. Többféle módon lehet feliratokat elhelyezni egy szöveges dokumentumban: elhelyezhetjük egy alakzat közepébe, vagy a **Wordnek** speciális objektumait használjuk erre a célra, **Szövegdobozt** vagy a **WordArtot** .

Szöveg hozzáadásához az alakzathoz válasszuk az alakzat helyi menüjében a **Szöveg hozzáadása** parancsot. Ha szöveges feliratot grafikai objektumként szeretnénk elhelyezni, használjuk a **Beszúrás** menü **Ábrák** csoportjának **Szövegdoboz**  eszközét!

A szövegben **WordArt** objektumokat is elhelyezhetünk, ezek a szöveg művészi formázását teszik lehetővé. A **WordArt** objektumoknak kétféle – *szöveges* (betűtípus, méret, szín, stílus) és *grafikai* (méret, elhelyezkedés, kitöltés, vonalszín, árnyék stb.) – tulajdonságai vannak. A **WordArt** beillesztéséhez válasszuk a **Beszúrás** ⇒ **Szöveg** ⇒ **WordArt**  lehetőséget!

A **SmartArt** objektumok a **Microsoft Office** grafikus objektumai, amelyeket arra terveztek, hogy strukturált adatokat jelenítsenek meg egy szöveges dokumentumban különböző sémák formájában. A **Word** szövegszerkesztő beépített **SmartArt** objektumok gyűjteményével rendelkezik, amelyeket több kategóriába sorolnak. **SmartArt** objektum létrehozásához végezzük el a **Beszúrás** ⇒ **Ábrák** ⇒ **SmartArt**  műveletet, amely megnyitja a **SmartArt** listát, ahol ki lehet választani a nekünk tetsző típust.



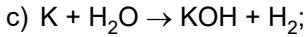
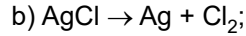
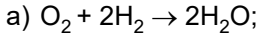
### Felelj a kérdésekre!

- 1°. Milyen matematikai szimbólumokat használhatunk a képletekben? Hogy illeszthetjük be ezeket?
- 2°. Milyen struktúrákat tartalmazhat egy képlet? Nevezd meg néhányat ezek közül!
- 3°. Milyen szerkesztési és formázási műveleteket végezhetünk a képlettel? Hogyan végezzük ezeket?
- 4°. Hogyan készíthetünk feliratot egy alakzat közepébe? Milyen szerkesztési és formázási műveleteket végezhetünk el rajta?
- 5°. Mire használják a **WordArt** objektumokat? Hogyan lehet elkészíteni?
- 6°. Milyen típusú objektumokhoz tartoznak a **WordArt** objektumok? Milyen tulajdonságokkal rendelkeznek?
- 7°. Milyen típusú **SmartArt** objektumokat ismertek és mi a rendeltetésük?
- 8°. Milyen tulajdonságokkal rendelkeznek a **SmartArt** ábrák? Hogyan módosíthatók?
- 9°. Milyen szerkesztési és formázási műveletek végezhetők el a **SmartArt** ábrákon?
- 10\*. Milyen grafikus objektumok hozhatók létre egy szöveges dokumentumban? Miben különböznek? Mi bennük a közös?



### Végezd el a feladatokat!

1•. Hozz létre egy szöveges dokumentumot! Írd be az alábbi kémiai képleteket, majd mentsd a fájlt **feladat 3.5.1.docx** néven a mappádba!



2•. Hozz létre egy szöveges dokumentumot! Írd be az alábbi fizikai képleteket, majd mentsd a fájlt **feladat 3.5.2.docx** néven a mappádba!

a) a forgás periódusa  $T = \frac{t}{N}$ ;

b) a közlekedőedények egyenlete  $\frac{h_1}{h_2} = \frac{\rho_2}{\rho_1}$ ;

c) a kinetikus energia  $E_k = \frac{mv^2}{2}$ ;

d) a hatásfok  $\eta = \frac{A_1}{A_2} \cdot 100 \%$ .



3•. Hozz létre egy szöveges dokumentumot! Írd be az alábbi matematikai képleteket, majd mentsd a fájlt **feladat 3.5.3.docx** néven a mappádba!

a)  $\frac{3}{4}(12 - x) = \frac{3}{8}$ ;

b)  $\frac{7}{8}(2 - x) = \frac{1}{8}$ ;

c)  $\frac{2}{3}(x - 4) + \frac{1}{3} = 5$ ;

d)  $\frac{3}{7}(1 - 2x) + \frac{1}{7} = -2$ .

4\*. Hozz létre egy szöveges dokumentumot! Írd be az alábbi matematikai képleteket, majd mentsd a fájlt **feladat 3.5.4.docx** néven a mappádba!

a)  $y = \begin{cases} x^2 + 3, & \text{ha } x \leq 0, \\ \frac{x}{2} + 3, & \text{ha } x > 0; \end{cases}$

b)  $y = \begin{cases} -3x^3 - 2, & \text{ha } x < 0, \\ \frac{7x}{4} - 2, & \text{ha } x \geq 0; \end{cases}$

c)  $y = \begin{cases} 3x - y = 1, \\ x + y = 3, \\ 4y = 8; \end{cases}$

d)  $y = \begin{cases} x - y = 1, \\ x + y = 3, \\ 2x = 6. \end{cases}$



5\*. Hozz létre egy szöveges dokumentumot! Írd be az alábbi matematikai képleteket, majd mentsd a fájlt **feladat 3.5.5.docx** néven a mappádba!

a)  $\frac{5a^4 - 20b^2}{2a^2 + b^2} : \frac{30(a - 4b)^2}{9a^4 - b^4}$ ;

b)  $\frac{x}{7a^2b^3} : \frac{6a^3c^2}{b} = \frac{y}{4c}$ ;

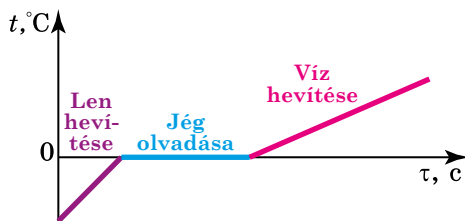
c)  $\frac{a + \frac{25}{a+10}}{\frac{25}{a} - a}$ ;

d)  $1 - \frac{1}{1 - \frac{1}{1 - \frac{1}{a+1}}}$ .



6\*. Hozz létre születésnapi üdvözlőkártyát egy barátodnak **WordArt** objektumok, feliratok és képek segítségével! Mentse a dokumentumot **feladat 3.5.6.docx** néven a mappádba! Cseréljétek üdvözlőkártyát a barátoddal! A barátod kártyájának melyik része tetszett a legjobban? Magyarázd el, hogyan lett ez elkészítve!

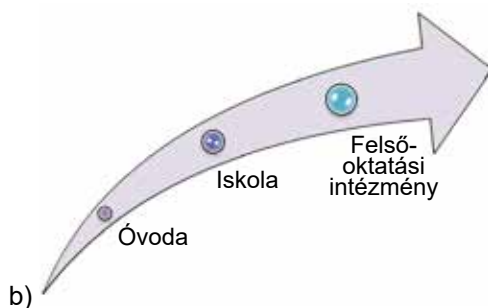
- 7\*. Készítsd el a következő képet egy szöveges dokumentumban ábrák és feliratok felhasználásával! Mentsd a dokumentumot **feladat 3.5.7.docx** néven a mappába!



- 8\*. Hozz létre **SmartArt** ábrákat egy szöveges dokumentumban a következő min-ták szerint! Mentsd a dokumentumot **feladat 3.5.8.docx** néven a mappába! Kérdezd meg családtagjaidat, hogyan használnák az ilyen ábrákat munkájuk során!



a)



b)

- 9\*. Hozz létre egy **SmartArt** ábrát, amely a képletek elkészítésének algoritmusát mutatja be a **Képletszerkesztő** felhasználásával! Mentsd a dokumentumot **feladat 3.5.9.docx** néven a mappába!
- 10\*. Készítsd el a szövegszerkesztőben a négyszögek csoportosítását a **SmartArt** szervezeti diagram formájában! Mentsd a dokumentumot **feladat 3.5.10.docx** néven a mappába!



- 11\*. Alkossatok három fős csoportokat! Készítsetek a szövegszerkesztőben **SmartArt** sugárdiagrammot, amely bemutatja, hogy hol helyezkedtek el a kö-zások szicsei! Mentsétek a dokumentumot **feladat 3.5.11.docx** néven a mappátokba! Hogyan osztottátok fel a munkát egymás között? Miért éppen így? Hogyan segítettetek egymásnak?
- 12\*. Alkossatok párokat! Ismerkedjete meg az internet segítségével az infografi-kával és tudjátok meg, hogy mire szolgál! Készítsetek utasításokat a kedvenc játékokhoz bármilyen szövegszerkesztő segítségével! Beszéljétek meg a kedvenc játékokhoz készített utasítások végrehajtását! Válasszátok ki azt, amelyik a legjobban tetszett! Végezzétek el a feladatot egy új szöveges doku-mentumban! Mentsétek a dokumentumot **feladat 3.5.12.docx** néven a mappátokba! Miért épp ezt a játékot választottátok? Miért ezt a végrehajtási módot? Milyen nehézségekbe ütköztetek? Hogyan sikerült ezt leküzdeni?



# 4. fejezet

## Webes felületek létrehozása és közzététele

*Ebben a fejezetben megismerkedtek:*

- a **HTML** nyelv fogalmával
- a hiperhivatkozások, grafikai és multimédiás elemekkel weboldalas használatával
- a weboldalak ergonomikus elkészítésének szabályaival
- a weboldalak készítésének és szerkesztésének lépéseivel
- azzal hogyan lehet weblapokat készíteni és különböző objektumokat elhelyezni rajtuk
- azzal hogyan kell weboldalakat közzétenni

### 4.1. A weboldalak felépítése. A HTML nyelv

1. Mi az a *weboldal*? Mi az a *weblap*?
2. Milyen objektumokat tartalmazhat egy weboldal?
3. Mi a *dizájn*? A prezentáció dizájnos elkészítésének milyen elveit ismeritek?

#### *A weboldalak szerkezeti elemei*

A weboldalak információforrások a **Word Wide Webes** világhálón. Már tudjátok, hogyan lehet információkat keresni az interneten, hogyan lehet weboldalakat böngészni. Most megismerkedünk a weboldalak elkészítésével.

A weboldal felépítésében a következő összetevőket különböztetjük meg (4.1.ábra):

- **tartalom** – (ang. *content* – tartalom) – a weboldal tartalmi része, erre koncentrálódik a felhasználó figyelme; szöveget, grafikát, videót, hangfájlokat tartalmazhat;
- **navigációs elemek** – lehetővé teszi, hogy átléphessünk a webhely más lapjaira vagy más weblapokra;
- **tervezés (dizájn)** – a weboldal kialakítását, látványát, tartalmának struktúráltságát biztosítják.

A weboldalakon közzétett tartalmi résznek érdekesnek, hasznosnak és lehetőleg egyedinek, szerzői joggal védettnek kell lennie. Először is, a webhely sikere az anyagok minőségétől függ.

A weboldalak és a számítógépes prezentációk tervezésénél figyelembe kell venni a kolorisztika és az ergonómia szabályait.

Most pedig részletesebben foglalkozunk az ergonómia szabályaival.



1. Navigációs elemek

2. Tervezés (dizájn)

3. Tartalom

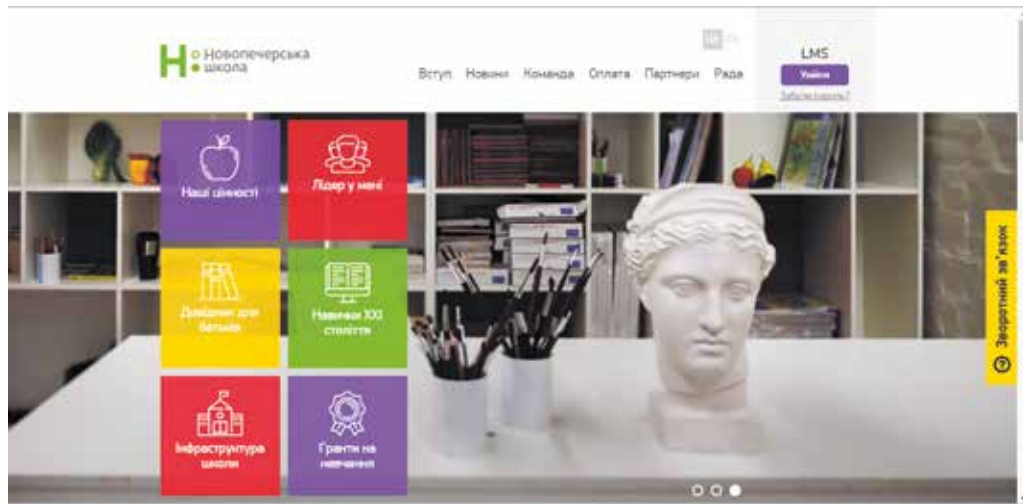
4.1. ábra. A weboldal felépítése

### Az objektumok weboldalakon történő elhelyezésének ergonómiai szabályai

Az **ergonómia** (a görög. έργος – munka, νόμος – törvény szavakból) – az a tudomány, amely a különböző eszközök, készülékek, szerszámok alkalmazásának hatásfokát tanulmányozza, valamint azt, hogyan hat ezek használata az ember tevékenységére, termelékenységére.

Egy **weboldal** akkor **ergonómikus**, ha érzékelése kényelmes, vezérlőelemei jól áttekinthetők és kényelmesen használhatók, a felhasználó jól követhetően tekintheti meg a weboldalt.

A weboldal ergonómiájának szerves része a kényelmes és egyszerű kezelhetőség, a felhasználó speciális képzettsége nélkül. Mindenkinek képesnek kell lennie arra, hogy intuitív módon kapcsolja össze a weboldalon végrehajtandó műveleteket a látott objektumokkal (4.2. ábra).



4.2. ábra. Ergonómikus weboldal példája



A weboldalak szerkezetének megtervezésekor az alábbi ajánlásokat kell figyelembe venni:

- a webhely nevét az összes oldal címében kell elhelyezni;
- minden oldalon jelenítsük meg, hogy melyik tematikus részhez tartozik az oldal;
- ajánljunk több linket ugyanarra az oldalra lépéshez, hogy minden felhasználó kiválaszthassa a legkényelmesebbet vagy legkézenfekvőbbet;
- készítsünk olyan lehetőségeket, hogy bármelyik oldalról a webhely fő (kezdőlap) oldalára ugorhassunk;
- el kell távolítani minden kétértelműségre utaló jelet, amikor kiválasztunk egy vezérlőelemet egy weboldalon;
- a láblécben fel kell tüntetni a weboldal tervezőjének és tulajdonosának szerzői jogait (copyright) és elérhetőségeit.

Ügyeljünk az objektumok ergonomikus elhelyezésére a weboldalakon. E célból kívánatos a következő szabályok betartása:

- a legfontosabb adatokat az oldal tetején helyezzük el;
- a cikkek címeit nagyított betűmérettel készítsük;
- legfeljebb három oszlopnyi tartalmat helyezzünk el az oldalon;
- ne engedjük a vízszintes gördítősávok megjelenését a weboldalakon;
- vizuálisan emeljük ki a hipervivatkozásként használt szövegrészt vagy grafikai elemet;
- a nagy mennyiségű szöveget tartalmazó oldalakon hozzunk létre hipervivatkozásokat, amelyek az oldal közepére mutatnak.

Weboldal tervezésénél próbáljunk meg kényelmet nyújtani a speciális igényű felhasználók számára. Ehhez ügyeljünk a következőkre:

- a betűméretnek olyannak kell lennie, hogy a legtöbb felhasználó olvashassa anélkül, hogy megerőltetné a szemét vagy nagyítaná az oldalakat; látássérült emberek számára kívánatos, hogy változtatni lehessen a betűméreten;
- kívánatos, hogy a képeken legyenek feliratok, valamint ne legyenek apró részletek; lehetővé kell tenni a kép nagyítását megtekintés céljából;
- ha lehetséges, adjunk a weboldalakhoz olyan elemeket, amelyek fontos információkat közölnek a webhelyen.

## A HTML nyelv fogalma

A weboldalakat szöveges dokumentumokként hozzák létre, és általában a **html** kiterjesztésű fájlokban tárolják. A **HTML** nyelv (ang. *Hyper Text Markup Language* – *hiperszöveges jelölőnyelv*) a weboldalak létrehozására és fejlesztésére szolgál. Az ezen a nyelven írt speciális formázó utasításokat a böngésző értelmezi és végrehajtja, objektumokat helyez el a weboldalon a fejlesztő terve szerint. Az oldal HTML-kódja kétféle elemből áll:

- **szövegből**, amely megjelenik az oldalon;
- **TAG**ből (ang. *tag* – címke) – a html dokumentum formázó utasításaiából, amelyek meghatározzák a szöveg kinézetét: felépítését, formátumát és a szövegrészek elhelyezkedését.

A *tageket* minden esetben a „kisebb” és „nagyobb” jelek közé kell tenni **< tag >**. Például, a **<hr>** – vízszintes elválasztó vonal.

A tag nyitó részébe az azonosító után megadhatunk ún. paramétereket is. Ezek a paraméterek a tagre vonatkozó tulajdonságokat állítják be. Például, ha képet szeretnénk beszúrni egy oldalra, akkor a paraméterértékek határozzák meg, hogy honnan vesszük a képet és milyen méretben kell megjeleníteni az oldalon.

A paraméter neve és az egyenlőségjel után a paraméter értékét dupla idézőjelek közé kell tenni. A paramétereket egy szóközzel választjuk el egymástól.

Például ha be szeretnénk illeszteni az **1.jpg** nevű képfájlt az oldalunkra, amely ugyanabban a mappában található, mint a weblapfájl, valamint, ha be szeretnénk állítani hozzá a kép 200 képpont szélességét és magasságát, akkor használjuk az IMG taget (ang. *image* – kép) a következő paraméterek értékeivel:

- **src** (ang. *source* – forrás) – a képfájl nevének megadása;
- **width** (ang. *width* – szélesség), **height** (ang. *height* – magasság) – a kép szélességének és magasságának beállítása.

A tag a következőképpen fog kinézni:

```
<img src = "1.jpg" width = "200" height = "200">
```

Annak a képnek a tage, amely az interneten a következő URL-címen helyezkedik el: [http://vsviti.com.ua/wp-content/uploads/2017/02/arheol\\_01.jpg](http://vsviti.com.ua/wp-content/uploads/2017/02/arheol_01.jpg), valamint a *szélessége* 200 pixel, a *magassága* pedig 100 pixel, a következőképpen néz ki:

```
<img src = "http://vsviti.com.ua/wp-content/uploads/2017/02/arheol_01.jpg"
width = "200" height = "100">
```

Léteznek olyan tagek, amelyek a szövegrész bizonyos részeit formázzák csak meg. Például egy mondatban csak egy szót kell kiemelni dőlt betűvel. Ebben az esetben páros tageket használnak (**a tag nyitó és záró része**). Minden tagnek van nyitó és záró része. A záró részbe az azonosító elé egy „/” jelet kell tenni. A tag nyitó része tartalmazza a paramétereket. Ezek a paraméterek a tagre vonatkozó tulajdonságokat állítják be. Például, ha a **HTML – hipertext jelölőnyelv** mondatban csak a *HTML* részt kell dőlt betűssé tenni, akkor azt így írhatjuk fel:

```
<i>HTML</i> – hipertext jelölőnyelv.
```

A **<i>** – nyitó és a **</i>** záró tag közé írt szöveg lesz dőlt betűs.

Ha az *Ukraina tradíciói* szavakat piros színnel szeretnénk leírni nagyított betűmérettel, akkor használjuk a **<font>** (ang. *font* – típus) taget a **color** és **size** (ang. *color* – szín, és a *size* – méret) paraméterekkel:

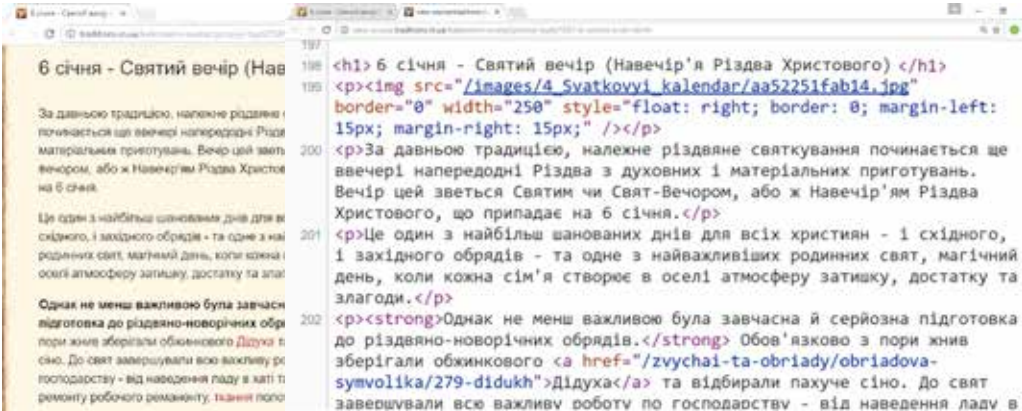
```
<font color = "red" size = "5"> Ukraina Tradíciói </font>.
```

A böngészőablakban megnyitott weboldal HTML-kód úgy tekinthető meg, hogy az oldal helyi menüjéből kiválasztjuk a **Forráskód megtekintése elemet**, vagy a **Ctrl+U** billentyűkombinációt. A 4.3. ábra a weboldal egy részének képét és a megfelelő HTML kódrészletet mutatja. Itt a címkéket, a paramétereket és a paraméterértékeket általában színessel emelik ki.

A 4.3 ábrán látható kódrészletben vannak olyan tagek, amelyek meghatározzák az oldal felépítését:

- **<h1> ... </h1>** – a weblap címét tartalmazó tag. A példában létrehozott cím szövege: *6 січня – Святий вечір (Навечір'я Різдва Христового)*;
- **<img>** – kép beillesztésének a tagje, amely tartalmazza a kép paramétereinek értékeit. A példában az *52251fab14.jpg* nevű kép a következő helyről lett beillesztve: */images/4\_Svatkovyi\_kalendar/aa52251fab14.jpg*;
- **<p> ... </p>** – bekezdések készítésére szolgáló tag. Az első bekezdésben a következő szöveg szerepel: *За давньою традицією, належне різдвяне святкування починається ще ввечері напередодні Різдва з духовних і матеріальних приготувань. Вечір цей зветься Святим чи Святвечором, або ж Навечір'ям Різдва Христового, що припадає на 6 січня.* Az oldal természetesen több bekezdést is tartalmaz;





4.3. ábra. Egy weboldal HTML-kódjának részlete

- **<strong> ... </strong>** – félkövér betűvel kiemelt szövegrész tagje. A fenti példában a következő szövegrész félkövérrel írott: *Однак не менш важливою була завчасна й серйозна підготовка до різдвяно-новорічних обрядів;*
- **<a> ... </a>** – hiperlink beillesztésének a tagje. A fenti példában a hiperlink a *Дідуха* szóhoz van kötve és hozzáférést biztosít ugyanazon webhely weboldalához </zvychai-ta-obriady/obriadova-symvolika/279-didukh> címmel.



**Használjuk a számítógépet!**

Végezzétek el a gyakorlati feladatot a <https://cutt.ly/chdrqBa> linkről, vagy a QR-kód segítségével!



**Összefoglalás**

A weboldal felépítésében a következő összetevőket különböztetjük meg: tartalom, navigációs elemek és tervezés (dizájn).

Egy **weboldal** akkor **ergonomikus**, ha érzékelése kényelmes, vezérlőelemei jól áttekinthetők és kényelmesen használhatók, a felhasználó számára jól követhető a weboldal. Ügyelni kell az objektumok ergonomikus elhelyezésére a weboldalakon.

A weboldalakat szöveges dokumentumokként hozzák létre, és általában a **html** kiterjesztésű fájlokban tárolják. A **HTML** nyelv a weboldalak létrehozására és fejlesztésére szolgál. Az oldal **HTML-kódja** kétféle adatból áll: **szövegből**, amely megjelenik az oldalon és **TAGból** – a html dokumentum formázó utasításából, amelyek meghatározzák a szöveg kinézetét.

A böngészőablakban megnyitott weboldal HTML-kódját úgy tekintheti meg, hogy az oldal helyi menüjéből kiválasztjuk a **Forráskód megtekintése** elemet, vagy a **Ctrl+U** billentyűkombinációt.



**Felelj a kérdésekre!**

1. Milyen szerkezeti elemei vannak egy weboldalnak? Jellemezd minden csoportját!
2. Milyen weboldalt nevezhetünk ergonomikusnak?
3. Milyen szabályokat kell betartanunk ahhoz, hogy egy weboldal ergonomikus legyen?
4. Hogyan alakíthatjuk ki úgy a webhelyet, hogy az kényelmes legyen a különleges igényű felhasználók számára?
5. Mi az a HTML-kód? Milyen típusú adatokat tartalmaz?

- 6°. Hogyan használhatjuk a tageket; paraméteres tageket; páros tageket?
- 7°. Melyik TAG szolgál a képek beillesztésére? Melyik a hiperhivatkozás létrehozására?
- 8°. Hogyan tekinthetjük meg a weboldal HTML-kódját?



### Végezd el a feladatokat!



- 1°. Nézd meg az **Ukrán Nemzeti Gyermekkönyvtár** weboldalát ([chl.kiev.ua](http://chl.kiev.ua))! Határozd meg, majd írd be a füzetébe, hogy hány navigációs elemet helyeztek el az oldalon, milyen típusú a tartalma, milyen tervezési (dizájn) elemeket használtak?
- 2°. Tekintsd meg a **Зелена хвиля (Zöld Hullám)** ([ecoclubua.com](http://ecoclubua.com)) weboldalt! Írd be a füzetedbe, hogy a weboldal mely elemei felelnek meg az ergonómia szabályainak és melyek nem!
- 3°. Keresd meg **7 чудес України (Ukrajna 7 csodája)** ([7chudes.in.ua](http://7chudes.in.ua)) weboldalt! Írd be a füzetbe, hogy a weboldal mely elemei felelnek meg az ergonómia szabályainak és melyek nem!
- 4\*. Tekintsd meg a [kievoi.ippo.kubg.edu.ua](http://kievoi.ippo.kubg.edu.ua) weboldal HTML-kódját! Keress a kódban képek, hiperhivatkozások és vízszintes választóvonalak beillesztésére szolgáló TAGeket! Vonj le következtetést a kis- és nagybetűk használatával kapcsolatban a HTML-kódban! Próbáld meg kitalálni, mire szolgálhatnak a következő tagek és paramétereik:
  - a) a `<hr>` tag és a hozzá tartozó **color** paraméter;
  - b) az `<a>` tag és a hozzá tartozó **href** paraméter lehetséges értékei!
- 5°. Nyisd meg a **4fejezet\4.1rész\4.1.5feladat.txt** fájlt! MentSD újra ezt a fájlt html kiterjesztéssel és **UTF-8**-as kódolással a saját mappádba! Írj be szöveget és címeket a HTML-kódba tagekkel, hozz létre egy olyan részt az oldalon, amelyben szerepel a város neve, egy vízszintes vonal és egy link a lakhelyed weboldaláról! MentSD el a módosított fájlt, és nézd meg annak tartalmát egy böngészőablakban! Ellenőrizd, helyes-e a hiperhivatkozás, amit megadtál!
- 6\*. Nyisd meg a **4fejezet\4.1rész\4.1.6feladat.txt** fájlt! MentSD újra ezt a fájlt html kiterjesztéssel és **UTF-8**-as kódolással a saját mappádba! Tekintsd meg a mentett fájl tartalmát egy böngészőablakban! Módosítsd a HTML-kódot úgy, hogy:
  - a) a vezetékneved megjelenjen a böngésző címében;
  - b) az első szint címsorában szerepeljen a vezetékneved és keresztnéved;
  - c) a második szint címsora az oktatási intézményed nevét tartalmazza;
  - d) a vízszintes vonal vörös legyen;
  - e) a kép mérete 200 × 200 pixeles legyen;
  - f) a hiperhivatkozás a **Wikipedia** főoldalára mutasson!MentSD el a módosított fájlt és nézd meg annak tartalmát egy böngészőablakban! Ellenőrizd, hogy helyes-e a hiperhivatkozás, amit megadtál!

## 4.2. A webfejlesztés automatizálásának eszközei



1. Milyen objektumokat lehet kiemelni a weboldalon?
2. Milyen weboldalt tekinthetünk ergonomikusnak?
3. Mi az a weboldal HTML-kódja? Milyen típusú adatokat tartalmaz?





## A webfejlesztés eszközei

A HTML nyelv használatával egyedi kialakítású, különféle szolgáltatásokkal és képességekkel rendelkező webhelyet hozhatunk létre a fejlesztő összes kreatív ötletének megvalósításához.

Egyszerűbb webhely létrehozásához használhatjuk a webfejlesztés automatizált létrehozásának és közzétételének eszközeit.

Léteznek **speciális webszerkesztők**, amelyek rendeltetése a weboldalak létrehozása. Ilyenek az **Adobe Dreamweaver**, **Microsoft FrontPage**, **WYSIWYG Web Builder**... stb. Ezek a szoftverek speciális eszköztárakkal rendelkeznek statikus és dinamikus oldalak fejlesztéséhez. A fejlesztő objektumokat helyez el a weboldalakon – képeket, táblázatokat, listákat stb. Az egyes weboldalak HTML-kódját a kapott objektumkészlet alapján állítja elő. Ezeket a programokat szokás **WYSIWYG-szerkesztőknek** (ang. *What You See Is What You Get* – amit láatsz, azt kapsz) is nevezni.

A weboldalak fejlesztésének másik eszköze a **tartalomvezérlő rendszerek WCMS-ek** (ang. *Web Content Management System* – webes tartalomkezelő rendszer) – ezek kényelmes eszközöket biztosítanak a felhasználók számára, amelyekkel kidolgozhatják a weblap külalakját, feltölthetik azt tartalommal, kialakíthatják a navigációs rendszereket stb. Ezek közül a rendszerek közül a **Joomla**, **WordPress**, **MediaWiki** az ismertebbek. A **WCMS rendszerek** sablonok, és beépülő modulok alapján készítik a weblapokat, segítségükkel könnyen adhatunk a weblapokhoz fórumot, csevegőt, szalagokat, statisztikai eszközöket, egyéb dinamikai elemeket.

Az ezeken a rendszereken létrehozott oldalak tartalmát adatbázisban tárolják, és a HTML-kód automatikusan generálódik egy sablon alapján.

Az imént említett alkalmazásokat telepíthetjük a saját számítógépünkre. A segítségével kivitelezett weblapokat létrehozásuk után közzé kell tenni az interneten.

Ugyanakkor léteznek olyan WCMS rendszerek, amelyek ingyenes hosztिंग szolgáltatásokat is biztosítanak. Ilyenek az **Google Sites**, **Weebly**, **uCoz** stb. Amennyiben ilyen rendszer alkalmazása mellett döntünk, a weblapot közvetlenül a szerveren hozhatjuk létre online módban, és a fejlesztő nem tudja közvetlenül szerkeszteni a weboldalak HTML-kódját.

## Automatikus webfejlesztés webhelytervező rendszer segítségével

Az online webfejlesztés és tartalomvezérlés alkalmával a következő lépéseket kell megtennünk:

1. Regisztrálnunk kell a webhelytervező szerveren.
2. Ki kell választani a weblap nevét és az alkalmazandó sablont.
3. Létre kell hozni a weboldalakot és a navigációs elemeket.
4. Fel kell tölteni az oldalakat tartalommal.
5. Közzé kell tenni az oldalt.

Vizsgáljuk meg ezeket a lépéseket részletesebben a **Google Sites** ingyenes szolgáltatás példáján.

Ahhoz, hogy használni tudjuk ezt a szolgáltatást, **Google**-felhasználónak kell lennünk. Ti már használtátok a **Google** szolgáltatásait, amikor e-maileztetek és közösen dolgoztatok egy dokumentumon?

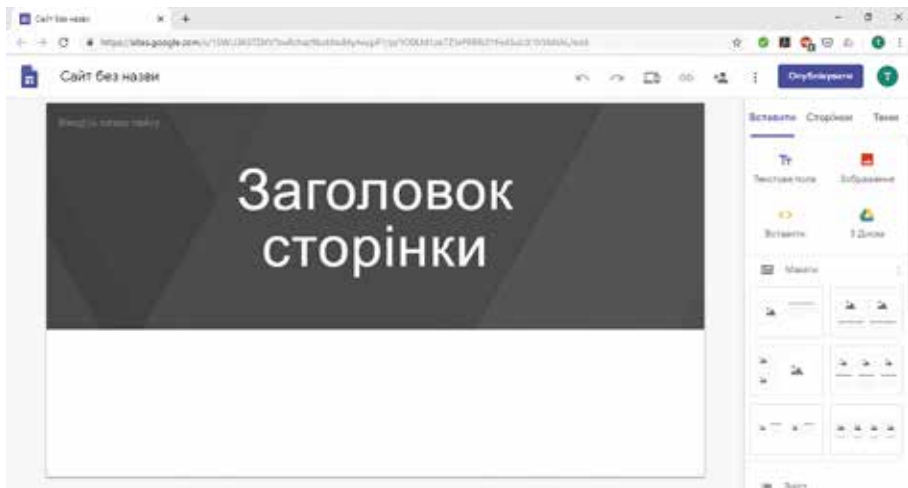
Webhelyet létrehozni különbözőképpen lehet:

1. Nyissuk meg a **Google Drive**-ot, a **Google** fiók használatával, majd pedig választunk az **Új** ⇒ **Továbbiak** ⇒ **Google webhely** lehetőségeket!

2. Jelentkezzünk be a **Google**-fiókba és nyissuk meg a [sites.google.com](https://sites.google.com) weboldalt!

Válasszuk az **Új webhely létrehozása**  gombot a **weboldal** jobb alsó sarkában!

Mindezek után a böngésző ablakában megjelenik a létrehozott honlap szerkesztő módban (4.4. ábra).



4.4. *ábra.* A frissen létrehozott honlap szerkesztő módban



4.5. *ábra.* Cím nélküli webhely vázlatképe

*Cím nélküli webhely* miniatűr vázlata megjelenik mind a **Google Drive**-on, mind pedig a **Webhelyek** ablakban (4.5. ábra). Ha duplán kattintunk erre a vázlatképre, szerkesztési módban nyílik meg a webhely a további munkához.

Az ablak jobb oldalán, a webhely fő oldala mellett van egy oldalsáv a **Beszűrés**, **Oldalak**, **Témák** fülekkel. A panel eszközei oldalak hozzáadására, törlésére és szerkesztésére, objektumok beszűrésére szolgálnak.

### ***A Webhely címének és témájának (stílusának) a kiválasztása***

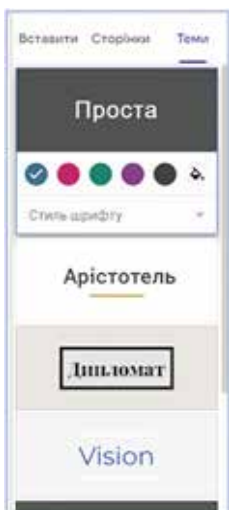
Minden webhelynek van egy neve, amely tükrözi annak tartalmát vagy szolgáltatását. Az oldal tetején található **Az oldal címe** mezőbe kerül az új webhely címe. Ugyanakkor megváltozik a webhely vázlatképe a **Google Drive**-on.

A **Google**-dokumentumokhoz hasonlóan a webhely szerkesztése során végrehajtott módosításokat automatikusan menti a rendszer.

A webhely kinézetének módosításához:

1. Kattintsunk a jobb oldali menü **Témák** pontjára!
2. Válasszunk egy új témát (4.6. ábra)!
3. Válasszunk színt és betűtípust a kiválasztott témához a javasoltak közül!

A webhely oldalain megváltoztathatjuk a cím megjelenését is. Amikor a cím fölé visszük az egérmutatót, megjelenik egy parancssáv **Kép módosítása** és **Fejléc típusa** lehetőségekkel. Az első lehetőség kiválasztásával megnyílik az oldal címében használható képek listája. A második parancs kiválasztásakor a kö-



4.6. *ábra.*  
A **Google**  
webhelyek témái



vetkezőkre változtathatjuk meg a cím típusát: **Borító, Nagy szalaghirdetés, Szalaghirdetés, Csak cím.**

A weboldalon **lábléceket** is elhelyezhetünk. Ehhez vigyük az egérmutatót az oldal aljára, és válasszuk a **Lábléc szerkesztése** gombot! A megjelenő mezőbe beírhatjuk a webhelyfejlesztő adatait, létrehozásának vagy frissítésének dátumát stb.

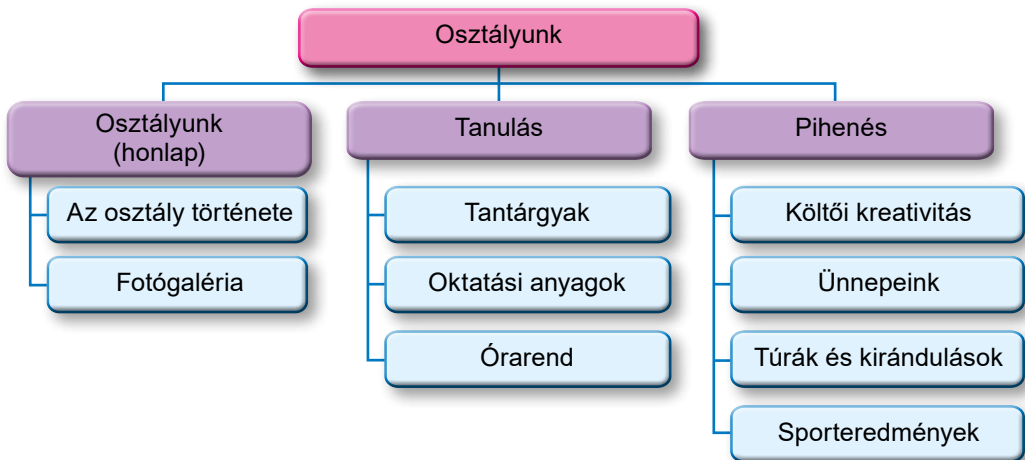
A fejlécek és láblécek automatikusan megismétlődnek a webhely minden oldalán, valamint a fejlécben lévő címsor lesz azonos az oldal címével.

### Webhelyek létrehozása

A frissen létrehozott webhely csak egyetlen oldalt **Főoldalt (Honlapot)** tartalmaz. A webhely minden oldalának megvan a saját címe. A főoldal címét a **Webhelydokumentum neve** mezőbe kell beírni. Gyakran a főoldal neve egybeesik a webhely nevével.

Általában egy webhely több oldalból áll, amelyek nevéből az úgynevezett **webhelytérkép** képződik. Ez egy diagram, amely megmutatja az oldalak hierarchiáját és a közöttük lévő kapcsolatokat. A legfelső szintű webhelytérkép azokat az oldalakat tartalmazza, amelyek a tematikus fejezetekben a legfontosabbak, alattuk helyezkednek el azok az oldalak, amelyek a megfelelő fejezetekhez tartoznak. Például a 4.7. ábra az *Osztályunk* webhelytérképét mutatja.

**Hierarchia** (a görög *ἱεραρχία* – szent, *ἀρχή* – uralom szavakból) – fölé- és alárendeltségek rendszere, amiben az emberek vagy dolgok valamilyen szempont alapján vannak elrendezve.

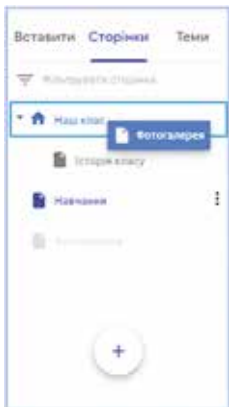


4.7.ábra. Az *Osztályunk* webhelytérképe

A fenti webhelytérképen három tematikus fejezet található – *Osztályunk*, *Tanulás*, *Pihenés*. Az *Osztályunk* tematikus fejezetéhez tartozik az *Osztályunk* – a tematikus fejezet főoldala, valamint *Az osztály története* és *Fotógalleria* oldalak is.

A webhelyen új oldalt a következőképpen készíthetünk:

1. Válasszuk ki az **Oldalak** fület az ablak jobb oldalán található oldalsávon!
2. Vezessük az egérmutatót a **+** gombra és válasszuk az **Új oldal** lehetőséget!
3. Adjunk nevet az új oldalnak a **Név** mezőbe!
4. Nyomjuk meg a **Kész** gombot!



Új oldalak létrehozásakor automatikusan létrejön egy navigációs sáv, amely alapértelmezés szerint minden oldal címének tetején, a jobb oldalon jelenik meg. Az oldalak sorrendjét úgy módosíthatjuk, hogy az oldal címblokkját az **Oldalak** fülben a kívánt helyre húzzuk. A tematikus fejezetek főoldalának létrehozásához húzzuk a fejezetek többi oldalának nevét tartalmazó blokkokat a szakasz főoldalának nevéhez! A 4.8. ábra az **Oldalak** fül megjelenését mutatja, amikor a *Fotógaléria* oldalt az *Osztályunk* tematikus fejezetbe helyeztük át.

Így a webhelyfejlesztő alkotja a webhely felépítését és a navigációs sávot.

4.8. ábra.  
Az **Oldalak** fül

### Objektumok elhelyezése a weboldalon









A weboldalakon különféle objektumokat – szövegmezőt, képeket, hivatkozásokat, **Google**-dokumentumokat – helyezhetünk el.

A weboldalhoz hozzáadott összes objektum külön blokkokban kerül elhelyezésre. Minden blokk áthelyezhető, átméretezhető vagy törölhető. Minden kiválasztott blokknak megvan a saját beállításpanelje.

A **Beszúrás** fül vezérlőelemeinek rendeltetését a 4.1. táblázat mutatja be.

4.1. táblázat

#### A Beszúrás fül vezérlőelemeinek rendeltetése

| Vezérlő elem  | Rendeltetés  |
|---|--|
| <br>Текстове поле      | Szövegmező. A szöveg formázható mint <i>Főcím</i> , <i>Alcím</i> , <i>Normál szöveg</i> és <i>Kicsinyített</i>   |
| <br>Зображення         | Kép beillesztése a <b>Google-Drive</b> -ból, URL-s internetcímből, számítógépes adathordozóból   |
| <br>Вставити           | Különböző objektumok beillesztése az internetről URL címük alapján, vagy HTML-kódrészlet beillesztése  |
| <br>Диск               | Különböző típusú dokumentumok behelyezése a <b>Google Drive</b> -ból   |
| <br>Макети             | Az objektumok elrendezésének lehetőségei   |
| <br>Зміст              | Az oldal tartalomjegyzékének létrehozása. A tartalomjegyzék automatikusan az oldal szövegdozsaiba helyezett fejlécekből jön létre, és gyors átmenetre szolgál az aktuális oldal kiválasztott címéhez |
| <br>Карусель зображень | Diavetítés létrehozása a kiválasztott képekből   |
| <br>Роздільник         | Vízszintes elválasztóvonal behelyezése   |



| Vezérlő elem | Rendeltetés   |
|--------------|---|
| YouTube      | <b>YouTube</b> videó beillesztése                     |
| Календар     | <b>Google Naptárba</b> létrehozott naptár behelyezése |
| Карта        | <b>Google Térképbe</b> létrehozott térkép behelyezése |

A beillesztett objektum törléséhez ki kell választani a **Törlés** gombot az objektum beállítási paneljén.

A létrehozott webhely oldalainak előlnézetét az **Előlnézet** gombbal tekinthetjük meg, amely az ablak tetején helyezkedik el. Ebben a nézetben láthatjuk, hogyan fog kinézni a webhelyünk az interneten, okostelefonon, táblagépen vagy nagy képernyőn.



Az előlnézeti kezelőpanel gombjai erre a célra vannak hozzárendelve (4.9. ábra). A gomb kiválasztásával ki léphetünk az előlnézeti módból.

4.9. ábra. Az Előlnézet kezelőpanel gombjai

### Webhely közzététele

Az általunk létrehozott webhely nem lesz elérhető az internet felhasználók számára mindaddig, amíg közzé nem tesszük. Ehhez a következőképpen járunk el:

1. Válasszuk a **Közzététel** gombot a böngészőablak tetején!
2. Írjuk be a webhely URL-címének utolsó részét kisbetűkkel, számokkal és kötőjelekkel! A cím első része (a szerver tartományneve és a fájl elérési útja) minden webhelynél megegyezik: [sites.google.com/view/](https://sites.google.com/view/). Az utolsó résznek egyedinek kell lennie minden webhelynél. Ha a megadott cím nem egyedi, erről értesítést kapunk, akkor meg kell változtatni.
3. Nyomjuk meg a **Közzététel** gombot!

A közzétett webhelyet úgy tekinthetjük meg, hogy a **Közzététel** listából kiválasztjuk a **Közzétett webhely megtekintése** lehetőséget. Az a cím, amelyet az internetfelhasználóknak el akarunk majd küldeni, hogy meg tudják tekinteni a weboldalunkat, a címsorban lesz a közzétett webhely nézetablakában.

Miután bármilyen változtatást végrehajtottunk a webhelyen, újra közzé kell tenni, hogy az internetfelhasználók láthassák a frissített weboldalt. A webhely URL-jének megadására csak a webhely első közzétételekor van szükség.

Szükség esetén a közzétételt megszakíthatjuk, ha a **Közzététel** gombra kattintva kiválasztjuk az **Eltávolítás** parancsot. A webhely az újbóli közzétételig nem lesz elérhető az internetfelhasználók számára.

### **Használjuk a számítógépet!**



Végezzétek el a gyakorlati feladatot a <https://cutt.ly/yhdrw8m> linkről, vagy a QR-kód segítségével!

## ! Összefoglalás

Webhely létrehozásához használhatjuk a webfejlesztés automatizált létrehozásának eszközeit: speciális webszerkesztőket, **WCMS** tartalomvezérlő rendszereket, online webhelytervezőket.

Webhely létrehozása online webhelytervezési rendszerek segítségével több lépésben zajlik: fiók regisztrálása az online webhelytervező szerveren, a webhelynév és a tervezősablon kiválasztása, a webhelyoldalak és a navigációs rendszer létrehozása, az oldalak tartalommal való kitöltése, az oldal közzététele.

## ? Felelj a kérdésekre!

- 1°. Milyenek lehetnek a webfejlesztés automatizált létrehozásának eszközei?
- 2°. Milyen lépései vannak a weboldal létrehozásának a **Google webhelyeken**?
- 3°. Hogyan hozhatunk létre webhelyet és nevezhetjük el azt a **Google webhelyek** segítségével?
- 4°. Hogyan változtathatjuk meg az oldal fejlécét és láblécét?
- 5°. Hogyan válasszuk ki a webhely tervezési témáját? Mely tulajdonságértékek választhatók ki a tervezési témával együtt?
- 6°. Hogyan lehet új oldalt létrehozni a webhelyen? Hogyan helyezhető át egy oldal egy másik tematikus fejezetbe?
- 7°. Milyen objektumok illeszthetők be a **Google webhelyek** által létrehozott webhelyek oldalaira?
- 8°. Miért teszik közzé a webhelyeket? Hogyan lehet közzétenni egy oldalt?
- 9°. Milyen a **Google webhelyek** által létrehozott webhelyek címszerkezete?
- 10\*. Szerinted milyen előnyei és hátrányai vannak a webhelyek létrehozásának az oldaluk HTML-kódjának megírásával és az online webhelytervezési rendszerek használatával?

## 📁 Végezd el a feladatokat!

- 1°. Válassz oldalcímeket, és hozz létre egy webhelyterképet a szövegszerkesztőben a *Veszélyes állatok* témához! Mentsd el **feladat 4.2.1.docx** néven a saját mappádba!
- 🏠 2°. Válassz oldalcímeket, és hozz létre egy webhelyterképet a szövegszerkesztőben a *Rekordok könyve* témához! Mentsd el **feladat 4.2.2.docx** néven a saját mappádba!
- 3°. Hozz létre egy webhelyet a **Google webhelyek** segítségével a 4.7. ábrán látható webhelyterképnek megfelelően! Nevezd el a webhelyet *Osztályunknak*! Válaszd a *Diplomata* tervezési témát! Tedd közzé a webhelyet szöveg feltöltése nélkül! Add meg a tanárodnak a létrehozott webhely URL-címét!
- 4°. Hozz létre egy *Úrszomszédaink* nevű webhelyet a **Google webhelyek** segítségével! A webhelyterkép, az oldalakon elhelyezett szöveg és képek a **4fejezet4.2rész4.2.4feladat** mappába található. Válaszd a *Benyomás* tervezési témát! Tedd közzé a webhelyet! Add meg a tanárodnak a létrehozott webhely URL címét!
- 🏠 5\*. Hozz létre egy webhelyet *A kedvenc sportágam* témához a **Google webhelyek** segítségével! Készíts webhelyterképet, válassz anyagokat a webhely tartalmi részének kitöltéséhez! Tedd közzé a webhelyet! Add meg a tanárodnak a létrehozott webhely URL-címét!





**6\***. Alkossatok 2-4 fős csoportokat! Beszéljétek meg a tartalmat, készítsetek webhelyterképet, gyűjtsetek anyagokat és készítsetek webhelyet a következő témák egyikére:

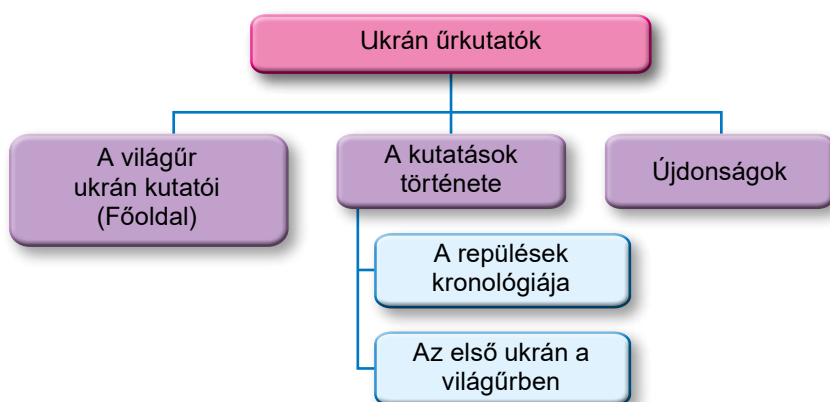
- A számítógépek előnyei és hátrányai;
  - A világűr ukrán kutatói;
  - Polgárok vagyunk;
  - Mentsük meg együtt a bolygót;
  - Az iskolások és az üzlet!
- Adjátok meg a tanárnak a létrehozott webhely URL-jét!

## 5. SZÁMÚ GYAKORLATI MUNKA

### Weboldalak készítése online webhelytervező rendszer segítségével

**Figyelem!** A számítógép használata során tartsd be a balesetvédelmi szabályokat és az egészségügyi előírásokat!

- Hozzunk létre a **Google webhelyeken** egy weblapot a *Ukrán űrkutatókról*!
- Válaszuk ki a *Vision* témának megfelelő sablont!
- A webhelyterkép legyen a következő:



- A *Főoldal*ra helyezzünk el információkat a webhely többi oldaláról!
- Adjuk meg adatainkat webhelyfejlesztőként a láblécben!
- A *kutatások története* oldalt töltsük fel adatokkal, például a **4fejezetű 5gyakorlati űrkutatás.docx** fájlból!
- Az *első ukrán a világűrben* oldalon helyezzünk el a **4fejezetű 5gyakorlati popovics** mappában található **Pavlo Popovics.docx** fájlban található szöveget és az ugyanebben a mappában található képet!
- Töltsük fel a **4fejezetű 5gyakorlati repülések.docx** fájlt a saját **Google Drive**-unkra! Ágyazzátok be a feltöltött fájlt *A repülések kronológiája* oldalra!
- Az *Újdonságok* oldalon tegyük közzé *Az orbitális pályákról* szóló **YouTube** videót, amelyet a *Mars* keresési lekérdezéssel találunk meg!
- Tegyük közzé a weboldalunkat *kosmos-vezeték* néven!
- Tekintsük át a közzétett weboldalunkat!
- Adjátok meg a tanárnak a létrehozott webhely URL-jét!

# 5. fejezet.

## Multimédiás objektumok feldolgozása

Ebben a fejezetben megismerkedtek:

- a multimédia fogalmával
- az audio- és videoformátumok digitalizálásának alapelveivel
- az audio- és videofájlok formátumaival
- a multimédiás objektumokat feldolgozó programokkal
- az audio- és videoformátumok átalakításának eszközeivel
- a hang és a videó rögzítésével
- az audio- és videofájlok létrehozásával
- az audio- és videofájlok közzétételével az interneten
- internetes forrásokból származó tartalom felhasználásával a szerzői jogok figyelembevételével

### 5.1. Multimédiás objektumok feldolgozása



1. Milyen üzeneteket neveznek multimédiásnak?
2. Az adatfeldolgozás mely eszközeit ismeritek?
3. Milyen eszközökkel tudunk képet, zenét, hangoskönyvet, videófilmet tárolni és megjeleníteni?

#### A multimédia fogalma

A korábban tanultak alapján már tudjátok, hogy az üzeneteket többféleképpen lehet megadni: szöveggel, számokkal, grafikával, hanggal, videóval, feltételes jelekkel, speciális szimbólumokkal és kombináltan – többféle módon kombinálva.

A kombinált módon megadott üzenet lehet:

- egy folyóirat cikke a város építészetét illusztráló fotókkal;
- egy népszerű dal videóklipje;
- egy oktató videó **Ukrajna történelmének eseményeiről**;
- egy előadás az osztály tanulójának kirándulásáról, a hozzá tartozó fényképekkel és emlékezetes helyek leírásával;
- a parkban tett sétátok videoanyaga.

A kombinált módon megadott üzeneteket **multimédiásnak** is nevezzük. A multimédia (latin *multum* – sok, *medium* – eszközök, módszerek, nyilvános) – az üzenetek megjelenítésének különböző módjainak kombinációja. Különböző érzékszervek vesz-

nek részt az ilyen üzenetek érzékelésében, és a **szöveget, a grafikát, az audiót és a videót** pedig **multimédiás objektumoknak** nevezzük.

Gyakran, amikor multimédiáról beszélünk, audio- és videoobjektumokra gondolunk.

A multimédiát az élet különböző területein használják. Különösen aktívan használják őket számítógépes alkalmazásokban, különösen azokban, amelyeket tanulásra terveztek. Ezeket az alkalmazásokat *elektronikus tankönyveknek* vagy *kézikönyveknek* is nevezik (5.1. ábra). Szöveges adatok mellett tartalmaznak, audio- és videoanyagokat, fotókat, diagramokat, grafikákat, animációkat, festmények reprodukcióit, valós vagy képzeletbeli tárgyak térbeli modelljeit stb, valamint lehetőséget biztosítanak a felhasználó önálló tanulására.



5.1. ábra. Az első osztályos **Művészet** elektronikus tankönyv részlete



5.2. ábra. Az **Új év** számítógépes játék

A multimédiát széles körben használják a filmgyártásban, a videókban és az animációkban. Lehetetlen elképzelni a modern számítógépes játékokat multimédiás objektumok nélkül (5.2. ábra).

### Az audió (hang) és a videó digitalizálásának alapelvei

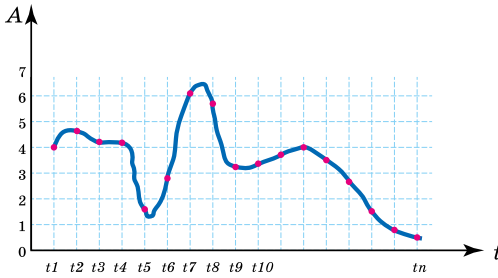
Amint azt a tankönyv első fejezetéből már tudjátok, az összes adatot, amellyel a számítógépes eszközök működnek, valamilyen módon kódolni kell. A kódolást nemcsak a különféle szimbólumok és grafikák, hanem az audio- és videoadatok számára is elvégzik.

A hang egy hullám, amely különböző közegekben terjed (gáz, folyadék, szilárd anyagok) (5.3. ábra). Az emberi érzékszervek, amint a biológiából tudjátok, a levegőtől származó hang érzékelésére vannak hangolva, bár bizonyos körülmények között a vízben vagy a szilárd anyaggal érintkezve is érzékeli a hangot. Az audiojel átalakítása a számítógépes eszközök által feldolgozható adatokká két szakaszban történik:

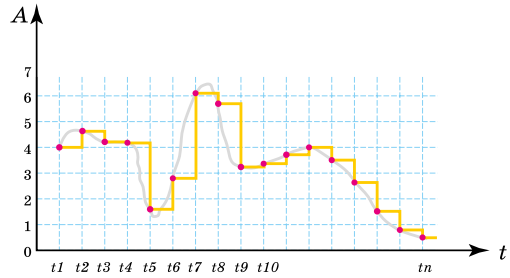
**Analóg** (gör. ανάλογος – arányos, ennek megfelelő) **jel** – a lejátszáskor keletkezett folyamatos jel.

**Diszkrét** (lat. *diskretus* – szakaszos, külön részekből álló) **jel** – olyan jel, amelynek értéke idővel a többszörösére változik.

- a hang átalakítása folyamatos (analóg) elektromos jellé mikrofonomok, elektronikus hangfelvevő eszközök segítségével történik;
- az analóg elektromos jel diszkrétre történő átalakítása (digitalizálása) úgy történik, hogy bizonyos időközönként megmérjük az analóg jel alapvető tulajdonságainak értékét, és ezeket az értékeket számsorozat formájában rögzítjük.



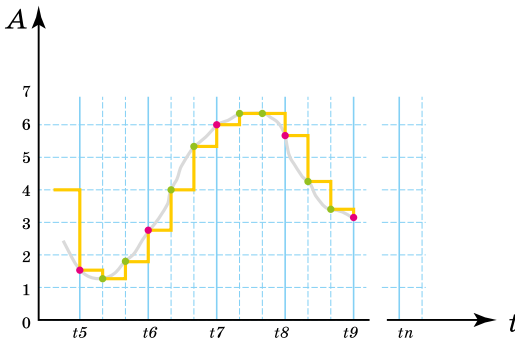
5.3. ábra. Az analóg amplitúdómodulált jel grafikonja



5.4. ábra. A diszkrét amplitúdómodulált jel egyszerűsített grafikonja (sárga vonal)

Az ilyen átalakításokat a számítógépes hangkártya végzi. A diszkrét jel egyszerűsített grafikonja az 5.4. ábrán látható.

Észrevehető, hogy az adott grafikon egy törött vonal. A töréspontok azok a pontok, ahol a jeltulajdonság értékét méri, és ez az érték a grafikonon változatlan marad a következő mérésig. A tulajdonság értékének kis változásával a diszkrét jel grafikonja közel áll az analóghoz (a grafikonon a  $t_1$ -től  $t_4$ -ig és  $t_{10}$ -tól  $t_n$ -ig). Ezekben a területeken a digitalizálás jelentős torzítás nélkül ment végbe. Más területen (például  $t_6$ -tól  $t_9$ -ig) a diszkrét jel grafikonja jelentősen eltér az analógtól. A kezdeti jel eltorzult.



5.5. ábra. A diszkrét amplitúdómodulált jel grafikonja az idő függvényében a mérések számának növelésekor

A torzítások elkerülése és a kiváló minőségű digitális hang biztosítása érdekében növeljük a mérések számát, azaz csökkentjük a mérések közötti időintervallumokat. Az 5.5. ábra a diszkrét jel grafikonját mutatja a  $t_6$  és  $t_9$  közötti intervallumon, amikor a mérések száma megháromszorozódott. Mint látható, a diszkrét jel grafikonja közelebb került az analóg jel grafikonjához. Az audiodigitalizálás minőségének javítása a hangfájlok méretének növekedésével jár.

Az adatak digitalizálásának fő tulajdonságai között szerepel a mintavételezési frekvencia – az audiojel tulajdonságai értékeinek, vagyis a másodpercenkénti méréseinek száma.

Mértékegysége a hertz (Hz). A frekvenciaértékek a digitalizálandó hang jellemzőitől függően változnak. Így telefonbeszélgetéshez elegendő 8 KHz frekvencia, kiváló minőségű zeneművek felvételéhez pedig legalább 44 KHz frekvencia.

A video-adatkódolás a grafikus és az audio-adatkódolás kombinációja. Mivel a videó minden képkockája kép vagy fénykép, grafikus kódolást használnak, és a hangsvot külön kódolják a fent leírt adatak kódolási szabályai szerint.



