

Генеза

НОВА УКРАЇНЬСЬКА ШКОЛА

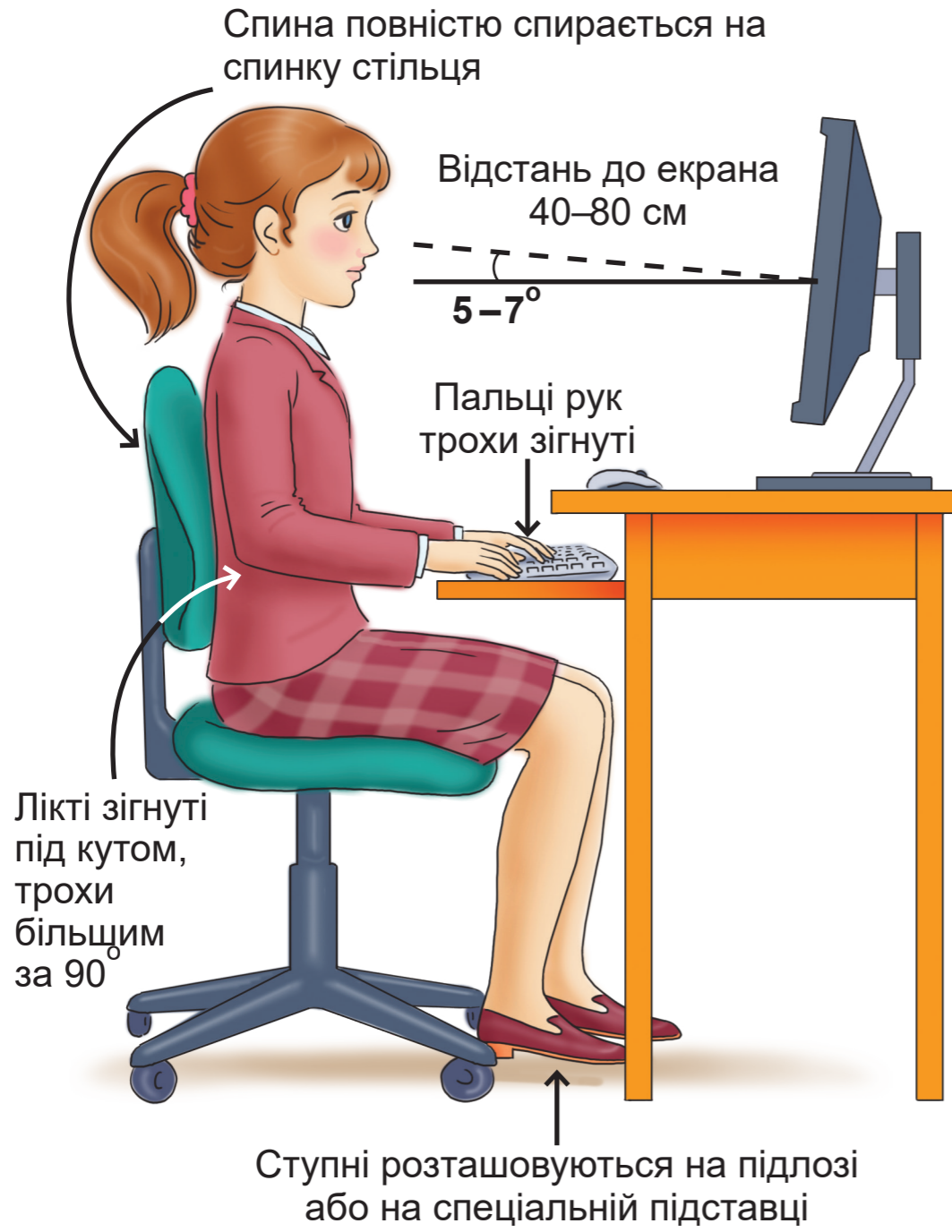
ІНФОРМАТИКА

ЧАСТИНА 1

7



ДОТРИМУЙСЯ ПРАВИЛЬНОЇ ПОСТАВИ ПІД ЧАС РОБОТИ З КОМП'ЮТЕРОМ



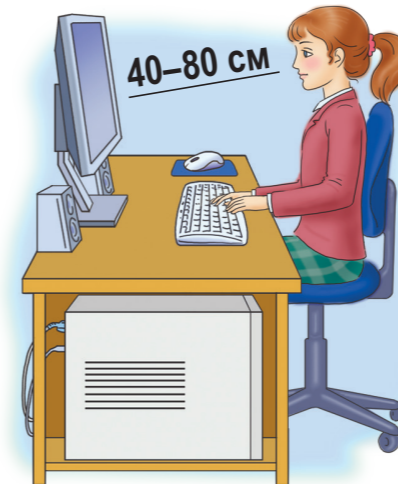
ПРАВИЛА БЕЗПЕЧНОЇ РОБОТИ З КОМП'ЮТЕРОМ



Готуючись до роботи з комп'ютером, наведи лад на столі



Тримай комп'ютер у чистоті. Протирай його спеціальною серветкою



Відстань від очей до екрана 40–80 см

40–80 см



Робоче місце повинно бути зручним



Не торкайся задньої панелі монітора й системного блока



Після 15 хвилин роботи з комп'ютером зроби перерву, розімнися

Відомості про користування підручником

№ з/п	Прізвище та ім'я учня/учениці	Клас	Навчаль- ний рік	Оцінка	
				на по- чатку року	в кінці року
1					
2					
3					
4					
5					

Навчальне видання

**РИВКІНД Йосиф Якович, ЛИСЕНКО Тетяна Іванівна,
ЧЕРНІКОВА Людмила Антонівна, ШАКОТЬКО Віктор Васильович**

ІНФОРМАТИКА

Підручник для осіб з особливими освітніми потребами (Н 54.1–Н 54.2)
7 клас (у 2 частинах) Частина 1

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»

У підручнику використано ілюстративний матеріал з відкритих джерел Інтернету,
зокрема сайтів *vecteezy.com*, *depositphotos.com*. Усі матеріали в підручнику
використано з навчальною метою відповідно до законодавства України
про авторське право і суміжні права.

Редактор *Наталія Дашко*. Обкладинка *Олени Мамаєвої*. Макет, художнє
оформлення, комп'ютерна обробка ілюстрацій *Василя Марущинця*.
Комп'ютерна верстка *Юрія Лебедєва*. Коректор *Олена Симонова*

Формат 84×108/16. Ум. друк. арк. 19,32. Обл.-вид. арк. 9,75.
Тираж 1 666 пр. Вид. № 0045. Зам. № 24-07-0406.

ТОВ «Гене́за», вул. Генерала Алмазова, б. 18/7 (літ. В), офіс 404, м. Київ, 01133, Україна.
Свідоцтво суб'єкта видавничої справи серія ДК № 7692 від 24.10.2022.

Віддруковано у ТОВ «ПЕТ», вул. Максиміліанівська, 17, м. Харків, 61024, Україна.
Свідоцтво суб'єкта видавничої справи серія ДК № 6847 від 19.07.2019.

ІНФОРМАТИКА В УКРАЇНІ

В Україні в середині 50-х років ХХ століття була створена мова програмування, яку назвали «Адресна мова» (1955 р.). Це була одна з перших мов програмування у світі. Саме завдяки цій мові зникла залежність від місця розташування програми в пам'яті комп'ютера. Значний вклад в її розробку зробила Катерина Логвинівна Ющенко – керівниця групи програмісток і обчислювачок МЕОМ, засновниця першої в країні школи теоретичного програмування. Вона також стала першою в країні жінкою – доктором фізико-математичних наук за досягнення у програмуванні.



*Катерина Логвинівна
Ющенко*



Фото групи програмісток та обчислювачок (1956 р.)



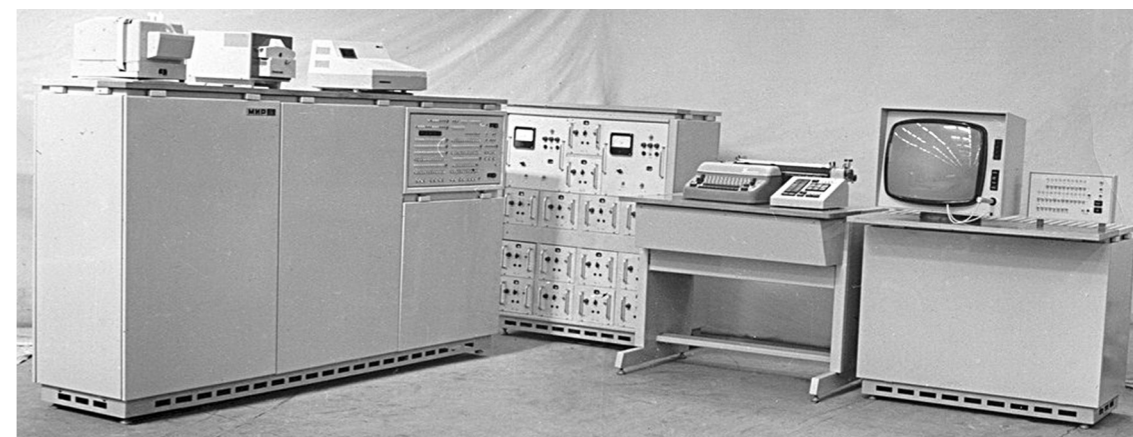
ЕОМ «MIP»

Перша машина в сімействі «MIP» (Машина для інженерних розрахунків) була створена в Інституті кібернетики (м. Київ) в 1965 році. Це одна з перших ЕОМ, які надалі отримали назву «персональні», тобто призначені для одночасного використання однією особою. Використовувалась вона у навчальних закладах, інженерних бюро, наукових організаціях. Мала низку унікальних особливостей, таких як оригінальна мова програмування «АЛМІР-65», що була близька за можливостями до мов програмування високого рівня, модульна архітектура, виведення даних на друкарську машинку тощо.

В ЕОМ «MIP-2» вперше в Україні було реалізовано діалоговий режим роботи. У цьому комп'ютері, крім стандартних на той час засобів введення даних, використовувався новий пристрій – дисплей зі світловим пером. Новою була й спеціальна мова програмування «АНАЛІТИК».



*ЕОМ
«MIP-2»*



УДК 004(075.3.056.262)
І-74

Авторський колектив:
Йосиф Ривкінд, Тетяна Лисенко, Людмила Чернікова, Віктор Шакотько

Рекомендовано Міністерством освіти і науки України
(наказ Міністерства освіти і науки України від 05.02.2024 № 124)

Видано за рахунок державних коштів. Продаж заборонено

Відповідно до модельної навчальної програми
«Інформатика. 7–9 класи» для закладів загальної середньої освіти
(автори: Ривкінд Й.Я., Лисенко Т.І., Чернікова Л.А., Шакотько В.В.)

Інформатика : підруч. для осіб з особливими
І-74 освіт. потребами (Н 54.1–Н 54.2) : 7-й кл. (У 2 ч.).
Ч. 1 / [Й. Ривкінд та ін.]. — Київ : Генеза, 2024. —
184 с. : іл.

ISBN 978-617-8353-51-3

ISBN 978-617-8353-52-0 (ч. 1)

УДК 004(075.3.056.262)

ISBN 978-617-8353-51-3
ISBN 978-617-8353-52-0 (ч. 1)

© Ривкінд Й.Я., Лисенко Т.І.,
Чернікова Л.А., Шакотько В.В., 2024
© «Генеза», оригінал-макет, 2024

Дорогі семикласниці та семикласники! Шановні вчительки та вчителі!

Учні та учениці сьомого класу продовжують вивчення найсучаснішого шкільного предмета – **інформатики**.

Інформатика – це наука про інформаційні ресурси та інформаційні процеси, про комп'ютери та їх використання. Чому потрібно вивчати інформатику? Тому, що сучасний світ – це світ, у якому інформаційно-комунікаційні технології є одним з головних інструментів для досягнення успіху, а вміння доцільно й ефективно їх використовувати у своїй навчальній діяльності, у повсякденному житті – одна з основних вимог до учня та учениці Нової української школи.

Підручник, який підготував наш авторський колектив, сприятиме успішному навчанню інформатики. Він допоможе навчитися логічно мислити, обґрунтовувати власні дії та свій вибір цифрових засобів для розв'язування задач, створення моделей об'єктів, явищ і процесів з використанням інформаційних технологій. А також діяти творчо, генерувати власні ідеї та добросовісно використовувати чужі для створення нових інформаційних об'єктів.

На уроках інформатики у 7-му класі ви дізнаєтеся, що таке електронна пошта та хмарні сервіси, які вони є і як їх можна використовувати для організації спільної роботи. Навчитеся створювати проекти однією з сучасних мов програмування – **Python** або **Object Pascal**. Продовжите створювати та опрацьовувати комп'ютерні презентації. Будете розробляти аудіо- та відеопроєкти, а також створювати анімовані графічні зображення. Набуті компетентності ви застосовуватимете під час виконання навчальних проєктів на уроках інформатики та інших предметів.

Матеріал підручника поділено на шість розділів. Розділи складаються з пунктів, а кожний пункт – з підпунктів. Корисними та цікавими є рубрики:



Пригадайте



Працюємо з комп'ютером



Інтеграція (поєднання) наук



Поміркуйте



Практична робота



Працюємо у групах/ в парах



Обговоріть і зробіть висновки



Найважливіше в цьому пункті



Дослідіть



Для тих, хто хоче знати більше



Дайте відповіді на запитання



Готуємось до вивчення нового матеріалу



Цікаві факти з історії



Виконайте завдання



Оцініть свої знання та вміння

Значна увага в підручнику приділяється практичній діяльності, у процесі якої й формуються відповідні інформатичні компетентності. Учням та ученицям буде запропоновано обдумати та обговорити в парах або в невеликих групах відповіді на деякі запитання, виконати завдання в зошиті або з використанням комп'ютера для створення чи вдосконалення певного інформаційного продукту.

Детальніше зрозуміти, наскільки сподівання учнів та учениць справилися, яких результатів досягнуто, вони зможуть, відповівши на запитання рубрики **Оцініть свої знання та вміння**, яку розміщено в кінці кожного розділу підручника.

Матеріали для виконання завдань, вправ, у тому числі файли-заготовки, розміщено на сайті **Інформатика для всіх** за адресою <http://sites.google.com/ru.org.ua/allinf> або за QR-кодом.



Бажаємо плідного та захопливого навчання!

Авторський колектив

Розділ 1

ПОШУК В ІНТЕРНЕТІ. ЕЛЕКТРОННА ПОШТА. ХМАРНІ СЕРВІСИ

У цьому розділі ви отримаєте нові, а також поглибите та розширите свої знання й удосконалили навички з таких тем:

- ▶ пошук відомостей в Інтернеті;
- ▶ розширений пошук;
- ▶ перевірка достовірності зібраних даних;
- ▶ розпізнавання та протидія маніпуляціям і пропаганді;
- ▶ електронне листування;
- ▶ хмарні сервіси;
- ▶ створення та використання документів зі спільним доступом;
- ▶ синхронізація даних.



1.1. Пошук відомостей в Інтернеті. Розширений пошук. Створення списків джерел і закладок

У цьому пункті йтиметься про:

- ▶ інструменти пошуку відомостей в Інтернеті;
- ▶ розширений пошук;
- ▶ формування списку джерел;
- ▶ створення та використання закладок на знайдені ресурси Інтернету.

ІНСТРУМЕНТИ ПОШУКУ ВІДОМОСТЕЙ В ІНТЕРНЕТІ



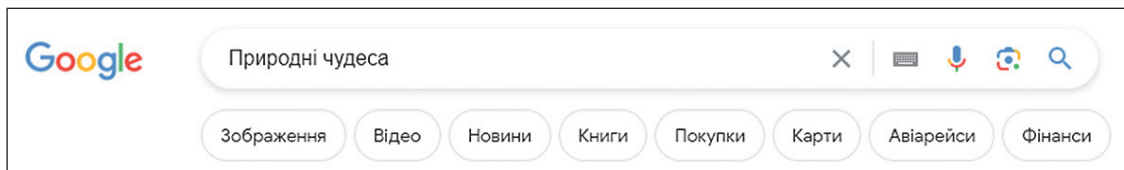
Пригадайте

- Які сервіси використовують для пошуку відомостей в Інтернеті? Наведіть приклади таких сервісів і алгоритм пошуку.
- Чим відрізняється пошук в Інтернеті текстових повідомлень і зображень?
- Які права на використання зображень, знайдених в Інтернеті, дає ліцензія **Creative Commons**?

Вам вже неодноразово доводилось шукати в Інтернеті текстові відомості, зображення, аудіо- та відеофайли з різних тем, які вас цікавлять або потрібні вам для навчання чи для домашніх справ. Зазвичай під час пошуку відомостей в Інтернеті ви отримуєте мільйони посилань.

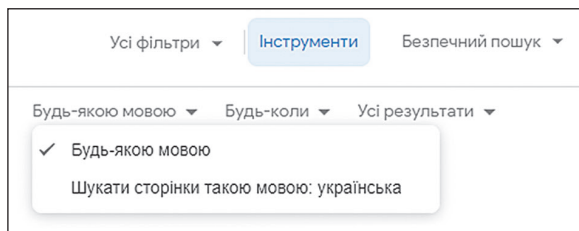
Фільтр (лат. *filtrum* – повсть) – пристрій, прилад або речовина для відокремлення потоку (рідини, газу, інформації, сигналу тощо) за його певними характеристиками від непотрібних домішок.

Для відображення в результатах пошуку посилань лише на дані певних видів (зображення, відео, карти) або певного призначення (новини, книги, покупки тощо) у пошуковій системі **Google** можна використовувати кнопки, що застосовують **фільтри** до результатів пошуку (мал. 1.1). Фільтри призначено для відбору результатів, які відповідають певній умові.



Мал. 1.1. Кнопки для фільтрування результатів пошуку в пошуковій системі **Google**

Але не всі сторінки, на які отримано посилання, точно відповідають вашому запиту та потребам. Частина з них створена іноземними мовами, інші застарілі або містять не всі введені вами ключові слова.



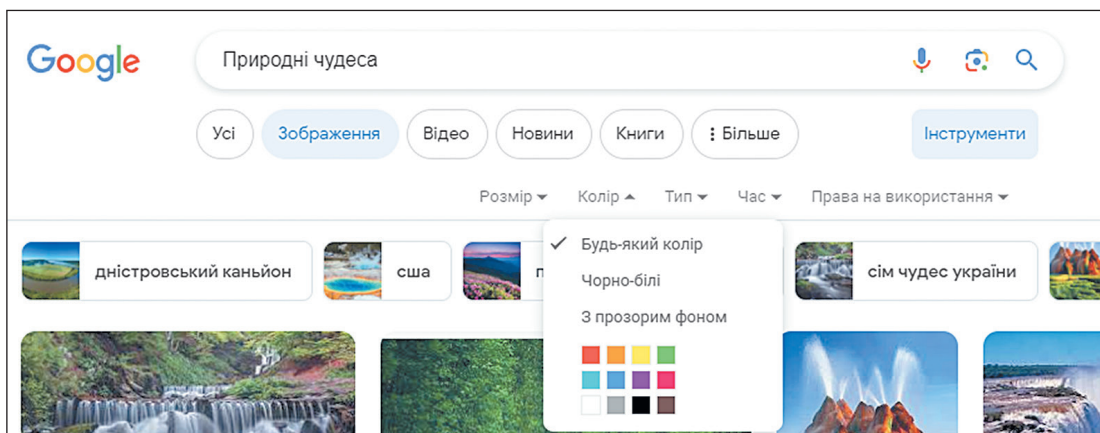
Мал. 1.2. Списки з інструментами пошуку

Для обмеження результатів пошуку в пошуковій системі **Google** також можна використати кнопку **Інструменти**. Після її вибору

відображаються списки для уточнення пошукового запиту (мал. 1.2).

У списку **Будь-якою мовою** можна вибрати, виводити посилання на сторінки будь-якою з доступних мов або лише українською. У списку **Будь-коли** можна вибрати період, упродовж якого матеріали було розміщено в Інтернеті: за минулу годину, добу, тиждень, місяць, рік, будь-коли або у визначений вами проміжок часу. Вибір у списку **Усі результати** визначає, будуть виведені посилання на всі сторінки, у яких містяться введені вами ключові слова в будь-якому відмінку, чи лише ті сторінки, у яких ваш запит присутній дослівно.

Якщо до результатів пошуку застосувати фільтр **Зображення**, то інструменти пошуку будуть іншими (мал. 1.3).



Мал. 1.3. Інструменти пошуку зображень

Зображення можна фільтрувати за значеннями таких властивостей:

- **Розмір:** *Будь-якого розміру*, *Великі* (ширина або висота зображення понад 1000 пікселів), *Середні* (ані ширина, ані висота зображення не перевищують 1000 пікселів), *Значок* (ширина та висота зображення не більші за 256 пікселів);
- **Колір:** *Будь-який колір*, *Чорно-білі*, *З прозорим фоном*, на зображенні переважає вибраний колір або його відтінок;

- **Тип:** *Будь-які, Ілюстрації* (кольорові мальовані зображення), *Графічний об'єкт* (чорно-білі мальовані зображення), *GIF* (анімовані зображення);
- **Час** розміщення в Інтернеті: *Будь-коли, За минулі 24 години, За минулий тиждень, За минулий місяць, За минулий рік;*
- **Права на використання:** *Усі, Ліцензії Creative Commons, Комерційні та інші ліцензії.*

РОЗШИРЕНИЙ ПОШУК



Поміркуйте

- Розгляньте фрагменти сторінок з результатами пошуку (мал. 1.4) та поміркуйте, чи відповідають вони пошуковому запиту з ключовою фразою *природні чудеса*. Як це залежить від розташування ключових слів на сторінці?

Десять природних чудес України, які треба обов'язково ...

Пропонуємо дізнатися про десять **природних чудес** нашої країни, які варто побачити кожному туристу. Тунель кохання. Казковий тунель є ботанічним феноменом, ...

а)

Допомога. Відповіді на запитання про Планету Кіно

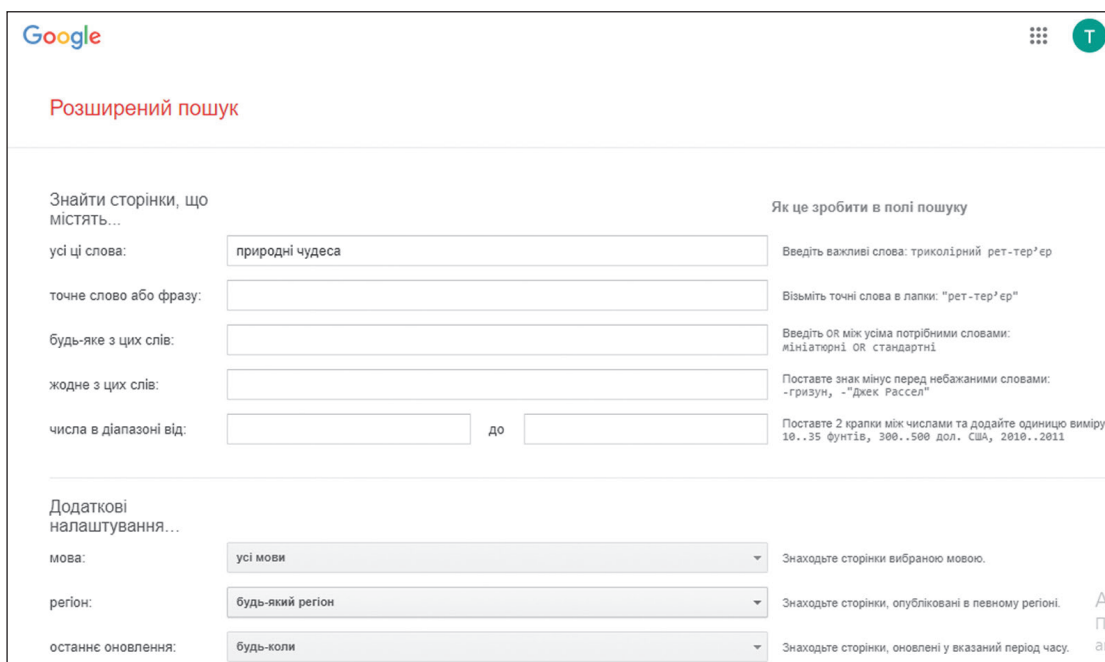
... Сем Мендес («007 Координати Скайфол»), Тім Бертон («Аліса в країні чудес», ... У кінотеатрі дуже реалістично та видовишно відтворюються такі **природні** ...

б)

Мал. 1.4

З метою налаштування ще ретельнішого відбору сторінок під час пошуку відомостей в Інтернеті можна відкрити сторінку **Розширений пошук** (мал. 1.5). Для цього потрібно на сторінці пошукової системи **Google** після виконання простого пошуку вибрати кнопку **Інструменти** та посилання **Розширений пошук**.

У розділі **Знайти сторінки, що містять...** можна у відповідному полі ввести слова або фрази, за якими буде викону-



Мал. 1.5. Сторінка **Розширений пошук** у пошуковій системі **Google**

ватися відбір посилань на знайдені сторінки. У таблиці 1.1 наведено опис сторінок, посилання на які будуть отримані під час заповнення вмісту полів, якщо у відповідне поле введено ключові слова *природні чудеса*.

Таблиця 1.1

Результат заповнення полів сторінки Розширений пошук

Поле	Опис результату	Пояснення результату
усі ці слова	Кожне з ключових слів міститься на сторінці, але не обов'язково в одному реченні	На знайдених сторінках може міститися текст <i>природні чудеса</i> , а також, наприклад, текст <i>природні явища</i> та <i>чудеса героїзму</i>
точне слово або фразу	На сторінці міститься точна ключова фраза, у якій слова розміщено в указаній послідовності	На знайдених сторінках обов'язково міститься текст <i>природні чудеса</i>

Поле	Опис результату	Пояснення результату
будь-яке з цих слів	На сторінці міститься хоча б одне з указаних слів	На сторінці може міститися, наприклад, текст <i>природні явища</i> , або текст <i>чудеса героїзму</i> , або і перше, і друге
жодне з цих слів	На сторінці не міститься жодного з уведених слів	На сторінках не може міститися ані слово <i>природні</i> , ані слово <i>чудеса</i>

Поля **числа в діапазоні** заповнюють, якщо потрібно, щоб на сторінці обов'язково містилися числа з указанного діапазону, наприклад від 100 до 200.

У списках і полях розділу **Додаткові налаштування** можна вибрати для виведення в результатах пошуку посилань лише на сторінки, які:

- **мова** – мають вміст указаною мовою;
- **регіон** – зареєстровані в певній країні або регіоні;
- **останнє оновлення** – оновлення яких відбулося в зазначений період часу;
- **сайт чи домен** – уходять до вказаного сайту, наприклад wikipedia.org, або мають визначену частину (домен) в адресі, наприклад .gov або .ua;
- **терміни, які відображаються** – у яких ключові слова містяться в певній частині сторінки: у заголовку, в тексті, в адресі сторінки або ін.;
- **тип файлу** – містять файли вибраного формату, наприклад комп'ютерні презентації формату *pptx*;
- **права на використання** – дозволені для використання, розповсюдження чи змінення або мають певні заборони.

Для отримання результатів пошуку після заповнення полів або вибору значень у списках потрібно вибрати кнопку **Розширений пошук**.

Якщо на сторінці пошукової системи ви обирали фільтр **Зображення** або **Відео** для уточнення результатів пошуку, то після вибору кнопки **Інструменти** та посилання **Розширений пошук** будуть відкриті сторінки **Розширений пошук зображень** або **Розширений пошук відео**.

У розділі **Додаткові налаштування** для пошуку зображень, крім іншого, надано можливість відбирати зображення бажаного розміру, якості, форми, типу і формату, а для пошуку відео – потрібної тривалості, якості та наявності субтитрів.

На сторінках розширеного пошуку в розділі **Також можна...** подано гіперпосилання для ознайомлення з іншими рекомендаціями довідкового центру **Google** щодо налаштування пошуку.

СТВОРЕННЯ СПИСКІВ ДЖЕРЕЛ І ЗАКЛАДОК



Поміркуйте

• Як за потреби можна повторно знайти сторінку, яку ви відвідували раніше? • Що потрібно знати для дотримання закону про авторські права?

Деякі вебсторінки, знайдені в Інтернеті, містять відомості, які можуть бути вам корисні в майбутньому. Адреси таких сторінок варто зберігати для повторного їх перегляду.



Одним зі способів збереження адрес вебсторінок є створення **списку джерел**. Зазвичай його створюють у текстовому документі та зазначають такі відомості:

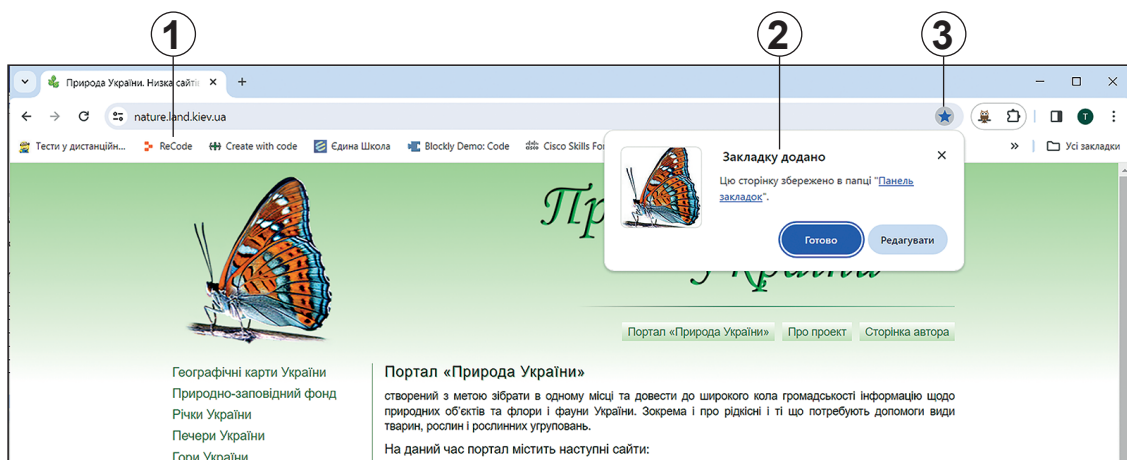
- адреса сторінки, скопійована з адресного рядка браузера;
- назва сайту та сторінки;
- імена авторів матеріалів, якщо вони зазначені на сторінці;
- дата перегляду сторінки.

Такі списки джерел використовують у ході виконання проєктів для обміну адресами корисних ресурсів між учасниками та учасницями групи, а також для розміщення у презентаціях, документах, доповідях, роботах МАН тощо для дотримання вимог закону про авторські права.

Крім цього, у браузері можна створювати **закладки** на сторінки, завдяки яким зберігають та упорядковують адреси відвіданих вебсторінок.

Для створення закладки на відкриту для перегляду сторінку призначена кнопка **Зробити закладку для цієї вкладки**

 , яку розміщено в адресному рядку браузера праворуч від адреси. Після вибору цієї кнопки змінюється її колір  і назва на **Змінити закладку для цієї вкладки** та відкривається вікно **Закладку додано** (мал. 1.6).



Мал. 1.6. Створення закладки на відкриту сторінку:
1 – **Панель закладок**; 2 – вікно **Закладку додано**;
3 – кнопка **Змінити закладку для цієї вкладки**

Щоб підтвердити створення закладки з налаштуваннями за замовчуванням, потрібно вибрати кнопку **Готово**.

Для розміщення закладок призначена **панель закладок**, яка залежно від налаштувань браузера відображається

у верхній частині вікна під адресним рядком на кожній сторінці або лише на сторінці нової вкладки.

Кожна закладка має ім'я та пов'язана з адресою вебсторінки. Перехід до перегляду потрібної сторінки відбувається вибором відповідної закладки на панелі закладок.

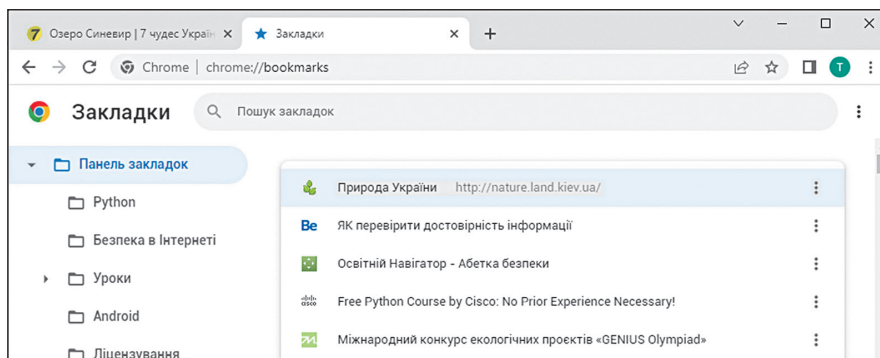
Закладки на сторінки схожого змісту можна групувати в папки. Для створення нової папки потрібно:

1. Вибрати у вікні **Закладку додано** кнопку **Редагувати**.
2. Вибрати у списку **Папка** команду **Вибрати іншу папку**.
3. Вибрати на панелі **Редагування закладки** місце для розташування нової папки.
4. Вибрати кнопку **Нова папка** та ввести ім'я папки.
5. Вибрати кнопку **Зберегти**.


Після виконання цього алгоритму закладка на сторінку буде розміщена у створеній папці. Якщо під час створення закладки її потрібно розмістити в папці, яка вже існує, то ім'я цієї папки можна вибрати у списку **Папка** вікна **Закладку додано**.

Для редагування закладки на відкриту для перегляду сторінку можна вибрати кнопку **Змінити закладку для цієї вкладки** та змінити ім'я, місце розміщення закладки або видалити її.

Побачити всі створені закладки та відредагувати їх можна на сторінці **Закладки** – службовій сторінці браузера **Google Chrome** (мал. 1.7).




Мал. 1.7. Сторінка **Закладки**

Щоб відкрити цю сторінку, потрібно вибрати кнопку **Налаштування та керування Google Chrome**  і виконати **Закладки й списки** ⇒ **Диспетчер закладок**.

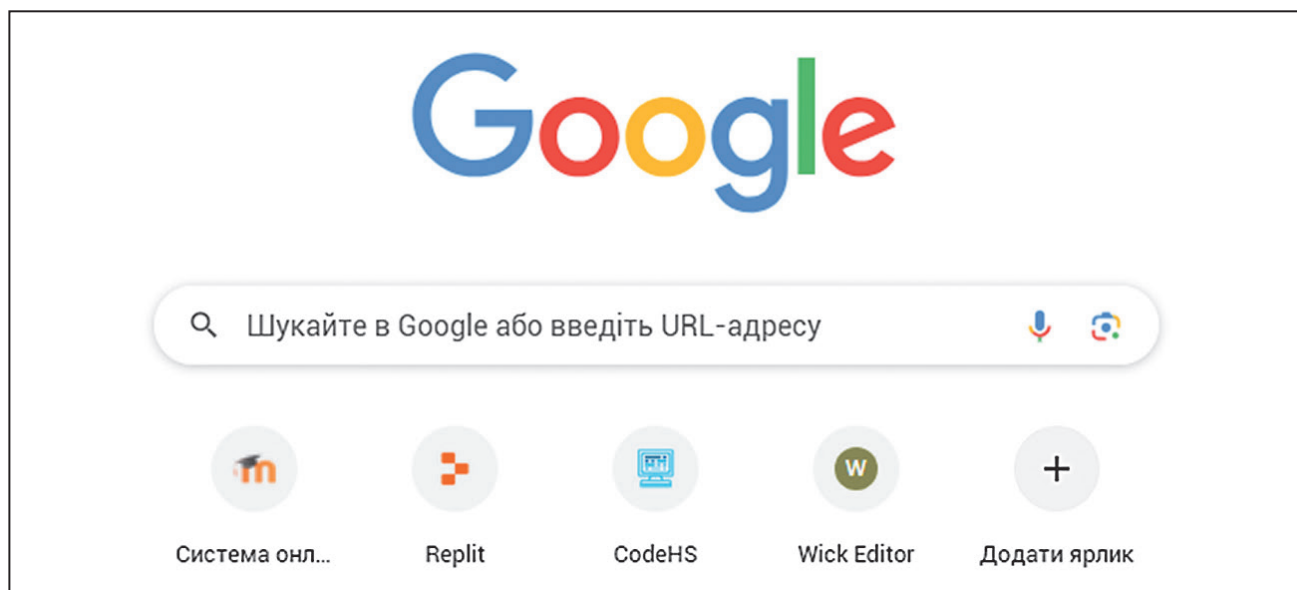
Для тих, хто хоче знати більше

Якщо деякі сторінки в Інтернеті ви переглядаєте регулярно, то для десяти з них можна створити ярлик на сторінці нової вкладки браузера (мал. 1.8).

Для цього слід:

1. Скопіювати адресу потрібної сторінки.
2. Відкрити у браузері нову вкладку.
3. Вибрати кнопку **Додати ярлик**  на новій вкладці.
Додати ярлик
4. На панелі **Додати ярлик**, що відкриється, увести назву сторінки та вставити скопійовану адресу.
5. Вибрати кнопку **Готово**.

У такий спосіб можна створити до 10 ярликів на сторінки, які ви часто переглядаєте, для швидкого переходу до них.



Мал. 1.8. Ярлики вебсторінок на сторінці нової вкладки браузера



Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви можете знайти за адресою <https://cutt.ly/ewNR6dXv> або QR-кодом.



Найважливіше в цьому пункті

Для відображення в результатах пошуку посилань лише на дані певних видів або певного призначення в пошуковій системі **Google** можна застосовувати **фільтри** до результатів пошуку.

Для обмеження результатів пошуку також можна використати кнопку **Інструменти**. Після її вибору відображаються списки для уточнення пошукового запиту.

З метою ретельнішого відбору сторінок під час пошуку відомостей в Інтернеті можна використовувати **Розширений пошук**.

Одним зі способів збереження адрес вебсторінок є створення **списку джерел**. Зазвичай у ньому зазначають такі відомості: адреса сторінки, назва сайту та сторінки, імена авторів матеріалів, дата перегляду сторінки.

У браузерях можна створювати **закладки** на сторінки, завдяки яким зберігають та упорядковують адреси відвіданих вебсторінок.



Дайте відповіді на запитання

1. Для чого призначені кнопки фільтрування результатів пошуку в пошуковій системі **Google**? Які фільтри можна застосувати до результатів пошуку?

2. За значеннями яких властивостей можна звужувати загальні результати пошуку, а також результати пошуку зображень і відео, використовуючи інструменти пошуку в пошуковій системі **Google**?

3. Як виконати розширений пошук у пошуковій системі **Google**?

4. Які відомості про вебсторінки включають у список джерел?

5. Як створити закладку в браузері на відвідану сторінку?




Обговоріть і зробіть висновки

1. З якою метою може використовуватись інструмент пошуку зображень **Права на використання**?
2. Чим відрізнятимуться результати розширеного пошуку, у якому встановлено пошук сторінок з *усіма словами* ключової фрази та пошук сторінок з *будь-яким словом* ключової фрази?
3. Чим відрізнятимуться результати розширеного пошуку, у якому встановлено пошук сторінок українською мовою та пошук сторінок з регіону *Україна*?
4. Навіщо у списку джерел зазначають дату перегляду сторінки?




Виконайте завдання

1. Знайдіть, використовуючи інструменти пошуку в пошуковій системі **Google**, кольорову ілюстрацію Сонячної системи середнього розміру, яку можна використовувати з посиланням на автора. Збережіть файл із зображенням у вашій папці, надайте файлу ім'я **завдання 1.1.1**.
2. Знайдіть, використовуючи інструменти пошуку в пошуковій системі **Google**, коротке (до 4 хв) відео про планети Сонячної системи, яке розміщено в Інтернеті за останній рік. Створіть закладку на знайдене відео.
3. Знайдіть, використовуючи розширений пошук, сторінки з України з описом найбільших островів світу, які мають площу розміром від 700 000 км² до 2 000 000 км². Запишіть у зошит ключові слова, які відображаються в полі пошуку, кількість знайдених сторінок, назву та площу найбільшого острова.
4. Знайдіть, використовуючи розширений пошук, презентацію про материки Землі, яку розміщено в Інтернеті на сторінці з України. Збережіть знайдену презентацію у вашій папці, надайте файлу ім'я **завдання 1.1.4**. Запишіть у зошит ключові слова, які відображаються в полі пошуку, та кількість знайдених ресурсів.

 **5.** Проведіть дослідження та з'ясуйте, які позначення можна вводити в полі пошуку на сторінці пошукової системи **Google** для уточнення результатів замість використання сторінки розширеного пошуку. Створіть текстовий документ, у якому заповніть таблицю:

Критерій	Позначення	Приклад
Точне слово або фразу		
Жодне зі слів		
Діапазон чисел		
Сайт чи домен		
Тип файлу		

Збережіть документ у вашій папці у файлі з іменем **завдання 1.1.5**.

 **6. Працюємо у групах.** Об'єднайтеся у групи по 3 особи та реалізуйте проєкт *Туристичні маршрути нашого краю*. Виконайте пошук в Інтернеті цікавих місць вашого регіону, проаналізуйте їх розташування на картах. Прокладіть маршрут, розрахуйте його довжину, запропонуйте послідовність проходження та список об'єктів для огляду на маршруті. Складіть у текстовому документі список посилань на знайдені ресурси, збережіть його у вашій папці у файлі з іменем **завдання 1.1.6.docx**. Створіть комп'ютерну презентацію з представленням запропонованого вашою групою маршруту. Збережіть презентацію у вашій папці у файлі з іменем **завдання 1.1.6.pptx**.

Готуємось до вивчення нового матеріалу

1. Пригадайте, що таке *факт* і *фейк*. Як їх можна розрізнити?
2. Що таке *дослідження*? Якими способами можна збирати дані для проведення дослідження?

1.2. Збирання даних. Перевірка достовірності. Ознаки мови ворожнечі, маніпуляції, пропаганди

У цьому пункті йтиметься про:

- ▶ методи збирання даних для проведення досліджень;
- ▶ перевірку достовірності зібраних даних;
- ▶ розпізнавання фактів і фейків;
- ▶ ознаки мови ворожнечі;
- ▶ розпізнавання та протидію маніпуляціям і пропаганді.

ЗБИРАННЯ ТА АНАЛІЗ ДАНИХ



Поміркуйте

- На які дані ви орієнтуєтесь, коли плануєте купити для себе новий одяг? Як ви отримуєте ці дані?
- Які дані вам потрібно врахувати, щоб вирішити, у яку спортивну секцію записатися?

Під час навчання та в побуті ви часто проводите дослідження, навіть не помічаючи цього. Який фасон і колір одягу модний в цьому сезоні? Які спортивні секції є у вашому районі? Які природні та історичні пам'ятки є привабливими туристичними об'єктами у вашій місцевості? **Шукаючи відповіді на подібні запитання, доводиться збирати та аналізувати певні дані.**

Методи збирання даних можуть бути різними: спостереження за навколишнім світом, опитування груп людей, опрацювання друкованих джерел та історичних знахідок, пошук потрібних відомостей в Інтернеті, фіксація даних з датчиків (сенсорів), результатів експериментів та інші (мал. 1.9).

Зібрані дані використовують для формулювання висновків та ухвалення на їх основі рішень щодо подальших дій. Правильність висновків суттєво залежить від якості та обсягу зібраних даних. Покупка одягу буде невдалою, якщо ви вивчали застарілі джерела про модні тенденції або не врахували особливості своєї фігури. Записатися у вибрану вами спор-



Мал. 1.9. Методи збирання даних

тивну секцію не вдасться, якщо ви не з'ясували, для якої вікової категорії проводяться заняття, або не врахували час проведення занять. Туристичний бізнес не принесе прибутку, якщо не з'ясувати вподобання можливих клієнтів і не створити для них належні умови.

Для прийняття надійних рішень важливо вибирати правильні методи, інструменти та джерела збирання даних (інформаційні сайти, соцмережі, ЗМІ, книжки, результати експерименту тощо) та здійснювати перевірку достовірності зібраних даних.

