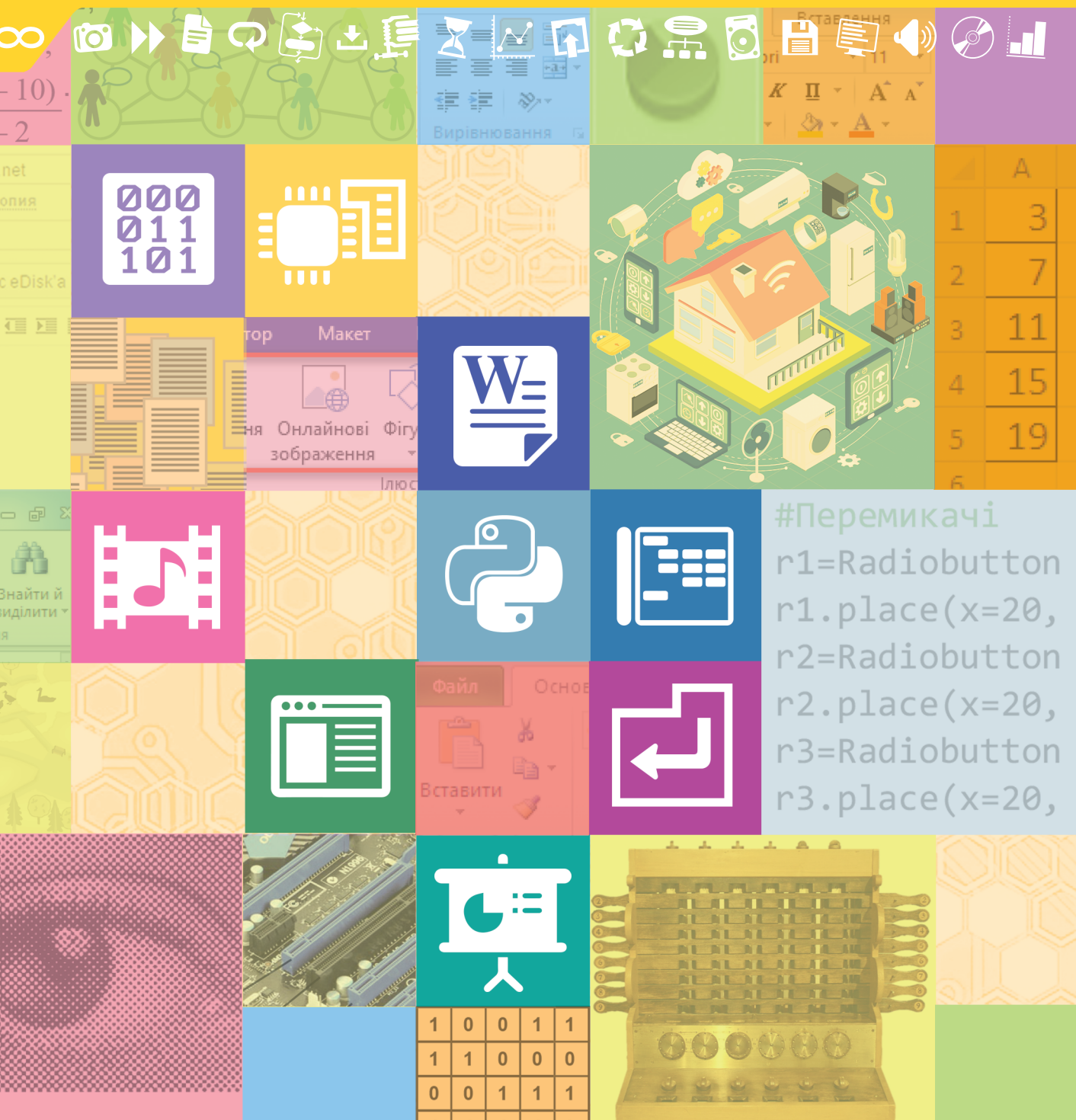


Ольга Казанцева, Ірина Стеценко

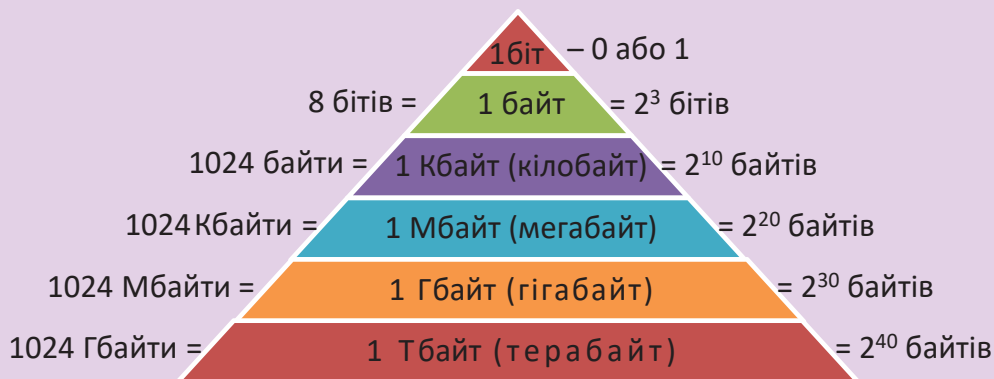
ІНФОРМАТИКА 8

К л а с



КОДУВАННЯ ДАНИХ

Вимірювання інформації



Змістовий підхід

Кількість інформації залежить від її змісту

b – кількість інформації

N – кількість ймовірних подій



$$2^b = N$$

Алфавітний підхід

Кількість інформації залежить від потужності алфавіту

N – потужність алфавіту

V – обсяг повідомлення

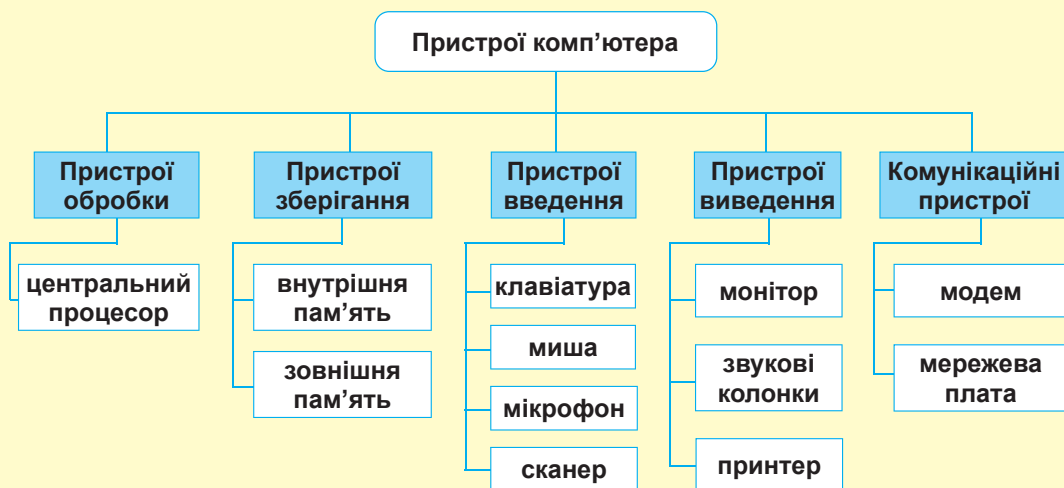
k – кількість символів у тексті

b – інформаційна вага символу

$$V = k \cdot b$$



КЛАСИФІКАЦІЯ ПРИСТРОЇВ КОМП'ЮТЕРА ЗА ЇХНІМ ПРИЗНАЧЕННЯМ



СТРУКТУРА HTML-ДОКУМЕНТА

```

<html>
  <head>
    <title>Заголовок</title>
  </head>
  <body>
    Тіло документа
  </body>
</html>
  
```

ТЕГИ ДЛЯ СТВОРЕННЯ СТРУКТУРИ ДОКУМЕНТА

<p></p>	створення абзацу
 	розрив рядка
<hr>	горизонтальна лінія
<h1></h1> ... <h6></h6>	заголовки

ТЕГИ ФОРМАТУВАННЯ ШРИФТУ

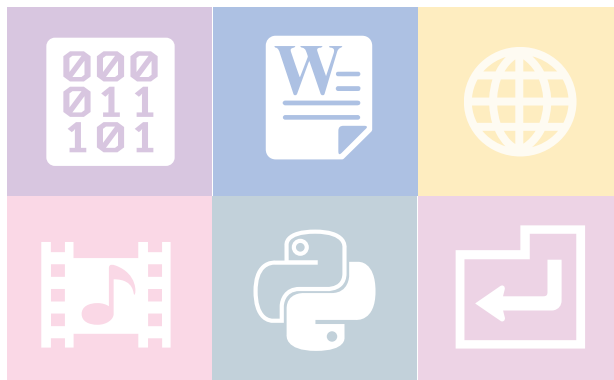
	напівжирне накреслення
<i>...</i>	курсив
<u>...</u>	підкреслений
	підрядковий текст
	надрядковий текст
<strike></strike>	перекреслений текст
<big></big>	збільшення розмірів шрифту на 1
<small></small>	зменшення розмірів шрифту на 1
	параметри шрифту

Ольга Казанцева
Ірина Стеценко

ІНФОРМАТИКА

ПІДРУЧНИК ДЛЯ 8 КЛАСУ
закладів загальної середньої освіти

Рекомендовано Міністерством освіти і науки України



ТЕРНОПІЛЬ
НАВЧАЛЬНА КНИГА – БОГДАН
2021

УДК 004(075.3)
К 14

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

ВИДАНО ЗА РАХУНОК ДЕРЖАВНИХ КОШТІВ. ПРОДАЖ ЗАБОРОНЕНО

Казанцева О. П.

К 14 Інформатика : підручник для 8 кл. закладів загальн. середн. освіти / О.П. Казанцева, І.В. Стеценко. — Тернопіль : Навчальна книга – Богдан, 2021. — 256 с. : іл.

ISBN 978-966-10-6440-8

Підручник містить систематизований виклад інформатики для учнів 8-го класу. Матеріал відповідає новій програмі Міністерства освіти і науки України.

Для учнів 8-х класів закладів загальної середньої освіти та вчителів інформатики.

УДК 004(075.3)

*Охороняється законом про авторське право.
Жодна частина цього видання не може бути відтворена
в будь-якому вигляді без дозволу видавництва.*

© Казанцева О.П., Стеценко І.В., 2021
© Навчальна книга – Богдан, виключна ліцензія на видання, оригінал-макет, 2021

ISBN 978-966-10-6440-8

УМОВНІ ПОЗНАЧЕННЯ



інтерактивна
електронна складова

Шановні восьмикласники та восьмикласниці!

Цей підручник допоможе вам продовжити вивчення інформатики. У розділі 1 ви познайомитеся з кодуванням та вимірюванням інформації, дізнаєтеся про призначення складових комп'ютера. Вивчаючи матеріал розділу 2, ви будете працювати в текстовому редакторі MS Word. Розділ 3 познайомить вас з основами вебдизайну при створенні власного сайту. З розділу 4 ви дізнаєтеся про мультимедійні продукти та технології. В розділі 5 розпочнете вивчення основ програмування в середовищі мови Python, створюючи як обчислювальні програми, так і Windows-додатки. Підручник містить практичні вправи для виконання на комп'ютері. Файли-заготовки до деяких вправ знаходяться в інтерактивній складовій підручника в папці **Files**.

ПУТІВНИК ПІДРУЧНИКОМ

РОЗДІЛ 4. Опрацювання об'єктів мультимедіа

Контрольні запитання та завдання

1. Які служби розміщення відеоконтенту ви знаєте? Чи користувалися ви ними? Якщо так, опишіть роботу з цими службами.
2. Як ви впорядкуєте свою колекцію знімків?
3. Опишіть можливості відеохостингу YouTube.
4. Як здійснити завантаження відео з YouTube?

Питання для роздумів

- 1*. В якому напрямку, на вашу думку, розвиватимуться платформи відеохостингу?
2. Яких функцій, на ваш погляд, не вистачає відеохостингу YouTube?

Завдання для досліджень

1. Дізнайтеся, як підписатися на канал в YouTube та як відписатися від нього.
2. Які можливості редагування відео містить вбудований редактор YouTube?
- 3*. Знайдіть інформацію про відеохостинг Vimeo.
4. Як, завантажуючи власні відео на YouTube, можна заробляти гроші?

Висновки

Мультимедіа — це програмний продукт, що містить колібражен, текстів, а також даних, які супроводжуються звуком, анімацією та іншими візуальними ефектами, оснащені терактивним інтерфейсом з елементами керування.

Захоплення відео — процес перетворення відеосигналу із знісного джерела на цифровий потік і запис його у відеофайл з метою обробки, зберігання та відтворення.

Конвертер — це програма, що перетворює файли з одного формату в інший.

Для здійснення захоплення відео зручно використовувати програму **FastStone Capture**.

Найбільш поширені формати фотографій є <https://www.faststone.org/>.

Для завантаження цієї популярності наб...

Вправа 18

Завдання. Виконайте завдання вправ у парі. Ви маєте надати доступ до власного документа сусіду/сусідці та отримати посилання на документ, доступ до якого вам надано сусідом/сусідкою.

1. Відкрийте документ **Анкета.docx**.
2. Додайте у нього власні три запитання.
3. Збережіть документ у хмарі з назвою **Вправа_18_Прізвище.docx**.
4. Надайте доступ до свого документа сусідові/сусідці для редагування та читачу/читачці для перегляду. Попередньо дізнайтесь адреси їхніх електронних скриньок.

У рубриці підсумовано матеріал розділу та виокремлено основні поняття і терміни


Пропоновані питання вимагають творчого пошуку і поглиблених знань:
* — достатній рівень,
** — високий рівень навчальних досягнень

Практичні завдання дають можливість закріпити вивчений матеріал, працюючи за комп'ютером

Вміщені запитання і завдання дають змогу перевірити засвоєні знання

Ці завдання розраховані на учнів, які захоплюються інформатикою

ПУТІВНИК ІНТЕРАКТИВНОЮ ЕЛЕКТРОННОЮ СКЛАДОВОЮ ПІДРУЧНИКА

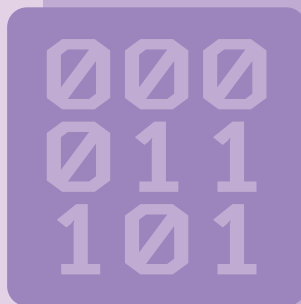
Зручним супроводженням усього матеріалу підручника інформатики є його інтерактивна електронна складова, посилання на використання якої позначено піктограмою .

Головне вікно програми



Розділ 1

Кодування даних та апаратне забезпечення



§1.

Роль і місце кодування в інформаційних процесах

1.1.

ЗБЕРІГАННЯ, ПЕРЕДАВАННЯ ТА ОБРОБКА ІНФОРМАЦІЇ

У 5 класі ви вже ознайомилися з тим, що таке інформація та інформаційні процеси, а саме: пошук, зберігання, передавання, обробка, використання та захист інформації.

Для передавання і зберігання інформації застосовують різні системи знаків. За способом сприйняття знаки можна розділити на:

- зорові (наприклад, літери, ноти, дорожні знаки тощо);
- слухові (це може бути усне мовлення, звукові сигнали й т. ін.);
- дотикові (шриффт Брайля для незрячих людей).

Сукупність знаків, що відображають певну інформацію, називають *повідомленням*.

Повідомлення передають на відстань за допомогою *матеріальних носіїв* (папір, флешпам'ять, магнітний диск) або *сигналів* (звукових, електричних, оптичних, радіосигналів).

Важливим інформаційним процесом є *обробка інформації*, тобто перетворення її з одного виду на інший (чи той самий), що виконується за строгими формальними правилами. Обробка може бути пов'язана з отриманням нової інформації. Проте нас цікавитиме обробка, внаслідок якої змінюється форма, а не зміст інформації. До цього виду обробки належать структурування та кодування інформації. Ці процеси необхідні для



Пригадайте поняття інформація та інформаційні процеси



забезпечення інших інформаційних процесів, оскільки інформація часто існує у формі, не прийнятній для її передавання та зберігання. Тому для перетворення інформації в прийнятну форму застосовують *кодування*, про яке йтиметься далі.

1.2.

КОДУВАННЯ ІНФОРМАЦІЇ

Розглянемо докладніше, що таке кодування і як його застосовують у комп'ютерних технологіях.

Кодування інформації — це процес перетворення інформації з форми, зручної для безпосереднього використання, у форму, зручну для передавання, зберігання або автоматичної обробки.

Прикладом кодування є запис усних повідомлень у текстовій формі, музики — у вигляді нот.