## Audio- és videoformátumok

Már megismerkedtünk a fájlformátum fogalmával. A fájl típus a fájlban tárolt adatok meghatározott struktúráját jelenti. Ismeritek a legelterjedtebb szöveges és grafikus formátumokat, a prezentációk és az elektronikus táblázatok formátumait. Az audio- és videoadatok szintén meghatározott formátumú fájlokban tárolják, ezeket multimédiás fájl típusokként is emlegetik. Ezeket a fájl típusokat foglaltuk össze az 5.1. táblázatban.

5.1. táblázat

### Multimédiás fájlformátumok

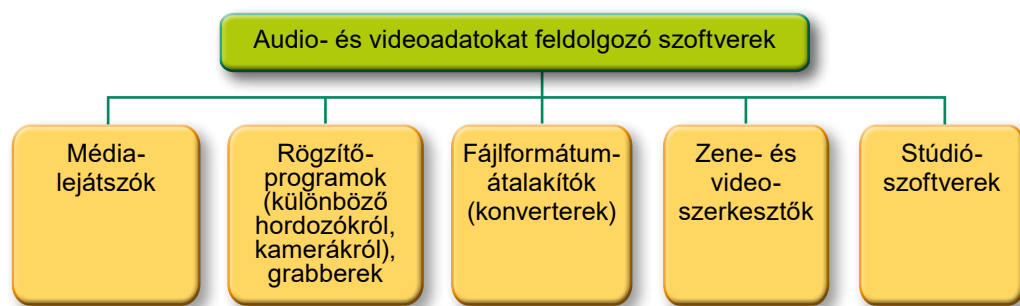
| Formátum  | Leírás   | Kiterjesztés     |
|---|--|------------------|
| <b>Tömörítés nélküli vagy veszteségmentes tömörítést használó formátumok</b>  |  |                  |
| <i>Audioformátumok</i>  |  |                  |
| <b>WAV</b> vagy <b>WAVE</b> (az ang. <i>waveform</i> – hullámalak szóból)   | A <b>Windows</b> alapértelmezett hangformátuma tömörítetlen audioadatok tárolására                 | <b>wav</b>       |
| <b>MIDI</b> (ang. <i>Musical Instrument Digital Interface</i> – hangszeres digitális interfész)   | Speciális eszköz vagy szoftver számára értelmezhető utasításokat tartalmazó fájl                   | <b>mid, midi</b> |
| <b>FLAC</b> (ang. <i>Free Lossless Audio Codec</i> – szabad veszteségmentes audiokodek)   | Szabad szoftver licenc alatt közzétett veszteségmentesen tömörített hangfájl                       | <b>flac</b>      |
| <b>Veszteségesen tömörített formátumok</b>  |  |                  |
| <i>Audioformátumok</i>  |  |                  |
| <b>MP3</b> vagy <b>MPEG-1/2/2.5 Layer 3</b> (ang. <i>Motion Picture Experts Group</i> – mozgóképfelkészítők szakértői csoport, <i>Layer 3</i> – harmadik szint) | A legelterjedtebb hangfájl típus. Fájlcserélőkben, interneten alkalmazzák hangadatok továbbítására | <b>mp3</b>       |
| <i>Videoformátumok</i>  |  |                  |
| <b>MP4</b> (vagy <b>MPEG-4 Part 14</b> )  | Videofilmek tárolására, valamint digitális televíziózásban használt fájl típus                     | <b>mp4</b>       |
| <b>WMA/WMV</b> (ang. <i>Windows Media Audio/Video</i> )   | A <b>Windows</b> szabványos fájl típusa audio- és videoadatok tárolására és lejátszására           | <b>wma, wmv</b>  |
| <b>QuickTime</b> (ang. <i>Quick Time</i> – gyors idő)   | Az <b>Apple</b> által kidolgozott fájl típus audio- és videoadatok tárolására és lejátszására      | <b>mov</b>       |

Mivel az audio- és videofájlok jelentős része tömörített, lejátszásuk során szükség van az adatok visszaállítására is. Ehhez szükség van az adatok **kódolására** és **dekódolására** alkalmas szoftverre, az úgynevezett **kodekekre**. A kodekek általában részei az audio- és videofeldolgozó programoknak, de léteznek különálló programként is, ilyen például a **K-Lite Codec**.

### Multimédiás objektumokat feldolgozó szoftverek

Már megismerkedtünk egyes multimédiás objektumokat feldolgozó programokkal, médialejátszókkal, mivel zenét vagy videót játszottak le a számítógépeken. Lehet, hogy már video- vagy hangfelvételeket készítettetek okostelefonok vagy táblagépek segítségével. Bizonyos alkalmazásokat is használtatok erre a célra. A 5.6. ábrán a multimédiás objektumokat feldolgozó programok általános osztályozását mutatjuk be.

Ismerkedjünk meg részletesebben ezekkel a programokkal és működésük algoritmusával.



5.6. ábra. Audio- és videofeldolgozó programok osztályozása



### Ha többet szeretnél tudni

#### Audio- és videolejátszók

Többféle multimédiás lejátszók léteznek. Egyesek képesek megjeleníteni a audio- és videoadatokat, mások viszont csak egyféle adat lejátszására specializálódnak, ez esetben őket vagy audiorejátszóknak vagy videolejátszóknak nevezik. A legelterjedtebb és legnépszerűbb lejátszók a **Winamp**, **Media Player Classic**, **QuickTime Player**, **Flv Player** stb.

Az operációs rendszerek rendelkeznek egy vagy több médialejátszóval. A **Windows 10** operációs rendszer médialejátszója a **Windows Media Lejátszó**. Segítségükkel meghallgatható az audiofájl és megnézhető a videofájl, valamint sorba rendezhetjük a kívánt médiafájljainkat.

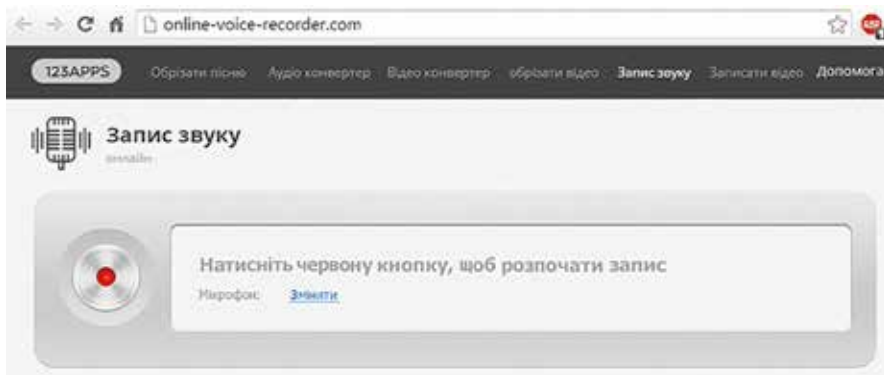
#### Audió és videó rögzítése

Az audio- és videoadatokat multimédiás **rögzítőprogramok** – **grabberek** (ang. *grabber* – fogó, rabló), **rekorderek** (ang. *record* – felvevő) – segítségével olvassuk be a mikrofonomokról, kamerákról, egyéb eszközökről. Az adatokat ezek a programok mentik el a megfelelő formátumokba.







Ezek a programok teszik lehetővé, hogy hangos kommentárt csatoljunk a prezentációhoz, rögzítsük a **Skype** vagy a **Zoom** videokonferenciát vagy a webes sportközvetítést stb. Ilyen rögzítőprogram a **Windows 10 Hangrögzítő**, az **Audacity**, **Audio-grabber**, **Streamripper**, **AML Easy Audio Recorder**, **Free Sound Recorder** stb. Manapság már internetes hang- és videorögzítő szolgáltatásokat is igénybe vehetünk, például: <http://vocaroo.com>, <http://online-voice-recorder.com> (5.7. ábra).








5.7. ábra. Az internet 123APPS-s hang és videorögzítő szolgáltatása

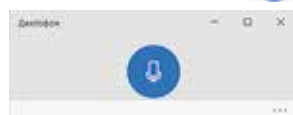
Ha tehát a mikrofonról szeretnénk hangot felvenni, a következőt kell tenni:

1. Kapcsoljuk be a mikrofont!
2. Indítsuk el a rögzítőprogramot **Start** ⇒ **Diktafon!**
3. A **Felvétel** gomb megnyomásával indítsuk el a felírást  (5.8. ábra)!
4. Mondjuk a mikrofonba a szöveget!
5. A megfelelő gombra kattintva állítsuk le a rögzítést (például a **Stop** gomb  megnyomásával)!

A rögzített fájl az **Ez a gép** ⇒ **Dokumentumok** ⇒ **Hag-felvételek** mappába automatikusan mentődik **rekord(n)** néven (ahol  $n$  – a rögzített hanganyag sorszáma).

A felvett hang meghallgatásához és szerkesztéséhez válasszuk ki a nevét a programablakban! A megfelelő gombok kiválasztásával a következő műveleteket lehet végrehajtani a létrehozott felvételen (5.9. ábra):


- a felvétel meghallgatása – **Lejátszás** gomb  ;
- jelölés hozzáadása a felvétel bizonyos részének gyors ugrásához vagy levágásához – **Jelölő hozzáadása**  gomb.
- a létrehozott hangot más felhasználókkal való megosztása – **Megosztás**  gomb;
- felvétel megvágása – **Vágás** gomb.  ;
- felvétel átnevezése – **Átnevezés** gomb  .



5.8. ábra. A **Diktafon** hangrögzítő

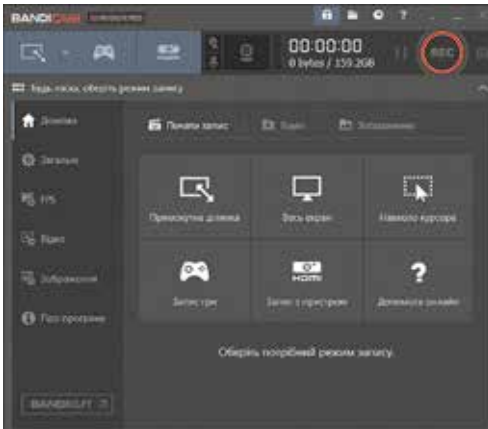


5.8. ábra. A **Diktafon** hangrögzítő lejátszási nézetben


A felvétel kivágása csak a felvétel elején vagy a végén lehetséges, ehhez a **Vágás** gomb  (5.10. ábra) kiválasztása után helyezzük a bal vagy jobb alsó jelölőt (vagy mindkettőt együtt) abba a helyzetbe, ahová vagy ahonnan a felvétel levágásra kerül.



5.10. ábra. A **Diktafon** hangrögzítő vágás közben



5.11. ábra. A **Bandicam** alkalmazás ablaka



A szerkesztett felvétel mentéséhez kattintsunk a **Mentés**  gombra, és válasszuk a mentési lehetőségek egyikét – *meglévő felvétel lecserélése, vagy mentés új fájlba*.

A továbbiakban az elmentett felvételek lejátszhatóak ugyanezzel a programmal, vagy egy másik lejátszóval.

**Videó rögzítése.** Videó rögzítésekor kiválaszthatjuk, hogy a képet és a hangot, vagy csak a képet, ha csak azt szeretnénk rögzíteni. A számítógéphez természetesen a választásnak megfelelő külső eszközt (amely tehát videót vagy hangot képes lejátszani) kell csatlakoztatni.

Videót webkameráról, a lejátszó ablakából vagy egy webhelyről rögzíthetünk. Minden esetben a célnak megfelelő programot kell használni. Videó rögzítésére alkalmas program az **OBS Studio**, **FlashBack Express**, **ApowerREC**, **XSplit Broadcaster**, **VLC Media Player**, **Bandicam** stb.

Az ingyenes **Bandicam**mal <https://www.bandicam.com/ua/> (5.11. ábra) a következőképpen történik a rögzítés:

1. Nyissuk meg a programot, amelyből a videót rögzíteni szeretnénk, vagy nyissuk meg a weboldalt!
2. Indítsuk el a **Bandicam**ot!
3. Válasszuk ki a program üzemmódját, vagyis az eszközt, ahonnan a rögzítés történik (például a képernyő egy téglalap alakú területe)!
4. A speciális kurzor segítségével jelöljük ki azt a képernyőrészt, ahonnan a videó rögzítése történik!
5. Indítsuk el a rögzítést a **REC**  gombbal!
6. Indítsuk el a lejátszást a programból vagy a webhelyről (5.12. ábra)!
7. A rögzítés befejeztével állítsuk azt le a **Stop**  gombbal!

A videó automatikusan mentésre kerül az alapértelmezett vagy a felhasználó által beállított beállítások szerint.

A létrehozott videót ugyanabban a programban tekinthetjük meg. Ehhez:

1. Válasszuk a **Videó** fület!
2. Válasszuk ki a mentett fájl nevét a listából!
3. Nyomjunk a **Lejátszás** gombra!

A fenti algoritmus kisebb változtatásokkal más rögzítőprogram estében is alkalmazható.



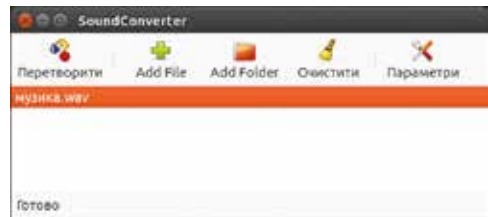
5.12. ábra. A **Bandicam**-alkalmazás, amint a képernyő téglalap alakú területéről rögzítést végez

A hang- és videoanyagok rögzítésekor figyelembe kell venni a szerzői jogok védelmének előírásait. Az online szolgáltatásokban, például a **YouTube**-on közzétett anyagok nagy része már feltételezi, hogy a szerző hozzáférést biztosít annak megtekintéséhez. Ezek az anyagok azonban általában nem biztosítanak kereskedelmi célú terjesztést vagy terjesztést a szerző csatornájára való hivatkozás nélkül.


### Audio- és videoformátum konverterek

A felhasználói igényektől függően sokszor szükség lehet a rögzített anyagok fájlformátumának átalakítására. Megegyezik, hogy a médialejátszón nem alkalmas bizonyos formátumok lejátszására. Ilyen esetekben speciális szoftvereszközökre, úgynevezett **konverterekre** van szükség. Ilyenek a **HAMSTER Free Audio Converter**, **SoundConverter XRECODE II**, **TAudioConverter**, **AudioConverter Studio** stb.

Miután elindítottuk a **SoundConverter** ([https:// soundconverter.org](https://soundconverter.org)) (5.13. ábra), az audiofájlok átalakítását a következőképpen végezzük:



5.13. ábra. A **SoundConverter** ablaka átalakításra váró fájlal

1. Kattintsunk az **Add File**  (ang. *Add File* – fájl hozzáadása) gombra, és jelöljük ki a konvertáló fájlt!
2. A **Paraméterek** gombra kattintva a megnyíló **Format** listán válasszuk ki, milyen formátumra szeretnénk a fájlt átalakítani!
3. Indítsuk el az átalakítást az **Átalakítás** gombbal!

Az átalakított fájlt alapértelmezés szerint az eredetivel azonos mappába lesz mentve. A **Paraméterek** gombra kattintva más mappát is kijelölhetünk az átalakított fájl mentésére.



5.14. ábra.  
A HAMSTER Free  
Video Converter  
ablaka

Videofájlok konvertálására rengeteg különböző licenc alatt terjesztett szoftver létezik. Ilyenek például a **HAMSTER Free Video Converter**, az **AVS Video Converter** az **Any Video Converter Free**, a **SUPER** stb.

Vizsgáljuk meg, hogy történik az átalakítás a **HAMSTER Free Video Converter** esetében. Miután elindítottuk a programot:

1. A **Fájl hozzáadása** gombbal jelöljük ki az átalakítandó fájlokat!
2. Kattintsunk a **Tovább** gombra!
3. Adjuk meg a célformátumot, majd szükség esetén annak paramétereit (5.14. ábra)!
4. Kattintsunk az **Átalakítás** gombra!
5. Jelöljük ki, hogy melyik mappába kívánjuk menteni az átalakított fájlt!

A konvertereknek léteznek webes változatai is, mind audio-, mind videofájlok esetében, például a <http://audio.online-convert.com>, <https://convert-video-online.com/> stb.



### Linux-használóknak

Ha a telepítés során nem lettek volna telepítve az audio- és videofeldolgozó programok, az **Ubuntu szoftverközpont** lehet a segítségünkre. A **Programok** listájából keressük ki a hang- és videofeldolgozó programokat, választuk ki ezek közül azokat, amelyek nekünk szükségesek, majd kattintsunk a **Telepítés** gombra (5.15. ábra)!



5.15. ábra. Az Ubuntu szoftverközpont ablaka



### Használjuk a számítógépet!



Végezték el a gyakorlati feladatot a <https://cutt.ly/XhdreNg> linkről, vagy a QR-kód segítségével!



## Összefoglalás

A kombinált módon megadott üzeneteket **multimédiásnak** is nevezzük. A multimédia az üzenetek megjelenítésének különböző módjainak kombinációja. Különböző érzékszervek vesznek részt az ilyen üzenetek érzékelésében, és a **szöveget, a grafikát, az audiót és a videót** pedig **multimédiás objektumoknak** nevezzük.

Figyelj arra, hogy gyakran, amikor multimédiáról beszélünk, audio- és videoobjektumokra gondolunk!

Az audiojel átalakítása a számítógépes eszközök által feldolgozható adatokká két szakaszban történik:

- a hang átalakítása folyamatos (analóg) elektromos jellé mikrofonom, elektronikus hangfelvevő eszközök segítségével történik;
- az analóg elektromos jel diszkrétre történő átalakítása (digitalizálása) úgy történik, hogy bizonyos időközönként megmérjük az analóg jel alapvető tulajdonságainak értékét, és ezeket az értékeket számsorozat formájában rögzítjük.

Az audioadatok digitalizálásának fő tulajdonságai között szerepel a **mintavételezési frekvencia** – az audiojel tulajdonságai értékeinek másodpercenkénti méréseinek száma. Hertzben (Hz) mérjük.

A video-adatkódolás a grafikus és az audio-adatkódolás kombinációja. Mivel a videó minden képkockája kép vagy fénykép, grafikus kódolást használnak, és a hangsvot külön kódolják a fent leírt audioadatok kódolási szabályai szerint.

Az audio- és videoadatok különböző formátumokban tárolódnak. Az audiofájlok ismertebb kiterjesztései a **wav**, a **mid**, a **flac** és az **mp3**. A videofájloké, az **mp4**, a **wmv**, a **mov**, az **flv**.

A tömörített állományok **kódolását** és **dekódolását** végző programokat **kodekeknek** nevezzük.

Az audio- és videofájlokat feldolgozó programok között vannak lejátszók (**playerek**), rögzítőprogramok (**grabberek**, **rekorderek**), fájlformátumok átalakítására szolgáló programok (**konverterek**), szerkesztők, stúdióprogramok.

A hang- és videoanyagok rögzítésekor fontos figyelembe venni a szerzői jogokat.



## Felelj a kérdésekre!

- 1°. Mi az a multimédia? Az élet mely területén alkalmazzuk? Hozz fel példákat!
- 2°. Hogyan történik a hang digitalizálása? Milyen eszközök szükségesek ehhez?
- 3°. Mi az a mintavételezési frekvencia? Miben mérjük?
- 4°. Melyek a videoadatok kódolásának jellemzői?
- 5°. Milyen audioformátumokat ismersz?
- 6°. Milyen videoformátumokat ismersz?
- 7°. Milyen típusai vannak a multimédiás adatokat feldolgozó szoftvereknek?
- 8°. Mire szolgálnak a grabberek? Nevez meg ilyen programokat!
- 9°. Milyen forrásokból rögzíthetünk videókat? Hogyan ne sértsünk ilyenkor szerzői jogokat?
- 10°. Milyen lépésekből áll az audio- és videofájlok konvertálása?



### Végezd el a feladatokat!



1\*. Keress információkat az interneten, majd készíts prezentációt azokról a tudósokról, akik megfogalmazták a mintavételi frekvencia minimális értékeit a különféle típusú hangüzenetek kiváló minőségű digitalizálásához!

2°. Rögzítsd a saját előadásodban Tarasz Sevcsenko *Gondolataim* című versét (**5fejezet\ 5.1rész\ gondolataim.doc**) a **Diktafon** hangrögzítőjével! Mentsd a hanganyagot **feladat 5.1.2.wav** néven a mappádba!



3°. A **123APPS** <http://online-voice-recorder.com> internetes rögzítőszolgáltatás segítségével rögzítsd Petőfi Sándor *Itt van az ősz, itt van újra* című versét (**5fejezet\ 5.1rész\ Itt van az ősz, itt van újra.doc**) a saját előadásodban! Mentsd a hanganyagot **feladat 5.1.3.mp3** néven a mappádba!



4°. Rögzíts a webkameráról egy 30 másodperces videót a **Bandicam** program segítségével! Mentsd a fájlt a **Dokumentumok** mappába **feladat 5.1.4.mp4** néven!



5°. Videorögzítő segítségével rögzítsd a videolejátszóval éppen lejátszott **video2.avi** (**5fejezet\ 5.1rész\ video2.avi**) fájlt! Mentsd a rögzített anyagot **feladat 5.1.5.wmv** néven a mappádba!

6\*. Végezd el a tanárod által megadott két **MP3** formátumú fájl (vagy az **5fejezet\ 5.1rész\ Audio** mappa két fájljának) átalakítását **WMA** formátumba valamely audiokonverter segítségével! Mentsd a fájlokat a mappában létrehozott **feladat 5.1.6** mappába! Hasonlítsd össze az eredeti és az átalakított fájlok méretét! Mivel magyarázható a tapasztalt különbség?

7\*. Végezd el a tanárod által megadott két **MOV** formátumú fájl (vagy az **5fejezet\ 5.1rész\ Video** mappa két fájljának) átalakítását **WMV** formátumba valamely konverter segítségével! Mentsd a fájlokat a mappában létrehozott **feladat 5.1.7** mappába! Hasonlítsd össze az eredeti és az átalakított fájlok méretét! Mivel magyarázható a tapasztalt különbség?

## 5.2. Audiók és videók szerkesztése



1. Milyen audio- és videoformátumokat ismertek?

2. Milyen típusai vannak a multimédiás adatokat feldolgozó programoknak?

3. Milyen műveleteket végezhetünk a lejátszók, rekorderek, konverterek segítségével?

### Audio- és videoszerkesztők

Az előző pontban nem vizsgáltuk a multimédiás programok két típusát: a *szerkesztőket* és a *stúdiókat*. Ezekkel a programokkal végezhetjük a multimédiás fájlok szerkesztését. Segítségükkel részleteket törölhetünk, másolhatunk, illeszthetünk be, megváltoztathatjuk a fájlok hosszát, szöveget és képeket illeszthetünk be a videókba stb.

A két típus között az a különbség, hogy a stúdióprogramok funkcionalitása magasabb, vagyis ezek több szerkesztési lehetőséget biztosítanak és professzionális hang-





film- és animációs stúdiókban használatosak. Ilyen programok a hangfeldolgozásban a **Linux MultiMedia Studio**, **Psycledelics Psycle**, **Steinberg Cubase**, **Cakewalk Sonar**, **Adobe Audition** stb; a videofeldolgozásban ide sorolható a **Pinnacle STUDIO**, **Kino**, **Adobe Premiere Pro**, **Avid Media Composer** stb. A stúdióprogramok jellemzően kereskedelmi alkalmazások.

Egyszerű hang- és videoszerkesztés céljára leginkább a különböző szerkesztőprogramok ajánlhatók. Hangfájlok esetén ez lehet az **Audacity**, **Free Audio Editor**, **Wave Editor**, videók estében pedig a **Windows Live Movie Maker**, **Open Shot**, **Shotcut**, **Virtual Dub**, **ZS4 Video Editor** stb.

### Windows Live Movie Maker

A **Windows Live Movie Maker** a **Windows Movie Maker** új verziója. A **Windows 7** és **Windows 10** operációs rendszer alá a program ingyenesen letölthető a **Microsoft** webhelyéről. A **Windows** régebbi verziói esetében a **Movie Maker** része volt az telepítő csomagnak.

A **Windows Live Movie Maker** videoszerkesztő, amellyel videorészleteket, fényképeket, hanganyagot, feliratokat fűzhetünk össze videóvá. A kész videót a program által támogatott formátumban elmenthetjük, vagy közzétehetjük az interneten.

A **Windows Live Movie Maker** videoszerkesztő ablakát az 5.16. ábra szemlélteti.



5.16. ábra. A Windows Live Movie Maker ablaka



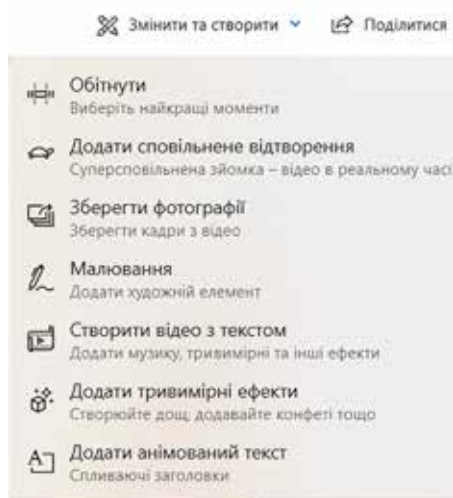
### Azoknak, akik Videoszerkesztő alkalmazással dolgoznak

A **Windows 10** legújabb verzióiban az alapértelmezett operációs rendszerrel telepített alkalmazások tartalmazzák a **Videoszerkesztő** programot, amely integrálva van egy másik szabványos alkalmazással, a **Képnézegetővel** grafikák és videók megtekintésére. A szerkesztőablakot a következő módon nyithatjuk meg: **Start** ⇒ **Video-szerkesztő**. A **Videoszerkesztő** ablak általános nézete videoszerkesztő módban az 5.17. ábrán látható.



1. Videócím
2. Videócím módosítása gomb
3. Fotó és videókönyvtár
4. Megtekintési ablak
5. Videoszerkesztési parancsok
6. Üres terület
7. Videoszerkesztési parancsok
8. A képkocka idő kijelzője
9. Beillesztett videorészlet

5.17. ábra. A **Videoszerkesztő** alkalmazás munkafelülete



5.18. ábra. **Kreatív kép létrehozása** ablak

A meglévő videofájlok egyedi módosításához a **Képnézegető** módban a listából válasszuk ki a **Kreatív kép létrehozása ezzel a fényképpel** gombot, és szerkeszthetjük a képet, például az **Animált szöveg hozzáadása** paranccsal (5.18. ábra).

### Általános fogalmak a videóról

**Video-** vagy **mozifilm** (ang. *cine-film* – filmszalag) alatt általában filmművészeti alkotásokat értünk. Napjainkban nincs jelentős különbség közöttük. Úgy tűnik, hogy a videofilm kifejezés a videomagnók és videokamerák széles körű használatára utal, amelyhez mágnesszalagot használtak adathordozóként,

szemben azokkal a mozifilmekkel, amelyekhez speciális átlátszó anyagból készült fényérzékeny réteggel ellátott filmet használtak.

Ma mind a professzionális film, mind a saját videó készítésekor ugyanazt a technológiát használjuk, hogy rögzítsük a videoadatokat a számítógépes adathordozókra – merevlemezeket vagy elektronikus chipeket. Ezért a *film* és a *videó* kifejezések szinte szinonimává váltak. Bár a videó leggyakrabban nem szakemberek által készített vagy nem moziban való megtekintésre szánt videó.

A rövid videoanyagokat manapság gyakran emlegetik **klip** (ang. *clip* – tömörítés, vágás)vagy **videóklip** néven.



A videó **kockákból** – videorészletekből, állóképekből – áll. A kockák alkotják a videosávot, ezt a szerkesztők a videovonalon helyezik el. A videók a következő elemekből épülnek fel:

- **cím** – néhány szöveges objektum a film elején;
- **feliratok** – szöveges objektumok, amelyek a képi anyag fölött jelennek meg, magyarázzák a látványt, vagy a szerzőket mutatják be;
- **hangzó kíséret**, ami több elemet tartalmazhat:
  - a videó saját hangja;
  - narráció – hangos magyarázat a videóhoz;
  - zene – zenemű részlete, amit általában hangos háttérként használnak;
- **vizuális effektek** – a kép megjelenítésében bekövetkező változás, például áttérés fekete-fehér megjelenítésre régi film hatás céljából;
- **áttűnések** – a kockák váltását kísérő hatások.

Amikor filmet hozunk létre a videoszerkesztőben, a képkockák képsorozatot alkotnak, és a videoszerkesztőbe a **videosávra** kerülnek. A vizuális és átmeneti effektusok általában a videosávon jelennek meg. A filmzene különálló **hangsávokra** helyezhető el. A **Windows Videoszerkesztő** alkalmazásban a videóval rögzített hang megjelenik a videosávon.

A film készítése során célszerű bizonyos algoritmus szerint eljárni:

- **az első lépés** a forgatókönyv elkészítése. Ez egy részletes terv, amelyben meghatározzuk a kockák sorrendjét, az egyes objektumok elhelyezését. Minél részletesebb a forgatókönyv, annál könnyebb a film elkészítése;
- **a második lépés** a szükséges anyagok előkészítése: a helyszínek előkészítése, a forgatás, a képek és kísérőzene kiválasztása;
- **a harmadik lépés** a vágás – az egyes kockák és a film szerkezetének feldolgozása abból a célból, hogy egy egységes mű jöhesse létre. A vágás során történik a nyersanyag szerkesztése, a szükséges narráció elkészítése, a szövegek, feliratok létrehozása.

## A forgatókönyv elkészítése

Vizsgáljuk meg a forgatókönyv elkészítésének menetét a *Papírmentes informatika* videó példáján. A videó V. M. Hluskovnak a számítógépes eszközök felhasználására vonatkozó elképzeléseinek megvalósulásáról szól. Hluskov előre látta, hogy a XXI. század elejére az információ döntő többsége nem papíron lesz tárolva, hanem számítógépek memóriájában.

Hluskov vezetésével ezen elképzelések megvalósítása céljából kifejlesztették a **MIR** (mérnöki célú számítógép) számítógépsorozatot, a személyi számítógép elődjét. A fejlesztés legfontosabb célja a számítógépek alkalmazásának elterjesztése volt a lakosság széles köreiben kortól és végzettségtől függetlenül.

A videót az informatikaórán készülünk bemutatni V. M. Hluskov munkásságának ismertetése céljából. A forgatókönyv a kockák tartalmának, a kockák eseményeinek leírásából áll.

1. kocka **Címoldal:**

- cím – *Papírmentes informatika*;
- háttér – *világoszöld*, betűszín – *sötétzöld*;

- megjelenítés – 7 s;
  - hang – a **Physical.wma** fájl háttérhangja;
  - effektus – kioltás bal oldalról;
  - áttűnés – mozaik.
2. kocka. **V. M. Hluskov fényképe:**
- fénykép a **Hluskov.jpg** fájlból;
  - megjelenítés – 6 s;
  - hang – *narratív szöveg, ebben a kockában kezdődik, a következőben ér véget*; a hangzás hossza a *felolvasás idejétől függ*;
  - effektus – *nincs*;
  - áttűnés – *mozaik*;
  - a narratíva szövege: *V. M. Hluskov, az Ukrán TA Kibernetikai Intézetének igazgatója a papírmentes informatikára történő áttérést a számítástechnika egyik legfontosabb alkalmazási területének tekintette. Úgy gondolta, hogy a XXI. század elejére az információ mennyiségének jelentős része nem papíron, hanem a számítógépek memóriájában lesz tárolva. Az az ember pedig, aki nem lesz képes ezt az információt felhasználni, a XX. század elejének írástudatlanához lesz hasonló.*
3. kocka. **Videofelvétel V. M. Hluskovról:**
- videó – **Hluskov video0.wmv** videó fájlból;
  - időtartam – *a narrátor szövegével egyszerre ér véget*;
  - hang – *általános háttérhang fájlból*; a hang időtartamát a *szöveg reprodukciójának sebessége határozza meg*;
  - effektus – *nincs*.
4. kocka. **Cím:**
- szöveg: *Hallgassuk meg Viktor Mihajlovcis Hluskovot*;
  - háttér – *világoszöld*, betűszín – *sötétzöld*;
  - időtartam – 7 s;
  - hang – *általános háttérhang a Physical.wma fájlból*;
  - effektus – *kioltás balról*; áttűnés – *mozaik*;
5. kocka. **Hluskov ismertetője a papírmentes informatikáról:**
- videó – **Hluskov video1.wmv** videó fájlból;
  - időtartam – 48 s;
  - effektus – *nincs*;
  - áttűnés – *mozaik*.
6. kocka. **V. M. Hluskov a MIR-1 számítógép fejlesztőinek körében:**
- fénykép a **Hluskov\_MIR-1.jpg** fájlból;
  - időtartam – 6 s;
  - hang – *általános háttérhang a Physical.wma fájlból*;
  - felirat: *Ahhoz, hogy az emberek megtanulják a számítógép használatát, át kell térni a kis számítógépekre. V. M. Hluskov felveti a MIR család kifejlesztésének ötletét*;
  - betűszín – *narancs*, háttér – *sötétzöld*;
  - effektus – *nincs*;
  - áttűnés – *mozaik*.



7. kocka. **A MIR-1 számítógép fényképe:**
- fénykép a **MIR-1.jpg** fájlból;
  - képfelirat: *A MIR számítógépek kis méretűek voltak és kényelmes beviteli eszközzel – elektromos írógéppel – voltak felszerelve;*
  - minden más a 6. kockának megfelelően.
8. kocka. **A MIR-2 számítógép fényképe:**
- fénykép a **MIR-2.jpg** fájlból;
  - képfelirat – *a MIR-2 új beviteli eszköze, a fényceruza;*
  - minden más a 6. kockának megfelelően.
9. kocka. **A MIR-3 számítógép fényképe:**
- fénykép a **MIR-3.jpg** fájlból;
  - képfelirat – *a MIR-3 számítógép – a sebesség a MIR-2-höz képest 20-szorosára nőtt;*
  - minden más a 6. kockának megfelelően.
10. kocka. **Zárókép:**
- felirat – **Jövő iskolai stúdió;**
    - operatőr – *Sztepan Mihajlenko;*
    - szöveg – *Okszana Kovalenko;*
    - vágó – *Mikola Ivanenko;*
  - háttér – *sárgászöld, betűk színe – fehér;*
  - időtartam – *10 s.*

## Anyaggyűjtés

Nyomatott kiadványok, internetről letöltött anyagok alapján összegyűjtjük a videó összeállításához szükséges információkat. Ezek listáját tartalmazza az 5.2. táblázat

5.2. táblázat

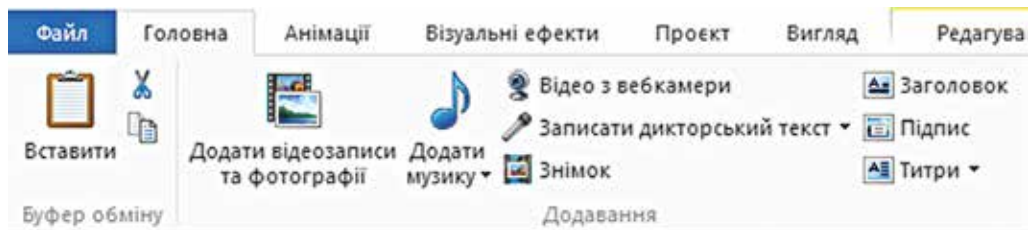
### A videóhoz szükséges fájlok

| Fájl neve           | Mappa                    | Hosszúság, s | Kocka         |
|---------------------|--------------------------|--------------|---------------|
| Physical.wma        | 5.fejezet\5.2.pont\audio | 220          | 1-9, kivéve 5 |
| Hluskov videó 0.wmv | 5.fejezet\5.2.pont\video | 53           | 3             |
| Hluskov videó 1.wmv | 5.fejezet\5.2.pont\video | 48           | 5             |
| Hluskov.jpg         | 5.fejezet\5.2.pont\foto  |              | 2             |
| Hluskov_MIR-1.jpg   | 5.fejezet\5.2.pont\foto  |              | 6             |
| MIR-1.jpg           | 5.fejezet\5.2.pont\foto  |              | 7             |
| MIR-2.jpg           | 5.fejezet\5.2.pont\foto  |              | 8             |
| MIR-3.jpg           | 5.fejezet\5.2.pont\foto  |              | 9             |

## Videó létrehozása

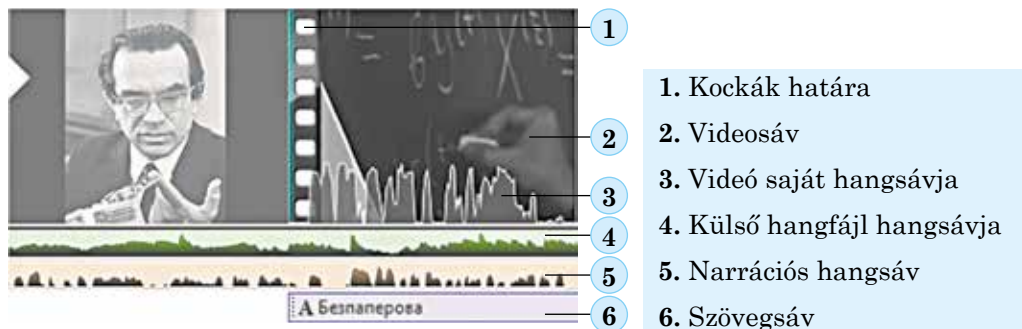
A **Windows Live Movie Maker**-ben a következőképpen készítjük el a filmet:

1. Indítsd el a programot!
2. Mentsd el az új projektet a **Fájl** ⇒ **Projekt mentése** paranccsal! Alapértelmezetten a program **wlmp** kiterjesztésű fájlba menti a projekteket.
3. A **Kezdőlap Hozzáadás** csoportjának vezérlőelemi segítségével helyezzük el a képeket és videoanyagokat a kockákban a forgatókönyvnek megfelelő rendben (5.19. ábra)!



5.19. *ábra.* A **Hozzáadás** csoport elemei

4. A **Kezdőlap** fül **Hozzáadás** csoport elemeinek felhasználásával adjuk hozzá a címetek, feliratokat a forgatókönyv szerint!
5. A **videoeszközök, hangeszközök, kísérszövegek, cím és stáblista** ideiglenes eszköztárak alkalmazásával végezzük el a videó objektumainak szerkesztését! Figyeljünk oda, hogy a videók és képek a videosávon (5.20. 2.), a hangfájlok a hangsávon (5.20. 4.), a narráció egy másik hangsávon (5.20. 5.), a címetek, feliratok, stáblista egy szövegsávon (5.20. 6. *ábra*) jelenjen meg! Amennyiben a videóhoz tartozik saját hang, az a videosáv alsó részében jelenik meg (5.20. 3. *ábra*).
6. Állítsuk be a videosávon a lejátszás idejét, a hangerőt, használjuk az objektumok mozgatását a sávokon, a lejátszás kezdetének és végének beállítását!
7. Szükség esetén állítsunk be effektusokat az egyes kockákra (**Vizuális hatások** lap)!
8. Az **Animáció** lap alkalmazásával állítsuk be a kockák közötti áttünéseket!
9. Mentsük a fájlt a **Fájl** ⇒ **Mentés** parancs végrehajtásával!



5.20. *ábra.* Szerkesztési terület

A szerkesztés során célszerű a projektet időről időre elmenteni. Ha a videó szerkesztését nem fejeztük be, akkor a projektfájl megnyitása után folytathatjuk a szerkesztést. Ne feledjétek, hogy az összes médiafájlnak abban a mappában kell lennie, ahonnan a projektbe illesztették, lehetőleg az egyes projektek külön mappájában! A mentett projekt lehetőséget biztosít arra, hogy egy-egy videót különböző változatokban készítsük el.





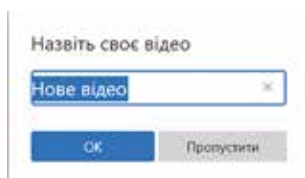
## Azoknak, akik a Videoszerkesztő alkalmazással dolgoznak

A Windows **Live Movie Maker** alkalmazás és a **Videoszerkesztő** alkalmazás közötti fő különbségek:

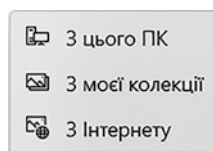
- nincs lehetőség átfedésre a film egyik töredékéből a másikba;
- a feliratokban a betűtípusok tulajdonságai értékeinek megválasztására nincs lehetőség. A feliratok szövegének kialakítását a stíluslistából választják ki;
- a szerző hangkommentárjának hozzáadásához először rögzíteni kell (például a **Hangrögzítő** alkalmazásban) és fájlba kell azt menteni.

Miután elindítottuk az új videó létrehozására szolgáló alkalmazást, válasszuk a **Kezdőlap** ablakban az **Új videoprojekt** gombot, és írjuk be a videó nevét a párbeszédpanelbe (5.21. ábra)! Egy meglévő projekt szerkesztésének folytatásához válasszuk a **Saját videoprojektek** listát!

Videó létrehozásának alapvető lépéseinek algoritmus a **Videoszerkesztő** alkalmazásban.



5.21. ábra. A videó elnevezése ablak



5.22. ábra. Fényképek és videók hozzáadása gombok listája

### Videó és fénykép hozzáadása a projekt könyvtárhoz:

1. A **Projekt dokumentumára** területen válasszuk a **+ Додати** gombot!
2. Válasszuk ki a kívánt videók vagy képek forrását a megjelenő listájából (5.22. ábra)!
3. Hozunk létre egy aktuális forrást (például egy mappát a számítógép adathordozóján), amely tartalmazza a kívánt video- és képfájlokat!
4. Válasszuk ki a videóba beilleszteni kívánt fájlokat, majd kattintsunk a **Megnyitás** gombra!
5. Ismételjük meg az algoritmus 2-4 parancsát az összes szükséges objektum felvételére a projektbe!

### Grafikus vagy videoobjektumot hozzáadása a videóhoz:

1. Jelöljük ki a kívánt objektumot a **Projekt dokumentumára** területen!
2. Húzzuk a kijelölt objektumot a **Történet** területre a szabad téglalapok egyikébe vagy a már kitöltött téglalapok közé!

### Szöveges feliratokat hozzáadása a videó vagy a képkockák elejére:

1. Válasszuk ki a kívánt objektumot a **Történet** területen!
2. A **Történet** területen válasszuk a **Додати картку назви** **Címkartya hozzáadása** gombot!
3. A **Történet** területen válasszuk a **Текст** a **Cím vagy felirat hozzáadása** gombot!
4. Írjuk be a kívánt szöveget a **Szöveg** részbe!
5. Válasszuk ki az egyik szövegstílust az **Animált szövegstílus** listából!
6. Az **Elrendezés** listában válasszuk ki az egyik javasolt módot a szöveg elhelyezésére az ablakban!

7. Válasszuk az előnézeti ablak fölötti **Háttér** фон gombot!
8. Válasszuk ki a kívánt háttérszínt a **Háttér** palettáról!
9. Nyomjuk meg a **Kész** gombot!

### Szöveges feliratok hozzáadása a képkockákhoz:

1. Válasszuk ki a kívánt objektumot a **Történet** területen!
2. A **Történet** területen válasszuk a **Szöveg** Текст részben a **Cím vagy felirat hozzáadása** gombot!
3. Írjuk be a kívánt szöveget a **Szöveg** részbe!
4. Válasszuk ki az egyik szövegstílust az **Animált szövegstílus** listából!
5. Az **Elrendezés** listában válasszuk ki az egyik javasolt módot a szöveg elhelyezésére az ablakban!
6. Nyomjuk meg a **Kész** gombot!

### A szöveges és kép objektumok megjelenítési idejének módosítása:

1. Válasszuk ki a kívánt objektumot a **Történet** területen!
2. Válasszuk az **Időtartam** Тривалість lehetőséget!
3. Válasszuk ki a kívánt értéket a listából, vagy írjuk be magunk!

### Videoobjektumok szerkesztése:

#### 1. Videóklip vágása:

1. Válasszuk ki a kívánt objektumot a **Történet** területen!
2. Válasszuk a **Történet** területen a **Felosztás** Обітнути gombot!
3. Az előnézeti ablak segítségével állítsuk be a vágás kezdetének és végének jelölőit a kívánt helyzetbe (5.23. ábra)!
4. Nyomjuk meg a **Kész** gombot!



5.23. ábra. A vágás folyamatának egy részlete

#### 2. Videóklip kettéosztása:

1. Válasszuk ki a kívánt objektumot az **Történet** területen!
2. Válasszuk a **Történet** területen a **Kettéosztás** Розділити gombot!
3. Az előnézeti ablak segítségével állítsuk a jelölőt oda, ahol ketté szeretnénk osztani a videóklipet (5.24. ábra)!
4. Nyomjuk meg a **Kész** gombot!




5.24. ábra. A Kettéosztás nézet egy részlete

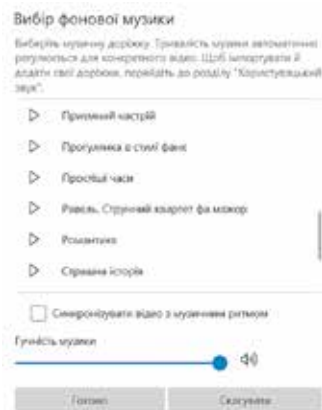
### Hang hozzáadása:

#### 1. Narráció hozzáadása:

1. Kattintsunk az ablak tetején lévő **Egyéni hang** Користувачський звук gombra!
2. Az előnézeti ablak segítségével állítsuk a jelölőt oda, ahol kezdődni fog a hangfájl!




3. Az **Egyéni hang** területen válasszuk az **Audiofájl hozzáadása** + Додати аудіофайл gombot!
  4. Válasszuk ki a kívánt fájlt a **Fájl megnyitása** ablakban, majd kattintsunk a **Megnyitás** gombra!
2. **Háttérhang hozzáadása:**
1. Kattintsunk az ablak tetején lévő **Háttérzene**  Фонова музика gombra!
  2. Válasszuk ki a kívánt zenét a listából (5.25. ábra)!
  3. Állítsuk be a zene hangerejét a csúszka segítségével, majd válasszuk a **Kész** gombot!



5.25. ábra. A **Háttérzene** hozzáadása ablaka

**Videó befejezése:**

1. Kattintsunk az ablak tetején lévő **Videó befejezése**  Завершити відео gombot!
2. Nyissuk meg a **Videó befejezése** ablak **Videóminőség** párbeszédpanelen listát (5.26. ábra), és állítsuk be a kívánt minőségi értéket!
3. Nyomjuk meg az **Exportálás** gombot!
4. A **Fájl mentése** ablakban adjuk meg a videófájl mentési helyét!



5.26. ábra. A **Videó befejezése** ablaka

Az exportálás befejeztével a videó külön ablakban jelenik meg.



**Linux-használóknak**

**Linux** alatt az **OpenShot** programot használhatjuk videók szerkesztésére (5.27. ábra). A program felhasználói felülete, a műveletek elvégzésének módja hasonlít a **Live Movie Maker**-ben megismertekre.



5.27. ábra. Az **Open Shot** ablaka



**Használjuk a számítógépet!**



Végezzétek el a gyakorlati feladatot a <https://cutt.ly/7hrrZx> linkről, vagy a QR-kód segítségével!



### Összefoglalás

A zeneszerkesztők, videoszerkesztők, stúdióprogramok rendeltetése a hangfájlok és a videófájlok szerkesztése. Ezekkel a programokkal másolhatjuk, áthelyezhetjük, törölhetjük a hangfájlok és a videók részeit, képeket, szövegeket illeszthetünk a videóba.

A két típus között az a különbség, hogy a stúdióprogramok funkcionalitása magasabb, vagyis ezek több szerkesztési lehetőséget biztosítanak és professzionális hang-, film- és animációs stúdiókban használatosak.

A **Windows Live Movie Maker** segítségével videókból, képekből, narratív szövegből, feliratokból készíthetünk videókat. A létrehozott videót elmenthetjük a program által támogatott formátumokban, vagy közzétehetjük az interneten.

Videók készítése során célszerű betartani a műveletek következő sorrendjét:

- **első lépésként** el kell készíteni a forgatókönyvet;
- **második lépésként** készítsük elő a szükséges videót, hangfelvételeket, szöveges információkat;
- **harmadik lépésként** végezzük el a vágást!



### Felelj a kérdésekre!

- 1°. Mi a hang- és videoszerkesztők rendeltetése?
- 2°. Mi a különbség a szerkesztőprogramok és a stúdióprogramok között?
- 3°. Milyen hangszerkesztőket ismersz? Milyen videoszerkesztőket ismersz?
- 4°. Mi a **Windows Live Movie Maker** rendeltetése?
- 5°. Mi a **Live Movie Maker Szerkesztőterületének** rendeltetése?
- 6°. Milyen objektumokat lehet beilleszteni egy videóba? Milyenek a tulajdonságaik?
- 7°. Hogy készítenek videofilmet?
- 8°. Mi a forgatókönyv és miért van rá szükség?
- 9°. Hogyan illesztünk a videóba fényképet és videóanyagot?
- 10°. Hogyan illesztjük be a narratív szöveget?
- 11\*. Mit értünk a videosáv és a hangsáv szinkronizálásán? Hogy végezzük el ezt **Live Movie Make**-ben?



### Végezd el a feladatokat!

- 1°. Állíts össze forgatókönyvet, és készíts videót Leszja Ukrajinkáról az ukrán irodalomórára! A videóban használj egy részletet a Leszja Ukrajinka bemutató filmből, illeszd be a költőné képét és az emlékműveit bemutató fotókat! (**5fejezet\ 5.2rész\ feladat5.2.1\Ukrainka**)! Készíts a filmhez címkockát, feliratokat! Keress háttérzenét, ami a film elejétől a végéig szóljon! Mentd a videót **feladat 5.2.1.wmv** néven a mappádba!
- 2°. Készíts a 8. osztályos történelemlétkönyv, valamint az **5fejezet\5.2rész\ feladat5.2.1\ Kolumb** anyagai alapján filmet Kolumbusz Kristófról! A narratív szöveget és a zenei aláfestést válaszd ki önállóan! Mentd a videót **feladat 5.2.2.wmv** néven a mappádba!
- 3°. Állíts össze forgatókönyvet, és készíts videót Olekszandr Dovzszenko neves ukrán íróról és filmrendezőről az ukrán irodalomórára! Használd fel a 8. osztályos irodalomkönyv, valamint az **5fejezet\5.2rész\ feladat5.2.1\ feladat5.2.3\ Dovzszenko** mappa anyagait! A narratív szöveget és a zenei aláfestést válaszd ki önállóan! Mentd a videót a **Dokumentum** mappába **feladat 5.2.3.wmv** néven!





4\*. Készíts az általad választott szerkesztőben egy videót, amiben a nyolcadik osztályban tanulmányozott fizikai folyamatokat mutatod be! A forgatókönyvet és a szükséges anyagokat állítsd össze önállóan! Mentsd a videót a **Dokumentum** mappába **feladat 5.2.4.wmv** néven!



5\*. Készíts az általad választott szerkesztőben egy videót, amiben az osztályodat vagy egy aktuális iskolai eseményt mutatsz be! A forgatókönyvet és a szükséges anyagokat állítsd össze önállóan! Mentsd a videót **feladat 5.2.5.wmv** néven a mappádba!

## 6. SZÁMÚ GYAKORLATI MUNKA

### Videoklip készítése

**Figyelem!** A számítógép használata során tartsd be a balesetvédelmi szabályokat és az egészségügyi előírásokat!

Készíts a videoszerkesztővel egy hőmérőket bemutató 2,5 perces videót! Ehhez:

- Készítsd el a forgatókönyvet a következők figyelembevételével:
  - tartalmazzon egy címkockát a videó nevével, aminek a címe lehet például **Hőmérők**;
  - tartalmazza a hőmérők különböző típusainak bemutatását (4–6 kép) az **5fejezet\gyakorlati 6\foto** mappából! Minden képhez készíts feliratot a hőmérő nevével, ami megegyezik a fájl nevével! A képek bemutatása ne tartson 5 másodpercnél tovább;
  - Galilei hőmérőjének bemutatását az **5fejezet\gyakorlati 6\Foto** mappa 2–3 fotója és az **5fejezet\gyakorlati 6\video** mappában található videó alapján végezd! A fotókon helyezz el feliratokat, a videót kísérd narratív szöveggel! A feliratok és a kísérszöveg összeállítása során alkalmazd az **5fejezet\gyakorlati 6\Galileihőmérő.doc** fájl anyagait;
  - záró képkockára írd rá a saját neved, mint a forgatókönyvíró és filmrendező;
  - zenei aláfestést minden olyan kockához, amihez nem tartozik narratív szöveg, használd az **5fejezet\gyakorlati 6\zene** mappa fájljait;
  - válaszd ki a képkockák közötti átmenetet (legfeljebb két különbözőt)!
- Mentsd a videót **gyakorlati 6** néven a mappádba!


### 5.3. Multimédiás fájlok közzététele az interneten



- Mi a felhasználói fiók? Hogy hozunk létre felhasználói fiókot a postaszerveren?
- Hogyan lehet hozzáférést biztosítani a **Google Drive**-on lévő adatokhoz?
- Hogy adhatunk a filmhez háttérzenét a videoszerkesztőben?

A videók megosztásához és a barátaink számára történő könnyebb hozzáféréshez online közzétehetjük őket. Ez megtehető különféle felhőalapú tárolási szolgáltatások használatával, de használhatunk olyan szolgáltatásokat is, mint a **Vimeo** (<https://vimeo.com>), a **dailymotion** (<http://www.dailymotion.com>), **Ukrhomenet Video** (<http://>

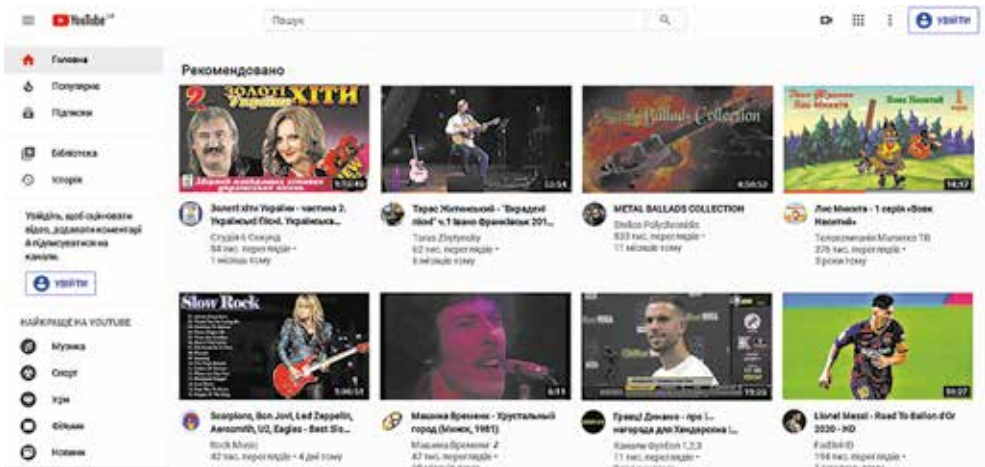
video.ukrhome.net), vagy megoszthatjuk a közösségi hálózatokon, mint például a **Facebook**.

A **YouTube**  (ang. *You Tube* – a te csatornád) az egyik legnépszerűbb videomegosztó szolgáltatás az interneten. Bármelyik **Gmail** felhasználó jogosult elhelyezni az anyagait a megosztón, és hozzáférési jogokat rendelni a feltöltött fájlokhoz.


### Saját YouTube-csatorna létrehozása videofájlok tárolásához

Ahhoz, hogy multimédiás fájlokat helyezz el a **YouTube**-on, a következőképpen kell eljárni:

1. Nyisd meg a szolgáltatás főoldalát (5.28. ábra) – <https://www.youtube.com/>!



5.28. ábra. A YouTube nyitóoldala

2. Az ablak bal oldalán kattints a **Bejelentkezés** gombra, majd add meg a **Google-fiókod** felhasználónevét és jelszavát!
3. A **YouTube** ablak jobb felső sarkában kattints a saját fíkdóhoz, például -re, és válaszd a **Csatorna létrehozása** gombot!
4. Az **Általános Szerződési Feltétel** ablakban kattintson a **Csatorna létrehozása** gombra!
5. A **Csatorna létrehozása** nézetben megadhatod, hogy milyen néven szeretnél publikálni a **YouTube**-csatornán!
6. Csatorna testreszabása részben végezd el a következőket:
  1. Állítsd be a profilképedet!
  2. Írd be csatorna adatait a **Csatorna személyre szabása** nézetbe – miről és kiknek fog szólni a videó!
  3. Szükség esetén add meg webhelyed, e-mailed és közösségi oldalad elérhetőségét!
  4. Kattints a **Mentés és folytatás** gombra, ha megadtad a szükséges adatokat, vagy a **Konfigurálás később** gombra, ha nem most kívánod megadni az adataidat!





A létrehozott saját **YouTube**-csatorna oldalelrendezését az 5.29. ábra mutatja. A továbbiakban lehetséges:

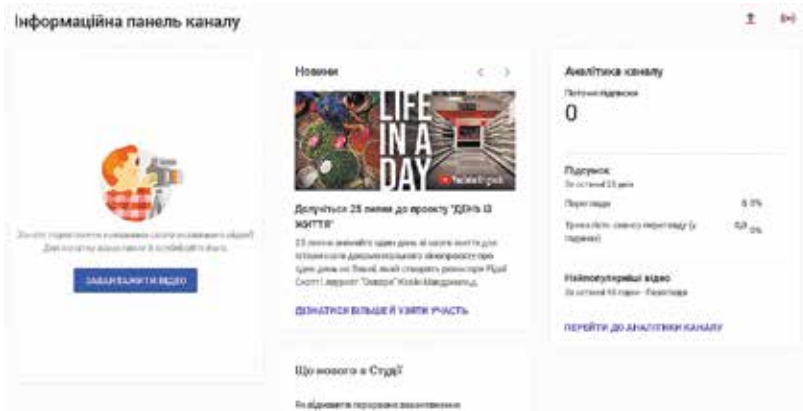
- a csatorna adatainak szerkesztése (**Csatorna testreszabása** gomb);
- megismerkedni a videomunkában segítséget nyújtó különféle videóanyagokkal, és felkeresni a **Csatorna irányítópultját** (**YouTube Alkotói Stúdió** gomb);
- videofájlokat feltölteni a saját csatornára (**Videó Feltöltése** gomb).



5.29. ábra. A létrehozott saját **YouTube**-csatorna kezdőoldala

A **YouTube Alkotói Stúdió** elsősorban a videók feldolgozására szolgál, valamint megismerkedhetünk különféle elemző anyagokkal, webhelyi hírekkel stb.


A **YouTube Alkotói Stúdió**nak **Csatorna irányítópultjában** a csatornára vonatkozó információk jelennek meg, beleértve a csatorna előfizetéséről, annak látogatásáról és egy bizonyos időtartamú videómegtekintésről szóló statisztikákat stb. (5.30. ábra).



5.30. ábra. **YouTube**-csatorna irányítópultja

### Videók feltöltése a **YouTube**-ra

Videofájl feltöltéséhez a csatornára kövesd az alábbi utasításokat:

1. Válaszd a **Videó feltöltése** gombot (**Csatorna irányítópultjában** vagy a főoldalon)!
2. **Videó feltöltése** nézetben válaszd a **Fájlok kiválasztása** vagy a  gombot!
3. Válaszd ki a kívánt fájlokat a számítógépről vagy a felhőben tároltakból, majd nyomd meg a **Megnyitás** gombot!
4. Hajtsd végre a három lépést a letöltött videó paramétereinek beállításához (5.31. ábra):



5.31. ábra. Az első lépés tulajdonságainak beállítása – **Részletek**

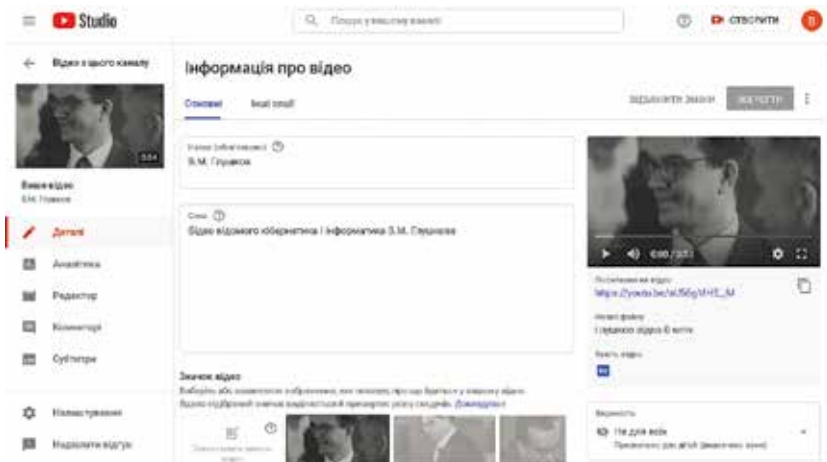
1. **Első lépés – Részletek.**
  1. Adjuk meg a videó címét a **Cím** mezőbe!
  2. A **Leírás** mezőbe adjunk meg egy áttekintő leírást a videónkról!
  3. Az **Indexkép** részen válasszunk vagy töltsünk fel képet, amely bemutatja, hogy mit tartalmaz a videó (ha indexképet szeretnénk feltölteni, válasszuk az **Indexkép feltöltése** lehetőséget)!
  4. Adjuk hozzá a videót egy témájában a videóhoz kapcsolódó **Lejátszási listához**!
  5. Válasszuk a közönség menü megfelelő pontját, amelyben megjelölhetjük, hogy a videó tartalma megfelel-e egy bizonyos korosztálynak (függetlenül attól, hogy ez a tartalom gyerekeknek szól-e vagy sem)! Ha szükséges, állítsunk be egyéb életkori korlátozásokat!
2. **Második lépés – A videoelemeket csak azoknál a videónál jelzik, amelyek nem gyermekek számára készültek.** Felnőttek számára hirdetéseket adhatunk hozzá saját videóikhoz.
3. **Harmadik lépés – Láthatóság.**
  1. A **Mentés vagy közzététel** csoportban válasszuk a választógombot, amely meghatározza a videónkhoz való hozzáférés korlátozásait: **Privát** (csak a mi és az általunk kiválasztott személyek nézhetik meg a videót), **Nem nyilvános** (korlátozott számú néző számára elérhető, akinek videólinket adunk), **Nyilvános**.
  2. **Ütemezés** – kiválaszthatjuk, hogy mikor legyen **Nyilvános** a videónk.
4. Szükség esetén másoljuk a videó címét, amely az előnézeti ablak alatt jelenik meg!
5. Nyomjuk meg a **Mentés** gombot!