ПЕРЕВІРКА ДОСТОВІРНОСТІ ДАНИХ. ФАКТИ ТА ФЕЙКИ



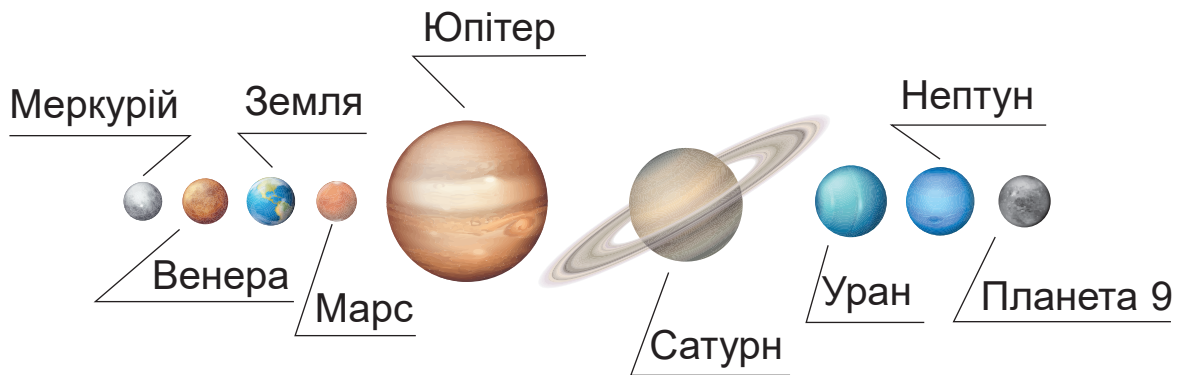
Пригадайте

- Що таке *факт*? Що таке *фейк*?
- Як перевірити надійність медіа-тексту?
- Як виконати пошук за зображенням у пошуковій системі **Google**?

Ви вже знаєте, що достовірність будь-яких даних, які ви використовуєте у своїх дослідженнях, потрібно перевіряти. Особливо тих даних, які ви отримуєте під час пошуку в Інтернеті. Дані можуть бути **недостовірними**, якщо вони:

- **застарілі**, не відповідають реаліям сьогодення. Наприклад, на основі даних про користувачів Інтернету найпопулярнішим браузером в Україні кілька років тому вважався **Internet Explorer**, але наразі ці дані застарілі, популярність цього браузера суттєво зменшилась;

- **не відповідають сучасним науковим поглядам.** Наприклад, дані про кількість планет у нашій Сонячній системі неодноразово переглядалися вченими за результатами астрономічних досліджень. Продовжується дослідження цього питання і в наш час (мал. 1.10);



Мал. 1.10. Сучасне уявлення про планети Сонячної системи

- **неповні.** Наприклад, дані про найпопулярніший серед українців месенджер будуть неповними, а тому недостовірними, якщо проводити опитування з цього питання лише серед підлітків і не враховувати думку людей старшого віку;
- **неправдиві,** тобто свідомо або випадково спотворені. Наприклад, людина може випадково помилитися в записах під час фіксування результатів експерименту або навмисно використати хибні дані, які вигідні їй, та в інших випадках.

Ви знаєте, що неправдиві відомості, які розповсюджують між людьми, у тому числі розміщують в Інтернеті, називають **фейками**. Збираючи дані, потрібно вміти розпізнавати фейки та відрізнити їх від **фактів**. Перевірка достовірності відомостей отримала назву **фактчекінг** (англ. *fact checking* – перевірка фактів).

Щоб переконатися в достовірності відомостей, знайдених в Інтернеті:

- перевіряйте знайдені відомості щонайменше у трьох різних джерелах. Якщо відомості правдиві, то вони обов'язково будуть підтверджуватися на кількох ресурсах;

- аналізуйте, з якого джерела отримано відомості. Якщо матеріали ви отримали із сайту офіційної організації (в адресі сайту присутня частина **.gov** – урядова організація або **.edu** – освітня установа), то цим даним зазвичай можна довіряти;
- з'ясовуйте, чи наведено відомості про автора знайдених вами матеріалів і чи є він фахівцем з питання, яке ви вивчаєте. А якщо відомості про автора відсутні або він не є фахівцем, то такі дані є сумнівними;
- перевіряйте дату публікації або останнього оновлення матеріалів на сайті. Можливо, відомості, які ви знайшли, уже застарілі.

Варто перевіряти достовірність не лише текстових повідомлень, а й зображень, знайдених в Інтернеті. Їх можуть підробити з використанням графічного редактора, додати неіснуючі об'єкти або видозмінити існуючі. Іноді зображення не відповідають часу або місцю події, про яку йдеться в повідомленні поруч із зображенням. Такі недостовірні зображення називають **фотофейками**. На малюнку 1.11 наведено один з відомих фотофейків – Альберт Ейнштейн на фоні ядерного вибуху.

Для перевірки достовірності зображення можна спробувати знайти початкове зображення (**першоджерело**), на основі якого міг бути створений фотофейк. На малюнку 1.12 наведено дві фотографії – першоджерела, на основі яких було створено фотофейк про Альберта Ейнштейна.



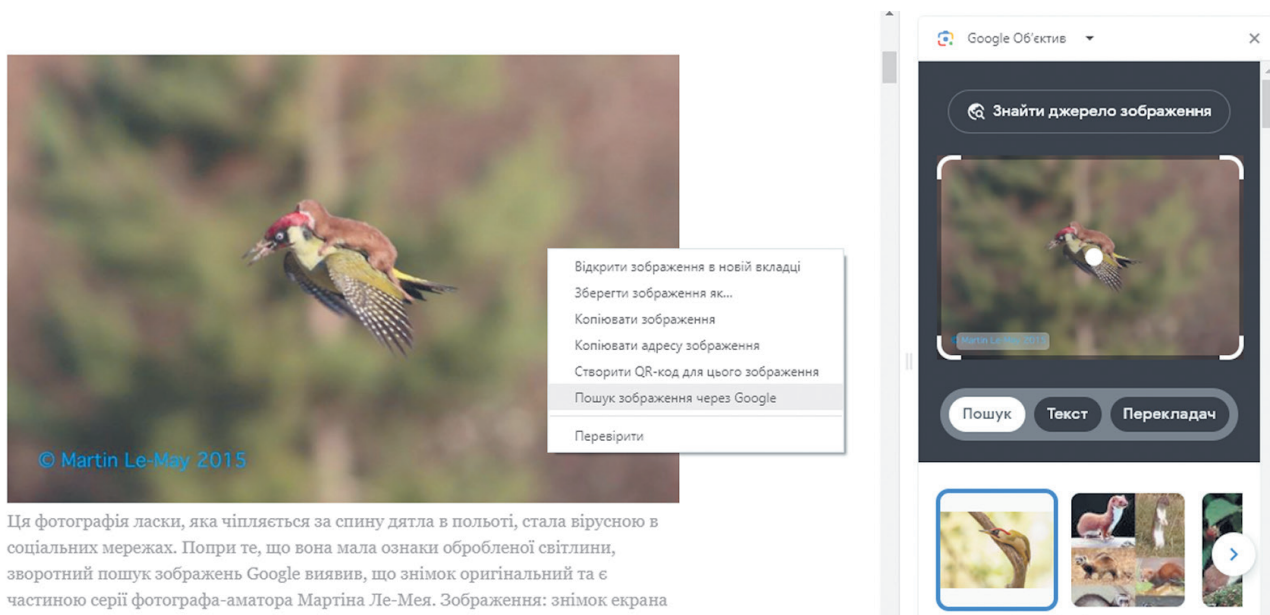
Мал. 1.11. Фотофейк: Альберт Ейнштейн на фоні ядерного вибуху



Мал. 1.12. Першоджерела для фотофейку

Пошук першоджерел, з яких утворено фотофейк, називають **зворотним пошуком**. Виконати зворотний пошук можна різними способами, наприклад:

- виконати пошук за зображенням у пошуковій системі **Google** сумнівного зображення;
- вибрати команду **Пошук зображення через Google** у контекстному меню сумнівного зображення на вебсторінці з повідомленням (мал. 1.13);
- здійснити пошук за зображенням з використанням спеціалізованих вебресурсів, наприклад **TinEye** (tineye.com) або **Photo Sherlock** (photosherlock.com).



Ця фотографія ласки, яка чіпляється за спину дятла в польоті, стала вірусною в соціальних мережах. Попри те, що вона мала ознаки обробленої світлинки, зворотний пошук зображень Google виявив, що знімок оригінальний та є частиною серії фотографа-аматора Мартіна Ле-Мея. Зображення: знімок екрана

Мал. 1.13. Результати виконання команди **Пошук зображення через Google**

Після завершення зворотного пошуку потрібно проаналізувати отримані результати – переглянути вміст знайдених сторінок, порівняти вигляд знайдених зображень, дату їх розміщення в мережі тощо, щоб зробити висновок про достовірність зображень.

Останнім часом в Інтернеті почали з'являтися недостовірні фальшиві відео, створені з використанням штучного інтелекту, так звані **діпфейки** (англ. *deepfake* – підроблене відео). У них на справжнє відео накладають інше відео або зображення для отримання недостовірного сюжету. Назва *діпфейк* походить від поєднання термінів *deep learning* (англ. *deep learning* – глибинне навчання, назва алгоритму штучного інтелекту) та *fake* (англ. *fake* – підробка). Розпізнати такі підробки складно, для цього потрібен аналіз фахівців.

МАНІПУЛЯТИВНІ ПОВІДОМЛЕННЯ, ПРОПАГАНДА, МОВА ВОРОЖНЕЧІ



Поміркуйте

- Чому можна довіряти фактам і не можна довіряти судженням?
- Чим може нашкодити людині фейкове повідомлення?

Повідомлення, які ви читаєте в Інтернеті, можуть виявитися **маніпулятивними**.

Маніпуляція – це спосіб впливу на думки, настрої, ставлення до подій, вчинки людей для отримання вигоди; техніка цілеспрямованого викривлення відомостей заради формування певного погляду, ставлення до тієї або іншої проблеми.

Маніпулятори орієнтуються на людські слабкості та емоції, особливості психіки людини, її способів мислення. Вони намагаються примусити людину не обмірковувати ситуацію, а приймати рішення та здійснювати вчинки на підставі емоцій і почуттів – страху, сорому, гніву, жорстокості, жадібності, почуття провини тощо.

Існують різні техніки маніпулювання думками людей.

Пропаганда – це поширення інформації (фактів, аргументів, чуток, напівправди чи брехні) з метою впливу на суспільну думку для змінення системи цінностей і поведінки. Наприклад, російська пропаганда намагається у свідомості людей виправдати збройну агресію проти України.

Пропагандистські повідомлення поширюються через телебачення, радіо, друковані засоби масової інформації, дописи в соцмережах, публікації в Інтернеті та нав'язують людям точку зору, вигідну пропагандистам (мал. 1.14).



Мал. 1.14. Засоби поширення пропаганди

Дискримінація (лат. *discriminatio* – розрізнення) – будь-яка відмінність, виключення, обмеження або перевага, що заперечує або зменшує рівне здійснення прав.

Мова ворожнечі (хейт) (англ. *hate speech* – мова ненависті) – це агресивні висловлювання з метою створення негативного образу людини або групи людей за ознакою раси (расизм), національності, політичних поглядів, релігії, статі (сексизм) тощо. Наприклад, мовою ворожнечі є твердження про те, що українців не існувало в історії, а заснована Україна лише у ХХ столітті.

Ознаками мови ворожнечі є:

- принизливі висловлювання та заклики до насильства або обмеження прав (дискримінації) відносно людей певної групи;
- безпідставні звинувачення, твердження про неповноцінність або моральні недоліки лише на основі належності до визначеної групи;
- виправдання історичних випадків насильства відносно етнічних або релігійних груп тощо.

Щоб розпізнати маніпулятивне повідомлення, користуйтеся такими порадами:

- Зверніть увагу, які емоції у вас викликає повідомлення. Якщо заголовок публікації або його зміст занадто емоційний, сенсаційний, викликає злість або тривожність, це може бути спробою вплинути на ваше сприйняття. Критично ставтеся до таких повідомлень.
- Відділяйте в повідомленні факти від суджень. Судження можуть бути чиєюсь упередженою точкою зору, спробою маніпуляції. Спирайтесь у своїх висновках на факти, які можна перевірити.
- Зверніть увагу, чи зазначені в повідомленні імена людей або назви засобів масової інформації, які є джерелами інформації. Якщо в тексті посилаються на анонімні джерела або не називають конкретних прізвищ експертів, думку яких поширюють, то це може бути ознакою маніпуляції.
- Проаналізуйте повідомлення, у ньому подано лише одну точку зору чи різні. Одностороннє подання інформації є ознакою маніпуляції.



Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви можете знайти за адресою <https://cutt.ly/5wNR6K3U> або QR-кодом.



Найважливіше в цьому пункті

Методи збирання даних: спостереження за навколишнім світом, опитування груп людей, опрацювання друкованих джерел та історичних знахідок, пошук потрібних відомостей в Інтернеті, фіксація даних з датчиків (сенсорів), результатів експериментів та інше. Зібрані дані використовують для формулювання висновків та ухвалення на їх основі рішень щодо подальших дій.

Дані можуть бути **недостовірними**, якщо вони застарілі, не відповідають сучасним науковим поглядам, неповні, неправдиві та в інших випадках. Перевірка достовірності відомостей отримала назву **фактчекінг**.

Недостовірні зображення називають **фотофейками**. Для перевірки достовірності зображення можна спробувати знайти початкове зображення (**першоджерело**), на основі якого міг бути створений фотофейк. Пошук першоджерел, з яких утворено фотофейк, називають **зворотним пошуком**.

Недостовірні фальшиві відео, створені з використанням штучного інтелекту, називають **діпфейками**.

Маніпуляція – це спосіб впливу на думки, настрої, ставлення до подій, вчинки людей для отримання вигоди; техніка цілеспрямованого викривлення відомостей заради формування певного погляду, ставлення до тієї або іншої проблеми.

Пропаганда – це поширення інформації (фактів, аргументів, чуток, напівправди чи брехні) з метою впливу на суспільну думку для змінення системи цінностей і поведінки.

Мова ворожнечі (хейт) – це агресивні висловлювання з метою створення негативного образу людини або групи людей за ознакою раси (расизм), національності, політичних поглядів, релігії, статі (сексизм) тощо.



Дайте відповіді на запитання

1. Які методи збирання даних ви знаєте?
2. Які дані можуть бути недостовірними?
3. Як перевірити достовірність даних, знайдених в Інтернеті?
4. Що таке *фейк*, *фотофейк*, *діпфейк*?
5. Як можна перевірити достовірність зображень?



Обговоріть і зробіть висновки

1. Як залежать результати досліджень від якості та обсягу зібраних даних? Наведіть приклади.
2. Як можна розпізнати пропаганду та протистояти їй?
3. Які ознаки мови ворожнечі? Як їй протистояти?



Виконайте завдання


1. Уявіть, що ви збираєтесь у туристичний похід. Запишіть у зошит відповіді на запитання:

а) На основі яких даних ви будете приймати рішення про те, які речі вам потрібно взяти із собою в туристичний похід?

б) Як ви збиратимете ці дані?

в) Як можуть на вас вплинути помилкові дані?

2. Знайдіть в Інтернеті відомості про те, як материки Землі отримали свої назви. Створіть текстовий документ, скопіюйте до нього фрагменти тексту з версіями утворення назв материків і адреси трьох ресурсів, які підтверджують ці версії. Збережіть документ у вашій папці у файлі з іменем **завдання 1.2.2.**

3. Знайдіть з використанням сайту **TinEye** (tineye.com) першоджерело фотографії з адресою https://medialab.online/wp-content/uploads/2021/08/77108672_584637765608381_3135717809917526016_n.jpg. . Визначте достовірність фотографії.

4. Проаналізуйте заголовки дописів у засобах масової інформації. Визначте, які з цих дописів містять фейки, маніпуляції, судження, пропаганду, мову ворожнечі:


а) Найкращий урожай серед усіх фруктів у цьому році буде в яблук.

б) Сенсація! У Дніпрі виловили невідому науці істоту!


в) Усі рудоволосі – хитруни. Не дай себе обдурити.

г) Лише нашою ручкою ти напишеш усі контрольні на 12 балів!

д) Наша команда – найсильніша! Приєднуйся до нас!

 **5.** Знайдіть в Інтернеті пояснення поняття *клікбейт*. Ознайомтесь з його характерними ознаками. Доберіть приклади клікбейту із сайтів у Інтернеті, соціальних мереж, засобів масової інформації. Створіть текстовий документ, у якому наведіть знайдені приклади та опишіть, за якими ознаками ви

розпізнали клікбейт. Збережіть документ у вашій папці у файлі з іменем **завдання 1.2.5**.

 **6.** Проведіть дослідження та з'ясуйте, на сайтах яких організацій можна перевірити сенсаційну інформацію про:

- а) сходження лавини в Карпатах;
- б) запуск нового потяга з України до Болгарії;
- в) можливі ризики вакцинації проти коронавірусу.

Створіть презентацію з відомостями про офіційні організації, які можуть підтвердити або спростувати відомості з наведених питань. Збережіть презентацію у вашій папці у файлі з іменем **завдання 1.2.6**.

Готуємось до вивчення нового матеріалу

- 1.** Які засоби ви використовуєте для спілкування з близькими та друзями на відстані?
- 2.** Чи доводилося вам отримувати паперові листи? Які відомості зазначають на конверті? Що можна вкладати до поштового конверта?

1.3. Електронне листування. **Електронна поштова скринька**

У цьому пункті йтиметься про:

- ▶ поняття *електронна пошта*;
- ▶ вміст електронної поштової скриньки;
- ▶ надсилання, отримання та інші операції з електронними листами.

ПОНЯТТЯ ПРО ЕЛЕКТРОННУ ПОШТУ

Пригадайте

- Якими засобами ви користуєтеся для спілкування з людьми на відстані?
- Завдяки чому поштовий лист потрапляє до тієї людини, якій цей лист призначено?

Комп'ютерні мережі створювалися для надійного та швидкого передавання повідомлень. Однією з найстаріших служб Інтернету, що створювалась для обміну повідомленнями, є **електронна пошта**.

Електронна пошта – це служба Інтернету, призначена для пересилання комп'ютерними мережами повідомлень (електронних листів) від деякого користувача одному чи групі адресатів. Часто електронну пошту називають **e-mail** (англ. *electronic mail* – електронна пошта).

Послугу електронної пошти надають різні ресурси Інтернету, наприклад **FREEMAIL** (mail.ukr.net), **I.UA** (i.ua), **Meta** (meta.ua), **Gmail** (gmail.com), **Outlook** (outlook.live.com) та інші.

Для того щоб мати можливість отримувати, зберігати та надсилати електронні листи, користувач повинен зареєструватися на тому чи іншому поштовому інтернет-ресурсі – створити **обліковий запис**.

Обліковий запис користувача – сукупність даних для розпізнавання користувача під час звернення до поштової служби. Обліковий запис інакше називають **екаунтом** (іноді вживають термін **акаунт**) (англ. *account* – обліковий запис).

Користувачу потрібно придумати умовне ім'я – **логін** (англ. *log in* – увійти, приєднатися) та **пароль** для захисту облікового запису. Під час реєстрації слід увести ці та інші дані, які потрібні для створення та захисту електронної поштової скриньки. Після реєстрації для користувача створюється **електронна поштова скринька**.

Кожна електронна поштова скринька має унікальну адресу. **Адреса електронної поштової скриньки** складається з логіну, під яким зареєструвався користувач, та адреси інтернет-ресурсу, на якому вона створена, розділених символом **@** (англ. *комерційне at* – на).

Наприклад, поштова адреса **pupil@ukr.net** означає, що для користувача з логіном **pupil** створена електронна пошто-ва скринька на сайті з адресою **ukr.net**.

Цікаві факти з історії

Перша поштова програма **SNDMSG** (англ. *send message* – надіслати повідомлення) була розроблена в 1971 році **Реем Томлінсоном** (мал. 1.15) для комп'ютерної мережі **ARPANet**.

Першою адресою електронної пошти була адреса її винахідника **tomlinson@bbn-tenexa**. Для з'єднання частин електронної поштової адреси Рей Томлінсон обрав символ **@**. Українські користувачі частіше називають його сленговими словами «собачка» або «равлик» за його зовнішній вигляд.



Мал. 1.15.
Рей Томлінсон

Першим листом, надісланим електронною поштою, був лист від Рея Томлінсона його колегам з повідомленням про появу нової можливості спілкування в мережі – електронної пошти.

ВМІСТ ЕЛЕКТРОННОЇ ПОШТОВОЇ СКРИНЬКИ

Пригадайте

- Що ви використовуєте в операційній системі **Windows** для впорядкування файлів на носіях даних?
- За якими ознаками ви впорядковуєте файли?

Розглянемо, як користуватися електронною поштою, на прикладі електронної пошти **Gmail** – однієї з послуг, яку надає користувачам Інтернету корпорація **Google**. Адреса відповідної електронної поштової скриньки має вигляд **логін@gmail.com**.

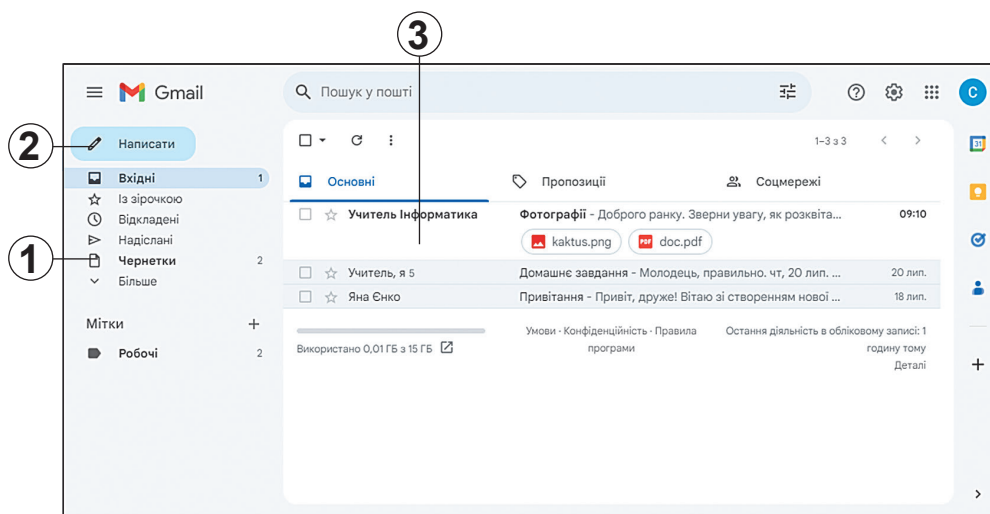
Звертаємо вашу увагу!

Якщо ви маєте екаунт у будь-якому сервісі **Google**, то ви вже маєте електронну поштову скриньку **Gmail**.

Щоб переглянути вміст електронної поштової скриньки, слід:

1. Відкрити головну сторінку сайту **Google** (google.com).
2. Вибрати у верхньому правому куті посилання **Gmail**.
3. Вибрати кнопку **Увійти**, якщо у браузері попередньо не зберігалися дані облікових записів користувачів, або, в іншому випадку, команду **Вибрати інший обліковий запис**.
4. Увести адресу електронної поштової скриньки та вибрати кнопку **Далі**.
5. Увести пароль доступу до електронної поштової скриньки та вибрати кнопку **Далі**.

Вигляд сторінки з вмістом електронної поштової скриньки відрізняється на різних інтернет-ресурсах, але всі вони мають подібні об'єкти (мал. 1.16).



Мал. 1.16. Вікно електронної поштової скриньки **Gmail**:


- 1 – список папок для зберігання електронних листів; 2 – кнопка для створення нового листа; 3 – список листів вибраної папки **Вхідні**

Здебільшого на лівій бічній панелі розміщені посилання, вибравши які можна відкрити папки, призначені для зберігання листів. Зазвичай у поштовій скриньці автоматично створюються такі папки:

- **Вхідні** – для зберігання листів, одержаних власником поштової скриньки;
- **Надіслані** – для зберігання копій відправлених листів;
- **Чернетки** – для зберігання незавершених листів, які користувач планує дописати та відправити пізніше;
- **Спам** (англ. *spam* – консерви, реклама яких була занадто нав'язливою) – для тимчасового зберігання листів, які поштова служба або користувач вважають небажаними: рекламними або випадковими;
- **Видалені (Кошик)** – для тимчасового зберігання видалених листів

та інші.

Після відкриття поштової скриньки у центральній частині вікна автоматично виводиться список листів папки **Вхідні**. Вибравши ім'я іншої папки, можна побачити її вміст. Дані про непрочитані листи відображаються напівжирним накресленням.

Про кожний з листів у відповідному рядку списку наведено, як правило, логін, ім'я або поштову адресу відправника, тему листа, яка стисло пояснює його вміст, перші слова тексту листа, дату або час відправлення та інше. Якщо разом з листом пересилаються файли, то в рядку такого листа може відображатися значок  або кнопки з іменами файлів



ОПЕРАЦІЇ НАД ЕЛЕКТРОННИМИ ЛИСТАМИ



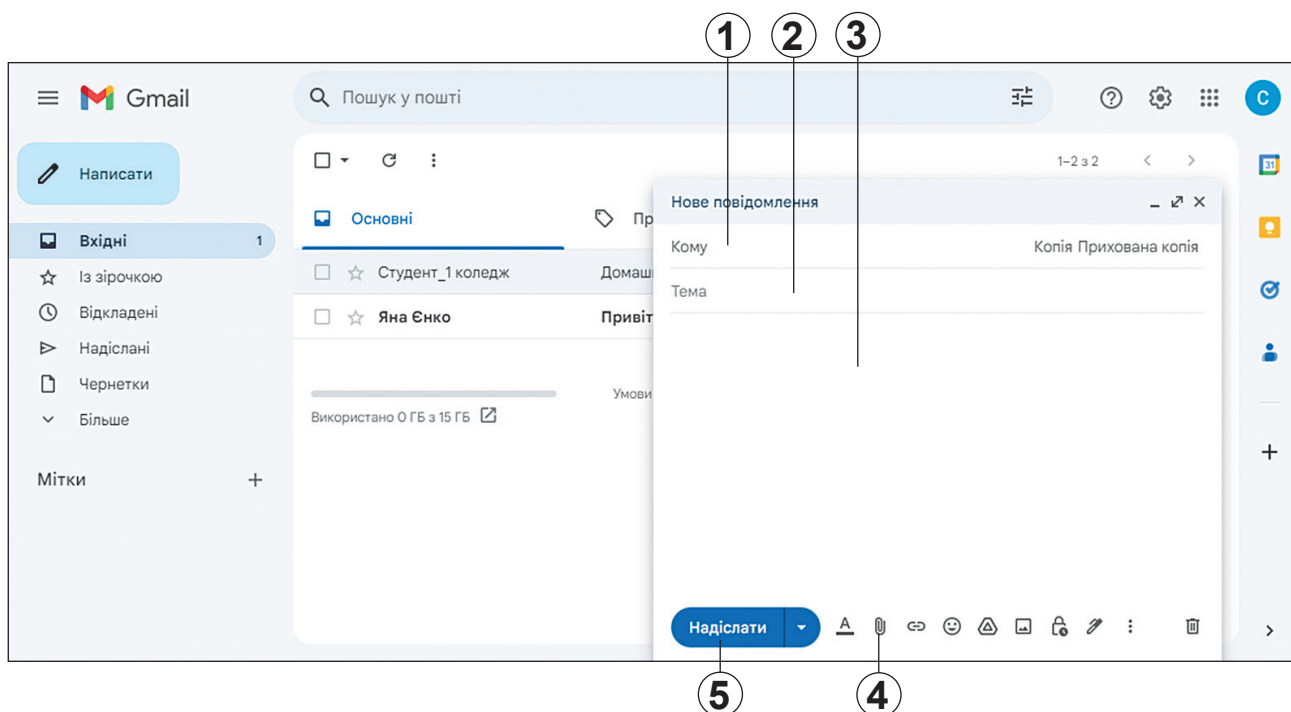
Поміркуйте

● У яких ситуаціях може виникнути потреба додати файли до електронного листа? ● Коли відомостями, які ви отримали в електронному листі, може бути корисно поділитися з іншими людьми? Як було б зручно це зробити?

Для створення нового листа потрібно вибрати кнопку **Написати** у вікні електронної поштової скриньки.

У вікні **Нове повідомлення**, що відкрилось (мал. 1.17), потрібно заповнити поля, розміщені в **заголовку листа**:


- **Кому** – увести одну або кілька адрес електронних пошто-вих скриньок одержувачів листа;
- **Тема** – увести слово або кілька слів, що коротко поясню-ють зміст листа.



Мал. 1.17. Вікно створення електронного листа:

1 – поле **Кому**; 2 – поле **Тема**; 3 – область для уведення тексту листа; 4 – кнопка для вкладання файлів; 5 – кнопка **Надіслати**

Частина вікна, яка розміщена під полем **Тема**, призначена для введення тексту листа.

Якщо разом з текстом листа потрібно надіслати деякі файли, наприклад фотографії, то можна вибрати кнопку **Вкласти файли** , у вікні **Відкриття файлу** вибрати на носії даних файли, призначені для надсилання, та вибрати кнопку **Відкрити**.

Звертаємо вашу увагу!

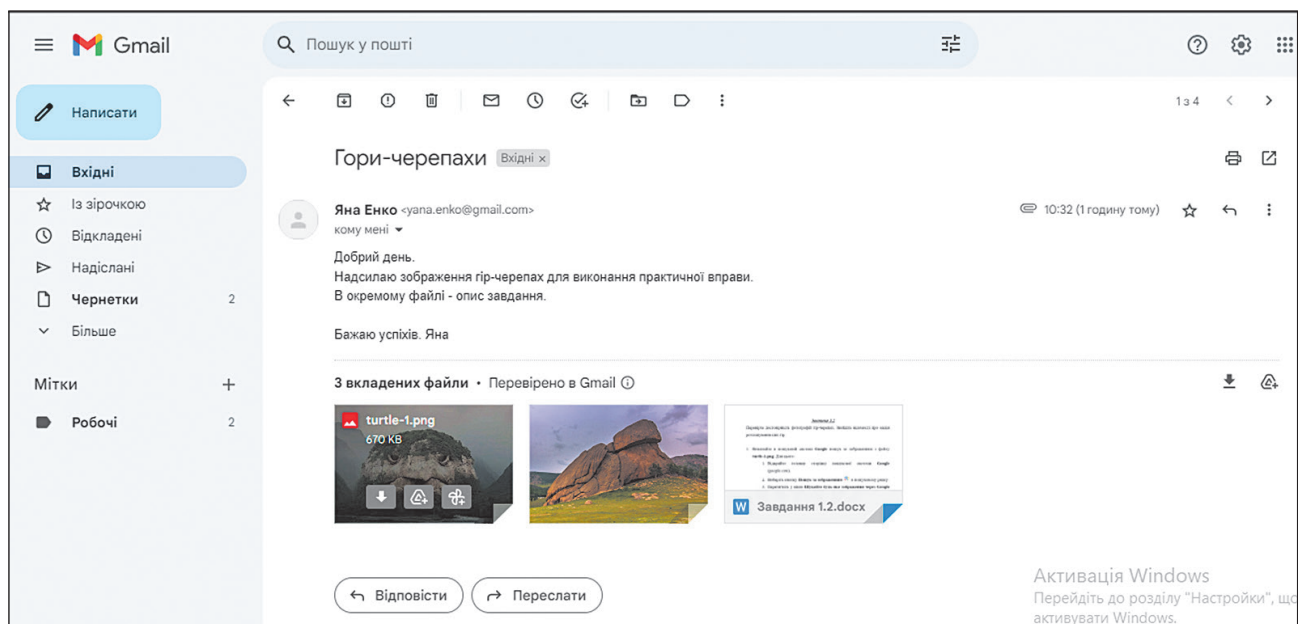
Виконувані файли з розширенням імені **exe** блокуються антивірусною програмою поштової служби та не будуть додані до листа. Вкладені файли, розмір яких перевищує 25 МБ, автоматично завантажуються на **Google Диск** відправника, а у лист додається лише посилання на них.

Після заповнення всіх полів, написання тексту листа та додавання файлів потрібно вибрати кнопку **Надіслати**. Ваш лист буде надіслано до електронних поштових скриньок адресатів, яких ви зазначили.




Щоб прочитати отриманий лист, слід вибрати його рядок у списку листів.

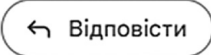
На сторінці відкритого листа (мал. 1.18) користувач може виконати з ним різні операції:

- ознайомитися з текстом листа;
- переглянути або зберегти на носії даних файли, додані до листа. Якщо в системі **Gmail** є додаток, що відповідає типу доданого файлу, то для перегляду потрібно вибрати

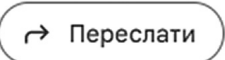


Мал. 1.18. Сторінка отриманого листа

ескіз цього файлу. Для повернення зі сторінки перегляду до вмісту листа слід вибрати кнопку **Закрити**  поруч з іменем файлу, що переглядається. Для збереження даного файлу потрібно вибрати кнопку **Завантажити** , що відображається після наведення вказівника на ескіз файлу, або кнопку **Завантажити**  на сторінці перегляду;

- відповісти на лист, вибравши кнопку **Відповісти**  в нижній частині листа. Тема листа-відповіді буде повторювати тему отриманого листа, лише на початку додаються символи **Re:** (англ. *reply* – відповідь, відповіді). У полі **Кому** буде автоматично розміщено адресу електронної поштової скриньки користувача, на лист якого ви відповідаєте. Якщо лист був надісланий кільком отримувачам, то відповідь можна також надіслати всім. У цьому випадку на сторінці отриманого листа буде кнопка **Відповісти всім**

 Відповісти всім ;

- переслати іншому користувачу, вибравши кнопку **Переслати** . До теми отриманого листа будуть додані символи **Fw:** (англ. *forward* – вперед, далі). Разом з текстом листа пересилаються також укладені файли, якщо вони були в отриманому листі;

- видалити, вибравши кнопку **Видалити** 

та інші.

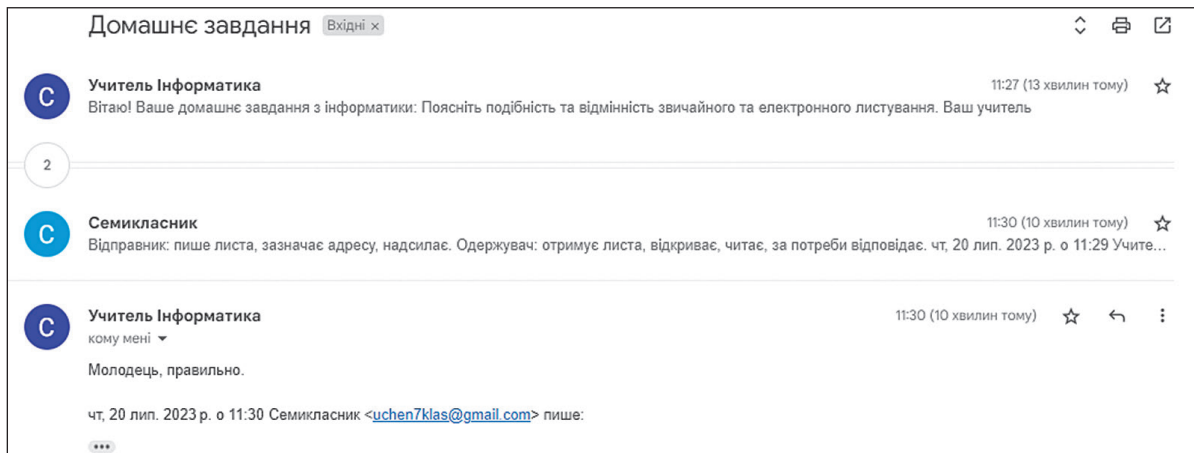
Якщо ви вели переписку з вашим адресатом, відповідаючи на отримані листи, то цей ланцюжок листів групується. У вікні електронної поштової скриньки у рядку такого ланцюжка буде

зазначено кількість листів (мал. 1.19), а вибравши цей рядок, можна побачити текст кожного листа та їх послідовність (мал. 1.20).


Учитель, я 5

Домашнє завдання - Молодець, правильно. чт, 20 лип. 2023 р. о 11:30 Семикласник <u...>

Мал. 1.19. Рядок ланцюжка листів у вікні електронної поштової скриньки



Мал. 1.20. Ланцюжок листів-відповідей

Від групування листів можна відмовитись. Для цього потрібно вибрати у вікні електронної поштової скриньки кнопку **Налаштування**  і на панелі **Швидкі налаштування**, що відкрилася, у розділі **Групування листів** зняти позначку прапорця **Перегляд повідомлень у вигляді ланцюжків**.

Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви можете знайти за адресою <https://cutt.ly/3wNR647w> або QR-кодом.



Найважливіше в цьому пункті

Електронна пошта (e-mail) – це послуга Інтернету, призначена для пересилання комп'ютерними мережами повідомлень (електронних листів) від деякого користувача одному чи групі адресатів.

Адреса електронної поштової скриньки складається з імені користувача, під яким він зареєстрований на сервері поштової служби (**логін**), та адреси поштової служби, розділених символом **@**.

Для створення та надсилання нового листа потрібно вибрати кнопку **Написати**, заповнити поля **Кому** та **Тема**, увести текст листа, вибрати кнопку **Надіслати**. Разом з текстом листа можна пересилати файли.

Щоб прочитати отриманого листа, слід вибрати його рядок у списку листів електронної поштової скриньки. На сторінці відкритого листа можна переглянути або зберегти на носій даних файли, додані до листа, відповісти на лист, переслати іншому користувачу, видалити та виконати інші операції.

Дайте відповіді на запитання

1. Що таке *електронна пошта*? Які вам відомі поштові інтернет-ресурси?
2. З чого складається адреса електронної поштової скриньки?
3. Які папки створюються автоматично в електронній поштовій скриньці? Яке їх призначення?
4. Як створити та надіслати електронний лист?
5. Які операції можна виконати над отриманим листом?

Обговоріть і зробіть висновки

1. У чому схожість і відмінність звичайного та електронного листування?
2. Чому на головній сторінці сайту **Google** створено окреме посилання на поштову службу **Gmail**?
3. Для чого потрібно заповнювати поле **Тема** під час створення електронного листа?
4. У чому зручність групування листів у ланцюжки?

Виконайте завдання


1. Надішліть електронний лист на адресу електронної поштової скриньки вашого однокласника чи однокласниці, що працює з комп'ютером ліворуч від вас. У темі листа вкажіть

ваше прізвище. У тексті листа привітайтеся та опишіть пояснення поняття *обліковий запис користувача*.

2. Надішліть електронний лист на адресу електронної поштової скриньки учителя/вчительки інформатики. Укажіть тему листа *Мій край*. У тексті листа привітайтеся та опишіть одне з улюблених місць вашого краю. До листа додайте фотографію указанного місця, яку знайдіть в Інтернеті або зробіть самостійно. Підпишіть лист вашим прізвищем.

3. Відкрийте електронний лист, отриманий від вашого вчителя/вчительки інформатики, ознайомтесь з умістом листа. Перешліть лист вашому однокласнику/однокласниці.

4. Запропонуйте тему для листа з таким змістом: *Мій собака з породи ньюфаундленд. Собак цієї породи називають «добрими гігантами». У них глибокі розумові здібності, вони легко навчаються, є прекрасними опікунами, надзвичайно лагідні з дітьми.*

 **5. Працюємо у групах.** Об'єднайтесь у групу з 2–3 осіб. Реалізуйте проект *Імена видатних особистостей на географічній карті*. Знайдіть опис і зображення місць, названих іменами видатних особистостей. Перешліть знайдені вами матеріали електронною поштою кожному з групи. Створіть презентацію, у якій розмістіть знайдені відомості. Збережіть презентацію у вашій папці у файлі з іменем **завдання 1.3.5**. Надішліть презентацію електронною поштою вашому вчителю/вчительці інформатики.



Готуємось до вивчення нового матеріалу

1. Як ви зберігаєте адреси друзів і рідних для надсилання їм листів?
2. Яких правил культури поведінки ви дотримуєтесь у спілкуванні з людьми?
3. Які ви знаєте правила безпечної поведінки в Інтернеті?



1.4. Адресна книга. Етикет і правила безпечного електронного листування

У цьому пункті йтиметься про:

- ▶ адресну книгу та список контактів;
- ▶ створення та використання списків розсилання;
- ▶ етикет електронного листування;
- ▶ правила безпечного користування електронною поштою.

АДРЕСНА КНИГА ТА СПИСОК КОНТАКТІВ



Пригадайте

- Які засоби ви використовуєте для зберігання поштових адрес і номерів телефонів ваших знайомих?

Для зберігання поштових адрес, на які ви надсилаєте традиційні паперові листи та вітальні листівки, ви використовуєте спеціальні записники. Під час електронного листування також зручно зберігати адреси для листування у спеціальному електронному записнику. Такий записник називають **адресною книгою**.

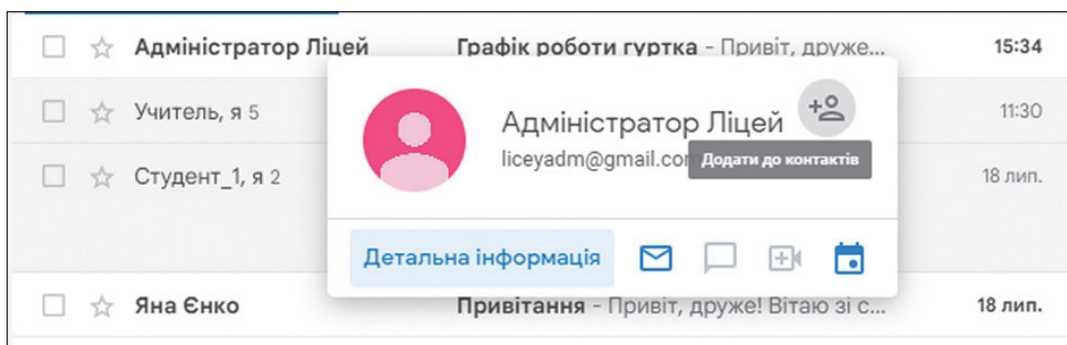
Адресна книга призначена для зберігання **контактів** – даних про осіб, з якими спілкується користувач: імен, адрес електронних поштових скриньок, інших особистих даних. Порожня адресна книга створюється автоматично після створення електронної поштової скриньки.

Користувач може додати дані осіб, з якими вже листувався, до власної адресної книги – списку **Мої контакти**. Для цього слід:

1. Відкрити електронну поштову скриньку.
2. Навести вказівник на ім'я відправника в рядку листа в папці **Вхідні** або на ім'я отримувача – в папці **Надіслані**.
3. Вибрати у спливаючому вікні кнопку **Додати до контактів**




(мал. 1.21).



Мал. 1.21. Додавання контакту до списку **Мої контакти**

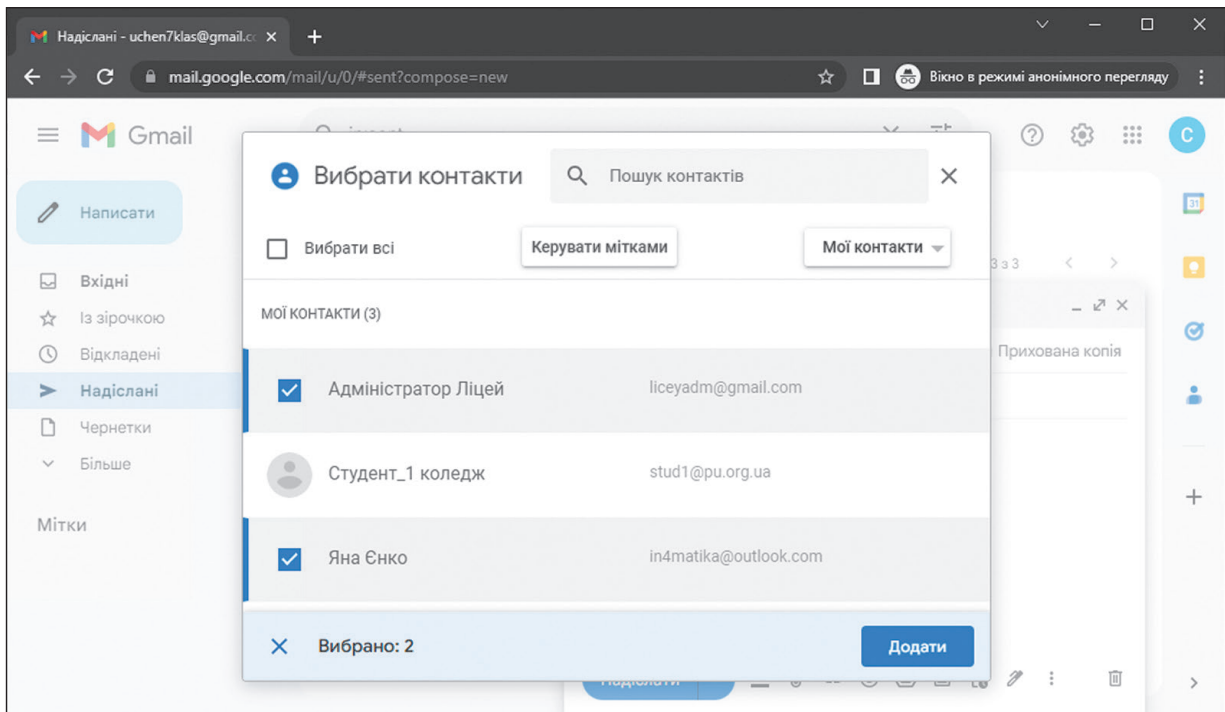
Щоб додати до адресної книги дані користувача, з яким ще не було листування, слід:

1. Відкрити електронну поштову скриньку.
2. Вибрати на правій бічній панелі кнопку **Контакти** .
3. Вибрати у вікні **Контакти**, що відкрилося, кнопку **Створити контакт**.
4. Увести дані нового користувача.
5. Вибрати кнопку **Зберегти**.