Текстове кодування надає можливість нескінченну кількість текстових повідомлень подати з допомогою алфавіту, що складається зі скінченної кількості знаків (наприклад, український алфавіт має 33 літери). Аналогічно, все різноманіття мелодій записують, використовуючи лише сім нот.

У комп'ютері застосовується двійкове кодування, коли інформація будь-якого виду (текстова, числова, графічна, звукова, відео) записується у вигляді коду, алфавіт якого складається лише з двох символів, що умовно позначаються, як «0» та «1».

Це пов'язано з тим, що апаратні складові сучасних комп'ютерів здатні обробляти лише сигнали у двох станах. Фізично змодельовати ці стани можна по-різному: намагнічено/ненамагнічено (жорсткий диск), розімкнено/замкнено, сигнал з амплітудами 1/0 (рис. 1.1).

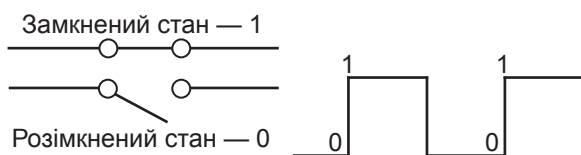


Рис. 1.1. Фізична реалізація станів «0» та «1»

Для того, щоб із закодованого повідомлення видобути інформацію, потрібно виконати операцію *декодування*.

Не слід плутати кодування інформації з її *шифруванням* — перетворенням, що виконують з метою ускладнити або унеможливити розуміння чи зміну інформації в разі її перехоплення.

Основна відмінність між кодуванням та шифруванням полягає в тому, що для відновлення закодованого повідомлення достатньо знати правило заміни, а для розшифрування зашифрованого повідомлення на додаток до правила потрібно мати ще й ключ до шифру.



Дізнайтеся більше про шифрування

Шифрування

перетворення, що виконують з метою **ускладнення** розуміння інформації в разі її перехоплення.

Для розшифрування необхідно знати **правило та ключ**



Кодування

перетворення інформації з форми, зручної для використання, у форму, зручну для автоматичної обробки.

Для розкодування достатньо знати **правило**

ВПРАВА 1



Завдання. Виконайте завдання у файлі **Вправа_1.doc**. Розкодуйте прислів'я, заповніть відповідні клітинки таблиць, представлені у файлі.



Контрольні запитання та завдання

1. Що таке повідомлення?
2. Які способи передачі повідомлень ви знаєте?
3. Що таке кодування інформації?
4. Чим кодування відрізняється від шифрування?



Питання для роздумів

1. Для чого використовують кодування?
2. Де в сучасній практиці використовують шифрування?



Завдання для досліджень

- 1*. Поясніть, що таке штрих-код, яке його походження. Наведіть приклади використання штрих-кодів.
- 2*. Що таке QR-код? Де він використовується?

§2.

Системи числення

2.1.

ПОНЯТТЯ СИСТЕМИ ЧИСЛЕННЯ

Із попередніх уроків ми дізналися, що у комп'ютері застосовується двійкове кодування. Ми ж з вами звикли до чисел, які записуються з допомогою десяти цифр. Далі розглянемо правила, за якими комп'ютер перетворює зрозумілі нам числа у двійковий код і навпаки.

Числа записують з допомогою певних символів та за певними правилами. Ці правила називають *системою числення*.



Дізнайтеся більше про системи числення

Символи, з допомогою яких записують числа, називають *цифрами*, їхню сукупність — *алфавітом*. Кількість цифр у алфавіті називають його *розмірністю*.

Згідно з цими визначеннями, знайома вам десяткова система числення є системою числення з розмірністю 10, у якій для запису чисел використовують цифри 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

У кодуванні даних на комп'ютері використовують усього дві цифри, отже, система числення, у якій виконує дії комп'ютер, має назву *двійкова система числення*. Взагалі можуть існувати системи числення з будь-якою розмірністю, наприклад, трійкова, вісімкова, шістнадцяткова тощо.

У таблиці нижче наведено приклади деяких систем числення.

Таблиця 2.1. Приклади систем числення

Система числення	Розмірність алфавіту	Цифри
Двійкова	2	0, 1
Трійкова	3	0, 1, 2
Вісімкова	8	0, 1, 2, 3, 4, 5, 6, 7
Десяткова	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Шістнадцяткова	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Зверніть увагу на нестандартне використання цифр у шістнадцятковій системі числення. Тут деякі цифри позначено літерами латинського алфавіту, цифра A відповідає числу 10 в десятковій системі, цифра B — числу 11 і т. д. до 15. Шістнадцяткове подання чисел доволі часто використовується в кодуванні різних даних на комп'ютері.

У комп'ютері інформація зберігається та обробляється у двійковій системі, що містить лише дві цифри — 0 та 1.

Послідовність двійкових цифр називають *двійковим кодом*. Кожна цифра (розряд) у двійковому коді називається *бітом*. На практиці частіше використовують більшу одиницю — байт, який дорівнює восьми бітам.

2.2.

ПЕРЕВЕДЕННЯ ЧИСЛА З ДЕСЯТКОВОЇ СИСТЕМИ ЧИСЛЕННЯ У ДВІЙКОВУ

Як записати десяткове число у двійковій формі, щоб його «зрозумів» комп'ютер? Необхідно перевести число з десяткової системи числення у двійкову.

Для переведення числа з десяткової системи числення у двійкову його потрібно послідовно ділити на 2 до тих пір, поки в частці не отримаємо 1. Відповідне число у двійковій системі числення утворюється як остання частка та послідовність остач від ділення, записаних у зворотному порядку, починаючи з останньої.

Приклад переведення числа 37 з десяткової системи числення у двійкову наведено на рис. 2.1.

$$\begin{array}{r|l}
 37 & 2 \\
 \hline
 36 & 18 \quad 2 \\
 & 18 \quad 0 \quad 2 \\
 & 8 \quad 1 \quad 2 \\
 & 4 \quad 0 \quad 2 \\
 & 2 \quad 0 \quad 2 \\
 & 1 \quad 0 \quad 1
 \end{array}$$

$37 = 100101_2$

Рис. 2.1. Переведення числа 37 з десяткової системи у двійкову

2.3.

ПЕРЕВЕДЕННЯ ЧИСЛА З ДВІЙКОВОЇ СИСТЕМИ ЧИСЛЕННЯ В ДЕСЯТКОВУ

Звичайно, переводити числа можна і навпаки — з двійкової системи в десяткову. Послідовність дій для переведення числа з двійкової системи числення в десяткову розглянемо на наступному прикладі.

Спочатку пронумеруємо цифри числа, причому нумерувати починаємо справа наліво від нуля.

$$\begin{array}{cccccccc}
 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\
 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0
 \end{array}
 \cdot 2^{} =$$

Далі записуємо цифри числа і множимо їх на 2 у тому степені, який номер має цифра.

$$= 1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 =$$

Даний вираз містить два доданки, що містять добуток числа на 0, відповідно ці доданки будуть дорівнювати нулю і ними можна знехтувати. Також можна спростити доданки, відкинувши одиниці. Після спрощення сума матиме такий вигляд:

$$= 2^7 + 2^5 + 2^4 + 2^2 + 2^1 + 2^0 =$$

Додавши числа, отримаємо відповідь 183.

$$= 128 + 32 + 16 + 4 + 2 + 1 = 183_{10}$$



Дізнайтеся, як перевести числа з однієї системи числення в іншу з допомогою калькулятора

Отже, для переведення числа з двійкової системи числення в десяткову визначають суму добутків цифр числа на 2 у тому степені, який порядковий номер має цифра.

ВПРАВА 2.1



Завдання. Виконайте завдання у файлі **Вправа_2_1.docx**. Розгадайте кросворд, увівши відповіді до сітки кросворда.

ВПРАВА 2.2



Виконайте додаткові практичні вправи



Завдання. Виконайте завдання у файлі **Вправа_2_2.docx**. Створіть зображення, перетворюючи координати точок з двійкової системи числення в десяткову.



Контрольні запитання та завдання

1. Що таке система числення?
2. Скільки цифр має двійкова система числення? Які це цифри?
3. Як позначаються цифри шістнадцяткової системи числення?



Питання для роздумів

1. Які існують системи числення?
- 2*. Чи може число, записане в десятковій системі числення, дорівнювати числу, записаному в двійковій системі числення?



Завдання для досліджень

- 1*. Де в інформатиці використовується шістнадцяткова система числення?
- 2*. Які системи числення мають своє застосування на практиці?



Пройдіть
тест

§3.

Кодування чисел

3.1.

ЦІЛІ ЧИСЛА В КОМП'ЮТЕРІ

Сучасні комп'ютери — це пристрої, здатні обробляти сигнали, які мають два стани (їх позначено цифрами 0 та 1). Отже, щоб комп'ютер міг зберегти та обробити числові значення, текст, зображення, звук чи відео, їх потрібно піддати двійковому кодуванню. Спосіб кодування інформації в комп'ютері насамперед залежить від виду інформації, а саме, від того, що потрібно кодувати: числа, текст, графічні зображення чи звук.

Двійкові коди цілих чисел будують, записуючи їх у двійковій системі числення. Пам'ять комп'ютера можна умовно зобразити, як аркуш паперу в клітинку. В кожній такій клітинці (комірці пам'яті) міститься 0 чи 1.

Такий вигляд після двійкового кодування має в пам'яті інформація будь-якого виду: текстова, числова, графічна (рис. 3.1). Як її декодувати? Адже ми не знаємо навіть, де починається і де закінчується елемент даних (число, літера, піксель тощо). Один із способів вирішення цієї проблеми полягає в тому, щоб заздалегідь домовитися, за яким правилом дані записуються в клітинки.



Рис. 3.1

Для більшої наочності перейдемо на десяткову систему. Домовимось, що для запису числа відводяться чотири клітинки. Знаючи це, легко визначити, що в наведеному повідомленні (рис. 3.2) записано три числа: 5671, 137 і 23.

Для зберігання чисел у пам'яті комп'ютера також відводиться певна кількість розрядів, і записуються числа у двійковій системі числення.

Але водночас, як показано на попередньому прикладі, завжди існує обмеження на максимальне число.

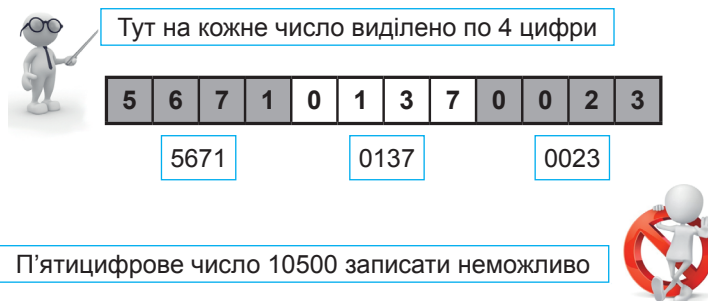


Рис. 3.2. Повідомлення, що містить три числа

3.2.

ЗНАКОВЕ ТА БЕЗЗНАКОВЕ ПОДАННЯ

З курсу математики ви знаєте, що є числа натуральні, цілі, раціональні. У комп'ютері для чисел різного типу застосовуються різні подання. Для цілих чисел їх два: *беззнакове* і *знакове*.

У беззнаковому поданні всі розряди відведено для цифр числа. Воно використовується, коли відомо, що значення, які записуватимуться в комірки пам'яті, завжди додатні (наприклад, порядкові номери, дати) (рис. 3.3).

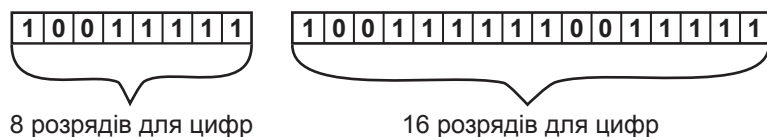


Рис. 3.3. Беззнакове подання цілого числа

Якщо для числа відведено k розрядів, то в найбільшого числа всі вони дорівнюватимуть одиниці, тобто це буде число, що складається з k одиниць. Його десяткове значення дорівнюватиме $2^k - 1$. Найменше число матиме нулі в усіх розрядах і завжди дорівнюватиме нулю. Максимальні десяткові числа для беззнакового подання при різних значеннях k наведено в табл. 3.1.

Таблиця 3.1. Максимальні десяткові числа для різного беззнакового подання

Кількість розрядів	Максимальне число
8	255 ($2^8 - 1$)
16	65535 ($2^{16} - 1$)
32	4294967295 ($2^{32} - 1$)
64	18446744073709551615 ($2^{64} - 1$)

У знаковому поданні старший (лівий) розряд відведено для знака числа (0 — додатне, 1 — від'ємне), а інші — для цифр (рис. 3.4). Воно

використовується для запису числових величин, які можуть набувати додатних і від'ємних значень (температура, результати арифметичних операцій тощо).



Рис. 3.4. Знакове подання цілого числа

Значення границь діапазонів для знакового подання при різних значеннях k наведено в табл. 3.2.

Таблиця 3.2. Максимальні і мінімальні десяткові числа для різного знакового подання

Кількість розрядів	Мінімальне число	Максимальне число
8	– 128	127
16	– 32768	32767
32	– 2147483648	2147483647
64	– 9223372036854775808	9223372036854775807

Ви бачите, що в комп'ютері як у знаковому, так і в беззнаковому поданні множина цілих чисел обмежена і залежить від кількості розрядів, що відводяться для подання числа. Натомість у математиці множина цілих чисел є нескінченною.

Зверніть також увагу на такі дві особливості знакового подання:

- абсолютна величина максимального числа на 1 менша від абсолютної величини мінімального;
- найбільше додатне число, яке можна записати в знаковому поданні, майже у два рази менше від найбільшого числа в беззнаковому поданні.



Контрольні запитання та завдання

1. Назвіть способи подання цілих чисел і поясніть їхню суть.
2. Як відрізняються максимальне число та мінімальне число у знаковому поданні?
3. Як відрізняються максимальні значення чисел у знаковому та беззнаковому поданнях однакової розрядності?

**Питання для роздумів**

1. Якої кількості розрядів достатньо для кодування наступних чисел у двійковій системі числення: 26_{10} , 108_{10} , 512_{10} ?
- 2*. Як ви думаєте, чому для кодування чисел у комп'ютері відводиться саме така кількість розрядів (8, 16, 32, 64)?

**Завдання для досліджень**

- 1*. Дізнайтеся, як у комп'ютері здійснюється кодування дійсних чисел.
- 2*. Чому дорівнює число, записане у такому вигляді: $+1,23E+14$?

§4.**Кодування символів****4.1.****АЗБУКА МОРЗЕ**

Одним із перших цифрових способів передачі інформації став код Морзе. Код Морзе — це нерівномірний телеграфний код, де спосіб кодування літер алфавіту, цифр та інших символів подано певною комбінацією «крапок» і «тире». Так, наприклад, літера «А» позначається, як «.-», а літера «Б» — «-...».



Дізнайтеся історичні відомості про азбуку Морзе та її винахідника

Азбука, з якої формується зазначений код, була названа на честь американського винахідника Семюела Морзе, який придумав її в 1838 році. Телеграф і радіотелеграф спочатку використовували азбуку Морзе, пізніше почали застосовувати код Бодо та ASCII, які є зручнішими для автоматизації.

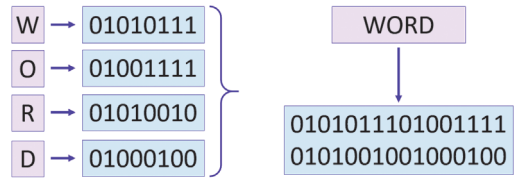
4.2.**КОДУВАННЯ СИМВОЛІВ У КОМП'ЮТЕРНІЙ ТЕХНІЦІ**

Символи тексту людина розрізняє за накресленням. Наприклад, кожен з нас швидко визначить у тексті літеру «а», яким би шрифтом її не було набрано.

Якщо слідувати аналогії зі сприйняттям тексту людиною, то в комп'ютері потрібно зберігати зображення літер, потім з них складати слова і зберігати зображення рядків. Зрозуміло, що цей спосіб

неефективний, оскільки доведеться зберігати безліч зображень, хоча кількість літер, з яких їх складено, обмежена (це літери алфавіту тієї мови, якою подано текст).

Тому для комп'ютерного подання текстової інформації застосовується інший спосіб: символи алфавіту кодуються двійковими числами, а текст подається у вигляді набору чисел — кодів символів, що його складають.



Щоб такий набір чисел можна було декодувати, слід знати, де закінчується і де починається код кожного символу, а для цього, як було показано на прикладі чисел, необхідно перед кодуванням визначити потрібну для кодування кількість розрядів.

Тож яку довжину коду символу взяти, щоб закодувати всі символи, які можуть трапитися в тексті? Якщо це двійковий код із довжиною 2 розряди, то з його допомогою можна буде закодувати алфавіт, що містить 4 символи, оскільки за цієї довжини коду існують чотири різні його комбінації.