### A YouTube-csatornán közzétett videók szerkesztése

Miután feltöltöttük a videófájlt, megtekinthetjük és szerkeszthetjük. Ehhez:

1. Nyissuk meg a **Lejátszási listát** a **YouTube Studio** ablak bal oldali ablaktáblájában!
2. Kattintsunk duplán a kívánt videó ikonra.

A megnyíló ablakban (5.32. ábra) megtekinthetjük és megváltoztathatjuk a videótokról korábban beírt adatokat, és a bal oldali ablaktábla linkszerkesztőjének kiválasztásával léphetünk a videoszerkesztés oldalra.



5.32. ábra. A YouTube Studio videoszerkesztési oldala

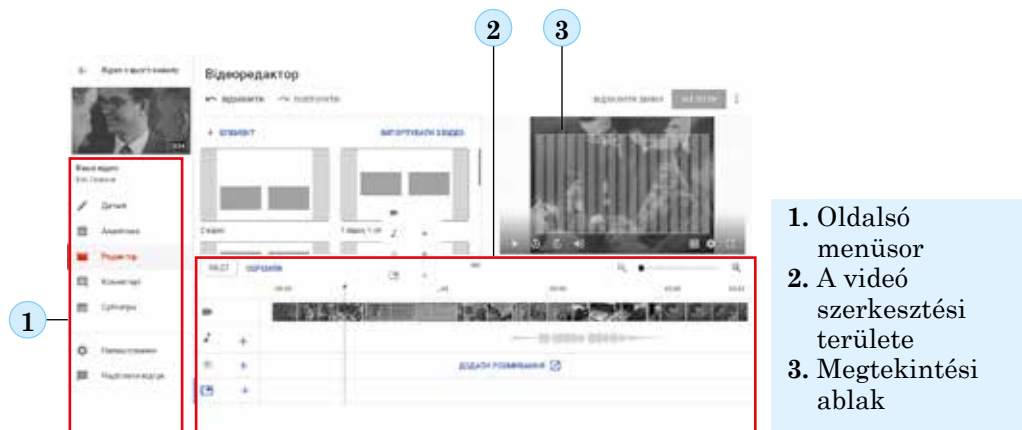


A **YouTube** videószerkesztő (5.33. ábra) lehetővé teszi számunkra, hogy a következő alapvető szerkesztési műveleteket hajthassuk végre:

- vágás, tetszőleges videórészletek törlése a videó bármely részéről;
- háttérzene hozzáadása a videóhoz;
- hozzáadhatunk a videóhoz linket, amely a csatornánkra vagy egy másik videónkra utal;
- elhomályosítás hozzáadása a videó bizonyos területeihez.

### A videó vágásához:

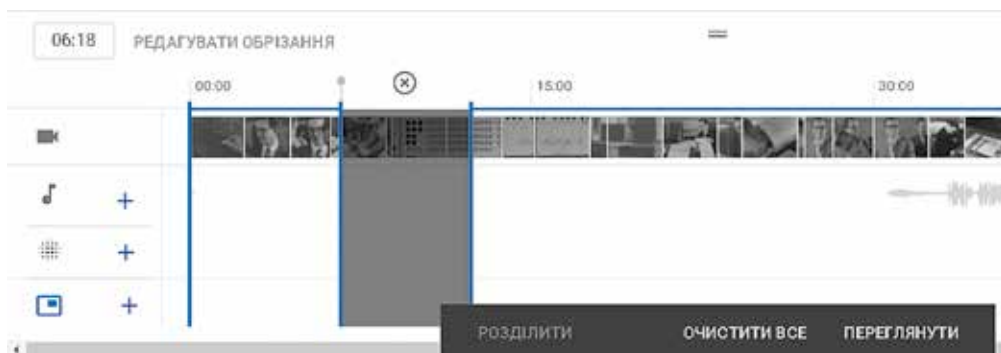
1. Válasszuk a **Vágás** lehetőséget!
2. Állítsuk be, hogy hol történjen a videó felosztása!
3. Válasszuk a **Felosztás** parancsot, amely a képernyőn a **Szerkesztő** alsó részében jelenik meg!



1. Oldalsó menü
2. A videó szerkesztési területe
3. Megtekintési ablak

5.33. ábra. A **YouTube Studio** videoszerkesztő nézete

4. Vigyük az egérmutatót a **Felosztás** parancs végrehajtása után kialakított függőleges vonalra (a mutatónak kétirányú nyíl formájában kell lennie ↔)!
  5. Húzzuk a függőleges vonalat balra vagy jobbra, amíg be nem fejezzük a vágást!
- A **Szerkesztési területen** megjelent a videón sötétszürkével kiemelt rész, amely a későbbiekben törlődik majd (5.34. ábra). A kijelölés visszavonásához az **Összes törlése** parancsra kell kattintanunk, vagy válasszuk a kiválasztott rész felett található **⊗** gombot!




5.34. ábra. A **YouTube Studio** videoszerkesztője vágás közben



Vágási módban több részletet is kiválaszthatunk, amelyeket a videó különböző részein vághatunk. Miután kiválasztottuk az összes törlendő területet, válasszuk a **Előnézet** parancsot a **Vágás** módból való kilépéshez, és nézzük meg, hogyan fog kinézni a videó a kivágott részek nélkül! Szükség esetén visszatérhetünk a Vágás módba a **Vágás szerkesztésére** kattintva.

A kiválasztott részek végső törlésére a **Mentés** parancs kiválasztása után kerül sor. A változtatások végrehajtása meglehetősen hosszú időt vehet igénybe, ez idő alatt a felhasználók hozzáférhetnek a videó változatlan verziójához.

A **YouTube Studio**nak csak egy lehetősége van a **hangadatok hozzáadására**, ez a háttérzene a **YouTube Alkotói Stúdió** részét képező dalokból. Háttérzene beszárahához kövessük az alábbi lépéseket:

1. Válasszuk a + gombot a videó szerkesztési területén, az audio sávától balra!
2. Válasszuk ki a kívánt dallamot a **YouTube** hangkönyvtárában (ha szükséges, hallgassuk meg a **Dal lejátszása**  gombra kattintva)!
3. Kattintsunk a **Hozzáadás** gombra!
4. Állítsuk be (ha a videó rendelkezik hangaláfestéssel) a video eredeti és a hozzáadott zene hangerejét (5.35. ábra)!
5. Válasszuk a **Változások mentése** gombot!

Bizonyos esetekben, például amikor megosztunk egy videót egy **YouTube**-csatornán, amelyen olyan személyek láthatók, akik nem egyeztek bele a videó megosztásába, vagy erőszakos jeleneteket tartalmaz, használjuk a kép elmosódása lehetőségét, ehhez:

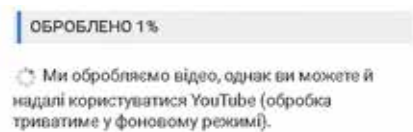
1. Válasszuk az **Elhomályosítás**  parancsot **ДОДАТИ РОЗМИВАННЯ**  !
2. A megnyíló ablakban (5.36. ábra) válasszuk az egyik elhomályosítási mód mellett szereplő **Szerkesztés** gombot, például az **Arcok elhomályosítása (Egyéni elhomályosítás)**!



5.35. ábra. Az eredeti és a beillesztett hang hangerőszabályozása



5.36. ábra. A YouTube Studio videoszerkesztője elhomályosítás módban



5.37. ábra. Videofeldolgozás homályosítási módban

3. Várjunk, amíg az alkalmazás videofeldolgozást végez (5.37. ábra), és elhomályosítási lehetőségeket kínál!
4. Az **Elhomályosítás** panelen válasszuk ki a videóban elmosni kívánt képeket (5.38. ábra)!
5. Nyomjuk meg a **Mentés** gombot!

A részletek egyéni elhomályosításakor állítsuk le a megtekintést annál a képnél, amit el szeretnénk homályosítani, és válasszuk ki a téglalap alakú elhomályosítási területet. Az alkalmazás feldolgozza a videót és felajánlja a képernyő elhomályosítási idejének friss paramétereit. A bal és jobb szegély húzásával önállóan megváltoztathatjuk az elmosódás kezdetének és befejezésének idejét.

### Videó feltöltése az internetre a Windows Live Movie Maker-rel

Amikor befejeztük a videó szerkesztését, a **Movie Maker** segítségével nemcsak elmenthetjük azt, de publikálhatjuk is az interneten. E célból nyissuk meg a projektet, majd:

1. Hajtsuk végre a **Fájl** ⇒ **Film közzététele** parancsot!
2. Válasszuk ki a szolgáltatók listájából azt (5.39. ábra), ahová el szeretnénk helyezni a fájlt, például a **YouTube**-ot!
3. A megnyíló párbeszédablakban adjuk meg a **YouTube** csatornánkhoz – való hozzáféréshez a felhasználói nevet és jelszót!
4. Adjuk meg a film nevét, leírását, hozzáférési beállításait, kategóriáját, tageit!
5. Kattintsunk a **Közzététel** gombra!
6. Várjuk meg a feltöltést és publikáljuk a videót!

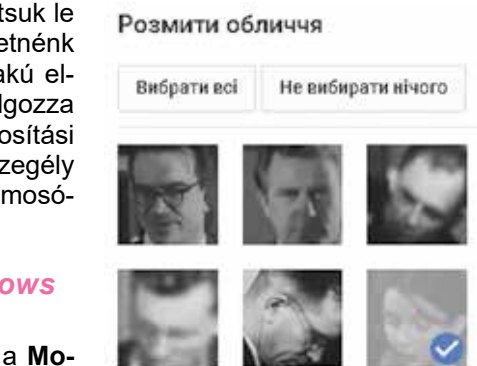
Hasonló algoritmust használnak a film más oldalakon történő közzétételére. A multimédiás adatok elhelyezésének előfeltétele a fiók létrehozása a szolgáltatás helyén.

### Hangfájlok közzététele az interneten

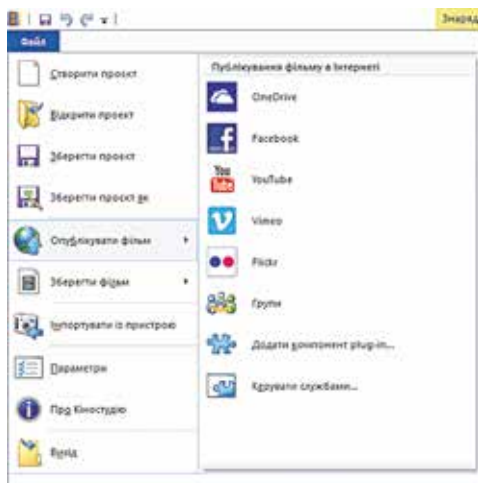
A hangfájlokot kezelő szolgáltatások döntő többsége azoknak lett szánva, akik saját zeneműveket hoznak létre. A másik irányzat azoknak szól, akik hangoskönyveket készítenek különböző – politikai, gazdasági, műszaki stb. – témákban.

A felsorolt hangfájlokat **podcast**oknak nevezik. A továbbiakban a *podcast* kifejezést használjuk a felhasználó által az interneten publikált hangfájlokra is.

A hangfájlok közzétételét támogató szolgáltatások a **Civil podcast csatorna** (<http://cpod.co/>), az **Ukrajnai podcast terminál** (<http://podcaster.org.ua>), a **Közzszolgálati rádió** (<https://soundcloud.com/hromads->



5.38. ábra. Arcképek az Elhomályosítási panelen



5.39. ábra. Szolgáltatók listája

A **Podcast** (ang. *ipod* – az Apple média-lejátszója; *broadcasting* – műsor-szórás) – olyan médiafájl (hang, videó és más fájlok), amelyeket a felhasználók (*podcasterek*) hoznak létre és teszik közzé sorozatszerűen az interneten.

ke-radio), a **BBC** médiavállalat oldala (<https://www.bbc.com/ukrainian/learning-english>) stb.

A **Google Drive** szolgáltatás segítségével audiofájlokat oszthatunk meg barátainkkal és ismerőseinkkel.



### Linux-használóknak

**Linux** alatt az **OpenShot** szerkesztő segítségével tehetünk közzé videókat az interneten. A program indítása után:

1. Hajtsuk végre a **Fájl** ⇒ **Videó betöltése** parancsot!
2. Jelöljük ki a feltöltendő videót, ehhez használjuk a **Videó feltöltése** ablakot (5.40. ábra)!
3. Válasszuk ki a szolgáltatót, ahová fel szeretnénk tölteni a fájlt (például a **YouTube**)!
4. Töltsük ki a megfelelő helyen a videó tulajdonságait tartalmazó mezőket (*név, leírás*)!
5. Adjuk meg a kiválasztott szolgáltatónál érvényes felhasználói nevünket és a jelszavunkat!
6. Kattintsunk a **Feltöltés** gombra!

Más szerkesztők esetében is a videók elhelyezésével kapcsolatos teendők hasonló a fentebb leírtakhoz.



5.40. ábra. A **Videó feltöltése** ablak



### Használjuk a számítógépet!

Végezzétek el a gyakorlati feladatot a <https://cutt.ly/AhdryjR> linkről, vagy a QR-kód segítségével!



### Összefoglalás

Az interneten sok olyan webhelyet találhatunk, amelyen multimédiás fájlokat tehetünk közzé és oszthatunk meg másokkal. A fájlok feltöltéséhez felhasználói hozzáférés szükséges.

A legelterjedtebb fájlmeosztó szolgáltatás a **YouTube** (<https://www.youtube.com>). Ide úgy tölthetünk fel videókat, ha létrehozunk egy csatornát, amely a felhasználó fájljait tartalmazza majd. A felhasználóknak lehetősége van arra, hogy szabályozzák mások hozzáférését a feltöltött fájljaikhoz.

A videoszerkesztők többsége támogatja a fájlok közzétételét a neten. Ilyenek például a **Windows Live Movie Maker** és az **OpenShot**.

Az internetre feltöltött multimédiás fájlokat **podcast**nak nevezik, a létrehozó felhasználókat pedig **podcaster**nek.



### Felelj a kérdésekre!



- 1°. Milyen szolgáltatóknál helyezhetünk el multimédiás tartalmakat az interneten?
- 2°. Hogy hozunk létre csatornát feltöltés céljából a **YouTube**-on?





- 3<sup>o</sup>. Milyen tulajdonságai vannak a **YouTube**-on elhelyezett videóknak?
- 4<sup>o</sup>. Milyen hozzáférési lehetőségeket biztosít a **YouTube**?
- 5<sup>o</sup>. Hogy rendelhetünk zenei háttérrel a **YouTube**-ra feltöltött fájlhoz?
- 6<sup>o</sup>. Hogy tehetünk közzé videókat a **Windows LiveMovie Maker** segítségével?
- 7<sup>o</sup>. Hogy töltünk fel hangfájlokat a **Google Drive**-ra?
- 8<sup>o</sup>. Hogy adhatunk hozzáférést a fájljainkhoz másoknak a **Google Drive**-on? Hogyan tájékoztathatjuk őket erről?

**Végezd el a feladatokat!**

- 1<sup>o</sup>. Helyezd el a **YouTube**-on az **5fejezet\5.3rész\video\szejm sztare ricsisce. mov** videót! Add meg a rövid leírást, zenei aláfestést és a tageket! Küldd el a tanárodnak a videó elérhetőségét!
-  2<sup>o</sup>. Töltsd fel a **YouTube**-ra a múlt foglalkozásokon **Dovzsenkóról** készült filmet a **Live Movie Maker** segítségével!
-  3<sup>o</sup>. Töltsd fel a **YouTube**-ra a tanárod által megadott fájlt (például az **5fejezet\5.3rész\video\csigák.mov**-ot)! Add meg a rövid leírást és a tageket! Keress az interneten információkat a csigákról, és helyezz el ezek alapján feliratokat a filmkockákon!
- 4<sup>o</sup>. Helyezd el a **Google Drive**-on a tanárod által megadott fájlt (például az **5fejezet\5.3rész\video\scsedrivocska.wma**-t)! Adj hozzáférést azoknak a felhasználóknak, akik megkapják a fájl linkjét! Küldd el a linket e-mailban egy barátodnak és a tanárodnak!

## 7. SZÁMÚ GYAKORLATI MUNKA

**Hangfájlok és filmek elhelyezése az interneten**

**Figyelem!** A számítógép használata során tartsd be a balesetvédelmi szabályokat és az egészségügyi előírásokat!

1. Töltsd fel a **YouTube**-ra a tanárod által megadott fájlt (például az **5fejezet\5.3rész\video\rovarok.wmv**-t)!
2. Állítsd be a következő tulajdonságokat:
  - név – *Rovarok*;
  - leírás – *Ukrajna központi régiójának rovarvilága*;
  - tagek – *természet, rovarok, szarvasbogár*;
  - hozzáférés szintje – *privát*;
  - vázlat – *szarvasbogár*;
  - feliratok – *00.01.0-tól 00.09.0-ig nagy szarvasbogár (Lucanus cervus) – a szarvasbogárfélék legtermetesebb képviselője*;
  - zene – stílus *coutry* és *folk*, név – *As We Go*.
3. Küldd el a linket e-mailban a tanárodnak!

# 6. fejezet

## Algoritmusok és programok

*Ebben a fejezetben megismerkedtek:*

- a programozási nyelvekkel és azok alkotóelemeivel
- a program és a projekt fogalmával
- a **Lazarus** és **IDLE** programozási környezettel
- **Pascal** és **Python** programozási nyelven létrehozott projektekkel
- projektek beállításaival
- az objektumokkal és azok tulajdonságaival
- az eseménykezelőkkel
- a **form**, a **gomb**, **címke**, a **vezérlőelemekkel**, **jelölőnégyzetekkel** és **kapcsolókkal**, azok tulajdonságaival, eseményeivel
- az elemek tulajdonságainak megváltoztatásával a program futtatása előtt és után
- a változók és állandók tulajdonságaival
- lineáris, elágazásos és ciklusos projektekkel

### 6.1. Számítógépes programok és programozási nyelvek



1. Mi az algoritmus? Mi az algoritmus-végrehajtó utasításkészlete?
2. Mi a számítógépes program?
3. Milyen programozási nyelvekkel ismerkedtetek meg régebben? Idézzétek fel ennek néhány utasítását!

#### *A számítógépes programok*

Már tudjátok, hogy a számítógép a szoftver vezérlése alatt működik, amit a különböző rendeltetésű programok alkotnak. Otthon vagy az iskolában használva a számítógépet, azon szövegszerkesztőt, képszerkesztőt, tömörítőprogramokat, táblázatkezelőt, prezentációkészítőt, oktatási és ellenőrző szoftvereket, játékokat futtathattok.

Azt is tudjátok, hogy a **számítógépes program** egy adatfeldolgozó algoritmus, ami a számítógép számára értelmezhető nyelven van leírva.

Működésük során a programok **adatokat** dolgoznak fel.

Azok az adatok, amelyeket a program a bemeneti perifériától (billentyűzet, egér, érzékelők), vagy más forrásból (például egy szöveges állományból) megkap, **bemeneti adatok**. Vannak programok, amelyek bemeneti adatok nélkül dolgoznak.

A program működése során az adatok feldolgozásra kerülnek, új adatok jönnek létre. Ezek a **köztes adatok**.



Ha a program célja bizonyos adatok előállítására, ezeket **kimeneti adatoknak** nevezzük.

A mai programok többsége működése során bizonyos lehetőséget biztosít a felhasználók részére, hogy beavatkozassanak a program és az eszközök működésébe. Ilyenek a gombok, menük, beviteli mezők, számlálók, címkék stb. Ezeknek az eszközöknek az összessége alkotja a **felhasználói felületet**.

A modern számítógépes programokat gyakran **számítógépes projekteknek** nevezzük.

## Programozási nyelvek

Amikor emberek számára állítunk össze algoritmust, emberi nyelveket használunk: ukránt, angol, orosz, magyar, németet stb.

A számítógép részére összeállított algoritmus esetében nem használhatjuk a bonyolult és nem mindig egyértelmű (homónímiák) emberi nyelveket. Ezért a számítógépes algoritmusokat speciális nyelveken, úgynevezett programozási nyelveken írják.

Azt a nyelvet, amelynek rendeltetése, hogy a számítógépnek értelmezhető algoritmusokat fogalmazzunk meg, **programozási nyelvnek** nevezzük.

Az utóbbi 70 évben nagyjából 3000 programozási nyelvet hoztak létre. Egyeseket ma már nem használják, másoknak új, kézenfekvőbb változatai jelentek meg. Időről időre új programozási nyelvek is megjelennek.

Vannak olyan **univerzális** nyelvek, amelyek a tudományos-műszaki, termelési és mindennapi problémák széles körében alkalmazhatóak, míg másokat **speciális** problémák megoldása céljából hoztak létre.

2020-ban a legtöbb program **Java, JavaScript, C# (C Sharp), PHP, Python, C++** nyelveken íródtak.

A programozási nyelvek **mesterséges nyelvek**. A kémiai reakciók leírásának nyelve, a matematikai egyenletek és egyenlőtlenségek nyelve, a postai irányítószámok nyelve, a sakkjátszmák leírásának nyelve szintén mesterséges nyelvek.

Egy nyelv a következő elemeket tartalmazza:

- **Ábécé** – azok a szimbólumok, amelyeket a nyelv szavainak és mondatainak lejegyzésére használunk.
- **Szótár** – a nyelvben használható szavak jegyzéke.
- **Szintakszis** – a nyelvi szerkezetek megalkotásának szabályai.
- **Szemantika** – a nyelvi szerkezetek értelmezése, alkalmazásának szabályai.

Az 5–7. osztályban a **Scratch2** programozási nyelvet tanultátok. Idézzük fel: ennek a nyelvnek az ábécéje az angol és a magyar nyelv betűiből, a **:**, **=**, **+**, **\***, **<** stb. jelekből áll. A szótár tartalmazza az **ismétlés**, **mozgás**, **ha**, **amikor**, **pattanj**, **vissza** utasításokat. A nyelv szintakszisa előírja, hogy az előtesztelő ciklus az **ismételd**, **amíg** szavakkal kezdődik. A **lépj 10 lépést** utasítás az algoritmus végrehajtóját 10 lépéssel mozdítja el az állásának megfelelő irányba.

A meg nem engedett szimbólumok használata, a szótár szavainak nem pontos leírása, a szintaktikai szabályok megsértése értelmezhetetlenné teszi az algoritmust a számítógép számára. Az ilyen hibákat **szintaktikai hibáknak** nevezzük.



## A Pascalt használóknak

A 8. és 9. osztályban a **Pascal** (objektumorientált pascal) nyelvvel ismerkedünk majd meg. A nyelv alkalmas arra, hogy a tanulók és hallgatók megismerkedjenek a programozás alapjaival, de arra is, hogy professzionális programozók különféle programokat hozzanak létre a segítségével.



### Érdekes történelmi tények



6.1. ábra.  
Nicolaus Wirth

A **Pascal** programozási nyelvet a svájci műszaki egyetem szakemberei hozták létre 1970-ben **Nicolaus Wirth** (szül. 1934-ben, lásd 6.1. ábra) professzor vezetésével. A nyelvet a híres francia matematikus, fizikus, irodalmár és filozófus, Blaise Pascal tiszteletére nevezték el.

A **Pascal** nyelvet kifejezetten oktatási céllal hozták létre, hogy a diákokat a programozási eljárások alkalmazására tanítsák. Idővel kiderült, hogy a programozási nyelv használata elég kényelmes, ezért elkezdtek gyakorlati problémák megoldására is alkalmazni. Megjelentek az első compilerek, amelyek a **Pascal** nyelven írott programokat a számítógép által végrehajtható programokká fordítják, később pedig a **Turbo Pascal** programozási környezet, ami nagy népszerűsége tette szert.

Az évek során a nyelvet átalakították, kibővítették, tökéletesítették, így jött létre a XX. sz. 80-as éveinek közepén a **Pascal objektum-orientált** változata, az **Object Pascal**. A 2000-es évek elején megjelent az **Object Pascal** nyelvre támaszkodó **Delphi** programozási környezet.

### A projektek programozási környezete

A **Pascal** programozási nyelven írt programok létrehozására, szerkesztésére, tesztelésére és javítására a **Lazarus programozási környezetet** fogjuk használni.


A programozási környezet a következő részekből áll:

- **szövegszerkesztő** a program szövegének beírására és szerkesztésére;
- **compiler**, ami a **Pascal** nyelven írt programot lefordítja a számítógép számára értelmezhető utasításokra;
- **debugger**, ami a hibák keresésében és javításában lesz segítségünkre;
- **súgó**.

**Lazarusban** a fejlesztés során rendelkezésünkre áll a **vizuális komponensek** könyvtára (**Lazarus Component Library** – **Lazarus** komponenskönyvtár). Ezek tartalmazzák a számotokra már jól ismert gombokat, mezőket, címkéket, jelölőnégyzeteket, kapcsolókat, számlálókat, listákat, gördítősávokat stb. Amikor ezeket a komponenseket elhelyezzük a formon, automatikusan létrejön a megfelelő **Pascal**-kód, amit nem kell begépelni, ezáltal sokkal könnyebb és gyorsabb a programok létrehozása. A programozásnak ez a módszere a **vizuális** programozás.

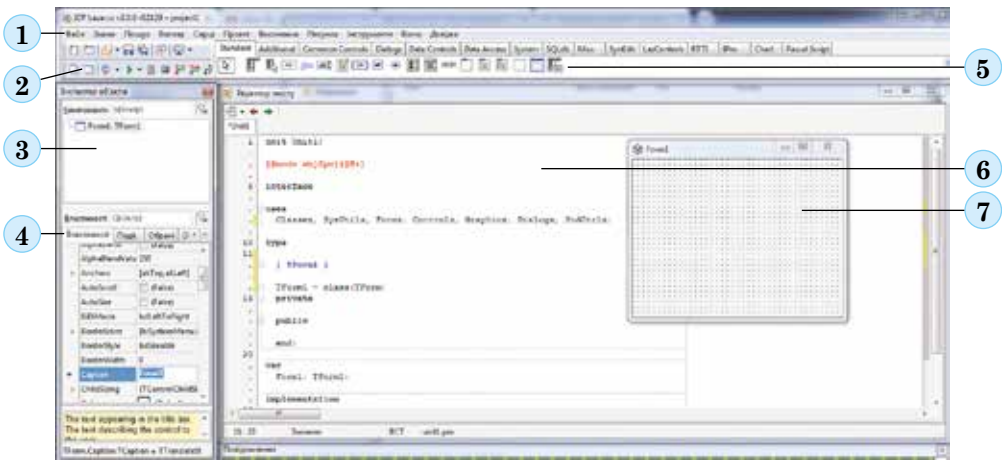


A **Lazarus** a következőképpen indíthatjuk:

- a **Start** menü **Lazarus** parancsal;
- a munkaasztal  parancsikonyájával.

A program indulásakor megnyílik a **Lazarus** rendszer ablaka (6.2. ábra), ami tartalmazza a **Főmenüt** (6.2. 1. ábra), az eszköztárakat (6.2. 2. ábra), a vezérlőelemek lapjait (6.2. 5. ábra) és néhány alárendelt ablakot, amelyek között megtaláljuk:


- az **Objektum felügyelőt** (6.2. 3. ábra), amelynek segítségével beállíthatjuk az objektumok tulajdonságait, elkészíthetjük az eseményvezérlőket (6.2. 4. ábra) stb.;
- **Szövegszerkesztőt** (6.2. 6. ábra), ahová beírhatjuk a **Pascal** nyelven írt program szövegét;
- a **Formot** (6.2. 7. ábra), ahol el fogjuk helyezni a program vezérlőelemeit.



- |   |                            |
|---|----------------------------|
| 1 – Főmenü                                | 5 – Vezérlőelemek          |
| 2 – Eszköztárak                           | 6 – Szövegszerkesztő ablak |
| 3 – Objektum felügyelő ablak              | 7 – Form ablak             |
| 4 – Az Objektum felügyelő eseményvezérlői |                            |

6.2. ábra. A **Lazarus** környezet az alárendelt ablakokkal

### Projektek mentése

A **Lazarus** környezetben készített projekt több fájlból áll. Ezért minden projektet célszerű külön mappába menteni. A mentés a **Fájl** menü **Mindent menti**  vagy a **Projekt** menü **Projekt mentése** parancssal történhet.

A rendszer egymás után a következő fájlokat ajánlja:

- mentésre: **Project1** (ang. *project* – terv) nevű **lpi** kiterjesztésű fájlt;
- az **Unit1** (ang. *unit* – blokk) nevű **pas** kiterjesztésű fájlt.

A fájlok neveit megváltoztathatjuk, a kiterjesztéseiket azonban nem.

A projekt valamennyi fájlja a megadott mappába történik (6.3. ábra).

Ha a már mentett fájlt megváltoztatjuk, az újbóli mentés ugyanabba a mappába történik majd, a mentés során nem nyílnak meg újabb ablakok.

A mentett projektet elmenthetjük másik mappába is, ezt a **Projekt** menü **Projekt mentése másként** paranccsal célszerű megtenni. Másolat mentése során meg lehet változtatni a fájlok neveit is.

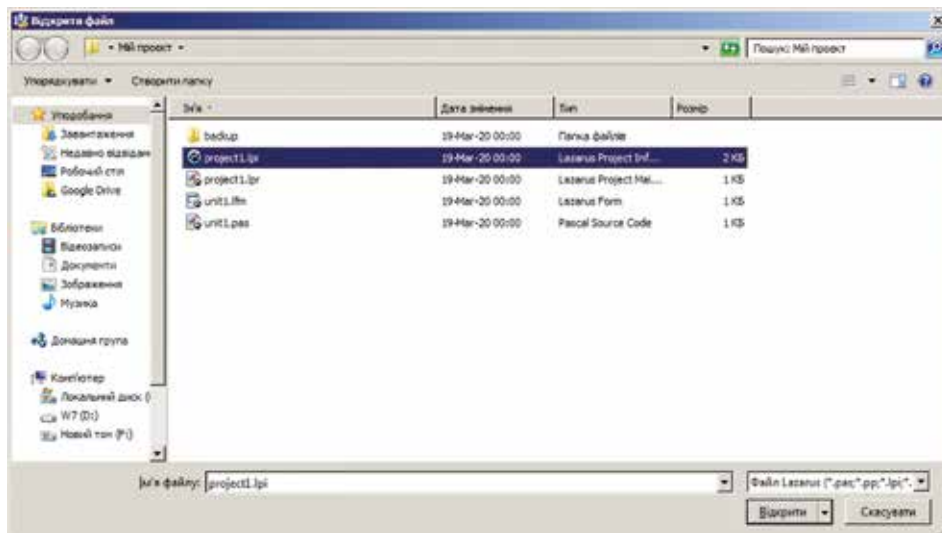
|              |                  |                        |        |
|--------------|------------------|------------------------|--------|
| backup       | 18.03.2020 23:19 | Папка с файлами        |        |
| project1     | 18.03.2020 23:18 | Значок                 | 134 КБ |
| project1     | 18.03.2020 23:19 | Lazarus Project Inf... | 2 КБ   |
| project1     | 18.03.2020 23:19 | Lazarus Project M...   | 1 КБ   |
| project1.lps | 18.03.2020 23:19 | Файл "LPS"             | 2 КБ   |
| project1.res | 18.03.2020 23:19 | Файл "RES"             | 136 КБ |
| unit1        | 18.03.2020 23:19 | Lazarus Form           | 1 КБ   |
| unit1        | 18.03.2020 23:19 | Pascal Source Code     | 1 КБ   |

6.3. ábra. A projekt fájlijait tartalmazó mappa

### Mentett projekt megnyitása

A mentett projektet **Fájl Megnyitás** paranccsal, az eszköztár **Megnyitás** gombjára kattintva, vagy a **Projekt** menü **Projekt megnyitás** paranccsal nyithatjuk meg. A megnyíló ablakban válasszuk ki az **Ipi** kiterjesztésű fájlt, majd kattintsunk az **OK** gombra (6.4. ábra)!

A projekt megnyitásának másik módja a projektfájl mappájának megnyitása az **Ipi** kiterjesztésű fájlra való kétszeri kattintás után. Ez automatikusan megnyitja a **Lazarus** környezetet és megnyitja a projektfájlokat.



6.4. ábra. A projekt megnyitása ablak


### Projektek futtatása

Mint már említettük, a modern vizuális projektfejlesztési környezetek, amelyek közül az egyik a **Lazarus**, részben automatizálják létrehozásuk folyamatát, megkönnyítve és felgyorsítva mind a szakemberek, mind a kezdők munkáját. Még abban az esetben






is, amikor új projektet nyitunk meg, amikor a projektszöveg sora nincs megadva, a **Lazarus** automatikusan létrehoz (generál) egy bizonyos kezdeti projektszöveget. Ez a projektszöveg elegendő a futtatásához.

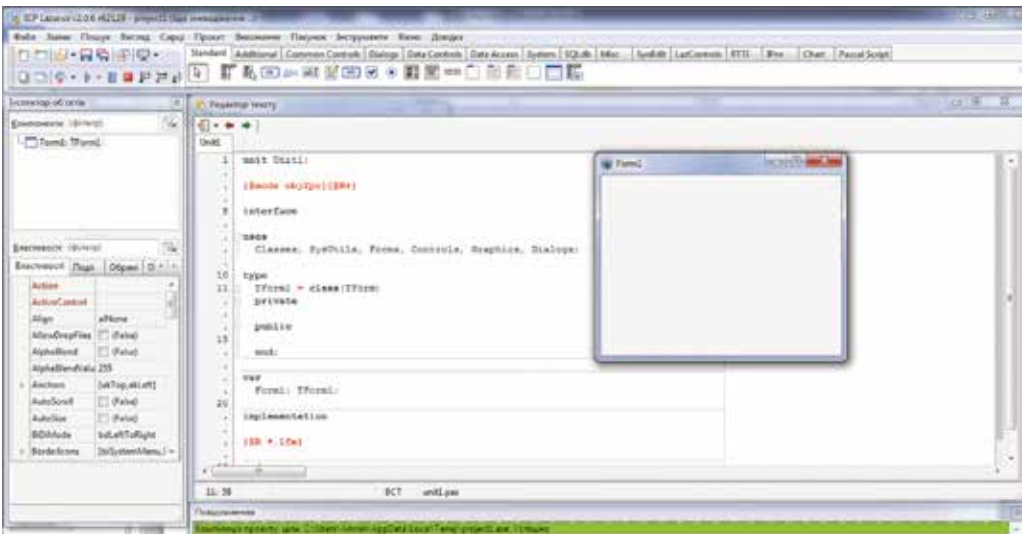
A projekt futtatását a **Futtatás** menü **Futtatás** paranccsal, az eszköztár **Végrehajtás**  gombjával vagy az **F9** funkcióbillentyű leütésével kezdeményezhetjük.

Amikor a projektet futtatjuk, először is a **compiler** hajtja végre a munkáját. Először elvégzi a program szövegének szintaktikai elemzését. Ha hibát talál, megszakítja munkáját és megmutatja az első megtalált hibát. A hiba kijavítása után újból el kell indítani a futtatást. Ha nem talál hibát, akkor átalakítja a **Pascal** nyelvű programot a számítógép számára értelmezhető utasításokká, majd átadja azt a processzornak végrehajtás céljából.

**Compiler** (ang. *compiler* – az aki össze- gyűjt) – számítógépes program, amely egy bizonyos programozási nyelven írt programot konvertál (fordít) egy másik programozási nyelv egyenértékű kódjává, amelyet egy számítógép végrehajthat.

Az alapértelmezett projekt futtatásakor a képernyőn megjelenik egy ablak, amelynek beállításai (szín, méretek, elhelyezkedés, cím stb.) megegyezik az alapértelmezett értékekkel (6.5. ábra).

Az **Ablak bezárása**  gombra kattintva vagy az **Alt + F4** billentyűkombinációt leütve leállíthatjuk a projekt futását.



6.5. ábra. Az alapértelmezetten létrejövő projekt és futtatásának eredménye

A **Lazarus** környezet ablak bezárásához választhatjuk a **Fájl** menü **Kilépes** parancsot, vagy a **Lazarus** ablak címsorában a környezet ablak bezárása gombot.

## Projektek ablakkal. A form és tulajdonságai

A **form** az az objektum, ahol el fogjuk helyezni a program vezérlőelemeit, például a gombokat, címkéket, menüket, gördítősávokat. A program futtatása során a formból lesz a **programablak**, ami tartalmazza az elhelyezett vezérlőket.

A formnak vannak **tulajdonságai**, ezeket az **Objektum felügyelő** ablakában (6.2. 4. ábra) tekinthetünk meg. Az ablak bal oldali része tartalmazza a tulajdonságokat, a jobb oldali pedig azok értékeit. A tulajdonságok egy részét beírhatjuk a billentyűzetről, másokat a legördülő listákból választhatjuk ki.

Lássunk néhány tulajdonságot:

- **Name** (*név*) – a formnak, mint objektumnak neve;
- **Caption** (*cím*) – ez jelenik meg az ablak címsorában;
- **Color** (*szín*) – a form háttérszíne;
- **Height** (*magasság*) – az ablak magassága pixelekben (a **pixel** a képernyő grafikus képének egy pontja);
- **Width** (*szélesség*) – a form szélessége pixelekben;
- **Top** (*teteje*) – az ablak bal felső sarkának pixelben mért távolsága a képernyő tetejétől;
- **Left** (*bal*) – az ablak bal felső sarkának pixelben mért távolsága a képernyő jobb szélétől.

A **Top** és **Left** tulajdonságokat megváltoztathatjuk a form képernyőn történő mozgásával, a **Height** és **Width** tulajdonságokat pedig a form szélein található markerek mozgásával.

Módosíthatjuk az eredeti projektet, valamint bármely más projektet. Például az **Objektum felügyelő** ablak **Tulajdonságok** lapján a következő form tulajdonságértékeket állíthatjuk be:

**Caption** – *Első ablak*;

**Color** – *clRed* (*color* – szín, **red** – vörös);

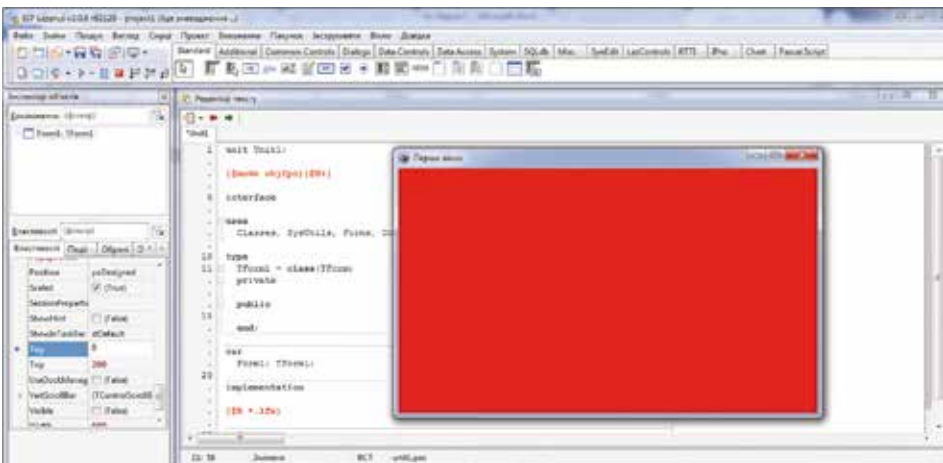
**Height** – *350*;

**Width** – *600*;

**Left** – *550*;

**Top** – *200*.

Ha elmentjük a projekt új verzióját, és újra futtatjuk, akkor egy másik színű és méretű ablak is megnyílik a képernyőn, valamint a címsor más szöveget fog tartalmazni (6.6. ábra).



6.6. ábra. Egy megváltoztatott projekt eredménye



## A Pythont használóknak

A 8. és 9. osztályokban megtanuljátok a **Python** programozási nyelvet. Ezt a nyelvet széles körben használják a különböző területeken: megtanítják a diákokat a programozás alapelveire, a tudományos számításokra, a grafikus adatok megjelenítésére, a háromdimenziós képek feldolgozására, a hálózatok és az operációs rendszerek adminisztrációjára, a kész programok képességeinek bővítésére, a saját alkalmazások létrehozására és még sok másra.



### Érdekes történelmi tények

A **Python** nyelvet először 1991-ben Guido van Rossum (szül. 1956-ban, 6.7. ábra) holland programozó alkotta meg, aki annak idején a Holland Nemzeti Matematika és Informatikai Kutatóintézet munkatársa volt.

A névadás nem a hullóktól ered. A szerző a XX. sz. 70-es éveiben népszerű *Monty Python repülő cirkusza* című vígjátéksorozat szereplőjéről nevezte el. A projekt létrehozásának környezete az **IDLE** nevet kapta, ahol a **Python** nyelv programjai íródnak. Ezt a környezetet szintén a sorozat egy szereplőjéről, Eric Idle-ről nevezte el.



6.7. ábra. Guido van Rossum

### A projektek létrehozásának környezete

Projektek **Python** programozási nyelven történő létrehozásához, szerkesztéséhez és hibakereséséhez az **IDLE 3.7 projektfejlesztési környezetet** (ang. *Integrated Development and Learning Environment* – Integrált fejlesztési és tanulási környezet) fogjuk használni.

Letölthetjük számítógépünkre a **Python** legújabb verzióját, az **IDLE fejlesztői környezetet** ingyenes telepítéséhez használjuk a **Python Software Foundation** (ang. *Python Software Foundation*) hivatalos webhelyét ([python.org](http://python.org)).


Az **IDLE** projektfejlesztési környezet a következőkből áll:

- **szövegszerkesztő** a program szövegének beírására és szerkesztésére;
- **compiler**, ami a **Pascal** nyelven írt programot lefordítja a számítógép számára értelmezhető utasításokra;
- **debugger**, ami a hibák keresésében és javításában lesz segítségünkre;
- **súgó**

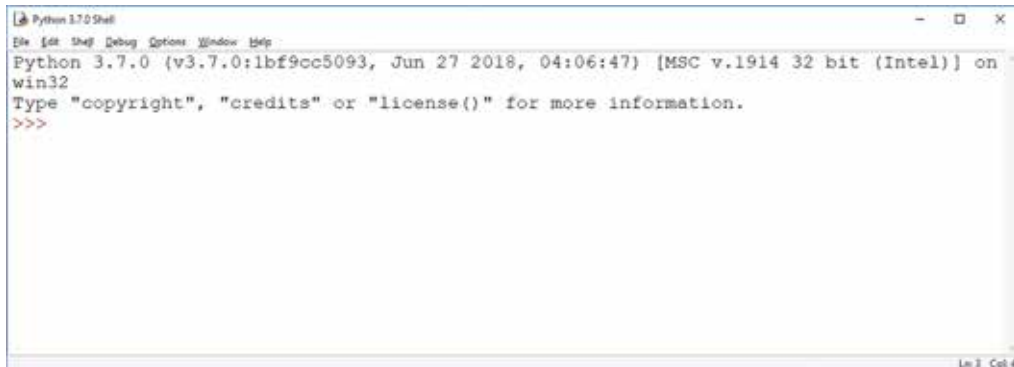
és egyéb vezérlők.

Ez a fejlesztői környezet lehetővé teszi nagy **modulkönyvtár** – kész programkóddarabok új projektekhez csatolását – használatát, ami egyszerűsíti és felgyorsítja azok létrehozását.

Az **IDLE** ablak megnyitható:

- a **Start** menü **IDLE (Python 3.7)** paranccsal;
- a munkaasztalon lévő  parancsikkal.

Ennek eredményeként megnyílik az **IDLE** projektfejlesztési és végrehajtási környezet **Python 3.7.0 Shell** ablaka (6.8. ábra), amely a főmenüt és a munkaterületet tartalmazza.



6.8. ábra. Az **IDLE** projektfejlesztési és végrehajtási környezet **Python 3.7.0 Shell** ablaka

A munkaterületen a `>>>` karakterek azt a helyet jelölik, amelyben a parancsokat a **Python** programozási nyelven adhatjuk meg. A parancs beírása az **Enter** billentyű lenyomásával fejeződik be, ezután a parancs végrehajtásra kerül, és a végrehajtás eredménye a parancs után a következő sorban jelenik meg. A parancsok bevitelének és végrehajtásának ezt a módját **interaktív**nak nevezik.

Az interaktív mód kényelmes a számításokhoz, a parancsok helyességének ellenőrzéséhez és a legegyszerűbb projektek végrehajtásához. Bonyolultabb projektek létrehozásához tanácsos a projektparancsokat külön fájlba írni, amely lehetővé teszi a projekt ismételt végrehajtását anélkül, hogy újra megadnánk az egyes parancsokat.

A **File** ⇒ **New File** paranccsal létrehozhatunk egy új ablakot a projekt szövegének megadásához. Ez megnyitja a szövegszerkesztő ablakot. A **Untitled** (cím nélküli) ablak neve lecserélésre kerül annak a fájlnak a nevére, amelyben a projekt szövege a **File** ⇒ **Save** parancs végrehajtása után az adathordozóra kerül. A fájlmentés ablakban válasszuk ki a mappát, ahova menteni szeretnénk, és írjuk be a fájl nevét! A fájl a **py** kiterjesztést kapja. A korábban létrehozott projekt újbóli mentésekor nem kell fájlnevet megadni.

A szövegszerkesztő ablakban megadott parancsok nem kerülnek azonnal végrehajtásra, és az **Enter** billentyű lenyomásával a kurzor új sorba kerül, ahol megadhatjuk a következő parancsot.

A projekt futtatásához el kell menteni a fájlt, majd futtassuk a **Run** ⇒ **Run Module** paranccsal vagy nyomjuk meg az **F5** billentyűt!

**Interpretátor** (ang. *interpreter* – tolmács, fordító) – olyan számítógépes program, amely a parancsokat gépi kóddá alakítja és végrehajtja azokat az utasításokat, amelyek valamilyen programozási nyelven íródtak


A projekt végrehajtásának megkezdése után az **interpretátor** megkezd munkáját. Ez a program felváltva elemzi a projektparancsokat és hibákat keres. Ha a parancsban nincs hibája, az értelmező lefordítja a **Pythonból** a számítógép processzorának végrehajtható parancsok nyelvére, és továbbítja a lefordított parancsot a

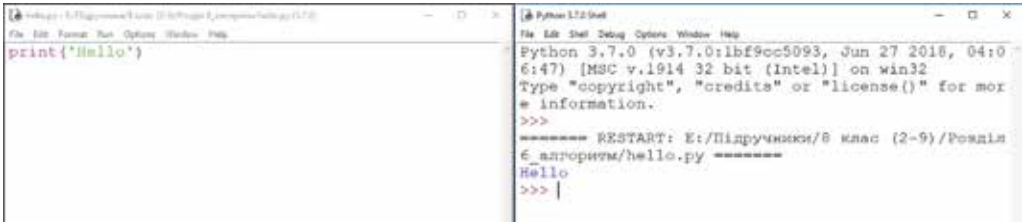
processzornak. Ezt követi a következő parancs elemzése és végrehajtása. Ha vannak szintaktikai hibák, az interpretátor megszakítja a projektet, és megjelöli az első hiba helyét. Javítanunk kell a hibát, majd újra kell futtatni a projektet.



A projekt eredményei és a hibaüzenetek a **Python 3.7.0 Shell** projekt fejlesztése és végrehajtása ablakban jelennek meg.

Projekt kidolgozása során a szövegszerkesztő ablak és a **Python 3.7.0 Shell** ablak kényelmesen elhelyezhetők a képernyőn egymás mellett (6.9. ábra). Ehhez:

1. Tegyük aktuálissá a szövegszerkesztő ablakot!
2. Nyomjuk meg a  + ← billentyűkombinációt!
3. Válasszuk ki a **Python 3.7.0 Shell** fejlesztési ablakát a képernyő jobb felében!



6.9. ábra. A szövegszerkesztő ablak – baloldalt,  
Python 3.7.0 Shell ablak – jobboldalt

A **py** kiterjesztésű fájlba mentett, korábban létrehozott projekt megnyitható, szerkeszthető és futtatható. A projekt szövegével rendelkező fájl megnyitásához hajtsuk végre a **File** ⇒ **Open** parancsot az **IDLE** ablakban, és válasszuk ki a kívánt fájlt.

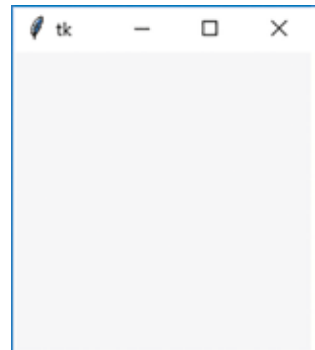
### Projektek ablakkal

A **Python** ablakkal rendelkező projekt létrehozásához csatolni kell a **tkinter** modult a projekthez, amely a **Python** modulok standard könyvtárának része. A modul a következő parancs végrehajtásával kapcsolódik a projekthez: **from tkinter import\*** (ang. *from* – valahonnan, *import* – behozni).

Ablak létrehozásához adjunk neki nevet és hajtsuk végre a következő parancsot: **<ablak\_neve> = Tk()**! A projekt főablakát gyakran *root*-nak nevezik (ang. *root* – gyökér). Ezután a parancsokban ezt a nevet fogjuk használni az ablakhoz.

A *root* nevű ablak létrehozásának parancsa: **root = Tk()**.

A parancs végrehajtása után megnyílik egy ablak, amelynek megjelenését a 6.10. ábra mutatja. Az alapértelmezett ablak a képernyő bal felső sarkában nyílik meg, mérete 200 × 200 pixel (**pixel** – a képernyőn megjelenő grafikus kép pontja), az ablak címsorában található a *tk* felirat, a háttér színe pedig világosszürke.



6.10. ábra. Az alapértelmezett értékekkel rendelkező ablak

Az összes ablak tulajdonsága megváltoztatható.

A *root* nevű ablak címsor feliratának módosításához futtassuk a **root.title('<szöveg>')** parancsot! Például a **root.title('Az első projektem')** parancs megjeleníti az ablak címsorában *Az első projektem* feliratot.

Az ablak méretének és helyzetének megváltoztatását a **root.geometry** ('<parameter sor>') parancs hajtja végre. A paraméterek sorában megadhatjuk: az ablak szélességét és magasságát, amely értékek között „x” szerepel, valamint az ablak bal szélének a képernyő bal szélétől való távolságát és a képernyő felső széle és az ablak felső széle közötti távolságot. Előtte „+”-ot teszünk. Így a **root.geometry parancs ('600x400+200+100')** az ablak szélességét 600 képpontra, az ablak magasságát 400 képpontra, az ablak bal szélének a képernyő bal szélétől való távolságát 200 pixelre, az ablak felső határának távolságát a felső képernyőtől 100 pixelre állítja be.

*Figyeljünk arra, hogy a paraméter értékeinek megadásakor ne használjunk szóközt!*

A **root.geometry('400x250')** paranccsal az ablak szélesség 400 pixelre, az ablak magassága pedig 250 pixelre lesz állítva, és nem változtatja meg az ablak elhelyezkedését a képernyőn. Ha pedig a **root.geometry('+300+200')** parancsot futtatjuk, akkor az ablak mérete nem változik, csak a képernyő bal oldalától és tetejétől számított távolság módosul.

Az ablak háttérszínének megváltoztatásához meg kell adni a szín nevét a **['bg']='<color>** (ang. *background* – háttér) parancsban. Például a **root['bg']='yellow'** parancs futtatásakor az ablak háttérszíne sárgára változik. A 6.11. ábrán a parancsok végrehajtása utáni formája látható az ablaknak:



6.11. ábra. Az ablak megjelenése a Címsor feliratának, méretének, helyzetének és a háttérszínének megváltoztatása után

Az ablak háttérszínének megváltoztatásához meg kell adni a szín nevét a **['bg']='<color>** (ang. *background* – háttér) parancsban. Például a **root['bg']='yellow'** parancs futtatásakor az ablak háttérszíne sárgára változik. A 6.11. ábrán a parancsok végrehajtása utáni formája látható az ablaknak:

```
from tkinter import *
root=Tk()
root.title('Az első projektem')
root.geometry('600x400+200+100')
root['bg']='yellow'
```

### Használjuk a számítógépet!

Végezzétek el a gyakorlati feladatot a <https://cutt.ly/hhdrubZ> linkről, vagy a QR-kód segítségével!



### Összefoglalás

A **számítógépes program** a számítógép által értelmezhető nyelven megírt adatfeldolgozó algoritmus.

A számítógép **adatok** feldolgozását végzi.

A külső eszközök (billentyűzet, egér, érzékelők), vagy más források (például szöveges fájl, vagy másik program) által szolgáltatott adatok a **bemeneti adatok**. Vannak programok, amelyek nem használnak bemeneti adatokat.

A program futása során az adatok feldolgozásra kerülnek és **köztes adatok** jöhetnek létre.

Amennyiben a program célja valamilyen adatok előállítása, ezek lesznek a program **kimenő adatai**.

Azt a nyelvet, amelyen a számítógép számára értelmezhetően fogalmazunk meg algoritmusokat, **programozási nyelvnek** nevezzük.

A programnyelvek összetevői:





- **Ábécé** – azok a szimbólumok, amelyeket a nyelv szavainak és mondatainak lejegyzésére használunk.
- **Szótár** – a nyelvben használható szavak jegyzéke.
- **Szintakszis** – a nyelvi szerkezetek megalkotásának szabályai.
- **Szemantika** – a nyelvi szerkezetek értelmezése, alkalmazásának szabályai.

A **Lazarus** programozási környezet részei:

- **szövegszerkesztő** a program szövegének beírására és szerkesztésére;
- **compiler**, ami a **Pascal** nyelven írt programot lefordítja a számítógép számára értelmezhető utasításokra;
- **debugger**, ami a hibák keresésében és javításában lesz segítségünkre;
- **súgó**

és egyéb vezérlők.

Az **IDLE** projektfejlesztési környezet a következőkből áll:

- **szövegszerkesztő** a program szövegének beírására és szerkesztésére;
- **compiler**, ami a **Pascal** nyelven írt programot lefordítja a számítógép számára értelmezhető utasításokra;
- **debugger**, ami a hibák keresésében és javításában lesz segítségünkre;
- **súgó**

és egyéb vezérlők.

A **form** az az objektum, amelyen komponenseket – gombok, mezők, címkék, menük, gördítősávok stb. – helyezhetünk el. A program futtatása során a formból jön létre a program ablaka a vezérlőelemekkel.

A form tulajdonságai:

- **Name** (név) – a formnak mint objektumnak neve;
- **Caption** (cím) – ez jelenik meg az ablak címsorában;
- **Color** (szín) – a form háttérszíne;
- **Height** (magasság) – az ablak magassága pixelekből;
- **Width** (szélesség) – a form szélessége pixelekből;
- **Top** (teteje) – az ablak bal felső sarkának pixelben mért távolsága a képernyő tetejétől;
- **Left** (bal) – az ablak bal felső sarkának pixelben mért távolsága a képernyő jobb szélétől.

A **Python** ablakokkal rendelkező projekt létrehozásához csatolni kell a **tkinter** modult a projekthez, ami a következő parancs végrehajtásával kapcsolódik a projekthez: **from tkinter import\***

A **root** nevű ablak címsor feliratának módosításához futtassuk a **root.title('<szöveg>')** parancsot. Az ablak méretének és helyzetének megváltoztatását a **root.geometry('<paraméter sor>')** parancs hajtja végre. Az ablak háttérszínének megváltoztatásához meg kell adni a szín nevét a **['bg'] = '<color>'** parancsban.





### Felelj a kérdésekre!

- 1°. Mi a számítógépes program?
- 2°. Mik a program bemenő, köztes és kimenő adatai?
- 3°. Mi a programozási nyelv?
- 4°. Mikből áll egy programozási nyelv? Ismertesd ezeket!
- 5°. Mi a szintaktikai hiba?

- 6°. Mi a **Lazarus (IDLE)** környezet rendeltetése?
- 7°. Ismertesd a **Lazarus (IDLE)** környezet elemeit!
- 8°. Mi a compiler rendeltetése? Mik a kompiláció lépései?
- 9°. Mi a form? Ismertesd néhány tulajdonságát! Hogyan lehet megváltoztatni értéket? Hogyan hozunk létre egy projektet **Python**ban, és hogyan állítsuk be az ablak tulajdonságainak értékeit?
- 10°. Hogy nyitunk meg egy **Lazarus**-projektet? Hogy menthetjük azt? Hogyan futtathatjuk?



### Végezd el a feladatokat!

- 1°. Hozz létre projektet, ami megnyit egy 400 x 350 pixeles piros háttérű ablakot, amelynek címsora családnevedet tartalmazza! Mentse a projektet a mappában létrehozott **feladat 6.1.1** nevű mappába!
-  2°. Hozz létre projektet, ami megnyit egy 200 x 450 pixeles kék háttérű ablakot, amelynek címsora nevedet tartalmazza! Mentse a projektet a mappában létrehozott **feladat 6.1.2** nevű mappába!
- 3°. Hozz létre projektet, ami megnyit egy 300 x 300 pixeles kék háttérű ablakot, amelynek bal felső sarka 300 pixelnyire van a képernyő bal szélétől és 400 pixelnyire a tetejétől! Mentse a projektet a mappában létrehozott **feladat 6.1.3** nevű mappába!
-  4°. Hozz létre projektet, ami megnyit egy 200 x 100 pixeles szürke háttérű ablakot, amelynek bal felső sarka 150 pixelnyire van a képernyő bal szélétől és 100 pixelnyire a tetejétől! Mentse a projektet a mappában létrehozott **feladat 6.1.4** nevű mappába!

## 6.2. Események. Eseménykezelők



1. Mi a form? Milyen tulajdonságai vannak?
2. Hogyan lehet megváltoztatni a form tulajdonságait?
3. Mi a compiler rendeltetése, és milyen lépéseket végez a munkája során?
4. Hogyan lehet projektablakot létrehozni és a tulajdonságait megváltoztatni?
5. Mire szolgál az interpretátor? Milyenek a működésének főbb lépései?

### Események és azok kezelői

A formhoz a tulajdonságokon kívül még **események** is tartoznak, amelyek a program futása során bekövetkezhetnek.

Ilyen események például a

- **Click** (*kattintás*) – bármely objektum kijelölése;
- **Close** (*bezárni*) – az ablak bezárása;
- **Create** (*létrehozás*) – a program ablakának megnyitása;
- **KeyPress** (*billentyűleütés*) – a billentyűzet valamely gombjának lenyomása;
- **MouseMove** (*egér mozgása*) – az egérmutató mozgatása az ablak területén és mások.



Események végbemehetnek a billentyűzet használatával. Ilyen esemény lehet például a **KeyPress** (ang. *key press* – billentyűlenyomás) esemény – a billentyűzet bármely gombjának megnyomása. Események más objektumokkal is végbemehetnek: gomb, felirat, mező stb. A tankönyv további részeiben tanulmányozzuk majd őket.

Minden olyan eseménnyel, ami a program futása során bekövetkezhet, össze lehet kapcsolni a program egy részletét. Az ilyen programrészletet **eseménykezelőnek** nevezzük.