Використовуючи список **Мої контакти**, можна швидко вводити адреси одержувачів під час створення електронних листів. Для цього потрібно в заголовку нового електронного листа вибрати посилання **Кому**. Відкриється вікно **Вибрати контакти** зі списком **Мої контакти** (мал. 1.22).

У цьому вікні потрібно встановити позначку одного або кількох прапорців у рядках з даними осіб, яким ви плануєте надіслати лист. За вибору кнопки **Додати** вибрані адреси буде уведено в поле **Кому**, і лист з одним і тим самим умістом ви надішлете всім вибраним адресатам.

Також під час введення будь-яких літер у поле **Кому** у спадному списку буде запропоновано адреси з **Контактів**, які починаються з цих літер. Для прискорення введення адреси електронної поштової скриньки її можна вибрати з цього списку.



Мал. 1.22. Вікно **Вибрати контакти** в поштовій службі **Gmail**

СПИСКИ РОЗСИЛАННЯ



Поміркуйте

• Для чого може бути корисним надсилання листів однакового вмісту кільком адресатам?

Контакти можна об'єднати у групи для впорядкування списку контактів, швидкого пошуку потрібних адресатів і швидкої підготовки листів однакового вмісту групі адресатів. Групи контактів також називають **списками розсилання**.

Групам контактів надають імена. Для цього групи позначають мітками. Кожен контакт може входити до кількох різних груп контактів.

Для створення нової групи контактів слід:

1. Відкрити вікно **Вибрати контакти**.
2. Установити позначки прапорців поруч з даними тих контактів, які ви плануєте включити до списку розсилання.

3. Вибрати кнопку **Керувати мітками**.
4. Вибрати команду **Створити мітку** (мал. 1.23).
5. Увести ім'я, що буде іменем нової групи контактів.
6. Вибрати кнопку **Зберегти**.
7. Закрити вікно **Вибрати контакти**.

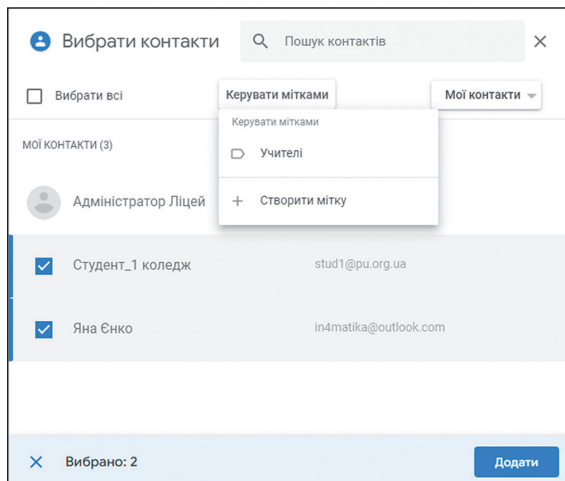
Щоб додати контакт до існуючої групи, слід:

1. Установити у вікні **Вибрати контакти** позначку прапорця в рядку контакту.
2. Вибрати кнопку **Керувати мітками**. Після цього відкриється список імен груп, у якому будуть позначені ті групи, у яких уже перебуває вибраний контакт.
3. Вибрати ім'я групи, до якої додається контакт.
4. Вибрати кнопку **Застосувати**.

Для створення листа на адресу всіх членів деякої групи слід:

1. Вибрати кнопку **Написати**.
2. Вибрати посилання **Кому** в заголовку листа.
3. Вибрати у вікні **Вибрати контакти** ім'я потрібної групи у списку кнопки **Мої контакти**.
4. Установити позначку прапорця **Вибрати всі**, який розміщено вище від списку контактів. У результаті буде встановлено позначки прапорців поруч з іменами всіх контактів групи.
5. Вибрати посилання **Додати**.

Після цього у вікні створення нового електронного листа в поле **Кому** будуть уведені адреси всіх членів групи.



Мал. 1.23. Створення групи контактів

ЕТИКЕТ ЕЛЕКТРОННОГО ЛИСТУВАННЯ



Пригадайте

• Для чого зазначають тему в електронних листах? • Яких правил культури поведінки ви дотримуетесь у спілкуванні з людьми?

Електронна пошта – засіб спілкування людей, тому вона передбачає дотримання відповідних правил увічливості. Деякі з них не відрізняються від загальноприйнятих норм людського спілкування:

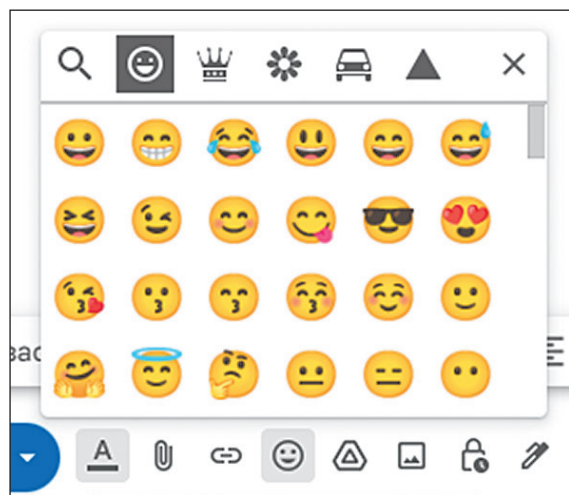
Етикет (фр. *étiquette* – етикетка, напис) – норми й правила гідної поведінки людей у суспільстві.

- починайте текст листа з привітання, завершуйте підписом;
- під час звертання до людини, з якою ви особисто не знайомі, назвіть себе;
- не забувайте вживати слова «будь ласка», якщо звертаєтесь до кого-небудь із проханням, подякуйте, якщо хтось допомагає вам;
- намагайтеся уникати фраз, які можуть спричинити конфлікт;
- перевіряйте текст на відсутність граматичних помилок.

Окремі правила пов'язано з особливостями спілкування у мережі:

- зазначайте тему свого листа: часто отримувач орієнтується саме на тему, коли переглядає пошту та обирає листи для читання;
- не пишіть весь текст листа великими літерами – його важко читати, це сприймається як крик;
- не додавайте без потреби вкладені файли до листа; повідомляйте адресатові, які файли додаєте до листа, бо часто через укладені файли розповсюджуються комп'ютерні віруси.

Електронна пошта позбавлена можливості передавати міміку та жести, які під час особистого спілкування виражають



Мал. 1.24. Колекція графічних смайлів поштової служби **Gmail**

ваше ставлення до предмета обговорення. Для передавання емоцій в Інтернеті використовують **смайли** (англ. *smile* – посмішка) – послідовності символів, що нагадують обличчя. Наприклад, послідовність символів :-) нагадує обличчя, що посміхається, а послідовність :- (– сумне обличчя. Правильне використання смайлів додає листу настрій, робить його більш емоційним.

До тексту листа можна вставляти і графічні смайли, або **емодзі** (яп. 絵 – картинка, 文字 – знак, символ). Для цього потрібно вибрати кнопку **Вставити смайли** 😊 у вікні створення нового листа та вибрати потрібне зображення з колекції (мал. 1.24).

Колекцію поділено на групи: **Емоції** 😊, **Смайли-предмети** 👑, **Природа** 🌸, **Транспорт** 🚗, **Символи** ▲. Потрібне зображення можна шукати за ключовими словами, вибравши кнопку **Пошук смайлів** 🔍.

ПРАВИЛА БЕЗПЕЧНОГО ЕЛЕКТРОННОГО ЛИСТУВАННЯ



Поміркуйте

- Чому потрібно обережно ставитися до спілкування в Інтернеті?
- На які ризики можна наразитися, використовуючи електронну пошту?

Спілкуючись в Інтернеті, потрібно пам'ятати про можливі небезпеки, пов'язані з особливостями роботи в мережі. Як і під час пошуку інформації в Інтернеті, під час електронного листування потрібно дбати про захист **особистих даних**, захист **від шкідливих програм** і **від загрозливого вмісту**.

Якщо відправник листа вам невідомий, то ви не можете перевірити правдивість його слів і справжні наміри. Крім того, листи навіть з адреси знайомої людини можуть бути надіслані шкідливою програмою, якщо її комп'ютер інфіковано. А тому варто критично ставитися до вмісту всіх отриманих листів.

Однією з проблем, пов'язаних з електронною поштою, є **спам** – небажані для одержувачів електронні листи, що масово розсилаються користувачам електронної пошти. Часто такі листи є рекламними, з намаганням привернути вашу увагу до певної продукції або послуг. Часом це реклама незаконних або заборонених товарів.

Іноді надходять шахрайські листи про начебто отримання вами спадку або грошового виграшу. Відправник такого листа пропонує вам допомогу в отриманні спадку чи коштів за певну винагороду для себе. Ця винагорода і є метою шахраїв.

Ще один вид шахрайства отримав назву **фішинг** (англ. *phishing* – фішинг, звучить як *fishing* – рибальство). Він полягає в намаганні отримати від вас («виудити» у вас) коди доступу до ваших банківських систем і платіжних карток. У фішинговому листі може міститися пропозиція підтвердити ваш обліковий запис на певному інтернет-ресурсі, але за посиланням у листі ви потрапите на фішинговий сайт, який копіює вигляд справжнього сайту банківської установи або магазину. Таким способом шахраї намагаються отримати логіни та паролі користувачів до різних ресурсів Інтернету.

Під час листування електронною поштою рекомендується дотримуватися таких правил безпеки:

- ніколи не наводьте в листах приватні дані про себе та своїх близьких, якщо особисто не знайомі з адресатом;
- ніколи не погоджуйтеся на особисту зустріч з особами, знайомими вам тільки через електронне листування, – вони можуть виявитися не тими, за кого себе видавали;
- не відповідайте на рекламні листи, що приходять від невідомих осіб; відповідаючи на ці листи, ви підтверджуєте існування адреси, що дозволить надсилати і надалі непотрібні вам листи зі спамом;
- не відкривайте вкладені файли, отримані в листах від невідомих осіб: у них можуть бути шкідливі програми; такі листи рекомендується видаляти;
- не повідомляйте пароль від електронної поштової скриньки стороннім особам, щоб ваше листування не потрапило в чужі руки та щоб ніхто не міг відправити листа від вашого імені;
- завжди завершуйте роботу зі своїм обліковим записом, якщо працювали не на власному комп'ютері;
- періодично змінюйте пароль доступу до вашої електронної поштової скриньки.



Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви можете знайти за адресою <https://cutt.ly/bwNTqgbV> або QR-кодом.



Найважливіше в цьому пункті

Адресна книга призначена для зберігання **контактів** – даних про осіб, з якими листується користувач: імен, адрес електронної пошти, інших особистих даних адресатів. Використовуючи адресну книгу,

можна швидко вводити адреси в поле **Кому** під час створення електронних листів.

Контакти в адресній книзі можна об'єднати у групи для швидкої підготовки листів однакового змісту для групи адресатів. Групи контактів також називають **списками розсилання**.

Електронна пошта передбачає дотримання правил етикету електронного листування. Спілкуючись в Інтернеті, потрібно пам'ятати про можливі небезпеки, пов'язані з особливостями роботи в мережі, та дотримуватися правил безпечного електронного листування.

Дайте відповіді на запитання

1. Для чого призначена адресна книга? Як переглянути її вміст?
2. Як додати дані з адресної книги **Мої контакти** до листа, що створюється?
3. Для чого призначений список розсилання? Як його створити? Як надіслати лист групі контактів?
4. Як додати емоційності електронному листуванню?
5. Яких правил безпеки слід дотримуватися під час електронного листування?

Обговоріть і зробіть висновки


1. Які переваги надає використання списків розсилання порівняно з надсиланням листів без використання цих списків?
2. Дотримання яких правил безпечного листування допоможе захистити ваші персональні дані?
3. Дотримання яких правил безпечного листування допоможе захистити ваш комп'ютер від зараження шкідливими програмами?

Виконайте завдання

1. Створіть і надішліть електронний лист на адреси вчителя/вчительки інформатики та трьох учнів/учениць, використавши адресну книгу **Мої контакти**. Тема листа *Спорт*. Уміст листа –

привітання, назви трьох олімпійських видів спорту, які ви знаєте, ваш підпис.


2. Створіть в адресній книзі **Мої контакти** групу контактів *Учні та учениці*. Включіть до групи контакти трьох учнів/учениць. Надішліть усім контактам з групи *Учні та учениці* та вчителю/вчительці інформатики лист з темою *Музика*. Уміст листа – привітання, три відомі вам жанри музики, ваш підпис. У тексті листа використайте графічні смайли.

 3. Створіть презентацію, описавши в ній відомі вам правила етикету електронного листування, поясніть потребу в їх дотриманні. Збережіть презентацію у файлі з іменем **завдання 1.4.3.pptx**. Надішліть файл електронною поштою вчителю/вчительці інформатики.

4. Визначте правила етикету листування, які порушено в такому листі:

ПРИВІТ! МЕНІ ПОТРІБНО ПІДГОТУВАТИ ПОВІДОМЛЕННЯ ПРО ІСТОРІЮ УКРАЇНСЬКОГО КОЗАЦТВА. ТИ ХВАЛИВСЯ КОЗАКАМИ З ТВОГО РОДУ. ТЕРМІНОВО НАПИШИ МЕНІ ЇХ ІСТОРІЮ. НАДСИЛАЮ ТОБІ ТРИ ФОТОГРАФІЇ – ПОДИВИСЬ, ДЕ Я ВІДПОЧИВАВ. БУВАЙ!

Створіть і надішліть лист вчителю/вчительці інформатики з описом визначених порушень.

 5. Створіть презентацію, описавши в ній відомі вам правила безпечного електронного листування та ризику, які вимагають дотримання цих правил. Збережіть презентацію у файлі з іменем **завдання 1.4.5.pptx**. Надішліть файл електронною поштою вчителю/вчительці інформатики.

6. Визначте правила безпеки, які порушено в листі з таким умістом:

Доброго дня, невідомий друже! До мене випадково потрапив твій лист з розповіддю про можливість поїздки на відпо-

чинок до моря. Мене звать Іванов Сергій. Я живу в Харкові, навчаюсь у 7 класі школи № 1024, мені 13 років. Я дуже хочу поїхати на море. Гроші для оплати я знайду – мій батько працює начальником відділення Новобанку, у нього висока зарплатня. Готовий зустрітися з тобою особисто та обговорити деталі поїздки. До зустрічі!

Створіть і надішліть лист учителю/вчительці інформатики з описом визначених порушень.



Практична робота № 1.

«Пошук в Інтернеті. Електронне листування»

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

1. Створіть текстовий документ, у якому опишіть у 2–3 реченнях традиції, пов'язані з вашим улюбленим народним святом. Збережіть документ у вашій папці. Ім'я файлу – *ваше прізвище*.
2. Знайдіть в Інтернеті та збережіть у вашій папці *анімоване* зображення *середнього* розміру, розміщене в Інтернеті за *останній місяць*, що пов'язане з вашим улюбленим святом.
3. Створіть в адресній книзі **Мої контакти** групу *Мій клас*, включіть у групу контакти вчителя/вчительки інформатики та двох учнів/учениць.
4. Створіть і надішліть усім членам групи *Мій клас* лист з темою *Практична робота 1*. У тексті листа наведіть список групи *Мій клас*. Додайте до листа створений текстовий документ і знайдене зображення.
5. Відкрийте та прочитайте лист, отриманий вами від учителя/вчительки інформатики. Збережіть файл, доданий до листа, у вашій папці. Створіть і надішліть відповідь на лист, додайте до тексту відповіді графічний смайл.
6. Продемонструйте результат учителю/вчительці.



Готуємось до вивчення нового матеріалу

1. Які інтернет-ресурси ви використовуєте, готуючись до уроків з різних предметів?
2. У чому переваги та недоліки використання онлайн-додатків порівняно з установленим аналогічним додатком на носії даних власного комп'ютера?



1.5. Хмарні сервіси

У цьому пункті йтиметься про:

- ▶ поняття «хмарні сервіси», їх переваги та недоліки;
- ▶ використання онлайн-перекладачів;
- ▶ хмарні сервіси **Google**;
- ▶ зберігання та впорядкування файлів у хмарному сховищі **Google Диск**;
- ▶ синхронізацію файлів.

ХМАРНІ СЕРВІСИ



Поміркуйте

• Які ресурси Інтернету можуть бути корисними для вашого навчання? У чому може бути їх користь? • Які дії, крім перегляду матеріалів, ви виконували на сайтах?

Вам вже доводилося на різних сайтах в Інтернеті читати, переглядати та прослуховувати інформаційні матеріали, спілкуватися, проходити тестування, брати участь в олімпіадах з різних предметів, проводити комп'ютерні експерименти, грати в комп'ютерні ігри тощо. Для цього на вашому комп'ютерному пристрої повинен бути встановлений лише браузер. А дані, які опрацьовуються, та потрібні для цього програми зберігаються на самих сайтах.

Інформаційно-комунікаційні технології, які надають засоби для віддаленого опрацювання та зберігання даних, називаються

вають **хмарними технологіями** (англ. *cloud technologies* – хмарні технології).

Засобами хмарних технологій створено **хмарні сервіси** (мал. 1.25), які надають користувачам Інтернету різноманітні послуги:



Мал. 1.25. Види хмарних сервісів

- онлайн-програми (текстові процесори, редактори презентацій та електронних таблиць, графічні та відеоредактори та переглядачі, калькулятори, перекладачі тощо) без установлення цих програм на свої комп'ютери;
- онлайн-сховища для зберігання файлів (текстових документів, презентацій, електронних таблиць, фотографій, малюнків, відеороликів тощо) і доступу до них з будь-якого комп'ютера, який має підключення до Інтернету;
- зберігання власних списків контактів і закладок на потрібні ресурси Інтернету для доступу до них з будь-яких комп'ютерних пристроїв;
- визначення місцезнаходження користувача та побудова маршрутів переміщення на онлайн-картах та інші.

Переваги використання хмарних сервісів:

- можливість зберігати власні дані в онлайн-сховищі для економії місця на власних носіях даних і захисту даних від втрат у випадку збою на комп'ютерному пристрої;
- доступність даних і програм для їх опрацювання з будь-якого комп'ютерного пристрою незалежно від установленної операційної системи;
- автоматичне оновлення програм для опрацювання даних, що забезпечує користувачу доступ до останньої версії програмного забезпечення;
- можливість спільно використовувати файли (редагувати, копіювати, пересилати тощо)

та інші.

Недоліками хмарних сервісів є потреба у постійному доступі до Інтернету, можливість змінення власниками ресурсів умов щодо використання сервісів (підвищення плати за користування, змінення обсягу хмарного сховища, відключення певних послуг та інше), можливість витоку персональних даних користувачів тощо.

Хмарні послуги можна отримати на багатьох інтернет-ресурсах. Для доступу до деяких хмарних сервісів потрібно мати обліковий запис на відповідному ресурсі, інші можуть бути доступні без реєстрації.

ОНЛАЙН-ПЕРЕКЛАДАЧІ



Поміркуйте

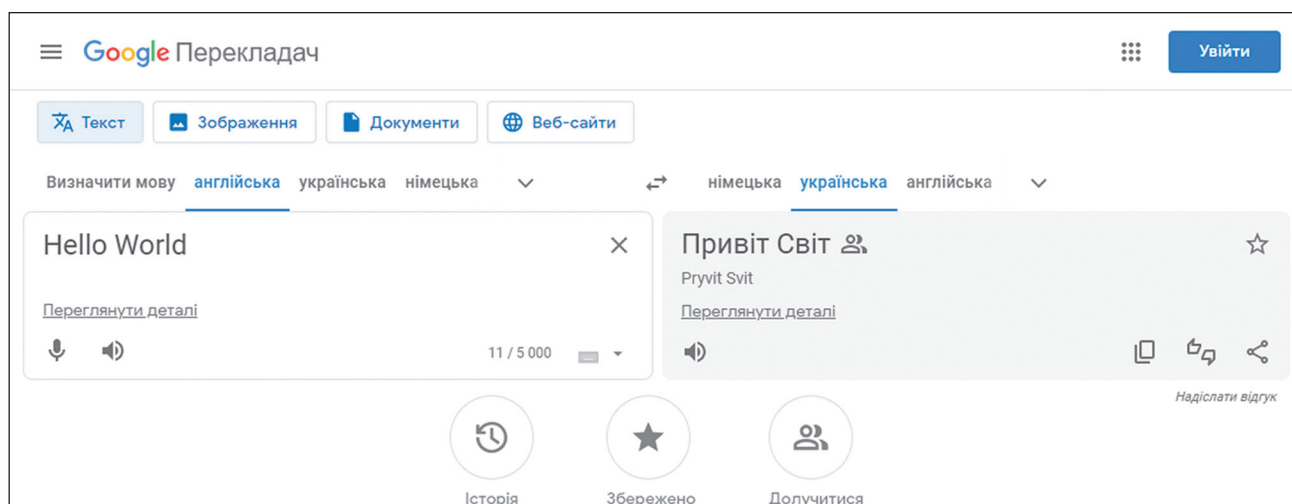
- Які засоби ви використовуєте, коли виконуєте завдання з іноземної мови?
- У чому полягає складність перекладу речень з однієї мови іншою?

Одним з популярних хмарних сервісів є сервіс перекладу текстів. Для автоматизованого перекладу текстів з однієї мови іншою розроблено спеціальні програми – **онлайн-**


перекладачі. Їх робота базується на використанні електронних словників. В онлайн-перекладачах реалізовано алгоритми для узгодження слів у отриманому тексті відповідно до правил мови, якою здійснюється переклад.


Популярними онлайн-перекладачами є, наприклад, **Google Перекладач** (translate.google.com), **Microsoft Bing Translator** (bing.com/translator), **МЕТА Перекладач** (translate.meta.ua), **DeepL Translate** (deepl.com) та інші.




Розглянемо використання онлайн-перекладача **Google Перекладач** (мал. 1.26), доступного за адресою translate.google.com. Користуватися ним можна без реєстрації облікового запису.



Мал. 1.26. Сторінка **Google Перекладача**

На сторінці **Google Перекладача** можна вибрати з указаних на сторінці або у списку  будь-які мови та виконати переклад:





- у режимі **Текст**  – тексту завдовжки до 5000 символів, уведеного з клавіатури або вставленого з **Буфера обміну**;

- у режимі **Зображення**  – тексту із зображення у графічному файлі формату **jpg, jpeg** або **png**;
- у режимі **Документи**  – тексту з документа формату **docx, pdf, pptx** або **xlsx**;
- у режимі **Вебсайти**  – тексту на вебсторінці з указаною URL-адресою.

Крім того, користувачам **Google Перекладача** доступні інші можливості з використанням елементів керування, призначення яких наведено в таблиці 1.2.

Таблиця 1.2

Призначення елементів керування на сторінці
Google Перекладача

<i>Кнопка</i>	<i>Назва</i>	<i>Призначення</i>
	Перекладати голосовий ввід	Для голосового введення тексту з використанням мікрофона
	Прослухати	Для прослуховування тексту мовою оригіналу або мовою перекладу
	Копіювати переклад	Для копіювання тексту перекладу в Буфер обміну
	Поділитися перекладом	Для надсилання перекладу іншим користувачам

ХМАРНІ СЕРВІСИ GOOGLE

Пригадайте

- Якими сервісами **Google** ви вже користувалися?
- Користування якими з цих сервісів не потребувало введення даних облікового запису?





Популярними в Україні є хмарні сервіси компанії **Google**. Ви вже користувалися такими з них, як **Google Пошук**, **Gmail**, **Google Перекладач**. Можливо, у вашому закладі освіти для організації навчання використовують сервіс **Google Клас**, а для проведення уроків онлайн – сервіс відеозв'язку **Google Meet** (англ. *meet* – зустріти). Обліковий запис, створений для користування будь-яким із сервісів **Google**, буде надавати доступ і до всіх інших сервісів.

У таблиці 1.3 наведено приклади деяких хмарних сервісів **Google**.


Таблиця 1.3

Приклади хмарних сервісів Google

Значок	Назва	Призначення
 Пошук	Google Пошук	Для пошуку відомостей в Інтернеті
 Gmail	Gmail	Для електронного листування
 Переклад...	Google Перекладач	Для перекладу текстів, файлів, вебсайтів з однієї мови іншою
 Клас	Google Клас	Для забезпечення взаємодії вчителів/вчительок і учнів/учениць під час освітнього процесу
 Meet	Google Meet	Для організації відеоконференцій
 Диск	Google Диск	Для впорядкованого зберігання та спільного використання файлів у хмарному сховищі
 Контакти	Google Контакти	Для зберігання даних про контакти користувача

Значок	Назва	Призначення
	YouTube	Для перегляду, розміщення та редагування відео
	Google Карти	Для перегляду онлайн-карт, прокладання маршрутів, позначення та опису місць на картах
	Google Календар	Для позначення та планування подій
	Google Фото	Для зберігання та редагування фотографій

Для доступу до цих та інших хмарних сервісів **Google** слід:

1. Відкрити у вікні браузера головну сторінку сайту **Google** (google.com).
2. Вибрати кнопку **Додатки Google**  у верхньому правому куті сторінки.
3. Вибрати у списку, що відкрився, значок відповідного сервісу.
4. Увести дані власного облікового запису, якщо їх не було введено попередньо та вони потрібні для роботи із сервісом.

Користування деякими із сервісів (**Google Пошук**, **Google Карти**, **Google Перекладач**, **YouTube** тощо) можливе без уведення даних облікового запису.

ХМАРНЕ СХОВИЩЕ GOOGLE ДИСК



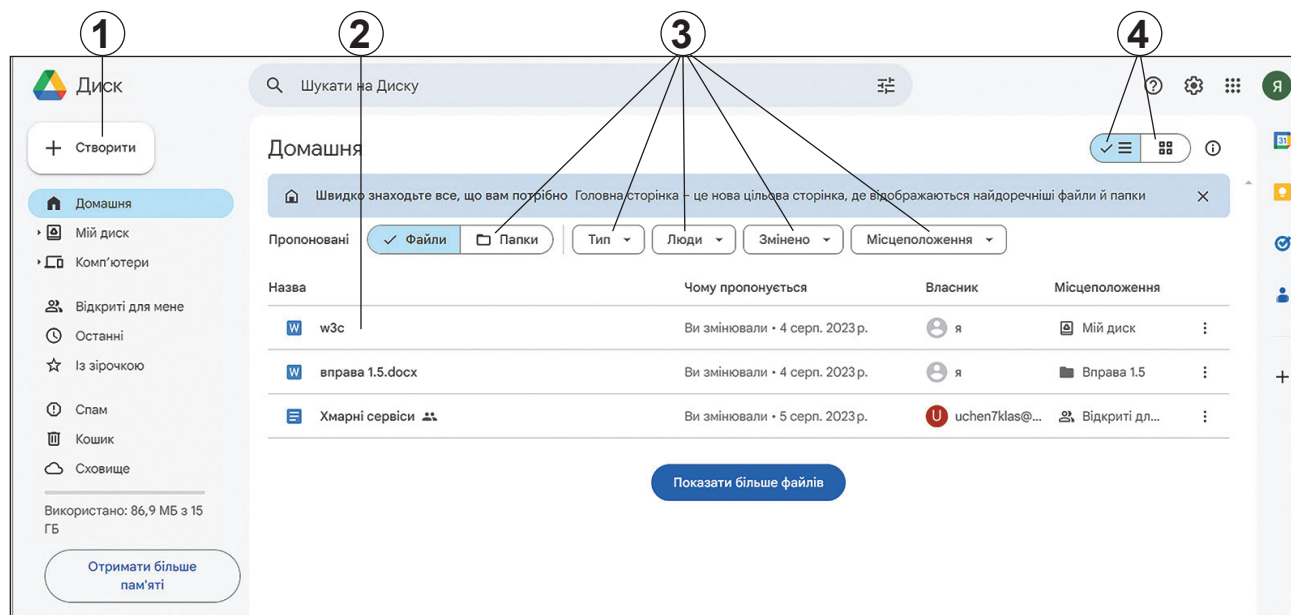
Пригадайте

- Які операції над файлами та папками можна виконувати в операційній системі **Windows**?
- За якими ознаками можна впорядковувати файли на носіях даних?

Google Диск  – це хмарний сервіс **Google**, призначений для впорядкованого зберігання файлів і надання до них доступу іншим користувачам. Такий вид сервісу називають **хмарним сховищем даних**.

Коли користувач реєструє обліковий запис **Google**, йому автоматично надається безкоштовне хмарне сховище обсягом 15 ГБ.

Вигляд домашньої сторінки **Google Диска** наведено на малюнку 1.27.

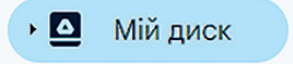




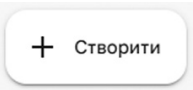
Мал. 1.27. Вигляд домашньої сторінки **Google Диска**:

- 1 – кнопка **Створити**;
- 2 – файли з розділу **Пропоновані**;
- 3 – кнопки та списки для вибору об'єктів;
- 4 – кнопки переключення режимів відображення об'єктів

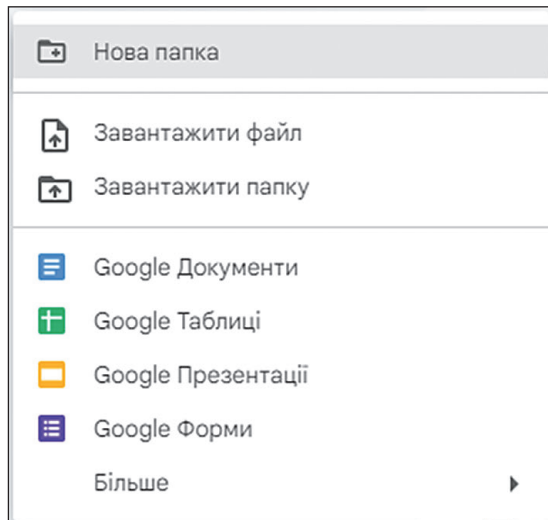
У центральній частині домашньої сторінки **Google Диска** відображаються імена кількох останніх файлів, з якими працював користувач. Завдяки цьому можна швидко знайти та відкрити подвійним клацанням пропоновані файли для продовження роботи з ними.

Вибравши в лівій бічній панелі кнопку **Мій диск**

, можна побачити список усіх об'єктів, що зберігаються на **Google Диску** користувача, – файлів, папок та ярликів. Вигляд списку можна змінити:

- після вибору кнопки **У вигляді сітки**  об'єкти **Google Диска** відображатимуться разом з ескізами їх вмісту;
- після вибору кнопки **У вигляді списку**  про кожен об'єкт **Google Диска** буде виведено відомості: значок, ім'я, власник, дата або час останніх змін, розмір файлу. На малюнку 1.27 об'єкти **Google Диска** виведено у вигляді списку. Щоб створити папку або файл на **Google Диску** або завантажити існуючі з інших носіїв даних, потрібно використати кнопку **Створити**  та в її меню вибрати відповідну команду (мал. 1.28).

Для завантаження файлу або папки з носія даних на **Google Диск** також можна перетягнути вибрані об'єкти з вікна **Провідника** операційної системи **Windows** безпосередньо на сторінку **Google Диска** або на сторінку однієї з папок.



Мал. 1.28. Меню кнопки **Створити** на **Google Диску**

Як і в операційній системі **Windows**, з об'єктами на **Google Диску** можна виконувати різні операції: відкривати для перегляду або змінення у відповідній онлайн-програмі, створювати копії, перейменовувати, переміщувати, переглядати значення властивостей

файлів і папок, видаляти в **Кошик**. Також об'єкти з **Google Диска** можна зберегти на інші носії даних.

Для виконання цих операцій потрібно вибрати об'єкт і використати елементи керування на **панелі інструментів**, що з'являється у верхній частині сторінки (мал. 1.29).



Мал. 1.29. Панель інструментів вибраного об'єкта


Призначення деяких елементів керування для виконання операцій над об'єктами на **Google Диску** наведено в таблиці 1.4.

Таблиця 1.4

Елементи керування панелі інструментів вибраного об'єкта

Значок	Назва	Призначення
	Скасувати вибір	Для відмови від вибору об'єкта
	Завантажити	Для збереження вибраного об'єкта на іншому носії даних
	Перемістити	Для переміщення вибраного об'єкта в іншу папку
	Перемістити в кошик	Для видалення об'єкта в Кошик
	Додаткові дії	Для відкриття меню з командами виконання інших операцій над об'єктом: <ul style="list-style-type: none"> ● Відкрити за допомогою ● Перейменувати ● Зробити копію ● Інформація про файл

Також операції над об'єктами на **Google Диску** можна виконати, використовуючи відповідні команди контекстного меню цього об'єкта.

Як і в операційній системі **Windows**, об'єкти, видалені з **Google Диска**, тимчасово зберігаються в **Кошику**. Вони автоматично видаляються назавжди через 30 днів після їх переміщення в **Кошик**. До цього часу їх можна відновити в попередньому розташуванні або видалити назавжди. Щоб повністю очистити **Кошик**, потрібно вибрати на лівій бічній панелі сторінки **Google Диска** кнопку **Кошик**  **Кошик**, а потім посилання **Очистити кошик**.

СИНХРОНІЗАЦІЯ ФАЙЛІВ




Поміркуйте

- Якими способами можна перенести файли з одного носія даних на інший?
- Як визначити, який з файлів з однаковими іменами, що зберігаються на різних носіях даних, новіший?


Хмарне сховище часто використовують як віртуальну флешку, яку не потрібно носити із собою. Якщо необхідно, щоб на вашому комп'ютері та у хмарному сховищі завжди зберігалася остання версія ваших файлів, навіть якщо вам іноді доводиться працювати з ними без доступу до мережі, то можна налаштувати *синхронізацію файлів*.

Синхронізація файлів – це процес відстеження змін файлів у двох або більше місцях їх зберігання та автоматичного їх оновлення до останньої версії.

Наприклад, у деякій папці на носії даних вашого комп'ютера ви видалили або змінили файли – такі самі дії відбудуться і з відповідними об'єктами хмарного сховища після синхронізації.

Для налаштування синхронізації файлів потрібно встановити на своєму комп'ютері програму **Диск для комп'ютера**, яка і буде відслідковувати зміни у визначених папках локального та хмарного дисків. Щоб завантажити цю програму, потрібно вибрати на сторінці **Google Диска** в меню кнопки **Налаштування**  команду **Завантажити Диск для комп'ютера**.

Після встановлення цієї програми потрібно її налаштувати: вибрати папки на комп'ютері, які синхронізуватимуться з **Google Диском**, та встановити значення інших властивостей.

У хмарному сховищі синхронізовані папки зберігатимуться в розділі **Комп'ютери**, для перегляду якого потрібно вибрати кнопку  **Комп'ютери** в лівій бічній панелі сторінки **Google Диска**.

Звертаємо вашу увагу!

Користувачі, які мають обліковий запис **Google**, можуть синхронізувати не лише файли та папки на дисках, а також закладки браузера на корисні ресурси Інтернету, контакти користувача та інші дані.

Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви можете знайти за адресою <https://cutt.ly/TwNTqTI9> або QR-кодом.



Найважливіше в цьому пункті

Інформаційно-комунікаційні технології, які надають засоби для віддаленого опрацювання та зберігання даних, називають **хмарними технологіями**. Засобами хмарних технологій створено **хмарні сервіси**, які надають користувачам Інтернету різноманітні послуги.

Для автоматизованого перекладу текстів з однієї мови іншою розроблено спеціальні програми – **онлайн-перекладачі**, наприклад **Google Перекладач**.

Google Диск – це хмарний сервіс **Google**, призначений для впорядкованого зберігання файлів і надання до них доступу іншим користувачам. Над файлами та папками на **Google Диску** можна виконувати різні операції: відкривати для перегляду або змінення у відповідній онлайн-програмі, створювати копії, перейменовувати, переміщувати, переглядати значення властивостей файлів і папок, видаляти в **Кошик**. Також об'єкти з **Google Диска** можна зберегти на інші носії даних.

Синхронізація файлів – це процес відстеження змін файлів у двох або більше місцях їх зберігання та автоматичного їх оновлення до останньої версії.

Дайте відповіді на запитання

1. Які ви знаєте хмарні сервіси? Які переваги та недоліки використання хмарних сервісів?
2. Як отримати доступ до хмарних сервісів **Google**?
3. Для чого призначено хмарний сервіс **Google Диск**?
4. Які операції над папками та файлами можна виконувати на **Google Диску**?

Обговоріть і зробіть висновки


1. Які хмарні сервіси та з якою метою можуть бути використані під час планування туристичного походу?
2. У чому схожість і відмінність виконання операцій над файлами та папками в операційній системі **Windows** і на **Google Диску**?
3. У чому зручність автоматичної синхронізації файлів? Які можуть бути ризики її налаштування?

Виконайте завдання

1. Перекладіть, використовуючи **Google Перекладач**, з української мови англійською термін *хмарне сховище*. Прислухайте, як вимовляються ці слова англійською. Поділіться перекладом електронною поштою з вашим учителем/вчителькою інформатики.


2. Перекладіть українською мовою сторінку **Google Cloud** з адресою cloud.google.com. Запишіть у зошит перекладений заголовок отриманої сторінки та відомості про кількість хмарних сервісів **Google**.

3. Перекладіть українською мовою з англійської текст, розміщений на зображенні з файлу **Розділ 1\Пункт 1.5\завдання 1.5.3.png**. Збережіть отриманий графічний файл з перекладом у вашій папці.


 4. **Працюємо в парах.** Об'єднайтеся у пари. Порівняйте особливості роботи онлайн-перекладачів **Google Перекладач** (translate.google.com) і **DeepL Translate** (deepl.com). Обговоріть переваги та недоліки кожного з онлайн-перекладачів. Створіть текстовий документ з описом переваг і недоліків. Збережіть документ у вашій папці у файлі **завдання 1.5.4**. Надішліть документ електронною поштою вашому вчителю/вчительці інформатики.

5. Створіть на вашому **Google Диску** папку **Завдання 1.5.5** і завантажте до неї файл, наприклад **Розділ 1\Пункт 1.5\завдання 1.5.5.pptx** з вашого комп'ютера. Створіть копію завантаженого файлу та перейменуйте скопійований файл. Нове ім'я файлу **stopfake**.

6. Завантажте на ваш **Google Диск** папку, наприклад **Розділ 1\Пункт 1.5\Завдання 1.5.6** з вашого комп'ютера. Перемістіть файл **Google.jpg** із завантаженої папки **Завдання 1.5.6** у кореневу папку **Google Диска**.

 7. Порівняйте дослідним шляхом способи виконання операцій над групами файлів на **Google Диску** та в операційній системі **Windows**. Створіть текстовий документ і опишіть у ньому, якими способами можна виділити групи файлів на **Google Диску** та в операційній системі **Windows**, яким буде результат виконання операцій копіювання, переміщення,

видалення. Збережіть документ у вашій папці у файлі з іменем **завдання 1.5.7**. Завантажте документ на **Google Диск**.

 **8.** Дослідіть особливості використання онлайн-калькуляторів на сайті **OnlineMSchool** (ua.onlinemschool.com). Створіть комп'ютерну презентацію про види онлайн-калькуляторів і правила користування ними. Збережіть презентацію у вашій папці у файлі з іменем **завдання 1.5.8**. Завантажте презентацію на **Google Диск**.

Готуємось до вивчення нового матеріалу

1. Які програми з пакета програм **Microsoft Office** ви використовували?
2. Які засоби ви використовуєте під час групової роботи над проектами для обміну матеріалами та спілкування?

1.6. Створення онлайн-документів і керування доступом до них

У цьому пункті йтиметься про:

- ▶ офісні онлайн-додатки;
- ▶ створення онлайн-документів різних видів;
- ▶ налаштування спільного доступу до об'єктів у хмарному сховищі;
- ▶ рівні доступу до спільного документа.




ОФІСНІ ОНЛАЙН-ДОДАТКИ

Пригадайте



- Які програми з пакета програм **Microsoft Office** ви використовували?
- Які електронні документи можна створювати з їх використанням?

Популярними хмарними сервісами є так звані **офісні онлайн-додатки**, які призначені для створення та опрацювання онлайн-документів різних видів: текстових документів, електронних таблиць, комп'ютерних презентацій, графічних

зображень та іншого. У пакеті хмарних сервісів **Google** є, наприклад, такі офісні онлайн-додатки:

-  **Google Документи** – текстовий процесор;
-  **Google Таблиці** – редактор електронних таблиць;
-  **Google Презентації** – редактор презентацій

та інші.

Для створення онлайн-документа можна вибрати в меню кнопки **Додатки Google**  значок відповідного сервісу та кнопку **Створити**  на сторінці, що відкриється.

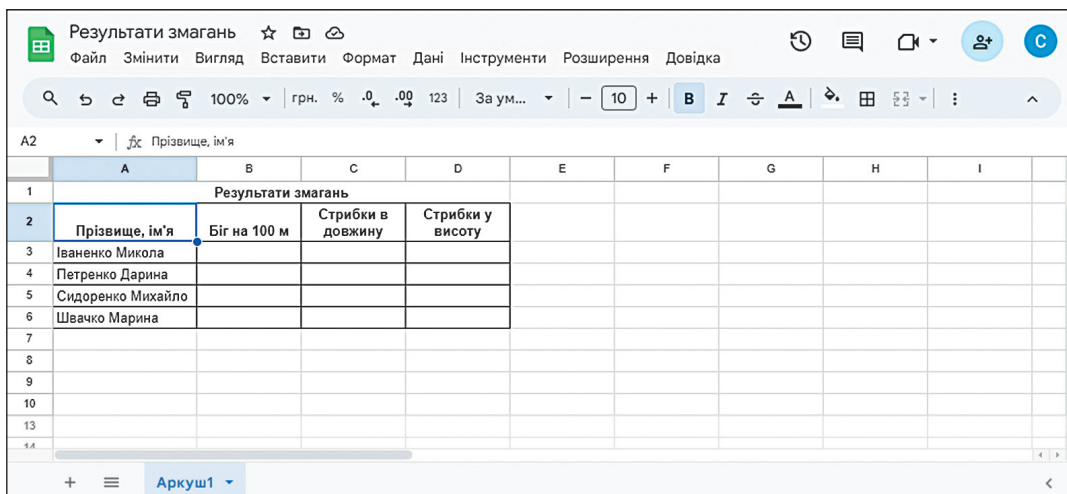
Іншим способом створення онлайн-документів є вибір відповідних команд у меню кнопки **Створити** на сторінці **Google Диска**.

Сторінки офісних онлайн-додатків (мал. 1.30–1.32) мають схожі елементи:

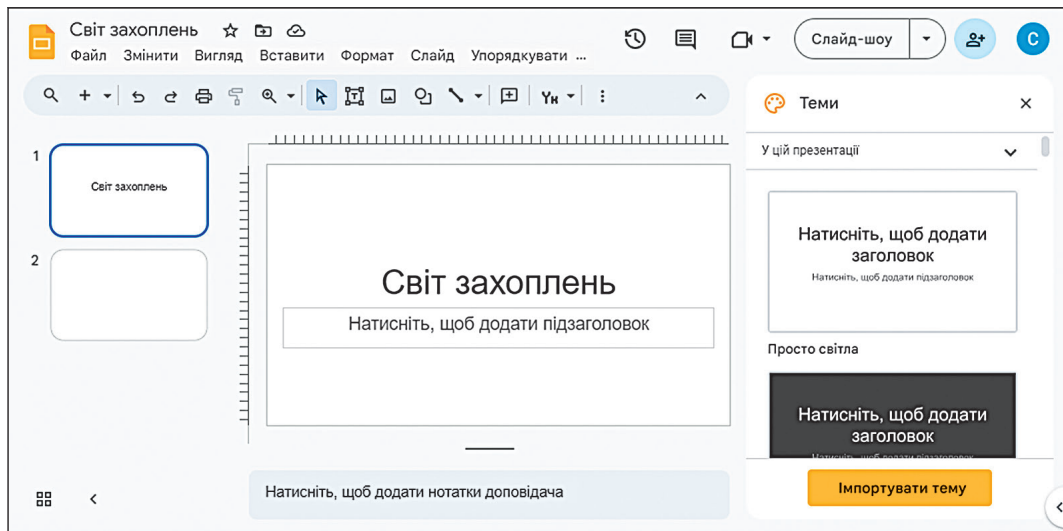
- **рядок заголовка** зі значком програми та іменем файлу;
- **рядок меню** з командами **Файл**, **Змінити**, **Вигляд**, **Вставити**, **Формат** та іншими;



Мал. 1.30. Сторінка онлайн-додатка **Google Документи**



Мал. 1.31. Сторінка онлайн-додатка **Google Таблиці**



Мал. 1.32. Сторінка онлайн-додатка **Google Презентації**

- **панель інструментів** з елементами керування. Ці елементи керування подібні до тих, що мають у програмах пакета **Microsoft Office** аналогічне призначення;
- **робоча область** для введення даних і вставлення об'єктів. Основні алгоритми зі створення й опрацювання документів з використанням онлайн-додатків майже не відрізняються від алгоритмів, які ви опанували під час роботи у програмах аналогічного призначення з пакета **Microsoft Office** або іншого.