Двійковий код із довжиною 3 розряди дозволяє закодувати алфавіт, що містить 8 символів (рис. 4.1).

Код із довжиною 2 розряди		
Код	Літера	Десятькове значення коду
00	А	0
01	Б	1
10	В	2
11	Г	3

Код із довжиною 3 розряди		
Код	Літера	Десятькове значення коду
000	А	0
001	Б	1
010	В	2
011	Г	3
100	Ґ	4
101	Д	5
110	Е	6
111	Є	7

Рис. 4.1. Кодування символів двійковими кодами з різною довжиною

Прослідковується така закономірність:

2 розряди — $2 \cdot 2 = 4 = 2^2$ символів;

3 розряди — $2 \cdot 2 \cdot 2 = 8 = 2^3$ символів;

4 розряди — $2 \cdot 2 \cdot 2 \cdot 2 = 32 = 2^4$ символів;

n розрядів — $2 \cdot 2 \cdot 2 \cdot \dots \cdot 2 \cdot 2 = 2^n$ символів.

З допомогою двійкового коду завдовжки n розрядів можна закодувати алфавіт, що містить 2^n символів.

Таким чином, щоб закодувати текст, слід обрати довжину коду, достатню для кодування потрібної кількості символів, і скласти таблицю, де буде вказано, який код якому символу відповідає. Такі таблиці вже складено, і називають їх *наборами символів*.

Сьогодні найпоширенішими є набори символів ASCII, Unicode («юнікод») і сумісні з ними.

Фрагмент розширеного набору символів ASCII наведено в таблиці нижче. У цьому наборі, як ви можете пересвідчитися, для кодування застосовують 8 розрядів. Розмір алфавіту, який можна закодувати з допомогою такого набору, становить 256 символів (пронумеровані десятковими цифрами від 0 до 255). Цього достатньо, щоб закодувати цифри, знаки пунктуації, латинські літери (великі й малі) та літери кирилиці (великі й малі).

Таблиця 4.1. Фрагменти набору символів ASCII та їхні коди

Символ	Десяткове значення коду	Код	Символ	Десяткове значення коду	Код	Символ	Десяткове значення коду	Код
пробіл	32	00100000	0	48	00110000	A	65	01000001
!	33	00100001	1	49	00110001	B	66	01000010
#	35	00100011	2	50	00110010	C	67	01000011
\$	36	00100100	3	51	00110011	D	68	01000100
*	42	00101010	4	52	00110100	E	69	01000101
+	43	00101011	5	53	00110101	F	70	01000110
,	44	00101100	6	54	00110110	G	71	01000111
-	45	00101101	7	55	00110111	H	72	01001000
.	46	00101110	8	56	00111000	I	73	01001001
/	47	00101111	9	57	00111001	J	74	01001010

A → 65 → 01000001

Так у даній таблиці представлено десяткове та двійкове подання символів. Наприклад, велика латинська літера **A** має десятковий код **65** і двійковий код **01000001**.

У наборі ASCII *перші 32 коди* (з 0 по 31) відведені для операцій (перенесення рядка, скасування попередньої операції, подавання звукового сигналу). Ці коди мають сприйматися пристроєм виведення як команди.

Наступні коди, з 32 по 127, є інтернаціональними і відповідають символам латинського алфавіту, цифрам, знакам арифметичних операцій та знакам пунктуації.

Коди з 128 по 255 — національні, тобто в кодуваннях для різних мов тому самому коду відповідають різні символи.

Недолік такої таблиці кодування — неможливість закодувати текст, що містить фрагменти багатьма мовами. Розрізняють символи, що належать до алфавітів кирилиці. Це алфавіти української, російської, білоруської мов тощо. А також символи латиниці, що є символами алфавітів англійської та деяких західноєвропейських мов. Значно більшими є різноманіття ієрогліфів.

Щоб усунути це обмеження, в 1991 році було запропоновано стандарт кодування Unicode, який дозволяє використовувати в текстах будь-які символи будь-яких мов світу.

З використанням Unicode закодовано всі алфавіти відомих мов, зокрема і «мертвих» (єгипетські ієрогліфи, писемність мая, етрусський алфавіт). Для мов, що мають кілька алфавітів або варіантів написання (наприклад, японська, індійська), закодовано всі варіанти.

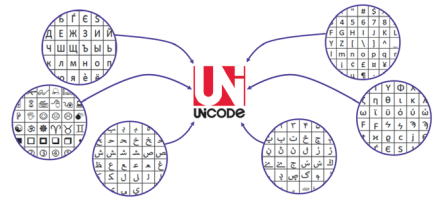
В Unicode внесено всі математичні, хімічні, музичні та інші наукові символні позначення. Потенційна ємність Unicode така велика, що наразі використовується лише незначний відсоток доступних кодів символів.

Перші 128 символів у стандарті Unicode збігаються з таблицею ASCII.

Далі розміщено основні алфавіти сучасних мов. Вони містяться в першій частині таблиці, значення їхніх кодів не перевищують $2^{16} = 65536$.

Для українських і російських літер є кілька кодових таблиць, що різняться розташуванням символів. Найчастіше вам траплятимуться такі: KOI8-U, KOI8-R, Windows-1251, Code Page 866 (CP 866), ISO 8859. Тексти, створені з використанням однієї таблиці, неправильно відображаються у разі використання іншої. Саме це є причиною того, що інколи в браузері замість тексту можна побачити беззмістовний набір знаків.

З удосконаленням вебтехнологій вдосконалюється і робота браузерів, і останні версії браузера Google Chrome самостійно пристосовуються під кодування тексту сторінки, яка завантажувється.



Дізнайтеся більше про кодування символів в MS Word

ВПРАВА 4.1



Завдання. Виконайте завдання у файлі **Вправа_4_1.docx**, використовуючи таблицю кодів азбуки Морзе.

ВПРАВА 4.2



Завдання. Виконайте завдання у файлі **Вправа_4_2.docx**, використовуючи таблицю ASCII кодів.



Контрольні запитання та завдання

1. Який підхід до кодування запропонував свого часу Семюел Морзе?
2. Який підхід застосовано до кодування текстової інформації в комп'ютерних технологіях?
3. Яку назву має стандартний набір, що складається з 256 символів?
4. Яку назву має стандарт кодування, що дозволяє здійснити кодування всіх алфавітів відомих мов?



Питання для роздумів

- 1*. Яку мінімальну довжину коду матиме повідомлення, подане за допомогою алфавіту з 65 символів?
- 2**. Чому азбука Морзе є нерівномірним кодом?



Завдання для досліджень

- 1*. Підготуйте повідомлення про використання на практиці різних способів кодування текстової інформації.
- 2*. Дізнайтеся, що таке стенографія і яке її призначення.



Пройдіть
тест

§5.

Вимірювання інформації

5.1.

ЯК ВИМІРЯТИ ІНФОРМАЦІЮ

Подібно до того, як вимірюють вагу, довжину або час, можна вимірювати й інформацію. Про те, як це зробити і які одиниці вимірювання при цьому використовують, дізнаємося далі.

Якщо не видаляти зайві знімки та відео зі свого смартфона, його пам'ять рано чи пізно вичерпається. А якщо, користуючись «повільним» інтернет-підключенням, переглядати онлайн-відео, воно буде «гальмувати». Щоб уникнути таких ситуацій, слід враховувати, який обсяг інформації бере участь в інформаційному процесі.

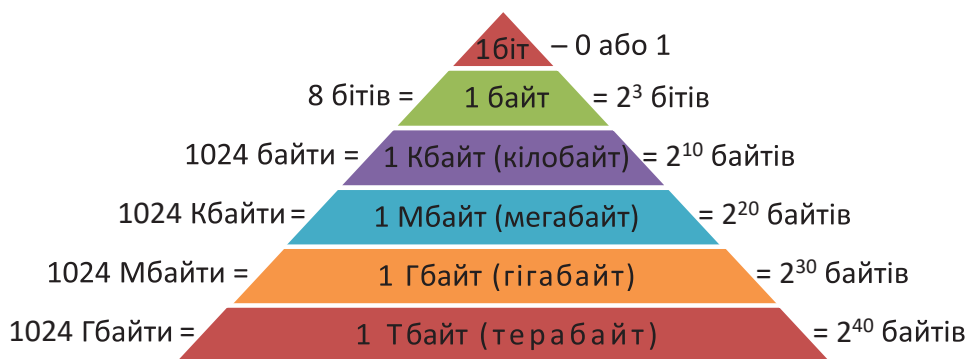
Під інформацією як правило розуміють щось корисне і зрозуміле користувачеві. Виходить, що текстовий файл, який містить 100 літер «А» або пробілів і займає на диску певний простір, не несе інформації. Отже, до вимірювання інформації мають існувати принаймні два підходи — такий, що враховує *зміст повідомлення*, і такий, що дає змогу оцінити *обсяг його даних*.

Окрім цього, потрібно ввести еталонну одиницю, подібно до того, як для вимірювання маси введено еталонну одиницю кілограм, для вимірювання довжини — метр.

Найменшою одиницею вимірювання інформації є 1 біт, який, як нам уже відомо, може приймати значення 0 або 1.

Наступною за величиною одиницею є 1 байт — він дорівнює 8 бітів, або 2^3 бітів.

На схемі нижче також представлені й інші одиниці вимірювання:



5.2.

ЗМІСТОВИЙ ПІДХІД

Основою змістового підходу до вимірювання інформації є зміст інформаційного повідомлення. Згідно з цим підходом, кількість інформації розглядається як ступінь зменшення невизначеності знань у результаті отримання повідомлення.

Розглянемо процес підкидання монети. Вона може впасти на один із двох боків («аверс» чи «реверс»), отже, перед її підкиданням невизначеність дорівнює 2. Невизначеність перед підкиданням грального кубика дорівнює 6 (у кубика шість граней), невизначеність того, скільки балів можна отримати за відповідь на уроці, становить 12 (за дванадцятибальної системи оцінювання).

Невизначеність знання про результат певної події дорівнює кількості можливих результатів цієї події.

Закодуємо повідомлення про можливі результати деяких із описаних вище подій з допомогою двійкового коду. Згадаємо методи кодування символів, а саме, той факт, що двійковий код із довжиною n дає можливість закодувати 2^n варіантів. Таким чином, для кодування варіантів падіння монети достатньо одного двійкового розряду. Тобто для того, щоб дізнатися, яким боком упала монета, достатньо один раз задати запитання з приводу цього. І отримати одну відповідь: «так» чи «ні».

Згідно із змістовим підходом таке повідомлення містить 1 біт інформації.

Для кодування варіантів випадання значень у разі підкидання кубика двох розрядів не вистачить, їх буде 3. Чому? Найближче до 6 число, що є степенем числа 2, — це 8, а $8 = 2^3$. Отже, показник степеня 3 і визначає кількість бітів інформації, яку одержує людина після підкидання кубика.



Код	Повідомлення
0	Аверс
1	Реверс

Підкидання грального кубика

невизначеність дорівнює 6

Код	Варіанти випадіння кубика
000	1
001	2
010	3
011	4
100	5
101	6

$$6 < 8 = 2^3$$

$$N = 3$$



Для кодування оцінки, яку одержує учень на уроці, знадобляться 4 розряди. 12 — число, менше від 16. А 16 є найближчим до 12 числом, що є степенем числа 2, тобто $16 = 2^4$. Отже, показник степеня 4 і визначає кількість бітів інформації, яку одержує учень після оголошення оцінки.

Одержання оцінки на уроці

невизначеність дорівнює 12

Код	Повідомлення	Код	Повідомлення
0000	1	0111	7
0001	2	1000	8
0010	3	1001	9
0011	4	1010	10
0100	5	1011	11
0101	6	1100	12

$$12 < 16 = 2^4$$

$$N = 4$$



Розглянемо ряд практичних задач, у яких застосовано змістовий підхід до вимірювання інформації.

Задача 1. У коробці лежать 24 олівці різного кольору. Скільки інформації несе повідомлення про те, що з коробки дістали синій олівець?

Розв'язання

Перед тим, як витягти олівець, невизначеність становила 24, оскільки усі олівці різного кольору.

Найближчим більшим за число 24 числом, що є степенем числа 2, є число 32.

$$24 < 32 = 2^5.$$

Показник степеня дорівнює 5, отже, повідомлення несе 5 бітів інформації.

Відповідь: 5 бітів.

Задача 2. Визначити, скільки бітів інформації несе повідомлення «Концерт відбудеться 25 жовтня».

Розв'язання

Оскільки в повідомленні вказано дату одного із днів року, то невизначеність перед повідомленням становила 365. Найближчим більшим числом, що є степенем числа 2, є число 512.

$$365 < 512 = 2^9.$$

Показник степеня дорівнює 9, отже, повідомлення несе 9 бітів інформації.

Відповідь: 9 бітів.



5.3.

АЛФАВІТНИЙ ПІДХІД

Ще один підхід до вимірювання інформації — *алфавітний*. З допомогою алфавітного підходу можна виміряти кількість інформації в тексті, поданому будь-якою мовою. Водночас не враховуються такі її властивості, як корисність, актуальність або повнота. Кількість інформації не пов'язана зі змістом тексту, а залежить від *інформаційної ваги символу в алфавіті*.

Обсяг інформації (V) визначається як кількість (k) символів у повідомленні, помножена на інформаційну вагу символу (b). Тобто $V = k \cdot b$.

Алфавіт — це весь набір букв, символів пунктуації та інших символів і знаків (включаючи пробіли), що застосовуються в тексті. Загальна кількість символів алфавіту визначає його *потужність*.

Інформаційна вага символу та потужність алфавіту пов'язані між собою такою формулою: $N = 2^b$, де N — потужність алфавіту, b — інформаційна вага символу.

Так, наприклад, потужність алфавіту української мови — 33 символи. Відповідно, потужність алфавіту десяткової системи числення — 10 символів, що представлені цифрами від 0 до 9.

Аа	Бб	Вв	Гг	Ґґ	Дд
Ее	Єє	Жж	Зз	Ии	Іі
Її	Йй	Кк	Лл	Мм	Нн
Оо	Пп	Рр	Сс	Тт	Уу
Фф	Хх	Цц	Чч	Шш	Щщ
Ьь	Юю	Яя			

0	1	2	3	4
5	6	7	8	9

N = 10

N = 33

Найменшу потужність (рівну 1) має алфавіт із двома символами ($2 = 2^1$).

Саме такий алфавіт, *двійковий*, застосовується в комп'ютері.

Інформаційна вага символу двійкового алфавіту прийнята за одиницю інформації, і називається ця одиниця також *бітом*. Як і в разі змістового підходу, від неї утворюються більші одиниці (байт, кілобайт тощо).

Якщо алфавіт містить 256 символів, наприклад, таблиця кодів ASCII, то b — інформаційна вага одного його символу становитиме 8 бітів ($2^8 = 256$). Згадайте принципи кодування тексту: саме за такої довжини двійкового коду можна закодувати всі символи зазначеного алфавіту.

Розглянемо задачі на вимірювання інформації з використанням алфавітного підходу.

Задача 3. Текст, що має довжину 104 символи, записаний з використанням алфавіту потужністю 64 символи. Який інформаційний обсяг тексту в байтах?

Розв'язання

$$N = 64; N = 2^b, 64 = 2^6, \text{ отже, } b = 6 \text{ (бітів).}$$

$$V = k \cdot b, 6 \cdot 104 = 624 \text{ (біти)} = 624 : 8 = 78 \text{ (байтів).}$$

Відповідь: 78 байтів.

Задача 4. Інформаційний обсяг набраного на комп'ютері тексту становить 1 Кбайт. Скільки символів містить цей текст, якщо інформаційна вага символу становить 4 біти?

Розв'язання

$$V = k \cdot b, k = V/b;$$

$$V = 1 \text{ Кбайт} = 1024 \text{ (байти);}$$

$$b = 4 \text{ біти} = 0,5 \text{ (байта).}$$

$$k = 1024 : 0,5 = 2048.$$

Відповідь: 2048 символів.

Задача 5. Привітання учасникам олімпіади від марсіан записано за допомогою всіх символів марсіанського алфавіту: ТЕВИРП!КИ! Визначити вагу повідомлення в бітах.

Розв'язання

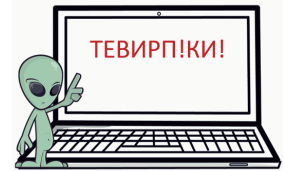
Спочатку підрахуємо, скільки символів містить марсіанський алфавіт. Як бачимо, у вислові є 8 різних символів: ТЕВИРП!К.

$$8 = 2^3.$$

Отже, вага символу становить 3 біти.