Valamely objektummal összekapcsolt eseménykezelőt az objektum **metódusának** nevezzük.



## A Pascalt használóknak

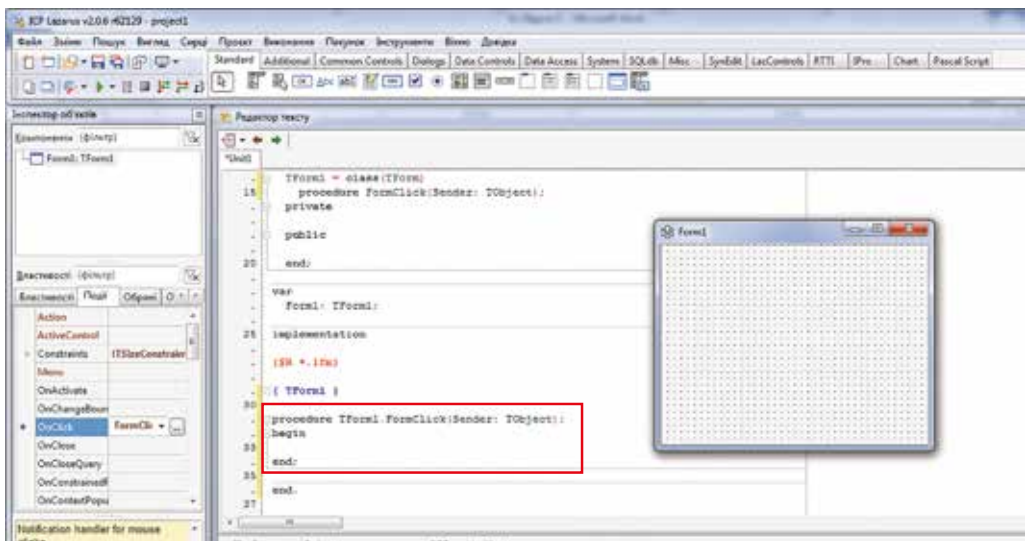
A **Lazarus** környezet az eseménykezelőt alapértelmezetten **eljárásként** hozza létre. Az **eljárás** (procedúra) a program névvel ellátott részlete, amelyet erre a névre hivatkozva indíthatunk el.

Az eljárás végrehajtása szintén esemény. Ha az eljárás annak eredményeképpen kerül végrehajtásra, hogy a formmal bekövetkezett egy esemény – például bekövetkezett a **Click** esemény –, akkor az eljárás futását **OnClick** eseménynek nevezik (ang. *on click* – a kattintásra).

Ha egy eljárást a **Click** eseményhez kívánjuk kötni, a következőket kell tenni:

1. Kattintsunk a **Formra**!
2. Az **Objektum felügyelő** ablakában kattintsunk az **Események** fülre!
3. Jelöljük ki az **OnClick** eseményt! Az **OnClick** melletti mező aktívává válik és megjelenik benne a kurzor.
4. Kattintsunk duplán erre a mezőre!

Ennek eredményeképpen az **OnClick** esemény mellett megjelenik a **FormClick** (kattintás a formon) felirat, az **Unit1** (ang. *unit* – blokk, az egész része) lapon pedig – ez tartalmazza a projekt szövegét – megjelenik a **TForm1.FormClick** eljárás sablonja, ahová be kell írni azokat az utasításokat, amelyeket a **Click** esemény bekövetkezésekor végre kell hajtani (6.12. ábra).



6.12. ábra. TForm1.FormClick eljárás sablonja

Az eljárás előkészítése a **procedure** kulcsszóval kezdődik (ang. *procedure* – eljárás), és vízszintes vonallal zárul.

Az eljárás utasításait a **begin** (kezdet) és **end** (vége) szavak közzé kell elhelyezni. *Itt jegyezzük meg*, hogy a **begin** és az **end** nem utasítások, inkább az eljárás elejét és végét jelző **utasítás zárójelek**. Ezeket a szavakat gyakran **kezelő zárójeleknek** nevezik.

Az eljárás végét jelző **end** kulcsszó után pontosvesszőnek (;) kell következni. Az eljárás valamennyi utasítását pontosvesszőnek (;) kell követni. *Itt jegyezzük meg*, hogy az egész projekt szövegének végén egy **utasítás zárójel** van, amelyet egy pont követ.

Ha azt szeretnénk, hogy az eljárás eredményeképpen megváltozzon a form valamely tulajdonsága, az eljárásnak egy **értékadó** utasítást kell tartalmazni. Ennek az utasításnak az általános alakja a következő:

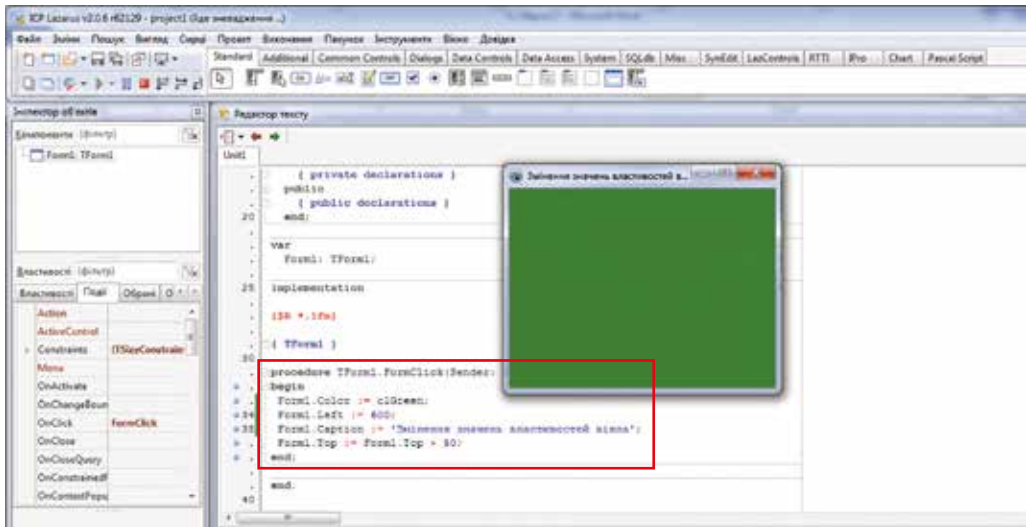
**<objektum neve>. <tulajdonság neve> := <érték vagy kifejezés>;**

A := jelet **hozzárendelési előjelnek** nevezzük. A billentyűzetről két karakter egymás utáni megadásával lehet felírni: **kettőspontot** (: ) és **egyenlő** (=), szóköz nélkül.

Például a **Form1.Color := clGreen** utasítás zöldre változtatja a form hátterét. A **Form1.Left := 600** utasítás 600 pixelre állítja a form és a képernyő bal széle közötti távolságot. A **Form1.Caption := 'a megváltozott cím'** értékadás eredményeképpen az idézőjelbe vett szöveg bekerül a form címsorába, a **Form1.Left := 300** pedig 300 pixelre állítja a form és a képernyő bal széle közötti távolságot.

A **Form1.Top := Form1.Top + 50** utasítás 50 pixellel növeli az ablak tetejének és a formnak a távolságát. Ez utóbbi utasítás értelmében a **Top** tulajdonság értéke előbb 50-nel növekszik, majd ez a szám lesz a **Top** tulajdonság új értéke.

Állítsuk össze most a **TForm1.FormClick** eljárást a fenti utasításokból, amelyek végrehajtása során megváltozik a form **Caption**, **Color**, **Left** és **Top** tulajdonságainak értéke (6.13. ábra).



6.13. ábra. A form **Caption**, **Color**, **Left** és **Top** tulajdonságainak megváltoztatása és a végrehajtás eredménye



A projekt indítása után meg kell várni az ablak létrejöttét, majd annak tetszőleges pontjára kell kattintani. Ekkor bekövetkezik a **Click** esemény, ennek eredményeképpen bekövetkezik az **OnClick** esemény, ami elindítja a **TForm1.FormClick** eseménykezelőt.

*Itt jegyezzük meg, hogy egy projektben több különböző esemény feldolgozásának eljárásai hozhatók létre egy és ugyanazon vagy különböző objektumokkal.*



### A Pythont használóknak

A **Pythonban** egy ablak eseménykezelője külön **alprogramként** jön létre, és a **bind** (angol. *bind* – kötni) módszer segítségével társul az ablakhoz. Az **alprogram** a projekt külön része, amelynek saját neve van, és ezen a néven futtatható. A **Pythonban** kétféle alprogram létezik: **eljárások** és **függvények**.

Az **eljárás** – egy alprogram, amely egy összetevő (ablakok, gombok stb.) létrehozását vagy a meglévő tulajdonságok értékének megváltoztatását, adott szöveg kiadását, kép létrehozását stb. eredményezi. Az eseménykezelők általában eljárások.

A **függvény** egy alprogram, amely egy vagy több értéket eredményez: számokat, szövegsorokat stb.

A **metódus** olyan eljárás vagy függvény, amely egy adott objektumhoz van társítva.

Az eseménykezelő és a *root* nevű ablak társítására szolgáló parancs általános alakja így néz ki:

**root.bind('<esemény>', <eseménykezelő\_esemény>)**

Az ablakkal együtt előforduló események **Pythonban** való megnevezését a 6.1. táblázat tartalmazza.

6.1. táblázat

#### A Python eseményeinek jelölése

| Esemény   | Jelölés a Pythonban   |
|---|---|
| <b>Click</b> – bal egér kattintás                     | <b>&lt;Button-1&gt;</b> vagy <b>&lt;1&gt;</b>                     |
| <b>DbClick</b> – dupla bal egér kattintás             | <b>&lt;Double-Button-1&gt;</b> vagy <b>&lt;Double-1&gt;</b>       |
| <b>MouseMove</b> – az egérmutatató mozgása az ablakon | <b>&lt;Motion&gt;</b> (ang. <i>motion</i> – mozgás)               |
| <b>KeyPress</b> – bármely billentyű megnyomásával     | <b>&lt;KeyPress&gt;</b>   |
| <b>Create</b> – ablak létrehozása                     | <b>&lt;Expose&gt;</b> (ang. <i>expose</i> – nyilvánosságra hozni) |
| <b>Close</b> – ablak bezárása                         | <b>&lt;FocusOut&gt;</b> (ang. <i>focus out</i> – fókusz kimenet)  |

Például, ha a **Click** (bal egér kattintás) eseménykezelőnek a *click* nevet adjuk, akkor az a parancs, amely ezt az eljárást egy *root* nevű ablakhoz társítja, így néz ki:

**root.bind('<1>', click).**

Az eseménykezelő általános kinézete a következő:

**def <eseménykezelő\_esemény> (event):**

**<az esemény során végrehajtandó parancsok>**

(ang. *define* – meghatározni, *event* – esemény).

Az eseménykezelő első sorát címsornak nevezzük. Például egy *click* nevű eseménykezelő esetében a címsor így néz ki:

**def click (event):**

Az eseménykezelő parancsokat behúzással kell írni a kódszerkesztő ablak bal szélétől jobbra. A behúzás automatikusan jön létre az **Enter** billentyű lenyomásával a „.” karakter után, vagy megtehetjük saját magunk is a **Tab** billentyű egyszeri lenyomásával.

Ha azt szeretnénk, hogy az esemény megváltoztassa az ablak egyes tulajdonságainak értékeit, az eseménykezelőnek parancsokat kell tartalmaznia ezen tulajdonságok értékeinek megváltoztatásához. Ezek lehetnek parancsok bizonyos módszerek alkalmazásához egy ablakban, például **geometry()**, **title()**, vagy **hozzárendelési** parancs. A hozzárendelés parancs általános nézete a következő:

**<objektumnév>['<tulajdonság>'] = <érték>**

Az = jelet **hozzárendelési jelnek** nevezzük.

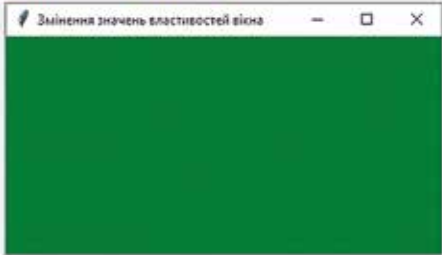
Például a **root['bg'] = 'green'** hozzárendelési parancs végrehajtása a *root* nevű ablaknak zöld színt állít be. A **root.title ('Az ablak tulajdonságainak megváltoztatása')** módszert alkalmazva, a szöveget idézőjelben jeleníthetjük meg az ablak címsorában. Ahhoz pedig, hogy az ablak mérete 400 × 200 pixeles legyen, a képernyő bal szélétől 500 pixelt, a felső részétől 70 pixelig terjedő távolságot kell használni a **root.geometry('400x200+500+70')** módszerrel.

Az eseménykezelőt használat előtt rögzítik, általában a projekt szövegének elején. A parancsok rögzítésének az eseménykezelő után az ablak bal szélétől kell kezdődnie. A **Backspace** gombbal vigyük a kurzort ebbe a helyzetbe. Igen, az eseménykezelő után fel kell írni az ablak létrehozásának és az eseménykezelőnek az ahhoz való összekapcsolásának parancsát:

**root = Tk()**

**root.bind('<1>', click)**

A 6.14. ábra egy példát mutat be a *click* eljárásra a *click* nevű eseménykezelő esetében, amely a fentebb említett parancsokból áll, a végrehajtás során megváltozik a **bg** tulajdonság értéke, a címszöveg, az ablak mérete és helyzete, valamint a végrehajtás eredménye.



```
from tkinter import *
def click (event):
    root['bg'] = 'green'
    root.title('Змінення значень властивостей вікна')
    root.geometry('400x200+500+70')
root=Tk()
root.bind('<1>', click)
```

6.14. ábra. A projekt és az eredménye a **Click** esemény bekövetkezésekor

Néha szükség van egy objektum tulajdonság új értékének beállítására az esemény feldolgozása során, amelyet ugyanannak a tulajdonságnak az aktuális értéke alapján számolnak. Például az egér bal gombjára kattintás után az ablakot 50 pixelnyire kell leengedni, vagyis az ablak távolságát a képernyő felső határától 50 pixellel meg kell növelni. Ebben az esetben meg kell szerezni az aktuális távolságértéket, 50-re kell megváltoztatni, és az eredményt külön változóban kell megjegyezni az ablak új pozíciójának beállításához.





A következő ablakmódszereket használjuk az ablak méretének és helyzetének aktuális értékeinek megszerzéséhez:

- **winfo\_width()** – (ang. *window information* – ablak információ) ablak szélessége;
- **winfo\_height()** – ablakmagasság;
- **winfo\_x()** – az ablak bal szegélyének távolsága a képernyő bal szélétől;
- **winfo\_y()** – az ablak felső szegélyének távolsága a képernyő felső részétől.

Például a *root* ablak aktuális távolságának megismerésére a képernyő felső részétől, valamint 50-nel növelni is azt, a következő parancsot kell végrehajtani:

**t = root.winfo\_y() + 50**

Új érték beállításához a **geometry()** metódus ablak helyzetét és méretét jelző sorba a szám helyett be kell helyettesíteni a változó értékét. Ehhez, ahova be akarjuk illeszteni a változó értékeit, tegyünk { } zárójeleket, és alkalmazzuk a **format ()** helyettesítési módszert a változó nevű karakterláncra, pontokkal elválasztva:

**root.geometry('300x200+500+{}'.format(t))**

Ha egy sorban több változó értékét akarjuk helyettesíteni, akkor minden megfelelő helyen be kell írnia a { } zárójeleket, a **format ()** metódusba pedig vesszőn keresztül fel kell sorolni a változók nevét.

Például annak érdekében, hogy az ablakot 50 pixellel lejjebb lehessen csökkenteni, és szélességét 40 pixellel növelni az *egér bal gombjára való kattintás* után, a 6.15. ábrán látható projektet kell elvégezni.

```
from tkinter import *
def click (event):
    t = root.winfo_y()+50
    w = root.winfo_width()+40
    root.geometry('{}x320+200+{}'.format(w, t))
root = Tk()
root.geometry('480x320+200+200')
root.bind('<1>', click)
```

6.15. ábra. Az ablak mérete és helyzete megváltoztatásának a projektje, az előző értékeitől függően

Miután elindítottuk a projektet a futtatáshoz, meg kell várnia, amíg az ablak megnyílik, és ki kell választani annak bármely pontját. Ez létrehoz egy **Click** eseményt az ablakhoz, és végrehajtásra kerül egy eljárás – az eseménykezelő, amelynek eredményeként megváltozik az ablak helye és mérete.

*Itt jegyezzük meg*, hogy egy projektben eljárásokat lehet létrehozni több különböző esemény feldolgozására ugyanazokkal vagy más objektumokkal.

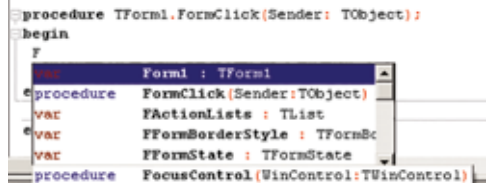
## A projekt létrehozásának és szerkesztésének eszközei



### A Pascalt használóknak

A **Lazarus** környezet kényelmes eszközökkel támogatja a projekt megírását:

- az objektum nevének első vagy néhány első betűjének beírása után meg lehet nyomni a **Ctrl + Szóköz** gombot, aminek eredményeképpen megnyílik egy lista, amiből kiválaszthatjuk az objektum nevét (6.16. ábra);
- miután az objektum neve után beírtuk a pontot, megnyílik az objektum tulajdonságait tartalmazó lista (6.17. ábra). Ezután már csak ki kell választani



6.16. ábra. Az objektumok nevének kiválasztása

ni a tulajdonságot, és le kell ütni az **Entert**, vagy kattintsunk kétszer a megfelelő sorra! Ha beírjuk a tulajdonság első néhány betűjét, a tulajdonságok listája egyre rövidebb lesz, mert csak azok maradnak meg, amelyek kezdőbetűi megegyeznek a beírt betűkkel. A módszer egyrészt meggyorsítja a program beírását, másrészt csökkenti a hibalehetőségek számát, különösen azok számára, akik nem ismerik az angol nyelvet.

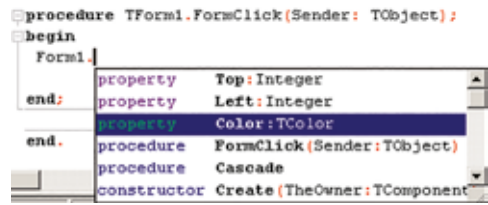
- A rendszer sok esetben kiteszi a megfelelő helyre a műveleti zárójeleket, értékadás (műveleti) jeleket, pontosvesszőket stb.

A rendszer nem különbözteti meg a kis- és nagybetűket. Ha azonban a szó összetett, a könnyebb értelmezhetőség kedvéért a szókezdő betűket nagybetűvel szokás írni.

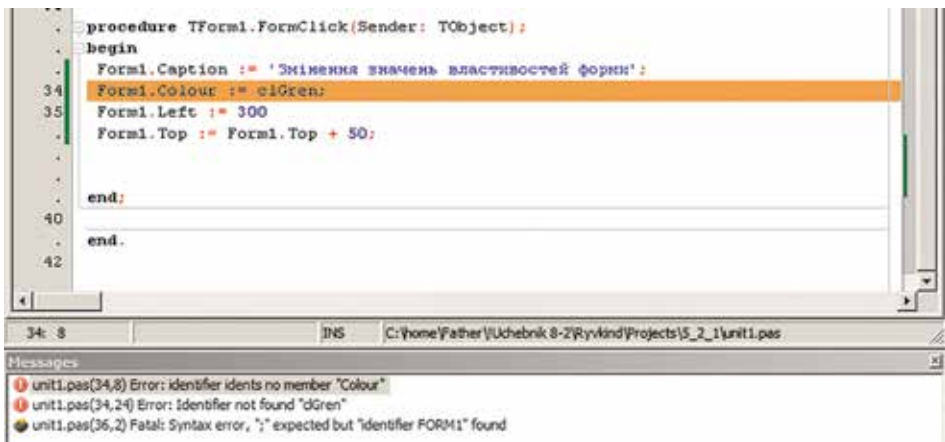
Már tudjátok, hogy a fordítás kezdetén a compiler szintaktikai ellenőrzést végez. Ha talál ilyeneket, akkor a **Szövegszerkesztő** alatt található **Üzenetek** ablakban megjelenik a hibák listája azoknak a soroknak a számával, ahol a compiler a hibát találta, valamint a hiba valószínű okára utaló kommentárral. Ezenkívül háttérszínnel lesz kiemelve az első hibás sor, vagy az utána következő sor.

A 6.18-as ábrán a 34. sor van kiemelve, mivel szintaktikai hibát tartalmaz: hibásan vannak leírva **Color** és **Green** szavak. Ugyancsak a következő, 35. sor végéről hiányzik a pontosvessző. Ilyen esetben az üzenetek között azt az információt találjuk, hogy helytelen szimbólummal kezdődik a következő, a 36. sor.

Ha a kompilátor (fordítóprogram) hibákat talál, akkor ki kell javítani a hibákat, majd újra kell indítani a projekt futtatását.



6.17. ábra. Az objektum tulajdonságainak listája



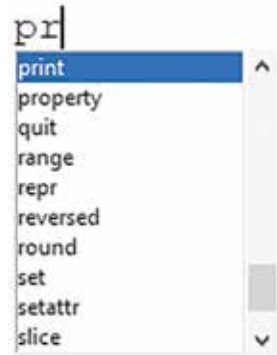
6.18. ábra. Szintaktikai hibák jelölése a Lazarusban



## A Pythont használóknak

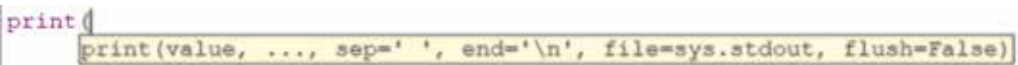
Az **IDLE** környezet tartalmaz néhány eszközt a projekt-szöveg írásához:

- a kulcsszó első vagy néhány első betűjének megadása után megnyomhatjuk a **Ctrl + Szóköz** billentyűt, és megnyílik egy lista, amelyből kiválaszthatjuk a kívánt kulcsszót (6.19. ábra);
- a függvény nevének és a nyitó zárójel beírása után automatikusan megjelenik egy tipp, hogy mely argumentumok lehetnek ebben a függvényben (6.20. ábra);
- a rendszer automatikusan elhelyezi a behúzásokat a projekt szövegének megfelelő helyein a kettősponttal (: ) végződő sorok után.



6.19. ábra. Kulcsszavak listája

A rendszer különböztet meg nagy- és kisbetűket. Ha egy szó több szóból áll, akkor az a szokás, hogy az egyes szavak első betűjét nagybetűvel írjuk (a jobb megértés érdekében).

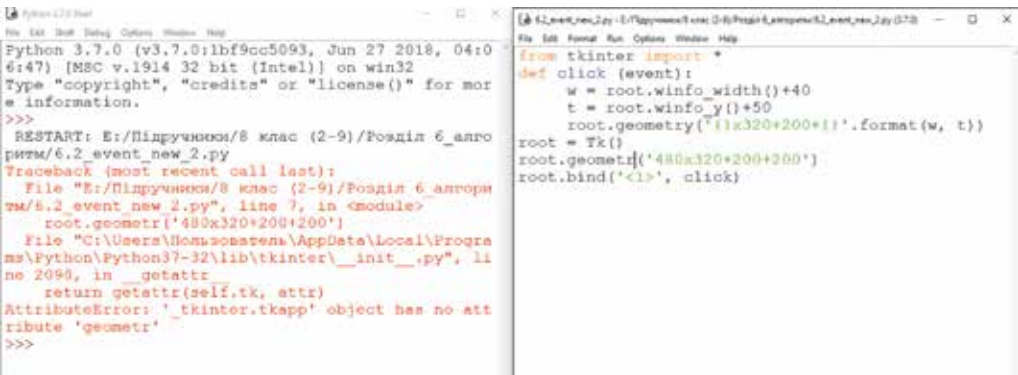


6.20. ábra. A lehetséges argumentumok tippje

Már tudjuk, hogy a projekt végrehajtása során az interpretátor ellenőrzi a projekt szövegének minden egyes szintaktikai hibáját. Ha ilyen hibák fordulnak elő, a projekt megszakad, és az **IDLE** ablakban annak a sornak a száma jelenik meg, ahol a hiba történt, valamint a hibával kapcsolatos megjegyzés is megjelenik.

A 6.21. ábrán a 7. sorban található hibaüzenet és annak leírása pirosan jelenik meg. A hibaüzenet utolsó sorában található annak a 'geometr' metódusnak a neve, amely miatt a projekt leállt.

Javítania kell a projekt szövegében található szintaxis hibákat, és újra kell indítania a projektet a végrehajtáshoz.



6.21. ábra. Szintaktikai hibaüzenet az **IDLE** környezetben

## Üzenetablak és azok alkalmazása

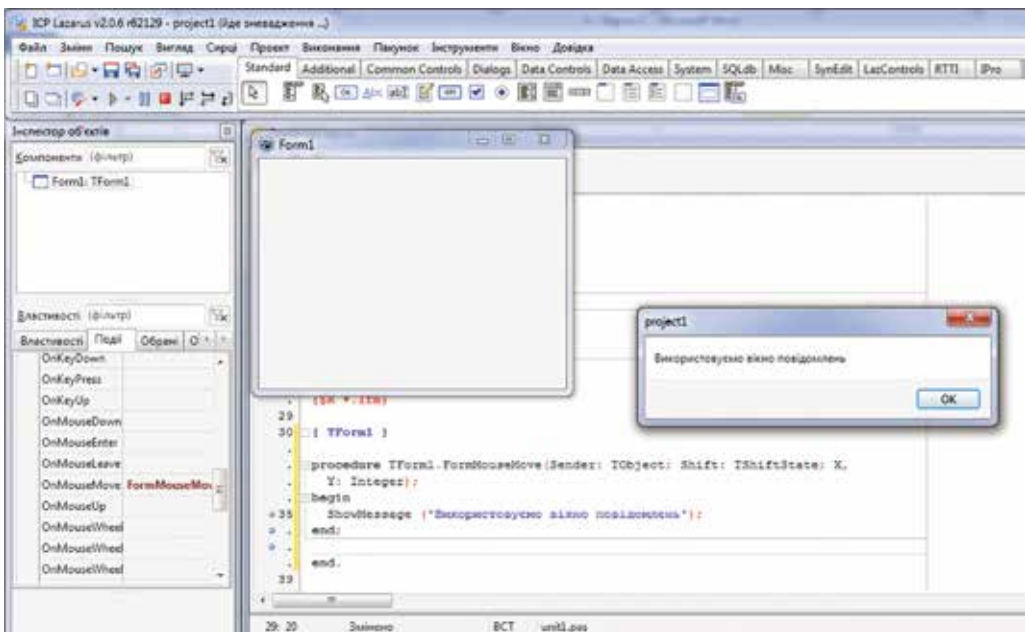
Üzenetablak segítségével szöveges üzeneteket jeleníthetünk meg a projektünk során.



### A Pascalt használóknak

Szöveges üzeneteket **üzenetablakok** segítségével helyezhetünk el a képernyőn. Üzenetablakot a **ShowMessage** ( <szöveg> ) (ang. *show messag* – üzenet megjelenítése) parancs segítségével nyithatunk meg, ahol zárójelek és idézőjelek között kell megadni a megjelenítendő üzenetet.


Az 6.22. ábrán látható esetben a **MouseMove** eseménykezelőjében helyeztünk el egy ilyen utasítást, ugyanezen az ábrán láthatjuk a végrehajtás eredményét is.



6.22. ábra. Üzenetablak alkalmazása

Ha elolvastuk az üzenetet, az üzenetablakot be kell zárni az **OK** gombbal. A projekt ablakát csak ezután tudjuk bezárni.




*Felhívjuk a figyelmeteket arra, hogy a példában a **MouseMove** eseményt használtuk. Ebben az az érdekes, hogy ez esetben a program ablakát nem lehet bezárni az egérrel. Ez azért van, mert amikor az egér megmozdul az ablak területén, azonnal bekövetkezik a **MouseMove** esemény, lefut az **OnMouseMove** eseménykezelő, ami elindítja a **TForm1. FormMouseMove** eljárást, ez pedig megjelenít egy üzenetablakot.*

A programot az **Eszköztár Bezárás**  gombjával, vagy az **Alt + F4** billentyűkombinációval lehet ilyenkor bezárni. Ez a billentyűkombináció minden projekt esetében alkalmazható.



## A Pythont használóknak

A Pythonban többféle üzenetablakot hozhatunk létre. Ehhez a következő parancsok szükségesek:

- **showinfo** ('<szöveg a címsorban>', '<üzenet szövege>') (ang. *show info* – információ megjelenítése) – létrehoz egy  ikont és szöveges üzenetet tartalmazó információs ablakot;
- **showwarning** ('<szöveg a címsorban>', '<üzenet szövege>') (ang. *warning* – figyelmeztetés) létrehoz egy figyelmeztető ablakot  ikonnal és figyelmeztető szöveggel;
- **showerror** ('<szöveg a címsorban>', '<üzenet szövege>') (ang. *error* – hiba) létrehoz egy hibajelző ablakot a hibaüzenet  ikonjával, valamint a hibáról informáló szöveggel.

Az üzenetablak megnyitásához a fejléc megadott szövegével és az üzenet megadott szövegének megjelenítéséhez importáljuk a **tkinter.messagebox** modult a projektbe, és az ablak céljának megfelelően használjuk a parancsot!

Például meg kell nyitnunk egy üzenetablakot, amelynek szövege az *Események* fejléc sorában található, és az *Üzenetablakot használom* szöveget tartalmazza, amikor a mutatót áthelyezzük a projekt végrehajtási ablakába. Egy példát, a **MouseMove** eseménykezelő szövegére egy projektablakhoz, amely parancsot tartalmaz az üzenetablak és a megfelelő projekt eredményének megjelenítésére, a 6.23. ábra mutatja.



6.23. ábra. Az üzenetablak használata

Az üzenet elolvasása után be kell zárnia ezt az ablakot az **OK** vagy az ablak bezárásához szolgáló gomb kiválasztásával. Csak ezután lehet bezárni a projekt végrehajtási ablakát.

*Itt jegyezzük meg*, hogy ebben a példában a **MouseMove** eseményt használjuk (az esemény megnevezése **Pythonnon** – **<Motion>**). Érdekes módon ebben az esetben, ha az üzenetablak megnyílik a projekt végrehajtási ablakának tetején, akkor a projektet nehéz lesz befejezni az egérrel. Végül is, amint a mutató elmozdul egy projektablakon, bekövetkezik a **MouseMove** esemény, és ennek eredményeként az eseménykezelő, amely ismét megnyit egy üzenetablakot, elindul a végrehajtásakor. Ezért az **Alt + F4** billentyű lenyomásával befejezhetjük a projektet. Ez a billentyűparancs használható bármely projekt végrehajtási ablakának bezárására.



### Használjuk a számítógépet!



Végezzétek el a gyakorlati feladatot a <https://cutt.ly/hhdriEE> linkről, vagy a QR-kód segítségével!

## ! Összefoglalás

Minden olyan eseménnyel, ami megtörténhet egy formával, össze lehet kapcsolni egy programrészletet, ami az esemény bekövetkeztekor fog lefutni. Az ilyen programrészletet **eseménykezelőnek** nevezzük. Az objektummal összekapcsolt eseménykezelőt **metódusnak** nevezzük.



### A Pascalt használóknak

A **Lazarus** környezet az eseménykezelőt alapértelmezetten **eljárásként** hozza létre. Az **Eljárás** a program névvel rendelkező része, amelyet erre a névre hivatkozva lehet futtatni.

Az eljárás utasításait a **begin** és **end** kulcsszavak között kell elhelyezni. Ezek a szavak nem utasítások, hanem az eljárás elejét és végét jelölő **utasítás zárójelek**

Az eljárás végét jelölő **end** kulcsszó után pontosvesszőt (;) kell tenni. Ugyancsak pontosvesszővel (;) kell lezárni az eljárást alkotó utasításokat.

Ha azt szeretnénk, hogy az eljárás eredményeképpen megváltozzon a form valamely tulajdonsága, az eljárásnak egy **értékadó** utasítást kell tartalmaznia. Ennek az utasításnak az általános alakja a következő:

**<objektum neve>.<tulajdonság neve> := <érték vagy kifejezés>;**

Üzenetablakot a **ShowMessage** (<szöveg>) utasítás segítségével nyithatunk, ahol zárójelek és idézőjelek között kell megadni a megjelenítendő üzenetet.



### A Pythont használóknak

A **Pythonban** egy ablak eseménykezelője külön **alprogramként** jön létre, és a **bind()** módszer segítségével társul az ablakhoz.

Az **eljárás** a program külön része, amelynek saját neve van, és ezen a néven futtatható.

Az eseménykezelő és a root nevű ablak társítására szolgáló parancs általános alakja így néz ki:

**root.bind('<esemény>', <eseménykezelő\_esemény>)**

#### A Python eseményeinek jelölése

| <b>Esemény</b>   | <b>Jelölés a Pythonban</b>  |
|--|---|
| <b>Click</b> – bal egér kattintás                      | <b>&lt;Button-1&gt;</b> vagy <b>&lt;1&gt;</b>                     |
| <b>DbClick</b> – dupla bal egér kattintás              | <b>&lt;Double-Button-1&gt;</b> vagy <b>&lt;Double-1&gt;</b>       |
| <b>MouseMove</b> – az egérmutatató mozgása az ablakban | <b>&lt;Motion&gt;</b> (ang. <i>motion</i> – mozgás)               |
| <b>KeyPress</b> – bármely billentyű megnyomásával      | <b>&lt;KeyPress&gt;</b>   |
| <b>Create</b> – ablak létrehozása                      | <b>&lt;Expose&gt;</b> (ang. <i>expose</i> – nyilvánosságra hozni) |
| <b>Close</b> – ablak bezárása                          | <b>&lt;FocusOut&gt;</b> (ang. <i>focus out</i> – fókusz kimenet)  |





Az eseménykezelő általános kinézete a következő:

**def <eseménykezelő\_esemény> (event):**

**<az esemény során végrehajtandó parancsok>**

Ha azt szeretnénk, hogy az esemény megváltoztassa az ablak egyes tulajdonságainak értékeit, az eseménykezelőnek parancsokat kell tartalmaznia ezen tulajdonságok értékeinek megváltoztatásához.

A következő ablakmódszereket használjuk az ablak méretének és helyzetének aktuális értékeinek megszerzéséhez:




- **winfo\_width()** – (ang. *window information* – ablak információ) ablak szélessége;
- **winfo\_height()** – ablakmagasság;
- **winfo\_x()** – az ablak bal szegélyének távolsága a képernyő bal szélétől;
- **winfo\_y()** – az ablak felső szegélyének távolsága a képernyő felső részétől.

Új érték beállításához a **geometry()** metódus ablak helyzetét és méretét jelző sorba a szám helyett be kell helyettesíteni a változó értékét. Ehhez, ahova be akarjuk illeszteni a változó értékeit, tegyünk { } zárójeleket, és alkalmazzuk a **format ()** helyettesítési módszert a változó nevű karakterláncra, pontokkal elválasztva: **root.geometry('300x200+500+{}`).format(t)**

Az **IDLE** környezetben vannak olyan eszközök, amelyek felgyorsítják a projekt szövegének írását. A nagy- és kisbetűket a rendszer különbözőnek tekinti.

**Üzenetablak** segítségével szöveges üzeneteket jeleníthetünk meg a projektünk során.

A **Python**ban többféle üzenetablakot hozhatunk létre. Ehhez a következő parancsok szükségesek:

- **showinfo ('<szöveg a címsorban>', '<üzenet szövege>')** (ang. *show info* – információ megjelenítése) – létrehoz egy ikont  és szöveges üzenetet tartalmazó információs ablakot;
- **showwarning ('<szöveg a címsorban>', '<üzenet szövege>')** (ang. *warning* – figyelmeztetés) – létrehoz egy figyelmeztető ablakot  ikonnal és figyelmeztető szöveggel;
- **showerror ('<szöveg a címsorban>', '<üzenet szövege>')** (ang. *error* – hiba) – létrehoz egy hibajelző ablakot a hibaüzenet  ikonjával, valamint a hibáról informáló szöveggel.

Az üzenetablak megnyitásához a fejléc megadott szövegével és az üzenet megadott szövegének megjelenítéséhez importáljuk a **tkinter.messagebox** modult a projektbe, és az ablak céljának megfelelően használjuk a parancsot!



### Felelj a kérdésekre!

- 1<sup>o</sup>. Milyen események történhetnek egy formával?
- 2<sup>o</sup>. Mi a kapcsolat a **Click** és **OnClick** események között?
- 3<sup>o</sup>. Mi az eseménykezelő?
- 4<sup>o</sup>. Hogy kapcsoljuk az eseményhez az eseménykezelőt?
- 5<sup>o</sup>. Mi a metódus?
- 6<sup>o</sup>. Mi az eljárás?
- 7<sup>o</sup>. Mi az utasítás zárójel a **Pascal**ban? Mire szolgálnak?
- 8<sup>o</sup>. Hogy lehet megtervezni egy eseménykezelőt a **Python**ban?
- 9<sup>o</sup>. Milyen utasításokkal változtathatjuk meg a form tulajdonságait? Hogy néznek ki? Mondj példákat!
- 10<sup>o</sup>. Mely esetekben fordulnak elő szintaxis hibák? Mit kell tennie ilyenkor?
- 11<sup>o</sup>. Milyen utasítással nyithatunk meg egy üzenetablakot?

**Végezd el a feladatokat!**

- 1°. Írd be a füzetedbe azt a parancsot, amely:
- a) beállítja az ablak magasságát 200 pixelre;
  - b) növeli az ablak magasságát 200 pixellel;
  - c) csökkenti az ablak szélességét 100 pixellel;
  - d) beállítja az ablak eltolást a képernyő bal szélétől 100 pixelre;
  - e) az ablakot 100 pixellel jobbra helyezi;
  - e) ablakot 100 pixellel balra helyezi;
  - g) az ablakot 200 pixellel feljebb helyezi;
  - h) az ablakot 100 pixellel lefelé helyezi!



- 2°. Írd be a füzetbe azt a parancsot, amely:
- a) beállítja az ablak szélességét 150 pixelre;
  - b) növeli az ablak szélességét 150 pixellel;
  - c) csökkenti az ablak magasságát 100 pixellel;
  - d) eltolja az ablakot a képernyő felső határától 100 pixelre;
  - e) az ablakot 200 pixellel jobbra helyezi;
  - e) az ablakot 200 pixellel balra helyezi;
  - g) az ablakot 100 pixellel felfelé helyezi;
  - h) az ablakot 200 pixellel lefelé helyezi!

- 3°. Hozz létre projektet, amelyben az **OnClick** esemény hatására a form mérete 400 × 300 lesz, a háttér kékre vált, a címsorba beíródik a lakóhelyed neve! Mentse a projektet a **feladat 6.2.3** néven létrehozott mappába!



- 4°. Hozz létre projektet, amelyben az **OnKeyPress** esemény hatására a form mérete 300 × 200 lesz, a háttér vörösre vált, a címsorba beíródik a családneved! Mentse a projektet a **feladat 6.2.4** néven létrehozott mappába!

- 5°. Hozz létre projektet, amelyben az **OnDoubleClick** (*double* – dupla) esemény hatására a háttér is és a címsor is megváltozik, valamint 200 pixellel megnő a form és a képernyő teteje közötti távolság! Mentse a projektet a **feladat 6.2.5** néven létrehozott mappába!



- 6°. Hozz létre projektet, amelyben az **Create** (<**Expose**>) esemény hatására a form mérete, valamint a háttér megváltozik és 100 pixellel csökken a form és a képernyő bal széle közötti távolság! Mentse a projektet a **feladat 6.2.6** néven létrehozott mappába!

- 7°. Hozz létre projektet, amelyben az **Click** esemény hatására a form mérete 500 × 500 lesz, a háttér barnára vált, a címsorba beíródik a *Projekt két eljárással* szöveg! A **MouseMove** eseménye váltsa ki egy üzenetablak megjelenítését *Ukrájna állampolgára vagyok* üzenettel! Mentse a projektet a **feladat 6.2.7** néven létrehozott mappába!

- 8°. Hozz létre projektet, amelyik a **Close** (<**FocusOut**>) esemény hatására megjelenít egy üzenetablakot *A program futása befejeződött. Viszontlátásra!* üzenettel! Mentse a projektet a **feladat 6.2.8** néven létrehozott mappába!

- 9°. Hozz létre projektet, amelyik a **Click** esemény hatására megjelenít egy üzenetablakot, amelyben a címsorba található szöveg jelenik meg! Mentse a projektet a **feladat 6.2.9** néven létrehozott mappába!



## 8. SZÁMÚ GYAKORLATI MUNKA

### Üzenetablakot megjelenítő objektumorientált program létrehozása

**Figyelem!** A számítógép használata során tartsd be a balesetvédelmi szabályokat és az egészségügyi előírásokat!

- Hozz létre egy projektet, amely egy ablakot (formot) nyit meg a következő tulajdonságértékekkel:
  - háttér – szürke;
  - szélesség – 200 pixel, magasság – 100 pixel;
  - távolság a bal oldaltól – 150 pixel;
  - távolság a képernyő tetejétől – 100 pixel;
  - címsor – 8. számú gyakorlati munka!
- Mentsd a projektet a mappában létrehozott **Gyakorlati 8** mappába!
- Hozz létre a formon a **Click** eseményhez egy eseménykezelőt, ami piros háttérszint állít be, 200 pixelre állítja a form és képernyő teteje közötti távolságot, 300 pixellel növeli a form szélességét, 50 pixellel csökkenti a form és a képernyő bal széle közötti távolságot, a címsorba beírja az *Programozást tanulunk* szöveget!
- Mentsd el a projektet ugyanazon a néven ugyanabba a mappába!
- Adj a projekthez eseménykezelőt a **KeyPress** eseményhez, ami a hátteret kékre állítja, a szélességét 200 pixellel növeli, a képernyő teteje és a form közötti távolságot 300 pixellel növeli!
- Mentsd el a projektet ugyanazon a néven ugyanabba a mappába!

### 6.3. A gomb



1. A form milyen tulajdonságait ismeritek? Hogy változtatjuk meg ezeket a tulajdonságokat a projekt futtatása előtt?
2. Milyen események következhetnek be a formon? Mit jelentenek ezek?
3. Mi az eseménykezelő? Hogy hozzuk létre? Hogy futtatjuk?

#### Gomb tulajdonságok

A **gomb** az egyik legelterjedtebb vezérlőelem, amit a formon elhelyezhetünk.

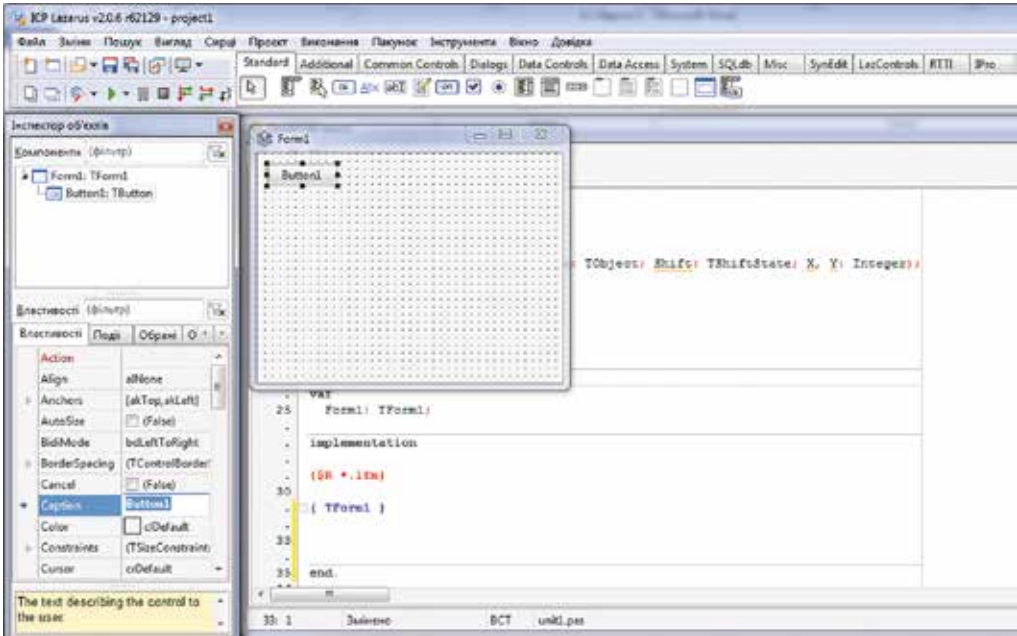


#### A Pascalt használóknak

Ha gombot szeretnénk elhelyezni, kattintsunk duplán az egér bal gombjával a **Standard** eszköztár **Ok** gombjára!

A gomb az ablak bal felső részében jelenik meg (6.24. ábra), a gomb neve hozzáadódik az **objektum felügyelő komponens**-listájához, amit a felügyelő ablakának felső részében láthatunk. Szükség esetén a gombot áthúzhatjuk az ablak tetszőleges helyére.

Ha kijelöljük a gombot a formon, vagy az **Objektum-felügyelő** ablakában, akkor ebben az ablakban a gomb tulajdonságai lesznek aktívak. Ahogy a form esetén megtapasztalhattuk, a gombnak is vannak alapértelmezett tulajdonságai (6.24. ábra). Természetesen a gomb tulajdonságait is megváltoztathatjuk.



6.24. ábra. Gomb a formon


Egyes tulajdonságok értékét be kell írni a megfelelő mezőbe, másokat a tulajdonsághoz tartozó listából választhatjuk ki; a gomb helyzetét áthúzással, méretét a méretezőpontok mozgatásával változtathatjuk meg.

A gomb rendelkezik mindazokkal a tulajdonságokkal, amikkel a formmal kapcsolatban már megismertedtünk (**Caption**, **Color**, **Height**, **Width**, **Top**, **Left**, **Name**). A **Name** tulajdonság értéke megadja a gomb nevét objektumként. A **Height** és **Width** tulajdonságok értékei határozzák meg a gomb magasságát és szélességét. A **Caption** tulajdonság természetesen a gombon látható feliratra vonatkozik, a **Top** és **Left** tulajdonságok viszonyítási pontja pedig nem a képernyő, hanem a form felső, illetve bal széle.

Ezen tulajdonságok értékét a **Tulajdonságok** lapon módosíthatjuk az **Objektum-felügyelő** ablakot, mielőtt elindítanánk a projekt végrehajtását. Mindezen tulajdonságok értékei a projekt végrehajtása során hozzárendelési parancsokkal is megváltoztathatók. Például **Button1.Width := 100**, **Button1.Left := Button1.Left + 200**, **Button1.Caption := 'Végrehajtani'**.

A gomb esetében a **Color** tulajdonság megváltoztatása nem változtatja meg a gomb alapértelmezett színét.

Tekintsünk át még néhány tulajdonságot, amellyel mind a form, mind pedig a gomb rendelkezik:

- **Enabled** (*engedélyezett, hozzáférhető*) – ennek értéke határozza meg, hogy az objektum tulajdonságain végezhető-e műveletek. Értéke lehet **True** (igaz, vagyis engedélyezett) vagy **False** (*hamis, azaz tiltott*) az elérhetetlen gomb .
- **Visible** (*látható*) – értéke azt határozza meg, hogy az objektum látható-e az ablakban. Ennek a tulajdonságnak is csak két értéke lehet, a **True** (látható), illetve a **False** (nem látható).



## A Pythont használóknak

Ahhoz, hogy gombot helyezünk el az ablakban:

1. Létrehozunk egy új Button típusú objektumot és hozzákapcsoljuk ahhoz a változóhoz, amely meghatározza az objektum nevét.
2. Beállítjuk a gomb tulajdonságainak értékeit vagy alapértelmezettként hagyjuk.
3. Elhelyezzük a létrehozott objektumot az ablakban.

A gomb létrehozásához és változóhoz való kapcsolásához az alábbi parancsot használjuk:

```
<gomb_neve> = Button()
```

vagy

```
<gomb_neve> = Button(<tulajdonságok és értékek halmaza>)
```

Az első esetben egy gomb jön létre **magasság**, **szélesség** és **szín** tulajdonságok értékeivel, amelyek alapértelmezetten be vannak állítva felirat nélkül. A második esetben egy gomb jön létre olyan tulajdonságok értékével, amely ebben az utasításban van megadva. A zárójelben vesszővel jelölik a párokat **<tulajdonság> = <érték>**. A gomb számára az alábbi tulajdonságok értékei állíthatóak be:

- **text** – a gomb szövege;
- **width** – a gomb szélessége (a vízszintesen egymás mellé helyezhető karakterek száma);
- **height** – a gomb magassága (a függőlegesen egymás alá helyezhető karakterek száma);
- **bg** – a háttér színe (az érték lehet a szín neve angolul);
- **fg** – a szöveg karaktereinek színe;
- **font** – a szöveg karaktereinek betűtípusa a gombon (az érték tartalmazhatja a betűtípust, karakterméretet és szövegkiemelést);
- **state** – a gomb állapota (**normal** – egyszerű, **disable** – nem érhető el).

Például *button* nevű gombot *Megváltoztatni* szöveggel, amely 15 karakter széles, az alábbi paranccsal hozhatunk létre:

```
button = Button(text = 'Megváltoztatni', width = 15).
```

A gomb ablakban való elhelyezéséhez az alábbi metódusok használhatóak:

- **pack()** – a gomb automatikusan az ablak címsora alá kerül, az ablak bal és jobb szélétől azonos behúzásra. Ha több gomb van az ablakban, akkor egymás alatt jelennek meg, szóközök nélkül. Ha valamilyen behúzással ellátott gombot kell elhelyezni a címsortól vagy az előző gombtól, akkor ezt pixelben megadhatjuk zárójelben, mint a **pady** tulajdonság értékét. Például a **button.pack(pady = 20)** parancs végrehajtása után, a gomb 20 pixel távolságra lesz az ablak tetejétől vagy az előző gombtól;
- **place(x = <behúzás az ablak bal szélétől>, y = <behúzás az ablak felső határától>)** – a gomb bal felső része az ablak **x** és **y** koordinátákkal ellátott pontjában helyezkedik el. Például a **button.place(x = 200, y = 100)** parancs végrehajtása után a gomb bal felső része az ablak (200, 100) koordinátájú pontjában fog elhelyezkedni

és egyebek.

A projekt végrehajtása során szükség lehet a gomb láthatatlanná tételére. Ehhez a gomb elhelyezésének módjától függően használható metódusok a **pack\_forget()** vagy **place\_forget()**. A gomb ismét láthatóvá tételéhez a **pack ()** vagy a **place ()** metódus használendő.

A 6.25. ábrán adott a projekt szövege a gomb létrehozásáról és elhelyezéséről az ablakban, valamint a végrehajtás eredménye.



6.25. ábra. Gomb az ablakban

## A gomb eseménykezelői

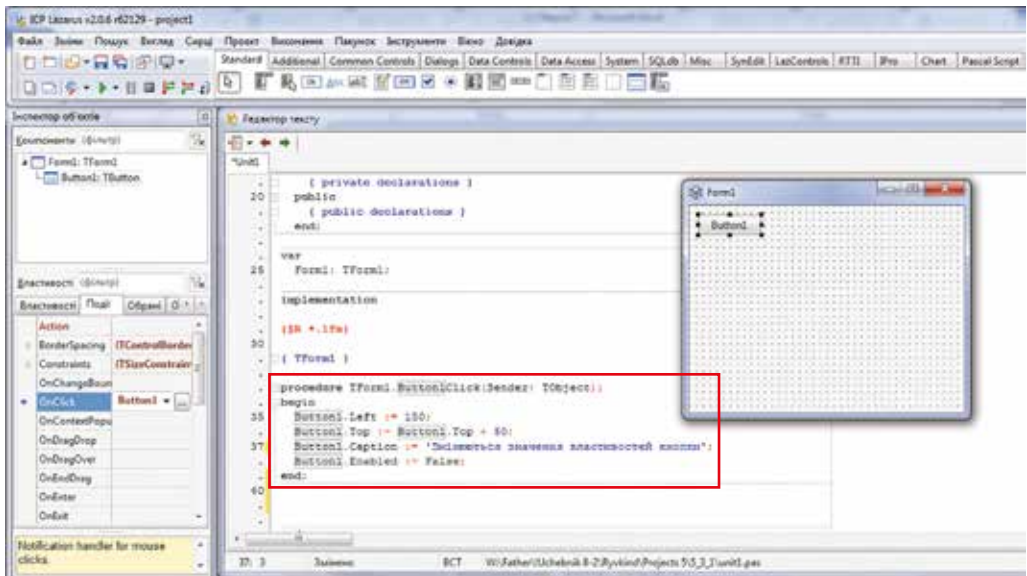


### A Pascalt használóknak

Ha a gomb ki van jelölve, és az **Objektum felügyelő Események** lapján kiválasztunk egy eseményt, a rendszer a program szövegében automatikusan kigenerálja megfelelő eseménykezelő sablonját. Ahogy a form esetében is, az eljárásokban elhelyezhetjük a gomb vagy a form tulajdonságait megváltoztató tulajdonságokat, majd futtathatjuk a projektet.

Az 6.26. ábrán látható példa esetében a gomb **Click** esemény kezelője **TForm1.Button1Click** eljárással a következő utasításokat tartalmazza:

- az ablak bal szélétől mért távolság legyen 150 pixel;
- az ablak felső szélétől mért távolság növekedjen 50 pixellel;
- a gombra íródjon ki a megfelelő szöveg;
- a gomb tulajdonságainak értékei váljanak *elérhetetlenné*!



6.26. ábra. A gomb Click eseményének kezelője





## A Pythont használóknak

Bármely eseménykezelő összeköthető a gombbal ugyanúgy, mint egy ablak a **bind()** metódussal. Például a 6.27 ábrán adott a **Click** eseménykezelő olyan gombra, amely az ablak sárga színre való beállítását tartalmazza.

```
from tkinter import *
def click(event):
    root['bg'] = 'yellow'
root = Tk()
root.geometry('600x400+350+200')
button = Button(text = 'Змінити', width = 15)
button.pack(pady = 30)
button.bind('<1>', click)
```

6.27. ábra. **Click** eseménykezelő gomb esetében

A gomb esetében az alapértelmezett esemény az egér bal gombjának kattintása. Ezért a gombbal könnyebben lehet társítani az esemény feldolgozásának módszerét:

- a gomb létrehozásakor megadjuk a megfelelő eseménykezelő nevét mint a **command** attribútum értékét;
- a gomb létrehozása előtt az eseménykezelőt leírjuk a megadott névvel; a címben nem szükséges feltüntetni a már megtörtént eseményre való hivatkozást – **event**.

A 6.28 ábrán adott a **Click** eseménykezelő gomb esetében – a *click()* nevű eljárás, amely az alábbi parancsokat tartalmaz:

- a gomb színének beállítása pirosra;
- *Arial* betűtípus beállítása 14 pixel mérettel;
- a megadott szöveg kiírása a gombon;
- a gomb állapotának beállítása *nem elérhetőre*.

```
from tkinter import *
def click():
    button['bg'] = 'red'
    button['font'] = 'Arial 14'
    button['text'] = 'Змінюються значення властивостей кнопки'
    button['state'] = 'disable'
root = Tk()
root.geometry('600x400+350+200')
button = Button(text = 'Вибери мене', command = click)
button.pack(pady = 30)
```

6.28. ábra. **Click** eseménykezelő gomb esetében



## Használjuk a számítógépet!



Végezzétek el a gyakorlati feladatot a <https://cutt.ly/ghdrpzl> linkről, vagy a QR-kód segítségével!

## ! Összefoglalás



### A Pascalt használóknak

Gombot úgy helyezhetünk el a formon **Lazarus** környezetben, hogy a **Standard**-eszköztár **Ok** gombjára duplán kattintunk.

A **Name** tulajdonság értéke megadja a gomb nevét objektumként. A **Height** és **Width** tulajdonságok értékei határozzák meg a gomb magasságát és szélességét. A **Caption** tulajdonság természetesen a gombon látható feliratra vonatkozik, a **Top** és **Left** tulajdonságok viszonyítási pontja pedig nem a képernyő, hanem a form felső, illetve bal széle.

Ezen tulajdonságok értékét a **Tulajdonságok** lapon módosíthatjuk az **Objektum-felügyelő** ablakot, mielőtt elindítanánk a projekt végrehajtását. Néhányuk értékét megváltoztathatjuk úgy, hogy elhúzzuk a gombot, vagy átméretezzük a határain lévő jelölők használatával. Mindezen tulajdonságok értékei a projekt végrehajtása során hozzárendelési parancsokkal is megváltoztathatók. Például: **Button1.Width := 100**, **Button1.Left := Button1.Left + 200**, **Button1.Caption := 'Végrehajtani'**.

A gomb esetében a **Color** tulajdonság megváltoztatása nem változtatja meg a gomb alapértelmezett színét.

Tekintsünk át még néhány tulajdonságot, amellyel mind a form, mind pedig a gomb rendelkezik:

- **Enabled** (*engedélyezett, hozzáférhető*) – ennek értéke határozza meg, hogy az objektum tulajdonságain végezhető-e műveletek. Értéke lehet **True** (igaz, vagyis engedélyezett) vagy **False** (hamis, azaz tiltott); az elérhetetlen gomb:

 ;

- **Visible** (*látható*) – értéke azt határozza meg, hogy az objektum látható-e az ablakban. Ennek a tulajdonságnak is csak két értéke lehet, a **True** (látható), illetve a **False** (*nem látható*).



### A Pythont használóknak

Ahhoz, hogy gombot helyezzünk el az ablakban, létre kell hoznunk egy új **Button** típusú objektumot, és hozzá kell kapcsolnunk ahhoz a változóhoz, amely meghatározza az objektum nevét.

A gomb létrehozásához és változóhoz való kapcsolásához az alábbi parancsot használjuk: **<gomb\_neve> = Button()**

Gomb létrehozásakor a tulajdonságainak értékei beállíthatók. A zárójelben vesszővel jelölik a párokat **<tulajdonság> = <érték>**. A gomb számára az alábbi tulajdonságok értékei állíthatók be: **text**, **width**, **height**, **bg**, **fg**, **font**, **state** stb.

A gomb ablakban való elhelyezéséhez a **pack()** vagy **place()** metódusok használhatóak. Bármely eseménykezelő összeköthető a gombbal ugyanúgy, mint egy ablak a **bind()** metódussal.

Ha egy **Click** eseménykezelési módot társítunk egy gombhoz, akkor a gomb létrehozásakor megadjuk a megfelelő eseménykezelő nevét mint a **command** attribútum értékét. A címben nem szükséges feltüntetni a már megtörtént eseményre való hivatkozást – **event**.



**Felelj a kérdésekre!**



**A Pascalt használóknak**

- 1<sup>o</sup>. Hogy helyezünk gombot a formra?
- 2<sup>o</sup>. Milyen tulajdonságai vannak a gombnak? Mit határoznak meg a gomb tulajdonságai?
- 3<sup>o</sup>. Miben különbözik a **Color** tulajdonság a form és a gomb esetében?
- 4<sup>o</sup>. Miben különbözik a **Caption** tulajdonság a form és a gomb esetében?
- 5<sup>o</sup>. Miben különböznek a **Top** és **Left** tulajdonságok a gomb és a form esetében?
- 6<sup>o</sup>. Milyen értékei lehetnek az **Enabled** és **Visible** tulajdonságoknak? Mit jelentenek ezek az értékek?



**A Pythont használóknak**

- 1<sup>o</sup>. Hogyan hozunk létre gombot?
- 2<sup>o</sup>. A gomb milyen tulajdonságait ismerjük? Mi határozza meg mindegyiket?
- 3<sup>o</sup>. Hogyan állítsuk be a gomb tulajdonságainak értékét?
- 4<sup>o</sup>. Hogyan helyezzünk el egy gombot egy ablakban?
- 5<sup>o</sup>. Milyen módon rendelhetünk eseménykezelőket egy gombhoz?
- 6<sup>o</sup>. Hogyan rendelhetünk egy gombhoz egy **Click** eseménykezelőt?



**Végezd el a feladatokat!**

1<sup>o</sup>. Hozz létre projektet, amelyben a gomb **Click** eseményének kezelője állítsa a gomb méretét 40 × 30 pixelre (Python szélessége – 30 karakter, magassága – 3 karakter), a gomb feliratában pedig elhelyezi lakóhelyetek nevét! Mentse a projektet a mappában létrehozott **feladat 6.3.1** mappába!



2<sup>o</sup>. Hozz létre projektet, amelyben a gomb **KeyPress** eseményének kezelője állítsa a gomb méretét 20 × 40 pixelre (Python szélessége – 20 karakter, magassága – 4 karakter), a gomb feliratában pedig elhelyezi a családnevedet! Mentse a projektet a mappában létrehozott **feladat 6.3.2** mappába!