Особливості роботи з документами в онлайн-додатках полягають у такому:

- документ за замовчуванням зберігається на **Google Диску** користувача;
- збереження відбувається автоматично після того, як користувач уносить навіть незначні зміни в документ. Таким чином, результати опрацювання документа не будуть втрачені, навіть якщо на комп'ютері відбудеться збій;
- користувач може відслідковувати зміни, які відбувалися під час опрацювання документа, та за потреби повертатися до попередніх версій документа;
- можна надавати іншим користувачам дозвіл на перегляд, редагування або коментування онлайн-документів.

Використовуючи офісні онлайн-додатки, можна також переглядати та редагувати файли, які були створені у відповідних програмах пакета **Microsoft Office** та завантажені на **Google Диск**.

А файли, створені на **Google Диску** в офісних онлайн-додатках, можна зберегти на носіях даних комп'ютера у форматі відповідних програм пакета **Microsoft Office**. Для цього слід виконати **Файл** ⇒ **Завантажити** та вибрати формат потрібної програми.

СПІЛЬНИЙ ДОСТУП ДО ОБ'ЄКТІВ НА GOOGLE ДИСКУ



Поміркуйте

- У чому може бути зручність одночасного опрацювання одного й того самого файлу всіма учасниками групи під час виконання групового проекту?


Користувачі Інтернету можуть надавати дозвіл іншим користувачам на спільне використання об'єктів, розміщених

у своєму хмарному сховищі. При цьому кажуть про **надання спільного доступу** до об'єктів.

Існують такі **рівні доступу** користувачів до об'єктів у хмарному сховищі:

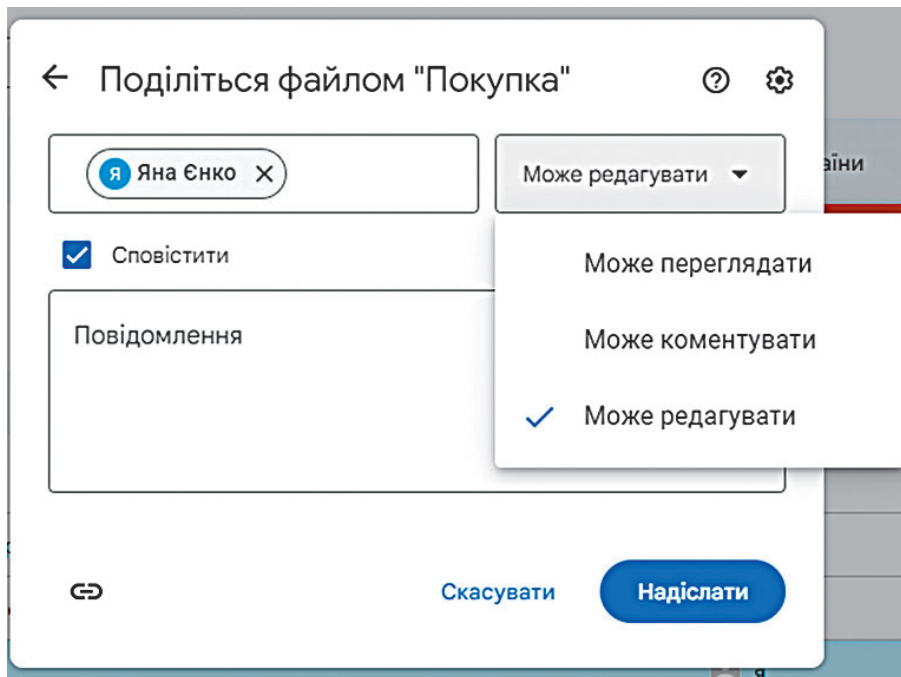
- **Може переглядати** вміст файлів і папок без можливості внесення будь-яких змін;
- **Може коментувати** вміст файлів, тобто залишати коментарі (текстові нотатки) до окремих фрагментів вмісту, не змінюючи при цьому сам вміст;
- **Може редагувати** вміст об'єктів, вносити будь-які зміни.

Для надання іншому користувачу спільного доступу до об'єкта на власному **Google Диску** потрібно:


1. Вибрати об'єкт на власному **Google Диску**.
2. Вибрати кнопку **Надати доступ**  на панелі інструментів або в його контекстному меню виконати **Надати доступ** ⇒ **Поділитися**.
3. Увести у вікні **Поділіться файлом**, що відкриється, у поле **Додайте користувачів і групи** адресу електронної поштової скриньки користувача, якому надається доступ. Якщо адреса є у вашому списку контактів, то можна вибрати її у спадному списку до завершення введення. Якщо адреса відсутня, то вибрати її у списку, що з'явиться після завершення введення.
4. Вибрати у списку рівень доступу для користувача (мал. 1.33).
5. Переконатися в наявності позначки прапорця **Сповістити**.
6. Увести в поле **Повідомлення** пояснення для користувача, до якого об'єкта та з якою метою він отримує доступ.
7. Вибрати кнопку **Надіслати**.

Після виконання власником файлу або папки цього алгоритму на його **Google Диску** в рядку об'єкта з'явиться значок

У спільному доступі  .



Мал. 1.33. Надання користувачеві спільного доступу до файлу

Користувачеві, якому надано доступ, надійде електронний лист з повідомленням про це. Для того щоб опрацювати об'єкт зі спільним доступом, потрібно вибрати кнопку **Відкрити** в отриманому листі або знайти об'єкт на власному **Google Диску** в розділі **Відкриті для мене**  **Відкриті для мене**.


Звертаємо вашу увагу!

Якщо спільний доступ установлено для папки, то такий самий рівень доступу буде одночасно надано й до всіх об'єктів у цій папці.

Спільний доступ до файлу або папки можна надати одночасно кільком користувачам. Для цього в поле **Додайте користувачів і групи** потрібно ввести кілька адрес або ім'я групи контактів.

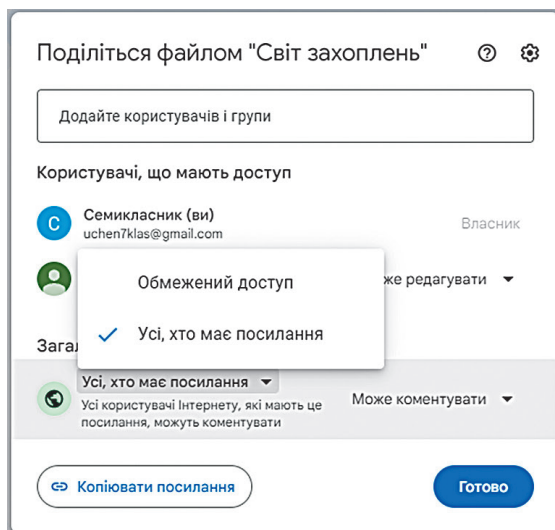
Можна також надати доступ усім користувачам, у кого буде посилання на вибраний об'єкт. Для цього потрібно у вікні **Поділіться файлом** у розділі **Загальний доступ** вибрати

у списку команду **Усі, хто має посилання** (мал. 1.34), установити рівень доступу для користувачів і вибрати кнопку **Копіювати посилання**. Скопійоване в **Буфер обміну** посилання потрібно вставити в електронні листи або поділитися іншим способом з користувачами, яким ви надаєте доступ.

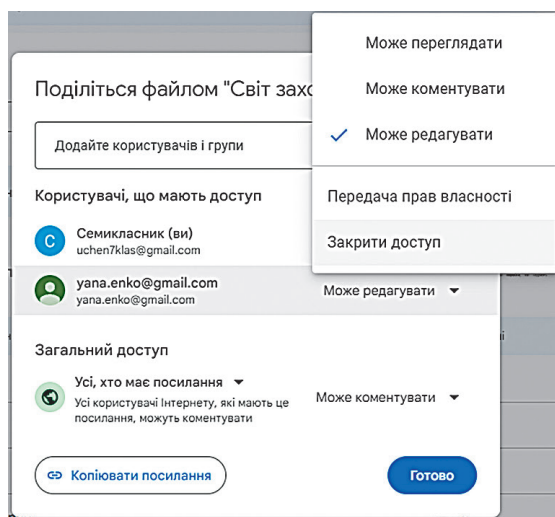
Спільний доступ до файлів можна також надавати під час їх створення або опрацювання у відповідних офісних онлайн-додатках. Для цього призначена кнопка **Поділитися** або  у вікні офісного додатка.

Відмінити спільний доступ до папки або файлу може тільки власник цього об'єкта. Якщо доступ було надано лише окремим користувачам, то для його відміни потрібно:

1. Вибрати об'єкт зі спільним доступом.
2. Відкрити вікно **Поділіться файлом** будь-яким способом, який ви знаєте.
3. Вибрати у списку поруч з адресою електронної поштової скриньки користувача команду **Закрити доступ** (мал. 1.35).
4. Вибрати кнопку **Зберегти**.



Мал. 1.34. Надання спільного доступу до файлу всім користувачам, хто має посилання



Мал. 1.35. Закриття спільного доступу до файлу для окремого користувача

Якщо потрібно відмінити спільний доступ для всіх користувачів за посиланням, то потрібно у вікні **Поділіться файлом** у розділі **Загальний доступ** вибрати у списку команду **Обмежений доступ** і підтвердити зміни.

СПІЛЬНА РОБОТА З ФАЙЛАМИ ТА ПАПКАМИ



Поміркуйте

- Навіщо може бути корисним знати, хто працює зі спільним документом одночасно з вами?

Після надання користувачам та отримання ними спільного доступу до файлів або папок можлива спільна робота з цими об'єктами.

Опрацьовувати об'єкт можуть усі користувачі, хто має доступ, одночасно або в різний час.

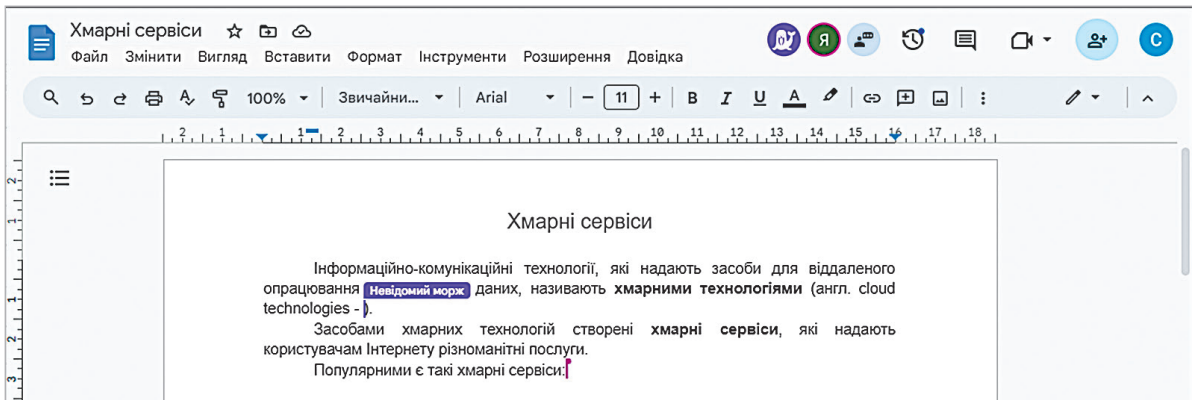
Якщо з файлом одночасно працюють кілька користувачів, то можна побачити на сторінці онлайн-додатка, хто саме працює зараз і що редагує інший користувач.

На малюнку 1.36 у верхньому правому куті в кольоровому колі з літерою (пиктограма користувача) відображено, хто працює з файлом (*Семикласник*), а в робочій області затінена та обведена зеленою рамкою клітинка, уміст якої редагує цей користувач.

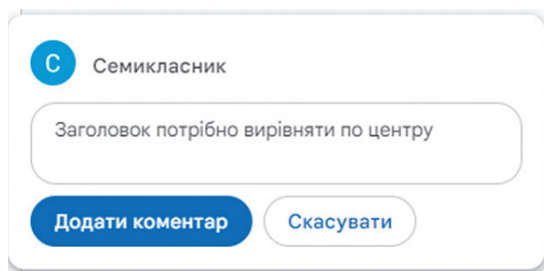
Покупка		Семикласник		
Продукт	Ціна, грн			
Хліб	25	2		50
Масло	56	1		56
Молоко	31	3		93

Мал. 1.36. Спільна робота з файлом електронних таблиць


Під час спільного редагування файлу кількома користувачами будуть відображатися всі їхні піктограми. Кожному користувачеві автоматично призначено окремий колір для позначення позиції редагування. Якщо користувач отримав не персональне посилання для спільного доступу, а загальне для всіх, то замість першої літери його імені на піктограмі буде зображення деякої тварини (мал. 1.37).






Мал. 1.37. Спільна робота з текстовим документом



Мал. 1.38. Вікно для додавання коментарів

Якщо користувачам було надано доступ для коментування, то вони не зможуть редагувати вміст файлів, але зможуть додавати коментарі. Для цього на сторінках онлайн-додатків є кнопка **Додати коментар** . Після її вибору відкривається вікно з полем для введення коментаря та кнопкою **Додати коментар** (мал. 1.38). Спроби редагування файлу також будуть оформлені як коментарі з пропозиціями внесення змін, з якими власник файлу може погодитись або відмовитись від них.

Користувачі, які мають доступ до цього файлу, можуть переглядати коментарі, вибравши кнопку **Переглянути історію**

коментарів  , і відповідати на них. Власник файлу може позначити обговорення як вирішене та прийняти пропозиції змін, вибравши у вікні коментаря кнопку **Прийняти пропозицію**  , або відмовитись від змін вибором кнопки **Відхилити пропозицію**  .

Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви можете знайти за адресою <https://cutt.ly/owNTqCHK> або QR-кодом.



Найважливіше в цьому пункті

Популярними хмарними сервісами є так звані **офісні онлайн-додатки**, які призначені для створення й опрацювання онлайн-документів різних видів: текстових документів, електронних таблиць, комп'ютерних презентацій та іншого.

Основні алгоритми зі створення й опрацювання документів з використанням онлайн-додатків майже не відрізняються від алгоритмів, які ви опанували під час роботи у програмах аналогічного призначення з пакета **Microsoft Office** або іншого.

Особливості роботи з документами в онлайн-додатках: документ за замовчуванням зберігається на **Google Диску** користувача, збереження відбувається автоматично після того, як користувач уносить зміни в документ, користувач може відслідковувати зміни, які відбувалися під час опрацювання документа, можна надавати іншим користувачам дозвіл на перегляд, редагування або коментування онлайн-документів.

Користувачі Інтернету можуть надавати дозвіл іншим користувачам на спільне використання об'єктів, розміщених у своєму хмарному сховищі. Існують такі **рівні доступу** користувачів до об'єктів: **Може переглядати, Може коментувати, Може редагувати**.

Спільний доступ до файлу або папки можна надати одному або кільком користувачам, а також усім користувачам, у кого буде посилання на вибраний об'єкт.



Дайте відповіді на запитання

1. Які ви знаєте офісні онлайн-додатки **Google**? Якими способами можна відкрити їх сторінки?
2. Які особливості роботи з документами в онлайн-додатках?
3. Які рівні доступу до об'єктів на **Google Диску** можна надавати користувачам?
4. Як надати спільний доступ до файлів і папок на **Google Диску**?



Обговоріть і зробіть висновки

1. У чому схожість і відмінність офісних онлайн-додатків **Google** та програм пакета **Microsoft Office**?
2. У чому зручність і небезпека надання доступу на редагування об'єктів на **Google Диску** для всіх користувачів, хто має посилання?
3. Навіщо потрібно знати, у якому місці онлайн-документа змінює його вміст інший користувач, який працює зі спільним документом одночасно з вами?
4. Які ви знаєте різні способи збирання, передавання та зберігання даних?





Виконайте завдання

1. Створіть на **Google Диску** папку **Завдання 1.6.1**, а в ній текстовий документ *Рівні доступу* з описом рівнів доступу до об'єктів на хмарному диску. Надайте доступ до створеної папки вашому вчителю/вчительці інформатики для редагування.
2. Створіть на **Google Диску** комп'ютерну презентацію *Рослинні символи України*. Розмістіть на титульному слайді назву презентації та ваше прізвище й ім'я, на другому слайді – зображення рослинних символів України, знайдені в Інтернеті, з їх назвами. Виберіть тему оформлення презентації із запропонованих у редакторі презентацій. Назвіть файл пре-

зентації іменем **завдання 1.6.2**. Надайте доступ до створеної презентації вашому вчителю/вчительці інформатики для перегляду.


3. Створіть на **Google Диску** електронну таблицю *Покупка*. Уведіть у першому стовпці таблиці назви трьох товарів, у другому – їх ціну, у третьому – придбану кількість. Уставте в четвертому стовпці формули для обчислення вартості придбання кожного товару та загальної вартості всієї покупки. Назвіть файл електронної таблиці іменем **завдання 1.6.3**. Надайте доступ до створеної електронної таблиці вашому вчителю/вчительці інформатики для редагування.

 **4. Працюємо в парах.** Об'єднайтесь у пари. Створіть кожен окремо на своєму **Google Диску** текстовий документ з іменем **завдання 1.6.4**. Уведіть у тексті документа список улюблених літературних творів. Надайте один одному доступ до власного документа з правом коментування. Прокоментуйте список один одного, у коментарях запропонуйте твори, які можна додати до списку. Перегляньте коментарі у власному документі, погодьтесь з ними або відхиліть.

 **5. Працюємо у групах.** Об'єднайтесь у групу з трьох осіб. Розподіліть між собою завдання з організації спільної роботи:

- учасник 1 – створення на **Google Диску** папки з іменем **завдання 1.6.5** зі спільним доступом для всіх учасників/учасниць групи та вчителя/вчительки інформатики з дозволом на редагування;
- учасник 2 – створення у спільній папці комп'ютерної презентації *Славенні міста України* з одного титульного слайда;
- учасник 3 – створення у спільній папці текстового документа із заголовком першого рівня *Використані джерела*.

Обговоріть, які міста України ви будете представляти та чому. Виконайте пошук відомостей в Інтернеті та оформте спільну презентацію (кожен учасник по одному слайду про одне з міст України). Посилання на сайти, з яких отримано відомості, вставте до текстового документа.

 **6. Дослідницьке завдання.** Створіть на **Google Диску** графічний файл, вибравши в меню кнопки **Створити** команду **Більше** ⇒ **Google Малюнки**. Створіть зображення будинку, використавши графічні примітиви. Задайте ім'я файлу **завдання 1.6.6**. Надайте доступ до створеного зображення вашому вчителю/вчительці інформатики для перегляду.

Готуємось до вивчення нового матеріалу

1. Що таке *алгоритм*?
2. Хто або що може бути виконавцем алгоритмів?
3. Для яких виконавців ви склали алгоритми в попередніх класах?

Оцініть свої знання та вміння з розділу «Пошук в Інтернеті. Електронна пошта. Хмарні сервіси»

Оцініть свої навчальні досягнення із цього розділу (початковий, середній, достатній, високий рівень).

- Я знаю та застосовую різні стратегії пошуку відомостей в Інтернеті.
- Я вмію створювати закладки у браузері на корисні ресурси Інтернету.
- Я можу оцінити достовірність даних різних типів і надійність джерел їх отримання.
- Я можу розпізнати техніку маніпуляцій і пропаганди в медіатекстах.
- Я вмію взаємодіяти з іншими особами засобами електронної пошти.

- Я дотримуюсь принципів етикету та кібербезпеки під час електронного спілкування.
 - Я знаю переваги та недоліки використання хмарних сервісів.
 - Я використовую хмарні сервіси для збирання, передавання та зберігання даних.
 - Я вмію використовувати онлайн-перекладачі.
 - Я вмію створювати онлайн-документи, використовуючи хмарні сервіси.
 - Я знаю та використовую хмарні сервіси та інтернет-ресурси для індивідуальної або групової діяльності та комунікації.
 - Я можу забезпечити інформаційну безпеку для себе, власних пристроїв і даних під час використання спільних документів.
- Повторіть той матеріал, який ви знаєте недостатньо.*



Практична робота № 2. «Хмарні сервіси»

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

1. Створіть на **Google Диску** папку з іменем **Практична робота 2**. Надайте вчителю/вчительці інформатики доступ до папки з правом перегляду. Створіть у папці **Практична робота 2** текстовий документ.
2. Перекладіть, використовуючи **Google Перекладач**, вебсторінку з адресою iucn.org українською мовою. Визначте, що означає аббревіатура *IUCN*.
3. Скопіюйте з вебсторінки *IUCN* 2–3 речення перекладеного тексту з поясненням призначення організації *IUCN* і вставте в текстовий документ. Відформатуйте текст. Надайте документу ім'я *Охорона природи*. Надайте вчителю/вчительці інформатики доступ до файлу з правом редагування.
4. Збережіть одне із зображень з вебсторінки *IUCN* до вашої папки. Завантажте збережений графічний файл з вашого комп'ютера на **Google Диск** у папку **Практична робота 2**.

Розділ 2

АЛГОРИТМИ ТА ПРОГРАМИ

У цьому розділі ви отримаєте нові, а також поглибите та розширите свої знання й удосконалисте навички з таких тем:

- ▶ програми та мови програмування;
- ▶ складові частини мов програмування;
- ▶ професії, у яких потрібно вміти створювати програми;
- ▶ проєкти з використанням вікна введення-виведення;
- ▶ віконні проєкти;
- ▶ лінійні алгоритми та проєкти;
- ▶ алгоритми та проєкти з розгалуженнями;
- ▶ проєкти для створення графічних зображень на вікні.

У підручнику цей розділ викладено з використанням мови програмування **Python** і середовища створення та редагування проєктів **IDLE**.

Текст цього розділу з використанням мови програмування **Object Pascal** і середовища створення та редагування проєктів **Lazarus** знаходиться за посиланням <https://cutt.ly/1w1cUMRe> або QR-кодом.



2.1. Комп'ютерні програми. Мови програмування. Програми-транслятори

У цьому пункті йтиметься про:

- ▶ комп'ютерні програми;
- ▶ інтерфейс користувача;
- ▶ мови програмування, їх складові;
- ▶ транслятори, їх види;
- ▶ професії, які потребують вміння створювати комп'ютерні програми.

АЛГОРИТМИ ТА КОМП'ЮТЕРНІ ПРОГРАМИ. ІНТЕРФЕЙС КОРИСТУВАЧА



Пригадайте

• Що таке *алгоритм*? • Хто або що може бути виконавцем алгоритмів? • Чи були ви виконавцями алгоритмів? Якщо так, то в яких ситуаціях? • Чи складали ви алгоритми? Якщо так, то де і для чого?

Ви вже знаєте, що алгоритм, записаний спеціальною мовою та призначений для виконання комп'ютером, називається **комп'ютерною програмою**.

Ви вже не тільки використовували різноманітні комп'ютерні програми: текстовий процесор, графічний редактор, табличний процесор, редактор комп'ютерних презентацій, навчальні та контролюючі програми, ігрові програми та багато інших, а й самі складали їх.

У багатьох сучасних комп'ютерних програмах (проєктах) використовуються елементи керування (кнопки, меню та інше), засоби введення даних (поля, лічильники та інше), засоби виведення даних (написи, поля та інше) тощо. Сукупність усіх таких засобів керування, а також методів взаємодії людини і комп'ютера утворюють **графічний інтерфейс користувача**. Прикладами програм з графічним інтерфейсом користувача є знайомі вам операційна система **Windows**, текстовий процесор **Word** та інші.

Крім графічного інтерфейсу користувача, сучасні комп'ютерні проєкти використовують і **текстовий інтерфейс користувача**. У такому інтерфейсі використовується тільки вікно введення-виведення, де користувач уводить дані і/або команди у вигляді чисел і текстів, а комп'ютер виводить результати також у вигляді чисел і/або текстів. Прикладами програм з текстовим інтерфейсом користувача є операційна система **Unix/Linux**, файловий менеджер **FAR** та інші (мал. 2.1).

```
C:\Users\rivki>md one  
C:\Users\rivki>
```

Мал. 2.1. Вікно програми з текстовим інтерфейсом і командою **md one**

Під час вивчення наступних пунктів ви спочатку будете вчитися створювати проекти з текстовим інтерфейсом, а потім – з графічним інтерфейсом.

МОВИ ПРОГРАМУВАННЯ



Поміркуйте

- Що таке *система команд виконавця*? • Чи довільні команди може виконувати виконавець? Поясніть свою відповідь. • Чи повинні виконавці алгоритмів розуміти команди алгоритмів? Поясніть свою відповідь.

Складаючи алгоритми, призначені для виконання людиною, користуються мовою спілкування людей (українською, англійською, німецькою та іншими), мовою жестів, спеціальних позначок, світловими та іншими сигналами тощо.

Але для запису алгоритмів, які повинен виконувати автоматичний пристрій (зокрема, комп'ютер), мова спілкування людей складна, має неоднозначні тлумачення (наприклад, слова-омоніми). Тому для запису алгоритмів, які призначені для виконання автоматичними пристроями, розробляють і використовують спеціальні мови.

Мова, яка використовується для запису алгоритмів, призначених для виконання комп'ютером, називається **мовою програмування**.

За останні 80 років створено приблизно три тисячі різних мов програмування. Деякі з них вже не використовують, для

деяких постійно з'являються нові версії, які зручніші у використанні та мають більше можливостей. Постійно створюються нові мови програмування.

Деякі мови програмування є **універсальними** й використовуються для складання програм для розв'язування задач з різних галузей науки, техніки, виробництва, сфери побуту та ін. Деякі мови програмування є **спеціальними** та створені для складання програм для розв'язування спеціальних задач, наприклад економічних.

У 2023 році найбільше комп'ютерних програм створювалися мовами програмування **Java, JavaScript, C# (C Sharp), PHP, Python, C++**.

Кожна мова програмування має такі складові:

1. **Алфавіт** – множина символів, з яких можна утворювати слова і речення цієї мови.
2. **Словник** – набір слів, які використовуються в цій мові.
3. **Синтаксис** – правила складання і запису мовних конструкцій: слів, що не увійшли до словника, і речень.
4. **Семантику** – установлене однозначне тлумачення мовних конструкцій, правил їх виконання.

Використання символів, що не входять до алфавіту мови програмування (крім спеціально обумовлених випадків), неправильне написання словникових слів, порушення синтаксичних правил призводить до неможливості виконання комп'ютером відповідної команди. Такі порушення називаються **синтаксичними помилками**.

У 5–6 класах ви вивчали мову програмування **Scratch**. Пригадаємо, що до алфавіту цієї мови програмування входять цифри, літери української абетки, англійського алфавіту, символи **:**, **=**, **+**, *****, **<** та інші. До її словника входять, наприклад, слова **повторити, перемістити, якщо, відбити, коли**.

Відповідно до синтаксичних правил цієї мови програмування команда циклу з передумовою має розпочинатися зі слів **повторити до** і саме в цьому порядку. А відповідно до семантики цієї мови програмування комп'ютер однозначно виконає команду **перемістити на 10 кроків** і перемістить виконавця на 10 кроків у напрямку його руху.

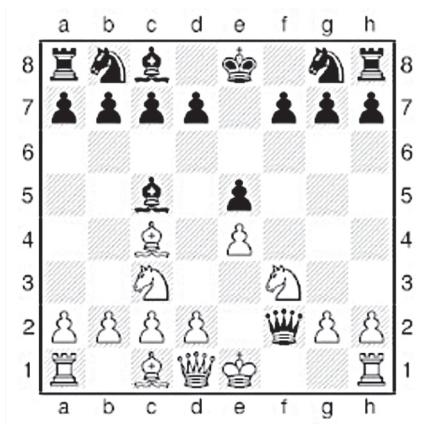
Для тих, хто хоче знати більше

Мови програмування є прикладами так званих **штучних мов**.

Іншими прикладами штучних мов є, наприклад, мова запису хімічних реакцій, мова запису розв'язування математичних рівнянь і нерівностей, мова запису поштових індексів на конвертах і листівках, мова запису ходів шахових партій, мова запису музичних композицій і багато інших (мал. 2.2).

Запис шахової партії:

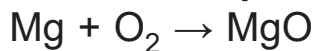
1. e4 e5
2. Kc3 Cc4
3. Cc4 Fh4
4. Kf3 Ф×f2#



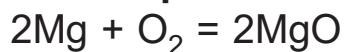
Запис музичної композиції:



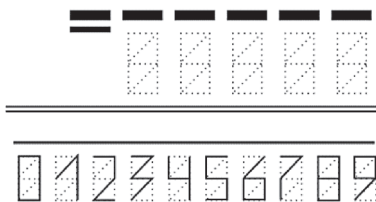
Схема хімічної реакції:



Хімічне рівняння:



Алфавіт мови для запису поштового індексу:



Мал. 2.2. Приклади штучних мов

ТРАНСЛЯТОРИ. КОМПІЛЯТОРИ ТА ІНТЕРПРЕТАТОРИ

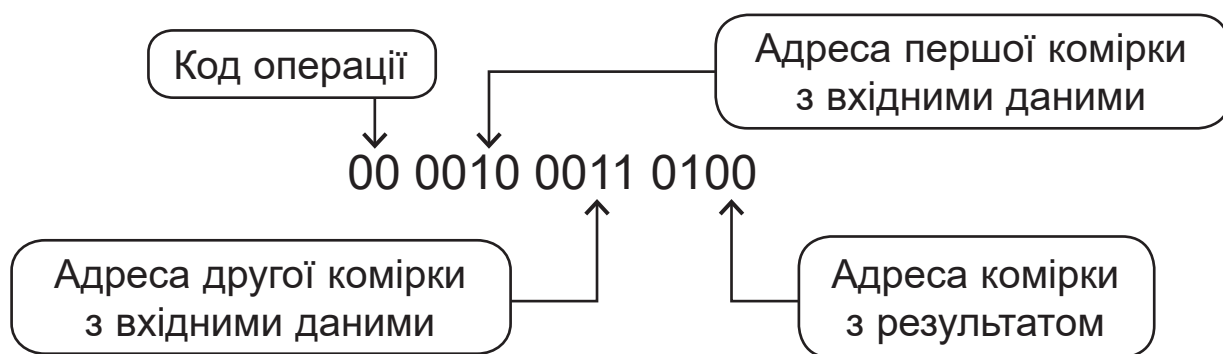


Пригадайте

- Що ви робите, якщо вам трапляється незнайоме слово іноземною мовою?
- Що ви чули про професію *синхронний перекладач*? Знайдіть, як звучить назва цієї професії англійською мовою.

Перші мови програмування з'явилися задовго до появи перших комп'ютерів. Ще в ХІХ ст. існували «програмовані» ткацькі верстати та піаніно-програвачі.

Перші мови програмування для комп'ютерів з'явилися в середині 40-х років ХХ ст. Команди цих мов записувалися числами, які складалися з цифр 0 та 1, і містили код операції, яку потрібно виконати, адреси комірок пам'яті, звідки потрібно взяти дані, адресу комірки пам'яті, куди потрібно помістити результат (мал. 2.3).



Мал. 2.3. Приклад команди перших мов програмування

Такі мови програмування були орієнтовані на конкретні процесори, тобто одні й ті самі команди для різних процесорів мали різні коди. Тому програми, складені для одних процесорів, не могли використовуватися на комп'ютерах з процесором іншого типу. Складання програм такими мовами програмування було справою важкою і тривалою. Такі мови називають **мовами програмування низького рівня**.

Перші мови програмування високого рівня, команди яких використовували не тільки числа, а й спеціальні слова, аналогічні або схожі на слова мов спілкування людей, з'явилися на початку 50-х років ХХ ст. Однією з перших була мова програмування **Fortran** (англ. *The IBM Mathematical Formula Translating System* – система перекладу математичних формул фірми IBM). Ось приклади кількох команд цією мовою:

1) read a 2) read b 3) x = a + b 4) write x.

А для того щоб перекласти програму, написану мовою програмування високого рівня, на мову команд процесора, створили спеціальні програми – **транслятори** (англ. *translate* – перекладати).

Програми-транслятори поділяються на 2 види: *інтерпретатори* і *компілятори*.

Інтерпретатор бере послідовно кожну команду проєкту, написаного мовою програмування високого рівня, і перш за все аналізує її на наявність синтаксичних помилок. Якщо такі помилки виявляються, то інтерпретація програми переривається і виводиться повідомлення про помилку, яку потрібно виправити. Якщо в команді відсутні синтаксичні помилки, то інтерпретатор перекладає її на мову команд процесора та передає процесору на виконання. І так далі з кожною командою проєкту.

Компілятор бере весь текст проєкту, написаного мовою програмування високого рівня, і аналізує всі команди на наявність синтаксичних помилок. Якщо такі виявляються, то компілятор перериває свою роботу й виводить повідомлення про всі ці помилки. Якщо в тексті проєкту синтаксичні помилки відсутні, то компілятор перекладає всі команди проєкту на мову команд процесора, створює і записує на носій **exe**-файл і передає його процесору на виконання.

Для тих, хто хоче знати більше

І інтерпретатор, і компілятор мають як переваги, так і недоліки (табл. 2.1).

Таблиця 2.1

Переваги та недоліки інтерпретатора та компілятора

	<i>Переваги</i>	<i>Недоліки</i>
Інтер- пре- татор	Займає менше місця в оперативній пам'яті комп'ютера	<ol style="list-style-type: none">1. Проєкт, який інтерпретується, виконується повільніше від проєкту, який компілюється.2. Команди циклу перекладаються (інтерпретуються) стільки разів, скільки вони виконуються. А це збільшує час виконання проєкту.3. Виконання проєкту може розпочатися, але не завершитися отриманням результату, якщо синтаксична помилка в останніх командах.4. При кожному наступному виконанні проєкту процес інтерпретації розпочинається спочатку.5. Програма-інтерпретатор повинна бути обов'язково встановлена на тому комп'ютері або сайті, де виконується проєкт
Компі- лятор	<ol style="list-style-type: none">1. Проєкт, який компілюється, виконується швидше за проєкт, який інтерпретується.2. Програма-компілятор не використовується після закінчення компіляції та створення exe-файлу	Займає більше місця в оперативній пам'яті комп'ютера



Цікаві факти з історії

Мова програмування **Python** була створена в 1991 році нідерландським програмістом **Гвідо ван Россумом** (нар. у 1956 р., мал. 2.4), на той час співробітником нідерландського **Національного дослідницького інституту математики та інформатики**.



Мал. 2.4. Гвідо ван Россум

Автор цієї мови програмування назвав її на честь героя популярного британського комедійного серіалу 70-х років XX ст. «Повітряний цирк Монті Пайтона». Назва **IDLE** середовища розробки проєктів, у якому можна створювати проєкти мовою **Python**, також походить від імені одного з героїв цього серіалу, Еріка **Айдла** (англ. *Eric Idle*).

Сьогодні ця мова набула широкого застосування в різних сферах: для навчання учнів і студентів принципів програмування, наукових розрахунків, графічної візуалізації даних, опрацювання тривимірних зображень, адміністрування мереж та операційних систем, розширення можливостей готових програм, створення мобільних додатків тощо.

ПРОФЕСІЯ «ПРОГРАМІСТ»



Поміркуйте

- Що ви чули про професію «програміст»? Які знання та вміння повинні бути в такої людини?
- Чи повинні вміти створювати комп'ютерні програми люди інших професій? Якщо так, то яких саме професій і чому?

Програміст – це професія людини, яка створює комп'ютерні програми (проєкти). Ця людина повинна знати одну або кілька мов програмування, які вона буде використовувати у своїй роботі.

Добре, якщо ця людина вміє за потреби створити математичну модель для розв'язування поставленої задачі та склас-

ти алгоритм у вигляді блок-схеми або в іншому вигляді. Якщо ні, то цю роботу для нього виконують інші люди, але в такому разі рівень кваліфікації та заробітної плати такого програміста значно нижче.

Програмістів можна розподілити на 2 групи: *системні програмісти* і *прикладні програмісти*.

Системні програмісти створюють операційні системи, під управлінням яких працюють комп'ютери, програми для роботи комп'ютерних мереж, драйвери пристроїв тощо.

Прикладні програмісти створюють різноманітні редактори, спеціалізовані програми для використання бухгалтерами, економістами, працівниками інших професій, системи керування базами даних, системи опрацювання відеоспостережень, інтерактивні веб-сторінки тощо.

Такий поділ доволі умовний, тому що багато програмістів можуть створювати як системні, так і прикладні програми.

Але не тільки програмісти створюють комп'ютерні програми. Наприклад, **фінансовий аналітик**, який аналізує тенденції ринку валют, цінних паперів, нерухомості, може кожного разу складати нові таблиці в табличному процесорі та використовувати їх у своїй роботі. А може скласти спеціальну програму, яка значно автоматизує, спростить і прискорить його роботу.

Драйвер пристрою – програма, яка узгоджує роботу зовнішнього пристрою, наприклад принтера, з комп'ютером, до якого він підключений.



Найважливіше в цьому пункті

Алгоритм, записаний спеціальною мовою та призначений для виконання комп'ютером, називається **комп'ютерною програмою**.

Сукупність таких засобів керування, як вікно, кнопка, поле, прапорці, перемикачі та інші, а також методів взаємодії людини та комп'ютера утворюють **графічний інтерфейс користувача**.

У **текстовому інтерфейсі користувача** використовується тільки вікно введення-виведення, у якому користувач уводить дані і/або команди у вигляді чисел і текстів, а комп'ютер виводить результати також у вигляді чисел і/або текстів.

Мова, яка використовується для запису алгоритмів, призначених для виконання комп'ютером, називається **мовою програмування**.

Кожна мова програмування має такі складові:

1. **Алфавіт** – множина символів, з яких можна утворювати слова і речення цієї мови.
2. **Словник** – набір слів, які використовуються в цій мові.
3. **Синтаксис** – правила складання і запису мовних конструкцій: слів, що не увійшли до словника, і речень.
4. **Семантику** – установлене однозначне тлумачення мовних конструкцій, правил їх виконання.

Синтаксичні помилки виникають під час використання символів, що не входять до алфавіту мови програмування (крім спеціально обумовлених випадків), неправильного написання словникових слів, порушення синтаксичних правил.

Транслятор – спеціальна програма, призначена для того, щоб перекласти програму, написану мовою програмування високого рівня, на мову команд процесора.

Програми-транслятори поділяються на 2 види: **інтерпретатори** і **компілятори**.

Програміст – це професія людини, яка створює комп'ютерні програми (проекти). Ця людина повинна знати одну або кілька мов програмування, які вона використовуватиме у своїй роботі.



Дайте відповіді на запитання

1. Що таке *комп'ютерна програма*?
2. Що таке *мова програмування*?
3. Які складові частини мови програмування? Опишіть кожену з них.
4. Що таке *синтаксична помилка*?
5. Яке призначення програми-транслятора?
6. Які види трансляторів ви знаєте?

7. Чим відрізняється робота програми-інтерпретатора від роботи програми-компілятора?

Обговоріть і зробіть висновки

1. Порівняйте складові частини відомої вам мови програмування з аналогічними складовими частинами української мови.

2. Чим відрізняються наслідки допущення вами синтаксичної помилки під час написання диктанту українською мовою і під час уведення тексту комп'ютерної програми?

3. Чому перші програми-транслятори були інтерпретаторами, а не компіляторами, а в 60–70-х роках ХХ ст. програми-компілятори майже витіснили програми-інтерпретатори?

Виконайте завдання

1. Створіть текстовий документ-повідомлення про мови програмування:

- | | | |
|-----------------------|---------------------|-------------------|
| а) Assembler ; | б) Fortran ; | в) Cobol ; |
| г) Prolog ; | д) Rubi ; | е) Lisp . |

Збережіть створений документ у вашій папці у файлі з іменем **завдання 2.1.1**.


2. Створіть текстовий документ-повідомлення про перші інтерпретатори. Збережіть створений документ у вашій папці у файлі з іменем **завдання 2.1.2**.


3. Створіть комп'ютерну презентацію про перші компілятори. Збережіть створену комп'ютерну презентацію у вашій папці у файлі з іменем **завдання 2.1.3**.

4. Створіть електронну таблицю з даними про популярність 10 сучасних мов програмування за останні 2 роки. За даними цієї таблиці побудуйте діаграму.



Скористайтеся ресурсом <https://www.tiobe.com/tiobe-index> або іншими. Збережіть створену електронну книгу у вашій папці у файлі з іменем **завдання 2.1.4**.

 **5. Робота у групах.** Створіть текстовий документ з порівнянням довільних двох програм з текстовим і графічним інтерфейсом користувача. Збережіть створений документ у вашій папці у файлі з іменем **завдання 2.1.5**.

 **6. Робота у групах.** Створіть комп'ютерну презентацію про професії, у яких потрібно вміти створювати програми. Збережіть створену комп'ютерну презентацію у вашій папці у файлі з іменем **завдання 2.1.6**.



Готуємось до вивчення нового матеріалу

1. Які види величин ви знаєте?
2. Для чого використовується команда присвоювання? Наведіть кілька прикладів.

2.2. Мова програмування PYTHON. Середовище створення проєктів IDLE. Створення лінійних проєктів

У цьому пункті йтиметься про:

- ▶ мову програмування **Python**;
- ▶ середовище створення проєктів **IDLE Python**;
- ▶ створення лінійних проєктів.

МОВА ПРОГРАМУВАННЯ PYTHON. СЕРЕДОВИЩЕ СТВОРЕННЯ ПРОЄКТІВ IDLE



Пригадайте

- Як ви створювали проєкти в середовищі **Scratch**?

У попередніх класах ви вивчали мову програмування **Scratch** і створювали проєкти в середовищі **Scratch**.


У 7–9 класах ви вивчатимете мову програмування **Python** і будете створювати проєкти в середовищі **IDLE Python**.

Середовище створення проєктів IDLE (англ. *Integrated Development and Learning Environment* – інтегроване середовище розробки та навчання) складається з:

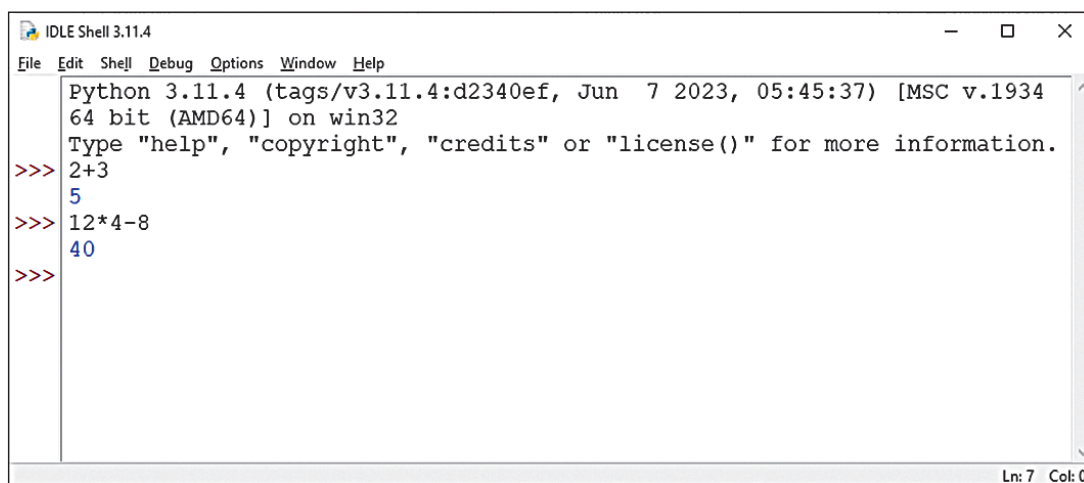
- **текстового редактора** для введення та редагування тексту проєкту;
- **інтерпретатора** для перекладу команд з мови програмування **Python** на мову команд, які може виконати процесор комп'ютера;
- **засобів налагодження проєкту** для пошуку в ньому помилок;
- **довідкової системи** та інших компонентів.