Усього в повідомленні 10 символів, відповідно вага усього повідомлення $3 \cdot 10 = 30$ бітів.

Відповідь: 30 бітів.



Виконайте практичні завдання



Контрольні запитання та завдання

1. Які підходи використовують для вимірювання інформації?
2. Які одиниці вимірювання інформації ви знаєте?
- 3*. Поясніть принцип змістового підходу вимірювання інформації.
4. Що таке інформаційна вага символу в алфавіті?
5. Що таке алфавіт?
6. Як вимірюється кількість інформації текстового повідомлення?



Питання для роздумів

- 1*. Марійка задумала число від 1 до 100. Яку найменшу кількість спроб має зробити Андрій, щоб відгадати число, яке задумала Марійка?
- 2*. Повідомлення про те, що Івасик живе на 12 поверсі, несе 4 біти інформації. Скільки поверхів у будинку?
3. Текст, що має довжину 100 символів, записаний із використанням алфавіту потужністю 64 символи. Який інформаційний обсяг тексту в байтах?
4. Інформаційний обсяг набраного на комп'ютері тексту становить 3,5 Кбайти. Скільки символів містить цей текст? Врахуйте, що інформаційна вага символу становить 8 бітів.



Завдання для досліджень

- 1*. Дізнайтеся про інші способи вимірювання інформації. Поясніть їхній принцип.



Пройдіть
тест

§6.

Кодування графічних даних

6.1.

РАСТРОВЕ ПОДАННЯ

Двійковому кодуванню піддається і графічна інформація. Розглянемо, як це відбувається.

Щоб зберегти у двійковій формі графічну інформацію, застосовують два подання — *растрове* і *векторне*. Як ми бачимо з рисунка 6.1, при масштабуванні растрового зображення його якість погіршується, а при масштабуванні векторного зображення якість не змінюється.



Рис. 6.1. Порівняння растрового та векторного подань

Растрове подання застосовується для зберігання фотографій, творів живопису, малюнків та подібних зображень.

Для створення растрового подання зображення розбивається на маленькі елементи. Така процедура називається *растеризацією*, звідси й назва подання.

Елементи, на які розбивається зображення, називаються *пікселями*. Отже, *піксель* — це найменша одиниця цифрового зображення у растровій

графіці. Він являє собою неподільний об'єкт прямокутної (зазвичай квадратної) форми, що має певний колір.

Растрове зображення — це зображення, що кодується як множина пікселів.



Растрезацію застосовували до появи комп'ютерів. Зокрема, для відтворення фотографій у книжках та газетах негатив знімка перефотографували через растр (пластину з нанесеними лініями). У результаті отримували зображення, утворене з точок. З нього робили форму для друкарської машини. У поліграфії застосовуються різні види растрів. На рис. 6.2 представлено збільшену копію растрованого зображення ока, що використовувалося для друку в газеті, та зображення комахи, розбите на пікселі з використанням комп'ютера.

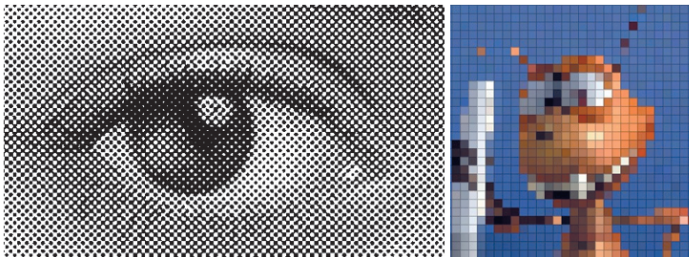


Рис. 6.2. Зображення ока та комахи

У комп'ютерній техніці застосовують лише прямокутні пікселі та прямокутний растр, у якого елементи розташовано у вигляді прямокутної матриці (рядками та стовпцями).

Колір пікселя для зберігання у комп'ютері кодується з допомогою двійкового коду з певною довжиною. Як і в разі кодування тексту, чим більшу довжину (кількість розрядів) має такий код, тим більше кольорів пікселя можна закодувати з його допомогою.

У чорно-білому зображенні, що не містить відтінків сірого, використовуються лише два кольори, а отже, для кодування кольору його пікселів достатньо одного розряду (0 — білий колір, 1 — чорний колір).

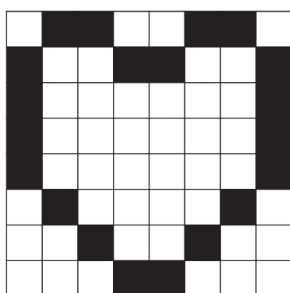
Код	Колір
0	білий
1	чорний

1 піксель = 1 біт



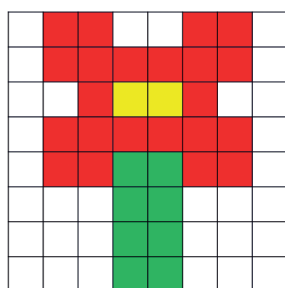
Двійкову послідовність, що міститиме дані зображення, можна побудувати, замінивши білі пікселі нулями, а чорні — одиницями, як показано на наступному прикладі. Ширина рисунка тут становить 8 пікселів, а отже, кожний рядок таблиці буде складатися із 8 бітів, або одного

байта. Все зображення матиме 8 таких рядків і для його зберігання знадобиться щонайменше 8 байтів.



0	1	1	0	0	1	1	0
1	0	0	1	1	0	0	1
1	0	0	0	0	0	0	1
1	0	0	0	0	0	0	1
1	0	0	0	0	0	0	1
0	1	0	0	0	0	1	0
0	0	1	0	0	1	0	0
0	0	0	1	1	0	0	0

А що ж робити, коли рисунок кольоровий? У цьому випадку для кодування кольору пікселя одним бітом уже не обійтися. Наприклад, для кодування одного з чотирьох варіантів кольорів потрібно вже застосувати 2 розряди двійкового коду. Отже, для зберігання даних про один піксель треба виділити 2 біти. Позначимо білий колір кодом 00, червоний — 01, зелений — 10, а жовтий — 11. Отримаємо таку таблицю.



00	01	01	00	00	01	01	00
00	01	01	01	01	01	01	00
00	00	01	11	11	01	00	00
00	01	01	01	01	01	01	00
00	01	01	10	10	01	01	00
00	00	00	10	10	00	00	00
00	00	00	10	10	00	00	00
00	00	00	10	10	00	00	00

□	-	00	■	-	01
■	-	10	■	-	11

У кольорових зображеннях для кодування кольору пікселя зазвичай застосовують 4, 8 або 24 розряди, що відповідають 16, 256 і 16 777 216 кольорам. Від цієї кількості розрядів залежить якість зображення.

Окрім кількості розрядів для зберігання кольору потрібно задати, який колір якому числу відповідатиме, тобто, як саме дані про колір будуть записані у цифровому вигляді. Це визначається *колірною моделлю*.

Колірна модель — система кодування кольорів, яка використовується для зберігання та відтворення зображення.

Найпоширеніших колірних моделей три — RGB, CMYK та HSB. Назва моделі RGB утворена першими літерами кольорів Red (червоний), Green (зелений), Blue (синій). Колір у цій моделі розглядається як результат змішування променів світла трьох кольорів. Кожна складова кодується цілим числом від 0 до 255, а отже, код кольору — це трійка чисел (R, G, B). Чорний колір має код (0, 0, 0) — тут відсутні будь-які світлові промені. Білий колір має код (255, 255, 255). Відповідно у червоного кольору буде присутня складова червоного кольору, а інші відсутні, аналогічно утворено коди зеленого та синього кольорів.

На рисунку нижче також наведено коди кольорів цієї палітри, утворені при попарному змішуванні червоного, зеленого та синього променів.

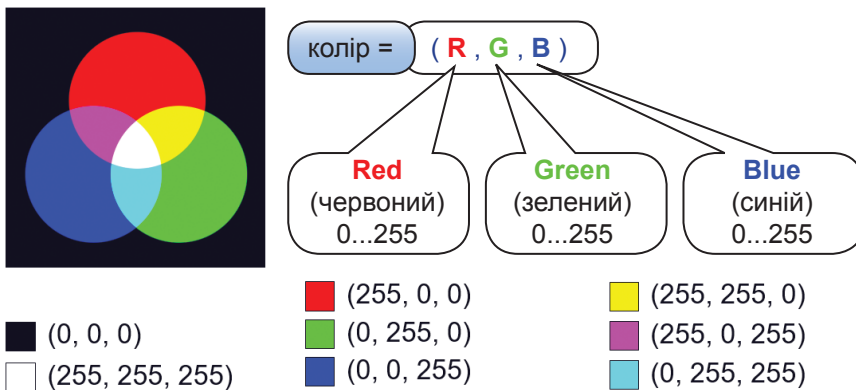


Рис. 6.3. RGB-палітра та коди основних її кольорів

Неважко здогадатися, що загальна кількість кольорів палітри RGB дорівнює $256 \cdot 256 \cdot 256 = 16777216$, що є більш ніж достатнім для сприйняття людиною. Оскільки $256 = 2^8$, кожна зі складових кольору містить 8 бітів інформації або 1 байт. Отже, для кодування певного кольору потрібно 24 біти або 3 байти. Це значення називають *глибиною кольору*.

Глибина кольору — це кількість бітів, що використовується для кодування кольору одного пікселя зображення.

24-бітне кодування називають True Color, що перекладається, як реальний колір, тобто колір, наближений до кольорів реальної дійсності.

Задача 1. Зображення, закодоване в режимі True Color, має розміри (100×150) пікселів. Визначити обсяг його даних.



150 пікселів

100 пікселів

Розв'язання

Для обчислення обсягу даних зображення необхідно скористатися формулою:

$$V = k \cdot b,$$

де k — кількість пікселів, b — глибина кольору.

Визначимо кількість пікселів зображення:

$$100 \cdot 150 = 15000 \text{ (пікселів).}$$

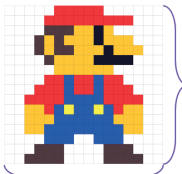
Оскільки зображення закодовано в режимі True Color, вага одного пікселя = 3 (байти).

Відповідно обсяг даних зображення дорівнює:

$$15000 \cdot 3 = 45000 \text{ (байтів).}$$

Відповідь: 45000 байтів.

Задача 2. На рисунку представлено зображення. Необхідно визначити обсяг його даних у байтах, якщо в ньому застосовуються тільки кольори, присутні на зображенні.



16 пікселів

16 пікселів

Розв'язання

Визначимо загальну кількість пікселів:

$$16 \cdot 16 = 256 \text{ пікселів.}$$

Визначимо глибину кольору — для цього підрахуємо кількість кольорів у зображенні. Це значення дорівнює 7.

$7 < 8 = 2^3$, отже, для кодування одного кольору необхідно 3 біти — це значення і є глибиною кольору.

Знаходимо обсяг даних зображення:

$$V = k \cdot b, 256 \cdot 3 = 768 \text{ бітів} = 768 : 8 = 96 \text{ байтів.}$$

Відповідь: 96 байтів.

Звичайно, в результаті дискретизації погіршується якість зображення. Для зменшення втрат якості зображення при дискретизації слід використовувати якомога менший розмір пікселя, а саме: збільшувати *роздільну здатність* зображення.

Роздільна здатність — це кількість пікселів на одиницю довжини.

Як правило, роздільна здатність вимірюється в пікселях на дюйм. 1 дюйм = 2,54 см.

Для роздільної здатності використовується англійське позначення *ppi* (pixel per inch, що перекладається як «піксель на дюйм»). Так, роздільна здатність 254 *ppi* означає, що в одному дюймі буде розташовано 254 пікселі. На рисунку представлені зображення з різними значеннями роздільної здатності. Звичайно, чим більша роздільна здатність, тим кращою є якість зображення, але й обсяг файлу, що містить це зображення, буде більшим.



300 ppi



96 ppi



48 ppi



24 ppi

Задача 3. Закодоване зображення має розміри (800×600) пікселів і роздільну здатність — 72 ppi. Який розмір буде в зображення, надрукованого на принтері?

Розв’язання

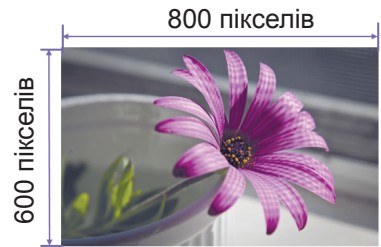
Для визначення ширини зображення необхідно його значення в пікселях помножити на довжину одного дюйма в сантиметрах та розділити на роздільну здатність зображення:

$$800 \cdot 2,54 : 72 = 28,2 \text{ (см)}$$

Аналогічно знаходимо висоту зображення:

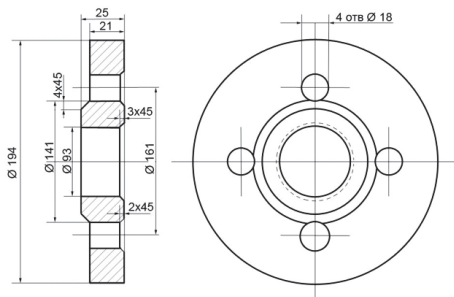
$$600 \cdot 2,54 : 72 = 21,2 \text{ (см)}$$

Відповідь: 28,2 см × 21,2 см.



6.2. ВЕКТОРНЕ ПОДАННЯ

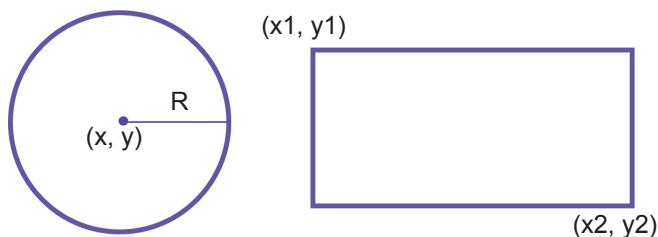
Крім растрового подання, використовують *векторне подання*. Його застосовують для зберігання контурних малюнків, креслень та інших структурованих рисунків, що складаються з контурних об’єктів, які можна описати математично (кіл, дуг, прямокутників, ліній тощо).



Векторне зображення — це зображення, побудоване з елементарних геометричних фігур, параметри яких (розміри, координати вершин, кути нахилу, колір контуру тощо) зберігаються у вигляді чисел.

В комп’ютері зберігають інформацію про тип об’єкта і дані, необхідні для його побудови. Для кола — це радіус або діаметр та координати.

нати центра, для прямокутника — координати протилежних вершин. Неважко здогадатися, що растеризація при цьому не застосовується.



Відповідно для кожної фігури зберігаються:

- розміри та координати зображення;
- колір та стиль контуру;
- колір та стиль заливки для замкнених фігур.

Отже, векторне подання має ряд переваг порівняно з растровим поданням:

- при кодуванні не втрачається інформація;
- при зміні розміру не відбувається спотворення зображення;
- файл має невеликий обсяг, адже він залежить не від розмірів зображення, а від кількості елементів у ньому;
- це подання є зручним для зберігання креслень, графіків, схем, карт тощо.

Звичайно, недоліком є те, що це подання абсолютно не придатне для роботи з фотографіями.

6.3.

ГРАФІЧНІ ФОРМАТИ

Для зберігання графічної інформації використовують як векторні, так і растрові формати. Розглянемо найпоширеніші формати графічних файлів.



BMP

Bitmap застосовують для зберігання растрових зображень без стиснення, з кодуванням інформації про кожен піксель.



GIF

Graphics Interchange Format призначений для стиснення растрових зображень, у яких міститься багато однорідних заливок (для логотипів, написів, схем). Основне обмеження цього формату полягає в тому, що кольорове зображення може бути записане тільки в режимі 256 кольорів.



JPG

Joint Photographic Experts Group формат розроблений для фотографій реалістичного кольору з використанням великого стиснення. Він дозволяє гнучко варіювати співвідношення рівня стиснення та якості зображення.



Portable Network Graphics використовують для зображень, які розміщують в Інтернеті. Формат задовольняє основну вимогу Вебу — забезпечення однакового вигляду зображення незалежно від використовуваного браузера та монітора. Колір зображення в цьому форматі відтворюватиметься однаково на будь-якому комп'ютері.



Tagged Image File Format на сьогодні є одним із найпоширеніших і найнадійніших растрових форматів. Його підтримують майже всі програми, пов'язані з графікою. Це найкращий вибір для зберігання сканованих малюнків, а також для імпортування растрової графіки у векторні редактори.



Scalable Vector Scalable відображає векторні зображення, призначені для вебсторінок в Інтернеті.

ВПРАВА 6.1



Завдання. Виконайте завдання у файлі **Вправа_6_1.docx**. Визначте кольори за їхніми RGB-кодами.

ВПРАВА 6.2



Завдання. Виконайте завдання у файлі **Вправа_6_2.docx**. Створіть зображення за їхніми двійковими кодами.