3<sup>o</sup>. Hozz létre projektet, amelyben a gomb **MouseMove** eseményének kezelője növelje meg a gomb és a form bal széle közötti távolságot 20 pixellel és megváltoztatja a gomb feliratát! Mentse a projektet a mappában létrehozott **feladat 6.3.3** mappába!

4<sup>o</sup>. Hozz létre projektet, amelyben a gomb **Click** eseményének kezelője csökkentse a gomb és a form bal széle közötti távolságot 20 pixellel, és megváltoztatja a gomb méretét! Mentse a projektet a mappában létrehozott **feladat 6.3.4** mappába!

5<sup>o</sup>. Hozz létre projektet, amelyben a gomb **Click** eseményének kezelője állítsa a gomb méretét 50 × 50 pixelre (Python szélessége – 50 karakter, magassága – 5 karakter), a felirat színét barnára állítsa, a gomb feliratában pedig elhelyezi a *Projekttem* szöveget! A **MouseMove** esemény hatására elmozdítja a gombot 30 pixellel jobbra és 40 pixellel le és már elérhetetlen lesz (Python a háttér sárga színt kap, állapota pedig – *tiltott*)! Mentse a projektet a mappában létrehozott **feladat 6.3.5** mappába!

**6\***. Hozz létre egy projektet, amelynek formjában helyezd el három gombot! A futtatás kezdetén az első gomb legyen *látható* és *elérhető*, a második *látható* és *elérhetetlen*, a harmadik *láthatatlan* és *elérhetetlen*! Mindhárom gomb **Click** eseményéhez rendelj eseménykezelőt a következők szerint:

- 1) az első gomb esetében tedd láthatatlanná és elérhetetlenné a gombot, láthatóvá, de elérhetővé a második gombot, láthatóvá, de elérhetetlenné a harmadikat;
- 2) a második gomb esetében tedd láthatatlanná és elérhetetlenné a gombot, láthatóvá és elérhetővé a harmadik gombot, láthatóvá, de elérhetetlenné az elsőt;
- 3) a harmadik gomb esetében tedd láthatatlanná és elérhetetlenné a gombot, láthatóvá és elérhetővé az első gombot, láthatóvá, de elérhetetlenné a másodikat!

Mentsd a projektet a mappádban létrehozott **feladat 6.3.6** mappába!

### 6.4. A címke



1. A gomb mely tulajdonságait ismeritek? Hogy változtathatjuk meg ezek értékeit?
2. Milyen események következhetnek be egy gombbal? Mit jelentenek ezek?
3. Milyen különbségek vannak a form és a gomb tulajdonságai között?
4. Milyen a különbség az eseményvégrehajtó létrehozásában a formra és a gombra nézve?

#### A címke tulajdonságai

A formon elhelyezhető objektumok közül most a *címkével* foglalkozunk. A **címkét** szöveges üzenetek megjelenítésére használjuk.

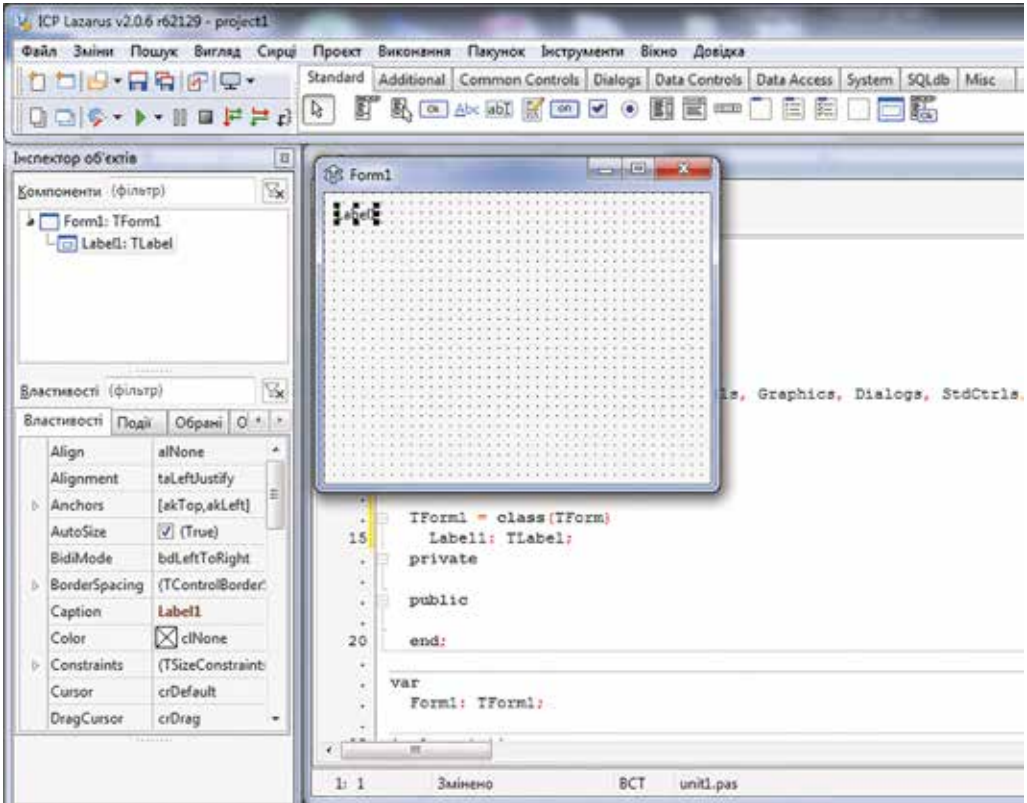


#### A Pascalt használóknak

A **Címke** (label) elhelyezése a gombhoz hasonlóan történik: kattintsunk duplán a **Standard** eszköztár **TLabel** | **Abc** | gombjára a **Lazarus** környezetben! A címke megjelenik a form bal felső sarkában (6.29. ábra), a neve bekerül az **Objektum felügyelő** ablakába. A címkét áthúzhatjuk a form tetszőleges helyére.

A címke kijelölése is a szokott módon, az ablakban vagy az **Objektum felügyelő**-ben történhet. Az **objektum felügyelő** ilyenkor a címke tulajdonságait mutatja. A **Tulajdonságok** értékeit itt változtathatjuk meg. Itt is vannak olyan tulajdonságok, amelyeket begépelünk, másokat a listából választjuk. A címke helyzetét áthúzással változtatjuk meg.

A **Top**, **Left**, **Name**, **Enabled**, **Visible**, **Font** tulajdonságok ugyanazt jelentik, mint a gombok esetében. A **Color** tulajdonság kihatással van a címke megjelenésére. Ezeknek a tulajdonságoknak az értékei megváltoztathatók a **Tulajdonságok** lapon, mielőtt elindítanánk a projektet, vagy parancsokat rendelhetünk a projekt végrehajtása során, például: **Label1.Top := 150**, **Label1.Left := Label1.Left + 150**, **Label1.Enabled := True**.



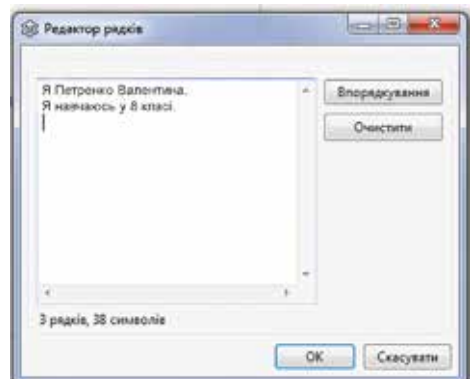
6.29. ábra. Címke a formon

A gombtól eltérően a címke **Height** és **Width** tulajdonságait nem változtathatjuk meg az egér segítségével vagy a **Tulajdonságlapon**. A címke szélességét és hosszát annak a szövegnek a mérete határozza meg, amit a címke tartalmaz. A **Caption** tulajdonság értékének változása, például **Label1.Caption := 'A nyolcadik osztályban tanulok'**, megváltoztatja a címke méreteit.

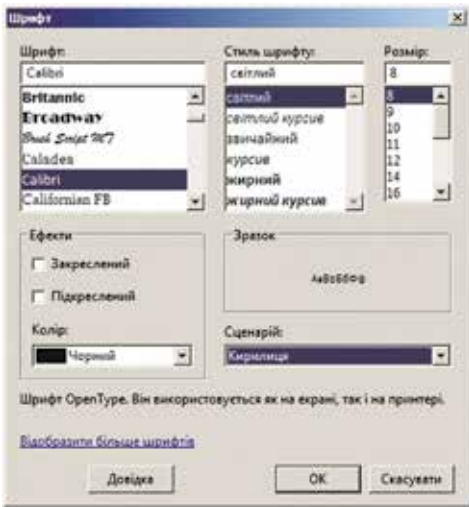
De ha a **False** értéket a címke **AutoSize** (ang. **Automatic Size** – automatikus méret) tulajdonságához állítjuk be, akkor annak szélessége és magassága a benne lévő szövegtől függetlenül megváltoztatható.

A címke **Caption** tulajdonságát a **Karakterlánc-szerkesztőben** adhatjuk meg (6.30. ábra). Az ablak megnyitásához aktuálissá kell tennie a **Caption** mezőt, és ki kell választania a **...** gombot. Ebben az ablakban többsoros szöveget írhatunk be.

A **Font** (betűtípus) értéke határozza meg, milyen betűtípussal jelennek meg a feliratok



6.30. ábra. A Karakterlánc-szerkesztő



6.31. ábra. Betűtípus ablak a karakterek tulajdonságai értékeinek beállítására

az objektumon. Az eddigi tulajdonságoktól eltérően ez egy **komplex** tulajdonság. Ez azt jelenti, hogy értéke több részből (**Color, Height, Name, Size**) áll. Ezek mindegyikének értékét a **Font**tól balra található **>** gomb megnyomása után megnyíló mezőbe érhetjük el. A tulajdonság értékét a **Font**tól jobbra található **Betűtípus** **...** gombra kattintva is elérhetjük (6.31. ábra).

A betűtípus kiválasztásakor ügyelni kell az **Írásrendszer** listára. Ha ebben a listában nincs **cirill** elem, akkor a felirat szövege nem jelenik meg helyesen ukránul.

A tulajdonságok értékét a projekt futása közben is megváltoztathatjuk. A karakterek színe megváltoztatható, például a következő parancs segítségével: **Label1.Font.Color := cIBlue**.



### A Pythont használóknak

Címkék is elhelyezhetők a **Python** nyelven létrehozott projekt ablakában. Erre a projekt szövegében szükség van.

1. A Címke létrehozása parancs bevitelére.
2. A tulajdonságok értékeinek beállítására.
3. A címke elhelyezésére az ablakban.

A címke létrehozásának parancsa:

```
<címke neve> = Label(text = '<címke szövege>')
```

(ang. *label* – címke, felirat). Például ahhoz hogy *label* nevű címként hozzunk létre *A nyolcadik osztályban tanulok* szöveggel az alábbi parancsot kell végrehajtani:

```
label = Label(text = 'A nyolcadik osztályban tanulok')
```

A címke létrehozásával egyidejűleg megadható egyes tulajdonságainak értéke is, például **bg** – címke háttér színe, **fg** – karakterek színe, **font** – betűtípus, a karakterek mérete és kiemelés típusa (**bold** – félkövérre, **italic** – dőlt, **underline** – aláhúzott), **height** – a címke magassága, **width** – a címke szélessége és egyebek. Ehhez a címke létrehozásának parancsában kör zárójelben az alábbi párt kell bevinni **<tulajdonság> = <érték>**. Például, az alábbi parancs végrehajtása után: **label = Label(text = 'A nyolcadik osztályban tanulok', bg = 'yellow', fg = 'green', font = 'Arial 14')** egy címke jön létre, zöld betűkkel sárga hátteren *A nyolcadik osztályban tanulok* szöveggel, melynek betűtípusa *Arial* és a karakterek mérete *14*.

A tulajdonságok értéke külön parancs formában is megadható:

```
<címke neve>[<tulajdonság neve>] = <tulajdonság értéke>
```

Például a *label* nevű címke szélességét 25 karakter hosszúra a **label['width'] = 25** paranccsal állíthatjuk be, a *Times New Roman* betűtípust 28-as karakter mérettel – **label['font'] = ('Times New Roman', 28)** paranccsal, a karakterek színét kékre – **label['fg'] = 'blue'** paranccsal. Az ilyen típusú parancsok használhatóak a címke tulajdonsága értékének megváltoztatására a projekt végrehajtása során.



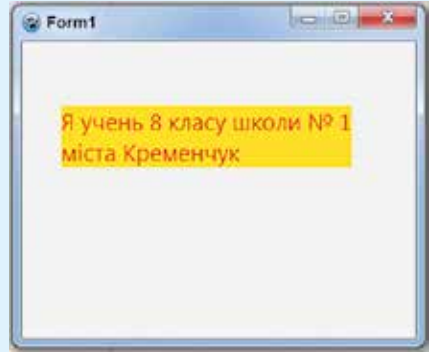


### Ha többet szeretnél tudni

Ha azt szeretnénk, hogy a címke szövege a szükséges helyeken új sorban folytatódjon, #13 új sor szimbólumot kell elhelyezni a szövegben. Az 6.34. ábrán látható szöveget például futásidőben a következő utasítással írhatjuk ki:

```
Label1.Caption := 'Я учень 8 класу  
школи №1' + #13 + 'міста Кременчук'.
```

A címke szövege új sorba kerülhet, ha a **WordWrap** (ang. *word wrap* – sortörés) tulajdonságának értékét **True**-ra, az **AutoSize** tulajdonság értékét pedig **False**-ra állítjuk és megadjuk a címke kívánt méretét. A szöveg ugyanúgy fog új sorba kerülni, mint a szövegszerkesztőkben.



6.34. ábra. A projekt eljárás futásának eredménye sortöréssel kiegészítve



### A Pythont használóknak

A címkéhez, akárcsak a projektablak vagy gomb esetében, eljárás társítható – valamilyen eseménykezelő a **bind()** metódussal:

```
<címke neve>.bind(<esemény>, <eljárás neve>)
```

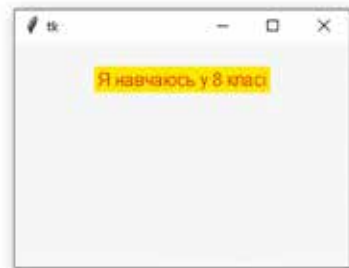
Külön kell írni az eljárás szövegét – az eseménykezelőt a megadott névvel.

A műveletek sorrendje a következő lehet:

1. Létrehozuk a címkét és beállítjuk tulajdonságainak értékét.
2. Hozzácsatoljuk a címkét az eseménykezelőhöz az esemény és az eljárás nevének megadásával.
3. Létrehozuk az eljárást – eseménykezelőt, a megadott névvel.
4. Beírjuk az utasításokat az eljárásba az objektumok tulajdonságainak kívánt értékeinek beállításához.

Például a *label* nevű címkéhez kapcsolt **Click** esemény bekövetkezése után az adott címke háttérszíne *sárgává* válik, a karakterek színe – *pirossá*, a karakterek mérete – *14*, valamint kiíródik a megadott szöveg. A 6.35. ábrán adott a projekt szövege, valamint az eljárás eredménye – a **Click** eseménykezelő címke esetében.

```
from tkinter import *
def label_click(event):
    label['text'] = 'Я навчаюсь у 8 класі'
    label['font'] = 14
    label['fg'] = 'red'
    label['bg'] = 'yellow'
root=Tk()
root.geometry('300x200')
label = Label(text = 'Це напис')
label.bind('<l>', label_click)
label.pack(pady = 20)
```

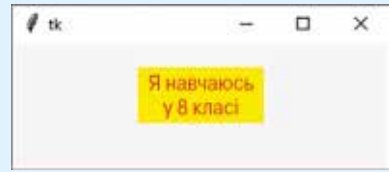


6.35. ábra. A **Click** eseménykezelő címke esetében valamint az eljárás eredménye



### Ha többet szeretnél tudni

A címke szövegét több sorba is kiírathatjuk. Ehhez oda, ahol kezdődni fog az új sor, `\n` karaktereket kell tenni. Például a `Label = Label(text = 'Я навчаюсь \n у 8 класі')` címkébe az első sorba a *Я навчаюсь* szöveg, a másodikba *у 8 класі* szöveg kerül. Ezen parancsok végrehajtásának eredményét a 6.36. ábra szemlélteti.



6.36. ábra. Címke szövegének kiírása két sorba



### Használjuk a számítógépet!



Végezzék el a gyakorlati feladatot a <https://cutt.ly/mhdraDg> linkről, vagy a QR-kód segítségével!



### Összefoglalás

A **címkét** szöveges üzenetek megjelenítésére használjuk.



### A Pascalt használóknak

A **Címke** elhelyezése a gombhoz hasonlóan történik: kattintsunk duplán a **Standard** eszköztár **Tlabel** | **Abc** | gombjára a **Lazarus** környezetben!

A **Top**, **Left**, **Name**, **Enabled**, **Visible**, **Font** tulajdonságok ugyanazt jelentik, mint a gombok esetében. A **Color** tulajdonság kihatással van a címke megjelenésére. Ezeknek a tulajdonságoknak az értékei megváltoztathatók a **Tulajdonságok** lapon, mielőtt elindítanánk a projektet, vagy parancsokat rendelhetünk a projekt végrehajtása során, például: `Label1.Top := 150`, `Label1.Left := Label1.Left + 150`, `Label1.Enabled := True`.

A gombtól eltérően a címke **Height** és **Width** tulajdonságait nem változtathatjuk meg az egér segítségével vagy a **Tulajdonságlapon**. A címke szélességét és hosszát annak a szövegnek a mérete határozza meg, amit a címke tartalmaz. A **Caption** tulajdonság értékének változása, például `Label1.Caption := 'A nyolcadik osztályban tanulok'` megváltoztatja a címke méreteit. De ha a **False** értéket a címke **AutoSize** (ang. **Automatic Size** – automatikus méret) tulajdonságához állítjuk be, akkor annak szélessége és magassága a benne lévő szövegtől függetlenül megváltoztatható.

A címke **Caption** tulajdonságát a **Karakterlánc-szerkesztőben** adhatjuk meg.

A **Font** (betűtípus) értéke határozza meg, milyen betűtípussal jelennek meg a feliratok az objektumon. Az eddigi tulajdonságoktól eltérően ez egy **komplex** tulajdonság. Ez azt jelenti, hogy értéke több részből (**Color**, **Height**, **Name**, **Size**) áll. A tulajdonságok értékét a projekt futása közben is megváltoztathatjuk. A karakterek színe megváltoztatható, például a következő parancs segítségével: `Label1.Font.Color := clBlue`.

A címke kijelölése is a szokott módon, az ablakban vagy az **Objektum felügyelőben** történhet.

Ha kijelöljük a címkét, az **Objektum felügyelő** ilyenkor a címke tulajdonságait mutatja. A **Tulajdonságok** értékeit itt változtathatjuk meg.



### A Pythont használóknak

A címke létrehozásának parancsa a **Python** nyelven a következő:

```
<címke neve> = Label(text = '<címke szövege>')
```

A címke létrehozásával egyidejűleg megadható egyes tulajdonságainak értéke is, például: **bg** – címke háttér színe, **fg** – karakterek színe, **font** – betűtípus, a karakterek mérete és a kiemelés típusa (**bold** – félkövérre, **italic** – dőlt, **underline** – aláhúzott), **height** – a címke magassága, **width** – a címke szélessége és egyebek. Ehhez a címke létrehozásának parancsában kör zárójelben az alábbi párt kell bevinni **<tulajdonság> = <érték>**.

A tulajdonságok értéke külön parancs formában is megadható:

```
<címke neve>[<tulajdonság neve>] = <tulajdonság értéke>
```

A címke elhelyezéséhez a projektablakban a **pack()** és **place()** metódusokat használják.

A címkéhez eljárás társítható – valamilyen eseménykezelő a **bind()** metódussal:

```
<címke neve>.bind('<esemény>', <eljárás neve>)
```

Külön kell írni az eljárás szövegét – az eseménykezelőt a megadott névvel.



### Felelj a kérdésekre!

1°. Mire szolgál a címke?



### A Pascalt használóknak

2°. Hogy helyezünk el címkét a formon?

3°. Mit jelentenek a **Top**, **Left** és **Name** tulajdonságok a címke esetében?

4°. Mit jelentenek az **Enabled** és **Visible** tulajdonságok a címke esetében?

5°. Mit jelentenek a **Font** komplex tulajdonság elemei a címke esetében?

6°. Miben különbözik a **Color** tulajdonság gomb és címke esetében?

7°. Miben különböznek a **Top** és **Left** tulajdonságok címke és form esetében?

8°. Miben különbözik a **Height** és **Width** tulajdonságok címke és form esetében?

9°. Hogy változtathatjuk meg a címke feliratát?

10°. Hogyan hozhatunk létre eseménykezelőt egy felirathoz?



### A Pythont használóknak

2°. Hogyan készíthetünk címkét?

3°. Hogyan lehet megállapítani egy címke tulajdonságainak értékét?

4°. A címke mely tulajdonságait ismeritek? Magyarazzátok el őket!

5°. Hogyan lehet feliratot elhelyezni egy ablakban?

6°. Hogyan lehet társítani egy eseménykezelőt egy címkéhez?

7°. Mi a különbség a címke és az ablak tulajdonságai között?

8°. Mi a különbség a címke és a gomb tulajdonságai között?



**Végezd el a feladatokat!**

- 1<sup>o</sup>. Hozz létre projektet, amelyben a címke **Click** eseményének eredményeképp:
- a címke és a form teteje közötti távolság *120 pixel* lesz, a címke szövegébe beírja annak az utcának a nevét, amelyen az iskolátok található;
  - a címke szélessége *30 karakter*, és a felirat szövege annak az utcának a nevére változik, amelyen az iskolátok található!

Mentsd a projektet a mappádban létrehozott **feladat 6.4.1.** mappába!



- 2<sup>o</sup>. Hozz létre projektet, amelyben a címke **DbIClick** eseményének eredményeképp:

- a címke és a form bal oldala közötti távolság *80 pixel* lesz, a címke szövegébe beírja a neved;
- a címke magassága *5 karakter*, és a felirat szövege a nevedre változik!

Mentsd a projektet a mappádban létrehozott **feladat 6.4.2.** mappába!

- 3<sup>o</sup>. Hozz létre projektet, amelyben a címke **MouseMove** eseményének eredményeképp:

- a címke szövegének színe piros lesz, és *20 pixellel* nő a címke és a form teteje közötti távolság;
- a címke új szöveget kap, amely piros színű és a felirat szélessége *5 karakterrel* megnő!

Mentsd a projektet a mappádban létrehozott **feladat 6.4.3.** mappába!



- 4<sup>o</sup>. Hozz létre projektet, amelyben a címke **Click** eseményének eredményeképp:

- a címke és a form bal széle közötti távolság *20 pixellel* nő, a betűméret 12-es lesz, a betűszín *kék* lesz, a címkébe pedig beíródik az iskolád neve;
- csökkentsük a címke szélességét *2 karakterrel*, a karakterek mérete 12-es, színe *kék* legyen, és az iskolátok neve jelenjen meg a címkében!

Mentsd a projektet a mappádban létrehozott **feladat 6.4.4.** mappába!

- 5<sup>o</sup>. Hozz létre projektet, amelyben a címke **Click** eseményének eredményeképp a címke háttere barnára vált, a címkébe beíródik a *Projekt* szöveg, a **MouseMove** hatására:

- elmozdul *30 pixellel* jobbra és *40 pixellel* le;
- a címke szélessége és magassága 3 karakterrel megnő!

Mentsd a projektet a mappádban létrehozott **feladat 6.4.5.** mappába!

- 6<sup>o</sup>. Hozz létre projektet, amelyben a címke **Click** eseményének eredményeképp:

- a címke *60 pixellel* jobbra és *10 pixellel* felfelé mozdul el, a címkébe pedig zöld színnel és szürke háttérrel beíródik a teljes neved;
- a felirat szélessége *2 karakterrel* csökken, a felirat magassága *3 karakterrel* növekszik, a vezeték- és utónév zöld színnel jelenik meg szürke háttéren!

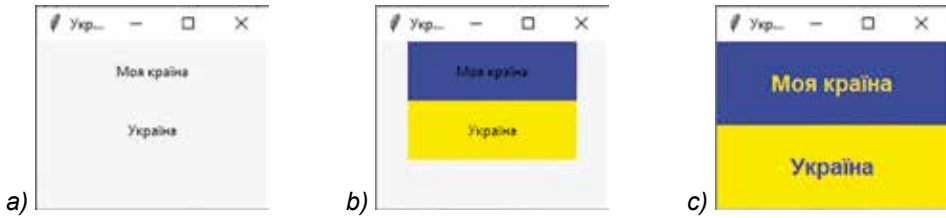
Mentsd a projektet a mappádban létrehozott **feladat 6.4.6.** mappába!



- 7<sup>o</sup>. Hozz létre projektet, amelyben a címke **Click** eseményének eredményeképp a címke *60 pixellel* jobbra és *80 pixellel* felfelé mozdul el, a címkébe szürke háttéren zöld betűkkel beíródik az iskolád neve, kék színnel, sárga háttérrel! Mentsd a projektet a mappádban létrehozott **feladat 6.4.7.** mappába!

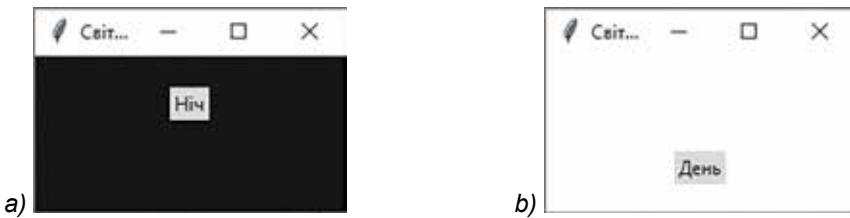
- 8<sup>o</sup>. Hozz létre egy projektet, amelyben a formban két feliratot helyezz el: *Országom* és *Ukrajna*! Állítsd be címsornak, hogy *Ukrajna*! A **Click** eseményének eredményeképp az első címke háttere *kék*re, a második címke háttere pedig *sárgá*ra vált, a karakterek betűtípusa és méretei mindkét esetben *Arial, 14-es, félkövér*.

A projekt formjainak kinézetét a 6.37. ábra szemlélteti. Mentsd a projektet a mappádban létrehozott **feladat 6.4.8.** mappába!



6.37. ábra. A formok kinézetete: a) indítás után; b) a **Click** esemény eredménye az első címkére; c) a **Click** esemény eredménye a második címkére

9\*. Hozz létre egy projektet, amelyben a formban két feliratot helyezz el: *Nappal* és *Éjszaka!* Állítsd be címsornak, hogy *Nappal!* A projekt elindítása után a form háttérszíne *fekete*, az első címke háttérszíne pedig *világosszürke*, a második címke *láthatatlan* legyen! A címke **Click** eseményének eredményeképpen a címke háttere fehérre vált, az első címke *eltűnik*, a második címke háttere pedig *világosszürke* legyen. A **Click** eseményének eredményeképpen a form második címkéje térjen vissza a kezdeti állapotba. A projekt formjainak kinézetét a 6.38. ábra szemlélteti. Mentsd a projektet a mappádban létrehozott **feladat 6.4.9.** mappába!



6.38. ábra. A projekt formja: a) kezdőhelyzet és a **Click** esemény eredménye a második címkére; b) a **Click** esemény eredménye az első címkére

## 9. SZÁMÚ GYAKORLATI MUNKA

### Gombokat és címkéket tartalmazó projektek létrehozása

**Figyelem!** A számítógép használata során tartsd be a balesetvédelmi szabályokat és az egészségügyi előírásokat!



#### A Pascalt használóknak

1. Indítsd el a **Lazarus** rendszert!
2. Helyezz el a formon két gombot és egy címkét!
3. Az első gomb tulajdonságai legyenek a következők:
  - szélesség – 60 pixel;
  - magasság – 20 pixel;
  - távolság a form bal szélétől – 120 pixel;
  - távolság a form tetejétől – 100 pixel;
  - felirat – *Form!*





4. A második gomb tulajdonságai legyenek a következők:
  - szélesség – *100 pixel*;
  - magasság – *30 pixel*;
  - távolság a form bal szélétől – *300 pixel*;
  - távolság a form tetejétől – *100 pixel*;
  - felirat – *Címke!*
5. Az címke tulajdonságai legyenek a következők:
  - távolság a form bal szélétől – *150 pixel*;
  - távolság a form tetejétől – *200 pixel*;
  - felirat – az osztályod neve!
6. Mentsd a projektet a mappádban létrehozott **Gyakorlati 9** mappába!
7. Hozd létre az első gomb **Click** eseményének kezelőjét, ami a form szélességét *800 pixelre*, magasságát *400 pixelre* állítja be, a form hátterét *zöldre*, a gomb szélességét *200-ra* állítja, a második gomb szélességét *10 pixellel* növeli, *50 pixellel* balra, *30 pixellel* felfelé mozdítja, majd tulajdonságait zárolja!
8. Hozd létre a második gomb **MouseMove** eseményének kezelőjét, ami a címke hátterét *sárgára* állítja, távolságát a form tetejétől *200 pixelre* állítja, csökkenti a címke és a form bal széle közötti távolságot *50 pixellel*, a szöveg színét *pirosra* változtatja és megjeleníti a *Pascal programozási nyelvet tanuljuk* szöveget!
9. Mentsd el a projekt új változatát ugyanazon néven ugyanarra a helyre!



## A Pythont használóknak

1. Hozzatok létre egy projektet, melyben az ablak mérete *400x300 pixel* és helyeztek el benne két gombot és egy címkét!
2. Állítsátok be az alábbi tulajdonságértékeket az első gombra:
  - szélesség – *20 karakter*;
  - magasság – *2 karakter*;
  - háttérszín – *szürke*;
  - karakterszín – *sárga*;
  - szöveg a gombon – *Projekt ablak!*
3. Állítsátok be az alábbi tulajdonságértékeket a második gombra:
  - szélesség – *10 karakter*;
  - magasság – *3 karakter*;
  - háttérszín – *szürke*;
  - karakterszín – *fehér*;
  - szöveg a gombon – *Címke!*
4. Állítsátok be az alábbi tulajdonságértékeket a címkére:
  - karakterszín – *kék*;
  - a címke szövege – az osztály neve;
  - betűtípus – *Arial 14!*
5. Hozzatok létre egy **Click** eseménykezelőt az első gombhoz, amely beállítja a projekt ablak szélességét – *800 pixelre*, a projekt ablak magasságát – *400 pixelre*, az ablak háttérszínét – *zöldre*, az első gomb szélességét – *30 karakterre*, növeli a második gomb szélességét *5 karakterrel*, megváltoztatja az első gomb színét *kékre* és a rajta lévő karakterek színét *fehérre*, a gomb állapotát *tiltottá* (nem elérhetővé) teszi!
6. Hozzatok létre egy **MouseMove** eseménykezelőt a második gombhoz, amely beállítja a címke hátterét – *pirosra*, a címke szélességét – *40 karakterre*, megnöveli a címke magasságát *5 karakterrel*, a címke karaktereinek színét *sárgára* állítja és a kiírja a címkében, hogy *A Python programozási nyelvet tanuljuk!*
7. Mentsétek el a projektet a mappátokba **Gyakorlati 9.py** néven!

## 6.5. Beviteli mező. Projektek bemeneti adatokkal és végeredményekkel

1. Milyen tulajdonságai vannak a *címkének*? Mit mutatnak ezek a tulajdonságok?
2. Mire szolgál az értékadó utasítás? Mi ennek az utasításnak az általános alakja?
3. Milyen állandókat (konstansokat) használtak a matematika-, a fizika- és a kémiaórákon? Miben különböznek az állandó mennyiségek a változó mennyiségektől?

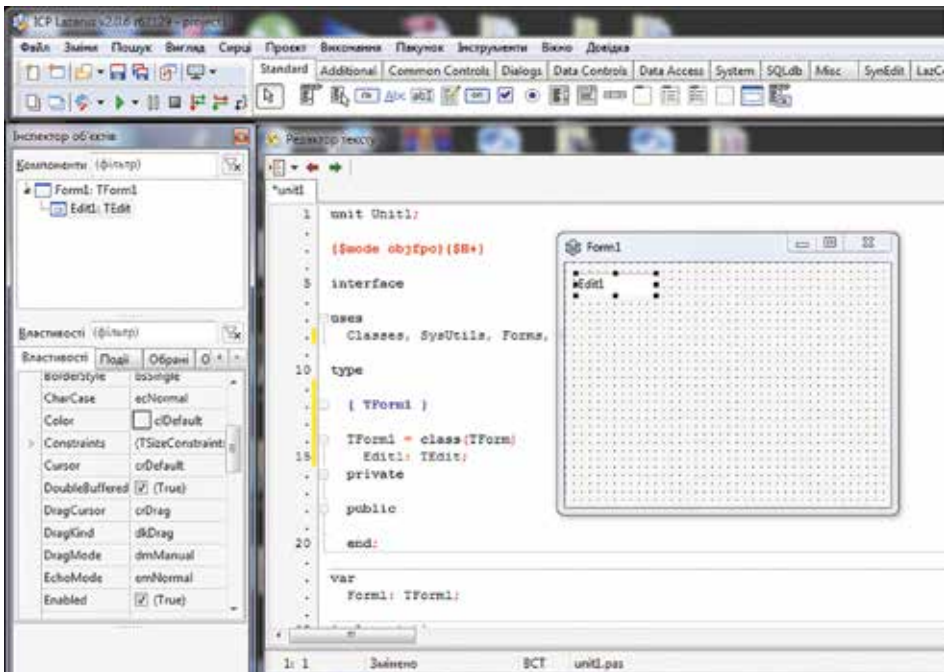
### A mező és tulajdonságai

A következő formra helyezhető komponens, amit megvizsgálunk, a **beviteli mező** lesz. A **mező** – az a komponens, amely adatok bevitelére szolgál. A mezőt eredmények bevitelére is használhatjuk.



### A Pascalt használóknak

A mezőt **Lazarus** környezetben úgy helyezzük el a formra, hogy duplán kattintunk a **TEdit** `ab1` elemére, ami a **Standart** fülben található. Ekkor a form bal felső sarkában megjelenik a beviteli mező, az **Objektum felügyelő Komponensek** ablakához hozzáadódik a mező (6.39. ábra). A mezőt is szabadon áthelyezhetjük a formon belül.



6.39. ábra. Beviteli mező a formon



A mezőt is a szokott módon, a formon vagy az **Objektum felügyelő** ablakában az egér bal gombjával jelöljük ki. Ekkor a felügyelő ablakában a **Tulajdonság** ablak a mező tulajdonságait tartalmazza. Ezeket a tulajdonságokat is szerkeszthetjük. Egyes tulajdonságok értékeit a **Tulajdonságok** ablak beviteli mezőibe kell begépelni, mások esetében a legördülő listákból választhatunk; amikor a mezőt mozgatjuk a formon, szintén megváltoznak a megfelelő tulajdonságai.

A **Top, Left, Name, Color, Enabled, Visible, Font** tulajdonságok értelmezése a mező és a címke esetében ugyanaz. A **Width** tulajdonságokat a **Tulajdonság** lapon és a méretezőmarkerek mozgatásával állíthatjuk be, ahogy az a gomb esetében is volt. A **Height** tulajdonságokat nem tudjuk megváltoztatni, ha az **AutoSize** tulajdonság értéke **True**.

Ennek a tulajdonságnak az értéke a projekt végrehajtása során is beállítható egy értékadó utasítással, például az **Edit1.Top := 200, Edit1.Left := Edit1.Left – 50, Edit1.Enabled := False**.

A beviteli mezőben olvasható szöveg megegyezik a **Text** (ang. *text* – szöveg) tulajdonság értékével. Ennek a tulajdonságnak az értékét a **Tulajdonság** lap megfelelő mezőjébe írhatjuk be. A projekt futtatása előtt célszerű ezt a mezőt üresen hagyni. Ennek a tulajdonságnak az értéke a projekt végrehajtása során is beállítható egy értékadó utasítással, például az **Edit1.Text := 'Szövegmező'**.

A címke megjelenítheti például a mezőben lévő szöveget. Ehhez futtatnia kell a **Label1.Caption := Edit1.Text** parancsot.

*Itt jegyezzük meg, hogy:* ha a mező szélessége kisebb, mint a bevitt vagy kimenő szöveg hossza, akkor a felirattal ellentétben az **AutoSize** tulajdonság értékétől függetlenül, a mező szélessége nem nő automatikusan, és a szöveg egyes részei a mezőn kívül lesznek, és láthatatlanná válnak a felhasználó számára.



## A Pythont használóknak

A **Python**ban a beviteli mező létrehozására szolgáló parancs így néz ki:

**<mező neve> = Entry(<tulajdonságok és értékeik halmaza>)**

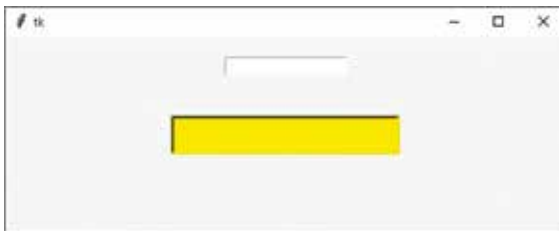
(ang. *entry* – bejegyzés, belépés).

A gombhoz és a címkéhez hasonlóan a beviteli mező rendelkezik **width, bg, fg, font** tulajdonságokkal, viszont a **height** hiányzik. A beviteli mező magasságát a benne lévő karakterek magassága határozza meg. A beviteli mezőnél beállítható a keretvastagság, mint a **bd** tulajdonság értéke.

A beviteli mező ablakban való elhelyezéséhez a **pack()** és **place()** metódusokat használhatjuk.

A 6.40 ábra két beviteli mezővel rendelkező projektablakot mutat. A felső beviteli mező az **entry1 = Entry()** parancssal jött létre – alapértelmezett tulajdonságértékekkel, az alsó beviteli mező pedig a:

**entry2 = Entry(width = 15, bg = 'yellow', fg = 'green', font = 'Arial 20', bd = 3)** parancssal.



6.40. ábra. Projektablak két beviteli mezővel

Ahhoz, hogy a mezőben szöveg kerüljön kiírásra, az alábbi parancsot használják:

**<mező neve>.insert(<pozíció>, <szöveg>),**

ahol a *pozíció* – a mezőben lévő hely, ahonnan kezdve a szöveg beillesztésre kerül. Ha a mező üres, akkor a szöveg a 0 pozícióba kerül, még akkor is, ha a parancsban másik pozíció van megadva.

Ha a mezőben már van némi tartalom, akkor a tartalom bármely helyétől beszúrható szöveg. Ahhoz, hogy a szöveg már a meglévő tartalom végére kerüljön, az **END** pozíciót adjuk meg.

Például ahhoz, hogy az **entry** nevű üres mezőbe *A 8. osztályban tanulok* szöveget írjuk ki, az alábbi parancsot kell végrehajtanunk:

**entry.insert(0, 'A 8. osztályban tanulok')**

Ahhoz, hogy megváltoztassuk ennek a mezőnek a tartalmát *A 8. A osztályban tanulok* szövegre, az előző parancs után hozzá kell adni az alábbi:

**entry.insert(14, 'A').**

Ahhoz pedig, hogy a mezőben a *A 8. A osztályban tanulok az 1. számú iskolában* szöveget kapjuk, az előző parancsok után hozzáadhatjuk a következőt:

**entry.insert(END, ' az 1.számú iskolában')**.

A mező tartalmát a **get()** metódussal kaphatjuk meg. Például ahhoz, hogy a *label* nevű címkébe kiírjuk az *entry* nevű mező tartalmát, az alábbi parancsot kell végrehajtani:

**label['text'] = entry.get().**

Az **entry** nevű mező tartalmának törlését az alábbi parancssal hajthatjuk végre:

**entry.delete(0, END).**

A projekt végrehajtása során előfordul, hogy célszerű a kurzort közvetlenül a kezdés után a mező közepére helyezni. Ebben az esetben azt mondják, hogy a mező fókuszbán van. Ehhez a **focus\_set()** metódust használjuk. Például az **entry.focus\_set()** parancs futtatásával a kurzor az **entry** nevű mező közepére kerül.

### **A beviteli mező eseménykezelői**

A mező, csakúgy, mint más komponensek, a vele előforduló különféle események kezelőjéhez társíthatók.



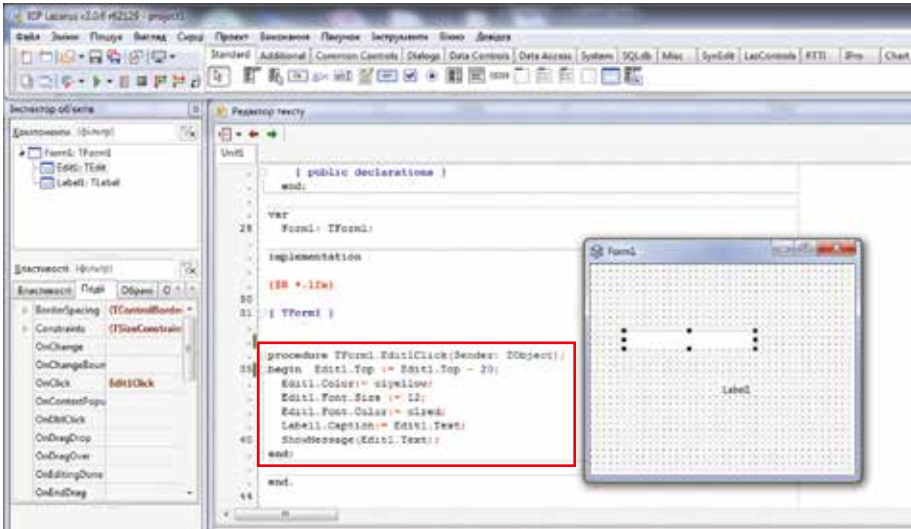
### **A Pascalt használóknak**

Ha kijelölünk egy mezőt, akkor az **Objektum felügyelő** ablak **Események** fülén, mint a korábban tárgyalt komponenseknél, kiválaszthatunk eseményt, amelyhez létrehozhatunk egy eljárást (hasonlóan a formhoz, a gombhoz és a címkéhez), amely az esemény kezelője lesz.

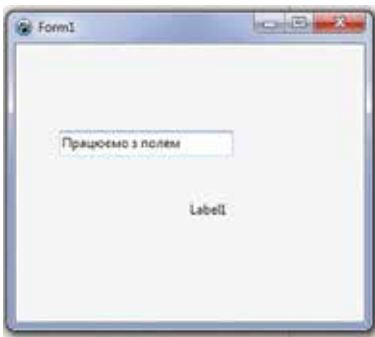
Például a 6.41 ábra egy eljárást tartalmaz – a mező **Click eseménykezelője** – a következő utasításokkal:

- 20 *pixel* csökken a mező teteje és a form teteje közötti távolság;
- a mező *sárga* hátterének beállítása;
- a mező szöveges karaktereinek a mérete 12-es;
- a mező szöveges karaktereinek a színe *piros*;
- a mezőbe, a címkébe és az üzenetablakba beírt szöveg megjelenítése.

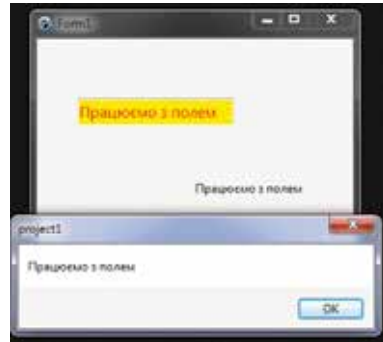
Miután elindítottuk a projektet a végrehajtáshoz, bármilyen szöveget beírhatunk a mezőbe (a 6.42. ábrán a mezőbe szereplő szöveg: *Mezőkkel dolgozunk*), helyezük a egérmutatót a mezőben, és kattintsunk az egér bal gombjával! Az eljárás eredménye – a mező **Click** eseménykezelőjét a 6.43. ábra mutatja. A mező háttérszíne *sárgává* változik, a karakterek színe – *piros*, a karakterek mérete – 12-es, a mezőbe beírt szöveg megjelenik a feliratban és az üzenetablakban.



6.41. ábra. A mező **Click** eseménykezelője



6.42. ábra. A form kinézete a szöveg mezőbe történő bevitele után, a **Click** esemény bekövetkeztéig



6.43. ábra. A form kinézete a **Click** esemény bekövetkeztekor



## A Pythont használóknak

Akárcsak a többi ablakobjektumhoz, a mezőhöz is hozzákapcsolhatunk eljárásokat, eseménykezelőket a **bind ()** metódus használatával.

Például a 6.44 ábrán látható egy olyan projekt szövege, melyben a **Click** eseménykezelő a mezőhöz van kapcsolva:

- a mező hátterszínének beállítása *sárgára*;
- a mező karakterei színének beállítása *pirosra*;
- a mező karakterei méretének beállítása *12 pixelre*;
- a mezőbe beírt szöveg kiírása a címkébe;
- a mezőbe beírt szöveg kiírása az üzenetablakba.

A projekt elindítása után a mezőbe beírhatunk bármilyen szöveget (a 6.45 ábrán a bal oldali mezőben a *Mezővel dolgozunk* szöveg van beírva), majd vigyük a kurzort a mezőre, és kattintsunk az egér bal gombjára! A mező **Click** eseménykezelőjének eredményét a 6.45 ábra jobb oldalán láthatjuk.

```

from tkinter import *
from tkinter.messagebox import *

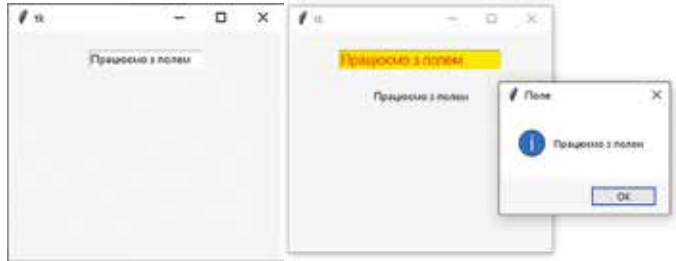
def click(event):
    entry['bg'] = 'yellow'
    entry['fg'] = 'red'
    entry['font'] = 12
    label['text'] = entry.get()
    showinfo("Info", entry.get())

root = Tk()
root.geometry("300x250")

entry = Entry()
entry.bind('<Return>', click)
entry.pack(pady = 20)
entry.focus_set()

label = Label()
label.pack()

```



6.44. ábra. A projekt szövege **Click** eseménykezelővel mező esetében

6.45. ábra. A projekt végrehajtásának ablaka a szöveg beírása után a mezőbe a **Click** esemény bekövetkezése előtt (balra) és utána (jobbra)

### Projektek bemeneti adatokkal és kimeneti eredményekkel

A projekt megvalósítása során adatokat vihetünk be és kimeneti eredményeket kaphatunk.

A bemeneti adatok megadásához a mezőt használjuk, a kimeneti eredmények megjelenítésére pedig a mezőt, címkét és szövegdobozt. Hozzunk létre egy projektet, amelyben két tetszőleges szám összegét számoljuk ki!

Készítsünk matematikai modellt ehhez a projekthez:

**bemeneti adat:** két tetszőleges szám  $x$  és  $y$ ;

**kimeneti eredmény:** a két szám összege  $z$ ;

**képlet:**  $z = x + y$ .

Ebben a projektben két mezőt fogunk használni, amelyekbe a felhasználó a projekt során összeadandókat fog beírni (ezek lehetnek egész számok és törtek is), egy címkét az összeg értékének megjelenítésére és egy gombot, melyet ha kiválasztunk, megtörténik a mezőkben megadott számok összeadása.



#### A Pascalt használóknak

Helyezzünk el a formon:

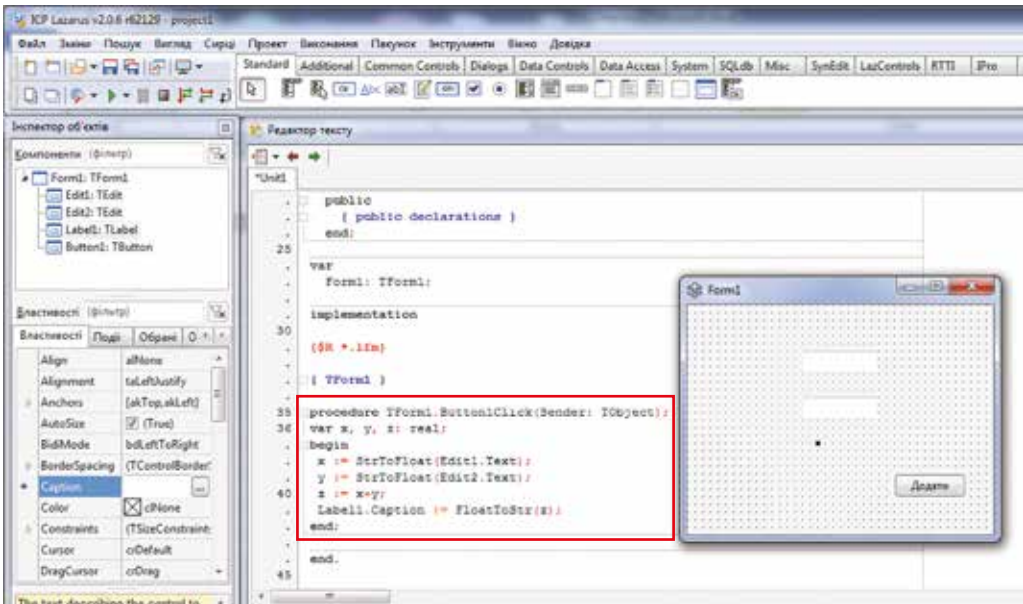
- két mezőt; tegyük üressé az **Objektum felügyelő** ablak **Tulajdonságok** lapjának a **Text** mezőjében található szöveg törlésével;
- egy címkét; tegyük üressé úgy, hogy a **Tulajdonságok** lapon a **Caption** mező szövegét töröljük;
- egy gombot; változtassuk a szövegét **Összeadásra** a **Tulajdonságok** lap **Caption** mezőjét felhasználva!

A 6.46. ábrán látható a gomb **Click** eseménykezelője, melyben a mezőkben megadott két szám összeadása történik. Tekintsük meg az említett eljárást!

Bár a felhasználó a projekt futása során számokat ír a beviteli mezőkbe, a rendszer ezeket szöveggként értelmezi. A szövegek az **Edit1** és **Edit2** mezők **Text** tulajdonságába íródnak be. Ezért az eljárás első két utasítása, az  $x := \text{StrToFloat}(\text{Edit1.Text})$  és az  $y := \text{StrToFloat}(\text{Edit2.Text})$  (ang. *string to float* – szöveg tizedes törtté) átalakítja ezeket a szövegeket tizedes törtekké, és elhelyezi ezeket az  $x$  és  $y$  változóban.

A harmadik utasítás összeadja ezt a két változót, és az eredményt elhelyezi  $z$ -ben.





6.46. ábra. Két mezőt, címkét és két szám összeadásának eljárását tartalmazó form

A negyedik utasítás – **Label1 Caption := FloatToStr (z)** (ang. *float to string* – tizedes tört szöveggé) – a kiszámított összeget számformátumból szöveggé alakítja, és elhelyezi ezt a szöveget a **Label1** címke **Caption** mezőjében.

A **Pascal** programozási nyelv megköveteli, hogy minden változónak ismert **típusa** legyen.

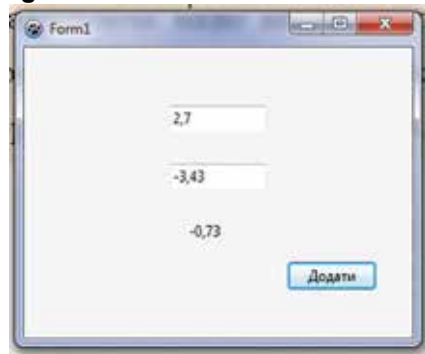
A **változó típusa** meghatározza, hogy:

- milyen értékeket vehet fel a változó;
- milyen műveleteket lehet a változóval végezni;
- mekkora operatív tár szükséges a változó tárolására.

A változók nevét és típusát a **var** (ang. *variable* – változó) kulcsszó után kell megadni. Ezt a sort helyezzük el az eljárás címe és a **begin** szó között!

Az eljárás során használt **x**, **y** és **z** változók a feltétel szerint tetszőleges valós értéket vehetnek fel. Az ilyen változókhoz a **real** (ang. *real* – valós) típus a megfelelő választás. Az ilyen változók egész és tört értéket is felvehetnek, abszolút értékük maximálisan  $1,7 \cdot 10^{38}$  lehet. Minden ilyen szám 8 bájtot foglal el.

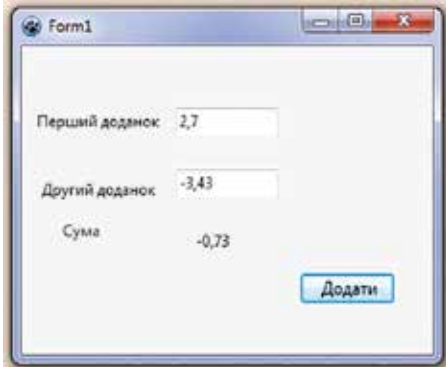
A létrehozott projekt végrehajtása során a felhasználó két számot ír be a mezőkbe (a 6.47. ábrán ezek a 2.7 és -3.43 számok). Kiválasztja az **Összeadás** gombot, és az összeadás végeredménye (-0,73) megjelenik a címkében.



6.47. ábra. A projekt formja két szám összeadására

Annak érdekében, hogy a felhasználó számára világosabbá váljon a projekt során, hogy melyik mezőbe kell megadni az értékeket és hogy melyikben jelenik meg az eredmény, célszerű az egyes formok bal oldalán található mezőkre, címkékre és az eredmény kimenetelére szöveges magyarázattal ellátott feliratokat helyezni, például: *Első összeadandó*, *Második összeadandó*, *Összeg* (6.48. ábra).

Vegyük figyelembe, ha a címke helyett egy harmadik mezőt helyezünk el a formon, akkor a következő parancsot végrehajtva **Edit3.Text := FloatToStr(z)** vagy a jelentésablakba a **ShowMessage (FloatToStr(z))** parancsot használva az eredmény kimenete a mezőben történik.



6.48. ábra. Szöveges magyarázattal ellátott, két szám összeadásának formja.

Ha egy változó csak egész értékeket vehet fel, akkor azt **integer** (egész) típusúnak kell deklarálni. Az ilyen változók megengedett értékei **-32 768** és **32 767** között vannak. A **longint** (ang. *long integer* – hosszú egész) típusú változók 4 bájtot foglalnak el, ezért értékeik **-2 147 483 648** és **2 147 483 647** közöttiek lehetnek.

Ha a projektben bármilyen típusú egész szám szerepel, akkor a mezőről való olvasásukhoz használjuk például az **x := StrToInt (Edit1.Text)** (ang. *string to integer* – szöveg egész számmá) parancsot, az egész számok kimeneteléhez a címkén pedig használjuk a **Label1.Caption := IntToStr(z)** (ang. *integer to string* – egész szám szöveggé) utasítást.

A **var** sorban a változókat az **integer**-be vagy **longint**-be kell deklarálni.

*Itt jegyezzük meg:* ha a **var** kulcsszó után különböző típusú változókat deklarálunk, akkor azt a következő alakban kell felírni: **var x, y: real; t: integer;**. A deklarálás több sorban is történhet, de a **var** szót csak egyszer használhatjuk.

**var x, y: real; t: integer;**



### A Pythont használóknak

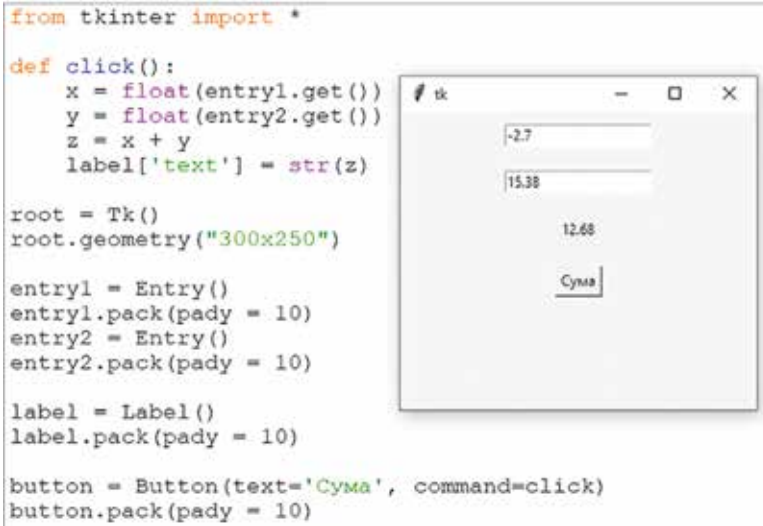
Helyezzünk el a projekt ablakában két mezőt, egy címkét és egy gombot Összeg szöveggel! Az ablakot létrehozó projekt az abban elhelyezett objektumokkal, a két szám hozzáadásának eljárása, valamint a projekt végrehajtási ablaka a 6.49 ábrán látható.

Tekintsük meg az adott eljárást!

Bár a felhasználó a projekt során számokat ír be a mezőkbe, a rendszer mégis szöveggként érzékeli őket. Ezeket a szövegeket az **entry1** és **entry2** nevű mezők tartalmazzák. Ezért az eljárás első két parancsa az **x = float(entry1.get())** és **y = float(entry2.get())** (ang. *float* – lebegőpontos szám) a számok szöveges ábrázolását közvetlenül számokká alakítja és az **x** és **y** változókhoz rendeli.

A harmadik parancs összeadja ezeket a számokat, és az összeget hozzárendeli a **z** változóhoz.

A negyedik parancs **label['text'] = str(z)** (ang. *string* – karakterlánc) a kiszámított összeget számból szöveggé alakítja és ezt a **label** nevű címke **text** tulajdonság értékéhez rendeli.



6.49. ábra. Ablak két mezővel, egy címkével és egy gombbal, valamint a projekt szövege két szám hozzáadásának eljárásával

A Python programozási nyelvben a változók bizonyos típusú értékeknek felelnek meg. A **típus** határozza meg, hogy:

- milyen értékeket vehet fel az adott változó;
- milyen műveletek végezhetők az értékekkel;
- milyen mennyiségű számítógépes RAM kerül lefoglalásra az ilyen típusú adatok tárolására.

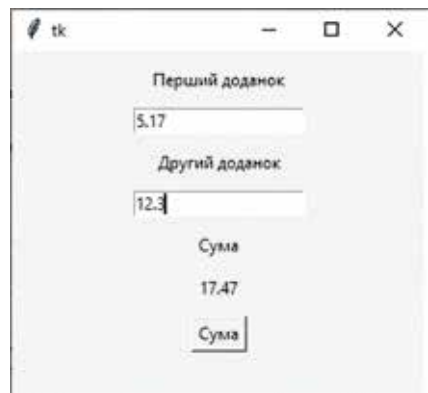
Az eljárásban használt **x**, **y**, **z** változók értékei a feladat feltételei szerint tetszőleges számok lehetnek, azaz egészek és törtek egyaránt. Ilyen értékeknél használhatjuk a **float** típust.

A létrehozott projekt végrehajtása során a felhasználó két számot ír be a mezőkbe (a 6.49. ábrán ezek a -2.7 és a 15.38 számok), kiválasztja az **Összeg** címet tartalmazó gombot, és az eredmény (12.68) megjelenik a címkében. A projekt végrehajtási ablakának bezárása nélkül a felhasználó további számokat írhat be a mezőkbe, ismét kiválaszthatja az **Összeg** címet tartalmazó gombot, és új eredményt kaphat. És így akárhányszor megismételheti ezt a folyamatot.

Annak érdekében, hogy a projekt végrehajtása során a felhasználó számára érthetőbb legyen, hogy melyik mezőbe mit adhatunk meg és hol jelenik meg az eredmény, célszerű a mezők és címkék mellé magyarázó címkéket helyezni, mint például *Első összeadandó*, *Második összeadandó*, *Összeg* (6.50 ábra).

Ha a változók csak egész értékeket vehetnek fel, akkor az **int** adattípus használható (ang. **integer** – egész).

Ha a projektben **int** típusú változókat használunk, akkor az értékek beviteléhez az **x = int(entry1.get())** parancsot alkalmazhatjuk.



6.50. ábra. Projekt végrehajtási ablak két szám összeadásának esetében magyarázattal

A **Python**ban a bemeneti és kimeneti problémák megoldhatók grafikus felhasználói felület létrehozása nélkül is.