Установити це середовище на свій комп'ютер можна безкоштовно з офіційного сайту **Python Software Foundation** (англ. *Python Software Foundation* – фонд програмного забезпечення **Python**) (python.org).

Відкрити вікно середовища **IDLE** можна, використовуючи:

- команду **IDLE** у розділі **Python** меню **Пуск**;
- значок  на **Робочому столі**.

У результаті відкривається вікно **IDLE Shell** (англ. *shell* – оболонка) середовища створення та виконання проєктів **IDLE** (мал. 2.5).



```
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934
64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 2+3
5
>>> 12*4-8
40
>>>
```

Мал. 2.5. Вікно **IDLE Shell** середовища створення та виконання проєктів **IDLE**

На лівому полі вікна символи >>> позначають рядок, у який можна вводити команди мовою програмування **Python**. Уведення команди завершується натисканням клавіші **Enter**, після чого команда виконується, результат її виконання відображається в наступному після команди рядку. Такий режим введення та виконання команд називається **інтерактивним**.

Інтерактивний режим зручний для проведення обчислень, перевірки правильності написання команд, реалізації найпростіших проєктів. Але для розробки складніших проєктів доцільно записувати команди проєкту в окремому документі, що дасть змогу зберегти проєкт у файлі на носії й далі неодноразово його виконувати, не вводячи повторно кожен команду.

Створити нове вікно для введення тексту проєкту можна командою **File** ⇒ **New File**. Після цього відкривається вікно *untitled* (англ. *untitled* – без назви) текстового редактора, у якому можна вводити текст проєкту мовою **Python**.

Для збереження проєкту на носії потрібно у вікні текстового редактора виконати команду **File** ⇒ **Save**. У вікні збереження файлу потрібно вибрати папку для зберігання та ввести ім'я файлу. Файл зберігається з розширенням імені **py**. Ім'я вікна текстового редактора буде замінено на ім'я файлу, у якому збережеться текст проєкту.

Під час повторного збереження проєкту в тій самій папці й з тим самим іменем ім'я файлу вводити не потрібно.

Проєкт можна також зберегти в іншій папці та змінити ім'я файлу. Для цього потрібно виконати команду **File** ⇒ **Save as**.

Щоб відкрити проєкт, збережений на носії, потрібно виконати команду **File** ⇒ **Open**. Після цього у вікні, що відкриється, вибрати файл з розширенням імені **py** та вибрати кнопку **Відкрити**.

Ще один спосіб відкриття проєкту – відкрити папку з файлом проєкту та вибрати в контекстному меню файлу команду ***Edit with IDLE*** ⇒ ***Edit with IDLE***. Після цього автоматично відкриється середовище **IDLE** і в ньому відкриється проєкт.

Звертаємо вашу увагу!

На багатьох ресурсах Інтернету встановлено середовища створення проєктів, які містять у своєму складі компілятори та інтерпретатори багатьох мов програмування, у тому числі інтерпретатор мови **Python**. Тому за наявності стабільного доступу до Інтернету можна створювати проєкти в одному з хмарних **IDE** (англ. ***Integrated Development Environment*** – інтегроване середовище розробки), наприклад **Ideone** (ideone.com), **Python IDE** (pythonide.online) або **Online Python** (online-python.com), які не потребують реєстрації користувача.

КОМАНДА ПРИСВОЮВАННЯ. ТИПИ ЗМІННИХ ВЕЛИЧИН



Пригадайте

- Які величини називаються *змінними* і які *сталими* (константами)? Наведіть приклади.
- Для чого використовується команда присвоювання? Наведіть приклади.
- У чому полягає *лінійний запис* математичного виразу? Чим він відрізняється від записів математичних виразів, які використовуються в математиці? Наведіть приклади.
- Які функції ви використовували для обчислень в електронних таблицях? Для яких обчислень вони призначені?

Ви вже знаєте, що команда присвоювання має такий загальний вигляд:

<ім'я змінної> <знак присвоювання> <значення або вираз>

Кожна змінна має ім'я, значення і тип.

Тип змінної визначає:

- яких значень може набувати ця змінна;
- які операції над нею можна виконувати;

- який обсяг оперативної пам'яті комп'ютера буде виділено для зберігання значення цієї змінної.

Числові вирази можуть містити:

- числа (цілі та дробові); ціла частина числа відокремлюється від дробової частини **крапкою**;
- знаки арифметичних операцій (+, −, *, /);
- дужки для змінення традиційного порядку виконання арифметичних операцій;
- спеціальні функції для виконання математичних та інших операцій; наприклад, функція **abs(x)** (англ. *absolute value* – абсолютна величина числа, модуль числа) обчислює модуль числа.

Як і в електронних таблицях, у проєктах мовами програмування використовують відомий вам *лінійний запис* математичних виразів. Наприклад, вираз $\frac{2x - 3}{3x - 7.5}$ у лінійному записі

буде таким: **(2 * x - 3) / (3 * x - 7.5)**.

У мові програмування **Python** як знак присвоювання використовують символ **=**.

Кілька команд у проєктах мовою **Python** можна записати в одному рядку, розділяючи їх символом **;** (крапка з комою).

Імена змінних у мові програмування **Python** можуть містити великі та малі українські та англійські літери, цифри та знак підкреслення (**_**).

Звертаємо вашу увагу!

1. Ім'я змінної в мові програмування **Python** не може розпочинатися з цифри.

2. У мові програмування **Python** великі та малі літери розрізняються, тобто **X** і **x** – це імена різних змінних.

Наведемо кілька прикладів команд присвоювання:

x = -24; Маса = 32.67; My_score = 12 - 45 * 56.3; i = i + 7;

У проєктах мовою програмування **Python** тип змінних не фіксований. Він визначається автоматично значенням змінної та може змінюватися в ході виконання проєкту.

Розглянемо два типи змінних:

- **float** (англ. *float* – плаваючий) – змінні цього типу можуть набувати як цілих, так і дробових значень;
- **int** (англ. *integer* – цілий) – змінні цього типу можуть набувати тільки цілих значень.

КОМАНДИ ВВЕДЕННЯ І ВИВЕДЕННЯ



Пригадайте

- Як вводяться значення вхідних даних у мові програмування **Scratch**?
- Як виводяться результати в мові програмування **Scratch**?

Для виведення значень змінних, результатів обчислення, текстів у мові програмування **Python** використовується команда **print(<текст у лапках>, <список виразів і/або імен змінних>)** (англ. *print* – друкувати).

Наприклад, команда **print('Сума чисел дорівнює ', s)** виводить у вікно **IDLE Shell** рядок, який розпочинається з тексту **Сума чисел дорівнює**, а далі виводить значення змінної **s**.

Для виведення значень змінних або виразів типу **float** доцільно для зручності округлювати ці значення до певної кількості цифр у дробовій частині. Для цього можна використати спеціальну функцію **round()** (англ. *round* – круглий), указавши в дужках ім'я змінної або вираз, а також кількість цифр у дробовій частині. Наприклад, команда **print(round(x,2))** виведе значення змінної **x**, округлене до двох цифр у дробовій частині.

Використовуючи команду **print**, можна:

- виводити тільки текст, наприклад **print('Ми вивчаємо мову Python!');**

- виводити тільки значення змінних, наприклад `print(x)`;
- виводити текст і значення виразу, округлюючи його, наприклад `print('Значення виразу дорівнює ', round((8.5*x-12)/(2-4.7*x),2))`;
- виводити кілька текстів і значення кількох змінних або виразів, округлюючи дробові числа, наприклад `print('x= ', x, 'y= ', y, 't= ', round((2*3.54-14)/(235-0.12*15),4))`; (змінна `x` типу `float`, а змінна `y` типу `int`).

Якщо в команді `print` потрібно вивести кілька текстів і значення кількох змінних або виразів, то можна розділити запис команди на кілька рядків для зручності її сприйняття. Для цього на місці поділу потрібно поставити символ `\` і натиснути клавішу **Enter**. Наприклад,

```
print('x= ', x, \
      'y= ', y, \
      't= ', round((2*3.54-14)/(235-0.12*15),4));
```

На малюнку 2.6 наведено проєкт з різними командами виведення і результат його виконання.

The screenshot shows two windows from the IDLE Python environment. The left window is a text editor with the following code:

```
1 x = -2.15
2 y = 26
3
4 print('Сума цих чисел: ', x+y)
5 print('Ми вивчаємо мову Python!')
6 print(x)
7 print('Значення виразу дорівнює ', \
8       round((8.5*x-12)/(2-4.7*x),3))
9 print('x=', x, \
10       'y=', y, \
11       't=', round((2*3.54-14)/(235-0.12*15),4))
12
```

The right window is the IDLE Shell, showing the output of the script:

```
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()"
for more information.
>>>
= RESTART: C:/Users/Пользователь/AppData/Local/Programs/Python/Python311/1.py
Сума цих чисел: 23.85
Ми вивчаємо мову Python!
-2.15
Значення виразу дорівнює -2.501
x= -2.15 y= 26 t= -0.0297
>>>
```

Мал. 2.6. Проєкт з командами виведення і результат його виконання

Для введення значень вхідних даних під час виконання проєктів у мові програмування **Python** використовуються команди:

- **<ім'я змінної> = int(input())**, наприклад **k = int(input());** **int** вказує тип значення, яке буде надано змінній (ціле), а **input()** – забезпечує введення значення з клавіатури (англ. **input** – введення);
- **<ім'я змінної> = float(input())**, наприклад **m = float(input());** **float** вказує тип значення, яке буде надано змінній (дробове).

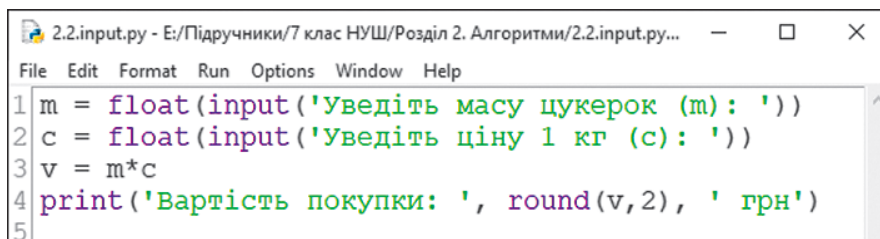
Під час виконання однієї з цих команд відкривається вікно **IDLE Shell**, у вікні з'являється курсор і виконання проєкту призупиняється. Користувач повинен увести значення та натиснути клавішу **Enter**. Після цього введене значення буде присвоєно відповідній змінній і виконання проєкту продовжиться.

Для пояснення того, які саме дані очікуються від користувача, усередині дужок у команді **input()** можна ввести пояснювальний текст. Наприклад:

```
k = int(input("Уведіть кількість днів: "));  
m = float(input("Уведіть масу цукерок: "));
```

У такому разі під час виконання проєкту у вікні **IDLE Shell** спочатку буде виведено пояснювальний текст, після нього з'явиться курсор і виконання проєкту буде призупинено до введення даних користувачем.

На малюнку 2.7 наведено приклад проєкту з використанням команди введення та етапи його виконання.



```
2.2.input.py - E:/Підручники/7 клас НУШ/Розділ 2. Алгоритми/2.2.input.py...
File Edit Format Run Options Window Help
1 m = float(input('Уведіть масу цукерок (m): '))
2 c = float(input('Уведіть ціну 1 кг (c): '))
3 v = m*c
4 print('Вартість покупки: ', round(v,2), ' грн')
5
```

а) проєкт з командами введення та виведення

Мал. 2.7. Проєкт з використанням команди введення та етапи його виконання (початок)

```
*IDLE Shell 3.11.4*
File Edit Shell Debug Options Window Help
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37)
[MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more
information.
>>>
= RESTART: Е:/Підручники/7 клас НУШ/Розділ 2. Алгоритми/2.2
.input.py
Уведіть масу цукерок (m): |
```

б) призупинення виконання проєкту на початку виконання команди введення

```
*IDLE Shell 3.11.4*
File Edit Shell Debug Options Window Help
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37)
[MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more
information.
>>>
= RESTART: Е:/Підручники/7 клас НУШ/Розділ 2. Алгоритми/2.2
.input.py
Уведіть масу цукерок (m): 2.5
Уведіть ціну 1 кг (c): 34.75|
```

в) введення значень змінних m і c перед останнім натисненням клавіші **Enter**

```
IDLE Shell 3.11.4
File Edit Shell Debug Options Window Help
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37)
[MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more
information.
>>>
= RESTART: Е:/Підручники/7 клас НУШ/Розділ 2. Алгоритми/2.2
.input.py
Уведіть масу цукерок (m): 2.5
Уведіть ціну 1 кг (c): 34.75
Вартість покупки: 86.88 грн
>>>
```

г) результат виконання проєкту

Мал. 2.7. Проєкт з використанням команди введення та етапи його виконання (продовження)

Проєкти, у яких кожна команда обов'язково виконується і тільки один раз, називаються **лінійними**.

ВИКОНАННЯ ПРОЄКТІВ



Пригадайте

- Як запустити проєкт на виконання в середовищі **Scratch**?

Для запуску проєкту на виконання можна скористатися одним з таких способів:

- установити курсор у вікно текстового редактора та натиснути клавішу **F5**;
- у вікні текстового редактора виконати **Run** \Rightarrow **Run Module**.

Після запуску проєкту на виконання починає працювати програма-інтерпретатор. Вона по чергово аналізує команди проєкту на наявність синтаксичних помилок. Якщо команда не має помилок, то інтерпретатор перекладає її з мови **Python** на мову команд, які може виконати процесор комп'ютера, і передає перекладену команду на виконання процесору. Результат виконання команди відображається у вікні **IDLE Shell**. Після цього виконується аналіз і виконання наступної команди. За наявності синтаксичної помилки інтерпретатор перериває виконання проєкту та виводить коментар про цю помилку (мал. 2.8).

```
project1.py - C:/Users/Пользователь/AppD...  IDLE Shell 3.11.4
File Edit Format Run Options Window Help  File Edit Shell Debug Options Window Help
x = float(input())
prin(x)
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 20
23, 05:45:37) [MSC v.1934 64 bit (AMD64)] on w
in32
Type "help", "copyright", "credits" or "licens
e()" for more information.
>>>
= RESTART: C:/Users/Пользователь/AppData/Local
/Programs/Python/Python311/project1.py
Traceback (most recent call last):
  File "C:/Users/Пользователь/AppData/Local/Pr
ograms/Python/Python311/project1.py", line 1,
in <module>
    x = float(input())
NameError: name 'float' is not defined. Did you
mean: 'float'?
>>> |
```

Мал. 2.8. Вікно з програмним кодом проєкту (ліворуч) і вікно **IDLE Shell** з коментарем до синтаксичної помилки (праворуч)

Потрібно виправити помилку та знову запустити проєкт на виконання.

Ознакою того, що виконання проєкту завершилось, є відображення символів >>> на лівому полі вікна **IDLE Shell**.

Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви можете знайти за адресою <https://cutt.ly/8wNTwacQ> або QR-кодом.



Найважливіше в цьому пункті

Для створення, редагування та налагодження проєктів мовою програмування **Python** використовуватимемо **середовище створення проєктів IDLE** (англ. *Integrated Development and Learning Environment* – інтегроване середовище розробки та навчання), яке складається з:

- **текстового редактора** для введення та редагування тексту проєкту;
- **інтерпретатора** для перекладу команд з мови програмування **Python** на мову команд, які може виконати процесор комп'ютера;
- **засобів налагодження проєкту** для пошуку в ньому помилок;
- **довідкової системи**

та інших компонентів.

Відкрити вікно середовища **IDLE** можна, використовуючи команду **IDLE** у розділі **Python** меню **Пуск** або значок  на **Робочому столі**.

Створити нове вікно для введення тексту проєкту можна командою **File** ⇒ **New File**. Після цього відкривається вікно текстового редактора, у якому можна вводити текст проєкту мовою **Python**.

Щоб зберегти проєкт на носії, потрібно у вікні текстового редактора виконати команду **File** ⇒ **Save**. У вікні збереження файлу потрібно вибрати папку для зберігання та ввести ім'я файлу. Файл зберігається з розширенням імені **py**.

Щоб відкрити проєкт, збережений на носії, потрібно виконати команду **File** ⇒ **Open**. Після цього у вікні, що відкриється, вибрати файл з розширенням імені **py** та вибрати кнопку **Відкрити**.

Для запуску проєкту на виконання можна натиснути клавішу **F5** або виконати **Run** ⇒ **Run Module**.

Загальний вигляд команди присвоювання:

<ім'я змінної> <знак присвоювання> <значення або вираз>

У мові програмування **Python** як знак присвоювання використовують символ **=**.

Кожна змінна має **ім'я**, **значення** і **тип**.

Тип змінної визначає:

- яких значень може набувати ця змінна;
- які операції над нею можна виконувати;
- який обсяг оперативної пам'яті комп'ютера буде виділено для зберігання значення цієї змінної.

Числові вирази можуть містити:

- числа (цілі та дробові); ціла частина числа відокремлюється від дробової частини **крапкою**;
- знаки арифметичних операцій (+, −, *, /);
- дужки для змінення традиційного порядку виконання арифметичних операцій;
- спеціальні функції для виконання математичних та інших операцій, наприклад функція **abs(x)** (англ. *absolute value* – абсолютна величина числа, модуль числа), яка обчислює модуль числа.

Імена змінних у мові програмування **Python** можуть містити великі та малі українські та англійські літери, цифри та знак підкреслення (**_**). Ім'я змінної не може розпочинатися з цифри. У мові **Python** великі та малі літери розрізняються, тобто **X** і **x** – це імена різних змінних.

У проєктах мовою програмування **Python** тип змінних нефіксований. Він визначається автоматично значенням змінної та може змінюватися в ході виконання проєкту.

Змінні типу **float** можуть набувати як цілих, так і дробових значень. Змінні типу **int** можуть набувати тільки цілих значень.

Для введення значень вхідних даних під час виконання проєктів у мові програмування **Python** використовують команди:

- **<ім'я змінної> = int(input())**, **int** вказує на тип значення, яке буде надано змінній (ціле), а **input()** – ввести;
- **<ім'я змінної> = float(input())**, **float** вказує на тип значення, яке буде надано змінній (дробове).

Для виведення результатів у мові програмування **Python** використовується команда

print(<текст у лапках>, <список виразів і/або імен змінних>).

Дайте відповіді на запитання

1. Які складові частини вікна створення проєкту і для чого призначена кожна з них?
2. Як зберегти проєкт на носії?
3. Що визначає тип змінної? Які типи змінних ви знаєте?
4. Який загальний вигляд команди присвоювання в мові програмування **Python**?
5. У чому полягає лінійний запис математичних виразів? Наведіть приклади.
6. Яка команда використовується для введення значень вхідних даних?
7. Яка команда використовується для виведення результатів?

Обговоріть і зробіть висновки

1. У мові програмування **Python** тип змінних нефіксований, його можна змінювати під час виконання проєкту. А в деяких інших мовах програмування, наприклад **Object Pascal**, тип змінних фіксований, його не можна змінити під час виконання проєкту. Які, на ваш погляд, переваги та недоліки кожного з цих підходів до визначення типів змінних?

Виконайте завдання

1. Запишіть у зошит наведені вирази, використовуючи лінійний запис:
 - а) $2x - (4x - 5)(3 - 7x)$;

б) $\frac{2x - 5}{3 - 4y} - 7zy$;

в) $7,23xyz - \frac{7x - 15y}{4z - 8,23y} - x$.

2. Складіть блок-схему алгоритму обчислення значення виразу $(2x - y)(3x + 4y)$. Запишіть її в зошит. Виконайте в зошиті складений алгоритм для двох наборів значень вхідних даних. Створіть за складеним алгоритмом проєкт. Виконайте проєкт і порівняйте результати з обчисленими в зошиті. Збережіть його у вашій папці у файлі з іменем **завдання 2.2.2.py**.

3. Створіть математичну модель і складіть блок-схему алгоритму обчислення периметра прямокутника за відомими його розмірами. Запишіть їх у зошит. Виконайте в зошиті складений алгоритм для двох наборів значень вхідних даних. Створіть за складеним алгоритмом проєкт. Виконайте проєкт і порівняйте результати з обчисленими в зошиті. Збережіть його у вашій папці у файлі з іменем **завдання 2.2.3.py**.

4. Щоб дістатися до школи, Надійка 10 хв іде пішки до автобуса й потім 20 хв їде на автобусі. Створіть математичну модель і складіть блок-схему алгоритму для визначення відстані від Надійчиного будинку до школи. Запишіть їх у зошит. Виконайте в зошиті складений алгоритм для двох наборів значень вхідних даних. Створіть за складеним алгоритмом проєкт. Виконайте проєкт і порівняйте результати з обчисленими в зошиті. Збережіть його у вашій папці у файлі з іменем **завдання 2.2.4.py**.

5. Туристична група 2 год йшла рівнинною місцевістю, а потім 30 хв спускалася вниз у долину річки. Створіть математичну модель і блок-схему алгоритму для визначення відстані, яку пройшла туристична група. Запишіть їх у зошит. Виконайте складений алгоритм для двох наборів значень вхідних

даних. Створіть за складеним алгоритмом проєкт. Виконайте проєкт і порівняйте результати з обчисленими в зошиті. Збережіть його у вашій папці у файлі з іменем **завдання 2.2.5.py**.

6. Відкрийте в середовищі розробки проєктів указаний учителем/учителькою проєкт, наприклад файл **заготовка 2.2.6.py**. Визначте та запишіть у зошиті, для розв'язування якої задачі його можна використати. Запустіть його на виконання. Проаналізуйте помилки, на які вказує програма-транслятор. виправте всі помилки і знову запустіть проєкт на виконання. Збережіть виправлений проєкт у вашій папці у файлі з іменем **завдання 2.2.6.py**.



Готуємось до вивчення нового матеріалу

1. Що таке *розгалуження*? Які два види розгалуження ви знаєте?
2. На яких уроках і де саме ви використовували розгалуження?
3. Чи використовуєте ви розгалуження в повсякденному житті? Якщо так, то в яких ситуаціях?



2.3. Розгалуження.

Проєкти з розгалуженнями

У цьому пункті йтиметься про:

- ▶ команди розгалуження в мові програмування **Python**;
- ▶ проєкти з розгалуженнями.

КОМАНДИ РОЗГАЛУЖЕННЯ В МОВІ ПРОГРАМУВАННЯ PYTHON

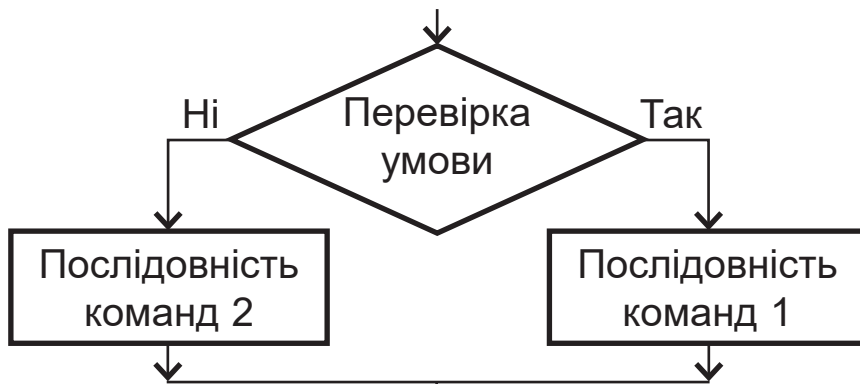


Пригадайте

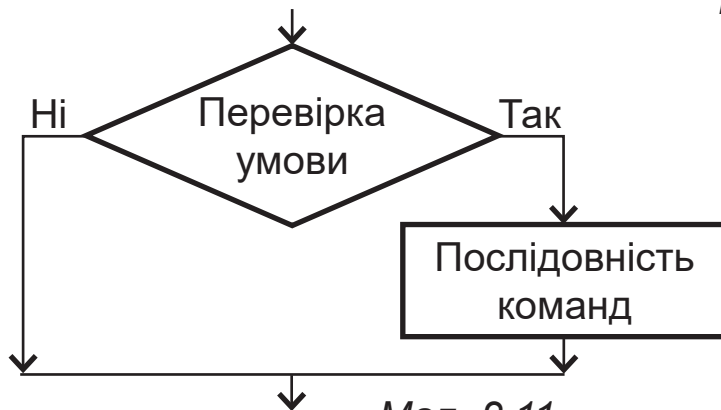
- Розгляньте малюнки 2.9–2.11 і дайте відповідь на запитання: як називаються наведені фрагменти алгоритмів?



Мал. 2.9



Мал. 2.10



Мал. 2.11

• Як виконується кожний з них? • У яких випадках ми використовуємо кожний з фрагментів?

Команда **повного розгалуження** в мові програмування **Python** має такий загальний вигляд:

```

if <перевірка умови>:
  <послідовність команд 1>
else:
  <послідовність команд 2>
  
```

(англ. *if* – якщо, *else* – інакше).

Виконання цієї команди відбувається так: **перевіряється умова**; якщо умова істинна (**Так**), то виконується **послідовність команд 1** і після цього виконується команда, наступна за розгалуженням; якщо умова хибна (**Ні**), то виконується **послідовність команд 2** і після цього виконується команда, наступна за розгалуженням.

Команда **неповного розгалуження** в мові програмування **Python** має такий загальний вигляд:

```
if <перевірка умови>:  
    <послідовність команд>
```

Виконання цієї команди відбувається так: **перевіряється умова**; якщо умова істинна (**Так**), то виконується **послідовність команд** і після цього виконується команда, наступна за розгалуженням; якщо умова хибна (**Ні**), то одразу виконується команда, наступна за розгалуженням.

Ще одним видом команди розгалуження в **Python** є команда **багатоальтернативного розгалуження**:

```
if <перевірка умови 1>:  
    <послідовність команд 1>  
elif <перевірка умови 2>:  
    <послідовність команд 2>  
elif <перевірка умови 3>:  
    <послідовність команд 3>  
...  
else:  
    <послідовність команд N>
```

Слово **elif** походить від з'єднання частини слова **else** та **if** та означає *інакше якщо*.

Виконання цієї команди відбувається так:

- **перевіряється умова 1**;
- якщо **умова 1** істинна (**Так**), то виконується **послідовність команд 1** і після цього виконується команда, наступна за розгалуженням;
- якщо **умова 1** хибна (**Ні**), то **перевіряється умова 2**;
- якщо **умова 2** істинна (**Так**), то виконується **послідовність команд 2** і після цього виконується команда, наступна за розгалуженням;

- якщо умова **2** хибна (**Hi**), то перевіряється умова **3** і т. д.
- якщо всі умови виявляться хибними, то виконується **послідовність команд N**, розміщена після **else**, і після цього виконується команда, наступна за розгалуженням.

Звертаємо вашу увагу:

- частина цієї команди **else: <послідовність команд N>** може бути відсутня;
- послідовності команд після **:** записують з відступом вправо відносно першої літери слів **if**, **elif** та **else**. Розмір відступу для кожної з команд послідовності повинен бути однаковим;
- якщо послідовності команд містять тільки по одній команді, то цю команду можна записувати після двокрапки без переходу на новий рядок.

Операції порівняння в **Python** позначають так:

<i>Операція порівняння</i>	<i>Python</i>	<i>Операція порівняння</i>	<i>Python</i>
Дорівнює	==	Більше або дорівнює	>=
Більше	>	Менше або дорівнює	<=
Менше	<	Не дорівнює	!=

РОЗВ'ЯЗУВАННЯ ЗАДАЧ З ВИКОРИСТАННЯМ РОЗГАЛУЖЕННЯ

Розглянемо задачі, алгоритми розв'язування яких містять розгалуження.

Задача 1. Мати доручила Богдану купити 2 л молока й 1 батон. Для цього вона дала сину x грн і сказала, що на решту він може купити собі порцію морозива. Чи вистачить Богдану решти, щоб купити порцію улюбленого морозива?

Складемо математичну модель для цієї задачі.

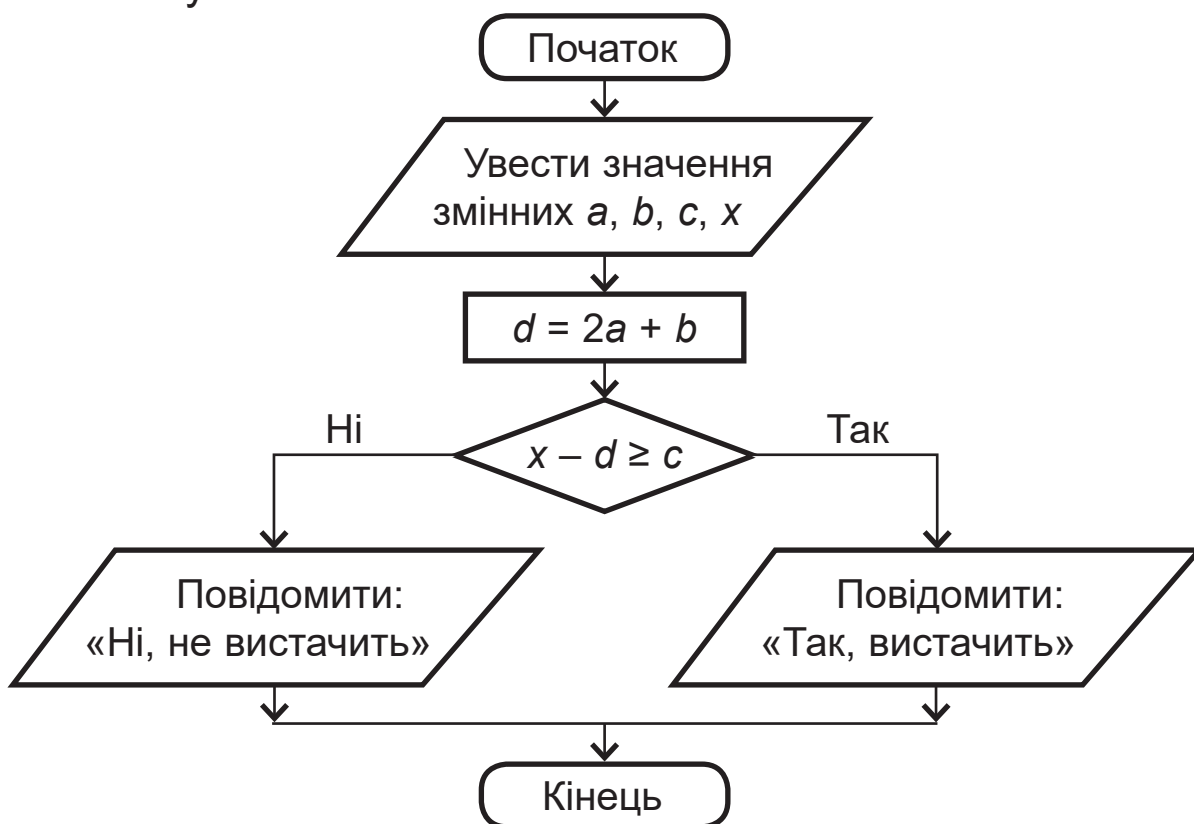
Вхідні дані: кількість літрів молока (2), ціна 1 л молока (a), кількість батонів (1), ціна 1 батона (b), ціна 1 порції улюбленого морозива (c), сума грошей, яку дала мама (x).

Кінцеві результати: повідомлення «Так, вистачить» або «Ні, не вистачить».

Проміжні результати: вартість покупки молока й батона (d).

Формули: $d = 2a + b$; якщо $x - d \geq c$, то «Так, вистачить», інакше «Ні, не вистачить».

Блок-схему алгоритму розв'язування цієї задачі наведено на малюнку 2.12.



Мал. 2.12. Блок-схема алгоритму розв'язування **Задачі 1**

Проект мовою програмування **Python** буде такий:

```
a = float(input('Уведіть ціну 1 л молока: '))  
b = float(input('Уведіть ціну 1 батона: '))
```



```

c = float(input('Уведіть ціну 1 порції улюбленого
морозива: '))
x = float(input('Уведіть суму грошей, яку дала мама: '))
d = 2*a + b;
if x - d >= c:
    print('Так, вистачить')
else:
    print('Ні, не вистачить')

```

Звертаємо вашу увагу!

1. Проєкти можна використовувати для проведення комп'ютерного експерименту.

Для *Задачі 1* можна вводити різні значення сум грошей, які дасть мама, і різні ціни 1 л молока, 1 батона й 1 порції улюбленого морозива. І залежно від цих значень отримувати різні результати виконання проєкту.

Після цього можна узагальнити отримані результати і зробити певні висновки.

2. Проєкти можна використовувати для **підтвердження або спростування гіпотези**.

Для *Задачі 1* можна висловити, наприклад, таку гіпотезу: *«Якщо мама дасть 80 грн, то Богдан зможе купити улюблене морозиво»*.

Після цього потрібно запустити проєкт на виконання, увести відповідні значення вхідних даних і отримати результат, який підтверджує або спростовує висловлену гіпотезу.

Задача 2. Обчислити значення змінної y , якщо

$$y = \begin{cases} 2x - 3, & \text{якщо } x \leq 2; \\ 3x - 2, & \text{якщо } x > 2. \end{cases}$$

Оскільки за різних значень x значення y мають обчислюватися за різними формулами, то в алгоритмі розв'язування цієї задачі потрібно використати розгалуження.

Оскільки в цій задачі формул для обчислення дві та будь-яке конкретне значення x відповідає або першій формулі, або другій, то в алгоритмі достатньо використати одне розгалуження.

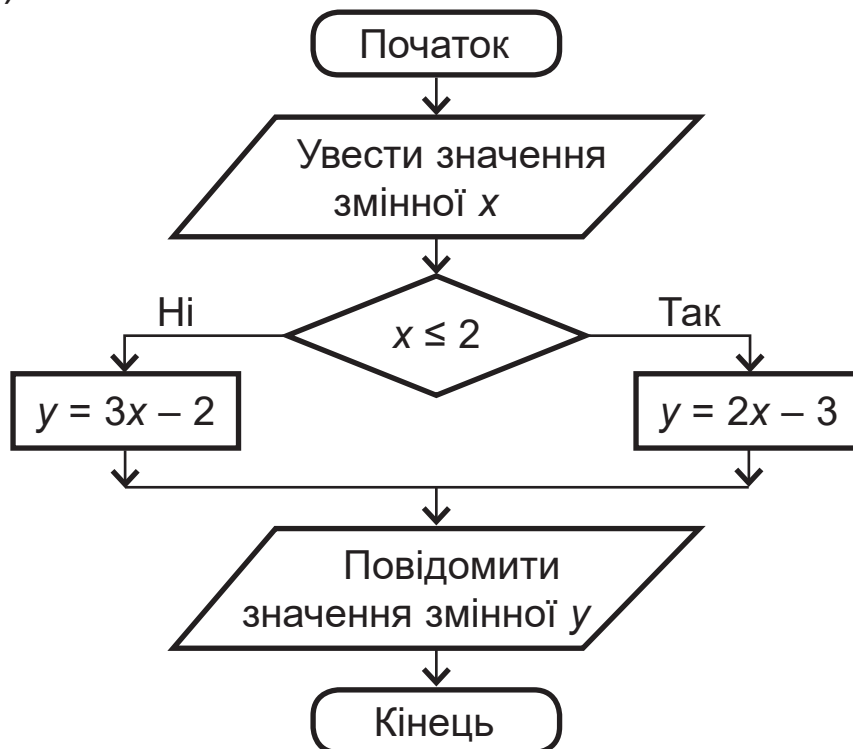
Складемо математичну модель для цієї задачі.

Вхідні дані: довільне число x .

Кінцеві результати: значення змінної y .

Формули: $y = 2x - 3$, якщо $x \leq 2$; $y = 3x - 2$, якщо $x > 2$.

Складемо блок-схему алгоритму розв'язування цієї задачі (мал. 2.13).



Мал. 2.13. Блок-схема алгоритму розв'язування **Задачі 2**

Фрагмент проекту для розв'язування цієї задачі буде такий:

```
x = float(input('Уведіть значення x: '))
if x <= 2:
    y = 2*x - 3
else:
    y = 3*x - 2
print('Значення змінної y: ', y)
```

Для перевірки правильності проєкту можна виконати його, наприклад, для таких значень: $x = -4$; $x = 0$; $x = 7$.

Для розв'язування деяких задач потрібно використати вкладені розгалуження або багатоальтернативне розгалуження. Розглянемо одну з таких задач.

Задача 3. Дано два числа. Визначити, чи рівні вони. Якщо ні, то яке з них більше?

Оскільки за різних значень x та y висновки мають бути різними, то в алгоритмі розв'язування цієї задачі потрібно використати розгалуження.

Оскільки таких випадків три (або $x = y$, або $x > y$, або $x < y$), то потрібно використати два розгалуження, друге з яких вкладено в перше.

Складемо математичну модель для цього проєкту.

Вхідні дані: два довільних числа x і y .

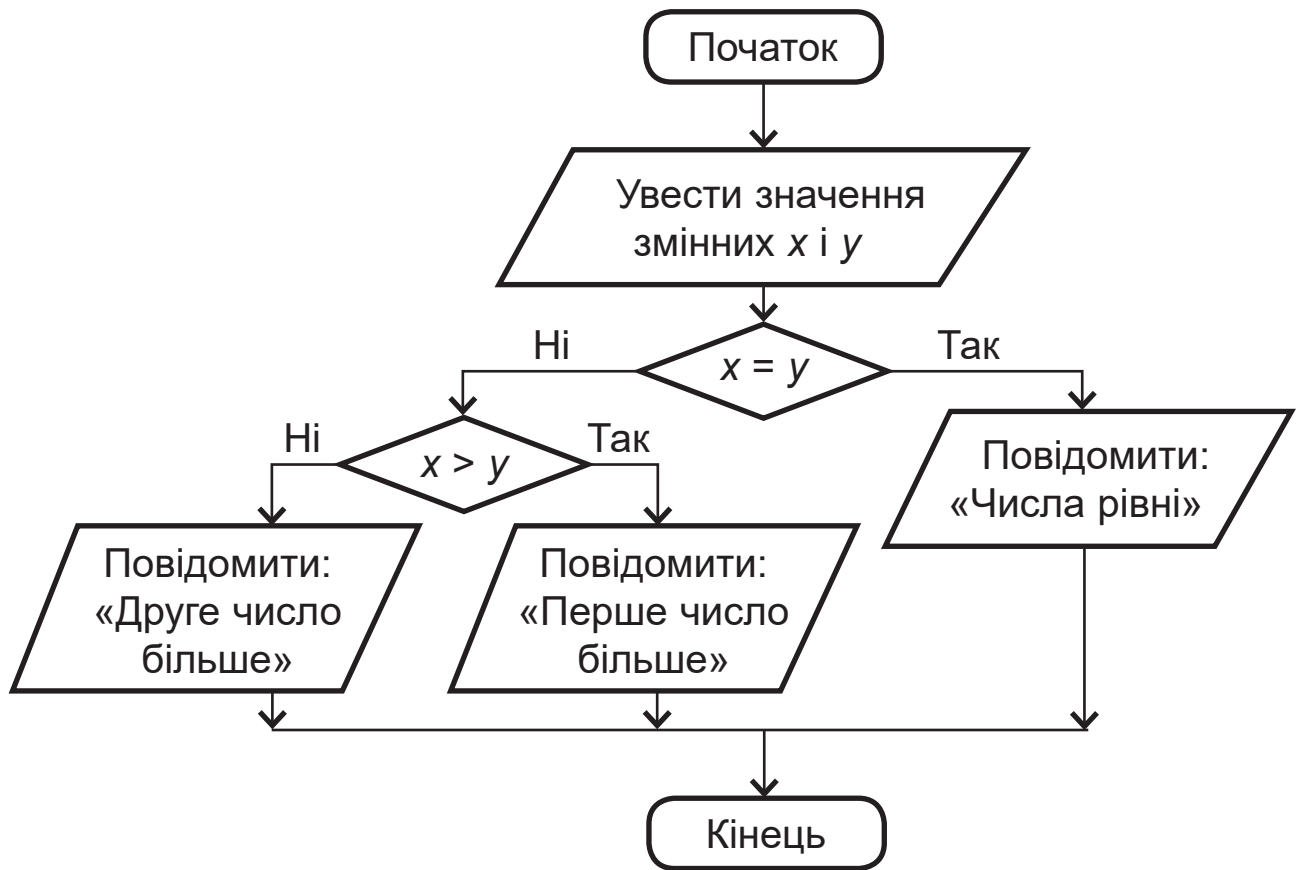
Кінцеві результати: висновок: або числа рівні, або перше число більше, або друге число більше.

Формули: «Числа рівні», якщо $x = y$; «Перше число більше», якщо $x > y$; «Друге число більше», якщо $x < y$.

Блок-схему алгоритму розв'язування цієї задачі наведено на малюнку 2.14.

Ось як виглядає фрагмент відповідного проєкту:

```
x = float(input('Уведіть x: '))
y = float(input('Уведіть y: '))
if x == y:
    print('Числа рівні')
elif x > y:
    print('Перше число більше')
else:
    print('Друге число більше')
```



Мал. 2.14. Блок-схема алгоритму розв'язування **Задачі 3**

Для перевірки правильності проєкту можна виконати його, наприклад, для таких наборів значень вхідних даних: $x = 3$, $y = 3$; $x = 5$, $y = -2$; $x = 4$, $y = 10$.

Для розв'язування деяких задач потрібно використати послідовні розгалуження. Розглянемо одну з таких задач.

Задача 4. Дано чотири нерівних між собою числа. Визначити найбільше з них.

Складемо математичну модель для цієї задачі.

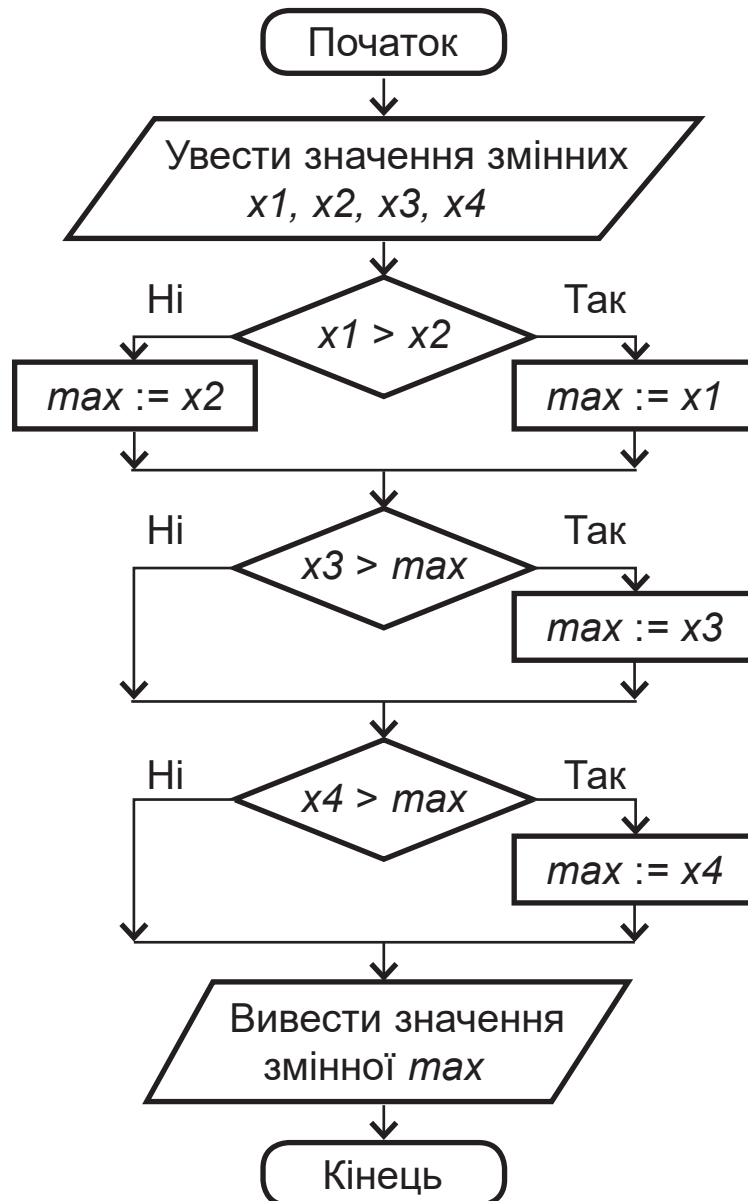
Вхідні дані: чотири довільних числа x_1 , x_2 , x_3 , x_4 .