Контрольні запитання та завдання

1. Які подання використовують для кодування графіки?
2. Що таке піксель?
3. Які складові має колірна модель RGB?
- 4*. Що таке True Color?
5. Що таке глибина кольору? Як змінюється якість зображення при збільшенні значення глибини кольору?
6. Що таке роздільна здатність? Як змінюється якість зображення при збільшенні значення роздільної здатності?
7. Які графічні формати ви знаєте? Поясніть їхнє призначення.



Питання для роздумів

- 1*. Що необхідно знати, щоб знайти ширину та висоту зображення в сантиметрах, якщо відомі його ширина та висота у пікселях?
- 2*. Як ви думаєте, чому в моделі RGB яскравість кольору кодується в діапазоні від 0 до 255?



Завдання для досліджень

- 1*. Знайдіть інформацію про історію походження та використання такої одиниці вимірювання, як дюйм.
- 2*. Дізнайтеся, що таке колірна модель СМЮК.



Пройдіть
тест

§7.

Апаратне забезпечення

7.1.

БУДОВА КОМП'ЮТЕРА

Комп'ютер — це програмований електронний пристрій, який приймає дані, обробляє їх, відображає результати у вигляді інформаційних повідомлень і за потреби зберігає дані для подальшого їхнього використання.

У цьому визначенні згадуються чотири *інформаційних процеси* — введення, обробка, виведення і зберігання. Розглянемо реалізацію цих процесів у комп'ютері детальніше.

Обробляє дані та виконує програми головний компонент комп'ютера — *центральний процесор*. Він також керує всіма блоками комп'ютера. Зберігання програм і даних відбувається в *пам'яті*. Щоб користувачі могли вводити до комп'ютера нову інформацію та керувати ним, застосовуються *пристрої введення* (клавіатура, миша, сканер, мікрофон тощо).

Щоб користувачі отримували інформацію про результати роботи комп'ютера, його оснащують пристроями *виведення* (монітором, звуковими колонками, принтером тощо).

Для обміну даними через мережу комп'ютери оснащують комунікаційними пристроями (модемом, мережевою платою). Залежно від того, що робить комп'ютер — передає дані в мережу чи приймає їх, ці пристрої виконують процеси введення або виведення.

Зв'язок між пристроями підтримується за допомогою спеціальної інформаційної магістралі — *шини*. Класифікацію пристроїв комп'ютера за їхнім призначенням наведено на рис. 7.1.

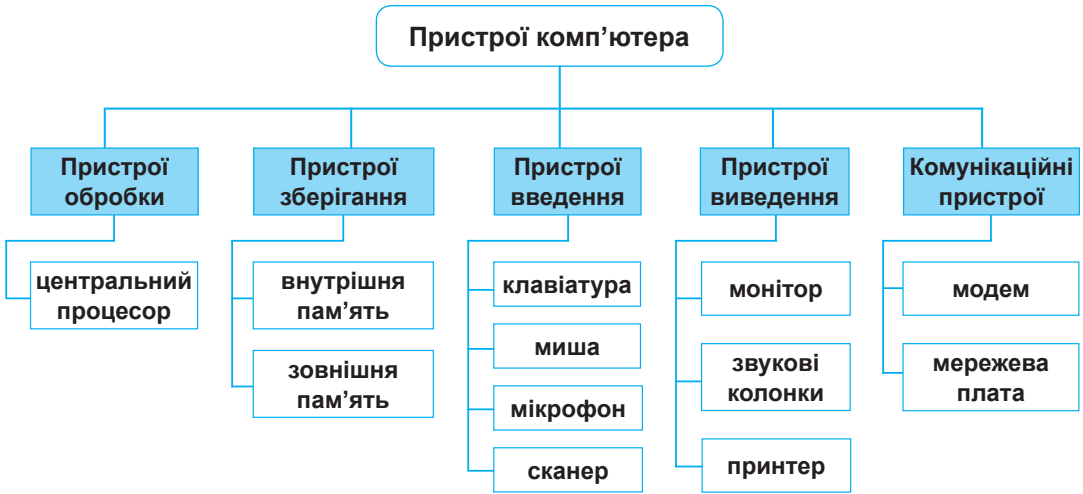


Рис. 7.1. Класифікація пристроїв комп'ютера за їхнім призначенням

7.2.

ЦЕНТРАЛЬНИЙ ПРОЦЕСОР

Центральний процесор — це основний компонент комп'ютера, призначений для керування всіма його пристроями та виконання арифметичних і логічних операцій над даними.

Це компактний (розміром близько $5\text{ см} \times 5\text{ см} \times 0,3\text{ см}$) модуль, що вставляється в гніздо на материнській платі (рис. 7.2).

Сьогодні серед виробників процесорів лідирують дві компанії — Intel Corporation та Advanced Micro Devices (AMD).

Можливості комп'ютера великою мірою залежать від характеристик процесора, насамперед від його *продуктивності*, яку зазвичай оцінюють за тактовою частотою і за кількістю ядер — головних обчислювальних модулів.

Тактова частота визначає, скільки елементарних операцій (тактів) процесор виконує за секунду, і вимірюється в мегагерцах ($1\text{ МГц} = 10^6\text{ Гц}$) і гігагерцах ($1\text{ ГГц} = 10^9\text{ Гц}$).

Багатоядерний процесор може мати два чи більше обчислювальних ядер на одному процесорному кристалі або в одному корпусі. Сучасні багатоядерні процесори мають 2, 4, 8 і більше обчислювальних ядер з тактовою частотою кожного від 1,66 ГГц до 4,7 ГГц.



Рис. 7.2. Мікросхеми центрального процесора

7.3. МАТЕРИНСЬКА ПЛАТА

Материнська плата — пристрій, на якому встановлено внутрішні компоненти комп'ютера. Використовується для передавання даних між усіма його пристроями.

Це складна багатошарова друкована плата, до якої підключено практично всі пристрої комп'ютера. Друкована плата є пластиною, що вкрита сіткою провідників-доріжок, по яких електричні сигнали надходять до змонтованих на платі мікросхем і роз'ємів, куди вставляють інші пристрої комп'ютера. Місця підключення інших пристроїв показано на рис. 7.3.

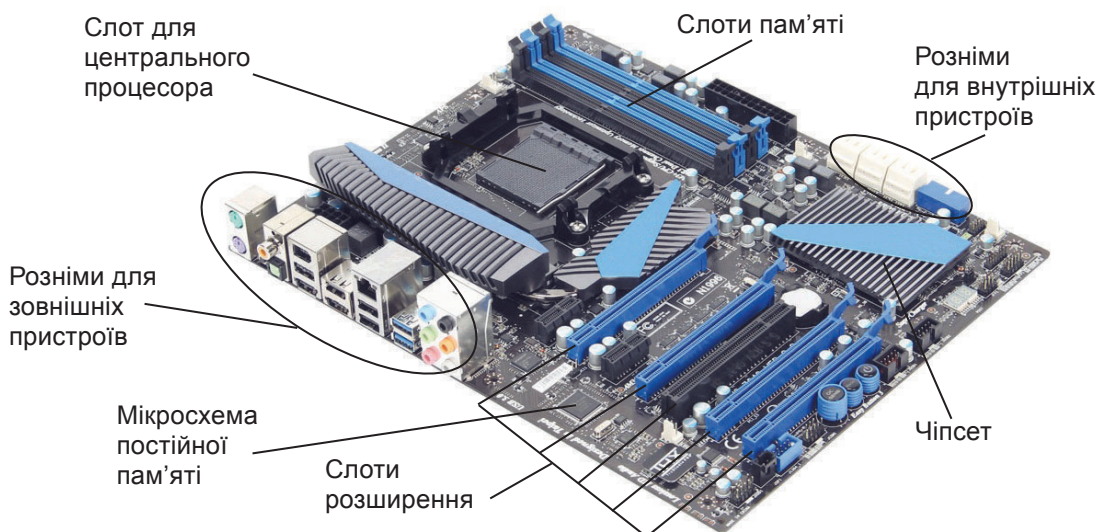


Рис. 7.3. Материнська плата

7.4. ВНУТРІШНЯ ПАМ'ЯТЬ

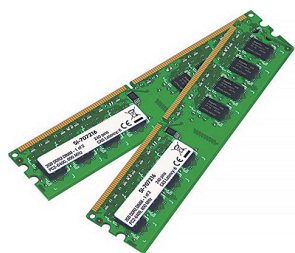


Рис. 7.4. Модулі оперативної пам'яті

Пам'ять комп'ютера традиційно поділяють на *внутрішню* і *зовнішню*. **Внутрішня пам'ять** складається з *оперативної* та *постійної пам'яті*. Основна характеристика внутрішньої пам'яті — обсяг, який вимірюють у мегабайтах і гігабайтах.

Оперативна пам'ять, або ОЗП (оперативний запам'ятовуючий пристрій), є основною частиною внутрішньої пам'яті (рис. 7.4), де зберігаються дані та програми для виконуваних у поточний момент завдань. У разі вимкнення електроживлення вміст оперативної пам'яті втрачається.

Оперативна пам'ять — швидкодіюча пам'ять, призначена для записування, зберігання і зчитування інформації у процесі її обробки.

Якби комп'ютер мав лише оперативну пам'ять, вміст якої після вимкнення живлення втрачається, то щоразу, ввімкнувши комп'ютер, доводилося б вводити програми і дані заново. Те саме можна сказати і про налаштування телевізора і мобільного телефону.

Цієї проблеми вдається уникнути завдяки наявності в комп'ютері постійної пам'яті, або ПЗП (постійного запам'ятовуючого пристрою). У неї записують інформацію, що не змінюється під час роботи комп'ютера. Постійна пам'ять енергонезалежна, тобто здатна зберігати інформацію і після вимкнення живлення. Виготовляють її у вигляді мікросхем (рис. 7.5).



Рис. 7.5. Мікросхема постійної пам'яті

Постійна пам'ять — швидкодіюча енергонезалежна пам'ять, призначена для зберігання інформації, що не змінюється під час виконання програм. Ця пам'ять забезпечує лише можливість зчитування інформації.

Проте є такі мікросхеми постійної пам'яті, які користувач може багаторазово програмувати, а інформація з них стирається в електричний спосіб. До пам'яті такого типу належить, наприклад, *флешпам'ять*.

7.5. ЗОВНІШНЯ ПАМ'ЯТЬ

Зовнішня пам'ять — це пам'ять, призначена для тривалого зберігання даних, цілісність яких не залежить від того, увімкнено чи вимкнено комп'ютер. Пам'ять цього типу виготовляють у вигляді різноманітних пристроїв для зберігання цифрових даних.

Пристрій зберігання даних складається з носія, на якому записано дані, та допоміжного обладнання, що забезпечує можливість їх записувати, зчитувати і передавати. Носій даних може бути знімним або стаціонарним. Інформація, що міститься на пристрої зберігання даних, після вимкнення живлення не втрачається.

До основних характеристик пристроїв зовнішньої пам'яті належать:

- ємність;
- тип інтерфейсу, тобто спосіб підключення;
- швидкість передавання даних;
- час доступу до даних.



Дізнайтеся більше про пристрої зовнішньої пам'яті

Нині найпоширенішими пристроями зберігання та носіями даних є *дисківі накопичувачі, флешпам'ять*.

Основними пристроями зберігання даних в інформаційних системах є **жорсткі диски**. Розрізняють два види жорстких дисків — магнітні дисківі накопичувачі (HDD) та твердотілі накопичувачі (SSD).

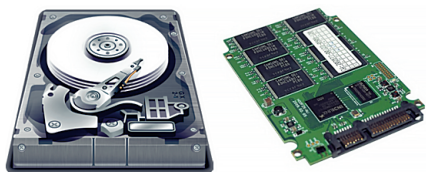


Рис. 7.6. Жорсткі диски HDD та SSD



Рис. 7.7. Флешнакопичувачі



Рис. 7.8. Оптичні носії

Зазвичай жорсткі диски розміщені в системному блоці комп'ютера, проте можуть бути виконані і як зовнішні пристрої. У такому разі диск підключають до комп'ютера через порт USB. Ємність жорстких дисків, що використовуються в комп'ютерах масового виробництва, сягає від 160 Гб до 8 Тб.

Вище вже згадувалося про флешпам'ять. Цю технологію використовують для виготовлення зовнішніх пристроїв зберігання (флешок), які підключають до комп'ютера через порт USB. Ємність флешок варіюється в діапазоні від 4 Гбайтів до 2 Тбайтів, і це ще не межа (рис. 7.7).

Ще й досі використовуються, але вже не мають масового застосування оптичні носії — диски DVD, CD та BD (Blu-ray Disk) (рис. 7.8). Записування даних на ці носії та зчитування з них здійснюють, застосовуючи лазер. Для цього використовують відповідні дисководи та плеєри.

ВПРАВА 7



Завдання. У файлі **Вправа_7.pptx** виконайте завдання, вказані у примітках кожного слайду.



Контрольні запитання та завдання

1. Для чого призначено центральний процесор?
2. Які функції виконує оперативна пам'ять?
3. Яка частина внутрішньої пам'яті є енергозалежною?
4. Які пристрої зберігання даних ви знаєте?



Питання для роздумів

- 1*. Який процесор буде працювати продуктивніше: одноядерний з частотою 3,2 ГГц чи двоядерний з частотою по 1,6 ГГц?

**Завдання для досліджень**

- 1*. Визначте ємність жорсткого диска, встановленого на вашому комп'ютері в класі та вдома.
- 2*. Відшукайте інформацію про еволюцію носіїв інформації.
- 3*. Дізнайтеся, які відмінності у дисках HDD та SSD.

Пройдіть
тест**§8.****Пристрої введення**

Розглянемо принципи роботи та різновиди пристроїв, які дозволяють уводити інформацію.

З допомогою пристроїв *уведення* користувачі вводять до комп'ютера нову інформацію та подають йому команди. Найпоширеніші з цих пристроїв — клавіатура, миша, сканер та мікрофон.

8.1.**КЛАВІАТУРА**

Основний пристрій, з допомогою якого користувач уводить до комп'ютера текстову інформацію, — це, звичайно, **клавіатура**. Крім того, клавіатура дозволяє вводити різноманітні команди.

Клавіатура має символні клавіші, призначені для введення букв, цифр і знаків пунктуації, а також спеціальні клавіші — керуючі і функціональні. Клавіші на клавіатурі згруповано у блоки.

Алфавітно-цифровий блок клавіш відповідає клавіатурі друкарської машинки.

Цифровий блок нагадує клавіатуру калькулятора. Його клавіші працюють у двох режимах — введення цифр і математичних знаків та керування курсором.

Режим змінюється клавішею **Num Lock**. Окремий блок утворюють *клавіші керування курсором*. Про призначення клавіш зі стрілками неважко здогадатися.

Home та **End** — клавіші переміщення курсору. Клавіша **Home** у вікні операційної системи здійснює перехід до першого об'єкта, а в текстовому редакторі — на початок рядка. Клавіша **End** у вікні операційної системи здійснює перехід до останнього об'єкта, в текстовому редакторі — на кінець рядка.

Використання клавіш **PgUp** та **PgDn** у вікні операційної системи є аналогічним до використання клавіш **Home** та **End**, тобто вони здійснюють перехід до першого та останнього об'єктів вікна. В текстово-

му редакторі клавіша **PgUp** здійснює перехід до попередньої сторінки тексту, **PgDn** — до наступної сторінки.



Рис. 8.1. Клавіатура з відповідними клавішами

Функціональні клавіші **F1**, **F2**, ..., **F12** не мають постійного призначення, у різних програмах воно різне. Наприклад, клавішу **F1** у більшості програм використовують для виклику довідкової системи, **F5** — для запуску програми в середовищі програмування.

Спеціальні клавіші розташовано в різних місцях клавіатури. Вони дозволяють виконувати різні команди комп'ютера. Інформацію про їхнє використання представлено в таблиці 8.1.

Таблиця 8.1. Призначення спеціальних клавіш

Enter	Підтвердження команди, запуск програми або відкриття файлу; а під час введення тексту — створення нового абзацу.
Esc	Припинення поточної операції, вихід із режиму роботи або закриття вікна
BackSpace	Перехід до попередньої папки у вікні операційної системи. У текстовому редакторі видаляє попередньо виділений фрагмент тексту або символ, розташований ліворуч від курсору
Delete	Видалення виділеного об'єкта. У текстовому редакторі видаляє символ, розташований праворуч від курсору
Insert	Перемикання між режимами вставки і заміни. У режимі вставки введений символ вставляється у позиції курсору, а в режимі заміни цей символ замінює символ, розташований праворуч від курсору
Caps Lock	Увімкнення режиму введення великих літер
Shift	При утриманні переходить до режиму введення великих літер або символів, розташованих над цифрами верхнього ряду алфавітно-цифрового блоку
Tab	Перехід по типах об'єктів вікна та створення відступів між символами в текстовому редакторі.
Ctrl, Alt	Використовуються в комбінації з іншими клавішами

За способом підключення до комп'ютера розрізняють дротові та бездротові клавіатури. Дротові клавіатури підключаються через порт USB чи спеціальний клавіатурний порт, а бездротові містять батарейку та надсилають сигнали до приймача, підключеного до порту USB.