A projekt futása során az adatok bevitelére az **input(<szöveg>)** függvényt használják, például az alábbi parancs végrehajtása után:

```
x = float(input('Írja be az első kifejezést: '))
```

- az **IDLE** környezet ablakában ki lesz írva a zárójelben lévő idézőjel közötti szöveg;
- a felhasználónak be kell írnia az első kifejezés értékét a billentyűzet segítségével, és meg kell nyomnia az **Enter** billentyűt;
- az első kifejezés beírt értéke (egész szám vagy tört) szöveggént lesz érzékelve, ezért a **float** függvény segítségével számmá alakítjuk, és ezt a számot az **x** változóhoz rendeljük.

Ha a beírni kívánt szám csak egész szám lehet, akkor **float** helyett az **int** függvényt használjuk.

Az értékek kiírásához a **print(<érték>)** parancsot használjuk. Például:

- **print(z)** a **z** változó értékét írja ki;
- **print('Az összeg egyenlő')** Az *összeg egyenlő* szöveget írja ki;
- **print('Az összeg egyenlő', z)** Az *összeg egyenlő* szöveget írja ki, utána pedig a **z** változó értékét szóközzel elválasztva tőle.

A fenti probléma megoldásának projektjét és az **IDLE** környezetben történő megvalósításának eredményét a 6.51 ábra mutatja.

|  |   |
|--|---|
| <pre>x = float(input('Уведіть перший доданок: ')) y = float(input('Уведіть другий доданок: ')) z = x + y print('Сума дорівнює', z)</pre> | <pre>Python 3.7.0 (v3.7.0:1bf9cc5093 [MSC v.1914 32 bit (Intel)] on Type "copyright", "credits" or rmatation. &gt;&gt;&gt; RESTART: E:/Підручники/8 клас .5 input.py Уведіть перший доданок: 12.3 Уведіть другий доданок: 7.25 Сума дорівнює 19.55 &gt;&gt;&gt;  </pre> |
|--|---|

6.51. ábra. Projekt két szám összegének meghatározására (balra) és végrehajtásának eredménye (jobbra)

## A számokkal végzett műveletek eredményeinek típusai

Már tudjátok, hogy a változó mennyiségekkel matematikai műveleteket lehet végezni.

+ – összeadás;    – – kivonás;    \* – szorzás;    / – osztás.



### A Pascalt használóknak

Ha az  **$x := a + b$** ,  **$x := a - b$** ,  **$x := a * b$**  i  **$x := a / b$**  műveletekben az **a** és a **b** **real** típusú, akkor az **x** **realként** kell legyen deklarálva.

Ha az  **$x := a + b$** ,  **$x := a - b$** ,  **$x := a * b$**  műveletekben az **a** és a **b** **integer (longint)** típusú, akkor az **x** lehet **integer (longint)**, de lehet **real** is.

Ha az  **$x := a / b$**  műveletben az **a** és a **b** **integer (longint)**, az **x** **real** kell legyen.



## A Pythont használóknak

A **Python** számtani kifejezései értékeinek kiszámításakor kapott eredmény típusa attól függ, hogy milyen értékek adóttak ebben a kifejezésben operandusként.

Ha tehát mindkét operandus egész típusú érték, akkor összeadásuk, kivonásuk és szorzásuk eredményeként egész típusú értékeket kapunk.

Ha az operandusok közül legalább egy valós típusú, akkor ezeknek a műveleteknek az eredménye is valós típusú lesz.

Az egész és a valós típusú mennyiségek elosztásának eredménye mindig valós szám lesz.

### Állandók (konstansok) alkalmazása a projektben

A projektekben a változókon kívül állandókat (**konstansok**at) is használhatunk.

Hozzunk létre projektet, amelyben meg kell határozni egy fenyőrönk tömegét a térfogata alapján. Ez fontos lehet annak eldöntésekor, hogy milyen teherautóra van szükség a fenyőrönk fafeldolgozó üzembe szállításához.

Készítsük el a projekt matematikai modelljét:

**bemeneti adat:** a fenyőrönk térfogata  $V$ ;

**kimeneti eredmény:** a fenyőrönk tömege  $m$ ;

**képletek:**  $m = \rho \cdot V$ , ahol  $\rho$  – a fenyő sűrűsége, melynek egy állandó (konstans) és 520 kg/m<sup>3</sup> az értéke!



## A Pascalt használóknak

Feltételezzük, hogy a térfogatot az **Edit1** mezőben köbméterekben adjuk meg, a tömeget pedig a **Label1** címkébe kilogrammokban fogjuk kiírni.

```
const p = 520;
```

```
var v, m: real;
```

```
begin
```

```
  v := StrToFloat(Edit1.Text);
```

```
  m := p*v;
```

```
  Label1.Caption := FloatToStr(m);
```

```
end;
```



## A Pythont használóknak

Tegyük fel, hogy a projektben a térfogatérték köbméterben adódik meg az *entry* nevű mezőbe, és a tömegérték kilogrammban íródik ki egy *label* nevű címkébe.

A **Python**ban nem hozhatók létre olyan állandó értékek (konstansok), mint más programozási nyelvekben, hogy azok számára tilos legyen az érték megváltoztatása. Ehelyett a programozók megegyeznek abban, hogy azoknak az értékeknek a nevét, amelyeket nem ajánlott megváltoztatni, nagy betűvel írják.

Akkor a projekt ennek megfelelő töredéke így fog kinézni:

```
P = 520
```

```
v = float(entry.get())
```

```
m = P * v
```

```
label['text'] = str(m)
```



### Használjuk a számítógépet!

Végezzék el a gyakorlati feladatot a <https://cutt.ly/YhdrsFE> linkről, vagy a QR-kód segítségével!



## Összefoglalás

A **mező** az a komponens, amely adatok bevitelére szolgál. A mezőt eredmények bevitelére is használhatjuk.



## A Pascalt használóknak

A **mezőt Lazarus** környezetben úgy helyezzük el a formra, hogy duplán kattintunk a **TEdit** elemére, ami a **Standart** fülben található.

A **Top, Left, Name, Color, Enabled, Visible, Font** tulajdonságok értelmezése a mező és a címke esetében ugyanaz. A **Width** tulajdonságokat a **Tulajdonság** lapon és a méretezőmarkerek mozgatásával állíthatjuk be, ahogy az a gomb esetében is volt. A **Height** tulajdonságokat nem tudjuk megváltoztatni, ha az **AutoSize** tulajdonság értéke **True**.

Ennek a tulajdonságnak az értéke a projekt végrehajtása során is beállítható egy értékadó utasítással, például az **Edit1.Top := 200, Edit1.Left := Edit1.Left – 50, Edit1.Enabled := False**.

A beviteli mezőben olvasható szöveg megegyezik a **Text** (ang. *text* – szöveg) tulajdonság értékével. Ennek a tulajdonságnak az értékét a **Tulajdonság** lap megfelelő mezőjébe írhatjuk be. A projekt futtatása előtt célszerű ezt a mezőt üresen hagyni. Ennek a tulajdonságnak az értéke a projekt végrehajtása során is beállítható egy értékadó utasítással, például az **Edit1.Text: = 'Szövegmező'**.

A **Pascal** programozási nyelv megköveteli, hogy minden változónak ismert **típusa** legyen. A **változó típusa** meghatározza, hogy:

- milyen értékeket vehet fel a változó;
- milyen műveleteket lehet a változóval végezni;
- mekkora operatív tár szükséges a változó tárolására.

Ha az eljárás során használt változók a feltétel szerint tetszőleges valós értéket vehetnek fel, akkor az ilyen változókhoz a **real** típus a megfelelő választás. Ha egy változó csak egész értékeket vehet fel, akkor azt **integer** vagy **longint** típusúnak kell deklarálni. A változók nevét és típusát a **var** kulcsszó után kell megadni, amit az eljárás címe és a **begin** szó között kell elhelyezni.

A számok mezőről való olvasásához használjuk az **x := StrToInt (Edit1.Text)** vagy az **x := StrToInt (Edit1.Text)** parancsot.

Megjelenítheti a változó értékét a címkén a **Label1.Caption := FloatToStr (z)**, és a mezőn – az **Edit1.Caption := FloatToStr (z)** paranccsal lehet.

A projektekben a változókon kívül állandókat (**konstansokat**) is használhatunk. Az állandók (konstansok) nevét és értékét a **const** sorba írjuk, amelyet az eljárás fejlécének sora után kell elhelyezni.



## A Pythont használóknak

A **Python**ban a beviteli mező létrehozására szolgáló parancs így néz ki:

**<mező neve> = Entry(<tulajdonságok és értékeik halmaza>).**

A gomhoz és a címkéhez hasonlóan a beviteli mező rendelkezik **width, bg, fg, font** tulajdonságokkal, viszont a **height** hiányzik, a mező magasságát a benne lévő karakterek magassága határozza meg. A beviteli mezőnél beállítható a keretvastagság, mint a **bd** tulajdonság értéke.





A beviteli mező ablakban való elhelyezéséhez a **pack()** és **place()** metódusokat használhatjuk.

Ahhoz, hogy a mezőben szöveg íródjon ki, az alábbi parancsot használják:

**<mező neve>.insert(<pozíció>, <szöveg>).**

A mező tartalmát a **get()** metódussal kaphatjuk meg, például:

**label['text'] = entry.get().**

Az **entry** nevű mező tartalmának törlését az alábbi parancssal hajthatjuk végre:

**entry.delete(0, END).**

Mint ahogy más ablakobjektumoknál, a címkéhez eljárás társítható – valamilyen eseménykezelő a **bind()** metódussal.

Beolvashatunk egy mezőből egy számot és hozzárendelhetünk egy változót az **x = float(entry1.get())** vagy **x = int(entry1.get())** parancssal.

A változó értékének a címkén való megjelenítésére a **label['text'] = str(z)**, és a mezőn való megjelenítésére az **entry.insert(0, z)** parancsot használjuk.

A **Python**ban a bemeneti és kimeneti problémák megoldhatók grafikus felhasználói felület létrehozása nélkül is.

A projekt futása során az adatok bevitelére az **input(<szöveg>)** függvényt használják, például **x = float(input('Írd be az első kifejezést: '))** vagy **x = int(input('Írd be az első összeadandót: '))**.

Az értékek kiírásához a **print(<érték>)** parancsot használjuk, például **print('Az összeg egyenlő', z)**.




### Felelj a kérdésekre!

- 1°. Mire használjuk a beviteli mezőt?
- 2°. Hogy helyezzünk el mezőt a formon?
- 3°. Mit jelentenek a mező esetében a **Top**, **Left**, **Name**, **Color**, **Text**, **Enabled** és **Visible** tulajdonságok?
- 4°. Miben különböznek a form és a mező **Top** és **Left** tulajdonságai?
- 5°. Miben különböznek a **Height** és **Width** tulajdonságok a mező és a címke esetében?
- 6°. Hogyan készíthetünk beviteli mezőt **Python**ba?
- 7°. A beviteli mező mely tulajdonságait ismered? Jellemezd ezeket!
- 8°. Hogyan lehet megszerezni a mező értékét; a mezőbe beleírni; törölni a mező adatait?
- 9°. Hogyan lehet megváltoztatni a mező szövegét a projekt előtt és alatt?
- 10°. Milyen utasításokkal alakítjuk át a szöveget számmá és vissza?
- 11°. Mi határozza meg a változó típusát? Milyen típusú változókat ismersz? Miben különböznek egymástól?
- 12°. Hol, és hogy adjuk meg a változó típusát?
- 13°. Miben különbözik az állandó a változótól? Hogyan adható meg egy konstans neve és értéke?
- 14°. Hogyan lehet grafikus felhasználói felület nélkül bevinni és kikapni adatokat a projekt végrehajtása során?





### Végezd el a feladatokat!

- 1°. Hozz létre projektet, amelyben a mező **Click** eseményének hatására 100 pixelrel növeli a mező és a form felső széle közötti távolságot  (a mező széles-


ségét 15 karakterrel), a mezőbe kiírja az *Informatika* szót! Mentsd a projektet a mappádban létrehozott **feladat 6.5.1** mappába!



2°. Hozz létre projektet, amelyben a mező **KeyPress** eseményének hatására 120 pixellel növeli a mező és a form bal széle közötti távolságot ( a mezőkeret vastagságát 3 pixelre), és a mezőbe kiírja a nevedet! Mentsd a projektet a mappádban létrehozott **feladat 6.5.2** mappába!

3°. Hozz létre projektet, amelyben a mező **MouseMove** eseményének hatására megváltoztatja a mező tartalmát, piros betűszínt és szürke hátteret állít be, és a mezőt 100 pixellel jobbra és 80 pixellel lefelé mozdítja ( a mező szélessége 5 karakterrel nő)! Mentsd a projektet a mappádban létrehozott **feladat 6.5.3** mappába!



4°. Hozz létre projektet, amelyben a mező **DbClick** eseményének hatására elmozdítja a mezőt 80 pixellel jobbra és 20 pixellel felfelé ( a mezőkeret vastagsága 5 pixellel megnő), kiírja a nevedet sárga háttérre zöld színnel! Mentsd a projektet a mappádban létrehozott **feladat 6.5.4** mappába!

5\*. Hozz létre egy olyan projektet, amelyben a mező szürkévé válik, 50 pixellel jobbra mozdul, és elérhetetlenné válik, ha a szövegét kijavítjuk a projekt során! Mentsd a projektet a mappádban létrehozott **feladat 6.5.5** mappába!

6°. Készítsd el a  $(2 * a + b) * (a + c)$  kifejezés matematikai modelljét és a kifejezés kiszámításának projektjét! Mentsd a projektet a mappádban létrehozott **feladat 6.5.6** mappába!



7°. Készítsd el a  $(3 * a - b) * (a + 2 * b)$  kifejezés matematikai modelljét és a kifejezés kiszámításának projektjét! Mentsd a projektet a mappádban létrehozott **feladat 6.5.7** mappába!

8°. Készíts matematikai modellt, és hozz létre egy projektet a körvonal hosszának meghatározására! A projektben használj állandót! Mentsd a projektet a mappádban létrehozott **feladat 6.5.8** mappába!

9°. Készíts matematikai modellt, és hozz létre egy projektet a fény útjának kiszámítására egy adott időre! A projektben használj állandót! Mentsd a projektet a mappádban létrehozott **feladat 6.5.9** mappába!

10°. Készíts matematikai modellt, és hozz létre egy projektet, amely meghatározza, hogy milyen összeget kell fizetnie a fogyasztónak az elfogyasztott villamos energiáért! Mentsd a projektet a mappádban létrehozott **feladat 6.5.10** mappába!

11°. A betétes betett a bankba  $x$  hrvnyát évi  $p\%$ -os kamatra. Év végén felveszi a nyereséget. Készíts matematikai modellt és hozz létre egy projektet, amely meghatározza, milyen nyereséget kap a befektető 3 év múlva! Mentsd a projektet a mappádban létrehozott **feladat 6.5.11** mappába!

12°. A betétes betett a bankba  $x$  hrvnyát évi  $p\%$ -os kamatra. A bank feltétele alapján a nyereség hozzáadódik a betéthez. Készíts matematikai modellt, és hozz létre egy projektet, amely meghatározza, milyen nyereséget kap a befektető 2 év múlva! Mentsd a projektet a mappádban létrehozott **feladat 6.5.12** mappába!

13\*. Az autó  $t$  órányit haladt  $v$  km/h-s sebességgel. Minden 100 km-es szakaszon  $k$  liter üzemanyagot fogyasztott. Az üzemagya ára  $p$  hrn literenként. Készíts matematikai modellt, és hozz létre egy projektet, amely meghatározza ennek az útnak az üzemanyag használatát! Mentsd a projektet a mappádban létrehozott **feladat 6.5.13** mappába!



## 10. SZÁMÚ GYAKORLATI MUNKA

### Mezővel rendelkező projektek adatbevitellel és kimeneti eredménnyel

**Figyelem!** A számítógép használata során tartsd be a balesetvédelmi szabályokat és az egészségügyi előírásokat!

**1. feladat.** Hozzatok létre egy projektet a mező és a gomb használatával! A mező tulajdonságainak állítsátok be a következő kezdőértékeit:



#### A Pascalt használóknak

- szélesség – 400 pixel;
- távolság a form bal szélétől – 70 pixel;
- távolság a form felső szélétől – 100 pixel;
- felirat a mezőn – az osztályotok neve.

Hozzatok létre projektet, amelyben a gomb **Click** eseményének hatására 150 pixelrel növeli a mező és a form bal széle közötti távolságot!

Hozzatok létre a gomb **Click** eseménykezelőjét, ami a mezőnek szürke háttérszínt állít be, a mező távolságát a form felső szélétől 150 pixelre állítja, csökkenti 50 pixellel a form bal széle és a mező távolságát, kék betűszínnel kiírja a mezőbe *A mezők tulajdonságait használjuk* szöveget!



#### A Pythont használóknak

- szélesség – 15 karakter;
- keretvastagság – 5 pixel;
- behúzás az ablak felső határától – 30 pixel;
- szöveg a mezőben – az osztály neve.

Hozzatok létre egy **Click** eseménykezelőt a gombhoz, amely beállítja a mező háttérszínét pirosra, a betűtípust **Arialre**, a karakterek méretét 14-re, növeli a mező szélességét 6 karakterrel, fehér színűre állítja a szöveget és kiírja, hogy *Mi a mező tulajdonságait használjuk!*

Mentsétek el a projektet a mappátokban létrehozott **Gyakorlati 10\_1** mappába!

**2. feladat.** Alkossatok egy projektet, amely kiszámítja az  $a^{(4 \cdot b - a)}$  kifejezés értékét!

1. Készítsetek matematikai modellt a feladat megoldásához! Határozzátok meg a bemeneti adatokat, a végeredményeket és a képleteket! Írjátok le a füzetbe!
2. Határozzátok meg, hogy mely elemeket kell elhelyezni a formon a bemeneti adatok megadásához és az eredmények kiírásához! Helyezzétek el őket! Írjátok alá a rendeltetésüket!
3. Hozzatok létre egy **Click** eseménykezelőt a gombhoz a kifejezés értékének kiszámításához és az eredmény kiírásához!




Mentsétek el a projektet a mappátokban létrehozott **Gyakorlati 10\_2** mappába!

**3. feladat.** Marikának az anyukája adott egy bizonyos pénzösszeget és megkérte, hogy vásároljon 400 g süteményt, 2 doboz tejet és kenyeret! Készítsetek egy projektet, amellyel meg tudjátok határozni a Marikánál megmaradó pénzösszeget!


1. Készítsetek matematikai modellt a feladat megoldásához! Határozzátok meg a bemeneti adatokat, a végeredményeket és a képleteket! Írjátok le a füzetbe!
2. Határozzátok meg, hogy mely elemeket kell elhelyezni a formon a bemeneti adatok megadásához és az eredmények kiírásához! Helyezzétek el őket! Írjátok alá a rendeltetésüket!
3. Hozzatok létre egy **Click** eseménykezelőt a gombhoz a kifejezés értékének kiszámításához és az eredmény kiírásához!
4. Mentsétek el a projektet a mappátokban létrehozott **Gyakorlati 10\_3** mappába!

## 6.6. A projekt hibakeresése



1. Mi jelent a projekt *kompilációja* ( *interpretációja*)?
2. Mi a compiler ( interpretator) rendeltetése? Mik a működésének sajátosságai?
3. Mi a szintaktikai hiba? Hogy jelennek meg ezek meg a kompiláció ( interpretáció) során?

### A program tesztelése

Már tudjátok, hogy a projekt létrehozása során szintaktikai hibák keletkezhetnek. A projekt szerzője által nem észlelt hibákat a compiler találja meg a program fordítása során ( interpretátor a parancsértelmezés szakaszában). Ezeknek a hibáknak a javítása után kerülhet sor a projekt újbóli futtatására.



Néha hibák jelentkeznek a projekt végrehajtásának szakaszában vagy annak befejezése után. Például, ha a program futása folyton megszakad, vagy a futtatás nem a helyes eredményre vezet, akkor azt mondják, hogy a program **logikai hibákat** tartalmaz.

A logikai hibák kiszűrésének egyik módja, hogy a projektet **tesztadatokkal** vizsgáljuk. Minden ilyen tesztadatsorhoz előre meghatározzuk az elvárt eredményt. Amennyiben a tényleges eredmény eltér az elvárttól, vagy a projekt megszakítása esetén meg kell határozni, milyen utasítások okozták ezt az eltérést és meg kell találni azokat a parancsokat, melyek okozták a projekt végrehajtásának megszakítását.

A logikai hibák tesztadatokkal történő felderítésének folyamatát nevezzük **tesztelésnek**. A projekt hibái kijavításának a folyamatát **hibakeresésnek** nevezzük.

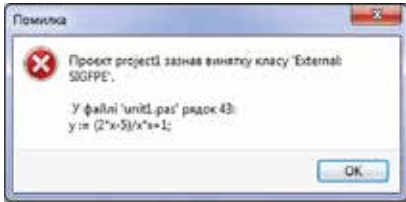
### A projekt lineáris töredékeinek tesztelése

Vegyünk egy példát egy projekt lineáris töredékének tesztelésére. Tegyük fel, hogy az  $y = \frac{2x - 5}{x^2 + 1}$  függvény értékének meghatározására készítünk projektet. A függvény értékének kiszámítására szolgáló eljárás, amelyben az  $x$  értékét egy mező segítségével adjuk meg, a következő lesz:

|  A Pascalt használóknak  |  A Pythont használóknak |
|---|--|
| <pre> procedure TForm1.Button1Click(Sender: TObject); var x, y: real; begin   x := StrToFloat(Edit1.Text);   y := (2*x-5)/(x*x+1);   Label1.Caption := FloatToStr(y); end;           </pre> | <pre> x = float(input('addjuk meg x: ')) y = (2*x-5)/(x*x+1) print(y)           </pre>                     |



Viszonylag gyakori hiba, hogy a tanulók a kifejezés értékének meghatározására szolgáló  $y := (2 \cdot x - 5) / (x \cdot x + 1)$  utasításban elfelejtik zárójelbe venni a nevezőt (időnként a számlálót is), ebből az utasításból tehát  $y := (2 \cdot x - 5) / x \cdot x + 1$  vagy  $y := 2 \cdot x - 5 / x \cdot x + 1$  lesz. Az ilyen hibát nem nehéz kiszűrni a megfelelő teszt segítségével. Ha a tesztérték  $x = 0$ , akkor a projekt az elvárt  $-5$  végeredmény helyett működése során hibajelzéssel leáll (6.52. ábra).



```
Уведіть x: 0
Traceback (most recent call last):
  File "D:/6.6 приклад.py", line 9, in <module>
    y = (2 * x - 5) / x * x + 1
ZeroDivisionError: float division by zero
>>>
```

6.52. ábra. A hibajelzés ablaka



### A Pascalt használóknak

Az **OK** gomb lenyomása után az ablak bezáródik, a program szövegében a hibás sor háttérszínnel lesz kiemelve (6.53. ábra).

```
procedure TForm1.Button1Click(Sender: TObject);
var x, y: real;
begin
  x := StrToFloat(Edit1.Text);
  y := (2*x-5)/x*x+1;
  Label1.Caption := FloatToStr(y);

end;
```

6.53. ábra. Az eljárás szövegében háttérszínnel kiemelt hibás sor

A projektet ilyen esetben a **Ctrl + F2** billentyűkombinációval, vagy a **Leállítás** gombot lenyomva állíthatjuk le.

Ezután meg kell érteni, mi a hiba, majd annak kijavítása után újból futtathatjuk a projektet.

A vizsgált esetben a hiba azért következett be, mert  $x = 0$  értéke mellett nullával való osztás következik be, aminek elvégzése nem lehetséges. Ha másik tesztértéket választunk az  $x$ -nek, például  $-1$ -et, akkor a projekt futása nem áll le, nem jelenik meg hibáüzenet, de az eredmény a helyes  $-3,5$  helyett  $-6$  lesz. Ebben az esetben is át kell vizsgálni az eljárást, meg kell találni a hibát és ki kell javítani azt.

### Kommentek

A hibakeresést jelentősen megkönnyíthetjük, magyarázatokat nyújthatunk a felhasználók számára a program szövegében, ha a program szövegében *kommenteket* helyezünk el. A **komment** olyan szöveg, amelyet a compiler (interpretator) nem hajt végre és logikai hibák keresésére, valamint a program működésének magyarázata céljából helyezük el.



### A Pascalt használóknak

Kommentek megjelöléséhez kapcsos zárójelekbe kell venni azokat, például {**y értékének meghatározása**}, vagy néha a kommentet zárójelek és csillagok közé teszik, például (**\*y értékének kiírása\***) (6.54. ábra).

Ha a komment rövid, vagyis nem foglal el egy sornál többet, akkor a zárójelek helyett két dőlt vonalat tehetünk a komment elé: **//x értékének kiírása** (6.54. ábra).

Amennyiben a komment nem hosszabb egy sornál, akkor az utasítás után is elhelyezhetjük egy új sorban. A komment nem helyezkedhet el az utasításban.

```
procedure TForm1.Button1Click(Sender: TObject);
var x, y: real;
begin
  x := StrToFloat(Edit1.Text); //Команда ввєдєння значєння x
  (Команда обчислення значєння y)
  y := (2*x-5)/x*x+1;
  (*Команда виведєння значєння y*)
  Label1.Caption := FloatToStr(y);

end;
```

6.54. ábra. A projekt szövege kommentárral

A projekt tesztelésekor néha előfordulhat, hogy ideiglenesen nem hajtanak végre bizonyos parancsokat. Ebben az esetben meg kell változtatnia őket kommentekké.

Például kommentté tehetünk egy parancssort  $y := (2 \cdot x - 5) / x \cdot x + 1$ . Ehelyett írjuk be a kimeneti parancsokat a projekt szövegébe, például a tört számlálójának és nevezőjének üzenetmezőjébe.

A logikai hibák kijavítása után vissza kell állítania a projektet az eredeti formájára.



### A Pythont használóknak

A Python projektben a megjegyzés a sor része, amely a # karaktertől a sor végéig terjed. Elhelyezhető egy adott parancs után, vagy akár külön sorba is (6.55. ábra). Parancsba közvetlenül nem lehet megjegyzést elhelyezni.

A Pythonban nem alkalmazandó többsoros megjegyzés. De ha többsoros leírást szeretnénk hozzáadni a projekthez, akkor a projekt elejére helyezve megadhatjuk azt, mint szöveget.

A szöveg elé és után három idézőjelet kell elhelyezni (6.55. ábra).

A projekt tesztelésekor néha előfordulhat, hogy ideiglenesen nem hajódnak végre bizonyos parancsok.

```
''' Проект для обчислення значєння виразу
  2*x - 5
  y = -----
  x*x + 1
  ...
# Команда ввєдєння значєння x
x = float(input('Уведіть x: '))
# Команда обчислення значєння y
y = (2 * x - 5) / x * x + 1
print(y) # Виведєння значєння y
```

6.55. ábra. A projekt szövege megjegyzésekkel

Ebben az esetben megjegyzésre kell őket változtatni. Ehhez kiválasztjuk a sorokat az ilyen parancsokkal, és megnyomjuk az **Alt + 3** billentyű kombinációt. A kijelölt sorok elején megjelenik a # karakter és a sorok megjegyzésekre változnak. Amikor újra futtatni szeretnénk ezeket a parancsokat, kijelöljük őket és megnyomjuk az **Alt + 4** billentyű kombinációt, majd a kijelölt sorok elejéről eltűnnek a # karakterek.





### Használjuk a számítógépet!

Végezzétek el a gyakorlati feladatot a <https://cutt.ly/Ehdrd8p> linkről, vagy a QR-kód segítségével!




### Összefoglalás

**A logikai hiba** egy olyan hiba a projektben, amely a végrehajtás váratlan megszakadásához vagy a végrehajtás helytelen eredményéhez vezet.

**A tesztadatok** a bemeneti adatok olyan értékei, melyekhez ismerjük az elvárt kimenő adatok értékeit, majd össze kell hasonlítani a projekt végrehajtásának bemeneti adatainak valós eredményével.

A **projekt tesztelése** – a logikai hibák keresésének folyamata tesztadatok igénybevételével.


**Hibakeresés** – a projekt hibái kijavításának folyamata.

A **komment** olyan szöveg, amelyet a compiler ( interpretátor) nem hajt végre és logikai hibák keresésére, valamint a program működésének magyarázata céljából helyezük el.



### A Pascalt használóknak

Kommentek megjelöléséhez kapcsos zárójelekbe `{ }` kell venni azokat, vagy néha a kommentet zárójelek és csillagok közé teszik (`* *`), vagy ha a komment rövid, vagyis nem foglal el egy sornál többet, akkor két dőlt vonalat tehetünk a komment elé: `//`.

A **Ctrl + F2** billentyűkombinációval vagy a **Befejezés**  gombra kattintva megszakíthatjuk a projektet.



### A Pythont használóknak

A **Python**ban a megjegyzés a sor része, amely a `#` karaktertől a sor végéig terjed. Elhelyezhető egy adott parancs után, vagy akár külön sorba is. Parancsba közvetlenül nem lehet megjegyzést elhelyezni. Ha többsoros leírást szeretnénk hozzáadni a projekthez, akkor megadhatjuk azt, mint szöveget azáltal, hogy elé és után három idézőjelet helyezünk.

Ahhoz, hogy a parancsokkal rendelkező sorokat megjegyzésekre változtassuk, kijelöljük azokat és az **Alt + 3** billentyű kombinációt alkalmazzuk. Ahhoz, hogy néhány sorban megszüntessük a megjegyzéseket, kiválasztjuk azokat és megnyomjuk az **Alt + 4** billentyű kombinációt.



### Felelj a kérdésekre!

- 1•. Milyenek a logikai hibák a projektben?
- 2◦. Mit értünk tesztelés alatt?
- 3•. Mi célból végezzük a projekt tesztelését?
- 4•. Mit értünk hibakeresésen?
- 5◦. Mi a komment a projekthez? Hogyan lehet a projekt szövegébe kommentet tenni?
- 6•. Mi célból írunk a projektbe kommenteket?

**Végezd el a feladatokat!**

**1°.** Nyisd meg a **6fejezet/feladat 6.6\projekt 6.6.1** projektet, melyet a következő feladat megoldására hoztunk létre: *Két vonat egyidejűleg indult el egymás felé! Határozd meg, mikor találkoznak, ha ismert a kezdeti távolság közöttük és mindegyik vonat sebessége!* A megadott bemeneti adatokhoz számítsd ki és írd be a füzetedbe az eredményeket:

- a) távolság – 1000 km, sebességek – 120 km/h és 130 km/h;
- b) távolság – 600 km, sebességek – 80 km/h és 120 km/h;
- c) távolság – 1200 km, sebességek – 115 km/h és 85 km/h!

Hajtsd végre a projektet az elsőre megadott bemeneti adatokkal! Ellenőrizd, hogy a kapott eredmények megegyeznek-e az elvártakkal! Teszteld a projektet és találd benne logikai hibát! Javítsd ki! Végezd el a projektet a másik két megadott bemeneti adattal! Ellenőrizd, hogy a kapott eredmények megegyeznek-e az elvártakkal! Mentsd a projektet a mappádban létrehozott **feladat 6.6.1** mappába!



**2°.** Hozz létre három tesztadatsort a következő feladat megoldására szolgáló projekt tesztelésére: *Két vonat indult el ugyanarról a helyről ellentétes irányokba. Határozd meg, milyen messze lesznek ezek egymástól, ha ismert az indulás óta eltelt idő és a vonatok sebessége!* Határozd meg a kimenő adatok értékeit a megfelelő bemeneti adatokhoz! Nyisd meg a **6fejezet/feladat 6.6\projekt 6.6.2** projektet! Teszteld a projektet az előkészített adatokkal! Döntsd el, tartalmaz-e a projekt logikai hibákat! Ha igen, javítsd ki azokat! Mentsd a projektet a mappádban létrehozott **feladat 6.6.2** mappába!

**3°.** A gyerekek ünnepre készülve cukorkát és süteményt vettek. Ismert a cukorka és a sütemény ára és a vásárolt mennyiség. Hozz létre projektet a kifizetett összeg meghatározására! Mentsd el a projektet a mappádban létrehozott **feladat 6.6.3** mappába! Írd le a bemeneti adatokat a projekt teszteléséhez!

**4\*.** Nyisd meg a **Téma 6\feladat 6.6\projekt 6.6.4** projektet! Elemezd ezt a projektet a programban elhelyezett kommentek alapján, és határozd meg, milyen feladat megoldására szolgál! A következtetésed alapján készíts három teszt-sort a projekt teszteléséhez! Határozd meg az elvart eredményeket! Teszteld a projektet a kidolgozott teszt-sorokkal! Döntsd el, tartalmaz-e a projekt logikai hibákat! Ha igen, javítsd ki azokat! Mentsd a projektet a mappádban létrehozott **feladat 6.6.4** mappába!

**6.7. Logikai kifejezések. Logikai változók. Logikai műveletek**

1. Mi a kijelentés? Mely kijelentéseket tekintünk igaznak, és melyeket hamisnak?
2. Milyen tulajdonságai vannak a változóknak?
3. Mit határoz meg az adattípus? Milyen típusai lehetnek a számot tartalmazó változóknak?

**Logikai kifejezések**

Már tudjátok, hogy a **kijelentés** egy olyan mondat, ami valamely objektumra, objektumok kapcsolatára nézve olyan állítást fogalmaz meg, amelyről egyértelműen eldönthető, hogy **igaz** vagy **hamis**. **Igaz kijelentés** például a *Kijev Ukrajna fővárosa* mondat, a  $2 * 5 - 4 = 6$ ,  $4 < 12$ . **Hamis kijelentés** például: *Január 30 napos*,  $2 * 7 + 3 = 12$ ,



3 > 5. Példák olyan mondatokra, amelyek nem állítások: *Kérlek, csukd be az ablakot, Neked tetszett ez a film?*

A kijelentést **logikai kifejezésnek** tekinthetjük.

A **logikai kifejezés** olyan kifejezés, amely **true** (ang. *true* – igaz, valós) vagy **false** (ang. *false* – hamis) értéket vehet fel.



Ha a kijelentés igaz, a megfelelő logikai kifejezés értéke is igaz (**true**). Vagyis *Kijev Ukrajna fővárosa* = **true**,  $(2 * 5 - 4 = 6) = \mathbf{true}$ ,  $(4 < 12) = \mathbf{true}$ .

Ha a kijelentés hamis, a megfelelő logikai kifejezés értéke is hamis (**false**). Tehát *Ez a tankönyv hetedikeseeknek szól* = **false**,  $(2 * 7 + 3 = 12) = \mathbf{false}$ ,  $(3 > 5) = \mathbf{false}$ .

*Itt jegyezzük meg, hogy* a logikai kifejezéseket elágazásokban és ciklusokban használják. A következő részekben ezekkel foglalkozunk.

**Logika** (gör. λόγος – szó, gondolat, értelem, nyelv szóból) – a helyes gondolkodás törvényeit, módszereit és formáit, a következtetés módozatait vizsgáló tudomány.

### Logikai változók



Az előző részekben szám típusú változókat használtunk. Emlékezzünk vissza, hogy **real** ( **float**) típusú változókat használtunk amelyek értékei tetszőleges számok lehetnek (egész és tört), valamint **integer** ( **int**) típust, melynek értékei csak egész számok lehetnek.


Az informatikában olyan változókat is használnak, amelyek értéke **true** vagy **false**.

Azokat a változókat, amelyek a **true** vagy a **false** érték egyikét vehetik fel, **logikai típusú változóknak** nevezzük.

A változók logikai típusát **booleannal** ( **boollal**) jelöljük.



*Itt jegyezzük meg, hogy* a logikai típusú változó egyfajta logikai kifejezés. A változóhoz hozzárendelési parancsokkal rendelhetünk logikai változókat. Például:

|  <b>A Pascalt használóknak</b> |  <b>A Pythont használóknak</b> |
|---|---|
| <b>x := true vagy x := false</b>  | <b>x = True vagy x = False</b>  |

Logikai típusú változóhoz hozzárendelhető egy másik logikai változó értéke, például **x := y** ( **x = y**).

Egy logikai típusú változóhoz hozzárendelhető egy logikai kifejezés értéke is.

Vagyis a következő hozzárendelési parancsokat használhatjuk:

|  <b>A Pascalt használóknak</b> |  <b>A Pythont használóknak</b> |
|---|---|
| <b>x := 2 + 12 / 3 = 6</b><br><b>y := 32 &lt; 13</b>  | <b>x = 2 + 12 / 3 == 6</b><br><b>y = 32 &lt; 13</b>   |

Ekkor az **x** változó értéke **true**, az **y**-é pedig **false**.



## A Pascalt használóknak

A logikai típusú változókat szintén a **var**ba kell deklarálni. Például **var x, y: boolean**.



## A Pythont használóknak

Annak ellenőrzésére, hogy két kifejezés értéke megegyezik-e, a **Python** nyelv az **==** jelet használja. A **True** és **False** értékeket nagybetűvel kell írni.

### Logikai műveletek

Tekintsünk meg három **logikai műveletet**: **tagadás**, **konjunkció** (a latin *conjunctio* – egyesítés szóból) és a **diszjunkció** (latin *disjunctio* – szétválasztás).

Az **x tagadása** olyan logikai művelet, amelynek eredménye **true**, ha az **x** logikai kifejezés értéke **false**, illetve **false**, ha az **x** logikai kifejezés értéke **true**.

Az **x** tagadását **not x**-szel jelöljük. Néha az **x** tagadását  $\bar{x}$  vagy  $\neg x$ -el jelöljük.

Vagyis ha **x = true**, akkor **not x = false**, és ha **x = false**, akkor **not x = true**. Ezek alapján össze tudjuk állítani a tagadás műveletének **igazságtáblázatát**.

| <b>x</b> | <b>not x</b> |
|----------|--------------|
| true     | false        |
| false    | true         |

A tagadás képzése a **nem** tagadószó használatával történik. Az **A = Létezik a legnagyobb szám** tagadása a **not A = Nem létezik a legnagyobb szám**.

Az **x** és **y** logikai kifejezések **konjunkciója** az a logikai kifejezés, amelynek értéke akkor **true**, ha az **x** és az **y** értéke is **true**, minden más esetben pedig **false**.

Az **x** és **y** kifejezések konjunkciójának szokásos jelölése **x and y** (az angol *and* – és szóból). Néha **x** és **y** logikai kifejezés konjunkciójának jelölése **x  $\wedge$  y**.

A konjunkció igazságtáblája a következőképpen néz ki:

| <b>x</b> | <b>y</b> | <b>x and y</b> |
|----------|----------|----------------|
| true     | true     | true           |
| true     | false    | false          |
| false    | true     | false          |
| false    | false    | false          |

Két logikai kifejezés konjunkcióját az **és** kötőszóval fogalmazzuk meg. A **36 többszöröse 3-nak és 36 többszöröse 2-nek** kijelentés a **36 többszöröse 2-nek**, illetve a **36 többszöröse 3-nak** kifejezések konjunkciója.

A **diszjunkció** az **x** és **y** logikai kifejezéseken végzett olyan művelet, amelynek eredménye akkor **true**, ha az **x** és **y** logikai kifejezés közül legalább az egyik értéke **true** és akkor **false**, ha **x** és **y** mindegyike **false**.



Az  $x$  és  $y$  logikai kifejezések diszjunkciójának jelölése  $x$  or  $y$  (ang. or – vagy).  
 Néha  $x$  és  $y$  logikai kifejezés diszjunkciójának jelölése  $x \vee y$ .  
 A diszjunkció igazságtáblája a következőképpen néz ki:

| $x$   | $y$   | $x$ or $y$ |
|-------|-------|------------|
| true  | true  | true       |
| true  | false | true       |
| false | true  | true       |
| false | false | false      |

Két kijelentés diszjunkciójának megfogalmazásakor a **vagy** kötőszót használjuk. Például a  $11 \leq 21$  (11 *kisebb vagy egyenlő* 21-gyel) kifejezés  $11 < 21$  és a  $11 = 21$  kijelentések diszjunkciója.

Gyakran használunk több **logikai** műveletet tartalmazó **kifejezéseket**. Például  **$x$  or  $y$  and not  $z$** .

A fentebb említett logikai műveleteknek is van **prioritási sorrendje**: először a tagadást hajtjuk végre, majd konjunkciót, végül a diszjunkciót.

A logikai kifejezések értékének kiszámításakor igazságtáblázatokat használunk. E célból a kifejezést alkotó logikai változók értékeinek valamennyi lehetséges kombinációját át kell tekinteni.

| $x$   | $y$   | $z$   | 1       | 2               | 3                      |
|-------|-------|-------|---------|-----------------|------------------------|
|       |       |       | not $z$ | $y$ and not $z$ | $x$ or $y$ and not $z$ |
| true  | true  | true  | false   | false           | true                   |
| true  | true  | false | true    | true            | true                   |
| true  | false | true  | false   | false           | true                   |
| false | true  | true  | false   | false           | false                  |
| true  | false | false | true    | false           | true                   |
| false | true  | false | true    | true            | true                   |
| false | false | true  | false   | false           | false                  |
| false | false | false | true    | false           | false                  |

A zárójelek a logikai műveletek sorrendjének módosítására szolgálnak. Például az  **$(x$  or  $y)$  and not  $z$**  logikai kifejezésben először a zárójelben lévő diszjunkciót hajtjuk végre, másodsor a tagadást és végül a konjunkciót.



### Érdekes történelmi tények

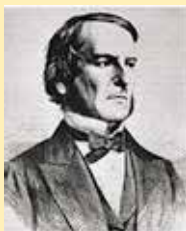
A logika mint tudomány az ókori Görögországban született. Eme tudomány megalkotójának az ógörög filozófust, **Arisztotelészt** tartják (i.e. 384-322) (6.56. ábra).

A logikai típusú változókkal végzett műveleteket először **George Boole** (1815–1864) angol matematikus és filozófus (6.57. ábra) definiálta. Ő kutatta először ezek tulajdonságait. 1854-ben jelent meg *A gondolkodás azon törvényeinek vizsgálata, amelyeken a matematikai logika és valószínűség elmélete nyugszik* című munkája, amiben szisztematikusan megfogalmazza azt, amit ma a *kijelentések algebrájának* vagy *Boole-logikának* neveznek.

A Boole-logika a számítógépek létrehozásának matematikai eszközévé vált, és az összes programozási nyelvhez használják. A logikai változók típusának nevét a tudósról kapta.



6.56. ábra.  
Arisztotelész



6.57. ábra.  
George Boole



6.58. ábra.  
Irina Homenko

Érdeemes megjegyezni, hogy jelentősen hozzájárultak a logika mint tudomány fejlődéséhez Jana Lukaszevics (1878–1956), Pavlo Kopnyin (1922–1971), Klaudiusz Rudenko (1919–1974), Irina Homenko (szül. 1961) ukrán tudósok.




### Összefoglalás

A **kijelentés** egy olyan mondat, ami valamely objektumra, objektumok kapcsolatára nézve olyan állítást fogalmaz meg, amelyről egyértelműen eldönthető, hogy **igaz** vagy **hamis**.

A **logikai kifejezés** olyan kifejezés, amely **true** vagy **false** értéket vehet fel.

A kijelentést **logikai kifejezésnek** tekinthetjük. Ha a kijelentés igaz, a megfelelő logikai kifejezés értéke is igaz (**true**). Ha a kijelentés hamis, a megfelelő logikai kifejezés értéke is hamis (**false**).

Azokat a változókat, amelyek a **true** vagy a **false** érték egyikét vehetik fel, **logikai típusú változóknak** nevezzük. A változók logikai típusát **booleannal** ( **boollal**) jelöljük.

A logikai típusú változókat logikai kifejezéseknek is tekintjük.

Logikai kifejezéseken **logikai műveletek** végezhetők: **tagadás**, **konjunkció**, **diszjunkció**.

Az **x tagadása** olyan logikai művelet, amelynek eredménye **true**, ha az **x** logikai kifejezés értéke **false**, illetve **false**, ha az **x** logikai kifejezés értéke **true**.

Az **x** és **y** logikai kifejezések **konjunkciója** az a logikai kifejezés, amelynek értéke akkor **true**, ha az **x** és az **y** értéke is **true**, minden más esetben pedig **false**.

A **diszjunkció** az **x** és **y** logikai kifejezéseken végzett olyan művelet, amelynek eredménye akkor **true**, ha **x** és **y** logikai kifejezés közül legalább az egyik értéke **true** és akkor **false**, ha **x** és **y** mindegyike **false**.

A fentebb említett logikai műveleteknek is van **prioritási sorrendje**: először a tagadást hajtjuk végre, majd konjunkciót, végül a diszjunkciót. A zárójelek a logikai műveletek sorrendjének módosítására szolgálnak.

Az igazságtáblázat segítségével kiszámíthatjuk egy logikai kifejezés értékeit. Ebben figyelembe kell venni a logikai típusú változók összes lehetséges értékét, amelyek szerepelnek ebben a kifejezésben.






**Felelj a kérdésekre!**

- 1<sup>o</sup>. Mi a kijelentés? Mondj példákat igaz kijelentésre, hamis kijelentésre és olyanra, amit nem tartunk kijelentésnek!
- 2<sup>o</sup>. Milyen kifejezést tekintünk logikainak? Mondj példákat!
- 3<sup>o</sup>. Milyen változókat nevezünk logikai változóknak?
- 4<sup>o</sup>. Hogy deklaráljuk a logikai változókat?
- 5<sup>o</sup>. Mi a logikai kifejezés tagadása?
- 6<sup>o</sup>. Mit értünk két logikai kifejezés konjunkcióján?
- 7<sup>o</sup>. Mit értünk két logikai kifejezés diszjunkcióján?
- 8<sup>o</sup>. Mi a logikai műveletek prioritásának sorrendje?
- 9\*. Mit tanulmányoz a logika?



**Végezd el a feladatokat!**

- 1<sup>o</sup>. Mondj példákat a logikai kifejezések és logikai műveletek matematikában, más tantárgyakban, élethelyzetekben történő használatára!
- 2<sup>o</sup>. Fogalmazd meg az alábbi állítások tagadását, valamint derítsd ki, hogy igazak-e vagy hamisak:
  - a) A 3 osztója 545-nek;
  - b) A gépkocsi nem haladhat át a piros jelzésen;
  - c) Léteznek derékszögű paralelogrammák;
  - d) A  $2x - 3 = 0$  egyenletnek van egész gyöke;
  - e) Nem létezik 2-vel osztható természetes szám;
  - f) Létezik olyan egész szám, ami minden más egész számmal osztható!
- 3<sup>o</sup>. Az alábbi kijelentések közül válaszd ki a konjunkciókat és diszjunkciókat! Határozd meg ezek logikai értékét!
  - a) A 27 többszöröse 3-nak és 9-nek.
  - b)  $17 < 42 < 18$ .
  - c) A 2 prímszám vagy páros.
  - d) Az  $ABC$  háromszög hegyesszögű, derékszögű vagy tompaszögű.
  - e) Minden paralelogramma átlói derékszögben metszik és felezik egymást.
  - f)  $7^2 = 49$  és  $(-7)^2 = 49$ .
  - g)  $21 \leq 21$ .
  - h)  $21 \leq 18$ .
- 
 4<sup>o</sup>. Legyen  $A = \text{Szerhijenko angol nyelvet tanul}$  és  $B = \text{Szerhijenko 8-as osztályzatot kapott informatikából}$ . Fogalmazd meg az alábbi kijelentéseket!
  - a)  $A \text{ and } \text{not } B$ ;
  - b)  $\text{not } A \text{ or } B$ ;
  - c)  $A \text{ or } \text{not } A \text{ and } B$ ;
  - d)  $\text{not } A \text{ and } \text{not } B \text{ and } A$ .
- 5<sup>o</sup>. Add meg a logikai műveletek sorrendjét az alábbi logikai kifejezésekben:
  - a)  $a \text{ or } \text{not } a \text{ and } b$ ;
  - b)  $\text{not } a \text{ or } \text{not } b \text{ and } a$ ;
  - c)  $(a \text{ or } \text{not } b) \text{ and } (\text{not } a \text{ or } \text{not } b)$ ;
  - d)  $\text{not } a \text{ and } \text{not } b \text{ or } c \text{ and } d$ .
- 6<sup>o</sup>. Készítsd el az  $a$  és  $b$  logikai változókat tartalmazó kifejezések igazságtáblázatait:
  - a)  $a \text{ or } a \text{ and } b$ ;
  - b)  $a \text{ or } \text{not } a \text{ and } b$ ;
  - c)  $a \text{ or } b \text{ or } \text{not } b$ ;



- d) **(a or not b) and (a or not b)**;  
 e) **not a or not b**;  
 f) **a and (not b or a) and b**;  
 g) **a and (b or c) not a**;  
 h) **a or b and c or not b**.

7\*. Készítsd el az alábbi kijelentések igazságtáblázatait:

- a) Darina szeret korcsolyázni, vagy Petrik atlétikával foglalkozik és Marika pedig egy zeneiskolában tanul.  
 b) Ma eső vagy szeles idő van és esőkabátot kell felvenni.  
 c) Katicának 8-asa van informatikából és ukrán nyelvből, vagy 9-ese Ukrajna történelméből.

8\*. Határozd meg a kijelentések igazságát vagy hamisságát a változók megadott értékeire:

- a)  $x > 0$  és  $y > 0$  vagy  $x < 0$  és  $y < 0$ ;  
 b)  $x > 0$  és nem  $y < 0$  vagy  $x < 0$  és  $y > 0$ !

Változók értékei:

- 1)  $x = 5, y = 8$ ;                      2)  $x = 5, y = -8$ ;  
 3)  $x = -5, y = 8$ ;                      4)  $x = -5, y = -8$ !

9\*. Az üzemben három szerelvény található. Érzékelők vannak felszerelve rájuk. Ha az érzékelők határértékei az első szerelvénynél és bármely más vagy mindhárom szerelvénynél egyszerre túllépnek, vészhelyzet áll elő. Ebben az esetben figyelmeztető jelzésnek kell megszólalnia. A rendszer automatikus működéséhez készíts hozzá egy logikai kifejezést és egy igazságtáblát, amely alapján ez a rendszer működni fog!



10\*. Misike akkor játszhat a számítógépen, ha elvégezte a házi feladatát vagy rendet tett a szobájában, és nővére befejezte a munkát a számítógépen. Az *Okos ház* rendszerben rögzíthetjük az egyes munkák teljesítményét. Ez lehetővé teszi, hogy automatikusan kapjunk üzenetet a számítógép használatáról. Készíts hozzá egy logikai kifejezést és egy igazságtáblát, amely alapján az üzenetkezelő rendszer működik!

## 6.8. Elágazások



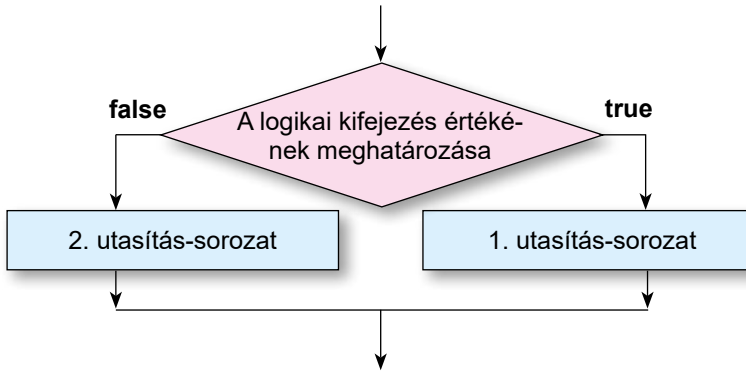
1. Milyen kifejezéseket nevezünk *logikainak*?
2. Milyen logikai műveletek ismertek? Mik lesznek az eredményeik?
3. Mi az elágazás? Milyen típusai vannak? Miben különböznek ezek?

### Elágazó algoritmusok

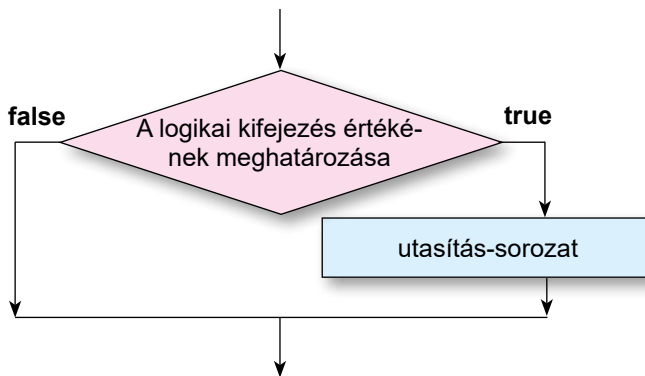
Már tudjátok, hogy az algoritmusok elágazásokat tartalmazhatnak. Azt is tudjátok, hogy az elágazás, mint az algoritmus része a feltétel ellenőrzését végző utasításból áll, aminek eredménye **igaz (true)** vagy **hamis (false)** lehet. Az ellenőrzés eredményétől függ, hogy ezt követően milyen utasítások kerülnek végrehajtásra.

Azt is tudjátok már, hogy a feltétel a logikai kifejezések egy példája.

Általános esetben ebben az utasításban egy logikai kifejezés értékét kell meghatározni, ami **true** vagy **false** lehet. Az elágazás folyamatábrája általános esetben a következő (6.59. és 6. 60. ábrák):





6.59. ábra. A teljes elágazás folyamatábrája



6.60. ábra. A nem teljes elágazás folyamatábrája

### Elágazások a programozásban

A teljes elágazás általános alakja a különböző programozási nyelvekben a következő:

|  <b>A Pascalt használóknak</b>                  |  <b>A Pythont használóknak</b> |
|--|---|
| <pre>if &lt;logikai kifejezés&gt; then begin     &lt;utasítás-sorozat1&gt; end else begin     &lt;utasítás-sorozat2&gt; end;</pre> | <pre>if &lt;logikai kifejezés&gt;:     &lt;utasítás-sorozat1&gt; else:     &lt;utasítás-sorozat2&gt;</pre>        |

(ang. *if* – ha, *then* – akkor, *else* – másként).

Az utasítás végrehajtása a következőképpen történik: kiértékelődik a logikai kifejezés; amennyiben az eredmény **true**, akkor az **1. utasítás-sorozat** kerül végrehajtásra, ami után az elágazás utáni első utasításra kerül a vezérlés; amennyiben az eredmény **false**, a **2. utasítás-sorozat** kerül végrehajtásra, ami után az elágazás utáni első utasításra kerül a vezérlés.

A nem teljes elágazás általános alakja a különböző programozási nyelvekben a következő:

|  <b>A Pascalt használóknak</b> |  <b>A Pythont használóknak</b> |
|---|---|
| <pre>if &lt;logikai kifejezés&gt; then begin     &lt;utasítás-sorozat&gt; end;</pre>                            | <pre>if &lt;logikai kifejezés&gt;:     &lt;utasítás-sorozat&gt;</pre>   |

Az utasítás végrehajtása a következőképpen történik: kiértékelődik a logikai kifejezés; amennyiben az eredmény **true**, akkor az **utasítás-sorozat** végrehajtásra kerül, ami után az elágazás utáni első utasításra kerül a vezérlés; amennyiben az eredmény **false**, az utasítás-sorozat nem kerül végrehajtásra, a vezérlés azonnal az elágazás utáni első utasításra kerül.



### A Pascalt használóknak

*Itt hívjuk fel a figyelmeteket a következőkre:*

- ha a **Then** és **Else** után csak egy utasítás következik, akkor a **begin** és **end** utasításmárkójeleket nem kötelező alkalmazni;
- az **Else** előtt pontosvesszőt tilos tenni.



### A Pythont használóknak

Az elágazási parancs másik formája a **Pythonban** a többszörös elágazási parancs:

```
if <logikai kifejezés 1>:
    <parancssor 1>
elif <logikai kifejezés 2>:
    <parancssor 2>
elif <logikai kifejezés 3>:
    <parancssor 3>
```

...

**else:**

```
<parancssor N>
```

Ez a parancs az alábbi módon megy végre:

- ✓ meghatározódik az 1. logikai kifejezés értéke;
- ✓ ha ez az érték **true**, akkor az 1. parancssor végrehajtásra kerül, utána pedig az ezt követő parancs az elágazásban;
- ✓ ha ez az érték **false**, akkor a 2. logikai kifejezés értékét kell kiszámítani;
- ✓ ha ez az érték **true**, akkor az 2. parancssor végrehajtásra kerül, utána pedig az ezt követő parancs az elágazásban;
- ✓ ha ez az érték **false**, akkor a következő logikai kifejezés értéke kerül kiszámításra, és így tovább;
- ✓ ha a parancsban szereplő összes logikai kifejezés értéke **false**, akkor az **else** után elhelyezkedő **N parancsok sorozata** kerül végrehajtásra, ezután pedig az elágazást követő parancs kerül végrehajtásra.

*Felhívjuk a figyelmeteket:*

- Azok a parancssorok, amelyek végrehajtása a logikai kifejezések értékétől függ, jobbra behúzással íródnak az **if**, **elif** és **else** szavak első betűjéhez képest. Az összes parancsnak azonos behúzással kell rendelkeznie.



- Ha a parancssorozatok csak egy parancsot tartalmaznak, akkor ez a parancs úgy írható, hogy a kettőspont után nem szükséges új sorra lépni.

**Feladatok megoldása elágazásokkal**

Vizsgáljunk most meg egy elágazást tartalmazó algoritmust!

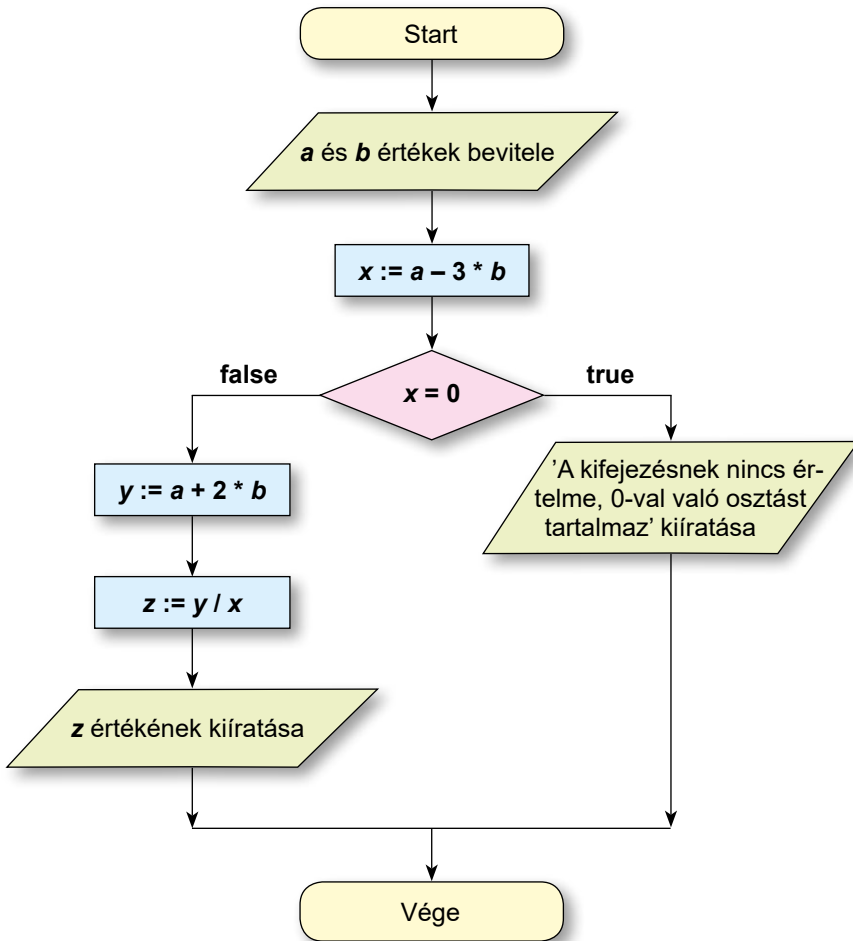
- 1. feladat.** Számítsuk ki a  $z = \frac{a+2b}{a-3b}$  kifejezés értékét, ahol  $a, b$  – valós számok.

Megalkotjuk a feladat matematikai modelljét:

- **bemeneti adat:**  $a$  és  $b$  valós szám;
- **kimeneti eredmény:**  $z$  értéke;
- **képletek:**  $z = (a + 2 * b) / (a - 3 * b)$ !



A kifejezés osztást tartalmaz az  $a - 3 * b$ -re. Az osztó  $a - 3 * b$  nullává válik (például  $a = 3$  és  $b = 1$  esetében), valamint nem válik nullává (például az  $a = 5$  és  $b = 2$  esetében). Ezért a feladat megoldásának algoritmusában elágazást kell tartalmazzon, amelyben két lehetséges eset lesz figyelembe véve: az osztó  $a - 3 * b$  0-val egyenlő, és az osztó  $a - 3 * b$  nem egyenlő 0-val.