Кінцеві результати: найбільше з цих чисел.

Формули: для розв'язування цієї задачі визначимо спочатку більше з перших двох чисел і присвоїмо це значення деякій змінній, наприклад змінній *max*; а далі порівняємо кожне

наступне число (третє і четверте) зі значенням цієї змінної; і якщо якесь число виявиться більше за значення змінної *max*, то замінимо значення цієї змінної на це число.

Блок-схему алгоритму розв'язування цієї задачі наведено на малюнку 2.15.



Мал. 2.15. Блок-схема алгоритму розв'язування **Задачі 4**

Відповідний фрагмент проєкту буде такий:

```
x1 = float(input('Уведіть x1: '))
x2 = float(input('Уведіть x2: '))
x3 = float(input('Уведіть x3: '))
x4 = float(input('Уведіть x4: '))
if x1 > x2: max = x1
else: max = x2
if x3 > max: max = x3
if x4 > max: max = x4
print('Найбільше значення: ', max)
```

Звертаємо вашу увагу: даний проєкт працюватиме правильно й у випадках, коли серед цих чотирьох чисел будуть рівні між собою числа.



Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви можете знайти за адресою <https://cutt.ly/iwNTwUO1> або QR-кодом.



Найважливіше в цьому пункті

Команда **повного розгалуження** в мові програмування **Python** має такий загальний вигляд:

```
if <перевірка умови>:
    <послідовність команд 1>
else:
    <послідовність команд 2> (англ. if – якщо, else – інакше).
```

Виконання цих команд відбувається так: **перевіряється умова**; якщо умова істинна (**Так**), то виконується **послідовність команд 1** і після цього виконується команда, наступна за розгалуженням; якщо умова хибна (**Ні**), то виконується **послідовність команд 2** і після цього виконується команда, наступна за розгалуженням.

Команда **неповного розгалуження** в мові програмування **Python** має такий загальний вигляд:

```
if <перевірка умови>:
    <послідовність команд>
```

Виконання цієї команди відбувається так: **перевіряється умова**; якщо умова істинна (**Так**), то виконується **послідовність команд** і після цього виконується команда, наступна за розгалуженням; якщо умова хибна (**Ні**), то одразу виконується команда, наступна за розгалуженням.

Ще одним видом команди розгалуження в **Python** є команда **багатоальтернативного розгалуження**:

```
if <перевірка умови 1>:  
    <послідовність команд 1>  
elif <перевірка умови 2>:  
    <послідовність команд 2>  
elif <перевірка умови 3>:  
    <послідовність команд 3>  
...  
else:  
    <послідовність команд N>
```

Слово **elif** походить від з'єднання частини слова **else** та **if** і означає *інакше якщо*.

Виконання цієї команди відбувається так:

- **перевіряється умова 1**;
- якщо **умова 1** істинна (**Так**), то виконується **послідовність команд 1** і після цього виконується команда, наступна за розгалуженням;
- якщо **умова 1** хибна (**Ні**), то **перевіряється умова 2**;
- якщо **умова 2** істинна (**Так**), то виконується **послідовність команд 2** і після цього виконується команда, наступна за розгалуженням;
- якщо **умова 2** хибна (**Ні**), то **перевіряється умова 3** і т. д.;
- якщо всі умови виявляться хибними, то виконується **послідовність команд N**, розміщена після **else**, і після цього виконується команда, наступна за розгалуженням.

Звертаємо вашу увагу:

- частина цієї команди **else: <послідовність команд N>** може бути відсутня;

- послідовності команд після **:** записують з відступом вправо відносно першої літери слів **if**, **elif** та **else**. Розмір відступу для кожної з команд послідовності повинен бути однаковим;
- якщо послідовності команд містять тільки по одній команді, то цю команду можна записувати після двокрапки без переходу на новий рядок.

Операції порівняння в мові програмування **Python** позначають так:

<i>Операція порівняння</i>	<i>Python</i>
Дорівнює	<code>==</code>
Більше	<code>></code>
Менше	<code><</code>
Більше або дорівнює	<code>>=</code>
Менше або дорівнює	<code><=</code>
Не дорівнює	<code>!=</code>

Дайте відповіді на запитання

1. У яких випадках алгоритм розв'язування задачі повинен містити розгалуження?

2. Який загальний вигляд команди повного розгалуження в мові програмування **Python**? Як вона виконується?

3. Який загальний вигляд команди неповного розгалуження в мові програмування **Python**? Як вона виконується?

4. Який загальний вигляд команди багатоальтернативного розгалуження? Як вона виконується?

Обговоріть і зробіть висновки

1. Чи відрізняється виконання наведених фрагментів проєктів? Якщо так, то чим саме?

Фрагмент 1

```
if <перевірка умови 1>:
    <послідовність команд 1>
```



```
else:  
    if <перевірка умови 2>:  
        <послідовність команд 2>  
    else:  
        <послідовність команд 3>  
                                Фрагмент 2
```


```
if <перевірка умови 1>:  
    <послідовність команд 1>  
else:  
    <послідовність команд 2>  
if <перевірка умови 2>:  
    <послідовність команд 3>  
else:  
    <послідовність команд 4>
```

2. У яких випадках два послідовних розгалуження можна замінити на вкладені розгалуження?

3. Скільки потрібно команд вкладених одне в одне розгалужень, якщо потрібно врахувати 5 можливих варіантів? Поясніть свою відповідь.

Виконайте завдання

1. Запишіть у зошит приклади життєвих ситуацій, які можна описати алгоритмом з розгалуженням. Зобразіть ці алгоритми у вигляді блок-схем.

 2. Запишіть у зошит приклади правил з української мови, математики, англійської мови, інших предметів, що містять розгалуження. Зобразіть ці правила у вигляді блок-схем.

3. Створіть математичну модель і проєкт, який обчислює модуль довільного числа. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.3.3.py**. Запишіть у зошиті значення вхідних даних, для яких ви тестували цей проєкт.

4. Створіть математичну модель і проєкт, який визначає менше з двох довільних дійсних чисел або виводить повідомлення, що числа рівні. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.3.4.py**. Запишіть у зошиті набори значень вхідних даних, для яких ви тестували цей проєкт.

5. Автоматизована система продажу авіаквитків запропонувала 5 варіантів перельоту з міста *A* до міста *B*. Створіть математичну модель і проєкт, який визначає найдешевший із запропонованих перельотів. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.3.5.py**.


6. Створіть математичну модель і проєкт, який визначає найменше з п'яти нерівних дійсних чисел. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.3.6.py**.


7. Створіть математичну модель і проєкт, який визначає найменше та найбільше з шести нерівних дійсних чисел. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.3.7.py**.


8. Відомо, що з трьох відрізків можна побудувати трикутник, якщо довжина більшого з цих відрізків менша, ніж сума довжин двох інших. Створіть математичну модель і проєкт для визначення, чи можна з даних трьох відрізків побудувати трикутник. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.3.8.py**. Запишіть у зошиті певну гіпотезу щодо взятих трьох відрізків. Використайте проєкт для перевірки цієї гіпотези.

9. До 1 липня 2023 року вартість витраченої електроенергії для населення України визначалася за таким правилом. Якщо споживач за місяць спожив не більше $100 \text{ кВт} \cdot \text{год}$ електроенергії, то потрібно заплатити 0,9 грн за кожну спожиту $\text{кВт} \cdot \text{год}$ електроенергії. Якщо споживач спожив за місяць понад $100 \text{ кВт} \cdot \text{год}$ електроенергії, то за перші $100 \text{ кВт} \cdot \text{год}$

потрібно сплатити 90 грн, а за кожну кВт · год спожитої електроенергії понад 100 кВт · год потрібно заплатити по 1,68 грн. Створіть математичну модель і проєкт для визначення, яку суму грошей має сплатити споживач за спожиту електроенергію. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.3.9.ру**.

 **10. Працюємо в парах.** За 1 м³ спожитого газу в Україні потрібно заплатити 7,96 гривні, а в Польщі – 1,61 злотого і ще й 9 злотих на місяць абонплати. Створіть математичну модель і проєкт для порівняння оплати за один і той самий об'єм спожитого газу в Україні та в Польщі. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.3.10.ру**.

 **11. Працюємо в парах.** Банк пропонує вкладнику два види депозитів: 15 % прибутку за рік і 7 % прибутку за пів року. В обох випадках після завершення терміну нарахований прибуток додається до вкладу. Створіть математичну модель і проєкт для порівняння прибутку за 2 роки за цими депозитами, якщо на обидва покласти одну й ту саму суму грошей. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.3.11.ру**. Сформулюйте і запишіть у зошит гіпотезу щодо того, який з цих двох депозитів вигідніший. Використайте проєкт для перевірки сформульованої гіпотези.

 **Готуємось до вивчення нового матеріалу**

1. У яких випадках в алгоритмах і проєктах використовують розгалуження?
2. Які види розгалужень ви знаєте? У яких випадках потрібно використовувати кожне з них?
3. У яких випадках потрібно використовувати послідовні розгалуження, а в яких – вкладені?

2.4. Підпрограми

У цьому пункті йтиметься про:

- ▶ підпрограми;
- ▶ процедури;
- ▶ функції.



Пригадайте

- Алгоритм знаходження найбільшого із заданих 4 чисел.

На попередніх уроках ви складали алгоритми для знаходження найбільшого та найменшого значень з кількох чисел. І яка б не була кількість цих чисел, все одно у процесі виконання алгоритмів усе зводилося до знаходження найбільшого або найменшого з певних двох чисел.

А от уявіть собі, що в мові програмування є команди **Знайти найбільше з даних двох чисел** і **Знайти найменше з даних двох чисел**. Наскільки б у такому разі спростився алгоритм знаходження, наприклад, найбільшого з 4 чисел!

Він би містив, крім команд введення і виведення, лише три прості команди:

1. **Знайти найбільше з перших двох чисел.**
2. **Знайти найбільше з числа, знайденого в команді 1, і третього числа.**
3. **Знайти найбільше з числа, знайденого в команді 2, і четвертого числа.**



Поміркуйте

- Складіть алгоритм знаходження найбільшого з 4 чисел, відмінний від вищенаведеного.

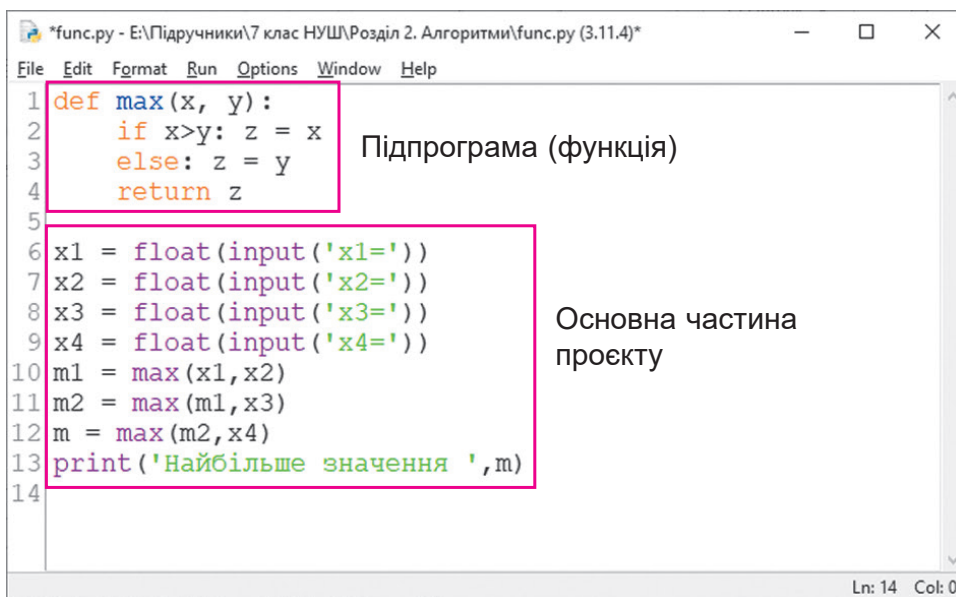
Сучасні мови програмування надають можливість користувачеві **створювати (задавати) свої команди**, яких немає в наборі команд цієї мови програмування, і:

- 1) надавати їм імена;
- 2) задавати, що вони будуть виконувати;
- 3) викликати ці команди на виконання будь-яку кількість разів у будь-якому місці проєкту.

Тобто будь-який користувач може доповнити множину команд мови програмування своїми командами, які він надалі зможе використовувати для розв'язування різноманітних задач. Кожну з таких команд користувач повинен створити (задати) як *підпрограму*.

Підпрограма – це окрема частина програми, яка має ім'я, містить команди і може бути викликана на виконання в будь-якому місці програми.

Розглянемо як приклад спосіб створення (задання) команди (підпрограми) **Знайти найбільше з двох чисел** і відповідний проєкт з використанням цієї команди (мал. 2.16).



```
*func.py - E:\Підручники\7 клас НУШ\Розділ 2. Алгоритми\func.py (3.11.4)*
File Edit Format Run Options Window Help
1 def max(x, y):
2     if x>y: z = x
3     else: z = y
4     return z
5
6 x1 = float(input('x1='))
7 x2 = float(input('x2='))
8 x3 = float(input('x3='))
9 x4 = float(input('x4='))
10 m1 = max(x1, x2)
11 m2 = max(m1, x3)
12 m = max(m2, x4)
13 print('Найбільше значення ', m)
14
```

Підпрограма (функція)

Основна частина проєкту

Ln: 14 Col: 0

Мал. 2.16. Проєкт знаходження більшого з 4 чисел з підпрограмою для знаходження найбільшого з двох чисел

Підпрограму, яка повертає результат у команду, у якій вона викликається, називають **функцією**.

Загальний вигляд функції в мові програмування **Python** такий:

```
def <ім'я> (<список імен змінних-аргументів>) :  
    <команди>  
    return <результат>
```

Рядок **def** <ім'я> (<список імен змінних-аргументів>) : називається **рядком заголовка функції** (англ. *definition* – визначення).

У наведеному прикладі ім'я функції **max**, аргументів два: **x** і **y**.

Команди функції повинні бути записані з однаковим відступом управо від позиції першої літери слова **def**.

Команда **return** <результат> (англ. *return* – повернення) завершує виконання функції та повертає значення результату в команду виклику функції. Як результат може бути вказано число, або ім'я змінної з певним значенням, або вираз, значення якого після обчислення буде результатом виконання функції.

У наведеному прикладі результатом функції є значення змінної **z**, яке буде повернуто в команду виклику функції.

У тексті проєкту функцію можна розміщувати в будь-якій його частині, але обов'язково перед командою, у якій вона викликається. Доцільно розміщувати функції на початку проєкту. Текст основної частини проєкту записують після останньої команди функції без відступу від лівої межі вікна текстового редактора.

Після запуску проєкту на виконання спочатку розпочнеться виконання основної частини проєкту, а не підпрограми, хоча вона й записана вище. Підпрограма почне виконуватися тільки після того, як буде викликана командою її виклику.

Функції викликаються у правій частині команди присвоєння або в команді виведення.

У наведеному прикладі перший раз ця функція викликається на виконання командою $m1 = \max(x1, x2)$. Виконується ця команда так:

1. Значення змінних $x1$ і $x2$ передаються змінним-аргументам функції x і y . Тобто по суті виконуються дві команди присвоювання: $x = x1$ і $y = x2$.

2. Виконуються команди функції зі значеннями змінних x і y .

3. Значення змінної z є результатом функції. Воно повертається в команду виклику і присвоюється змінній $m1$.

Так, наприклад, якщо введемо перші два числа **5** і **10**, то виконання функції буде таким:

<i>Команда</i>	<i>Виконання</i>
$x = x1$ $y = x2$	$x = 5$ $y = 10$
$x > y$	$5 > 10$ – Ні
$z = y$	$z = 10$
$m1 = z$	$m1 = 10$

Другий раз ця функція викликається на виконання командою $m2 = \max(m1, x3)$. Виконується ця команда так:

1. Значення змінних $m1$ і $x3$ передаються змінним-аргументам функції x і y . Тобто по суті виконуються дві команди присвоювання: $x = m1$ і $y = x3$.

2. Виконуються команди функції зі значеннями змінних x і y .

3. Значення змінної z як результат функції повертається в команду виклику і присвоюється змінній $m2$.

Так, наприклад, якщо введемо третє число **1**, то виконання функції буде таким:

<i>Команда</i>	<i>Виконання</i>
$x = m1$ $y = x3$	$x = 10$ $y = 1$
$x > y$	$10 > 1$ – Так
$z = x$	$z = 10$
$m2 = z$	$m2 = 10$

Третій раз ця функція викликається на виконання командою $m = \max(m2, x4)$. Виконується ця команда так:

1. Значення змінних $m2$ і $x4$ передаються змінним-аргументам функції x і y . Тобто по суті виконуються дві команди присвоювання: $x = m2$ і $y = x4$.

2. Виконуються команди функції зі значеннями змінних x і y .

3. Значення змінної z як результат функції повертається в команду виклику і присвоюється змінній m .

Так, наприклад, якщо введемо четверте число **15**, то виконання функції буде таким:

<i>Команда</i>	<i>Виконання</i>
$x = m2$ $y = x4$	$x = 10$ $y = 15$
$x > y$	$10 > 15$ – Ні
$z = y$	$z = 15$
$m = z$	$m = 15$

Загальний вигляд команди виклику функції в команді присвоювання такий:

<ім'я змінної> = <ім'я функції> (<список конкретних значень і/або імен змінних, які мають певні значення>)

Наприклад, $c = \max(14, x1)$, $k = \max(7.5, -12)$.

У команді виведення виклик функції може бути, наприклад, таким: `print('Найбільшим з чисел x і 15 є число: ', max(x, 15))`.

Звертаємо вашу увагу!

1. У команді виклику функції кількість конкретних значень або імен змінних повинна збігатися з кількістю аргументів функції. Тобто неправильно є команда виклику функції, наприклад, `c = max(14, x1, x2)`.

2. У функції може бути один аргумент, або кілька аргументів, або зовсім не бути аргументів.

3. Функція може повертати як результат одне значення або кілька значень.

4. Якщо підпрограма не повертає результати, тобто не має команди `return`, то таку підпрограму називають **процедурою**.

5. У проєкті можуть бути кілька функцій, а також і функції, і процедури.

Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви можете знайти за адресою <https://cutt.ly/uwNTw9ui> або QR-кодом.



Найважливіше в цьому пункті

Підпрограма – це окрема частина програми, яка має ім'я, містить команди і може бути викликана на виконання в будь-якому місці програми.

Підпрограму, яка повертає результат у команду, у якій вона викликається, називають **функцією**.

Загальний вигляд функції в мові програмування **Python** такий:

```
def <ім'я> (<список імен змінних-аргументів>) :
```

```
    <команди>
```

```
    return <результат>
```

Рядок `def <ім'я> (<список імен змінних-аргументів>)` : називається **рядком заголовка функції** (англ. *definition* – визначення).

Команда **return** <результат> (англ. *return* – повернення) завершує виконання функції та повертає значення результату в команду виклику функції. Як результат може бути вказано число, або ім'я змінної з певним значенням, або вираз, значення якого після обчислення буде результатом виконання функції.

Загальний вигляд команди виклику функції в команді присвоєння такий:

<ім'я змінної> = <ім'я функції> (<список конкретних значень і/ або імен змінних, які мають певні значення>)

Наприклад, **c = max(14, x1), k = max(7.5, -12).**

Функція може повертати як результат одне значення або кілька значень.

Якщо підпрограма не повертає результати, тобто не має команди **return**, то таку підпрограму називають **процедурою**.

Дайте відповіді на запитання

1. Що таке *підпрограма*?
2. Яку підпрограму називають *функцією*?
3. Які складові частини рядка заголовка функції?
4. Якою має бути команда виклику функції?
5. Яку підпрограму називають *процедурою*?

Обговоріть і зробіть висновки

1. У яких випадках доцільно використовувати процедури, а в яких – функції?
2. Чи можна викликати процедуру в команді присвоєння, аналогічно до виклику функції? Поясніть, чому. Як, на вашу думку, має виглядати команда виклику процедури?

Виконайте завдання

1. Запишіть у зошит виконання наведених фрагментів проєктів:

```
def perim(x, y, z):  
    t = x + y + z  
    return t
```

а) `p = perim(2, 6, 5)`

б) `a = 16.5; b = 22`
`p = perim(a, b, 10)`

в) `a = 19; b = 16; c = 17`
`print(perim(a, b, c))`

2. Створіть проєкт для знаходження суми модулів трьох чисел, використавши функцію для знаходження модуля числа. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.4.2.py**.

3. Створіть проєкт для знаходження найменшого серед шести нерівних між собою чисел, використовуючи функцію знаходження найменшого значення з двох чисел. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.4.3.py**. Висловіть гіпотезу, чи правильно працюватиме ваш проєкт, якщо серед цих шести чисел будуть рівні між собою. Перевірте вашу гіпотезу, використавши створений проєкт.

4. Створіть проєкт для знаходження периметра та площі трьох прямокутників, якщо відомо довжини їх сторін, використовуючи функції для знаходження периметра та площі прямокутника за довжинами його сторін. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.4.4.py**.

5. Створіть проєкт, у якому потрібно знайти суму довжин трьох кіл за відомими довжинами їх радіусів, або суму периметрів трьох квадратів, або суму периметрів трьох трикутників за відомими довжинами їх сторін, залежно від того, яке число (1, 2 або 3) введено на початку виконання проєкту. У проєкті використайте функції. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.4.5.py**.



Практична робота № 3.

«Проекти з підпрограмами та з розгалуженнями»

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

Задача. Створіть проект, у якому потрібно знайти суму площ двох кругів за відомими довжинами їх радіусів або суму площ двох квадратів за відомими довжинами їх сторін, залежно від того, яке число (1 або 2) введено на початку виконання проекту. У проекті використайте функцію для обчислення площі круга і площі квадрата. Порівняйте ці суми. Якщо вони різні, визначте наскільки. Збережіть проект у вашій папці у файлі з іменем **практична 3.py**.



Готуємось до вивчення нового матеріалу

1. Що таке *вікно програми*?
2. З вікнами яких програм ви працювали?
3. Які елементи керування ви використовували у вікнах програм?



2.5. Проекти з вікнами

У цьому пункті йтиметься про:

- ▶ створення проєктів з вікнами;
- ▶ властивості вікон та їх значення.

СТВОРЕННЯ ВІКОННИХ ПРОЄКТІВ



Поміркуйте

• Порівняйте проєкти з використанням вікна введення-виведення, які ви створювали на попередніх уроках, з проєктами з використанням вікон, на яких розміщено елементи керування (кнопки, поля, прапорці та ін.). Які їх переваги та недоліки?

Для створення проєкту з вікнами мовою **Python** потрібно приєднати до проєкту модуль **tkinter**, що входить до стандартної бібліотеки модулів **Python**. Модуль приєднується до проєкту виконанням команди **from tkinter import *** (англ. *from* – з, *import* – імпортувати, ввозити).

Для створення вікна потрібно вибрати для нього ім'я та виконати команду: **<ім'я_вікна> = Tk()**. Головному вікну проєкту часто надають ім'я **root** (англ. *root* – корінь). Далі в командах будемо використовувати саме це ім'я для вікна.

Команда створення вікна з іменем **root** повинна бути такою: **root = Tk()**.

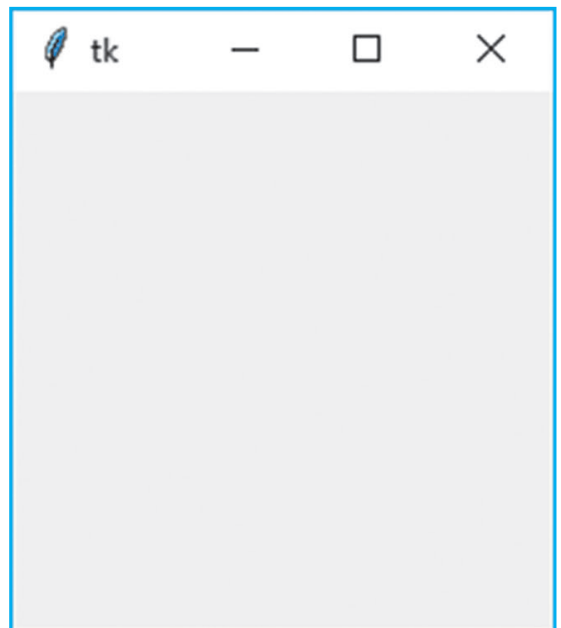
Звертаємо вашу увагу!

Для виконання віконних проєктів у онлайн-середовищах, наприклад на сайтах **Replit** (replit.com) або **CodeHS** (codehs.com), останньою у проєкті повинна бути команда очікування подій у вікні: **root.mainloop()**.

Після запуску проєкту та виконання цієї команди відкривається вікно, вигляд якого наведено на малюнку 2.17. За замовчуванням вікно відкривається у верхньому лівому куті екрана, має розміри 200 на 200 пікселів, рядок заголовка вікна містить напис *tk*, колір фону – світло-сірий.

Значення всіх властивостей вікна можна змінити.

Для змінення напису в рядку заголовка вікна з іменем **root** потрібно виконати команду **root.title('<текст>')** (англ. *title* – за-



Мал. 2.17. Вікно зі значеннями властивостей, установленними за замовчуванням

головок, назва). Наприклад, команда ***root.title('Мій перший проєкт')*** виводить у рядок заголовка вікна напис *Мій перший проєкт*.

Для змінення розмірів і положення вікна потрібно використати команду ***root.geometry('<рядок параметрів>')*** (англ. *geometry* – геометрія). У рядку параметрів можна вказати: ширину та висоту вікна, між якими стоїть символ «x», відстань лівої межі вікна від лівої межі екрана та відстань верхньої межі вікна від верхньої межі екрана, перед якими стоять символи «+». Так, команда ***root.geometry('600x400+200+100')*** установить ширину вікна 600 пікселів, висоту вікна – 400 пікселів, відстань лівої межі вікна від лівої межі екрана – 200 пікселів, відстань верхньої межі вікна від верхньої межі екрана – 100 пікселів.

Звертаємо вашу увагу!

У рядку параметрів, що визначає розміри та положення вікна, відсутні пропуски.

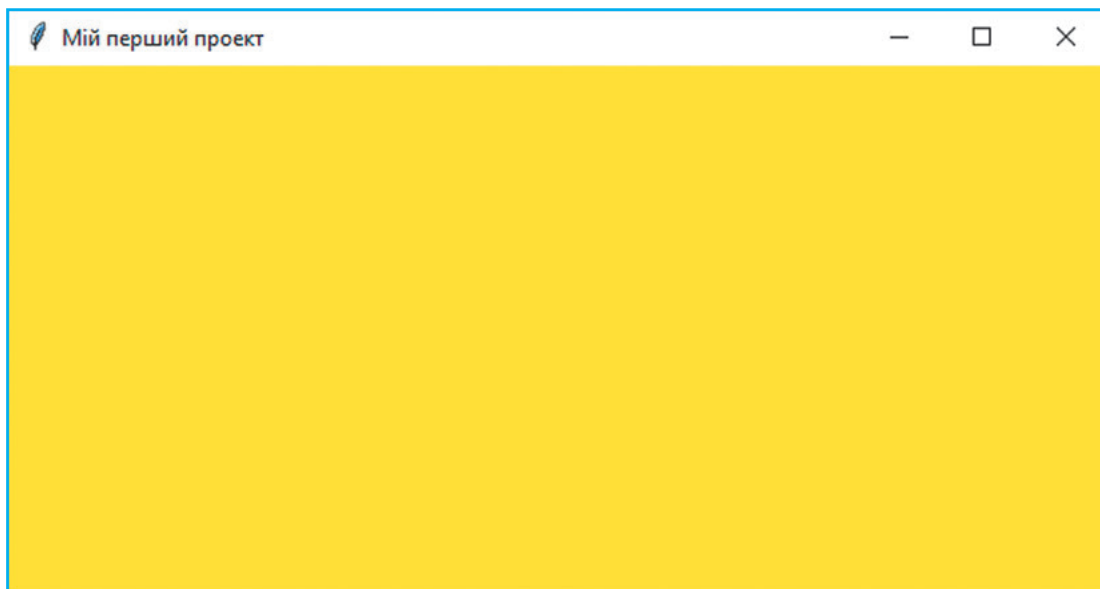
Команда ***root.geometry('400x250')*** установить ширину вікна 400 пікселів і висоту вікна – 250 пікселів, а положення вікна не змінить. А якщо виконати команду ***root.geometry('+300+200')***, то розміри вікна не зміняться, а зміняться лише відстані вікна від лівої та верхньої межі екрана.

Для змінення кольору фону вікна потрібно вказати назву кольору в команді ***root['bg']='<колір>'*** (англ. *background* – фон, тло). Наприклад, після виконання команди ***root['bg']='yellow'*** (англ. *yellow* – жовтий) колір фону вікна стане жовтим.

На малюнку 2.18 наведено вигляд вікна після виконання команд:

```
from tkinter import *
root=Tk()
root.title('Мій перший проєкт')
```

```
root.geometry('600x400+200+100')
root['bg']='yellow'
```



Мал. 2.18. Вигляд вікна після змінення напису в рядку заголовка, розмірів, положення та кольору фону

Для тих, хто хоче знати більше

Якщо команда `root.mainloop()` буде використана у проєктах, створених у середовищі розробки проєктів **IDLE**, то вікно виконання проєкту можна буде відкрити, двічі клацнувши у вікні **Провідник** на файлі, у якому зберігається проєкт.

ЗАСОБИ СТВОРЕННЯ ТА РЕДАГУВАННЯ ТЕКСТУ ПРОЄКТУ

Пригадайте

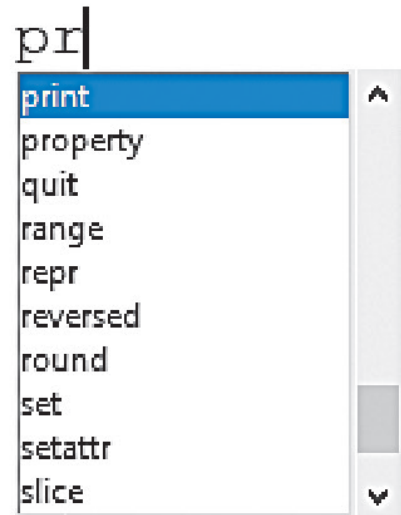
- Що таке *синтаксична помилка*?
- Що відбувається, якщо транслятор виявив у проєкті синтаксичні помилки?
- Що повинен робити автор проєкту за наявності в ньому синтаксичних помилок?

У віконних проєктах, так само як і в проєктах з вікном введення-виведення, за наявності в тексті проєкту синтаксичних помилок транслятор перериває його виконання та повідом-

ляє про них. Усі синтаксичні помилки потрібно виправити, після чого знову запустити проєкт на виконання.

Середовище **IDLE** має деякі засоби допомоги під час написання тексту проєкту:

- після введення першої або кількох перших літер ключового слова можна натиснути сполучення клавіш **Ctrl + Пропуск**, і відкриється список, з якого можна вибрати потрібне ключове слово (мал. 2.19);
- після введення імені функції та відкриваючої дужки автоматично виводиться підказка, які аргументи можуть бути у цієї функції (мал. 2.20);



Мал. 2.19. Список для вибору ключового слова

```
print(  
print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
```

Мал. 2.20. Підказка з можливими аргументами функції

- після введення у виразі закриваючої дужки на секунду підсвічується частина виразу до відповідної відкриваючої дужки, щоб можна було перевірити, чи всі дужки у виразі закриті (мал. 2.21);

```
a = int(input('Уведіть сторону квадрата '))
```

Мал. 2.21. Підсвічування частини виразу між відповідними дужками

- система автоматично розставляє в потрібних місцях тексту проєкту відступи після рядків, що завершуються символом *двокрапка* (:)

та інші.



Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви можете знайти за адресою <https://cutt.ly/HwNTeiLp> або QR-кодом.



Найважливіше в цьому пункті

У мові **Python** для створення проєкту з вікном потрібно приєднати до проєкту модуль **tkinter**, використовуючи команду **from tkinter import ***. Команда створення вікна з іменем **root** є такою: **root = Tk()**.

Для змінення напису в рядку заголовка вікна з іменем **root** потрібно виконати команду **root.title('<текст>')**. Змінення розмірів і положення вікна виконує команда **root.geometry('<рядок параметрів>')**. Для змінення кольору фону вікна потрібно вказати назву кольору в команді **root['bg']='<колір>'**.



Дайте відповіді на запитання

1. Як створити проєкт з вікнами?
2. Як змінити напис у рядку заголовка вікна?
3. Як змінити розміри вікна і його положення на екрані?
4. Як змінити колір фону вікна?



Обговоріть і зробіть висновки

1. Які переваги та недоліки віконних проєктів порівняно з проєктами, у яких використовується тільки вікно введення-виведення?

2. У чому переваги та недоліки середовищ візуальної розробки проєктів порівняно із середовищами невізуальної розробки?



Виконайте завдання

1. Створіть проєкт, виконання якого відкриє вікно червоного кольору розмірами 400 на 350 пікселів і змінить текст у рядку заголовка вікна на ваше прізвище. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.5.1.py**.

2. Створіть проєкт, виконання якого відкриє вікно синього кольору розмірами 200 на 450 пікселів і змінить текст у рядку заголовка вікна на ваше ім'я. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.5.2.py**.

3. Створіть проєкт, виконання якого відкриє вікно синього кольору розмірами 300 на 300 пікселів, ліва межа якого розміщена на відстані 300 пікселів від лівої межі екрана і верхня межа – на відстані 400 пікселів від верхньої межі екрана. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.5.3.py**.

4. Створіть проєкт, виконання якого відкриє вікно сірого кольору розмірами 200 на 100 пікселів, ліва межа якого розміщена на відстані 150 пікселів від лівої межі екрана і верхня межа – на відстані 100 пікселів від верхньої межі екрана. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.5.4.py**.



Готуємось до вивчення нового матеріалу

1. Що таке *вікно програми*?
2. З вікнами яких програм ви працювали?
3. Які елементи керування ви використовували у вікнах програм?



2.6. Події. Обробники подій.

Вікна повідомлень

У цьому пункті йтиметься про:

- ▶ події, що можуть відбуватися з вікном;
- ▶ обробники подій;
- ▶ обробники подій з вікном;
- ▶ вікна повідомлень;
- ▶ віконні проєкти з використанням подій та їх обробників.

ПОДІЇ ТА ЇХ ОБРОБНИКИ



Пригадайте

- Що таке *підпрограма*?
- Яка підпрограма називається *функцією*, а яка *процедурою*?
- Що таке *аргументи функції*, *результати функції*?
- Як викликається функція на виконання?
- Які властивості вікна ви знаєте? Як ви можете змінити їх значення?



Поміркуйте

- Які події можуть відбуватися з вікном проекту під час його виконання?
- Що може відбуватися з вікном у результаті настання певної події?

З вікном під час виконання віконного проекту можуть відбуватися **події**.

Такими подіями, наприклад, можуть бути:

- **Click** (англ. *click* – клацання) – вибір будь-якої точки вікна;
- **DbIClick** (англ. *double* – подвійний) – подвійне клацання в будь-якій точці вікна;
- **Close** (англ. *close* – закрити) – закривання вікна;
- **Create** (англ. *create* – створення) – створення вікна;
- **KeyPress** (англ. *key press* – натиснення клавіші) – натиснення будь-якої клавіші на клавіатурі;
- **MouseMove** (англ. *mouse move* – переміщення миші) – переміщення вказівника в межах вікна виконання проекту та інші.

З будь-якою подією, яка може відбутися з вікном, можна пов'язати **підпрограму**, яка буде виконуватися одразу після настання цієї події. Така підпрограма називається **обробник події**.

У проекті можуть бути кілька обробників різних подій.

Підпрограма, яка пов'язана з певним об'єктом, називається **методом цього об'єкта**.

У мові **Python** подію з вікном пов'язують з підпрограмою – обробником цієї події, використовуючи метод **bind** (англ. *bind* – в'язати, пов'язувати).

У загальному вигляді команда пов'язування події з вікном, ім'я якого *root*, з обробником цієї події має такий вигляд:

```
root.bind('<подія>', <ім'я_обробника_події>)
```

Позначення у мові **Python** деяких подій, які можуть відбутися з вікном, наведено в таблиці 2.2.

Таблиця 2.2

Позначення подій у мові Python

<i>Подія</i>	<i>Позначення в мові Python</i>
Click – клацання лівої кнопки миші	<Button-1> або <1>
DbClick – подвійне клацання лівої кнопки миші	<Double-Button-1> або <Double-1>
MouseMove – рух вказівника в межах вікна	<Motion> (англ. <i>motion</i> – рух)
KeyPress – натискання будь-якої клавіші на клавіатурі	<KeyPress>
Create – створення вікна	<Expose> (англ. <i>expose</i> – розкривати)
Close – закривання вікна	<Destroy> (англ. <i>destroy</i> – знищити)

Наприклад, якщо обробнику події **Click** (клацання лівої кнопки миші) надати ім'я *click*, то команда пов'язування цієї процедури з вікном з іменем *root* має такий вигляд:

```
root.bind('<1>', click).
```

Загальний вигляд обробника події такий:

```
def <ім'я_обробника_події> (event):
```

```
    <команди, які будуть виконуватися за настання події>  
    (англ. event – подія).
```

Для обробника події з іменем *click* рядок заголовка буде таким: **def click (event):**

Якщо потрібно, щоб у результаті настання події змінилися значення певних властивостей вікна, обробник події повинен містити команди змінення значень цих властивостей.

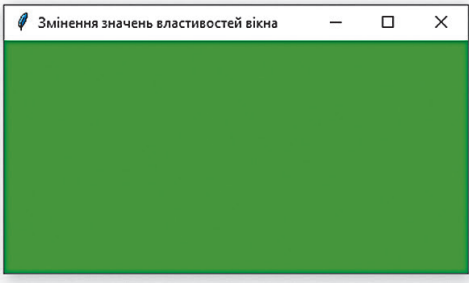
Значення властивостей вікна можна змінити:

- **командою присвоювання**, загальний вигляд якої такий: **<ім'я вікна>[<властивість>] = <значення>**; наприклад, команда присвоювання **root['bg'] = 'green'** змінить значення властивості вікна *колір фону* та встановить зелений колір вікна з іменем *root*;
- **командою використання метода вікна**, загальний вигляд якої такий: **<ім'я вікна>.<ім'я методу>(<параметри>)**; наприклад, відома вам команда **root.geometry(<рядок параметрів>)** змінить значення властивостей *розміри вікна* та *відступи вікна*, а команда **root.title(<текст>)** змінить значення властивості *заголовок вікна*.

Обробник події найчастіше записують на початку проєкту.

На малюнку 2.22 наведено приклад обробника події **Click** для вікна, під час виконання якого зміняться текст заголовка, розміри і положення вікна й результат виконання.

```
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)
```



Мал. 2.22. Проєкт і результат його виконання після настання події **Click** для вікна

Звертаємо вашу увагу!

У вищенаведеному проєкті підпрограма – обробник події не повертає нічого в команду її виклику, тобто є **процедурою**.

Після запуску проєкту на виконання потрібно дочекатися відкриття вікна та вибрати будь-яку його точку. Тим самим відбудеться подія **Click** для вікна та буде виконана процедура – обробник події, результатом виконання якої буде змінення кольору фону вікна, тексту в рядку заголовка, розмірів і положення вікна.

Іноді потрібно під час опрацювання події встановити нове значення властивості об'єкта, яке обчислюється на основі поточного значення цієї самої властивості. Наприклад, потрібно, щоб після клацання лівої кнопки миші вікно опустилося на 50 пікселів, тобто слід збільшити відстань вікна від верхньої межі екрана на 50 пікселів. У такому разі потрібно отримати поточне значення відстані, змінити його на 50 і запам'ятати результат в окремій змінній, щоб встановити нове положення вікна.

Для отримання поточних значень розмірів і положення вікна використовують такі методи:

- **winfo_width()** – (англ. *window information* – інформація про вікно) поточна ширина вікна;
- **winfo_height()** – поточна висота вікна;
- **winfo_x()** – поточна відстань лівої межі вікна від лівого краю екрана;
- **winfo_y()** – поточна відстань верхньої межі вікна від верхнього краю екрана.

Наприклад, щоб дізнатися поточну відстань вікна з іменем *root* від верхньої межі екрана та збільшити її на 50, потрібно виконати команду:

```
t = root.winfo_y() + 50
```

Щоб установити нове значення, потрібно в рядок параметрів метода **geometry()** замість числа підставити значення змінної. Для цього слід:

- перед рядком параметрів увести символ **f**;
- у рядку параметрів на місці, де потрібно вставити значення змінної, увести символи **{}**, а між ними ім'я змінної:
наприклад, **root.geometry(f'300x200+500+{t}')**

Щоб використати в команді значення кількох змінних, потрібно на всіх відповідних місцях увести символи **{}** з іменами відповідних змінних.

Наприклад, для того щоб після настання події *кляцання лівої кнопки миші* вікно опустилося на 50 пікселів униз, а його ширина збільшилася на 140 пікселів, можна виконати проєкт, наведений на малюнку 2.23.

```
from tkinter import *
def click(event):
    w = root.winfo_width() + 140
    t = root.winfo_y() + 50
    root.geometry(f'{w}x320+200+{t}')
root = Tk()
root.geometry('480x320+200+200')
root.bind('<1>', click)
```

Мал. 2.23. Проєкт для змінення положення та розмірів вікна залежно від попередніх значень положення та розмірів

ВІКНО ПОВІДОМЛЕНЬ І ЙОГО ВИКОРИСТАННЯ




Пригадайте



- У яких ситуаціях під час вашої роботи з різними програмами з'являлися вікна повідомлень?
- Які повідомлення ви бачили в таких вікнах?

Для виведення на екран у ході виконання проєкту текстових повідомлень можна використати спеціальне вікно – **вікно повідомлень**.

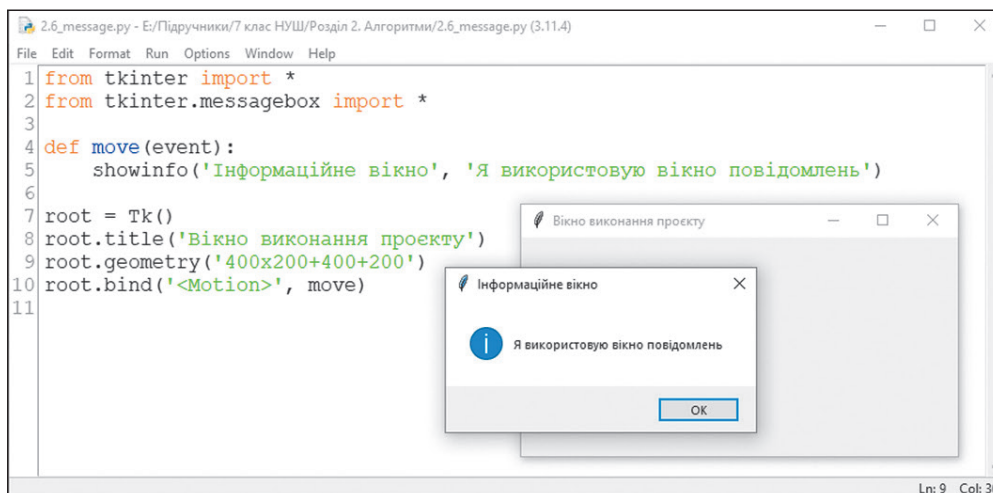
У мові **Python** можна створити спеціальні вікна для виведення в них повідомлень різного призначення: інформаційне повідомлення, повідомлення-попередження, повідомлення про помилку та інші.

Щоб мати можливість відкривати у проєкті такі вікна з повідомленнями, потрібно приєднати до проєкту модуль **tkinter.messagebox** командою **from tkinter.messagebox import ***.

Щоб відкрилося **інформаційне вікно** зі значком , з певним текстом у заголовку вікна і з певним **інформаційним повідомленням**, потрібно використати команду **showinfo(<текст у рядку заголовка>, <текст повідомлення>)** (англ. *show info* – показувати інформацію).

Команди відкриття вікон з повідомленнями інших призначень аналогічні. Наприклад, вікно з повідомленням-попередженням відкривається командою **showwarning** (значок на вікні ) , а вікно з повідомленням про небезпеку – командою **showerror** (значок на вікні ).

Приклад проєкту з обробником події **MouseMove** для вікна проєкту з командою відкриття інформаційного вікна та результат його виконання наведено на малюнку 2.24.