Клавіатури можуть бути різні за призначенням, деякі з них (ігрові, музичні тощо) показано на рисунку.



8.2.

МИША

Комп'ютерна миша — це вказівний пристрій уведення. Переміщуючи його, користувач керує вказівником — спеціальним курсором, що відображається на моніторі. Миша передає відомості про своє переміщення програмі, яка у відповідь переміщує зображення вказівника.

Миша має кілька кнопок, коліщатко та інші додаткові елементи керування. З допомогою миші запускають програми, відкривають файли, переміщують фрагменти зображень тощо. Сучасні миші, як дротові, так і бездротові, зазвичай підключаються до комп'ютера через порт USB.



Дізнайтеся про походження комп'ютерної миші

8.3.

СКАНЕР

Ще один пристрій уведення інформації — це **сканер**. З його допомогою створюють цифрові копії зображення з паперових або інших носіїв. Процес отримання такої копії називають *скануванням*. Відскановане зображення, на якому присутній текст, можна перетворити на текстовий документ із використанням спеціальних програм оптичного розпізнавання текстів. Сканери бувають різних видів. Частіше за все використовуються планшетні, листові (протяжні) та ручні сканери (рис. 8.2). Останні зазвичай використовують для читання штрих-кодів.



Рис. 8.2. Планшетний, листовий (протяжний), ручний сканери

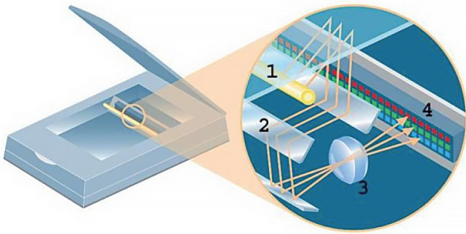


Рис. 8.3. Принцип роботи сканера

Принцип дії планшетного сканера такий. Об'єкт, що сканується, розташовують на склі планшета, під яким міститься рухома лампа, на рис. 8.3 вона позначена як 1. Відображуване від об'єкта світло крізь систему дзеркал (2) та лінзу (3) потрапляє на світлочутливу матрицю (4) — спеціальну мікросхему, яка складається зі світлочутливих елементів

(фотодіодів). Перетворена інформація передається в комп'ютер.

Основними характеристиками сканера є *роздільна здатність* (у сучасних сканерів вона зазвичай становить 2400 dpi × 2400 dpi, або точок на дюйм) і *глибина кольору* (кількість відтінків, що сприймає сканер), яка вимірюється в бітах (стандартне на сьогодні значення — 48 бітів).

8.4.

ВЕБКАМЕРА



Рис. 8.4. Вебкамери

Ще один пристрій уведення інформації — *вебкамера* (рис. 8.4). Так називають цифрову фотокамеру, що в реальному часі фіксує зображення, призначені для подальшого передавання через Інтернет. Зазвичай вебкамери транслюють зображення міських вулиць, приміщень, де проходять конференції, тощо. Вебкамера потрібна також для відеозв'язку через Інтернет з допомогою програм на зразок Zoom.

8.5.

МІКРОФОН



Гарнітура

Щоб записати звук або обмінятися голосовими повідомленнями через Інтернет, потрібен комп'ютер, оснащений *мікрофоном*. Це пристрій, що перетворює звукові коливання на коливання електричного струму. Комп'ютери деяких видів (наприклад, ноутбуки) мають

вбудований мікрофон і вебкамеру, натомість для настільних ПК ці пристрої зазвичай доводиться купувати окремо. Часто мікрофон поєднується з навушниками в одному пристрої, який називається *гарнітурою*.



Дізнайтеся більше про інші пристрої введення

ВПРАВА 8



Завдання. У файлі **Вправа_8.pptx** виконайте завдання, вказані у примітках кожного слайду.



Контрольні запитання та завдання

1. Які групи клавіш є у клавіатурі? Опишіть їхнє призначення.
2. Для чого призначено сканер? Назвіть основні характеристики сканерів.
3. Яке призначення вебкамери?
4. Перелічіть відомі вам пристрої введення.



Питання для роздумів

- 1*. Чому символічні клавіші на клавіатурі розміщені не в алфавітному порядку?



Завдання для досліджень

- 1*. Підготуйте доповідь про ігрові пристрої введення.
- 2*. Знайдіть інформацію про різні види сканерів, розкажіть про їхнє призначення.

§9.

Пристрої виведення та мультимедійні пристрої

Щоб користувач міг дізнатися про результати роботи комп'ютера, їх потрібно подати в прийнятній для людини формі, тобто у вигляді візуальної чи звукової інформації. Для цього призначені пристрої **виведення**, серед яких найбільшого поширення набули монітори, звукові колонки та пристрої друку.

9.1.

МОНІТОРИ

Одним із основних пристроїв виведення є **монітор**. На моніторі у вигляді текстової та графічної інформації відтворюються результати роботи комп'ютера. Крім того, на моніторі відображаються елементи керування, які користувач може вибрати з допомогою миші та ак-



Дізнайтеся більше про відтворення зображення на екрані монітора

тивізувати у такий спосіб певну дію (ініціювати виконання команди, видалити фрагмент тексту тощо). Зображення на моніторі формується з пікселів. Донедавна поширеними були монітори на основі електронно-променевих трубок (ЕПТ-монітори), але тепер їх замінили рідкокристалічні монітори. Основними характеристиками монітора є *роздільна здатність* (кількість пікселів по ширині та висоті) та *розмір діагоналі* його екрана.

9.2.

ЗВУКОВІ КОЛОНКИ ТА НАВУШНИКИ



Для виведення звукової інформації використовують звукові колонки. Ними обладнано більшість сучасних персональних комп'ютерів. Таких колонок може бути від двох до восьми. Колонки поділяються на пасивні та активні. Пасивні колонки не мають власного підсилювача і підключаються до виходу підсилювача звукової карти, з допомогою якої комп'ютер відтворює звукову інформацію. Активні колонки оснащені вбудованим підсилювачем, завдяки чому забезпечується краща якість звуку.

Виведення звуку відбувається також з використанням навушників, що є пристроєм для особистого прослуховування музики або інших звукових записів.

9.3.

ПРИСТРОЇ ДРУКУ

Для виведення результатів роботи комп'ютера на папір або спеціальну плівку використовують **принтери**. Хоча існує кілька різновидів цього пристрою, сьогодні користувачі надають перевагу двом із них, а саме: *струменевим* (як правило, їх використовують для кольорового друку) та *лазерним* (їх частіше використовують для чорно-білого друку). Основними характеристиками принтерів є якість та швидкість друку, а також *вартість друку* однієї копії.



Дізнайтеся більше про принцип роботи принтерів



Струменевий принтер



Лазерний принтер

Великої популярності сьогодні набуло використання багатофункціонального пристрою (БФП), який об'єднує в собі принтер, сканер та ксерокс. Його перевагою є те, що один такий пристрій займає значно менше місце та не вимагає окремого підключення кожного пристрою. Проте якість їхнього друку є трохи гіршою. БФП, як і принтери, бувають струменевими та лазерними.

Для побудови великих зображень, плакатів, карт тощо використовують спеціальні пристрої — *плотери*, які також мають технології струменевого та лазерного друку (рис. 9.1).



Рис. 9.1

9.4. МУЛЬТИМЕДІЙНЕ ОБЛАДНАННЯ

Сучасні комп'ютери широко застосовують для відтворення мультимедіа. Під цим терміном розуміють поєднання різних форм подання інформації (текстової, графічної, аудіо- та відеоінформації).

Для відтворення мультимедіа застосовують спеціальне обладнання:

- мультимедійні проєктори для проєктування на екрані відео та зображень з комп'ютера;
- інтерактивні дошки та панелі для взаємодії користувача з тим, що зображено на екрані;
- презентери для зручних презентацій.



Дізнайтеся більше про мультимедійне обладнання

ВПРАВА 9



Завдання. У файлі **Вправа_9.pptx** виконайте завдання, вказані у примітках кожного слайду.



Контрольні запитання та завдання

1. Перелічіть відомі вам пристрої виведення.
2. Назвіть основні характеристики моніторів.
3. Які існують види принтерів? Назвіть їхні характеристики.
4. Які існують ще пристрої друку? Які сфери їхнього застосування?
5. Назвіть мультимедійні пристрої. Опишіть їхнє призначення.



Питання для роздумів

1. Які переваги та недоліки у використанні БФП в порівнянні з окремими пристроями, що входять до БФП?
2. Чим відрізняються інтерактивні дошки від інтерактивних панелей?



Завдання для досліджень

- 1*. Визначте основні характеристики проєкторів.
- 2*. Дізнайтеся, які ще існують пристрої виведення, крім розглянутих у даному параграфі. Назвіть їхнє призначення.
- 3*. Відшукайте інформацію про 3D-принтери та перспективи їхнього застосування.

§10.

Технічні характеристики основних складових персонального комп'ютера

Настільні персональні комп'ютери та ноутбуки користувачі можуть використовувати для роботи з офісними документами, відтворення мультимедіа, для ігор тощо. Конфігурація комп'ютера і мінімальні параметри його складових, що дадуть змогу виконувати ці завдання, різні. Тому потрібно навчитися підбирати оптимальну комплектацію комп'ютера, щоб не витратити зайві кошти й отримати комп'ютер, який відповідатиме всім вимогам користувача під час роботи і відпочинку.

10.1.

ПОЗНАЧЕННЯ КОМПЛЕКТУЮЧИХ СИСТЕМНОГО БЛОКА КОМП'ЮТЕРА

Почнемо з дослідження інформації про комплектуючі, які ви можете побачити в магазині комп'ютерної техніки або на сайті інтернет-магазину. Наведемо приклади описів системних блоків, що пропонують в одному з інтернет-магазинів.

1. **Intel Core i5-9400F** (2.9–4.1 ГГц) / RAM 16 ГБ / HDD 1 ТБ + SSD 480 ГБ / nVidia GeForce GTX 1660 Super, 6 ГБ / без ОД / LAN
2. **AMD Ryzen 5 2600** (3.4–3.9 ГГц) / RAM 16 ГБ / HDD 1 ТБ / AMD Radeon RX 580, 8 ГБ / без ОД / LAN
3. **Intel Core i7-7700** (3.6–4.2 ГГц) / RAM 8 ГБ / SSD 512 ГБ / Intel HD Graphics 630 / DVD-RW / LAN

Проаналізувавши ці дані, розподілимо їх за компонентами, що дасть змогу порівняти їхні характеристики (табл. 10.1).

Таблиця 10.1. Порівняння характеристик системних блоків

Компонент	ПК1	ПК2	ПК3
Процесор	Intel Core i5-9400F (2.9–4.1 ГГц)	AMD Ryzen 5 2600 (3.4–3.9 ГГц)	Intel Core i7-7700 (3.6–4.2 ГГц)
Оперативна пам'ять	RAM 16 ГБ	RAM 16 ГБ	RAM 8 ГБ
Жорсткий диск	HDD 1 ТБ + SSD 480 ГБ	HDD 1 ТБ	SSD 512 ГБ
Відеокарта	nVidia GeForce GTX 1660 Super, 6 ГБ	AMD Radeon RX 580, 8 ГБ	Intel HD Graphics 630 (інтегрована відеокарта)
Оптичний дисковод	без ОД	без ОД	DVD-RW

Розглянемо характеристики комп'ютерних компонентів, на які слід звертати увагу при виборі комп'ютера.

10.2.

ЦЕНТРАЛЬНИЙ ПРОЦЕСОР І ОПЕРАТИВНА ПАМ'ЯТЬ

Сьогодні на ринку центральних процесорів переважають два виробники — Intel Corporation та Advanced Micro Devices Inc. (AMD). Кожний з них пропонує повну лінійку сучасних процесорів з інтегрованою графікою, що дозволяє в деяких випадках обійтися без зовнішньої відеокарти. Ці лінійки умовно можна розділити за призначенням. Так, в таблиці 10.2 наведено види процесорів відповідно до їхнього призначення.



Таблиця 10.2. Модель процесора та обсяг ОЗП для комп'ютерів різного призначення

Компонент	Потужні ігри	Ігри, обробка мультимедійних даних і графіки	Невимогливі ігри, любительська обробка відео і фото	Офісні завдання, навчання
Процесор Intel®	Intel® Core™ i7 (i7-4770, i7-4820K, i7-4960X)	Intel® Core™ i5 (i5-4440, i5-4570, i5-4670K)	Intel® Core™ i3 (i3-4130, i3-4340)	Intel® Pentium® (G3220, G3430)
Процесор AMD	AMD A10 APU (A10-5700, A10-6700T, A10-6800K)	AMD A8 APU (A8-5500, A8-6500T, A8-6600K)	AMD A6 APU (A6-5400K, A6-6400K)	AMD A4 APU (A4-4000, A4-5300, A4-6300)
ОЗП	16–32 Гбайти	8–16 Гбайтів	4–8 Гбайтів	2–4 Гбайти

Найбільш потужні процесори використовуються для потужних ігор, менші характеристики — для звичайних ігор та комп'ютерів, на яких передбачена обробка мультимедійних даних і графіки, менш вимогливими є комп'ютери, призначені для любительської обробки відео і фото, найменші вимоги у комп'ютерів, призначених для виконання офісних завдань та навчання.

На відміну від процесорів, уся сучасна *оперативна пам'ять*, представлена виробниками, належить до одного типу — **DDR4** і відрізняється лише робочою частотою (швидкодією) та можливістю працювати на підвищеній частоті. Вибір оперативної пам'яті для комп'ютера визначається насамперед фінансовими можливостями покупця та довірою до певного виробника.

Частоти найпоширеніших модулів пам'яті знаходяться у межах 1066–3200 МГц. Найпопулярніші на сьогодні виробники — Kingston, Samsung, Crucial. Обсяг однієї планки оперативної пам'яті має значення від 1 до 32 Гб.

Краще використовувати два однакових модулі пам'яті меншого обсягу, ніж один більшого, оскільки в цьому випадку вмикається більш швидкий двоканальний доступ до пам'яті.

Не варто гнатися за великим обсягом пам'яті та встановлювати 32 Гбайти, оскільки практичної користі від цього не буде. Windows 7/8/10 з усіма запущеними програмами та антивірусною програмою вимагає від 1 до 2 Гб, а потужні сучасні ігри зрідка використовують більше 4 Гб, тому для офісних комп'ютерів достатньо 2 Гб, для універсальних — 4 Гб, а для ігрових — 8 Гб.

10.3.

МАТЕРИНСЬКІ ПЛАТИ

Чи не першим параметром, на який потрібно звертати увагу при виборі комплекту процесор–материнська плата, є тип з'єднання (socket), за допомогою якого процесор кріпиться до плати. Для сучасних процесорів Intel® це материнські плати Socket LGA1150, а для AMD — Socket FM2(+).

10.4.

ВІДЕОКАРТИ

Далі мова піде про відеокарти. На ринку відеоадаптерів представлено продукцію двох виробників графічних процесорів: nVidia® та Ati™, які на сьогодні пропонують відеокарти на будь-який смак.

Відеокарти бувають *інтегровані, дискретні та зовнішні*.



Інтегровані відеокарти вбудовуються в материнську плату і заміни не підлягають. Подібний варіант ідеально підходить для вирішення невимогливих офісних завдань. Для обробки графіки та відео частину пам'яті ця карта забирає в оперативної пам'яті комп'ютера.

Дискретна відеокарта фактично є окремою платою, яка може бути встановлена у відповідний слот материнської плати. Таку карту можна замінити, а система на її основі може бути модернізована без особливих зусиль шляхом встановлення нової дискретної відеокарти. Цей вид відеокарти має свою пам'ять, отже, оперативна пам'ять для обробки відео не задіяна.

Зовнішні відеокарти найчастіше підключають до комп'ютера нестандартним способом, вони призначені для розширення можливостей малопотужних мобільних комп'ютерів (ноутбуки, нетбуки тощо).

Основні характеристики:

- обсяг відеопам'яті (може приймати значення від 512 Мб до 64 Гб);
- тип пам'яті (інтегрована, дискретна чи зовнішня);
- розрядність шини пам'яті (приймає значення від 64 до 4096 бітів);
- частота графічного процесора та пам'яті;
- спосіб охолодження (використовують пасивний та активний способи).