A feladat megoldásának algoritmusát a 6.61. ábra folyamatábrája mutatja.





6.61. ábra. Az 1. feladat megoldásának algoritmusában

A 6.61. ábrán látható algoritmus a különböző programozási nyelveken a következő alakban írható fel:

|  <b>A Pascalt használóknak</b>  |  <b>A Pythont használóknak</b>  |
|--|--|
| <pre> a := StrToFloat(Edit1.Text); b := StrToFloat(Edit2.Text); x := a - 3 * b; if x = 0 then Label1.Caption := 'A kifejezésnek nincs értelme, 0-val való osztást tartal- maz' else begin     y := a + 2 * b;     z := y / x;     Label1.Caption := FloatToStr(z); end; </pre> | <pre> a = float(input('Add meg a: ')) b = float(input('Add meg b: ')) x = a - 3 * b if x == 0:     print('A kifejezésnek nincs értelme, 0-val való osztást tartalmaz') else:     y = a + 2 * b     z = y / x     print(z) </pre> |

Felhívjuk a figyelmeteket, hogy az  $a - 3 * b$  értékének kiszámítására nem kötelező külön utasítást alkalmazni és nem szükséges ezt az értéket külön az  $x$  változóban tárolni. Az  $(a + 2 * b) / (a - 3 * b)$  értékét sem szükséges két lépésben kiszámítani. A vizsgált algoritmust a következőképpen is leírhatjuk:

|  <b>A Pascalt használóknak</b>  |  <b>A Pythont használóknak</b>  |
|--|--|
| <pre> a := StrToFloat(Edit1.Text); b := StrToFloat(Edit2.Text); if a - 3 * b = 0 then Label1.Caption := 'A kifejezésnek nincs értelme, 0-val való osztást tartal- maz' else begin     z := (a + 2 * b) / (a - 3 * b);     Label1.Caption := FloatToStr(z); end; </pre> | <pre> a = float(input('Add meg a: ')) b = float(input('Add meg b: ')) if a - 3 * b == 0:     print('A kifejezésnek nincs értelme, 0-val való osztást tartalmaz') else:     z = (a + 2 * b) / (a - 3 * b)     print(z) </pre> |

A projekt helyességének ellenőrzéséhez futtathatjuk például a következő bemeneti adatokkal:

- 1)  $a = 6, b = 2;$
- 2)  $a = 2, b = 3.$

Tekintsünk át most egy olyan feladatot, amelynek elágazása konjunkciót alkalmaz!

**2. feladat.** Számítsd ki a következő függvény értékét

$$y = \begin{cases} 2x - 12, & \text{ha } -3 < x < 5; \\ 7 - 8x & \text{minden egyéb értéke mellett!} \end{cases}$$



Megalkotjuk a feladat matematikai modelljét:

- **bemeneti adat:**  $x$  tetszőleges szám;
- **kimeneti eredmény:**  $y$  függvény értéke;
- **képletek:**  $y = \begin{cases} 2x - 12, & \text{ha } -3 < x < 5; \\ 7 - 8x & \text{minden egyéb értéke mellett!} \end{cases}$






A feladat megoldására szolgáló projektrészlet a következőképpen néz ki:

|  <b>A Pascalt használóknak</b>                                  |  <b>A Pythont használóknak</b>           |
|--|---|
| <pre>x := StrToFloat(Edit1.Text); if (x &gt; -3) and (x &lt; 5) then y := 2 * x - 12 else y := 7 - 8 * x; Label1.Caption := FloatToStr(y);</pre> | <pre>x = float(input('Add meg x: ')) if x &gt; -3 and x &lt; 5:     y = 2 * x - 12 else:     y = 7 - 8 * x print(y)</pre> |

A projekt helyességének ellenőrzéséhez futtathatjuk például a következő bemeneti adatokkal: 1)  $x = -4$ ; 2)  $x = 0$ ; 3)  $x = 7$ .

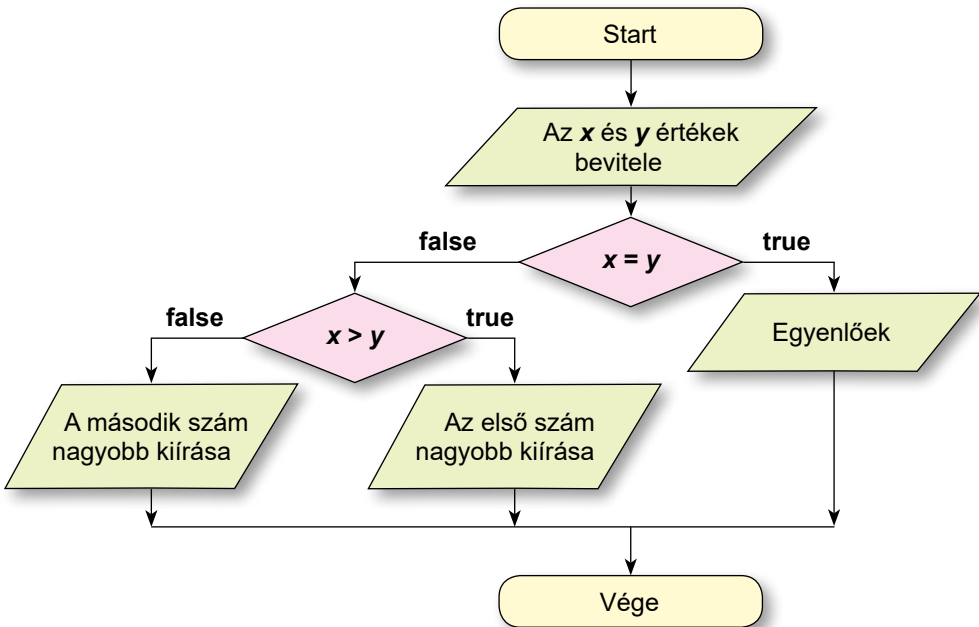
Lássunk egy olyan feladatot, amelynek megoldása során valamivel összetettebb elágazás alkalmazására van szükség ( több alternatív elágazás)!

**3. feladat.** Adva van két szám. Határozd meg, egyenlők-e ezek! Ha nem, akkor határozd meg, melyik a nagyobb! Megalkotjuk a feladat matematikai modelljét:

- **bemeneti adat:**  $x$  és  $y$  tetszőleges szám;
- **kimeneti eredmény:** felelet: a számok egyenlők, vagy az első szám nagyobb, vagy a második szám nagyobb;
- **képletek:**



$$\begin{cases} \text{a számok egyenlők, ha } x = y; \\ \text{az első szám nagyobb, ha } x > y; \\ \text{a második szám nagyobb, ha } x < y. \end{cases}$$

A feladat megoldásának folyamatábrás algoritmusát a 6.62. ábra mutatja.



6.62. ábra. A 3. feladat megoldásának folyamatábrás algoritmus

Az aktuális feladat projektrészlete:

|  <b>A Pascalt használóknak</b>  |  <b>A Pythont használóknak</b>   |
|--|---|
| <pre> x := StrToFloat(Edit1.Text); y := StrToFloat(Edit2.Text); if x = y then Label1.Caption := 'a számok egyenlők' else if x &gt; y then Label1.Caption := 'az első szám nagyobb' else Label1.Caption := 'a második szám nagyobb'; </pre> | <pre> x = float(input('Add meg x: ')) y = float(input('Add meg y: ')) if x == y:     print('a számok egyenlők') elif x &gt; y:     print('az első szám nagyobb') else:     print('a második szám nagyobb') </pre> |

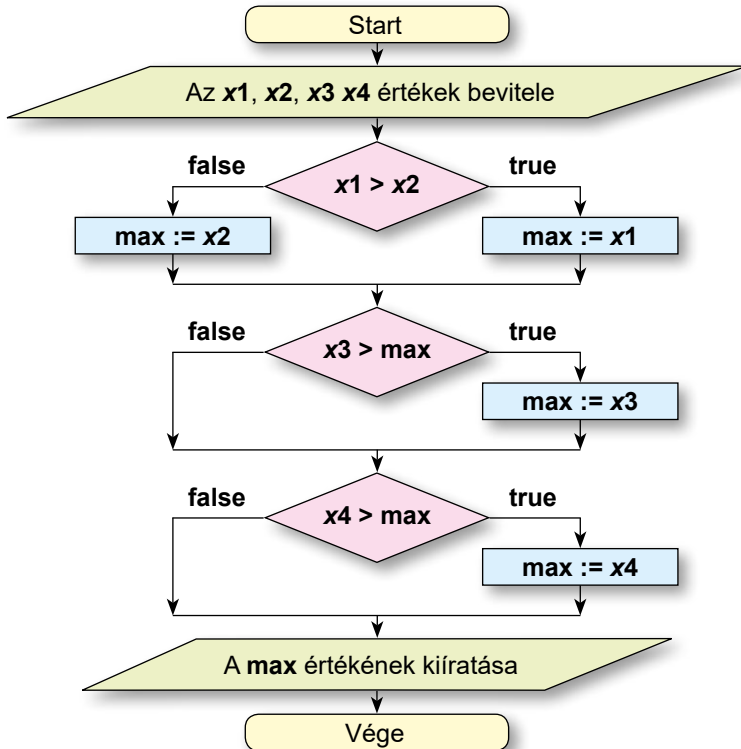
A projekt helyességének ellenőrzéséhez futtathatjuk például a következő bemeneti adatokkal: 1)  $x = 3, y = 3$ ; 2)  $x = 5, y = 2$ ; 3)  $x = 4, y = 10$ .

**4. feladat.** Adott négy egymással nem egyenlő szám. Határozzuk meg a legnagyobb. Elkészítjük a feladat matematikai modelljét:

- **bemeneti adat:** négy darab tetszőleges szám  $x_1, x_2, x_3, x_4$ ;
- **kimeneti eredmény:** a legnagyobb szám közülük.

A feladat megoldásához először meghatározzuk az első két szám közül melyik a nagyobb, és ezt az értéket hozzárendeljük egy változóhoz, például a **max** változóhoz. Ezután összehasonlítjuk ezt a változót egymás után a többi számmal (harmadikkal és negyedikkel). Ha pedig egy szám nagyobb, bizonyul a **max** változó értékénél, akkor ennek a változónak az értékét ezzel a számmal helyettesítjük.



A feladat megoldásának folyamatábrás algoritmusát a 6.63. ábra mutatja.



6.63. ábra. A 4. feladat megoldásának folyamatábrás algoritmus



A projekt a következőképpen néz ki:

|  <b>A Pascalt használóknak</b>  |  <b>A Pythont használóknak</b>  |
|--|--|
| <pre>x1 := StrToFloat(Edit1.Text); x2 := StrToFloat(Edit2.Text); x3 := StrToFloat(Edit3.Text); x4 := StrToFloat(Edit4.Text); if x1 &gt; x2 then max := x1 else max := x2; if x3 &gt; max then max := x3; if x4 &gt; max then max := x4; Label1.Caption := FloatToStr(max);</pre> | <pre>x1 = float(input('Add meg x1: ')) x2 = float(input('Add meg x2: ')) x3 = float(input('Add meg x3: ')) x4 = float(input('Add meg x4: ')) if x1 &gt; x2: max = x1 else: max = x2 if x3 &gt; max: max = x3 if x4 &gt; max: max = x4 print(max)</pre> |

A létrehozott projekt megfelelően fog működni abban az esetben is, ha e négy szám között egyenlő számok lesznek. Ellenőrizték le a következő számokkal: 4, 4, 8, 2.

### Projekttesztelés elágazással

A projekt elágazással történő tesztelése a bemeneti adatsorok szempontjából szükséges, hogy ellenőrizzük a munka helyességét a változók minden lehetséges változatánál.

Például az  $y = \frac{x-5}{x^2-9}$  kifejezés kiszámításának a projektje elágazást tartalmaz, mert végrehajtása attól függ, hogy a nevező nulla-e vagy sem. Ezért tesztelni kell az  $x$  értékeit, mely értékek mellett egyenlő 0-val, vagyis 3 és  $-3$ , és milyen értékeknél nem egyenlő 0-val, például 5 és 2.



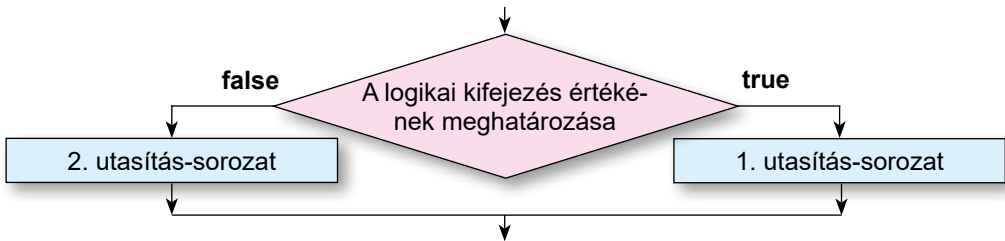
### Használjuk a számítógépet!

Végezzék el a gyakorló feladatot a <https://cutt.ly/ehdrhey> linkről, vagy a QR-kód segítségével!

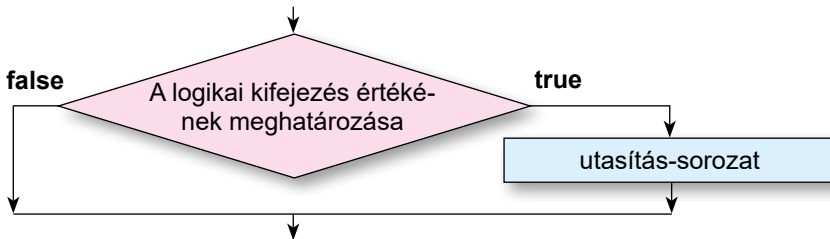


### Összefoglalás

A teljes és a nem teljes elágazások folyamatábráját a 6.64. és 6.65. ábrák mutatják:





6.64. ábra. A teljes elágazás folyamatábrája





6.65. ábra. A nem teljes elágazás folyamatábrája

A teljes elágazás általános alakja a különböző programozási nyelvekben a következő:

|  <i>A Pascalt használóknak</i>                      |  <i>A Pythont használóknak</i> |
|--|---|
| <pre>if &lt;logikai kifejezés&gt; then begin     &lt;utasítás-sorozat 1&gt; end else begin     &lt;utasítás-sorozat 2&gt; end;</pre> | <pre>if &lt;logikai kifejezés&gt;: &lt;utasítás-sorozat 1&gt; else:     &lt;utasítás-sorozat 2&gt;</pre>        |

Az utasítás végrehajtása a következőképpen történik: kiértékelődik a logikai kifejezés; amennyiben az eredmény **true**, akkor az **1. utasítás-sorozat** kerül végrehajtásra, ami után az elágazás utáni első utasításra kerül a vezérlés; amennyiben az eredmény **false**, a **2. utasítás-sorozat** kerül végrehajtásra, ami után az elágazás utáni első utasításra kerül a vezérlés.

A nem teljes elágazás általános alakja a különböző programozási nyelvekben a következő:

|  <i>A Pascalt használóknak</i> |  <i>A Pythont használóknak</i> |
|---|---|
| <pre>if &lt;logikai kifejezés&gt; then begin     &lt;utasítás-sorozat&gt; end;</pre>                            | <pre>if &lt;logikai kifejezés&gt;: &lt;utasítás-sorozat 1&gt;</pre>   |

Az utasítás végrehajtása a következőképpen történik: kiértékelődik a logikai kifejezés; amennyiben az eredmény **true**, akkor az **utasítás-sorozat** végrehajtásra kerül, ami után az elágazás utáni első utasításra kerül a vezérlés; amennyiben az eredmény **false**, az utasítás-sorozat nem kerül végrehajtásra, a vezérlés azonnal az elágazás utáni első utasításra kerül.





### Felelj a kérdésekre!

- 1<sup>o</sup>. Milyen a teljes elágazás folyamatábrája? Ismertesd ennek végrehajtását!
- 2<sup>o</sup>. Milyen a nem teljes elágazás folyamatábrája? Ismertesd ennek végrehajtását!
- 3<sup>o</sup>. Miben különbözik a teljes és nem teljes elágazás végrehajtása?
- 4<sup>o</sup>. Mi a teljes elágazás általános alakja a különböző programozási nyelvekben? Hogyan lehet végrehajtani?
- 5<sup>o</sup>. Mi a nem teljes elágazás általános alakja a különböző programozási nyelvekben? Hogyan lehet végrehajtani?
- 6<sup>o</sup>. Miben különbözik a teljes és a nem teljes elágazás végrehajtása a különböző programozási nyelvekben?
- 7<sup>\*</sup>. Milyen esetekben szükséges a teljes elágazást alkalmazni az algoritmusokban, és milyen esetekben a nem teljes?
- 8<sup>o</sup>. Hogyan válasszunk teszt példákat egy elágazással rendelkező projekt teszteléséhez?
- 9<sup>o</sup>. Hogyan néz ki a többszörös elágazási parancs **Python**ban? Hogyan működik?





**Végezd el a feladatokat!**

1•. Készíts folyamatábrát és magyarázatot a feladatokhoz!

|  A Pascalt használóknak   |  A Pythont használóknak  |
|--|---|
| <pre> if &lt;logikai kifejezés 1&gt; then begin     &lt;utasítás 1&gt; end else if &lt;logikai kifejezés 2&gt; then begin     &lt;utasítás 2&gt; end else begin     &lt;utasítás 3&gt; end;                     </pre> | <pre> if &lt;logikai kifejezés&gt;:     &lt;utasítás 1&gt; else:     if &lt;logikai kifejezés 2&gt;:         &lt;utasítás 2&gt;     else:         &lt;utasítás 3&gt;                     </pre> |

2•. Készíts folyamatábrát, és magyarázd el a parancssor végrehajtását!

|  A Pascalt használóknak   |  A Pythont használóknak                                     |
|--|--|
| <pre> if &lt;logikai kifejezés 1&gt; then begin     &lt;utasítás 1&gt; end; if &lt;logikai kifejezés 2&gt; then begin     &lt;utasítás 2&gt; end;                     </pre> | <pre> if &lt;logikai kifejezés&gt;:     &lt;utasítás 1&gt; if &lt;logikai kifejezés 2&gt;:     &lt;utasítás 2&gt;                     </pre> |

3\*. Magyarázd el, miben különbözik az 1. és 2. feladatok végrehajtása!

4\*. Hozz fel példákat a mindennapi életből, amelyek elágazásokkal írhatók le! Készíts folyamatábrát ezekről!




5\*. Hozz fel példákat a mindennapi életből, amelyek elágazásokkal írhatók le! Készíts folyamatábrát ezekről!

6•. Készíts matematikai modellt, és hozz létre projektet az  $(a + b) - c : (a - b)$  kifejezés értékének kiszámítására! Mentsd a projektet a mappádban létrehozott **feladat 6.8.6** mappába! Írd be a füzetedbe azokat a bemeneti adatokat, amelyekkel tesztelted ezt a projektet!



7•. Készíts matematikai modellt, és hozz létre projektet az  $a + b - c : (a - 2b)$  kifejezés értékének kiszámítására! Mentsd a projektet a mappádban létrehozott **feladat 6.8.7** mappába! Írd be a füzetedbe azokat a bemeneti adatokat, amelyekkel tesztelted ezt a projektet!

8•. Készíts matematikai modellt, és hozz létre projektet, amely összehasonlít két valós számot, kiírja a kisebbet, vagy jelzi, ha a számok egyenlők! Mentsd a projektet a mappádban létrehozott **feladat 6.8.8** mappába!

-  9°. Az automatizált repülőjegy vásárlási rendszer 5 repülési lehetőséget kínált fel  $A$  városból  $B$  városba. Készíts matematikai modellt, és hozz létre egy projektet, amely meghatározza a legolcsóbbat a javasolt járatok közül! MentSD a projektet a mappádban létrehozott **feladat 6.8.9** mappába!
- 10°. Készíts matematikai modellt, és hozz létre egy projektet, amely meghatározza a hat egyenlőtlen valós szám közül a legkisebbet és a legnagyobbat. MentSD a projektet a mappádban létrehozott **feladat 6.8.10** mappába!
-  11°. Készíts matematikai modellt, és hozz létre egy projektet, amely kiszámítja egy tetszőleges szám modulusát! MentSD a projektet a mappádban létrehozott **feladat 6.8.11** mappába! Írd be a füzetedbe azokat a bemeneti adatokat, amelyekkel tesztelted ezt a projektet!
- 12°. Hozz létre projektet, amelyben egy gomb 30 pixellel jobbra mozdul, ha az egérmutatót fölé visszük! Ha a gomb eléri a form jobb szélét, jelenjen meg a bal oldalon! MentSD a projektet a mappádban létrehozott **feladat 6.8.12** mappába!
- 13°. Készíts matematikai modellt, és hozz létre egy projektet, amelyben kiszámíthatja a függvény értékeit:  $y = \begin{cases} 2x - 4, & \text{ha } x < 2; \\ x^2 + 5, & \text{ha } 2 \leq x \leq 7; \\ 3 - 5x, & \text{ha } x > 7! \end{cases}$  MentSD a projektet a mappádban létrehozott **feladat 6.8.13** mappába! Írd be a füzetedbe azokat a bemeneti adatokat, amelyekkel tesztelted ezt a projektet!
-  14°. Készíts matematikai modellt, és hozz létre egy projektet, amelyben kiszámíthatja a függvény értékeit:  $y = \begin{cases} 15 - x, & \text{ha } -1 < x < 4; \\ 6 + 4 & \text{minden más } x\text{-re!} \end{cases}$  MentSD a projektet a mappádban létrehozott **feladat 6.8.14** mappába! Írd be a füzetedbe azokat a bemeneti adatokat, amelyekkel tesztelted ezt a projektet!
- 15\*. Készíts matematikai modellt, és hozz létre egy projektet annak meghatározására, hogy lehetséges-e háromszöget alkotni három, meghatározott hosszúságú szakaszból. Ha igen, akkor ez a háromszög egyenlő oldalú lesz-e? MentSD a projektet a mappádban létrehozott **feladat 6.8.15** mappába! Írd be a füzetedbe azokat a bemeneti adatokat, amelyekkel tesztelted ezt a projektet!
- 16°. Ha a fogyasztó havonta legfeljebb 100 kWh áramot fogyaszt, akkor minden kWh-ért 0,9 hrn-t kell fizetnie, és minden 100 kW-nál többet fogyasztott kWh-ért 1,68 hrn-t kell fizetnie. Készíts matematikai modellt, és hozz létre egy projektet annak meghatározására, hogy a fogyasztónak mennyi pénzt kell fizetnie az elfogyasztott villamos energiáért! MentSD a projektet a mappádban létrehozott **feladat 6.8.16** mappába!
- 17°. Ukrajnában 1 m<sup>3</sup> elfogyasztott gázért 5,45 hrn-t kell fizetni, Lengyelországban pedig 1,61 zlotyt és további 9 zlotyt havonta. Készíts matematikai modellt, és hozz létre egy projektet az azonos mennyiségű Ukrajnában és Lengyelországban elfogyasztott gázért fizetett összeg összehasonlítására! MentSD a projektet a mappádban létrehozott **feladat 6.8.17** mappába!
- 18°. A bank kétféle betétet kínál a betétesnek: az év eredményének 11%-át, hat hónapos eredményének 5%-át. Mindkét betétnél a megadott időnkénti nyereség hozzáadódik a betéthez. Készíts matematikai modellt, és hozz létre egy projektet, amely összehasonlítja az 1 év nyereségét ezeken a betéteken, ha ugyanannyi pénzt tesz be mindkettőre! MentSD a projektet a mappádban létrehozott **feladat 6.8.18** mappába! Írd be a füzetedbe azokat a bemeneti adatokat, amelyekkel tesztelted ezt a projektet!





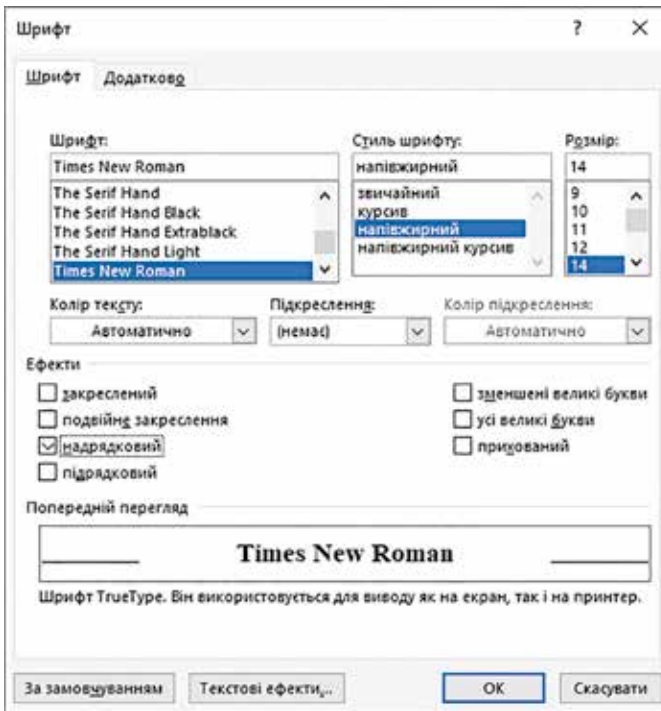
## 6.9. A jelölőnégyzetek. A választógombok

1. Milyen a teljes és a nem teljes elágazás általános alakja a programozási nyelvekben, és hogyan lehet őket végrehajtani?
2. Mire használják a címkéket és a mezőket? Milyen tulajdonságai vannak, hogy változtatjuk meg ezeket?
3. Mire használtak az általatok használt programokban a jelölőnégyzetet és választógombokat?

### A jelölőnégyzet és tulajdonságai

Vizsgáljuk meg a **jelölőnégyzet** komponenst! A jelölőnégyzetek arra szolgálnak, hogy egy lehetőséget, vagy többet, vagy egyet se válaszunk ki a felkínált lehetőségek közül (6.66. ábra).

Ha több jelölőnégyzet van az ablakban, kiválaszthatunk egyet vagy többet, esetleg egyet sem, vagy az összeset. Ha be van jelölve a jelölőnégyzet, akkor egy pipa jelenik meg benne  .



6.66. ábra. A jelölőnégyzetek használatának példája

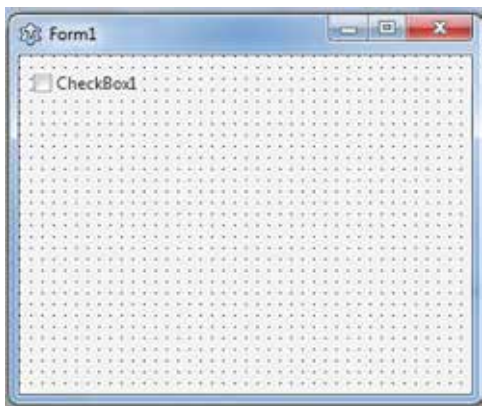


### A Pascalt használóknak

Jelölőnégyzetet elhelyezni a formon a **Standard** eszköztár **TCheckBox**  gomb (ang. *checkbox* – ellenőrző doboz) kétszeri rákattintásával lehet. A jelölőnégyzet a form bal felső sarkában (6.67. ábra) jelenik meg, tulajdonságai pedig megjelennek az

**Objektum Felügyelő komponenslistájában.** Szükség esetén a jelölőnégyzet a form területén áthelyezhető.

A jelölőnégyzet **Caption, Top, Left, Name, Color, Enabled, Visible, Font, Width, Height** tulajdonságok értelmezése megegyeznek azzal, amit a címke esetében tapasztaltunk. A **Width** és **Height** tulajdonságok értékei a jelölőnégyzet melletti szöveg hossza és betűmérete által meghatározott. Ez a szöveg a jelölőnégyzet **Caption** tulajdonságának értéke. Ezt a szöveget, ahogy a címkénél is, beírhatjuk a **Caption** mezőbe, vagy a  gombra kattintva megnyíló **Karakterlánc-szerkesztő**be.



6.67. ábra. Jelölőnégyzet a formon

Ki van-e jelölve a jelölőnégyzet, vagy sem, azt a **Checked** (ang. *checked* – ellenőrizve) tulajdonságának két értéke a **true** (a jelölőnégyzetben a pipa be van kapcsolva) vagy **false** (a pipa nincs bekapcsolva) határozza meg.



### A Pythont használóknak

A jelölőnégyzet létrehozásának parancsa a **Python**ban így néz ki:

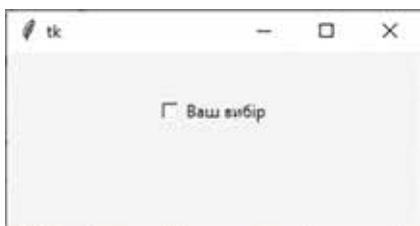
```
<jelölőnégyzet neve> = Checkbutton(text = '<aláírás szövege>')
```

Az így létrehozott jelölőnégyzet ablakban való elhelyezése után, a megadott aláírással fog rendelkezni. Például az alábbi parancs végrehajtása után:

```
flag = Checkbutton(text = 'Választás')
```

```
flag.pack()
```

egy jelölőnégyzet jön létre *flag* névvel és *Választás* aláírással (6.68 ábra).



6.68. ábra. Jelölőnégyzet az ablakban

A címkéhez, beviteli mezőhöz és a gombhoz hasonlóan a jelölőnégyzet is rendelkezik **width, height, bg, fg, font** tulajdonságokkal.

Ahhoz, hogy a projekt során megtudjuk a jelölőnégyzet (kijelölt vagy kijelöletlen) állapotát, hozzá kell rendelni egy speciális logikai változót. Ilyen változó létrehozásához az alábbi parancsot használjuk:

```
<változó neve> = BooleanVar()
```

Ezt a változót az alábbi parancs segítségével kapcsoljuk a jelölőnégyzethez:

```
<jelölőnégyzet neve> = Checkbutton(text = '<aláírás szövege>', variable = <változó neve>)
```

Például az alábbi parancs végrehajtása után:

```
choice = BooleanVar()
```

```
flag = Checkbutton(text = 'Választás', variable = choice)
```

```
flag.pack()
```



egy jelölőnégyzet jön létre *flag* névvel és *Választás* aláírással, valamint a *choice* nevű változó fogja elmenteni a jelölőnégyzet állapotát: **True** – ha a jelölőnégyzet ki van választva, **False** – ha a jelölőnégyzet nincs kiválasztva.

Ennek a változónak az alapértelmezett értéke **False**, és a projekt futtatásakor hiányzik a pipa. Ha a projekt végrehajtása során bejelölésre kerül a jelölőnégyzet, akkor a változó értéke **True** lesz. Ha ismét kijelöli a jelölőnégyzetet, annak állapota ellentétesre változik, és a változó értéke is megváltozik.

Ahhoz, hogy a projekt elindítása után közvetlenül be legyen jelölve a jelölőnégyzet, a **<változó neve>.set(True)** (ang. *set* – beállítás) parancsot használjuk, például **choice.set(True)**.

Ahhoz, hogy meghatározzuk ennek a változónak az értékét a projekt futtatása során, a **<változó neve>.get()** parancsot kell végrehajtani, például **choice.get()**. Ennek a változónak az értékét felhasználhatjuk az elágazást tartalmazó projektekben.

Ha a projektben több jelölőnégyzetet is szeretnénk használni, mindegyikhez külön változót kell rendelnünk, például:

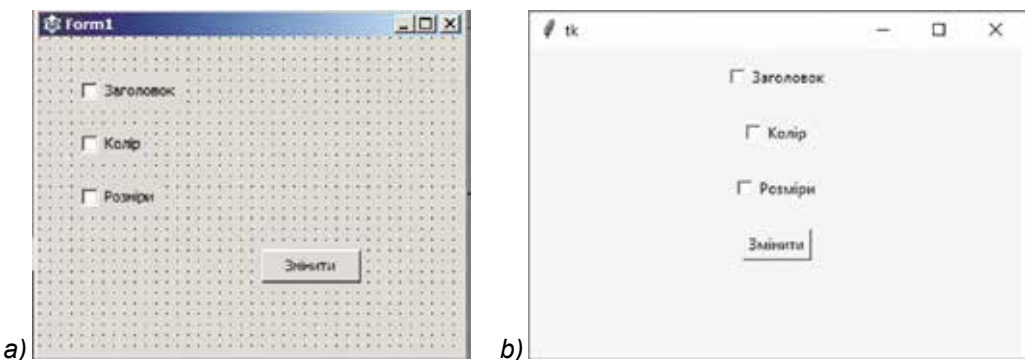
```
choice1 = BooleanVar()
flag1 = Checkbutton(text = 'Cím', variable = choice1)
choice2 = BooleanVar()
flag2 = Checkbutton(text = 'Szín', variable = choice2) choice3 = BooleanVar()
flag3 = Checkbutton(text = 'Méret', variable = choice3)
```

### Jelölőnégyzetek használata

Vizsgáljuk meg, hogy használható a jelölőnégyzet elágazás programozása során. Tegyük fel, hogy a projekt futása során az ablak bizonyos tulajdonságait szeretnénk megváltoztatni (cím, szín, méretek), vagy ezek közül néhányat, esetleg egyiket sem.

E célból helyezzünk el a formon egy gombot és három jelölőnégyzetet (6.69. ábra). Mivel a projekt futása során a jelölőnégyzetek állapota lehet **false**, célszerű ezek

**Checked** állapotát ( a logikai változó értékét) alapértelmezetten hagyni.



6.69. ábra. Jelölőnégyzetek használata: a) **Pascalban**; b) **Pythonban**

Hozzuk létre a gomb **Click** eseménykezelőjét, és helyezzünk el ebben három egymást követő (nem egymásba zárt) elágazásokat:



### A Pascalt használóknak

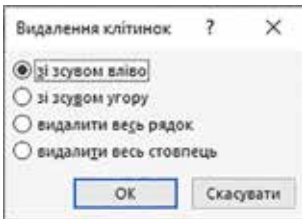
```
if CheckBox1.Checked
then Form1.Caption := 'A cím megváltozott';
if CheckBox2.Checked
then Form1.Color := clRed;
if CheckBox3.Checked
then begin
    Form1.Width := 500;
    Form1.Height := 200;
end;
```



### A Pythont használóknak

```
if choice1.get():
    root.title('A cím megváltozott')
if choice2.get():
    root['bg'] = 'red'
if choice3.get():
    root.geometry('500x200')
```


Ezekben az utasításokban egymástól függetlenül végezzük a jelölőnégyzetek állapotát. A jelölők állapotától – ki van-e jelölve a jelölőnégyzet vagy sem – függően történik meg az ablak tulajdonságainak megváltoztatása. Ha a jelölőnégyzet ki van pipálva, akkor megváltozik az ablak tulajdonsága, ha pedig nincs kijelölve, akkor nem.



6.70. ábra. Példa a választógombok használatára


### Választógombok és tulajdonságaik

Tekintsük meg a **választógomb** komponenst! A választógombok csak egy opció kiválasztására szolgálnak a lehetséges variációk közül (6.70. ábra).

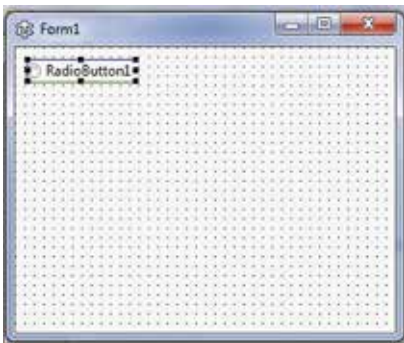
Ha a formon több választógomb van, ezek közül csak egyet lehet kiválasztani. Ha a választógomb ki van választva, akkor annak alakja  lesz.



### A Pascalt használóknak

Választógombot a **Standard** eszköztár **TRadioButton**  (ang. *radiobutton* – rádiógomb) gombjára duplán kattintva helyezhetünk el a formon (6.71. ábra).

A választógomb mindazokkal a tulajdonságokkal rendelkezik, mint a jelölőnégyzet, többek között a **Checked** tulajdonsággal is, amit az elágazás logikai kifejezésében használhatunk.




6.71. ábra. Választógomb a formon



6.72. ábra. A választógombok két csoportja a formon



A formon elhelyezhetünk két, egymástól független választógomb csoportot. Ehhez duplán kattintsunk a **Standard** eszköztár **TRadioGroup** (ang. *radiogroup* – rádió csoport)  gombjára. Ekkor minden csoportból ki lehet választani egy gombot (6.72. ábra).



## A Pythont használóknak

A jelölőnégyzetekkel ellentétben, egy választógomb csoportból csak egyet lehet kiválasztani. Ehhez az összes választógomb a csoportban egy egész típusú változóhoz kell, hogy legyen kapcsolva. Az ilyen változó létrehozására szolgáló parancs a következőképpen néz ki:

**<változó neve> = IntVar()**

A csoporthoz tartozó választógombokat az alábbi paranccsal hozzuk létre:

**<választógomb neve> = Radiobutton(text = '<aláírás szövege>', variable = <változó neve>, value = <változó értéke>)**, ahol:

- a **text** tulajdonság meghatározza a választógomb aláírásának szövegét;
- a **variable** tulajdonság meghatározza a változó nevét, amely a választógombhoz van kapcsolva. Ugyanaz a változó kell, hogy legyen az azonos csoport összes kapcsolójához;
- a **value** tulajdonság meghatározza, hogy milyen értéket vesz fel a változó egy adott választógomb kiválasztásakor.

A 6.73. ábra egy projekt töredékét mutatja, amelyben három, azonos csoportban található választógomb van – *switch1*, *switch2* és *switch3*, amelyek kiválasztása esetén a *choice* változó 1, 2 és 3 értékek egyikét veheti fel.

```
choice = IntVar()
switch1 = Radiobutton(text = "Заголовок", variable = choice, value = 1)
switch1.pack(pady = 10)
switch2 = Radiobutton(text = "Колір", variable = choice, value = 2)
switch2.pack(pady = 10)
switch3 = Radiobutton(text = "Розміри", variable = choice, value = 3)
switch3.pack(pady = 10)
```

6.73. ábra. Egy projekt töredéke három, azonos csoportban található választógomb esetében

A jelölőnégyzethez hasonlóan a választógomb is rendelkezik a **width**, **height**, **bg**, **fg**, **font** tulajdonságokkal.

Alapértelmezés szerint a **choice** változó értéke 0. Mivel ez az érték nem felel meg egyetlen választógombnak sem, a projekt indítása után egy választógomb se lesz kijelölve. Annak érdekében, hogy az egyik választógomb ki legyen jelölve, a változónak az adott választógomb értékét kell megadni az alábbi parancs segítségével:

**<változó neve>.set(<változó értéke>)**

Például a **choice.set (1)** parancs végrehajtása után kiválasztásra kerül az első választógomb.

Ahhoz, hogy a projekt futása során megtudjuk melyik választógomb volt kiválasztva, a **<változó neve>.get()** a parancsot használjuk, például **choice.get()**.

Ha több független választógomb csoportot szeretnénk elhelyezni az ablakban, akkor mindegyik csoporthoz külön változót kell társítanunk.

## Választógombok használata

Figyeljük meg, hogy változtathatjuk meg az ablak egy tulajdonságát (vagy a címét, a színét, a méretét) választógomb alkalmazásával!





## A Pascalt használóknak

A projektben használjunk egy gombot és három választógombot (6.74. ábra)!

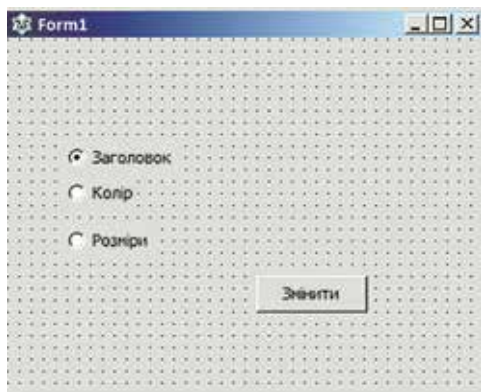
Állítsuk be a választógombok tulajdonságainak értékeit úgy, hogy közülük csak az első legyen kiválasztva! A választógombok beillesztésekor változtassuk meg az első választógomb **Checked** tulajdonságát **true**-ra, az összes többi választógombot pedig hagyjuk a **false**-on!

Hozzunk létre egy eljárást a gomb **Click** eseménykezelőjének, és helyezzük el benne a következő utasításokat:

```

if RadioButton1.Checked
then Form1.Caption := 'A cím megváltozott'
else if RadioButton2.Checked
then Form1.Color := clRed
else begin
    Form1.Width := 600;
    Form1.Height := 300;
end;

```



6.74. Választógombok használata a Pascalban

Az utasítás végrehajtása a **RadioButton1.Checked** kiértékelésével kezdődik.

Ha a **Click** esemény bekövetkezésekor az első választógomb volt bekapcsolva, akkor **RadioButton1.Checked = true**. Ezért az ablak címe lesz megváltoztatva, az elágazás egyéb tulajdonságai pedig kimaradnak.

Ha a **Click** esemény bekövetkezésekor a második választógomb volt bekapcsolva, akkor **RadioButton1.Checked = false**. Ezért az **Else** utáni utasítás kerül végrehajtásra. Ez az utasítás szintén egy elágazás. A végrehajtás során a **RadioButton2.Checked** kerül kiértékelésre.

Mivel most a második gomb volt bekapcsolva, ezért az érték itt **true**. Itt az ablak háttérszíne lesz megváltoztatva, az elágazás többi utasítása kimarad.

Ha a **Click** bekövetkezésekor a harmadik gomb volt bekapcsolva, akkor **RadioButton1.Checked = false**. Ezért az **Else** után utasításra kerül a vezérlés, vagyis a **RadioButton2.Checked** kerül kiértékelésre. Mivel a második gomb sem volt bekapcsolva, értéke ennek is **false**. Ezért a második **Else** után utasítás kerül végrehajtásra, vagyis az ablak méretének megváltoztatása.

Ha a projekt indítása előtt egyik gomb **Checked** tulajdonságát se változtattuk meg (alapértelmezetten hagytuk), akkor a projekt indításakor egyik sem lesz kijelölve. Ha a projekt futása során nem kapcsoljuk be egyiket sem, és lenyomjuk a gombot, akkor az ablak mérete változik meg (*gondoljátok át önállóan, miért történik ez!*). Hogy ez ne történjen így, mindenképpen be kell kapcsolni futtatás előtt az egyik választógombot, vagyis a második **Else** után el kell helyezni még egy elágazást:





```

if RadioButton1.Checked
then Form1.Caption := 'A cím megváltozott'
else if RadioButton2.Checked
then Form1.Color := clRed
else if RadioButton3.Checked
then begin
    Form1.Width := 600;
    Form1.Height := 300;
end;
    
```



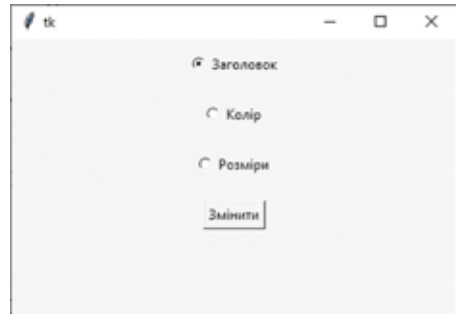
## A Pythont használóknak

Használjunk a projektben három választógombot és egy gombot (6.75 ábra)! Hozzunk létre egy Click eljárást – eseménykezelőt a gomb számára és vigyük be az alábbi parancsokat:

```

if choice.get() == 1:
    root.title('A cím megváltozott')
elif choice.get() == 2:
    root['bg'] = 'red'
else:
    root.geometry('600x300')
    
```

A parancs végrehajtása a **choice** változó értékének meghatározásával kezdődik. Ha a **Click** esemény bekövetkezésekor a gomb számára az első választógomb lesz kijelölve, akkor a **choice** logikai kifejezés értéke igaz lesz. **get() == 1**. Ezért az ablak címének megváltoztatására szolgáló parancs végrehajtásra kerül, és az elágazási parancs más részei nem kerülnek végrehajtásra.



6.75. ábra. Választógombok használata a Pythonban

Ha a **Click** esemény bekövetkezésekor a gomb számára a második választógomb lesz kijelölve, akkor a **choice.get() == 1** logikai kifejezés értéke **False** lesz. Ezért az **elif** kulcsszó után az állapotellenőrzés parancs teljesül. Végrehajtásakor meghatározza a **choice** változó értékét. Mivel a második választógombot választottuk, ez az érték 2. Ezért az ablak színének megváltoztatására szolgáló parancs kerül végrehajtásra, és az elágazási parancs más részei nem kerülnek végrehajtásra.

Ha a **Click** esemény bekövetkezésekor a gomb számára a harmadik választógomb lesz kijelölve, akkor a **choice.get() == 1** logikai kifejezés értéke **False** lesz. Ezért az **elif** kulcsszó után az állapotellenőrzés parancs teljesül. Mivel a második választógomb nincs kijelölve, ezért a **choice.get() == 2** logikai kifejezés értéke **False** lesz. Ezután az **else** kulcsszó után szereplő parancsok kerülnek végrehajtásra, vagyis az ablak átméretezésére szolgáló parancsok.

Ha a projekt végrehajtása során egyik kapcsolót sem választjuk ki, viszont a gombot megnyomjuk, akkor az ablak mérete változik meg (*gondoljátok át önállóan, miért történnek ilyen változások!*). Ennek megakadályozása érdekében egy másik **elif** parancsot kell használni az **else** kulcsszó helyett:

```

if choice.get() == 1:
    root.title('A cím megváltozott')
elif choice.get() == 2:
    root['bg'] = 'red'
elif choice.get() == 3:
    root.geometry('600x300')
    
```

**Használjuk a számítógépet!**

Végezzétek el a gyakorlati feladatot a <https://cutt.ly/ShdrjKU> linkről, vagy a QR-kód segítségével!

**Összefoglalás**

A jelölőnégyzetek arra szolgálnak, hogy egy lehetőséget, vagy többet, vagy egyet sem válasszunk ki a felkínáltak közül. Ha több jelölőnégyzet van az ablakban, kiválaszthatunk egyet vagy többet, esetleg egyet sem, vagy az összeset. A jelölőnégyzetek állapotának (kijelölt vagy jelöletlen) feldolgozásához a projekt mindegyikéhez egymás után rögzített, egymástól független elágazási utasításokat tartalmaz.

**A Pascalt használóknak**

Jelölőnégyzetet elhelyezni a formon a **Standard** eszköztár  **TCheckBox** gomb kétszeri rákattintásával lehet.

Hogy ki van-e jelölve a jelölőnégyzet, vagy sem, azt a **Checked** (ang. *checked* – ellenőrizve) tulajdonságának két értéke – a **true** (a jelölőnégyzetben a pipa be van kapcsolva) vagy **false** (a pipa nincs bekapcsolva) határozza meg. Alapértelmezetten ez a tulajdonság **false**-ra van állítva.

**A Pythont használóknak**

A jelölőnégyzet létrehozásának parancsa a **Python**ban így néz ki:  
`<jelölőnégyzet neve> = Checkbutton(text = '<aláírás szövege>', variable = <változó neve>)`, ahol egy logikai változót használnak a jelölőnégyzet állapotának meghatározásához a következő paranccsal: `<a változó neve> = BooleanVar()`.

Ahhoz, hogy a projekt elindítása után közvetlenül be legyen jelölve a jelölőnégyzet, a `<változó neve>.set(True)` parancsot használjuk, például `choice.set(True)`.

Ahhoz, hogy meghatározzuk ennek a változónak az értékét a projekt futtatása során, a `<változó neve>.get()` parancsot kell végrehajtani, például `choice.get()`.

A **választógombok** a lehetséges lehetőségek közül csak az egyik kiválasztására szolgálnak. Egy választógomb csoportból csak egyet lehet kiválasztani. Ezért a választógombokat olyan esetekben használják, amikor a projekt végrehajtása során csak az egyik lehetőséget kell kiválasztani. A választógombok állapotának (kijelölt vagy jelöletlen) feldolgozásánál mindegyikhez egy elágazási parancs tartozik a projektben, amelyek mindegyike be van ágyazva az előzőbe.

**A Pascalt használóknak**

Választógombot a **Standard** eszköztár **TRadioButton**  gombjára duplán kattintva helyezhetünk el a formon.

Hogy a választógomb ki van-e választva vagy sem, azt a **Checked** tulajdonsága határozza meg, hasonlóan a jelölőnégyzethez.



A formon elhelyezhetünk két egymástól független választógomb csoportot. Ehhez duplán kattintsunk a **Standard** eszköztár **TRadioGroup** gombjára! Ekkor minden csoportból ki lehet választani egy gombot.



### A Pythont használóknak

A választógomb létrehozásának parancsa a **Python**ban így néz ki:

**<választógomb neve> = Radiobutton(text = '<aláírás szövege>', variable = <változó neve>, value = <változó értéke>)**, ahol a változó a kiválasztott választógomb meghatározására szolgál, és a **<változó neve> = IntVar()** paranccsal hozható létre.

Ha azt akarjuk, hogy az egyik csoport egy jelölőnégyzete rögtön ki legyen jelölve, akkor a jelölőnégyzet változójának a **<változó neve>.set(<változó értéke>)** paranccsal állítsuk be a megfelelő értéket, például **choice.set(1)**.

Ahhoz, hogy a projekt futása során megtudjuk, melyik választógomb volt kiválasztva, a **<változó neve>.get()** a parancsot használjuk, például **choice.get()**.

Több független kapcsolócsoportot is elhelyezhetünk az ablakban. Ehhez minden kapcsolócsoportot külön változóhoz kell társítani. Ezután mindegyik csoportban kiválaszthatunk egy jelölőnégyzetet.





### Felelj a kérdésekre!

- 1°. Milyen célra használhatjuk a jelölőnégyzetet és a választógombot?
- 2°. Milyen értékeket vehet fel a jelölőnégyzet és a választógomb **Checked** tulajdonsága?
- 3°. Hogyan lehet meghatározni a jelölőnégyzet állapotát egy **Python**ban kifejlesztett projektben?
- 4°. Hogyan kell felírni az elágazásokat, amelyek ellenőrzik, hogy be van-e jelölve egy jelölőnégyzet?
- 5°. Hogyan kell felírni az elágazásokat, amelyek ellenőrzik, hogy be van-e jelölve egy választógomb?
- 6°. Hogyan tudjuk megállapítani, hogy melyik választógomb van bekapcsolva a **Python**ban kifejlesztett projektekben?
- 7°. Mire szolgál a választógombok csoportja?





### Végezd el a feladatokat!

- 1°. Rajzold meg az alábbi utasítások folyamatábráját, és magyarázd el a működését:

|  <b>A Pascalt használóknak</b>                  |  <b>A Pythont használóknak</b>  |
|--|--|
| <pre>if CheckBox1.Checked then begin     &lt;utasítás 1&gt; end; if CheckBox2.Checked then begin     &lt;utasítás 2&gt; end;</pre> | <pre>if choice 1.get():     &lt;utasítás 1&gt; if choice 2.get():     &lt;utasítás 2&gt;</pre> <p>ahol <b>choice1</b> és <b>choice2</b> – logikai típusú változók, amelyek a jelölőnégyzethez vannak kötve</p> |

2°. Rajzold meg az alábbi utasítások folyamatábráját, és magyarázd el a működését:

|  <b>A Pascalt használóknak</b>   |  <b>A Pythont használóknak</b>  |
|---|--|
| <pre> if RadioButton1.Checked then begin     &lt;utasítás 1&gt; end else if RadioButton2.Checked then begin     &lt;utasítás 2&gt; end else begin     &lt;utasítás 3&gt; end;</pre> | <pre> if choice.get() == 1:     &lt;utasítás 1&gt; elif choice.get() == 2:     &lt;utasítás 2&gt; else:     &lt;utasítás 3&gt;</pre> <p>ahol <b>choice</b> – egész típusú változó, amely jelölőnégyzetek csoportjához van társítva</p> |

3\*. Magyarázd meg, miben különbözik az 1. feladatban szereplő elágazás működése a 2. feladatban leírt elágazás működésétől!

4\*. Fogalmazz meg egy projektfeladatot, amelynek megoldására használhatjuk az 1. feladatot!



5\*. Fogalmazz meg egy projektfeladatot, amelynek megoldására használhatjuk a 2. feladatot!

6°. Hozz létre projektet, amiben két valós szám összegét vagy különbségét, vagy szorzatát, vagy hányadosát számítod ki egy lista kiválasztott elemének megfelelően! MentSD a projektet a mappádban létrehozott **feladat 6.9.6** mappába!



7°. Hozz létre projektet, amiben két valós szám összegét vagy különbségét, vagy szorzatát, vagy hányadosát számítod ki jelölőnégyzetek állapotának megfelelően! MentSD a projektet a mappádban létrehozott **feladat 6.9.7** mappába!

8°. A munkás alkatrészeket készít. Ismerjük az egy hónap alatt elkészített alkatrészek számát és az alkatrész árát. Amennyiben a munkás teljesíti a tervet, 10% jutalmat kap, ha túlteljesíti, akkor 30%-ot. Hozz létre projektet, amely meghatározza a munkás havi fizetését! MentSD a projektet a mappádban létrehozott **feladat 6.9.8** mappába!



9°. A munkás egy hónap alatt  $x$  alkatrészt készít el, amelyek mindegyikéért  $y$  hrinyát kap. Ha a munkásnak van gyereke, akkor 20% kiegészítést kap a béréhez. Ha a hónap során a munkás megsértette a munkarendet, akkor a fizetéséből 15%-ot levonnak. Hozz létre projektet, amely meghatározza a munkás havi fizetését! MentSD a projektet a mappádban létrehozott **feladat 6.9.9** mappába!

10°. Hozz létre egy projektet, amelyben egy gomb és a jelölőnégyzetek két csoportja található. Az első csoport jelölőnégyzetei határozzák meg az ablak színét, a második csoport jelölőnégyzetei pedig azt, hogy mennyire változik meg az ablak mérete. MentSD a projektet a mappádban létrehozott **feladat 6.9.10** mappába!



## 11. SZÁMÚ GYAKORLATI MUNKA

### Elágazásokat tartalmazó projektek

**Figyelem!** A számítógép használata során tartsd be a balesetvédelmi szabályokat és az egészségügyi előírásokat!

1. Készíts matematikai modellt, és hozz létre egy projektet az  $x = (a + 3 * b) / (2 * a - b)$  kifejezés kiszámítására! Mentsd a projektet a mappában létrehozott **gyakorlati11\_1** mappába!
2. Készíts matematikai modellt, és hozz létre egy projektet négy egyenlőtlen szám legkisebb értékének megtalálásához! Mentsd a projektet a mappában létrehozott **gyakorlati11\_2** mappába!
3. A betétes egy bizonyos összeget szeretne elhelyezni a bankba 2 évre. A bank háromféle betétet kínál: évi 9%-kal, évi 10%-kal és évi 11%-kal. Az első két betétnél a nyereséget évente hozzáadják a betéthez, a harmadikban – nem. Készíts matematikai modellt, és hozz létre egy projektet jelölőnégyzetek segítségével annak a pénzüsszegnek a kiszámításához, amelyet a betétes 2 év múlva kaphat, ha az egyik betétet választja. Mentsd a projektet a mappában létrehozott **gyakorlati11\_3** mappába!

## 6.10. Számlálás ciklus



1. Milyen folyamatokat nevezünk ciklikusnak? Hozzatok fel példákat!
2. Mit nevezünk ciklusnak egy algoritmusban? Mi a ciklusmag?
3. Mi a számlálás ciklus általános alakja a programozási nyelvekben? Hogyan működik?

### A számlálás ciklus utasításai

Már tudjátok, hogy sok algoritmusban egy vagy több utasítást egymás után többször kell megismételni. Ezeket az utasításokat nem tanácsos annyiszor beépíteni az algoritmusba, ahányszor meg kell ismételni őket, mert az ismétlések száma nagyon nagy, vagy akár előre ismeretlen lehet.

Ilyen esetekben *ciklusokat* kell használni az algoritmusokban. Emlékezzünk vissza arra, hogy **egy ciklus az algoritmusban** nem más, mint az algoritmus egy része, amelyet egymás után többször is meg lehet ismételni!

A programozási nyelvekben többféle ciklusokat megvalósítható utasítás létezik. Ezek egyike a **számlálás ciklus**. Ezek alkalmazása akkor célszerű, ha az ismétlések száma már a ciklus végrehajtása előtt ismert.



### A Pascalt használóknak

A számlálás ciklus általános alakja a **Pascalban** a következő:  
**for <változónév> := <kifejezés1> to <kifejezés2> do begin**

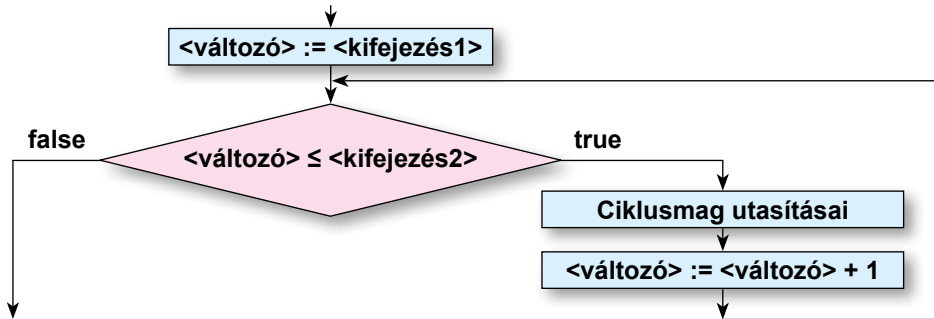
**<a ciklusmag utasításai>**

**end;**

A **for <változónév> := <kifejezés1> to <kifejezés2> do** (*for* – valami részére, *to* – valameddig, *do* – tenni) sor a **számlálás ciklus feje**. A ciklusfejben szereplő változó (i), amely a := előtt szerepel, a **ciklusváltozó**.

A **ciklusváltozó**, **kifejezés1** és **kifejezés2** csak egész értékeket vehet fel. Ha a ciklusmag csak egyetlen utasítást tartalmaz, a **begin** és **end** utasítás-zárójelek elhagyhatók.

A számlálós ciklus folyamatábráját a 6.76. ábrán láthatjuk.



6.76. ábra. A számlálós ciklus folyamatábrája

A számlálós ciklus a következőképpen működik:

1. A **ciklusváltozó** felveszi a **kifejezés1** értéket.
2. Kiértékelődik a **változó ≤ kifejezés2** logikai kifejezés.
3. Ha a kifejezés értéke **true**, végre kell hajtani a ciklusmagot, majd át kell lépni a 4. utasításra, ha **false**, a vezérlést át kell adni a ciklus utáni első utasításnak!
4. A ciklusváltozót meg kell növelni eggyel!
5. Át kell lépni a 2. utasításra!

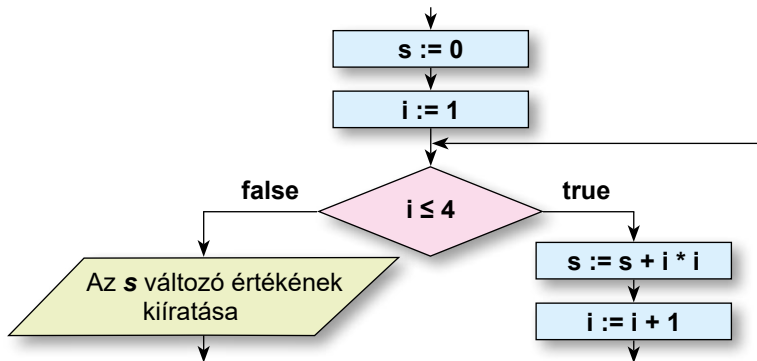
A ciklus működésének bemutatására lássuk az  $1^2 + 2^2 + 3^2 + 4^2$  összeg kiszámításának algoritmusát:

```

var s, i: integer;
begin
  s := 0;
  for i := 1 to 4 do
    begin
      s := s + i * i;
    end;
    Label1.Caption := IntToStr(s);
  end;

```

A projekt egy részletének a folyamatábráját a 6.77. ábra szemlélteti.