Мал. 2.24.
Використання
вікна
повідомлень

Після ознайомлення з повідомленням потрібно закрити інформаційне вікно, вибравши кнопку **OK** або кнопку закриття вікна. Тільки після цього можна закрити вікно виконання проєкту.

Звертаємо вашу увагу!

В останньому прикладі використано подію **MouseMove**. Цікавим є те, що в цьому випадку, якщо інформаційне вікно відкривається поверху вікна виконання проєкту, проєкт буде складно завершити, використовуючи мишу. Адже як тільки вказівник буде переміщуватися по вікну проєкту, наступатиме подія **MouseMove** і як результат запускатиметься на виконання обробник цієї події, який знову відкриватиме інформаційне вікно. Тому закінчити виконання цього проєкту можна, натиснувши сполучення клавіш **Alt + F4**. Це сполучення клавіш можна використовувати для закриття вікна виконання будь-якого проєкту.



Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви можете знайти за адресою <https://cutt.ly/HwNTeWiN> або QR-кодом.



Найважливіше в цьому пункті

З вікном під час виконання віконного проєкту можуть відбуватися події.

Такими подіями, наприклад, можуть бути:

- **Click** – вибір будь-якої точки вікна;
- **DbClick** – подвійне клацання у будь-якій точці вікна;
- **Close** – закривання вікна;
- **Create** – створення вікна;
- **KeyPress** – натиснення будь-якої клавіші на клавіатурі;
- **MouseMove** – переміщення вказівника в межах вікна виконання проєкту та інші.

З будь-якою подією, яка може відбутися з вікном, можна пов'язати **підпрограму**, яка буде виконуватися одразу після настання цієї події. Така підпрограма називається **обробник події**.

У проєкті можуть бути кілька обробників різних подій.

Підпрограма, яка пов'язана з певним об'єктом, називається **методом цього об'єкта**.

У мові **Python** обробник події для вікна створюють як окрему **процедуру** та пов'язують її з вікном, використовуючи метод **bind**.

У загальному вигляді команда пов'язування обробника події з вікном, що має ім'я *root*, має такий вигляд:

```
root.bind('<подія>', <ім'я_обробника_події>)
```

Обробник події у загальному вигляді записують так:

```
def <ім'я_обробника_події> (event):
```

```
    <команди, які будуть виконуватися за настання події>
```

Значення властивостей вікна можна змінити:

- **командою присвоювання**, загальний вигляд якої такий: **<ім'я вікна>[<властивість>] = <значення>**; наприклад, команда присвоювання **root['bg'] = 'green'** змінить значення властивості вікна *колір фону* та встановить зелений колір вікна з іменем *root*;
- **командою використання метода вікна**, загальний вигляд якої такий: **<ім'я вікна>.<ім'я методу>(<параметри>)**; наприклад, відома вам команда **root.geometry('<рядок параметрів>')** змінить значення властивостей *розміри вікна* та *відступи вікна*, а команда **root.title('<текст>')** змінить значення властивості *заголовок вікна*.

Для отримання поточного значення розмірів і положення вікна використовують функції:

- **winfo_width()** – ширина вікна;
- **winfo_height()** – висота вікна;
- **winfo_x()** – відстань лівої межі вікна від лівого краю екрана;
- **winfo_y()** – відстань верхньої межі вікна від верхнього краю екрана.

Щоб установити нове значення, потрібно в рядок параметрів метода **geometry()** замість числа підставити значення змінної. Для цього слід:

- перед рядком параметрів увести символ `f`;
- у рядку параметрів на місці, де потрібно вставити значення змінної, увести символи `{}`, а між ними ім'я змінної:
наприклад, `root.geometry(f'300x200+500+{t}')`

У мові **Python** можна створити вікна повідомлень кількох видів. Для цього призначені команди:

- `showinfo(<текст у рядку заголовка>, <текст повідомлення>')` – створення інформаційного вікна;
- `showwarning(<текст у рядку заголовка>, <текст повідомлення>')` – створення вікна попередження;
- `showerror(<текст у рядку заголовка>, <текст повідомлення>')` – створення вікна оповіщення про помилку та інші.

Для відкриття вікна повідомлень із заданим текстом заголовка та виведення у ньому заданого тексту повідомлення потрібно приєднати до проекту модуль `tkinter.messagebox` і використати команду відповідно до призначення вікна.

Дайте відповіді на запитання

1. Які події можуть відбуватися з вікном під час виконання проекту?
2. Що називають *обробником події*? Як пов'язати з подією її обробник?
3. Що називають *методом об'єкта*?
4. Як оформити обробник події в **Python**?
5. Якими командами можна змінити поточні значення властивостей об'єкта? Який їх загальний вигляд? Наведіть приклади таких команд.
6. Як відкрити вікно повідомлень? Для чого використовують такі вікна?

Обговоріть і зробіть висновки

1. Чому система налаштована так, що не можна продовжувати працювати з проектом, поки не закрито вікно повідомлення?
2. Навіщо передбачено різні види вікон-повідомлень?



Виконайте завдання

1. Запишіть у зошиті команду, яка:

- а) установить висоту вікна 200 пікселів;
- б) збільшить висоту вікна на 200 пікселів;
- в) зменшить ширину вікна на 100 пікселів;
- г) установить відступ вікна від лівої межі екрана 100 пікселів;
- д) перемістить вікно на 100 пікселів управо;
- е) перемістить вікно на 80 пікселів уліво;
- ж) перемістить вікно на 200 пікселів угору;
- и) перемістить вікно на 100 пікселів униз.

2. Створіть проєкт, у якому в результаті настання події **Click** для вікна встановляться розміри вікна 400 на 300 пікселів, синій колір фону та зміниться текст заголовка на назву вашого населеного пункту. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.6.2.py**.


3. Створіть проєкт, у якому в результаті настання події **KeyPress** для вікна встановляться розміри вікна 300 на 200 пікселів, червоний колір фону, збільшиться відступ від верхньої межі екрана на 200 пікселів і зміниться текст заголовка на ваше прізвище. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.6.3.py**.


4. Створіть проєкт, у якому в результаті настання події **DbIClick** для вікна зміниться колір фону вікна на вибраний вами, текст заголовка на вибраний вами і збільшиться на 200 пікселів відступ верхньої межі вікна від верхньої межі екрана. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.6.4.py**.

5. Створіть проєкт, у якому в результаті настання події **Create** зміняться розміри вікна, колір фону і збільшиться на 300 пікселів відступ його лівої межі від лівої межі екрана. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.6.5.py**.

6. Створіть проєкт, у якому в результаті настання події **Click** для вікна встановляться розміри вікна 500 на 500 пікселів, коричневий колір фону і зміниться текст заголовка на *Проект з двома процедурами*, а в результаті настання події **KeyPress** відкриється вікно повідомлень з текстом *Я – громадянин України!*. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.6.6.py**.

7. Створіть проєкт, у якому в результаті настання події *Закривання вікна проєкту* відкриється вікно повідомлень з текстом *Виконання проєкту закінчено! До побачення!*. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.6.7.py**.

 8. **Працюємо у групах.** Розгляньте 2–3 властивості вікна, які не розглянуті в цьому пункті. Створіть проєкт для змінення значень розглянутих вами властивостей. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.6.8.py**.

 **Практична робота № 4.**
«Проєкти для змінення значень властивостей вікна. Використання вікна повідомлень»

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

1. Створіть проєкт, за виконання якого буде відкриватися вікно з такими значеннями властивостей:
 - а) колір фону – *сірий*;
 - б) ширина – *200 пікселів*;
 - в) висота – *100 пікселів*;
 - г) відступ лівої межі – *150 пікселів*;
 - д) відступ верхньої межі – *100 пікселів*;
 - е) текст у рядку заголовка – *Практична робота*.

2. Збережіть проєкт у вашій папці у файлі з іменем **практична 4.py**.
3. Додайте до проєкту обробник події **KeyPress** для вікна, виконання якого змінить колір фону вікна на синій, збільшить на 300 пікселів відступ верхньої межі вікна від верхньої межі екрана, зменшить його ширину на 200 пікселів.
4. Збережіть проєкт з тим самим іменем у тому самому місці.
5. Додайте до проєкту обробник події **Click** для вікна, виконання якого встановить червоний колір фону вікна, установить відступ верхньої межі вікна 200 пікселів від верхньої межі екрана, збільшить ширину вікна на 300 пікселів, зменшить на 50 пікселів відступ лівої межі вікна від лівої межі екрана, відкриє вікно повідомлень з текстом *Ми вивчаємо програмування!*.
6. Збережіть проєкт з тим самим іменем у тому самому місці.



Готуємось до вивчення нового матеріалу

1. Чим процедура відрізняється від функції?
2. Чи може процедура містити аргументи?



2.7. Створення графічних зображень на вікні

У цьому пункті йтиметься про:

- ▶ систему координат на вікні;
- ▶ методи мови програмування для створення графічних зображень;
- ▶ створення проєктів з графічними зображеннями.

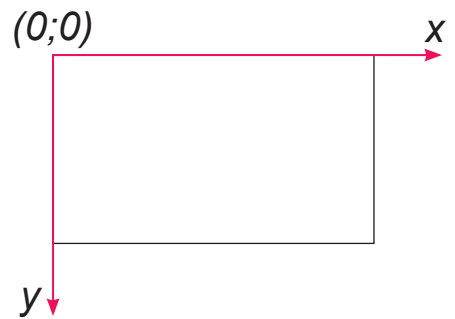
СИСТЕМА КООРДИНАТ НА ВІКНІ



Пригадайте

- Що таке *система координат на площині*?
- Як визначаються координати точки на координатній площині?
- Які події можуть відбуватися з вікном? Що таке *обробник події*?

Для створення зображень у середовищі **IDLE Python** на вікні задано умовну систему координат, початок якої розміщується в лівому верхньому куті вікна, вісь абсцис (x) напрямлено вправо, вісь ординат (y) напрямлено вниз (мал. 2.25). **Координати кожної точки вікна – цілі невід’ємні числа.**



Мал. 2.25. Система координат на вікні

СТВОРЕННЯ ГРАФІЧНИХ ЗОБРАЖЕНЬ

Зображення створюються на спеціальному об’єкті **Canvas** (англ. *canvas* – полотно, живопис). Щоб розташувати зображення на вікні, потрібно створити об’єкт **Canvas** і розмістити його на вікні, наприклад, так:

```
c = Canvas()  
c.pack()
```

Об’єкт **Canvas** за замовчуванням має розміри 378 на 265 точок, але йому можна задати потрібну ширину та довжину, а також колір фону під час створення:

```
c = Canvas(width=300, height=300, bg="white")  
c.pack()
```

Систему координат на об’єкті **Canvas** задано так само, як на вікні.

Для малювання на об’єкті **Canvas** можна використовувати методи, наведені в таблиці 2.3.

Таблиця 2.3

Методи малювання на об’єкті Canvas

<i>Метод</i>	<i>Призначення</i>	<i>Приклад команди</i>
delete("all")	Очищує весь уміст об’єкта Canvas	c.delete("all")

<i>Метод</i>	<i>Призначення</i>	<i>Приклад команди</i>
create_line (x1, y1, x2, y2)	Малює відрізок від точки з координатами (x1, y1) до точки з координатами (x2, y2)	c.create_line (50, 60, 150, 200)
create_rectangle (x1, y1, x2, y2)	Малює прямокутник, ліва верхня вершина якого має координати (x1, y1), а права нижня – (x2, y2)	c.create_rectangle (100, 100, 350, 400)
create_oval (x1, y1, x2, y2)	Малює еліпс, вписаний у прямокутник, ліва верхня вершина якого має координати (x1, y1), а права нижня – (x2, y2)	c.create_oval (100, 100, 350, 400)
create_polygon (x1, y1, x2, y2, ..., xn, yn)	Малює багатокутник, координатами вершин якого є точки з указаними координатами; остання точка буде з'єднана з першою	c.create_polygon (0, 30, 200, 200, 200, 30)

Звертаємо вашу увагу!

1. Щоб намалювати квадрат, потрібно використати метод **create_rectangle** і вказати координати вершин прямокутника такі, щоб $x_2 - x_1 = y_2 - y_1$. Тоді сусідні сторони прямокутника будуть рівні. Наприклад, **c.create_rectangle(100, 100, 300, 300)**.

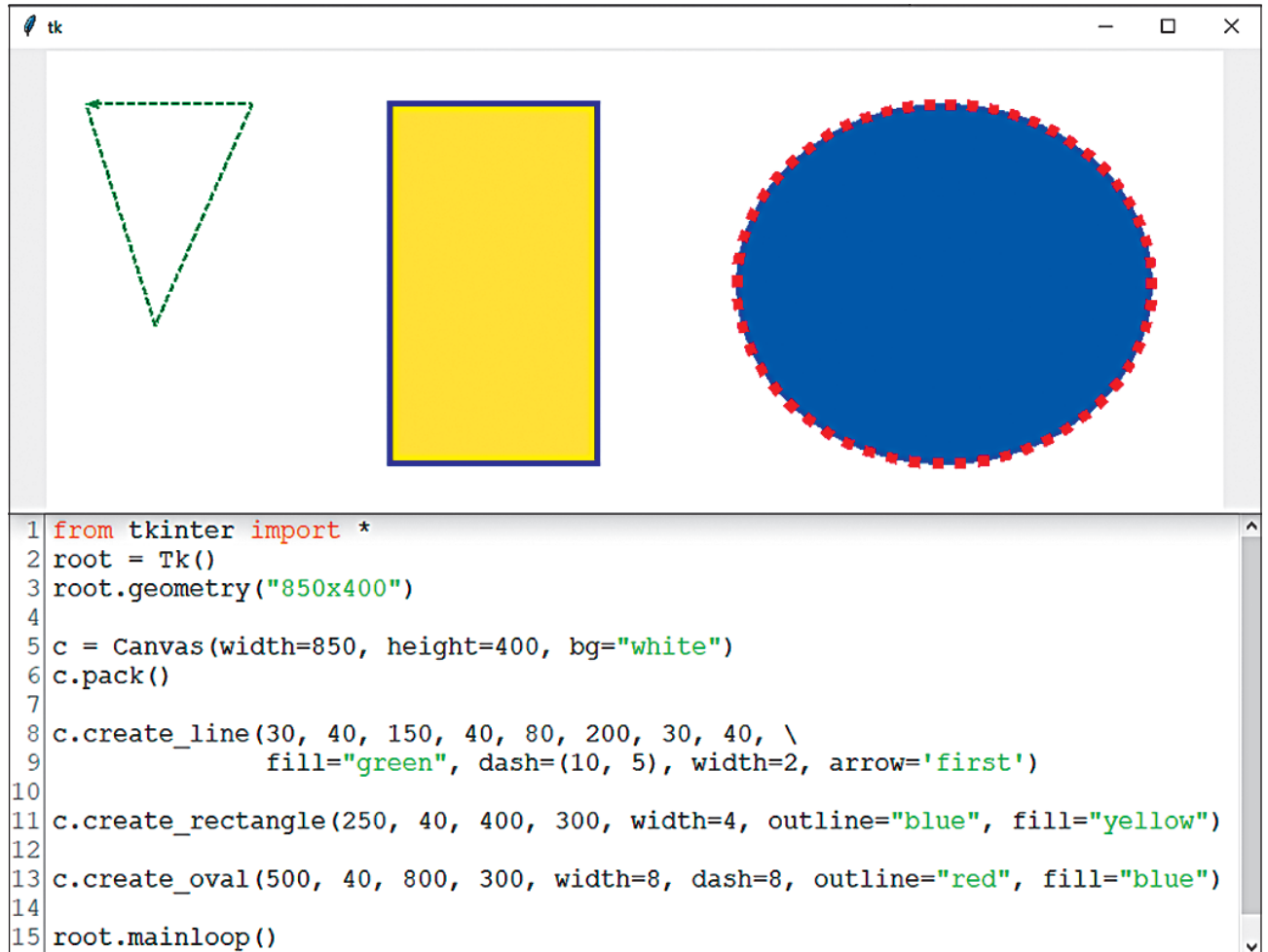
2. Щоб намалювати коло, потрібно використати метод **create_oval** і вказати такі координати вершин прямокутника, у який буде вписано еліпс, щоб прямокутник був квадратом. Наприклад, **c.create_oval(200, 200, 600, 600)**.

3. Щоб намалювати ламану, потрібно використати метод **create_line()** і ввести послідовно координати всіх її вершин. Для того щоб ламана була замкненою, потрібно вказати координати останньої вершини такі самі, як координати першої вершини. Наприклад, **c.create_line(55, 85, 155, 85, 105, 180, 55, 85)**.

Контури фігур на малюнках можуть бути різного кольору, товщини, стилю, а внутрішню область можна зафарбовувати різними кольорами. Для цього в командах малювання потрібно вказати значення таких властивостей фігур:

- **fill** (англ. *fill* – заповнити) – колір зафарбовування фігур; значення – назва кольору англійською. Наприклад, **c.create_oval(200, 200, 400, 600, fill="yellow")**;
- **outline** (англ. *outline* – контур) – колір контуру фігур; значення – назва кольору англійською. Наприклад, **c.create_rectangle(50, 100, 300, 200, outline="red")**;
- **width** – ширина (товщина) лінії малювання; значення – натуральне число. Наприклад, **c.create_line(55, 85, 155, 85, width=3)**;
- **dash** (англ. *dash* – тире) – штрихова лінія малювання; значенням є набір цілих чисел, який задає тип штриховки, тобто ширину штриха та ширину проміжку між штрихами. Якщо значенням є одне число, то ширина штриха та ширина проміжку будуть однаковими. Наприклад, **c.create_line(55, 85, 155, 85, dash=(7, 2))**;
- **arrow** (англ. *arrow* – стрілка) – наявність стрілки на кінці лінії; значення **first** (англ. *first* – перший) означає, що стрілка буде зображена біля першої точки лінії, **last** (англ. *last* – останній) – стрілка буде зображена біля останньої точки лінії, **both** (англ. *both* – обидва) – стрілки будуть з обох кінців лінії. Наприклад, **c.create_line(55, 85, 155, 85, arrow='both')**;

На малюнку 2.26 наведено приклад проєкту, у якому використовуються різні інструменти з різними значеннями їх властивостей для малювання геометричних фігур, і результат його виконання.



Мал. 2.26. Проєкт для малювання геометричних фігур і результат його виконання

Звертаємо вашу увагу!

Під час виконання проєктів для малювання графічних фігур у середовищі **IDLE Python** не завжди правильно відтворюються штриховані лінії, шаблон яких задає властивість **dash**.

ВИКОРИСТАННЯ ПРОЦЕДУР ДЛЯ МАЛЮВАННЯ ЗОБРАЖЕНЬ



Пригадайте

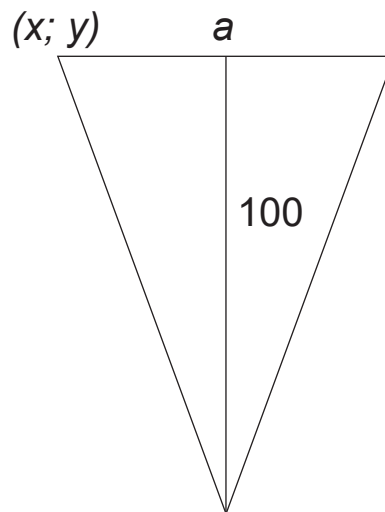
• Що таке *процедура*? • Що таке *аргументи процедури*, *результати процедури*? • Як викликається процедура на виконання? • Чим процедура з результатами відрізняється від процедури без результатів?

Якщо побудову деякої геометричної фігури оформити як процедуру, то одну цю процедуру можна використати для побудови цієї фігури різних розмірів і в різних місцях вікна.

Розглянемо як приклад проект з використанням процедури малювання рівнобедреного трикутника з висотою 100. У цієї процедури будуть три аргументи цілого типу: дві координати лівої вершини основи трикутника й довжина основи (мал. 2.27). Це дасть нам змогу використати цю процедуру для малювання таких рівнобедрених трикутників у різних місцях вікна та з різними довжинами основи.

Конкретні значення координат лівої вершини основи трикутника $(x; y)$ передаватимуться процедурі в команді її виклику. Координати правої вершини основи дорівнюватимуть $(x + a; y)$.

Щоб намальований трикутник був рівнобедреним, абсциса вершини рівнобедреного трикутника повинна дорівнювати $x + a : 2$, а ордината – $y + 100$. Оскільки координати всіх точок вікна (полотна) повинні бути цілими числами, то для визначення половини довжини



Мал. 2.27.
Рівнобедрений
трикутник
з координатами
лівої вершини основи
 (x, y) , довжиною
основи a і висотою 100

основи потрібно використати не операцію звичайного ділення ($/$), а операцію цілочисельного ділення, яку ви знаєте ще з молодшої школи як операцію знаходження частки від ділення цілих чисел з остачею.

У мові програмування **Python** операцію знаходження частки від ділення цілих чисел з остачею позначають $//$. Наприклад, $8 // 3 = 2$, $17 // 5 = 3$.

На малюнку 2.28 наведено текст проєкту малювання трикутників з використанням процедури.

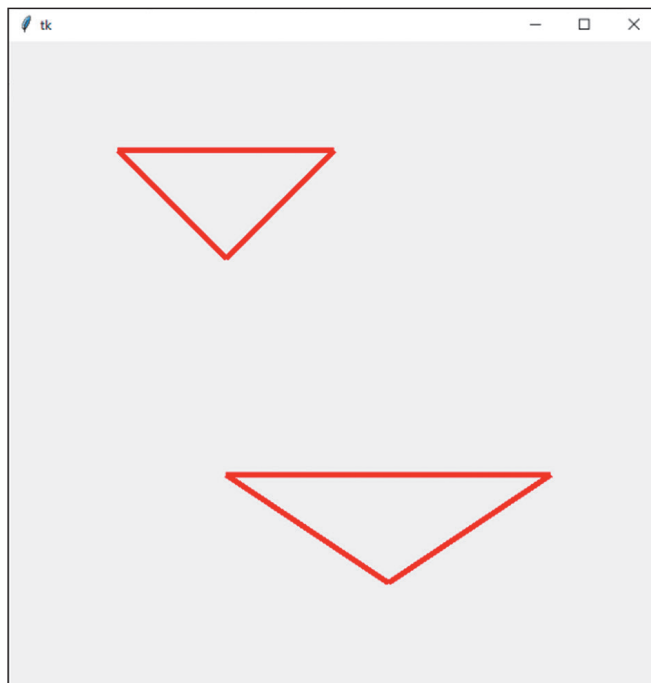
```
File Edit Format Run Options Window Help
1 from tkinter import *
2 def triangle(x, y, a):
3     c.create_line(x,y, x+a,y, width=5, fill="red")
4     c.create_line(x+a,y, x+a//2,y+100, width=5, fill="red")
5     c.create_line(x+a//2,y+100, x,y, width=5, fill="red")
6
7 root = Tk()
8 root.geometry("600x600")
9
10 c = Canvas(width=600, height=600)
11 c.pack()
12
13 triangle(100, 100, 200)
14 triangle(200, 400, 300)
15
16 root.mainloop()
17
```

Мал. 2.28. Проєкт з процедурою для малювання рівнобедреного трикутника

У наведеному тексті проєкту:

- у процедурі **triangle** для малювання трикутника:
 - команда **c.create_line(x,y, x+a,y, width=5, fill="red")** малює основу трикутника завдовжки **a** з точки з координатами **(x, y)** до точки з координатами **(x+a, y)**;
 - команди **c.create_line(x+a,y, x+a//2,y+100, width=5, fill="red")** і **c.create_line(x+a//2,y+100, x,y, width=5, fill="red")** малюють бічні сторони трикутника;

- в основній частині тексту проєкту:
 - дві команди виклику процедури **triangle**:
triangle(100, 100, 200) і **triangle(200, 400, 300)**;
 - у кожній з них указано конкретні значення двох координат лівої вершини основи трикутника й довжина цієї основи;
 - у процесі виконання першої з цих команд змінні **x**, **y**, **a** – аргументи процедури **triangle** – набувають указаних у команді значень (**x = 100**, **y = 100**, **a = 200**) і відбувається виконання команд процедури **triangle** із цими значеннями аргументів; після закінчення виконання процедури **triangle** на вікні з'являється рівнобедрений трикутник із заданими координатами лівої вершини основи та довжиною основи (мал. 2.29);
 - друга з цих команд виклику виконується аналогічно (мал. 2.29).



Мал. 2.29. Результат виконання проєкту з процедурою для малювання рівнобедреного трикутника



Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви можете знайти за адресою <https://cutt.ly/uwNTriEz> або QR-кодом.



Найважливіше в цьому пункті

Для створення зображень у середовищі **IDLE Python** на вікні задано умовну систему координат, початок якої розміщується в лівому верхньому куті вікна, вісь абсцис (x) напрямлено вправо, вісь ординат (y) – вниз. **Координати кожної точки вікна – цілі невід'ємні числа.**

Зображення створюються на спеціальному об'єкті **Canvas** (англ. *canvas* – полотно, живопис). Щоб розташувати зображення на вікні, потрібно створити об'єкт **Canvas** і розмістити його на вікні, наприклад, так:

```
c = Canvas()  
c.pack()
```

Об'єкт **Canvas** за замовчуванням має розміри 378 на 265 точок, але йому можна задати потрібну ширину та довжину, а також колір фону під час створення:

```
c = Canvas(width=300, height=300, bg="white")  
c.pack()
```

Для малювання можна використовувати такі методи: **delete("all")**, **create_line(x1, y1, x2, y2)**, **create_rectangle(x1, y1, x2, y2)**, **create_oval(x1, y1, x2, y2)**, **create_polygon(x1, y1, x2, y2, ..., xn, yn)**.

Контури фігур на малюнках можуть бути різного кольору, товщини, стилю, а внутрішню область можна зафарбовувати різними кольорами. Для цього в командах малювання потрібно вказати значення таких властивостей фігур: **fill**, **outline**, **width**, **dash**, **arrow**.



Дайте відповіді на запитання

1. Яку систему координат уведено на вікні?
2. Які ви знаєте властивості інструментів малювання? Яких значень вони можуть набувати?

3. Які ви знаєте методи малювання і для чого призначено кожний з них?

4. Для чого можна використати процедури під час малювання фігур?



Обговоріть і зробіть висновки

1. Чому, на вашу думку, на вікні вводиться саме така система координат, а не застосовується та, яку ви використовували на уроках математики?

2. Чи можна для малювання фігур використовувати не процедури, а функції?



Виконайте завдання

1. Створіть проєкт для малювання *Качечки* за точками з координатами: (230; 200), (210; 180), (190; 180), (230; 150), (210; 120), (170; 130), (150; 120), (170; 160), (140; 170), (170; 170), (150; 180), (150; 220), (180; 230), (160; 240), (210; 240), (230; 170), (260; 210), (230; 200), (190; 250). Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.7.1.py**.


2. Створіть проєкт з використанням процедури для малювання рівнобедреного трикутника із заданими координатами правої вершини основи, довжини основи та висоти. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.7.2.py**.

3. Створіть проєкт з використанням процедури для малювання трьох прямокутників різних розмірів лініями різної товщини та кольору. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.7.3.py**.

4. Створіть проєкт з використанням процедури для малювання трьох кругів різних розмірів лініями різної товщини. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.7.4.py**.

5. Намалюйте в зошиті якийсь малюнок: сніговика, світлофор, будинок, ялинку, пірамідку або малюнок на власний розсуд. Створіть проєкт для малювання цього малюнка на вікні. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.7.5.py**.

6. Створіть проєкт, у якому після відкриття вікна на ньому вже є намальований квадрат, а після настання події **Click** з вікном на вікні з'являється ще й зафарбований круг. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.7.6.py**.

 7. Створіть проєкт для малювання олімпійського прапора. Збережіть проєкт у вашій папці у файлі з іменем **завдання 2.7.7.py**.



Оцініть свої знання та вміння з розділу «Алгоритми та програми»

Оцініть свої навчальні досягнення із цього розділу (початковий, середній, достатній, високий рівень).

- Я знаю, що таке мова програмування, для чого використовують програми-транслятори і чим відрізняється робота інтерпретатора від роботи компілятора.
- Я вмію використовувати середовище для створення, редагування та запуску проєктів на виконання з використанням вікна **введення-виведення**.
- Я знаю, у яких випадках в алгоритмах потрібно використовувати розгалуження, які є види розгалужень.
- Я вмію використовувати різні види розгалужень для створення алгоритмів.
- Я вмію створювати проєкти з використанням розгалужень.
- Я вмію використовувати проєкти для проведення комп'ютерних експериментів і перевірки гіпотез.
- Я знаю, навіщо у проєктах використовують підпрограми, які є види підпрограм, що таке аргументи й результати підпрограми, як викликається підпрограма на виконання.

- Я вмію створювати проєкти з використанням підпрограм.
- Я вмію використовувати середовище створення віконних проєктів для створення, редагування та запуску проєкту на виконання.
- Я вмію змінювати значення властивостей вікна.
- Я знаю, що таке подія та обробник події.
- Я знаю, які події можуть відбуватися з вікном.
- Я вмію створювати проєкти з використанням обробників різних подій з вікном.
- Я вмію створювати проєкти для створення зображень на вікні.

Повторіть той матеріал, який ви знаєте недостатньо.



Готуємось до вивчення нового матеріалу

1. Які комп'ютерні презентації ви створювали і для чого?
2. Яка роль анімації об'єктів на слайді та анімації слайдів у комп'ютерній презентації?

Розділ 3

КОМП'ЮТЕРНІ ПРЕЗЕНТАЦІЇ

У цьому розділі ви отримаєте нові, а також поглибите та розширите свої знання й удосконалисте навички з таких тем:

- ▶ комп'ютерні презентації з розгалуженнями;
- ▶ створення гіперпосилань у комп'ютерних презентаціях;
- ▶ використання тригерів у комп'ютерних презентаціях;
- ▶ налаштування демонстрації та керування показом комп'ютерних презентацій.

3.1. Створення комп'ютерних презентацій з розгалуженнями




У цьому пункті йтиметься про:




- ▶ комп'ютерні презентації з розгалуженнями;
- ▶ створення зовнішніх і внутрішніх гіперпосилань у презентаціях;
- ▶ використання кнопок дій;
- ▶ налаштування користувацького показу слайдів.

КОМП'ЮТЕРНІ ПРЕЗЕНТАЦІЇ З РОЗГАЛУЖЕННЯМИ



Пригадайте

- Для чого призначені елементи керування  ,  та  , що з'являються під час демонстрації комп'ютерної презентації в режимі **Показ слайдів**?
- Яке призначення гіперпосилань на вебсторінках?
- Що таке *розгалуження* в алгоритмах?

Ви вже знаєте, що під час демонстрації комп'ютерної презентації слайди відображаються послідовно в порядку їх розміщення. Змінювати порядок показу слайдів можна, використовуючи елементи керування  ,  та  , які з'явля-

ються під час демонстрації. Ще більше можливостей з керування послідовністю показу слайдів комп'ютерної презентації можна отримати, використовуючи **гіперпосилання**.

Гіперпосилання в комп'ютерних презентаціях також можна використовувати для відкриття інших файлів або сайтів в Інтернеті.

Наприклад, під час виступу з презентацією про визначних мандрівників минулого буде корисним переглянути відомості у **Вікіпедії** про книгу Марко Поло «Книга чудес світу», інтерактивну карту подорожей Христофора Колумба, відео на **YouTube** про значення відкриття Фернана Магеллана. Структуру такої презентації наведено на малюнку 3.1.



Мал. 3.1. Структура комп'ютерної презентації
Відкриття нових земель

Завдяки використанню на слайдах комп'ютерної презентації гіперпосилань у ній утворюються **розгалуження**.

Переглянути презентацію з розгалуженнями *Відкриття нових земель* можна за посиланням <https://cutt.ly/Xw3NDrWH> або за QR-кодом.



СТВОРЕННЯ ГІПЕРПОСИЛАНЬ НА СЛАЙДАХ КОМП'ЮТЕРНОЇ ПРЕЗЕНТАЦІЇ



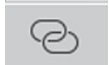
Пригадайте

- З якими об'єктами можуть бути пов'язані гіперпосилання на вебсторінках?
- Як відображаються гіперпосилання на вебсторінках?

Гіперпосилання на слайдах комп'ютерної презентації, як і на вебсторінках, можуть бути пов'язані з фрагментами тексту або з графічними об'єктами – фігурами, піктограмами, зображеннями та іншими.

Внутрішні гіперпосилання призначені для переходів до певних слайдів цієї самої презентації. **Зовнішні** гіперпосилання призначені для переходу на деяку вебсторінку в Інтернеті або відкриття деякого файлу, що зберігається на носіях даних.

У редакторі презентацій **PowerPoint**, щоб створити гіперпосилання, слід:

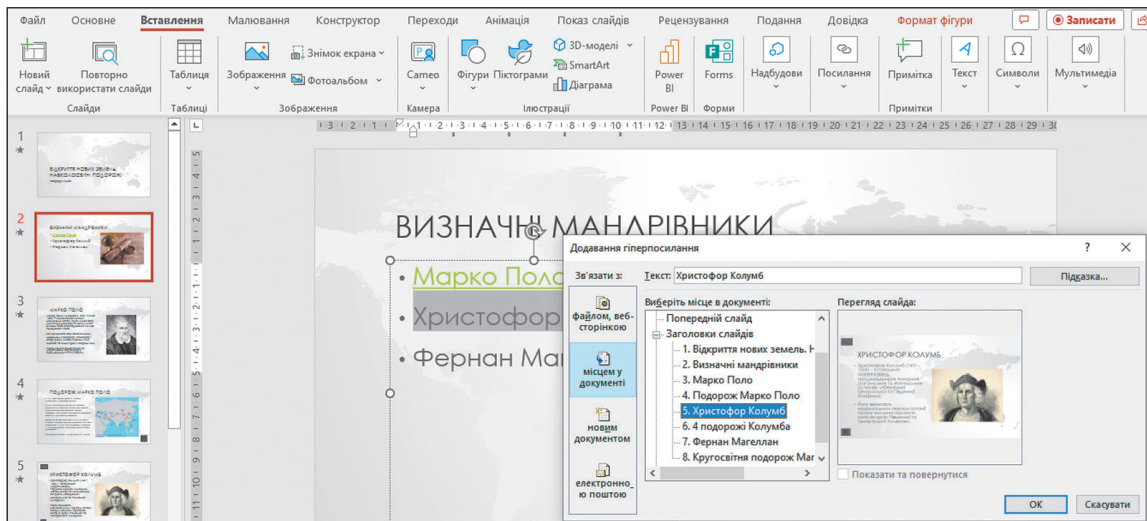
1. Виділити на слайді фрагмент тексту або графічний об'єкт, з яким буде пов'язано гіперпосилання.
2. Виконати **Вставлення** ⇒ **Посилання** ⇒ **Посилання** .
3. Вибрати у вікні **Додавання гіперпосилання** (мал. 3.2) в області **Зв'язати з** кнопку, яка відповідає об'єкту для переходу:

- **місцем у документі**

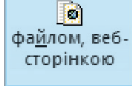


силання;

– для внутрішнього гіперпосилання;




Мал. 3.2. Вікно **Додавання гіперпосилання** під час створення внутрішнього гіперпосилання

- **файлом, вебсторінкою**  – для зовнішнього гіперпосилання.

4. Указати об'єкт для переходу:


- для внутрішнього гіперпосилання – вибрати у списку **Виберіть місце в документі** заголовков слайда, на який потрібно перейти;
- для зовнішнього гіперпосилання – вибрати потрібний файл на носії даних або ввести адресу потрібної вебсторінки в поле **Адреса**.

5. Вибрати кнопку **ОК**.

Фрагмент тексту на слайді, до якого додано гіперпосилання, виділяється кольором і підкреслюється. Під час демонстрації комп'ютерної презентації вказівник після наведення на текстовий або графічний об'єкт, до якого додано гіперпосилання, змінює свій вигляд на такий .

Після вибору внутрішнього гіперпосилання відкриється указаний слайд презентації. Після вибору зовнішнього гіпер-

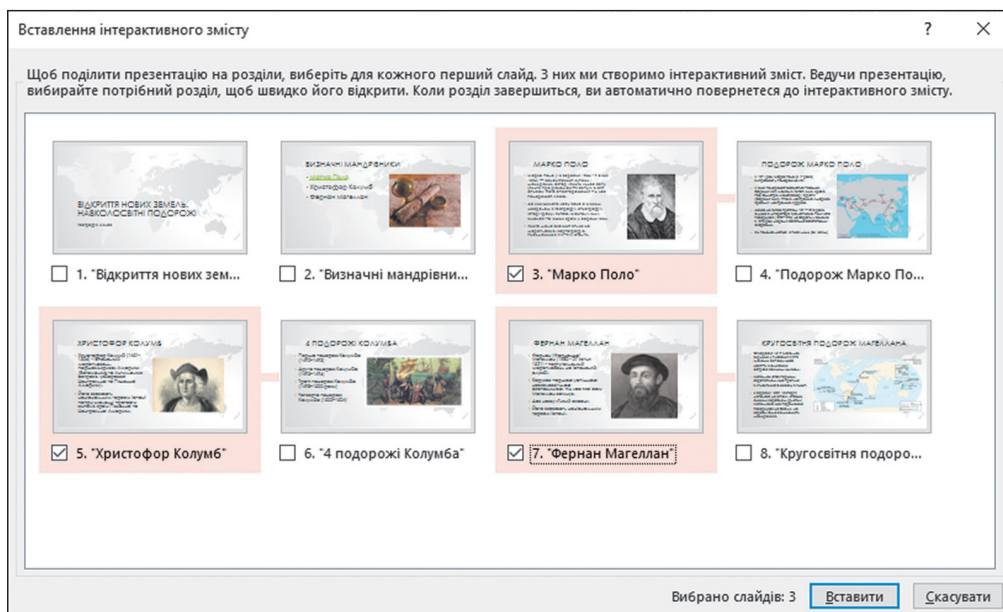
посилання відкриється відповідна програма, а в ній відобразиться вміст потрібного файлу або відкриється вебсторінка з указаною адресою. Після перегляду файлу або вебсторінки та їх закриття буде продовжена демонстрація презентації з того самого слайда, на якому розміщено гіперпосилання.

Для видалення гіперпосилання можна в контекстному меню відповідного текстового фрагмента або графічного об'єкта вибрати команду **Вилучити посилання**  Вилучити посилання .

Для тих, хто хоче знати більше

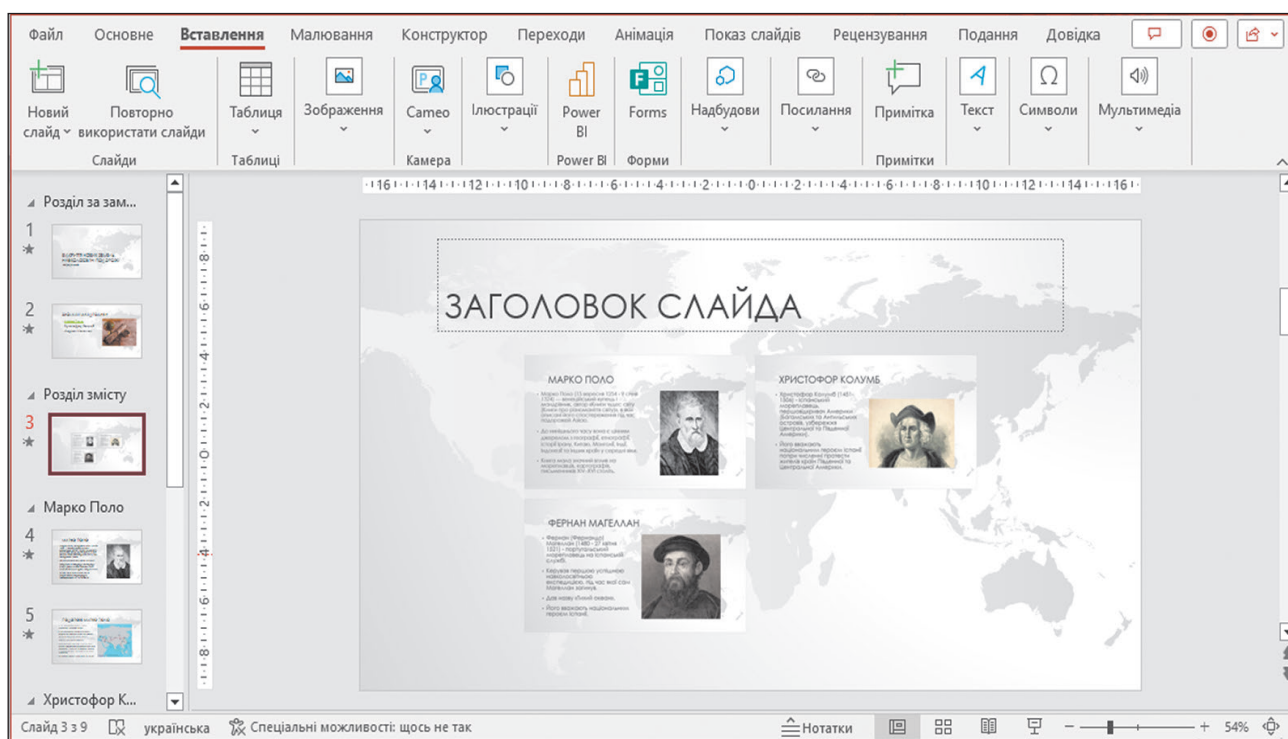
У презентаціях з розгалуженнями часто створюють слайд, який виконує роль меню з гіперпосиланнями для переходів до інших слайдів. Створити такий слайд можна з використанням механізму **швидких переходів**. Щоб створити слайд з меню переходів, слід:

1. Виконати **Вставлення** ⇒ **Посилання** ⇒ **Швидкі переходи** ⇒ **Інтерактивний зміст**.
2. Установити у вікні **Вставлення інтерактивного змісту** позначки прапорців поруч з ескізами слайдів, гіперпосилання на які потрібно створити (мал. 3.3).
3. Вибрати кнопку **Вставити**.



Мал. 3.3. Вікно **Вставлення інтерактивного змісту**

Після цього презентація поділяється на розділи, які відображаються в області **Слайди**. Перший розділ має назву **Розділ за замовчуванням**. Він включає титульний слайд і всі наступні до першого позначеного. Другий розділ – **Розділ змісту** – створюється перед першим з позначених слайдів. У ньому автоматично створюється новий слайд з ескізами вибраних слайдів (мал. 3.4). Назви наступних розділів утворено із заголовків позначених слайдів. Кожен розділ містить слайди, що розташовані між двома послідовними позначеними слайдами. Останній розділ містить слайди від останнього позначеного до кінця презентації.



Мал. 3.4. Створення слайда з меню переходів

Кожен ескіз на слайді з меню переходів є гіперпосиланням на відповідний слайд. Під час демонстрації комп'ютерної презентації після вибору деякого ескізу відбувається перехід на відповідний слайд, а після завершення перегляду всіх слайдів розділу – повернення на слайд меню.

ВИКОРИСТАННЯ КНОПОК ДІЙ НА СЛАЙДАХ КОМП'ЮТЕРНОЇ ПРЕЗЕНТАЦІЇ




Пригадайте

- Як уставити графічні примітиви на слайд комп'ютерної презентації?
- Які ви знаєте властивості графічних примітивів?





Ще один спосіб створення гіперпосилань у презентаціях – використання **кнопок дій**. Кнопки дій є графічними примітивами.

Вам уже доводилося вставляти графічні примітиви на слайди комп'ютерних презентацій, використовуючи список фігур

кнопки **Фігури** . Ви знаєте, що всі графічні примітиви поділені на кілька груп: **Лінії**, **Прямокутники**, **Основні фігури**, **Фігурні стрілки** та інші. У редакторі презентацій **PowerPoint** остання група фігур має назву **Кнопки дій** (мал. 3.5).



Мал. 3.5. Графічні примітиви з групи **Кнопки дій**



Значки на цих графічних примітивах схожі на зображення знайомих вам елементів керування, що використовуються в різних програмах: кнопок **Вперед**  і **Назад** , **На початок**  і **В кінець**  та інші.