10.5.

ЖОРСТКІ ДИСКИ

Як уже зазначалося вище, в сучасних комп'ютерах використовують два види жорстких дисків: HDD та SSD (див. рис. 7.6). Найбільш потужними виробниками жорстких дисків сьогодні вважаються компанії WD, Seagate, Hitachi та Samsung. Основною характеристикою жорсткого диска є його обсяг. У домашніх комп'ютерах в основному використовують моделі дисків обсягом від 250 Гбайтів до 6 Тбайтів.

Для офісного робочого комп'ютера достатньо 500 Гбайтів, а, скажімо, для відеомонтажу може і 15 Тбайтів не вистачати. Тому обсяг диска потрібно вибирати відповідно до потреб користувача. Ставити дорогий місткий диск у робочий ПК, на якому будуть установлені тільки операційна система й офісні програми, не доцільно, тим більше, що наразі широкої популярності набуло використання хмарних технологій і частину інформації можна зберігати у хмарі.

Ще одна характеристика — це *швидкість* обертання, що визначається кількістю обертів за одну хвилину. Чим вища швидкість обертання магнітних пластин, тим швидший (отже, продуктивніший) жорсткий диск. Однак із зростанням швидкості збільшується його енергоспоживання і шум. Найпоширеніші моделі мають швидкість обертання 5 400, 5 900, 7 200 або 10 000 об/хв.

Диски зі швидкістю обертання від 5400 до 5900 обертів за хвилину — оптимальний варіант для робочого комп'ютера або ноутбука. Вони тихо працюють, не дуже нагріваються і досить економічні. Швидкість передавання даних у таких моделях досягає 150 Мбіт/с. Як правило, чим менша швидкість обертання, тим довше служить накопичувач. Моделі зі швидкістю обертання 7200 об/хв підійдуть для стаціонарних комп'ютерів середнього і вищого класів. Реальна швидкість читання/запису таких накопичувачів досягає 180 Мбіт/с.

На відміну від HDD дисків, SSD-диски не бояться ударів, мають значно більшу швидкодію. При цьому їхня вартість є значно вищою. Основним мінусом цих пристроїв є те, що при виході з ладу дані, що зберігалися на диску, відновити неможливо. Основною характеристикою цих накопичувачів є обсяг. Зараз у продажу є SSD-накопичувачі обсягом від 32 Гбайтів до 10 Тбайтів.

ВПРАВА 10



Завдання. Використовуючи дані якого небудь інтернет-магазину комп'ютерної техніки, підберіть три системні блоки під потребу одного із користувачів (згідно з вашим варіантом) з розрахунку: найвища якість — найнижча ціна. У файлі **Вправа_10.docx** складіть порівняльну таблицю. Вкажіть, який з обраних комп'ютерів відповідає вимогам. Обґрунтуйте свій вибір.

Варіант 1. Для учня 8-го класу.

Варіант 2. Для бухгалтера підприємства.

Варіант 3. Для вебдизайнера.

Варіант 4. Для тестувальника сучасних ігор.

Варіант 5. Для вчителя математики.

Варіант 6. Для оператора відеомонтажу.

**Контрольні запитання та завдання**

1. Для яких практичних цілей зазвичай використовують персональні комп'ютери?
2. Як це впливає на вибір їхньої конфігурації?
3. Чи завжди наявність більшого обсягу ОЗП впливає на швидкість комп'ютера?
4. Що таке дискретна відеокарта? Які переваги вона має порівняно із вбудованою?

**Завдання для досліджень**

1. Порівняйте основні характеристики (центральный процесор, оперативна пам'ять) настільних комп'ютерів і смартфонів. Зробіть висновок.

Пройдіть
тест**§11.****Історія обчислювальних та комп'ютерних пристроїв**

Перш ніж досягти сучасного рівня, обчислювальна техніка пройшла тривалий шлях розвитку. Загалом усю її історію можна поділити на чотири періоди — *домеханічний, механічний, електромеханічний та електронний*.

11.1.**ДОМЕХАНІЧНИЙ ПЕРІОД**

Найдавнішим лічильним пристроєм, наданим людині самою природою, була її власна рука. Спочатку люди лічили на пальцях однієї руки, потім на пальцях двох рук, а в деяких племенах використовували для цього ще й пальці ніг.

Наступним кроком у розвитку лічби стало використання камінців та інших предметів. Для запам'ятовування чисел робили зарубки на кістках тварин, в'язали вузлики на мотузках.

Перший лічильний пристрій, який звався *абак* (рис. 11.1), з'явився у Вавилоні близько 3000 року до н.е. Його довгий час застосовува-

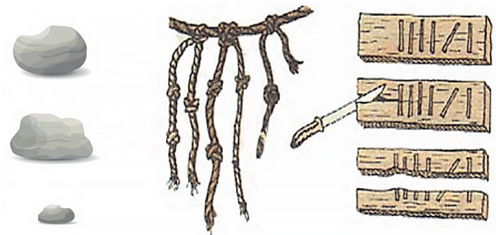




Рис. 11.1. Абак

ли у Стародавній Греції, Римі та в Західній Європі. Абак — це звичайна дошка, розділена на смуги, вздовж яких пересували камінці чи кістки. Поступово людина вдосконалювала цей пристрій. Кістки почали нанизувати на нитки, які натягували на рамку. Так з'явилася рахівниця, яку дехто використовує і нині.

11.2.

МЕХАНІЧНИЙ ПЕРІОД

Перший у світі автоматичний обчислювальний пристрій для обчислень було описано у 1623 році німецьким ученим Вільгельмом Шиккардом. Запропонована ним ідея пристрою із застосуванням підсумкового та розмножувального механізмів використовувалась надалі у всіх механічних обчислювальних машинах.

Першу механічну лічильну машину сконструював у 1645 році французький учений Блез Паскаль. Він виготовив її для полегшення роботи свого батька, що працював податковим інспектором.

Відомий німецький математик Готфрід Вільгельм Лейбніц у 1673 році створив перший арифмометр — машину, яка виконувала чотири арифметичні операції.

Таблиця 11.1. Механічні обчислювальні пристрої

1623 рік	1645 рік	1673 рік
Обчислювальний пристрій Вільгельма Шиккарда	Машина Паскаля	Арифмометр Лейбніца
		

У подальшому було створено багато конструкцій арифмометрів, зокрема арифмометр Томаса, машину Боле, арифмометр Орднера, і вони набули широко розповсюдження. Слід зазначити, що такі лічильні пристрої використовували до 80-х років ХХ століття.

У 1822 році англійський математик Чарльз Беббідж презентував свою різницеву машину. Машина була повністю механічною і складалася з безлічі шестерень і важелів. У ній використовувалася де-

сяткова система числення. Вона оперувала 18-розрядними числами з точністю до восьмого знака після коми і забезпечувала швидкість обчислень 12 членів послідовності за 1 хвилину.

Починаючи з 1834 року, Чарльз Беббідж розпочав проєктування аналітичної машини. Завдяки своїм дослідженням він сформулював принципи побудови обчислювальної машини нового типу, що мала всі основні складові сучасних комп'ютерів.

Історія донесла до нас такий цікавий факт: за проханням Беббіджа першу програму для його машини написала англійська математикиня леді Ада Лавлейс, дочка відомого англійського поета лорда Джорджа Байрона. Цю жінку вважають першою програмісткою у світі. На її честь було названо одну з мов програмування ADA.

На жаль, через нестачу коштів та низький розвиток технологій того часу, аналітична машина не була завершена при житті Беббіджа.

Проте у 1906 році його син Генрі спільно з фірмою Монро побудував діючу модель аналітичної машини, що містила арифметичний пристрій та пристрій для друку результатів.

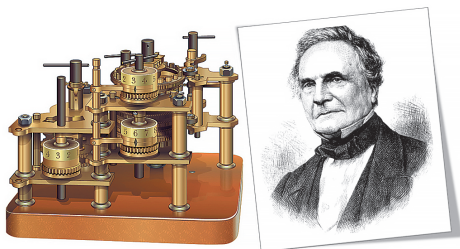


Рис. 11.2. Різницева машина Беббіджа



Рис. 11.3.
Ада Лавлейс

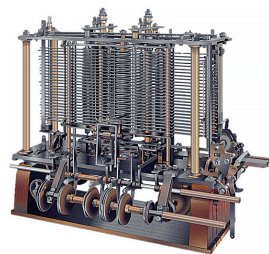


Рис. 11.4. Аналітична
машина Беббіджа

11.3.

ЕЛЕКТРОМЕХАНІЧНИЙ ПЕРІОД

З появою електроструму розпочався новий етап розвитку обчислювальної техніки — електромеханічний.

У кінці XIX ст. американець Герман Голлеріт, син німецьких емігрантів, сконструював електричний перфокартковий обчислювальний пристрій для розв'язування складних статистичних задач. Інформацію кодували за допомогою перфорацій у спеціальних картах, що розміщували в машині у певному порядку. Електричний пристрій розпізнавав отвори і посилав сигнали в обчислювальну машину. Пристрій Голлеріта був настільки вдалим, що Бюро перепису США

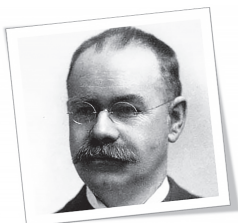


Рис. 11.5. Табулятор Голлеріта

використало його при опрацюванні даних перепису населення у 1880 році. У 1887 році Росія купила цю машину для опрацювання даних першого перепису населення. Пізніше Голлеріт очолив фірму з виробництва перфокарт, яка з часом перетворилася на відому фірму IBM.

На основі ідей Беббіджа у 1944 році в Америці було створено керуючий автоматичними послідовностями обчислювач MARK-1. Фактично це був удосконалений арифмометр завдовжки 17 метрів і заввишки два з половиною метри. Важив він біля 5 тонн. Це був програмований механізм, якому надавався рух з допомогою електродвигуна.

Проте першим автоматичним програмованим комп'ютером став Z3. Створений Конрадом Цузе програмований комп'ютер керувався перфострічкою, що була виготовлена з кіноплівки. У Z3 був повністю автоматичний контроль над операціями. Отже, в сучасному розумінні це був перший комп'ютер.

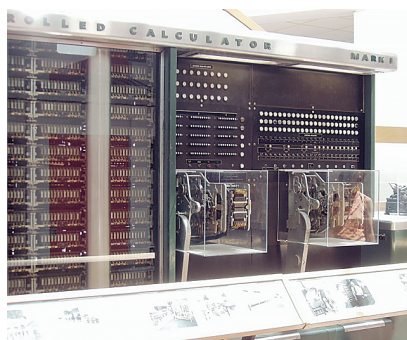


Рис. 11.6. Обчислювач MARK-1

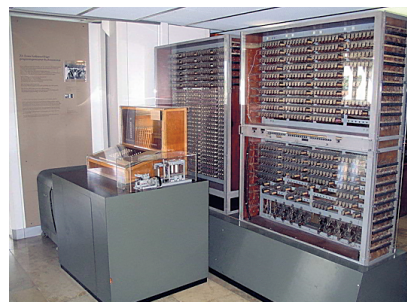


Рис. 11.7. Комп'ютер Z3

11.4.

ЕЛЕКТРОННИЙ ПЕРІОД. ПОКОЛІННЯ ЕОМ

Подальший розвиток обчислювальної техніки можна віднести до електронного періоду, який умовно поділяється на п'ять поколінь.

І покоління

Перше покоління розпочалося революцією у сфері обчислювальних машин після того, як у 1946 році в США було створено електронно-обчислювальну машину «Еніас» на основі електронних ламп. Вона була дуже великою, містила близько 18 000 електронних ламп,

важила 30 тонн і займала приміщення площею 300 кв. м. «Еніас» виконував 300 операцій множення або 5000 операцій додавання за одну секунду. Лампи дуже часто перегорали, тому Еніас міг працювати лише декілька годин поспіль. Носіями інформації в таких комп'ютерах були перфокарти та перфострічки.

Не можна оминати увагою досягнення вітчизняних учених у цьому напрямку. В 1950 році командою вчених під керівництвом Сергія Олексійовича Лебедева з Київського інституту електротехніки Української РСР була створена так звана «мала електронна обчислювальна машина» (МЕОМ). Вона містила близько 6000 електровакуумних ламп та споживала до 25 кВт. Машина могла виконувати близько 3000 операцій за секунду.

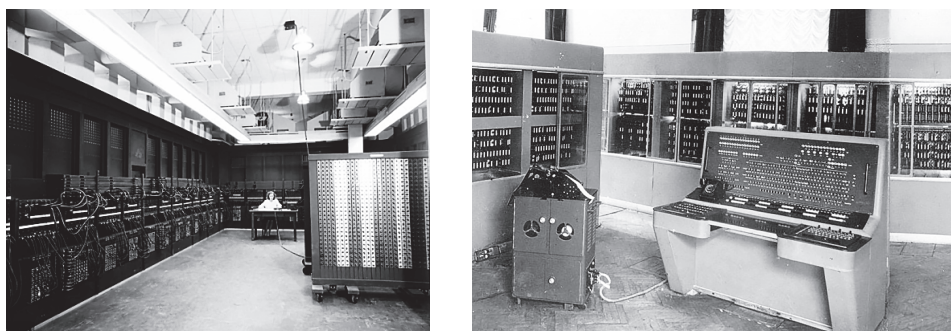


Рис. 11.8. Комп'ютери Еніас та МЕОМ

Основні характеристики комп'ютерів I покоління наведено на рис. 11.9.

Роки	Елементна база	Швидкодія	Обсяг ОЗП	Носії інформації	Програмне забезпечення	Приклади ЕОМ
з 1946 року	електронна лампа 	1 000 – 20 000 оп./с	2 Кб	перфокарти, перфострічки 	машинні коди	Еніас, МЕОМ, UNIVAC

Рис. 11.9. Характеристики комп'ютерів I покоління

II покоління

Використання електронних ламп породжувало багато проблем: виділення великої кількості тепла, що вимагало постійного охолодження приміщень потужними кондиціонерами, велика енер-

гоємність, громіздкість, ненадійність та дорожнеча. Ці проблеми були вирішені у 1947 році американськими інженерами Вільямом Шоклі, Джоном Бардіном та Волтером Бреттейном. Працюючи в компанії Bell, вони винайшли транзистор, який виконував ті ж самі функції, що й електронні лампи, використовуючи властивості напівпровідників.

З появою транзисторів розпочалося друге покоління обчислювальної техніки. Електронні обчислювальні машини почали збирати на транзисторах. Це дало змогу зменшити розміри обчислювальних машин та підвищити швидкість їхньої роботи.

У 1959 році з'явилися перші комп'ютери IBM на транзисторах. В 1960 році IBM випустила транзисторну IBM 1620, модель стала популярною як науковий комп'ютер. Їх було випущено близько 2000 екземплярів.

Застосування напівпровідників дозволило покращити не тільки центральний процесор, але й периферійні пристрої. Друге покоління пристроїв дозволяло зберігати вже десятки мільйонів символів та цифр. З'явилось розділення на жорстко закріплені пристрої зберігання та змінні пристрої. Заміна касети дисків на змінному пристрої вимагала лише кілька секунд. Хоча обсяг змінних носіїв був зазвичай нижчий, але можливість їхньої заміни давала можливість збереження практично необмеженого обсягу даних.

Найкращою вітчизняною ЕОМ II покоління вважається Велика електронно-обчислювальна машина ВЕОМ-6, створена у 1966 році, головним конструктором якої був Сергій Олексійович Лебедев.



Рис. 11.10. Комп'ютери IBM 1620 та ВЕОМ-6

Основні характеристики комп'ютерів II покоління наведено на рис. 11.11.



Роки	Елементна база	Швидкодія	Обсяг ОЗП	Носії інформації	Програмне забезпечення	Приклади ЕОМ
з 1958 року	транзистори 	100 000 – 2 000 000 оп./с	2–23 Кб	магнітні стрічки та барабани 	алгоритмічні мови, операційні системи	IBM 1620, BEOM–6, Atlas

Рис. 11.11. Характеристики комп'ютерів II покоління

III покоління

Бурхливе зростання використання комп'ютерів почалося з третього покоління обчислювальних машин. Початок цьому поклав винахід інтегральної схеми, який став можливим завдяки ланцюгу відкриттів, зроблених американськими інженерами в 1958–1959 роках.

У 1964 році був представлений мейнфрейм IBM System/360. Ці ЕОМ та її спадкоємці на довгі роки стали фактичним промисловим стандартом для потужних ЕОМ загального призначення. За шість років IBM випустила понад 30 тисяч машин.

В СРСР аналогом IBM System/360 були машини ЄС 1020.