6.77. ábra. A projekt egy részletének a folyamatábrája





Hajtsuk végre ezt a programrészletet:

| Utasítás                                   | A végrehajtás eredménye          |
|--|----------------------------------|
| <code>s := 0</code>                        | <code>s = 0</code>               |
| <code>i := 1</code>                        | <code>i = 1</code>               |
| <code>i &lt;= 4</code>                     | <code>(1 &lt;= 4) = true</code>  |
| <code>s := s + i * i</code>                | <code>s = 0 + 1 * 1 = 1</code>   |
| <code>i := i + 1</code>                    | <code>i = 2</code>               |
| <code>i &lt;= 4</code>                     | <code>(2 &lt;= 4) = true</code>  |
| <code>s := s + i * i</code>                | <code>s = 1 + 2 * 2 = 5</code>   |
| <code>i := i + 1</code>                    | <code>i = 3</code>               |
| <code>i &lt;= 4</code>                     | <code>(3 &lt;= 4) = true</code>  |
| <code>s := s + i * i</code>                | <code>s = 5 + 3 * 3 = 14</code>  |
| <code>i := i + 1</code>                    | <code>i = 4</code>               |
| <code>i &lt;= 4</code>                     | <code>(4 &lt;= 4) = true</code>  |
| <code>s := s + i * i</code>                | <code>s = 14 + 4 * 4 = 30</code> |
| <code>i := i + 1</code>                    | <code>i = 5</code>               |
| <code>i &lt;= 4</code>                     | <code>(5 &lt;= 4) = false</code> |
| <code>Label1.Caption := IntToStr(s)</code> | <code>Label1.Caption = 30</code> |

Felhívjuk a figyelmet arra, hogy a ciklus lefutása után a ciklusváltozó értéke 1-gyel nagyobb, mint a ciklusfejben a **kifejezés2** értéke. Ezt az értéket szükség esetén fel lehet használni a program következő részében.



### Ha többet szeretnél tudni

Van a számlálós ciklusnak olyan változata, amelyben a ciklusváltozó nem növekszik, hanem csökken:

```
for <változónév> := <kifejezés1> downto <kifejezés2> do
begin
  <a ciklusmag utasításai>
end;
```

(ang. *downto* – lefelé valaméddig).

Ez a ciklus addig fut majd, amíg a ciklusváltozó értéke nagyobb vagy egyenlő a **kifejezés2** értékénél (vagyis, amíg kisebb nem lesz annál).



### A Pythont használóknak

A számlálós ciklus parancsának általános kinézete Pythonban:

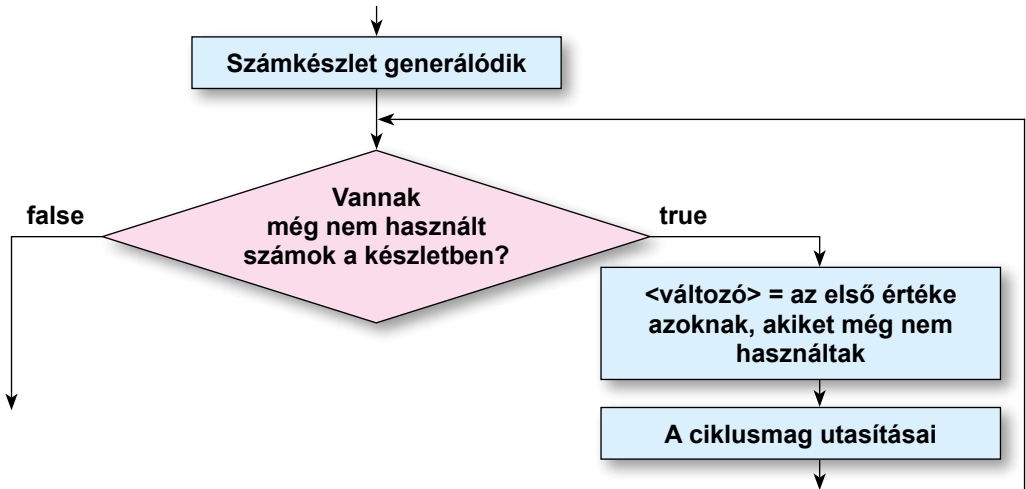
```
for <változó neve> in range(<ismétlések száma>):
  <a ciklusmag utasításai>
```

A **for <változó neve> in range(<ismétlések száma>)** (ang. *for* – valami részére, *in range* – hatótávolságban) sor a számlálós ciklus feje. A ciklusfejben szereplő változó a ciklusváltozó.

A **range(<ismétlések száma>)** függvény, egy sor egymást követő egész számot alkot (generál), amelyek közül az első 0, az utolsó pedig az adott ismétlésszámnál eggyel kevesebb (összesen annyi szám van amennyi az ismétlésszám). Például

`range(3)` függvény egy 3 számot tartalmazó számkészletet generál, melynek elemei a 0, 1, 2 számok. A ciklusszámláló változó következetesen vesz értékeket ebből a számkészletből.

A 6.78 ábra mutatja a számlálós ciklus folyamatábráját a **Python**ban.



6.78. ábra. A számlálós ciklus folyamatábrája **Python**ban

A számlálós ciklust a következőképpen hajtjuk végre:

1. A **`range(<ismétlések száma>)`** függvény 0-tól (**`ismétlések száma - 1`**)-ig generál számot.
2. Ellenőrzés megy végbe, hogy vannak-e olyan számok a készletben, amelyek még nem voltak használatban, és amelyek a ciklusváltozó új értékévé válhatnak.
3. Ha az ellenőrzés eredménye **True**, akkor a ciklusváltozó a még fel nem használt számkészletek közül az első értékét veszi fel, majd a 4. lépés kerül végrehajtásra; ha **False**, akkor a ciklus végrehajtása befejeződik, majd a ciklust követő parancs kerül végrehajtásra.
4. A ciklusmag utasításai végrehajtásra kerülnek.
5. Megismétlődik a 2. lépés.

A számlálós ciklus összes utasításnak, amely a ciklusmagban helyezkedik el, azonos behúzással kell rendelkeznie a **for** szó első betűjéhez képest.

A számlálós ciklus végrehajtásának szemléltetéséhez megfigyelünk egy olyan projektet, amelyben az  $1^2 + 2^2 + 3^2 + 4^2$  kifejezés összegét határozzuk meg.

```
s = 0
```

```
for i in range(4):
```

```
    s = s + (i + 1) * (i + 1)
```

```
print('Az összeg egyenlő', s)
```

Hajtsuk végre ezt a projektet:

| Utasítás   | A végrehajtás eredménye |
|--|-------------------------|
| <code>s = 0</code>   | <code>s = 0</code>      |
| <code>range(4)</code>  | 0, 1, 2, 3              |
| Vannak még nem használt számok a készletben?   | True                    |
| az <code>i</code> változó a fel nem használt számok halmazából az első értéket veszi fel | <code>i = 0</code>      |



A táblázat folytatása

| Utasítás  | A végrehajtás eredménye           |
|---|-----------------------------------|
| $s = s + (i + 1) * (i + 1)$   | $s = 0 + (0 + 1) * (0 + 1) = 1$   |
| Vannak még nem használt számok a készletben?                                  |                                   |
| Az $i$ változó a fel nem használt számok halmazából az első értéket veszi fel | $i = 1$                           |
| $s = s + (i + 1) * (i + 1)$   | $s = 1 + (1 + 1) * (1 + 1) = 5$   |
| Vannak még nem használt számok a készletben?                                  | True                              |
| Az $i$ változó a fel nem használt számok halmazából az első értéket veszi fel | $i = 2$                           |
| $s = s + (i + 1) * (i + 1)$   | $s = 5 + (2 + 1) * (2 + 1) = 14$  |
| Vannak még nem használt számok a készletben?                                  | True                              |
| Az $i$ változó a fel nem használt számok halmazából az első értéket veszi fel | $i = 3$                           |
| $s = s + (i + 1) * (i + 1)$   | $s = 14 + (3 + 1) * (3 + 1) = 30$ |
| Vannak még nem használt számok a készletben?                                  | False                             |
| print('az összeg egyenlő', s)   | Az összeg egyenlő 30              |



### Ha többet szeretnél tudni

A `range(<kezdet>, <vége>, <lépés>)` függvény általános esetben három egész típusú argumentummal rendelkezik:

- **kezdet** – az első egész szám, amely a generált halmazba kerül;
- **vége** – egész szám, amely nem kerül a generált számok halmazába és mindegyik halmazban található számnál nagyobb, ha a lépés  $> 0$ , viszont kisebb az összes számnál a halmazban, ha a lépés  $< 0$ ;
- **lépés** – a generált számhalmaz két szomszédos egész száma közötti különbség.

Például a `range(3, 9, 2)` függvény olyan számok halmazát generálja, hogy:  $3$ ,  $3 + 2 = 5$ ,  $5 + 2 = 7$ . A következő szám  $7 + 2 = 9$  már nem kerül a számhalmazba, mivel nem kisebb, mint a függvény második argumentuma.

### A projekt tesztelése számlálós ciklussal

Ha számlálós ciklust tartalmazó projektet tesztelünk, célszerű a következő bemeneti adatsorokat választani, hogy a ciklusmag különböző számú alkalommal kerüljön végrehajtásra: 1-szer; 2-szer; 3-szor; 5-ször; akárhányszor.



### Használjuk a számítógépet!



Végezzétek el a gyakorlati feladatot a <https://cutt.ly/lhdrkPJ> linkről, vagy a QR-kód segítségével!



### Összefoglalás

A **ciklus az algoritmusban** nem más, mint az algoritmus egy része, amelyet egymás után többször is meg lehet ismételni. Ezek alkalmazása akkor célszerű, ha az ismétlések száma már a ciklus végrehajtása előtt ismert.



### A Pascalt használóknak

A számlálós ciklus általános alakja a **Pascalban** a következő:

```
for <változónév> := <kifejezés1> to <kifejezés2> do  
begin  
  <a ciklusmag utasításai>  
end;
```

A **for** <változónév> := <kifejezés1> **to** <kifejezés2> **do** sor a számlálós ciklus feje. A ciklusfejben szereplő változó (i), amely a := előtt szerepel, a ciklusváltozó. A ciklusváltozó, kifejezés1 és kifejezés2 csak egész értékeket vehet fel.

A számlálós ciklus a következőképpen működik:

1. A ciklusváltozó felveszi a kifejezés1 értéket.
2. Kiértékelődik a változó  $\leq$  kifejezés2 logikai kifejezés.
3. Ha a kifejezés értéke **true**, végre kell hajtani a ciklusmagot, majd át kell lépni a 4. utasításra, ha **false**, a vezérlést át kell adni a ciklus utáni első utasításnak.
4. A ciklusváltozót meg kell növelni eggyel.
5. Át kell lépni a 2. utasításra.



### A Pythont használóknak

A számlálós ciklus parancsának általános kinézete **Pythonban**:

```
for <változó neve> in range(<ismétlések száma>):  
  <a ciklusmag utasításai>
```

A **for** <változó neve> **in** **range**(<ismétlések száma>) sor a számlálós ciklus feje. A ciklusfejben szereplő változó a ciklusváltozó.

A **range**(<ismétlések száma>) függvény, egy sor egymást követő egész számot alkot (generál), amelyek közül az első 0, az utolsó pedig az adott ismétlésszámnál egyel kevesebb. A ciklusszámláló változó következetesen vesz értékeket ebből a számkészletből.

A számlálós ciklust a következőképpen hajtjuk végre:

1. A **range**(<ismétlések száma>) függvény 0-tól (ismétlések száma – 1)-ig generál számot.
2. Ellenőrzés megy végre, hogy vannak-e olyan számok a készletben, amelyek még nem voltak használatban, és amelyek a ciklusváltozó új értékévé válhatnak.
3. Ha az ellenőrzés eredménye **True**, akkor a ciklusváltozó a még fel nem használt számkészletek közül az első értékét veszi fel, majd a 4. lépés kerül végrehajtásra; ha **False**, akkor a ciklus végrehajtása befejeződik, majd a ciklust követő parancs kerül végrehajtásra.
4. A ciklusmag utasításai végrehajtásra kerülnek.
5. Megismétlődik a 2. lépés.

A számlálós ciklus összes utasításnak, amely a ciklusmagban helyezkedik el, azonos behúzással kell rendelkeznie a **for** szó első betűjéhez képest.



### Felelj a kérdésekre!

- 1°. Mi a ciklus az algoritmusban?
- 2°. Milyen a számlálós ciklus általános alakja a programozási nyelvekben?





- 3°. Hogy működik a számlálós ciklus a programozási nyelvekben?
- 4°. Hogy néz ki a számlálós ciklus folyamatábrája? Ismertesd ennek végrehajtását!
- 5°. Előfordulhat-e, hogy a számlálós ciklus csak egyszer kerüljön végrehajtásra? Válaszodat indokold! Hozz fel példát!
- 6°. Előfordulhat-e, hogy a számlálós ciklus egyszer sem kerül végrehajtásra? Válaszodat indokold! Hozz fel példát!
- 7\*. Előfordulhat-e, hogy a számlálós ciklus végtelenszer kerül végrehajtásra? Válaszodat indokold! Hozz fel példát!
- 8°. Milyen bemeneti adatsorokat célszerű választani, hogy a ciklusmag különböző számú alkalommal kerüljön végrehajtásra?
- 9\*. Mi a különbség a szekvencia, az elágazás és a ciklus között?





**Végezd el a feladatokat!**

- 1°. Állítsd össze az alábbi programrészletek folyamatábráját, és hajtsd végre azt!




|    |  <b>A Pascalt használóknak</b> |  <b>A Pythont használóknak</b> |
|----|---|---|
| a) | <pre>n := 1; for i := 1 to 4 do n := n * i;</pre>   | <pre>n = 1 for i in range(4): n = n * (i + 1)</pre>   |
| b) | <pre>p := 0; a := 2; for i := 3 to 6 do begin a := 3 * a + 1; p := p + a; end;</pre>                            | <pre>p = 0 a = 2 for i in range(3, 7, 1): a = 3 * a + 1 p = p + a</pre>   |




- 2°. Állítsd össze az alábbi programrészletek folyamatábráját, és hajtsd végre azt!

|    |  <b>A Pascalt használóknak</b> |  <b>A Pythont használóknak</b>                     |
|----|---|---|
| a) | <pre>n := 0; for i := 1 to 4 do n := n + i * i * i;</pre>   | <pre>n = 1 for i in range(4): n = n + (i + 1)**3 Művelet ** hatványozást jelent, esetünkben az i + 1 a 3-ik hatványra emelkedik</pre> |
| b) | <pre>p := 0; a := 4; for i := 2 to 5 do begin a := 2 * a - 1; p := p + a; end;</pre>                              | <pre>p = 0 a = 4 for i in range(2, 6, 1): a = 2 * a - 1 p = p + a</pre>   |

- 3°. Állíts össze algoritmust 5 szám összegének kiszámítására, amelyek közül az első 7 és minden szám 3-mal haladja meg az előzőt! Hajtsd végre az algoritmust!

-  4°. Állíts össze algoritmust 6 szám összegének kiszámítására, amelyek közül az első 100 és minden szám 5-ször kisebb az előzőnél! Hajtsd végre az algoritmust!
- 5°. Készíts matematikai modellt, és állíts össze projektet  $n$  szám összegének kiszámítására, amelyek közül az első  $x$  és minden szám 2-szer nagyobb az előzőnél! Írd be a füzetedbe azokat a bemeneti adatokat, amelyekkel tesztelted ezt a projektet! MentSD a projektet a mappádban létrehozott **feladat 6.10.5** mappába!
-  6°. Készíts matematikai modellt, és állíts össze projektet  $n$  szám összegének kiszámítására, amelyek közül az első  $x$  és minden szám az előző négyzete! Írd be a füzetedbe azokat a bemeneti adatokat, amelyekkel tesztelted ezt a projektet! MentSD a projektet a mappádban létrehozott **feladat 6.10.6** mappába!
- 7°. Készíts matematikai modellt, és hozz létre projektet a következő feladat megoldására! Az ügyfél bizonyos összeget helyezett el a bankban. A bank 10% kamatot fizet az előző év végén a számlán lévő összeg után. Határozd meg, mekkora összeg lesz a számlán  $n$  év múlva! Írd be a füzetedbe azokat a bemeneti adatokat, amelyekkel tesztelted ezt a projektet! MentSD a projektet a mappádban létrehozott **feladat 6.10.7** mappába!
-  8°. Készíts matematikai modellt, és hozz létre projektet a következő feladat megoldására! Az ügyfél bizonyos összeget helyezett el a bankban. A bank meghatározott kamatot fizet az előző év végén a számlán lévő összeg után. Határozd meg, mekkora összeg lesz a számlán  $n$  év múlva! Írd be a füzetedbe azokat a bemeneti adatokat, amelyekkel tesztelted ezt a projektet! MentSD a projektet a mappádban létrehozott **feladat 6.10.8** mappába!
- 9\*. A maratonon való részvételre készülő paralimpiai sportoló az edzés első napján  $s$  km távolságot tett meg. Minden következő napon napi megtett távolsága  $p\%$ -kal nőtt az előző napi távolsághoz képest. Készíts matematikai modellt, és hozz létre projektet a következők meghatározására:
- a paralimpiai sportoló által megtett távolság az edzés  $n$ -edik napjáig;
  - teljes táv az  $n$  edzésnapra;
  - mennyivel lesz hosszabb a táv az edzés  $n$ -edik napján az első nap távjától!
- MentSD a projektet a mappádban létrehozott **feladat 6.10.9** mappába!

## 6.11. Elöltesztelő ciklus

-  1. Mi a számlálós ciklus általános alakja és hogyan működik?
2. Milyen az előltesztelő ciklus folyamatábrája és hogy működik?
3. Az algoritmusokban mikor alkalmazunk számlálós ciklust, mikor előltesztelőt?

### A ciklus előfeltétele

Már tudjátok, hogyha a ciklus végrehajtása előtt nem ismert, hogy a ciklusmag hányszor ismétlődik, akkor lehetetlen számlálós ciklust használni.

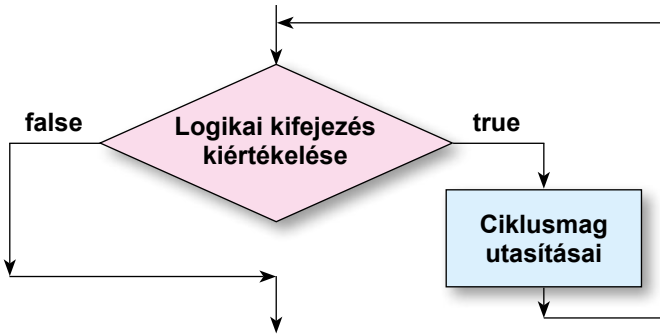
Ilyen esetekben **előltesztelő ciklus**okat alkalmaznak.





Az előltesztelő ciklus, mint az algoritmus része egy logikai kifejezés kiértékelésével kezdődik. A kiértékelés eredményétől függően, ha **true** az eredmény, a ciklusmag utasításai kerülnek végrehajtásra mindaddig, amíg a logikai kifejezés értéke **false** nem lesz, ekkor a ciklus utáni első utasítás kerül végrehajtásra.



Az előltesztelő ciklus általános folyamatábráját a 6.79. ábrán láthatjuk.



6.79. ábra. Az előltesztelő ciklus folyamatábrája

### Előltesztelő ciklusok a programozási nyelvekben

A programozási nyelvekben használt előltesztelő ciklusutasítás általános alakja a következő:

|  <b>A Pascalt használóknak</b> |  <b>A Pythont használóknak</b> |
|---|---|
| <pre>while &lt;logikai kifejezés&gt; do begin   &lt;ciklusmag utasításai&gt; end;</pre>                         | <pre>while&lt;logikai kifejezés&gt;:   &lt;ciklusmag utasításai&gt;</pre>                                       |

(ang. *while* – amíg).

Az utasítás végrehajtása a következőképpen történik: kiértékelődik a logikai kifejezés; amennyiben az eredmény **true**, végrehajtásra kerülnek a ciklusmag utasításai, ezután újra kiértékelődik a logikai kifejezés, és ha az eredmény újra **true**, újból végrehajtásra kerülnek a ciklusmag utasításai; ha a kiértékelés eredménye **false**, a **ciklusmag** nem kerül végrehajtásra, a vezérlés a ciklus utáni első utasításra kerül.

*Itt hívjuk fel a figyelmeteket a következőkre:*

- ha a ciklusmagot egyetlen utasítás alkotja, a **begin** és **end** utasítás-zárójelek alkalmazása nem kötelező;
- a ciklusmagok szekvenciákat, elágazásokat és további ciklusokat tartalmazhatnak.

Vizsgáljunk meg egy feladatot, amely megoldásának algoritmusá elöltesztelő ciklust tartalmaz.

**1. feladat.** Hány összeadandót kell összeadni ahhoz, hogy azok összege meghaladja a 100-at, ha az első összeadandó az adott valós szám és minden következő összeadandó 3-mal nagyobb az előzőnél?

Megalkotjuk a feladat matematikai modelljét:

- **bemeneti adat:** első összeadandó (**a**), az a szám, amely az előzőnél **3**-mal nagyobb, az a szám, amelynek meg kell haladnia a **100**-as összeget;

- **kimeneti eredmény:** összeadások mennyisége ( $n$ );
- **képlet:**  $a = a + 3$ ,  $s = s + a$ ,  $s \leq 100$ .

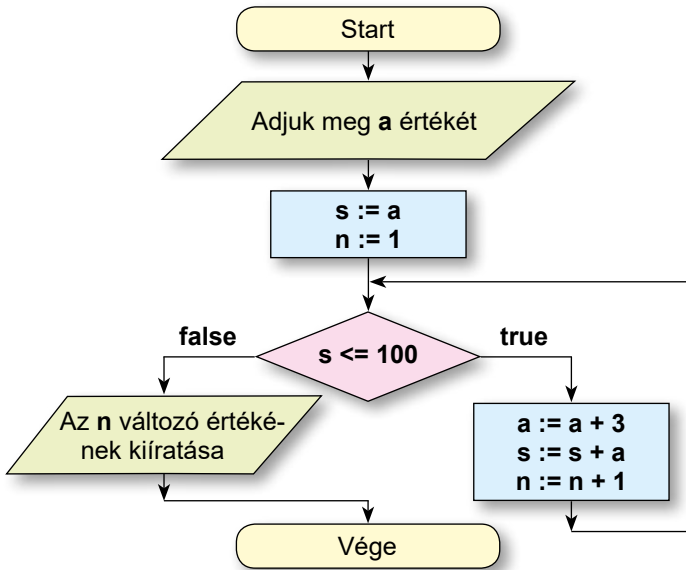
A három bemenő adat közül meg kell adni az első összeadandót  $a$ -t, a másik kettő – konkrét szám.

A feladat egy végeredménnyel rendelkezik, amit ki kell írni.

A ciklus kezdete előtt az összeg meghatározására szolgáló  $s$  változó értékét egyenlővé tesszük az első összeadandóval. Az összeadandók megszámlálására szolgáló  $n$  kezdőértéke legyen 1, mivel az összeg egyelőre az első összeadandóval egyenlő.



A számok összeadását addig kell folytatni, amíg az összeg kisebb, vagy egyenlő 100-zal. Ezért a ciklusfejben az  $s \leq 100$  logikai kifejezést helyezük el. A ciklusmagban meg kell határozni a következő összeadandót ( $a := a + 3$ ), azt hozzáadjuk az összeghez ( $s := s + a$ ) és az összeadandók számlálóját meg kell növelni 1-gyel.

A feladat megoldásának folyamatábráját a 6.80 ábra szemlélteti.



6.80. ábra. Az 1. feladat megoldásának folyamatábrája

A megfelelő projektrészlet a következő:

|  <b>A Pascalt használóknak</b>   |  <b>A Pythont használóknak</b>                   |
|---|---|
| <pre> var a, s: real; n: integer; begin   a := StrToFloat(Edit1.Text);   s := a; n := 1;   while s &lt;= 100 do   begin     a := a + 3;     s := s + a;     n := n + 1;   end;   Label1.Caption := IntToStr(n); end; </pre> | <pre> a = int(input('add meg az első összeadandót: ')) s = a n = 1 while s &lt;= 100:   a += 3   s = s + a   n += 1 print(n) </pre> |



Ha előtesztelő ciklust tartalmazó projektet tesztelünk, célszerű a bemenő adatokat úgy választani, hogy a ciklusmag egyszer sem, 1-szer, 2-szer; 3-szor, 5-ször hajtódjon végre és szükség esetén más adatkészletekkel.

Például a projekt egy részének a tesztelésére

|  |  |
|--|--|
| <pre>a := StrToFloat(Edit1.Text); s := 0; n := 0; while s &lt; a do begin   s := s + 5; n := n + 1; end;</pre> | <pre>a = float(input('Add meg az a: értékét')) s = 0 n = 0 while s &lt; a:   s += 5   n += 1</pre> |
|--|--|

a következő **a** értékeket vegyük:

- **a = -2** – a ciklusmag egyszer sem fog végrehajtódni;
- **a = 3** – a ciklusmag 1-szer hajtódik végre;
- **a = 7** – a ciklusmag 2-szer hajtódik végre és így tovább.



**Használjuk a számítógépet!**

Végezzétek el a gyakorlati feladatot a <https://cutt.ly/6hdrID2> linkről, vagy a QR-kód segítségével!

**Összefoglalás**

Az előtesztelő ciklus általános folyamatábráját a 6.79. ábrán láthatjuk.

A programozási nyelvekben használt előtesztelő ciklusutasítás általános alakja a következő:

|   |  |
|---|--|
| <b>A Pascalt használóknak</b>   | <b>A Pythont használóknak</b>  |
| <pre>while &lt;logikai kifejezés&gt; do begin   &lt;ciklusmag utasításai&gt; end;</pre> | <pre>while &lt;logikai kifejezés&gt;:   &lt;ciklusmag utasításai&gt;</pre> |

Az utasítás végrehajtása a következőképpen történik: kiértékelődik a logikai kifejezés; amennyiben az eredmény **true**, végrehajtásra kerülnek a ciklusmag utasításai, ezután újra kiértékelődik a logikai kifejezés, és ha az eredmény újra **true**, újból végrehajtásra kerülnek a ciklusmag utasításai; ha a kiértékelés eredménye **false**, a **ciklusmag** nem kerül végrehajtásra, a vezérlés a ciklus utáni első utasításra kerül.

**Felelj a kérdésekre!**

- 1°. Milyen az előtesztelő ciklus általános alakjának folyamatábrája? Ismertesd a ciklus végrehajtását!
- 2°. Hogyan hajtódnak végre az előtesztelő ciklusok algoritmusai?
- 3°. Hogyan néz ki a programozási nyelvekben használt előtesztelő ciklusutasítás általános alakja?
- 4°. Hogy történik az előtesztelő ciklus végrehajtása?
- 5°. Előfordulhat-e, hogy az előtesztelő ciklus magja egyszer sem kerül végrehajtásra? Magyarázd meg a válaszodat! Hozz fel példát!
- 6°. Előfordulhat-e, hogy az előtesztelő ciklus magja csak egyszer kerül végrehajtásra? Magyarázd meg a válaszodat! Hozz fel példát!

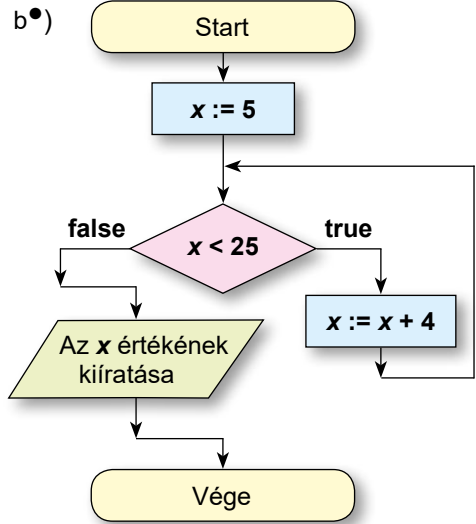
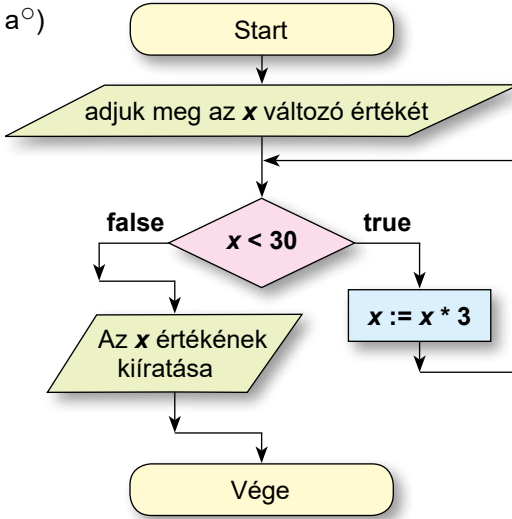
## 6. fejezet

- 7\*. Előfordulhat-e, hogy az előtesztelő ciklus a végtelenségig fut? Magyarázd meg a választodat! Hozz fel példát!
- 8\*. Milyen esetekben szükséges számlálós ciklust használni, és milyen esetekben előtesztelőt?
- 9\*. Mi a különbség a számlálós és az előtesztelő ciklusok között?
- 10\*. Miben különbözik egymástól a következő algoritmikus szerkezetek végrehajtása: szekvencia, számlálós ciklus, előtesztelő ciklus?





### Végezd el a feladatokat!

1. Hajtsd végre az alábbi algoritmusokat!





- 2\*. Hajtsd végre a programrészletet, és állapítsd meg, milyenek lesznek a változók értékei a végrehajtás után!

|    |  <b>A Pascalt használóknak</b> |  <b>A Pythont használóknak</b> |
|----|---|---|
| a) | <pre>s := 100; while s &gt; 30 do s := s - 20;</pre>  | <pre>s = 100 while s &gt; 30: s = s - 20</pre>  |
| b) | <pre>k := 1; a := 12; while a &lt; 100 do begin a := 2 * a - 4; k := k + 1; end;</pre>                            | <pre>k = 1 a = 12 while a &lt; 100: a = 2 * a - 4 k += 1</pre>  |
| c) | <pre>s := 0; a := 1; k := 1; while a &lt; 50 do begin s := s + a; k := k + 1; a := k * k; end;</pre>              | <pre>s = 0 a = 1 k = 1 while a &lt; 50: s = s + a k += 1 a = k * k</pre>  |



3\*. Hajtsd végre a programrészletet, és állapítsd meg, milyenek lesznek a változók értékei a végrehajtás után!

|    |  <b>A Pascalt használóknak</b> |  <b>A Pythont használóknak</b> |
|----|---|---|
| a) | <pre>k := 1; a := 100; while a &gt; 10 do begin   a := a / 2;   k := k + 1; end;</pre>                          | <pre>k = 1 a = 100 while a &gt; 10:   a = a / 2   k += 1</pre>  |
| b) | <pre>s := 0; a := 5; while a &lt;= 100 do begin   s := s + a;   a := a * 2; end;</pre>                          | <pre>s = 0 a = 5 while a &lt;= 100:   s = s + a   a = a * 2</pre>   |

4\*. Készíts szöveges feladatokat az 1. feladat megoldására szolgáló algoritmusok szerint!



5\*. Készíts szöveges feladatokat az 2. feladat megoldására szolgáló algoritmusok szerint!

6\*. Készíts matematikai modellt és egy projektet annak megállapítására, hogy a munkavállaló hány órán belül teljesíti az  $x$  alkatrész elkészítésének tervét, ha az első órában  $y$ -nyit gyártott le, és minden rákövetkező órában egyel több alkatrészt gyártott le, mint az előző órában! MentSD a projektet a mappádban létrehozott **feladat 6.11.6** mappába!



7\*. Készíts matematikai modellt és egy projektet annak megállapítására, hogy hány óra alatt teszi meg a turista a 20 km-es távot, ha az első órában 6 km-t tett meg, és minden egyes rákövetkező órában 1 km-rel kevesebbet halad, mint az előzőben! MentSD a projektet a mappádban létrehozott **feladat 6.11.7** mappába!

8\*. Hozz létre projektet, amely meghatározza azoknak a pozitív számoknak az átlagát, amelyek közül az első 100, és minden következő egy adott számmal kevesebb az előzőnél! MentSD a projektet a mappádban létrehozott **feladat 6.11.8** mappába!

9\*. Készíts matematikai modellt, és hozz létre projektet a következő feladat megoldására: a betétesnek hány évig kell letétbe helyeznie  $x$  hrivnyát a banknak, hogy több mint  $a$  hrivnyányi nyereséget kapjon a banktól, ha a bank évente  $p\%$ -kal növeli a betéti kamatot! MentSD a projektet a mappádban létrehozott **feladat 6.7.9** mappába!

10\*. Hozz létre egy projektet két természetes szám legnagyobb közös osztójának megtalálásához az euklideszi algoritmus szerint: ameddig a számok különbözőek, a nagyobbikat cseréljük le a nagyobbik és a kisebbik szám különbségére; amikor a számok egyenlők lesznek, akkor megkapjuk a szám  $L_{nko}$ -t. MentSD a projektet a mappádban létrehozott **feladat 6.7.10** mappába!

11\*. A maratonton való részvételre készülő sportoló az edzés első napján  $s$  km távolságot tett meg. Minden következő napon a megtett napi távolsága  $p\%$ -kal nőtt az előző naphoz képest. Készíts matematikai modellt, és hozz létre projektet a következők kiderítésére:

a) melyik edzésnapon hajtotta végre a sportoló az  $s_1$  km távot;

b) melyik edzésnapon nőtt a távolság  $s_2$  km-rel az előző naphoz képest!

MentSD a projektet a mappádban létrehozott **feladat 6.7.11** mappába!



**12\***. Alkossatok 2-3 fős csoportokat egy közös kutatási projekt kidolgozásához! Ismerkedjétek meg a számrendszerek fogalmával (például a [https:// nrs.rozh-2sch.org.ua/](https://nrs.rozh-2sch.org.ua/) weboldalon), keressetek információkat a számítástechnikában való szerepükről! Készítsetek projektet egy programozási nyelven, amely a számrendszerek között váltja át a számokat! Mutassátok be a projektet az osztályban!

### 12. SZÁMÚ GYAKORLATI MUNKA

#### Ciklusokat tartalmazó projektek

**Figyelem!** *A számítógép használata során tartsd be a balesetvédelmi szabályokat és az egészségügyi előírásokat!*

1. Készíts matematikai modellt, és hozz létre projektet a következő feladat megoldására! *Az áradás előtt a folyó vízszintje  $N$  cm volt. Az áradás során a vízszint óránként  $P$  cm-rel emelkedett. Milyen lesz a vízszint  $N$  órával az áradás után? Az áradás kezdete után hány órának kell eltelnie ahhoz, hogy a vízszint ne legyen kevesebb, mint  $K$  cm?*
2. Használj beviteli mezőket a projektben, egy mezőt az első kérdésre adott válasz megadásához és két gombot! Minden mező mellé helyezz el címkéket a mezők magyarázatához!
3. Állítsd össze az első gomb **Click** eseménykezelőjét, amely bekövetkeztekor kiíródik egy mezőbe az első kérdésre adott válasz!
4. Állítsd össze a második gomb **Click** eseménykezelőjét, amely bekövetkeztekor második kérdés kiszámításra majd pedig kiíratásra kerül egy üzenet ablakban!
5. Mentsd a projektet a mappádban létrehozott **Gyakorlati\_12** mappába!





## A

**Adatok feldolgozása (Опрацювання даних)** – az a folyamat, amelynek során új adatokat hozunk létre a meglévők alapján, 6 old.

**Alprogram (Підпрограма)** – a projekt külön része, amelynek saját neve van, és ezen a néven futtatható, 163. old.

**Az  $x$  logikai kifejezés tagadása (Заперечення логічного виразу  $x$ )** – olyan logikai művelet, amelynek eredménye **true**, ha az  $x$  logikai kifejezés értéke **false**, illetve **false**, ha az  $x$  logikai kifejezés értéke **true**, 210. old.

## B

**Bájt (Байт)** – nyolc bitből álló bitsorozat, 16. old.

**Belső hivatkozás (Внутрішнє гіперпосилання)** – egy dokumentumon belüli objektumokra mutató hivatkozás, 88. old.

**Belső memória (Внутрішня пам'ять)** – biztosítja a számítógép működését. A processzorhoz hasonlóan az alaplapon helyezkedik el. Alkotóelemei: állandó memória, operatív memória, cache memória, 23. old.

**Billentyűzet (Клавіатура)** – szimbólumok és utasítások bevitelére szolgáló eszköz, 30. old.

**Bináris kód (Двійковий код)** – a bináris kódolás után keletkezett kód, 15. old.

**Bináris kód hossza (Довжина двійкового коду повідомлення (обсяг даних))** – az üzenet byte-okban kifejezett értéke, 16. old.

**Bináris kód hosszának többszörös mértékegységei (Кратні одиниці вимірювання довжини двійкового коду):**

$1 \text{ Kbájt (kilobájt)} = 2^{10} \text{ bájt} = 1024 \text{ bájt};$

$1 \text{ Mbájt (megabájt)} = 2^{10} \text{ kbájt} = 2^{20} \text{ bájt} = 1\,084\,576 \text{ bájt};$

$1 \text{ Gbájt (gigabájt)} = 2^{10} \text{ Mbájt} = 2^{20} \text{ kbájt} = 2^{30} \text{ bájt};$

$1 \text{ Tbájt (terabájt)} = 2^{10} \text{ Gbájt} = 2^{20} \text{ Mbájt} = 2^{30} \text{ kbájt} = 2^{40} \text{ bájt}.$

**Bináris kódolás (Двійкове кодування)** – amikor az üzenetet két jel segítségével kódoljuk, 15. old.

**Bit (Біт)** – a 0 és az 1 bináris kódolásban, 15. old.

## C

**Compiler (Компілятор)** – ami a **Pascal** nyelven írt programot lefordítja a számítógép számára értelmezhető utasításokra, 153. old.

## D

**Dekódolás (Декодвання повідомлення)** – az a folyamat, amelynek elvégzése után a kódból visszafejtjük az eredeti üzenetet, 7. old.

**Digitalizáló tábla (Графічний планшет)** – grafikai információk bevitelére alkalmas eszköz. A tablett munkafelületének jellemzői kerülnek alkalmazásra. A felhasználó egy speciális tollal hozza létre az ábrát., 29. old.

**Diszjunkció (Диз'юнкція)** – az  $x$  és  $y$  logikai kifejezéseken végzett olyan művelet, amelynek eredménye akkor **true**, ha  $x$  és  $y$  logikai kifejezés közül legalább az egyik értéke **true** és akkor **false**, ha  $x$  és  $y$  mindegyike **false**, 111. old.

**Dokumentum szerkezete (Структура документа)** – a dokumentum részeiből álló hierarchikus szerkezet, 82. old.

**Dokumentumok stilisztikai tervezésének állami szabványa (Державні стандарти оформлення документів)** – a dokumentumok felépítésükre és tartalmukra vonatkozó egységes követelmények és szabályok, amelyek állami szabvány szerint készültek, 80. old.

## E

**Eljárás (procedúra) (Процедúra)** – a program névvel ellátott részlete, amelyet erre a névre hivatkozva indíthatunk el, 161. old.

**Ergonómia (Ергономіка)** – az a tudomány, amely a különböző eszközök, készülékek, szerszámok alkalmazásának hatásfokát tanulmányozza, valamint

azt, hogyan hat ezek használata az ember tevékenységére, termelékenységére, 106. old.

**Ergonómikus weboldal (Ергономічний сайт)** – olyan weboldal, amely használata kényelmes, vezérlőelemei jól áttekinthetők és használhatók, a felhasználó jól követhetően tekintheti meg a weboldalt, 106. old.

**Eseménykezelő (Обробник подій)** – olyan projektresztlet, amely az esemény bekövetkezése után kerül végrehajtásra, 161. old.

**Erintőképernyő (Сенсорний екран)** – adatbeviteli- és kiviteli eszköz, amely az ujjal vagy a speciális tollal való érintésre érzékeny, 29. old.

## F

**Fejléc (Колонтитули)** – a dokumentum „szolgálati” jellegű (például oldalszám) információkat tartalmazó ismétlődő része, 73. old.

**Felhasználói felületet (Інтерфейс користувача)** – ide tartoznak a vezérlőelemek (gombok, menük stb.), adatbeviteli eszközök (mezők, számlálók stb.), adatkimeneti eszközök (üzenetablakok, címkék, szövegmezők stb.) és használata, 149. old.

**Flash memóriák (Флешпам'ять)** – az elektronikus chipeken lévő külső memória három fő típusát különböztetjük meg: pendrive, memóriakártya és szilárdtest, vagy SSD-meghajtó, 25 old.

**Forma (Форма)** – olyan objektum, amelyen komponenseket (vezérlőelemeket), például gombokat, mezőket, feliratokat, menüket, gördítősávokat helyezhetünk el. A végrehajtás során a formából lesz a program ablaka, amelyeken megjelennek a megfelelő vezérlők, 154. old.

**Formázási jelek (Приховані недруківані символи)** – ide azokat a jeleket soroljuk, amelyeket a felhasználó visz be a beírás során, de nyomtatásban nem jelennek meg, 64. old.

**Függvény (Функція)** – egy alprogram, amely egy vagy több értéket eredményez: számokat, szövegsorokat stb., 163. old.

## G

**Grabberek (Граббер)** – nézd a rekordert a 122. oldalon

## H

**Hangszerkesztő (Музичні редактори)** – hangadatok módosítására szolgáló program, 128. old.

**Hasábelválasztó jelek (Маркери меж)** – egyezményes jelek, melyek mozgatásával megváltoztathatjuk a hasábszélességeket és a hasábok távolságát, 72. old.

**Hibakeresés (Налáгодження проєкту)** – a projekt hibái megtalálásának és kijavításának a folyamata, 204. old.

## I

**Interpretátor (Інтерпретатор)** – olyan számítógépes program, amely a parancsokat gépi kóddá alakítja és végrehajtja ezeket az utasításokat, amely valamilyen programozási nyelven íródott, 156. old.

## K

**Kijelentés (Висловлювання)** – egy olyan mondat, ami valamely objektumra, objektumok kapcsolatára nézve olyan állítást fogalmaz meg, amelyről egyértelműen eldönthető, hogy **igaz** vagy **hamis**, 112. old.

**Klip (Кліп)** – rövid terjedelmű videó vagy mozifilm, 130. old.

**Kodek (Кóдеки)** – multimédiás adatok tömörítésére és kitömörítésére szolgáló szoftver, 122. old.

**Kódtábla (Таблиця кодів символів)** – olyan táblázat, amiben minden felhasználandó szimbólumhoz számot rendelünk, 9. old.

**Komment (Коментár)** – a compiler által figyelmen kívül hagyott szöveg, amelyet a jobb érthetőség és a hibakeresés megkönnyítése céljából helyezünk el a programban, 205. old.

**Konjunkció (Кон'юнкція)** – az az **x** és **y** logikai kifejezés, amelynek értéke akkor **true**, ha az **x** és az **y** értéke is **true**, minden más esetben pedig **false**, 210. old.



**Könyvjelző (Закладка)** – olyan helyet jelölünk vele a dokumentumban, amelyet könnyen meg szeretnénk találni. Adhatunk neki egyedi nevet, így könnyen azonosítható, 88. old.

**Külső hiperhivatkozás (Зовнішнє гіперпосилання)** – külső forrásokra utaló hivatkozás, 88. old.

**Külső memória (Зовнішня пам'ять)** – a számítógépnek az adatok hosszútávú tárolására szolgáló része. Ide tartoznak a merevlemezek, optikai lemez, flash-memória stb., 24. old.

## L

**Link (Посилання)** – a dokumentum speciális eleme, amely a dokumentum másik helyére, vagy egy másik dokumentumra mutat, 88. old.

**Listák (Списки)** – a szöveges dokumentum speciálisan formázott bekezdései, objektumok felsorolására használjuk, 70. old.

**Logikai hibák (Логічні помилки)** – hibák, melyek során a program futása folyton megszakad, vagy a futtatás nem a helyes eredményre vezet, 204. old.

**Logikai művelet (Логічний вираз)** – olyan kifejezés, amely **true** vagy **false** értéket vehet fel, 209. old.

**Logikai típusú változók (Змінні логічного типу)** – azok a változók, amelyek a **true** vagy a **false** érték egyikét vehetik fel, 209. old.

## M

**Mikrofon (Мікрофон)** – olyan eszköz, amely a hangrezgéseket elektromos jellé alakítja, 29. old.

**Monitor (Монітор)** – a személyi számítógép elsődleges kiviteli eszköze. A képalkotás módja alapján megkülönböztetünk LCD (folyadékkristály), plazma OLED (szerves fénydióda), elektronikus tinta stb. monitorokat, 36. old.

## N

**Navigációs elemek (Елементи навігації)** – lehetővé teszi, hogy átléphessünk a webhely más lapjaira vagy más weblapokra, 105. old.

**Nyomatató (Прінтер)** – kiviteli eszköz, amely papíron jeleníti meg a különböző adatokat, jellemzően A3-at meg nem haladó méret, 36. old.

## P

**Plotter (rajzgép) (Плότηр)** – nagy-méretű (A2 vagy nagyobb) grafikai objektumok kiviteli szolgáló eszköz, 36. old.

**Podcast (Подкаст)** – az interneten terjeszthető multimédiás fájl, általában a szerzők véleménynyilvánítását tartalmazza politikai, gazdasági, műszaki, irodalmi, zenei kérdésekben, 145. old.

**Podcaster (Подкастер)** – podcastokat létrehozó személy, 145. old.

**Prioritási sorrend (Пріоритет виконання логічних операцій)** – először a tagadást hajtjuk végre, majd konjunkciót, végül a diszjunkciót, 211. old.

**Processzor (Процесор)** – ez az eszköz biztosítja a programok futását, végzi az adatok feldolgozását a számítógépben. A processzor, mint minden algoritmusvégrehajtó, saját utasításkészlettel rendelkezik, 22. old.

**Programozási nyelv ábécéje (Алфавіт мови програмування)** – azok a szimbólumok, amelyeket a nyelv szavainak és mondatainak lejegyzésére használunk, 149. old.

**Programozási nyelv (Мова програмування)** – az a nyelv, amelynek rendeltetése, hogy a számítógépnek értelmezhető algoritmusokat fogalmazzunk meg, 149. old.

**Projekt tesztelése (Тестування проєкту)** – logikai hibák keresésének folyamata tesztadatok alkalmazásával, 204. old.

## R

**Rekorder (Рекордер)** – hang- és videó rögzítésére szolgáló eszköz, 122. old.

## S

**Stílus (Стиль)** – bizonyos objektum tulajdonságainak névvel ellátott összessége, 78. old.

**Számítógépes program (Комп'ютерна програма)** – egy adatfeldolgozó algoritmus, ami a számítógép számára értelmezhető nyelven van leírva, 148. old.

**Szemantika a programozási nyelvben (Семантика в мові програмування)** – a nyelvi szerkezetek értelmezése, alkalmazásának szabályai, 149. old.

**Színmodell (Колірна модель)** – számszerűsített jellemzők, amelyek a spektrum színeinek előállítására szolgálnak, 12. old.

**Szintakszis a programozási nyelvben (Синтаксис мови програмування)** – a nyelvi szerkezetek megalkotásának szabályai, 149. old.

**Szintaktikai hibák (Синтаксичні помилки)** – hibák, melyek értelmezhetetlenné teszik az algoritmust a számítógép számára, ide tartozik a meg nem engedett szimbólumok használata, a szótár szavainak nem pontos leírása, a szintaktikai szabályok megsértése, 149. old.

**Szkenner (Сканер)** – grafikus beviteli eszköz. Működése szerint megkülönböztetünk kéziszkennert, lapszkennert és szkennerkamerát, 32. old.

**Szótár a programozási nyelvben (Словник мови програмування)** – a nyelvben használható szavak jegyzéke, 149. old.

**Szöveges dokumentum (Текстовий документ)** – a szövegfeldolgozó rendszerek feldolgozásának fő tárgya, amely különböző objektumokat tartalmazhat: karaktereket, szavakat, mondatokat, bekezdéseket, oldalakat, ábrákat, táblázatok és diagramokat, 62. old.

**Szövegfeldolgozó rendszerek (Системи опрацювання текстів)** – szöveg létrehozására és feldolgozására tervezett programok, 61. old.

**Szublímációs nyomtatás (Сублімаційний друк)** – a szublímáció az anyag szilárd állapotból gázállapotba történő átmenetének fizikai jelensége, megkerülve a folyékony állapotot. A szublímációs nyomtatásnak számos lehetősége van: közvetlen „nedves” nyomtatás, közvetlen

„száraz” nyomtatás, közvetett nyomtatás, 39. old.

## T

**Tabulátorok (Табуляція)** – a szövegszerkesztő azon eszköze, amelynek segítségével szöveges objektumokat pozícionálhatunk, 74. old.

**TAG (Тэги)** – a html dokumentum formázó utasításai, amelyek meghatározzák a szöveg kinézetét: felépítését, formátumát és a szövegrészek elhelyezkedését, 107. old.

**Tartalom (Контент)** – a weboldal tartalmi része, erre koncentrálódik a felhasználó figyelme; szöveget, grafikát, videót, hangfájlokat tartalmazhat, 105. old.

**Tartalomjegyzék (Зміст документа)** – a dokumentum szerkezeti elemeinek listája az oldalszámok feltüntetésével, 87. old.

**Tartalomvezérlő rendszerek WCMS-ek (Системи управління вебконтентом, WCMS)** – ezen programok kényelmes eszközöket biztosítanak a felhasználók számára, amelyekkel kidolgozhatják a weblap külalakját, feltölthetik azt tartalommal, kialakíthatják a navigációs rendszereket, 111. old.

**Téma (Тема документа)** – a dokumentum egészére alkalmazott stílus. A témáknak egyedi neve van, és stíluselőírásokat tartalmaznak a dokumentumban alkalmazott objektumokra (szimbólumok színe, mérete, betűtípusa, bekezdések igazítása, sorköz, ábrák formázása stb.), 79. old.

**Tesztadatok (Тестовий набір входних даних)** – olyan adatok, melyek segítségével meggyőződhetünk a projekt helyességéről, és amelyekhez előre ismerjük az elvárt eredményeket, 204. old.

**Többszintű lista (Багаторівневий список)** – amikor a listák egyes elemei alá új listák vannak rendelve, 70. old.

**Touchpad (Тачпад)** – notebookok, netbookok számítógép vezérlésére szolgáló adatbeviteli eszközei, 29. old.

**TV-tuner (ТВ-тюнер)** – olyan eszköz, amelynek segítségével televíziós műso-



rokat tekinthetünk meg a számítógépen, 29. old.

## U

**UNICODE** (*universal coded character set*) (**Юнікод**) – karakterkódolási rendszer, amely 17, egyenként 65 536 elemű kódtáblát tartalmaz, ezáltal 1 114 112 karakter kódolását teszi lehetővé, 10. old.

**Utasítás zárójelek** (**Операторні дужки**) – a **begin** és az **end** szavak, 162. old.

**Üzenetek kódolása** (**Кодування повідомлення**) – az üzenetet reprezentáló egyik szimbólumsorozat cseréje egy másik szimbólumsorozatra, 7. old.

## V

**Vetített billentyűzet** (**Проекційна клавіатура**) – hatékony adatbeviteli eszköz, melynek képe bármilyen síkfelületre vetíthető, 30. old.

**Videokártya** (**Відеоадаптер**) – speciális eszköz, amely feldolgozza a grafikai adatokat a képernyőn való megjelenítés előtt, 37. old.

**Videostudiók** (**Відеостудії**) – professzionális videószerkesztésre használt program, 128. old.

**Videoszerkesztők** (**Відеоредактори**) – videóadatok szerkesztésére szolgáló program, 128. old.

## W

**Webhelytérkép** (**Мапа сайту**) – egy diagram, amely megmutatja az oldalak hierarchiáját és a közöttük lévő kapcsolatokat, 113. old.

**Weboldal dizájn** (**tervezés**) (**Дизайн вебсторінки**) – a weboldal kialakítását, látványát, tartalmának strukturáltságát biztosítja, 105. old.

## Z

**Zenei stúdió** (**Музичні студії**) – professzionális hangszerkesztő szoftver, 128. old.

**3D nyomtató** (**Тривимірні, або 3D-прінтери**) – digitális modellekből háromdimenziós tárgyak alkotására képes

eszköz. A 3D-s nyomtatókat azon anyagok típusa szerint különböztetik meg, amikből „felépül” az adott tárgy, 40. old.

**3D tollak** (**3D-ручки**) – új adatkiméleti eszközök a térben történő „rajzolásához”. képeket készítenek egy olvadó műanyag vékony (0,5-1,5 mm) sugarának kinyomásával, amely azonnal megkeményedik és háromdimenziós tárgyat képez, 42. old.

**ASCII** – az angol nyelv betűit, számjegyeket, elválasztójeleket tartalmazó, összesen 128 karakter kódolását (0-tól 128-ig) lehetővé tevő karaktertábla, 9. old.

**CMYK** – színmodell, amely négy alapszintet tartalmaz: **türkizkék** (**Cyan**), a **bíbor** (**Magenta**), a **sárga** (**Yellow**) és a **fekete** (**Black**), 12. old.

**DOCX** – szöveget, képeket, beillesztett objektumokat, tulajdonságaik értékeit tároló szöveges dokumentum, melyet a **Microsoft Word** szövegszerkesztő használ, 62. old.

**FLAC** – szabad felhasználású, veszteségmentes tömörítést alkalmazó audio kodek, 121. old.

**HSB** – színmodell, melyben minden szint három számmal: árnyalat (**Hue**), **telítettség** (**Saturation**) és **fényerő** (**Brightness**) jellemeznek, 12. old.

**HTML** – szöveges dokumentumok weblap formátumban történő mentése, 107. old.

**IDLE** – projektfejlesztési környezet a **Python** programozási nyelvben, 155. old.

**MIDI** – speciális eszköz vagy program (szintetizátor) általi lejátszásra alkalmas fájlformátum, 121. old.

**MP3** – a legelterjedtebb hangfájl-típus. Fájlcsereelőkből, interneten alkalmazzák hangadatok továbbítására, 121. old.

**MP4** – videofilmek tárolására, valamint digitális televíziózásban használt fájl-típus, 212. old.

**ODT** – az **Open Writer** szövegszerkesztő által létrehozott szövegfájlok szabványos formátuma, 63. old.

**PDF** – nem szerkeszthető, megtekintésére **Adobe Reader**t vagy a böngészőket

használják. A fájl a formázott szöveget és az összes beillesztett objektumot tárolja, 63. old.

**RGB** – színmodell, melynek alapszínei (a piros (*Red*), a zöld (*Green*) és a kék (*Blue*), 12. old.

**RTF** – a fájl szövegét, képeket, beillesztett objektumokat és azok tulajdonságainak értékeit tárolja. Különböző operációs rendszerekben működő szövegszerkesztők által támogatott, 63. old.

**QuickTime** (gyors idő) – az **Apple** által kidolgozott, hang- és videoanyagok tárolására alkalmas fájlformátum, 121. old.

**SmartArt** – **Microsoft Office** grafikus objektumai, amelyeket arra terveztek, hogy strukturált adatokat jelenítsenek meg egy szöveges dokumentumban különböző sémák formájában, 98. old.

**TXT** – a szöveges dokumentumok egy típusa, **txt** fájlban csak formázatlan,

bekezdésekre tördelt szöveg tárolható, 62. old.

**WAV** vagy **WAVE** – a **Windows** operációs rendszerek alapértelmezett hangformátuma, 121. old.

**WMA** – a **Windows** operációs rendszerben szabványos, hang- és videofájlok tárolására szolgáló fájlformátum, 121. old.

**WMV** – a **Windows** operációs rendszerben szabványos, hang- és videofájlok tárolására szolgáló fájlformátum, 121. old.

**YouTube** (a te csatornád) – videomegosztó szolgáltatás az interneten, 140. old.

**WYSIWYG** – olyan technológia, amely lehetővé teszi, hogy a dokumentumot a képernyő ugyanabban a formában jelenítse meg, ahogyan kinyomtatják, 62. old.



# TARTALOM

|                                    |   |
|------------------------------------|---|
| <i>Kedves nyolcadikosok!</i> ..... | 3 |
|------------------------------------|---|

## 1. fejezet

### ADATOK KÓDOLÁSA

|   |    |
|---|----|
| 1.1. Az adatfeldolgozás, mint információs folyamat.<br>Üzenetek kódolása és dekódolása. ....          | 5  |
| 1.2. Szövegek és képek kódolása. ....   | 9  |
| 1.3. Bináris kódolás .....  | 14 |
| <i>1. számú gyakorlati munka.</i> Feladatok megoldása szöveges kód<br>hosszának meghatározására ..... | 20 |

## 2. fejezet

### A SZÁMÍTÓGÉP HARDVERE

|   |    |
|---|----|
| 2.1. A személyi számítógép felépítése .....   | 21 |
| 2.2. Adatbeviteli eszközök .....  | 29 |
| 2.3. Kiviteli eszközök .....  | 36 |
| 2.4. A számítástechnikai és számítógépes eszközök története .....                         | 44 |
| 2.5. A modern számítógépek típusai és alkalmazása .....                                   | 52 |
| <i>2. számú gyakorlati munka.</i> Számítógép-konfiguráció kialakítása igény szerint. .... | 60 |

## 3. fejezet

### SZÖVEGFELDOLGOZÁS

|  |    |
|--|----|
| 3.1. Szövegfeldolgozó rendszerek. Szövegtörédek keresése és cseréje .....  | 61 |
| 3.2. Többszintű listák, szakaszok, hasábok, fej- és láblécek beillesztése<br>a szövegbe. ....                              | 69 |
| <i>3. számú gyakorlati munka.</i> Különböző objektumokat tartalmazó<br>szöveges dokumentum létrehozása .....               | 77 |
| 3.3. Stílusok alkalmazása a dokumentumban.<br>A dokumentum szerkezetének kialakítása .....                                 | 77 |
| 3.4. Tartalomjegyzék létrehozása. Hivatkozások.<br>Dokumentumokkal való csapatmunka .....                                  | 86 |
| <i>4. számú gyakorlati munka.</i> A dokumentum felépítése.<br>Automatikus tartalomjegyzék készítése. Hiperhivatkozás. .... | 94 |
| 3.5. Speciális grafikai objektumok létrehozása a szövegszerkesztőkben .....  | 95 |

## 4. fejezet

### WEBES FELÜLETEK LÉTREHOZÁSA ÉS KÖZZÉTÉTELE

|   |     |
|---|-----|
| 4.1. A weboldalak felépítése. A HTML nyelv .....  | 105 |
| 4.2. A webfejlesztés automatizálásának eszközei. ....   | 110 |
| <i>5. számú gyakorlati munka.</i> Weboldalak készítése online webhelytervező<br>rendszer segítségével ..... | 117 |

## 5. fejezet

### MULTIMÉDIÁS OBJEKTUMOK FELDOLGOZÁSA

|   |     |
|---|-----|
| 5.1. Multimédiás objektumok feldolgozása . . . . .                                  | 118 |
| 5.2. Audiók és videók szerkesztése . . . . .  | 128 |
| 6. számú gyakorlati munka. Videoklip készítése . . . . .                            | 139 |
| 5.3. Multimédiás fájlok közzététele az interneten . . . . .                         | 139 |
| 7. számú gyakorlati munka. Hangfájlok és filmek elhelyezése az interneten . . . . . | 147 |

## 6. fejezet

### ALGORITMUSOK ÉS PROGRAMOK

|  |     |
|--|-----|
| 6.1. Számítógépes programok és programozási nyelvek . . . . .  | 148 |
| 6.2. Események. Eseménykezelők . . . . .   | 160 |
| 8. számú gyakorlati munka. Üzenetablakot megjelenítő objektumorientált program létrehozása . . . . .     | 173 |
| 6.3. A gomb . . . . .  | 173 |
| 6.4. A címke . . . . .   | 180 |
| 9. számú gyakorlati munka. Gombokat és címkéket tartalmazó projektek létrehozása . . . . .               | 188 |
| 6.5. Beviteli mező. Projektek bemeneti adatokkal és kimeneti végeredményekkel . . . . .                  | 190 |
| 10. számú gyakorlati munka. Mezővel rendelkező projektek adatbevitellel és kimeneti eredménnyel. . . . . | 203 |
| 6.6. A projekt hibakeresése . . . . .  | 204 |
| 6.7. Logikai kifejezések. Logikai változók. Logikai műveletek. . . . .                                   | 208 |
| 6.8. Elágazások . . . . .  | 214 |
| 6.9. A jelölőnégyzetek. A választógombok . . . . .   | 225 |
| 11. számú gyakorlati munka. Elágazásokat tartalmazó projektek . . . . .                                  | 235 |
| 6.10. Számlálás ciklus. . . . .  | 235 |
| 6.11. Elöltesztelő ciklus. . . . .   | 242 |
| 12. számú gyakorlati munka. Ciklusokat tartalmazó projektek . . . . .                                    | 248 |
| Szótár . . . . .   | 249 |

### A tankönyv állapota

| № | A tanuló neve | Osz-<br>tály | Tanév | A tankönyv állapota |          |
|---|---------------|--------------|-------|---------------------|----------|
|   |               |              |       | év elején           | év végén |
| 1 |               |              |       |                     |          |
| 2 |               |              |       |                     |          |
| 3 |               |              |       |                     |          |
| 4 |               |              |       |                     |          |
| 5 |               |              |       |                     |          |