Особливістю цих графічних примітивів є пов'язування їх з певними діями з керування переходами між слайдами під час демонстрації комп'ютерної презентації. Використовують

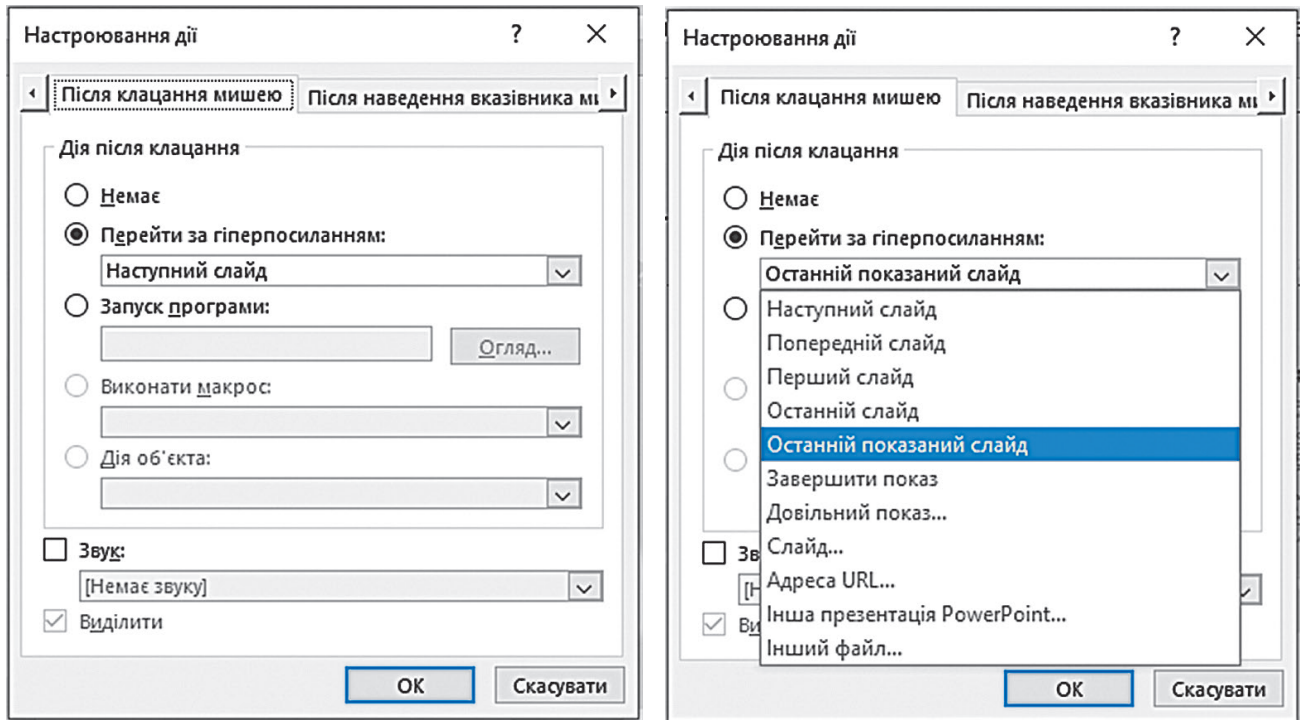
ці графічні примітиви як кнопки, після вибору яких відбувається пов'язана дія. Дії за замовчуванням для деяких кнопок дій наведено в таблиці 3.1.

Таблиця 3.1

Призначення деяких кнопок дій

Зображення кнопок	Назва кнопки дії	Призначення за замовчуванням
	повернутися назад / перейти далі	Для переходу на попередній / наступний слайд
	перейти на початок / перейти в кінець	Для переходу на перший / останній слайд презентації
	на домашню сторінку	Для переходу на слайд, визначений користувачем як домашній у комп'ютерній презентації
	повернення	Для повернення до останнього показаного слайда
	відео	Для запуску відеофайлу
	звук	Для запуску звукового файлу
	пуста	Для вибору будь-якої дії

Після вставлення певної кнопки дії на слайд відкривається вікно **Настроювання дії** (мал. 3.6). Для кнопок, пов'язаних з діями переходів, буде встановлено позначку перемикача **Перейти за гіперпосиланням**, а у списку нижче буде відобразитися дія за замовчуванням для цієї кнопки.



Мал. 3.6. Вікно **Настроювання дії** та список можливих дій

Дію за замовчуванням можна змінити, вибравши іншу дію у списку. Також на вкладці **Після наведення вказівника миші** можна вибрати дію, яка відбудеться під час демонстрації презентації не після клацання, а після наведення вказівника на кнопку дії.

Для відмови від установленної дії можна в контекстному меню кнопки дії вибрати команду **Вилучити посилання**.

Дії, указані у списку вікна **Настроювання дії**, можуть бути пов'язані не лише з кнопками дій, а й з іншими об'єктами слайдів – фрагментами тексту та графічними об'єктами різних типів. Для цього слід виділити потрібний об'єкт слайда,

виконати **Вставка** ⇒ **Посилання** ⇒ **Дія**



і виконати відповідні налаштування у вікні **Настроювання дії**.

КОРИСТУВАЦЬКИЙ ПОКАЗ СЛАЙДІВ

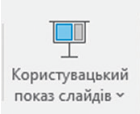


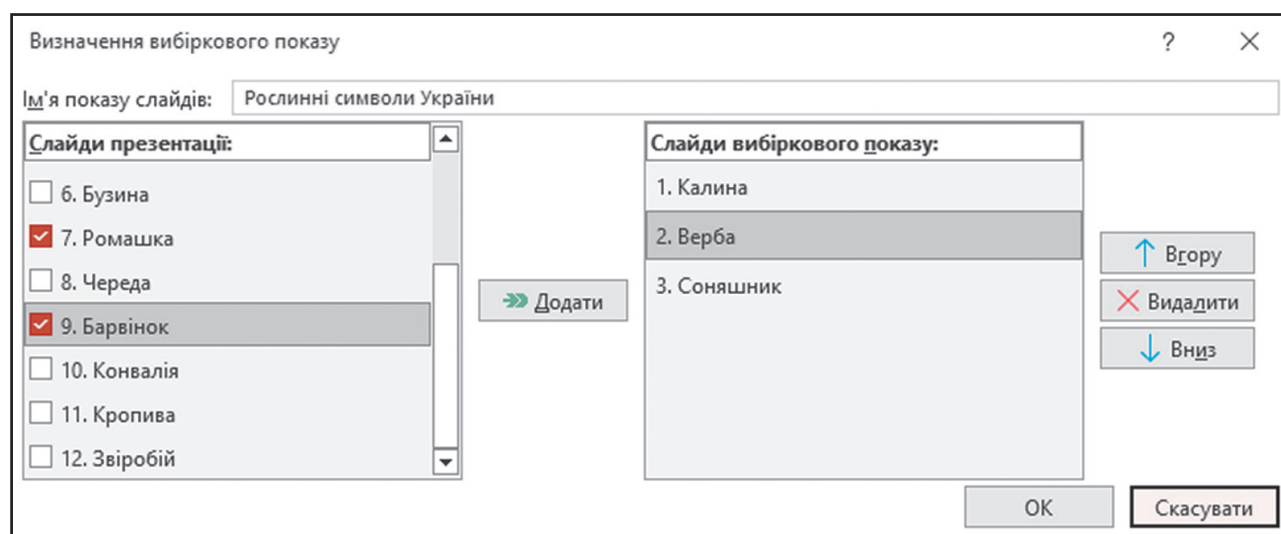
Пригадайте

- Які можна зробити налаштування показу комп'ютерної презентації?
- У яких режимах може демонструватися комп'ютерна презентація?

Одну й ту саму комп'ютерну презентацію можна використувати неодноразово, демонструючи різні набори слайдів у різних ситуаціях. Наприклад, один набір слайдів з презентації *Рослинний світ України* можна продемонструвати на уроці біології з теми *Лікарські рослини*, другий набір – на класній годині *Червона книга України*, третій – на святі *Рослинні символи України*.

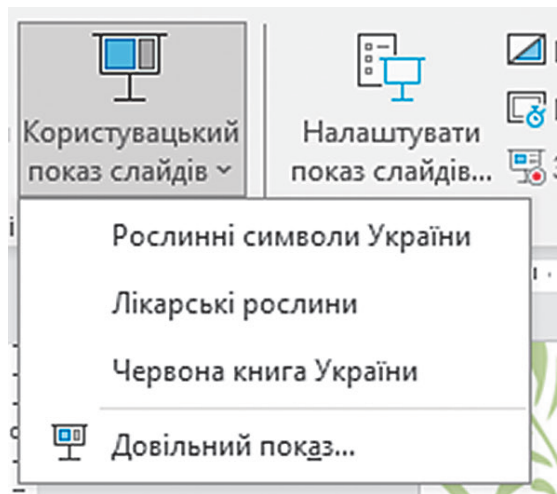
Щоб вибрати набори слайдів, які будуть продемонстровані в певних ситуаціях, використовують так званий **користувацький показ слайдів**.

Для створення різних варіантів показу використовують кнопку **Користувацький показ слайдів**  з групи **Початок показу слайдів** на вкладці **Показ слайдів**. Після її вибору



Мал. 3.7. Вікно **Визначення вибіркового показу**

у вікні **Визначення вибіркового показу** (мал. 3.7) можна ввести ім'я користувачького показу, вибрати слайди, які будуть демонструватися, та встановити потрібну послідовність їх показу.



Мал. 3.8. Список кнопки **Користувачький показ слайдів**

Після створення варіантів показу їх імена відображаються у списку кнопки **Користувачький показ слайдів** (мал. 3.8). Для початку демонстрації потрібного набору слайдів слід вибрати його ім'я в зазначеному списку.

Будь-який з підготовлених наборів слайдів можна встановити як показ презентації за замовчуванням. Саме він буде демонструватися, якщо натиснути клавішу **F5** або виконати **Показ**

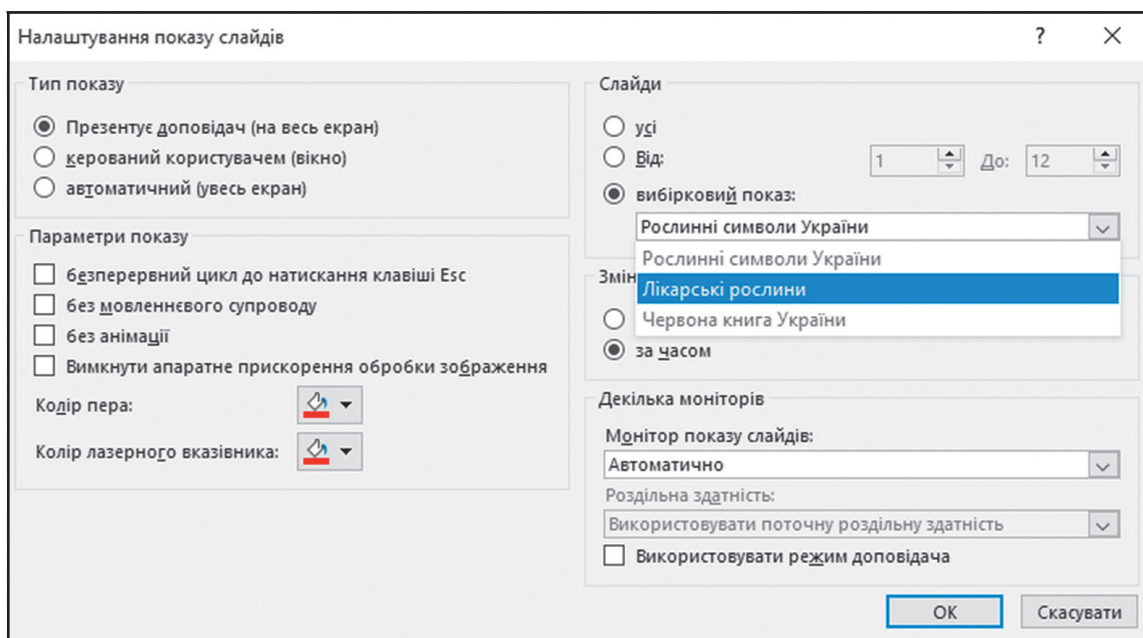
слайдів ⇒ **Початок показу слайдів** ⇒ **З початку**



Для цього слід:

1. Виконати **Показ слайдів** ⇒ **Налаштування** ⇒ **Налаштувати показ слайдів**.
2. Установити позначку перемикача **вибірковий показ** у групі **Слайди** у вікні **Налаштування показу слайдів** (мал. 3.9).
3. Вибрати у списку ім'я потрібного вибіркового показу.
4. Вибрати кнопку **ОК**.

Якщо після виконання цього алгоритму розпочати демонстрацію презентації, то будуть відтворені слайди лише з вибраного набору.



Мал. 3.9. Установлення визначеного набору слайдів для демонстрації комп'ютерної презентації за замовчуванням



Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви можете знайти за адресою <https://cutt.ly/4wNTrQgx> або QR-кодом.



Найважливіше в цьому пункті

Гіперпосилання в комп'ютерних презентаціях можна використовувати для керування послідовністю показу слайдів і для відкриття інших файлів або сайтів у Інтернеті. Завдяки використанню на слайдах комп'ютерної презентації гіперпосилань у ній утворюються **розгалуження**.

Гіперпосилання на слайдах комп'ютерної презентації, як і на вебсторінках, можуть бути пов'язані з фрагментами тексту або з графічними об'єктами – фігурами, піктограмами, зображеннями та іншими.

Внутрішні гіперпосилання призначені для переходів до певних слайдів цієї самої презентації. **Зовнішні** гіперпосилання призначені для переходу на деяку вебсторінку в Інтернеті або відкриття деякого файлу, що зберігається на носіях даних.

Графічні примітиви з групи **Кнопки дій** пов'язані з певними діями з керування переходами між слайдами під час демонстрації комп'ютерної презентації. Використовують ці графічні примітиви як кнопки, після вибору яких відбувається пов'язана дія.

Одну й ту саму комп'ютерну презентацію можна використовувати неодноразово, демонструючи різні набори слайдів у різних ситуаціях. Щоб вибрати набори слайдів, які будуть продемонстровані в певних ситуаціях, використовують так званий **користувацький показ слайдів**.

Дайте відповіді на запитання

1. З якою метою створюють зовнішні гіперпосилання в комп'ютерних презентаціях?
2. З якою метою створюють внутрішні гіперпосилання в комп'ютерних презентаціях?
3. Як створити кнопки дій? Для чого вони призначені?
4. Для чого створюється користувацький показ слайдів?


Обговоріть і зробіть висновки

1. Чому комп'ютерні презентації з гіперпосиланнями є презентаціями з розгалуженнями?
2. Що спільне та відмінне у створенні та використанні гіперпосилань і кнопок дій у комп'ютерних презентаціях?
3. У яких ситуаціях доцільно створювати та використовувати користувацькі покази слайдів?

Виконайте завдання

1. Відкрийте комп'ютерну презентацію, наприклад з файлу **Розділ 3\Пункт 3.1\завдання 3.1.1.pptx**. Уставте на слайд 2 гіперпосилання для переходу до інших слайдів презентації, пов'язавши їх з фрагментами тексту. Пов'яжіть із зображенням на слайді 2 гіперпосилання для перегляду сторінки **Вікіпедії** з адресою <https://uk.wikipedia.org/wiki/Мікроскоп>. Уставте на слайди 3–6 кнопки дій для повернення на слайд 2. Збережіть комп'ютерну презентацію у вашій папці у файлі з тим самим іменем.


2. Відкрийте комп'ютерну презентацію, наприклад з файлу **Розділ 3\Пункт 3.1\завдання 3.1.2.pptx**. Створіть слайд з меню для переходів до слайдів 3, 5, 7 презентації. Уставте на створений слайд кнопку дій для завершення перегляду презентації. Збережіть комп'ютерну презентацію у вашій папці у файлі з тим самим іменем.


 3. Створіть комп'ютерну презентацію для перевірки знань з фізики з теми *Рівномірний прямолінійний рух* такої структури:

1. Титульний слайд з назвою презентації та вашим прізвищем.
2. Три слайди із заголовками *Швидкість руху*, *Час руху*, *Довжина шляху*.
3. Два слайди з макетом *Заголовки розділу* із заголовками *Правильно* та *Помилка*.





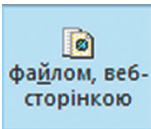


На слайдах *Швидкість руху*, *Час руху*, *Довжина шляху* розмістіть кнопки дій, підписами яких є математичні записи формул для розрахунку *швидкості*, *шляху*, *часу* під час рівномірного прямолінійного руху. Реалізуйте переходи з цих кнопок на слайди *Правильно* та *Помилка*. На слайді *Довжина шляху* розмістіть кнопку дії для завершення показу презентації. На слайдах *Правильно* та *Помилка* розмістіть кнопки дій для повернення на останній показаний слайд. Інші засоби переходів виберіть самостійно. Збережіть презентацію у вашій папці у файлі з іменем **завдання 3.1.3.pptx**.

4. Відкрийте комп'ютерну презентацію, наприклад з файлу **Розділ 3\Пункт 3.1\завдання 3.1.4.pptx**. Створіть користувачькі покази *Квіти*, *Пейзажі*, *Натюрморти*. Включіть до них титульний слайд і слайди з відповідними зображеннями. Збережіть комп'ютерну презентацію у вашій папці у файлі з тим самим іменем.

 **5.** Дослідіть можливості керування презентацією з використанням лише кнопок дій. Відкрийте комп'ютерну презентацію, наприклад з файлу **Розділ 3\Пункт 3.1\завдання 3.1.5.pptx**. У налаштуваннях переходів слайдів забороніть зміну слайдів клацанням миші. Розмістіть на слайдах кнопки дій для переходів між слайдами в порядку зменшення розміру площі материків так, щоб переглянути всі слайди та повернутися до початку маршруту. На слайді, з якого починається маршрут, розмістіть також кнопку для завершення перегляду презентації. Запустіть демонстрацію, перегляньте слайди на визначеному маршруті, завершіть перегляд. Збережіть презентацію у вашій папці у файлі з тим самим іменем.

 **6. Працюємо у групах.** Об'єднайтеся у групи з трьох осіб. Створіть презентацію за піснею «*Ой у лузі червона калина*». Розподіліть у групі завдання з пошуку даних і створіть слайди такого змісту: про автора слів пісні, про історію створення пісні, сучасне звучання та ілюстрування пісні. Створіть вибіркові покази для демонстрування слайдів указанного змісту. Збережіть презентацію у вашій папці у файлі з іменем **завдання 3.1.6.pptx**.

7. Установіть відповідність між зображеннями елементів керування та їх призначенням.

- | | | | |
|---|--|---|--|
| 1)  | 2) 
Користувацький показ слайдів ▾ | 3) 
місцем у документі | 4)  |
| 5) 
файлом, веб-сторінкою | 6)  | 7) 
Швидкі переходи ▾ | |

- А) Для створення внутрішнього посилання
- Б) Для вибору слайдів, які будуть демонструватися
- В) Для переходу на титульний слайд

- Г) Для переходу на домашній слайд
- Д) Для створення зовнішнього посилання
- Е) Для переходу на останній переглянутий слайд
- Ж) Для автоматизованого створення системи навігації



Готуємось до вивчення нового матеріалу

1. Як додати анімаційні ефекти до об'єктів слайдів комп'ютерної презентації?
2. Після настання яких подій може розпочинатися відтворення ефектів анімації на слайдах комп'ютерної презентації?



3.2. Використання тригерів у комп'ютерній презентації

У цьому пункті йтиметься про:

- ▶ поняття та призначення тригерів;
- ▶ створення тригерів;
- ▶ використання тригерів під час показу комп'ютерної презентації.

ТРИГЕРИ

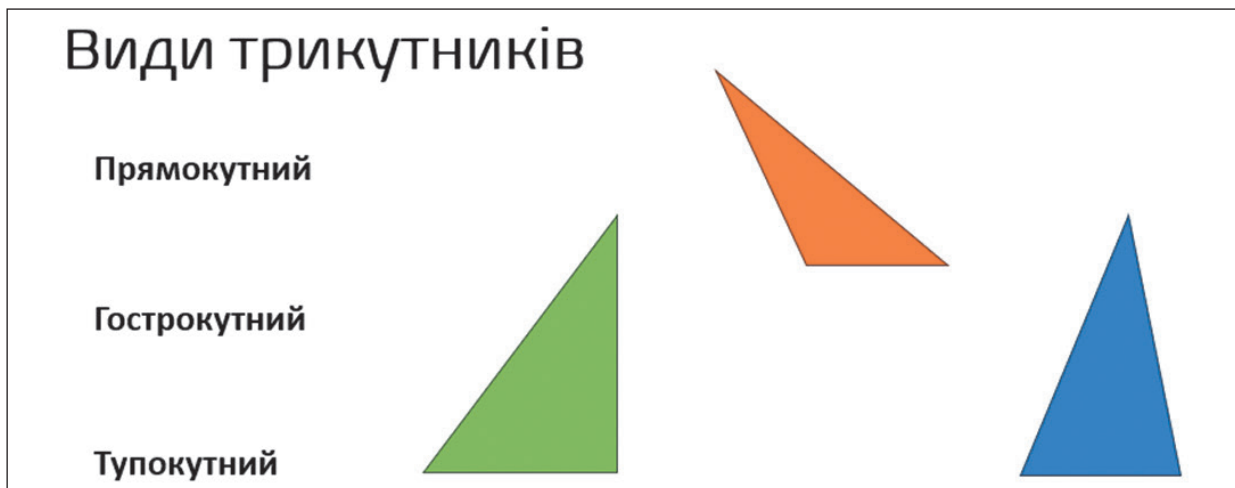


Пригадайте

- Як додавати ефекти анімації до об'єктів на слайдах комп'ютерної презентації?
- Після настання яких подій можуть відтворюватися ефекти анімації під час демонстрації комп'ютерної презентації?

Ви вже знаєте, що для ефектів анімації об'єктів на слайдах комп'ютерної презентації можна задати подію для початку відтворення ефекту: після клацання мишею; одночасно з попереднім ефектом; відразу після попереднього за номером ефекту. Але іноді буває потрібно, щоб анімаційні ефекти починали відтворюватися після вибору користувачем певного об'єкта на слайді.


Наприклад, потрібно, щоб у презентації про види трикутників (мал. 3.10) після вибору будь-якого трикутника відповідний йому напис перемістився та розташувався поруч із цим трикутником.



Мал. 3.10. Слайд презентації про види трикутників

Цього можна досягти завдяки використанню механізму **тригерів** (англ. *trigger* – курок, спусковий гачок).

Тригери призначені для початку відтворення деяких ефектів анімації після вибору певного об'єкта на слайді під час демонстрації комп'ютерної презентації. Об'єкт, після вибору якого починається анімація цього самого або інших об'єктів слайда, називають **перемикачем анімації**.

Під час демонстрації після наведення вказівника на перемикач анімації вигляд вказівника змінюється, як після наведення на гіперпосилання .

СТВОРЕННЯ ТРИГЕРІВ

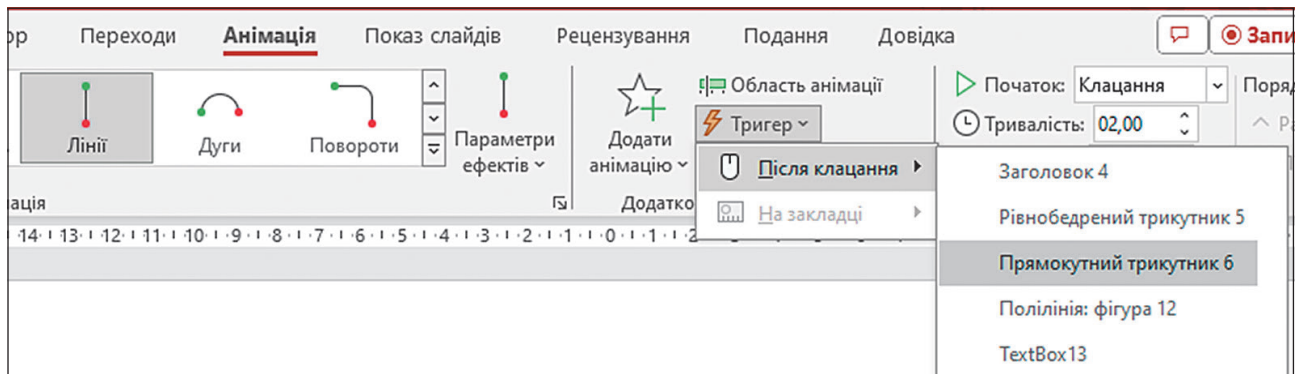


- Які ви знаєте властивості ефектів анімації об'єктів слайдів?
- Як установити значення властивостей ефектів анімації?
- Як відкрити **Область анімації**?

Під час розробки структури комп'ютерної презентації з тригерами потрібно визначити не лише до яких об'єктів слайда потрібно додати ефекти анімації, а й після вибору яких об'єктів (перемикачів анімації) повинні відтворитися ці ефекти.


Для створення тригера слід:


1. Додати до деякого об'єкта ефект анімації.
2. Виконати **Анімація** ⇒ **Додаткові параметри анімації** ⇒ **Тригер** ⇒ **Після клацання**.
3. Вибрати у списку, що відкриється, ім'я об'єкта, який буде перемикачем анімації (мал. 3.11).

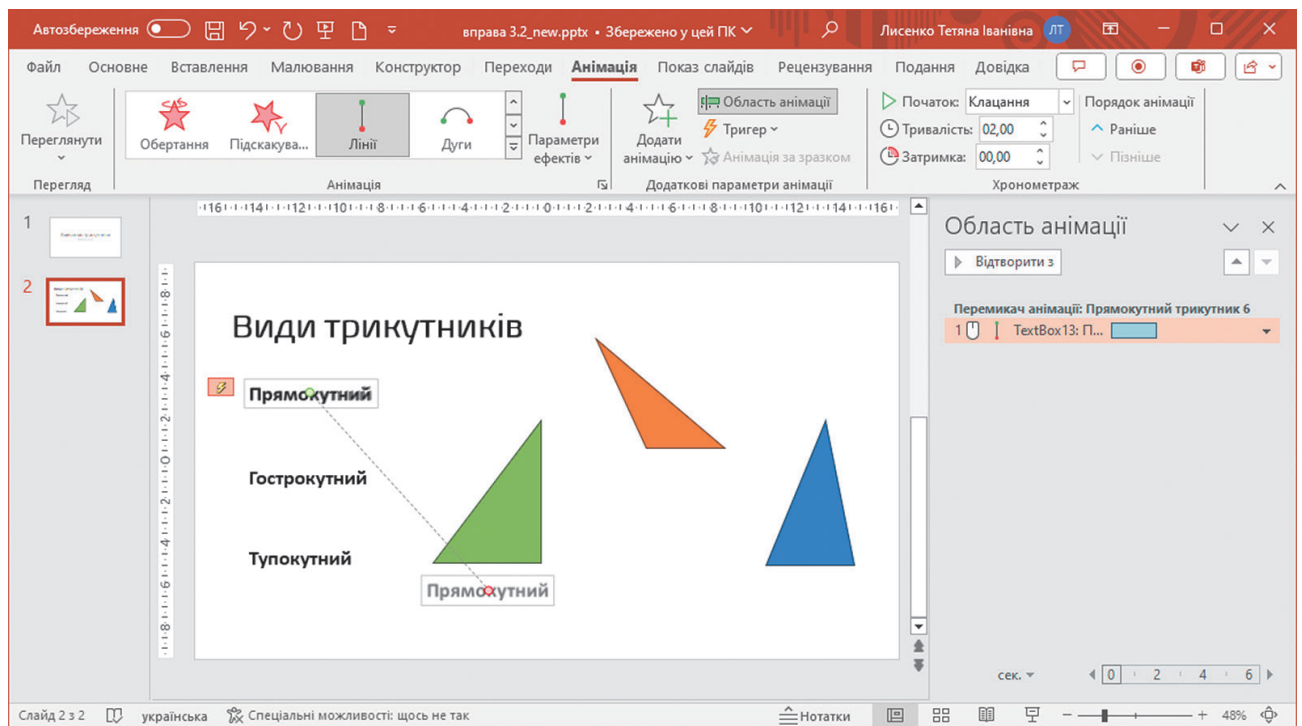


Мал. 3.11. Створення тригера


Після виконання цього алгоритму:

- на слайді поруч з об'єктом, до якого додано ефект анімації, з'являється значок ;
- в **Області анімації** з'являється напис **Перемикач анімації** та ім'я перемикача, а нижче від нього – рядок доданого ефекту анімації (мал. 3.12).

На малюнку 3.12 показано, що перемикачем анімації є об'єкт з іменем *Прямокутний трикутник 6*. Після його вибору розпочнеться анімація текстового об'єкта з текстом *Прямокутний*, який має ім'я *TextBox 13*, – почне відтворюватись ефект анімації *Лінії*  з групи *Шляхи переміщення*.

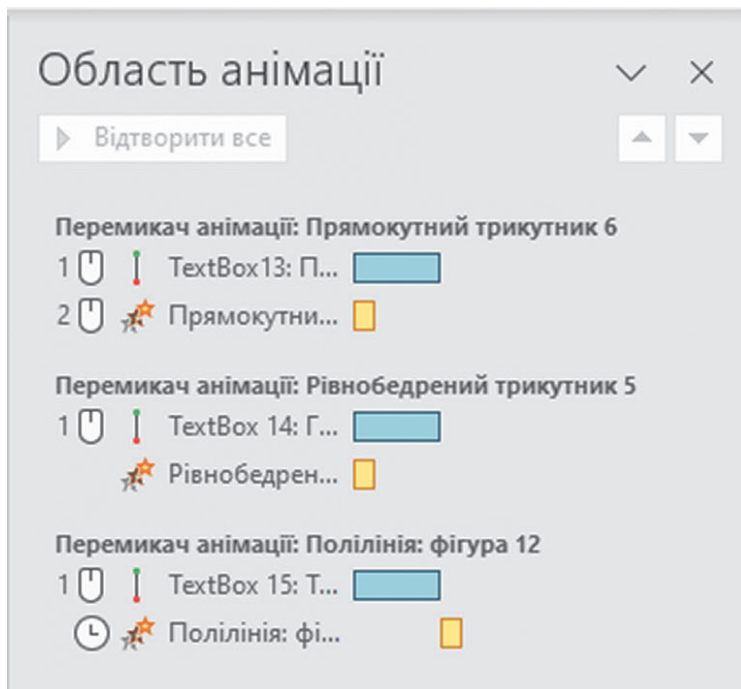


Мал. 3.12. Повідомлення про перемикач анімації в **Області анімації**



Один об'єкт може бути перемикачем анімації, який запускає кілька ефектів анімації. Наприклад, після вибору трикутника не лише відбувається переміщення напису, а й змінюється колір заливки вибраного трикутника. Для цього можна установити вищеописаним способом один і той самий перемикач анімації для кількох ефектів анімації або в **Області анімації** розмістити рядок ефекту під відповідним написом про перемикач анімації, використовуючи кнопки . Щоб ефекти анімації після вибору перемикача анімації почали відтворюватись одночасно або послідовно один після одного, потрібно змінити подію для початку відтворення ефекту.

На одному слайді може бути створена довільна кількість перемикачів анімації (мал. 3.13).

Щоб відмовитися від встановленого тригера, можна розмістити рядок ефекту анімації вище від усіх написів про



Мал. 3.13. Вигляд **Області анімації** з кількома визначеними тригерами

перемикачі анімації, використовуючи кнопки   в **Області анімації**. Якщо жоден ефект не буде залежним від деякого перемикача, то напис про цей перемикач анімації зникне.

Також, щоб відмовитися від тригера, можна вибрати на слайді об'єкт з ефектом анімації, виконати **Анімація** \Rightarrow **Додаткові параметри анімації** \Rightarrow **Тригер** \Rightarrow **Після клацання** та повторно вибрати ім'я об'єкта, який позначено як перемикач анімації.

ПЕРЕЙМЕНУВАННЯ ОБ'ЄКТІВ СЛАЙДІВ



Пригадайте

- Де можна побачити імена об'єктів слайда? З якою метою їх використовують?
- Як перейменувати об'єкти операційної системи?

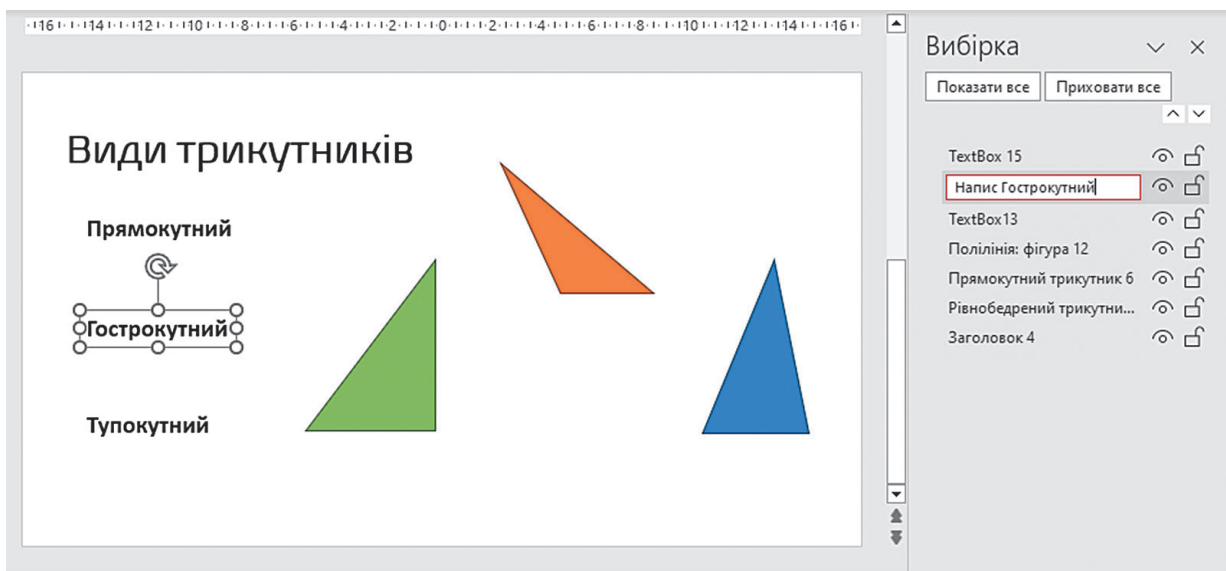
Імена об'єктів на слайдах комп'ютерної презентації створюються автоматично під час розміщення об'єктів на слайді.

Ім'я за замовчуванням складається з указівки типу об'єкта та номера об'єкта на слайді. Наприклад, ім'я заголовка титульного слайда за замовчуванням **Заголовок 1**, ім'я підзаголовка – **Підзаголовок 2**, ім'я покажчика місця заповнення на слайді макета *Назва і вміст* – **Місце для вмісту 2**. Іменами вставлених об'єктів може бути **Графіка 3** – ім'я піктограми, **Рисунок 5** – ім'я зображення з графічного файлу, **Овал 9** – ім'я графічного примітива тощо.

Якщо на слайді розміщено багато об'єктів, то під час вибору перемикачів анімації буває складно правильно вибрати потрібний об'єкт за його іменем. Тому бажано, щоб об'єкти мали зрозумілі імена, які відповідають змісту презентації.

Імена об'єктів можна змінювати, використовуючи область виділення **Вибірка**. Для цього слід:

1. Відкрити область виділення **Вибірка**, виконавши **Основне** ⇒ **Редагування** ⇒ **Виділити** ⇒ **Область виділення**.
2. Вибрати об'єкт на слайді.
3. Вибрати рядок з іменем об'єкта в області **Вибірка**.
4. Увести нове ім'я об'єкта (мал. 3.14).



Мал. 3.14. Перейменування об'єкта в області виділення **Вибірка**

5. Натиснути клавішу **Enter** або вибрати будь-яку точку за межами рядка з іменем об'єкта.

У такий спосіб можна перейменувати об'єкти слайда для спрощення вибору перемикачів анімації та правильного налаштування ефектів анімації.



Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви можете знайти за адресою <https://cutt.ly/WwNTrHU0> або QR-кодом.



Найважливіше в цьому пункті

Тригери призначені для початку відтворення деяких ефектів анімації після вибору певного об'єкта на слайді під час демонстрації комп'ютерної презентації. Об'єкт, після вибору якого починається анімація цього самого або інших об'єктів слайда, називають **перемикачем анімації**.

Під час розробки структури комп'ютерної презентації з тригерами потрібно визначити, до яких об'єктів слайда потрібно додати ефекти анімації та після вибору яких об'єктів (перемикачів анімації) повинні відтворитися ці ефекти.

Один об'єкт може бути перемикачем анімації, який запускає кілька ефектів анімації. На одному слайді може бути створена довільна кількість перемикачів анімації.

Імена об'єктів на слайдах комп'ютерної презентації створюються автоматично під час розміщення об'єктів на слайді. Для спрощення вибору перемикачів анімації та правильного налаштування ефектів анімації можна змінювати імена об'єктів, використовуючи область виділення **Вибірка**.



Дайте відповіді на запитання

1. Для чого призначені тригери?
2. Який об'єкт називають *перемикачем анімації*?
3. Як створити тригер для анімації об'єкта?

4. Як перейменувати об'єкти слайдів комп'ютерної презентації? Навіщо це потрібно робити?

5. Чи може об'єкт слайда бути перемикачем анімації для ефекту, доданого до цього самого об'єкта?



Обговоріть і зробіть висновки

1. Які переваги використання тригерів у комп'ютерних презентаціях? Наведіть приклади.

2. Чи важливий порядок розміщення повідомлень про тригери в **Області анімації**? Чи важливий порядок розташування рядків ефектів анімації, що запускаються одним і тим самим перемикачем анімації?

3. Чи може один і той самий ефект анімації бути пов'язаний з різними перемикачами анімації? Як забезпечити відтворення анімації об'єкта після вибору різних перемикачів?




Виконайте завдання


1. Відкрийте комп'ютерну презентацію, наприклад з файлу **Розділ 3\Пункт 3.2\завдання 3.2.1.pptx**. Створіть тригери для запуску ефектів анімації, доданих до зображення сонця, щоб зображення переміщувалось у певному напрямку після вибору відповідної стрілки. Збережіть комп'ютерну презентацію у вашій папці у файлі з тим самим іменем.

2. Відкрийте комп'ютерну презентацію, наприклад з файлу **Розділ 3\Пункт 3.2\завдання 3.2.2.pptx**. Створіть тригери для запуску ефектів анімації, доданих до зображення сонця та місяця, щоб після вибору кнопки *День* на слайді одночасно з'являлось сонце та зникав місяць, а після вибору кнопки *Ніч* з'являвся місяць і зникало сонце. Збережіть комп'ютерну презентацію у вашій папці у файлі з тим самим іменем.

3. Відкрийте комп'ютерну презентацію, наприклад з файлу **Розділ 3\Пункт 3.2\завдання 3.2.3.pptx**. Додайте до об'єктів

слайдів ефекти анімації та створіть тригери таким чином, щоб презентацію можна було використовувати для самоперевірки знань з теми *Пристрої комп'ютера*. Під час демонстрації після переходу до кожного слайда на ньому повинні відобразитися заголовки, зображення та кнопки *Так* і *Ні*, а інші об'єкти повинні бути приховані. Після вибору кнопок повинен з'являтися один з написів: *Правильно* або *Помилка*. Збережіть комп'ютерну презентацію у вашій папці у файлі з тим самим іменем.

 **4.** Проведіть експеримент з метою реалізувати один ефект анімації з різними перемикачами анімації. Відкрийте комп'ютерну презентацію з файлу **Розділ 3\Пункт 3.2\завдання 3.2.4.pptx**. Під час демонстрації після вибору кожної кнопки на слайді повинно з'являтися повідомлення *Правильно* або *Помилка* та зникати невідповідне повідомлення. Додайте до об'єктів слайда ефекти анімації та створіть тригери, щоб реалізувати указаний сценарій. *Підказка:* додайте до написів кілька однакових ефектів, використовуючи кнопку **Додати анімацію** на вкладці **Анімація**. Для кожного з однакових ефектів можна призначити окремий перемикач анімації. Збережіть презентацію у вашій папці у файлі з тим самим іменем.

 **5.** Створіть комп'ютерну презентацію з географії з теми *Карти материків та океанів* такої структури:

1. Титульний слайд з назвою презентації та вашим прізвищем.
2. Три слайди із заголовками *Який це материк?*, зображеннями контуру одного з материків і кнопками з назвами всіх материків.

Потрібні зображення знайдіть в Інтернеті. Додайте ефекти анімації до об'єктів слайдів і створіть тригери таким чином.

ном, щоб презентацію можна було використовувати для самоперевірки знань. Під час демонстрації після вибору кнопок з назвами материків повинен з'являтися один із написів – *Правильно* або *Помилка* – залежно від правильності відповіді. Збережіть презентацію у вашій папці у файлі з іменем завдання **3.2.5.pptx**.



Готуємось до вивчення нового матеріалу

1. Які дані називають *мультимедійними*?
2. Які пристрої використовують для введення та виведення звукових і відеоданих?



Оцініть свої знання та вміння з розділу «Комп'ютерні презентації»

Оцініть свої навчальні досягнення із цього розділу (початковий, середній, достатній, високий рівень).

- Я знаю, що таке гіперпосилання і для чого їх використовують.
- Я розумію, які комп'ютерні презентації називають презентаціями з розгалуженнями.
- Я знаю, які засоби можуть бути використані для створення презентацій з розгалуженнями.
- Я вмю додавати зовнішні та внутрішні гіперпосилання на слайди комп'ютерних презентацій.
- Я вмю додавати кнопки дій на слайди комп'ютерних презентацій.
- Я вмю налаштовувати користувацький показ слайдів комп'ютерних презентацій.
- Я вмю створювати тригери в комп'ютерних презентаціях для керування анімаційними ефектами.
- Я вмю перейменовувати об'єкти на слайдах комп'ютерних презентацій.

- Я вмію створювати комп'ютерні презентації, у яких відбувається опрацювання подій.

Повторіть той матеріал, який ви знаєте недостатньо.



Практична робота № 5. **«Комп'ютерні презентації»**

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

Завдання. Налаштуйте показ комп'ютерної презентації *Що я знаю про Україну*, яка міститься у файлі **Розділ 3\Пункт 3.2\практична робота.pptx**.

1. Створіть з відповідних написів на слайді 2 гіперпосилання для переходів на сторінку з адресою uk.wikipedia.org/wiki/Київ і на слайди з номерами 3, 4, 6.
2. Уставте на слайді 2 кнопку дій для завершення перегляду презентації.
3. Створіть на слайдах 3, 5, 7 кнопки для переходу на слайд 2.
4. Зробіть на слайді 6 піктограму гіперпосиланням для переходу на сторінку з адресою uk.wikipedia.org/wiki/Гімн_України.
5. Додайте ефекти анімації до текстових написів на слайді 7 для їх переміщення до відповідних зображень. Перемикачами анімації для початку відтворення ефектів анімації зробіть відповідні зображення.

З М І С Т

Дорогі семикласниці та семикласники!	
Шановні вчительки та вчителі!	3

Розділ 1. ПОШУК В ІНТЕРНЕТІ. ЕЛЕКТРОННА ПОШТА. ХМАРНІ СЕРВІСИ

1.1. Пошук відомостей в Інтернеті. Розширений пошук. Створення списків джерел і закладок	5
1.2. Збирання даних. Перевірка достовірності. Ознаки мови ворожнечі, маніпуляції, пропаганди	18
1.3. Електронне листування. Електронна поштова скринька	28
1.4. Адресна книга. Етикет і правила безпечного електронного листування	39
<i>Практична робота № 1. «Пошук в Інтернеті. Електронне листування»</i>	<i>49</i>
1.5. Хмарні сервіси	50
1.6. Створення онлайн-документів і керування доступом до них . .	64
<i>Практична робота № 2. «Хмарні сервіси»</i>	<i>77</i>

Розділ 2. АЛГОРИТМИ ТА ПРОГРАМИ

2.1. Комп'ютерні програми. Мови програмування. Програми-транслятори	78
2.2. Мова програмування PYTHON . Середовище створення проектів IDLE . Створення лінійних проектів	90
2.3. Розгалуження. Проекти з розгалуженнями	104
2.4. Підпрограми	120
<i>Практична робота № 3. «Проекти з підпрограмами та з розгалуженнями»</i>	<i>128</i>
2.5. Проекти з вікнами	128
2.6. Події. Обробники подій. Вікна повідомлень	134
<i>Практична робота № 4. «Проекти для змінення значень властивостей вікна. Використання вікна повідомлень»</i>	<i>145</i>
2.7. Створення графічних зображень на вікні	146

Розділ 3. КОМП'ЮТЕРНІ ПРЕЗЕНТАЦІЇ

3.1. Створення комп'ютерних презентацій з розгалуженнями	158
3.2. Використання тригерів у комп'ютерній презентації	173
<i>Практична робота № 5. «Комп'ютерні презентації»</i>	<i>183</i>