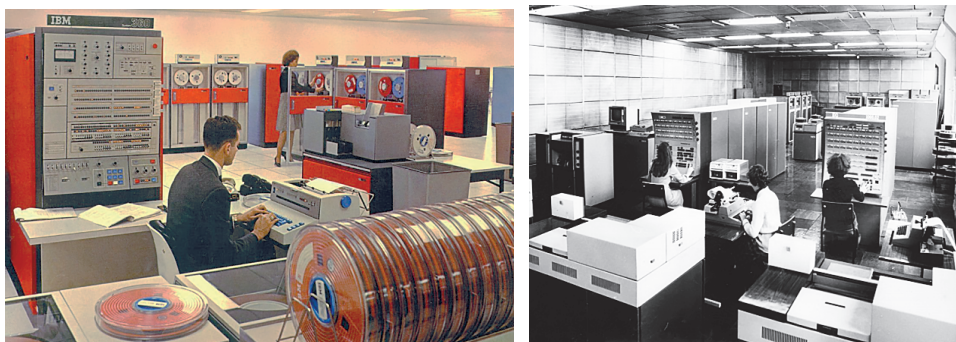


Рис. 11.12. Комп'ютери IBM System/360 та ЄС 1020

Основні характеристики комп'ютерів III покоління наведено на рис. 11.13.

Роки	Елементна база	Швидкодія	Обсяг ОЗП	Носії інформації	Програмне забезпечення	Приклади ЕОМ
з 1966 року	інтегральні схеми 	1 – 10 млн оп./с	64 Кб	магнітні диски та стрічки 	операційні системи, процедурні мови програмування	IBM System/360, EC 1020

Рис. 11.13. Характеристики комп'ютерів III покоління

IV покоління

Розвиток інтегральних схем спонукав появу надвеликих інтегральних схем. Якщо звичайна інтегральна схема могла містити біля тисячі елементів, надвеликі схеми містили у собі понад мільйон транзисторів.

У 1969 році співробітники компанії Intel створили центральний процесор на одному кристалі. Тобто, замість безлічі інтегральних мікросхем було створено одну головну інтегральну мікросхему, яка виконувала всі арифметичні, логічні операції та операції управління, записані в машинному коді. Такий пристрій отримав назву *мікропроцесор*.

У 1971 році компанія Intel на замовлення фірми Busicom випускає перший мікропроцесор Intel 4004 для використання в калькуляторі.

Поява мікропроцесорів дозволила створити невеликі комп'ютери, які могли собі дозволити купити маленькі компанії або окремі особи. У 1981 році на світ з'явився перший масовий персональний комп'ютер IBM PC. В якості основного мікропроцесора в комп'ютері використовувався 16-розрядний мікропроцесор Intel 8088, який дозволяв працювати з 1 Мегабайтом пам'яті. Незважаючи на високу ціну, у 80-х роках комп'ютери IBM PC набули широкого розповсюдження. Саме IBM PC поклав початок сім'ї сучасних комп'ютерів.

Перший масовий домашній комп'ютер був розроблений Стівом Возняком — одним із засновників компанії Apple Computer. Пізніше Возняк розробив перший масовий персональний комп'ютер.

Основні характеристики комп'ютерів IV покоління наведено на рис. 11.14.

Роки	Елементна база	Швидкодія	Обсяг ОЗП	Носії інформації	Програмне забезпечення	Приклади ЕОМ
з 1975 року	мікропроцесори 	1 – 100 млн оп./с	1–64 Мб	гнучкі диски 	бази та банки даних, прикладне ПЗ, мультимедійне ПЗ	IBM PC, Корвет

Рис. 11.14. Характеристики комп'ютерів IV покоління

Щодо комп'ютерів V-го покоління, то вони поки що не існують, передбачається, що вони будуть створені з використанням надвеликих інтегральних схем, базуватимуться на сучасних досягненнях біології, хімії та матимуть штучний інтелект.

У 1986 році в Японії розроблено проєкт комп'ютерів, стрижнем якого є комп'ютери з асоціативною організацією пам'яті, потужними засобами введення, виведення та опрацювання інформації, побудованих на принципах самоорганізації і самоадаптації. Інакше кажучи, це комп'ютери, наділені засобами штучного інтелекту.



Дізнайтеся більше про внесок вітчизняних учених у розвиток ЕОМ

ВПРАВА 11



Завдання. У файлі **Вправа_11.pptx** виконайте завдання, вказані у примітках кожного слайду.



Контрольні запитання та завдання

1. Охарактеризуйте основні етапи розвитку обчислювальної техніки.
2. Які елементні бази використовувалися в кожному поколінні комп'ютерів?
3. Які носії інформації використовувалися в кожному поколінні комп'ютерів?
4. Які перспективи розвитку V-го покоління комп'ютерів?



Питання для роздумів

1. Які причини появи примітивних засобів для виконання обчислень?
2. Які основні ознаки поділу комп'ютерів на покоління?
3. Які ви бачите перспективи розвитку обчислювальної техніки?



Завдання для досліджень

1. Підготуйте доповідь про історію походження різних комп'ютерних пристроїв.
2. Яким був внесок у розвиток комп'ютерної техніки Леонардо да Вінчі?
3. Які існували різновиди абаків?



Пройдіть
тест

§12.

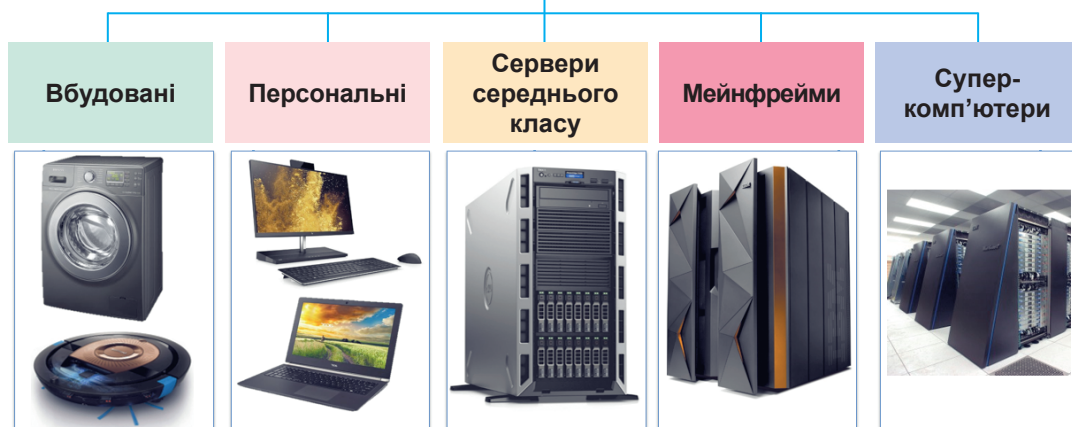
Види сучасних комп'ютерів та їхнє застосування

12.1.

КЛАСИФІКАЦІЯ КОМП'ЮТЕРІВ

Сучасні комп'ютери залежно від розміру, призначення і вартості поділяють на п'ять типів: вбудовані, персональні, сервери середнього класу, мейнфрейми та суперкомп'ютери.

Види комп'ютерів



12.2.

ВБУДОВАНІ КОМП'ЮТЕРИ

Вбудовані комп'ютери — це невеликі комп'ютери, якими оснащують побутові прилади та автомобілі.

Широкого застосування набуло використання вбудованих комп'ютерів у побутових приладах: роботи-пилососи, пральні машини, мультиварки, мікрохвильові печі, кондиціонери тощо.

Вбудованими комп'ютерами оснащені сучасні автомобілі. Зокрема, в автомобілі комп'ютер керує клімат-контролем, роботою двигуна та інших систем, спрощує паркування, надає різноманітну інформацію водієві.

12.3. ПЕРСОНАЛЬНІ КОМП'ЮТЕРИ

Персональні комп'ютери — це комп'ютери, якими в певний момент часу може користуватися одна особа. Персональні комп'ютери можна поділити на дві категорії: **стаціонарні** та **портативні**.

Під категорією **стаціонарного** ПК мається на увазі комп'ютер, який має постійне місце, і не передбачається його часта зміна. Живлення цих комп'ютерів здійснюється через електричну мережу.

Виділяються три основних типи таких пристроїв: настільні комп'ютери, неттопи та моноблоки.

Найбільш високопродуктивними представниками персональних комп'ютерів є саме *настільні комп'ютери*. Цей тип комп'ютера називається настільним тому, що місцем розміщення служить письмовий або комп'ютерний стіл. Основним компонентом тут вважається системний блок. До системного блока підключається монітор, клавіатура і миша, всі ці компоненти взаємодіють між собою. Системний блок є модульним пристроєм, що означає можливість заміни кожного комплектуючого модуля окремо, тим самим змінюючи конфігурацію під себе.

Другий тип стаціонарних комп'ютерів — *неттопи*. Неттоп — це мінімізована версія системного блока, крім розмірів, відрізняється меншим енергоспоживанням і виділенням шуму, і як наслідком, в більшості випадків — меншою продуктивністю. Але все це сприяє відмінній інтеграції з інтер'єром офісу або будинку.

Третій тип стаціонарних комп'ютерів — *моноблоки*. Моноблок є так само стаціонарним комп'ютером, але без видимого системного блока. Каркасом для такого типу комп'ютерів, як моноблок, служить один загальний корпус з монітором. Усі комплектуючі розміщуються в тилівій частині, за дисплеєм, що створює певну естетичність для роботи за комп'ютером.

Друга категорія персональних комп'ютерів — **портативні комп'ютери**. Це комп'ютери, які є невеликими за розмірами і передбачають легкість зміни місця їхнього перебування. Живлення цих комп'ютерів здійснюється переважно завдяки акумулятору.

Одним із найбільш розповсюджених типів портативних комп'ютерів є *ноутбуки* та *нетбуки*.

Ноутбуки — це комп'ютери, які можна легко переносити, мають можливість автономної роботи, яка можлива саме завдяки акумуляторній батареї. Крім цього, відмінна риса від стаціонарних комп'ютерів — суміщення складових системного блока, дисплея та клавіатури в одному корпусі. А здатність складатися робить їх ще компактнішими.

Нетбуки — це ті ж ноутбуки, які мають менші розміри та більше часу роботи від акумуляторної батареї. Проте ці переваги впливають на зменшення їхньої продуктивності, що не дає можливості ними використовувати ресурсомісткі додатки.



Ще один тип портативних комп'ютерів — *планшети та планшетні ноутбуки*, їх ще називають трансформерами. *Планшети* — це переносні комп'ютери, які в одному корпусі містять обчислювальний потенціал для перегляду інтернет-сторінок, відео та прослуховування музики. Основою тут є пристрій уведення — сенсорний дисплей (тачскрін). Планшети є справжніми помічниками, а в деяких ситуаціях зручніші, ніж ноутбуки, за рахунок менших розмірів. У планшетних ноутбуків, крім тачскріна, так само, як і в ноутбука, є клавіатура, і в залежності від моделі, вона може складатися, під'єднуватися через окремий дріт або висуватися. Основним завданням даного типу портативного пристрою є саме вихід і використання Інтернету, хоча в ньому і присутні основні мультимедійні можливості. Для таких мінікомп'ютерів час автономної роботи є важливішим, ніж продуктивність.

Третій тип портативних комп'ютерів — *смартфони*. На відміну від інших портативних пристроїв, смартфони мають також функцію телефону. Так само, як і інші види комп'ютерів, які є представниками портативних пристроїв, смартфони залежні від часу роботи акумулятора. Мають невеликі розміри, що сприяє можливості мати їх завжди під рукою. Звичайно ці переваги не можуть зменшувати їхню продуктивність у порівнянні з більшими за розмірами пристроями. Завдяки нижчій розподільній здатності є можливість використовувати пристрої для перегляду відео та фото, крім того, вони оснащені зручними можливостями фото- і відеозйомки. Пристроєм уведення смартфонів є сенсорний дисплей.

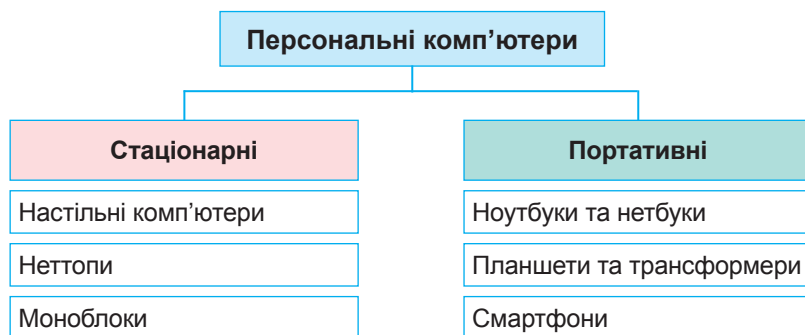


Рис. 12.1. Види персональних комп'ютерів

12.4. СЕРВЕРИ СЕРЕДЬОГО КЛАСУ

Сервери середнього класу — це комп'ютери, призначені для керування інформаційними системами невеликих і середніх підприємств. Такі сервери потужніші за персональні комп'ютери, можуть одночасно обслуговувати багато користувачів, які підключаються до них з допомогою своїх персональних комп'ютерів.

12.5. МЕЙНФРЕЙМИ

Мейнфрейми є серверами вищого класу. Це потужні комп'ютери, які встановлюють великі підприємства, банки, мобільні оператори тощо. Сервери цього класу розраховані на безперервну цілодобову роботу й одночасно обслуговують запити тисяч користувачів.

12.6. СУПЕРКОМП'ЮТЕРИ

Ще один вид комп'ютерів — суперкомп'ютери. Так називають комп'ютери з величезною обчислювальною потужністю. Їх використовують для виконання завдань, які вимагають інтенсивних обчислень, зокрема для прогнозування погодно-кліматичних умов, керування космічними польотами. Традиційний суперкомп'ютер коштує кілька мільйонів доларів. Щоб зменшити його вартість, почали використовувати суперкомп'ютерні кластери, що складаються з кількох серверів середнього класу, які діють як один.



Дізнайтеся більше про використання сучасних суперкомп'ютерів



Контрольні запитання та завдання

1. Які існують види комп'ютерів?
2. Яка відмінність стаціонарних персональних комп'ютерів від портативних?



Питання для роздумів

- 1*. Що сприяло поширенню персональних комп'ютерів?
- 2*. Визначте, з якими видами комп'ютерів ви працюєте у вашому навчальному закладі, з якими вам доводиться працювати вдома.



Завдання для досліджень

- 1*. З'ясуйте, що таке дата-центр і які послуги він надає.



Пройдіть тест



Висновки

Кодування інформації — це процес перетворення інформації з форми, зручної для безпосереднього використання, у форму, зручну для передавання, зберігання або автоматичної обробки.

Найменшою одиницею вимірювання інформації є **1 біт**.

Для кодування інформації в комп'ютері використовують **двійковий код**.

Для кодування тексту використовують набори символів ASCII, Unicode тощо.

Для вимірювання інформації використовують **змістовий** та **алфавітний** підходи.

Для кодування графіки використовують **растрове** та **векторне** подання.

Апаратна складова комп'ютера містить у собі пристрої обробки, зберігання, введення, виведення та комунікаційні пристрої.

Основні характеристики, на які слід звертати увагу при виборі **комп'ютера**, — це процесор, оперативна пам'ять, материнська плата, жорсткий диск та відеокарта.

Історію обчислювальної техніки можна поділити на чотири періоди — домеханічний, механічний, електромеханічний та електронний.

Сучасні комп'ютери залежно від розміру, призначення і вартості поділяють на п'ять типів: вбудовані, персональні, сервери середнього класу, мейнфрейми та суперкомп'ютери.



Розгадайте електронний кросворд

Розділ 2

Опрацювання текстових даних



§13.

Пошук та заміна фрагментів тексту

Ще в 5-му класі ви ознайомилися з роботою текстового редактора. Продовжимо знайомство з ним на вкладці текстового процесора Microsoft Word 2016.

Однією із можливостей текстового процесора є редагування тексту з використанням засобів пошуку та заміни.



Пригадайте основні поняття, що пов'язані з текстовим редактором

13.1.

ПОШУК ТЕКСТОВИХ ФРАГМЕНТІВ

При роботі з текстом часто виникає потреба пошуку в тексті деяких фрагментів. Ця можливість стане у нагоді для перевірки правильності написання тексту, виділення фрагментів тексту, вилучення зайвих символів тощо.

Для пошуку фрагмента тексту необхідно на вкладці **Основне** обрати інструмент **Пошук** або натиснути комбінацію клавіш **Ctrl+F**. Після цього ліворуч з'явиться панель **Навігація**, що містить поле для введення текстового фрагмента, який потрібно відшукати.

Для пошуку окремих слів краще скористатися розширеним пошуком. Для цього в меню інструмента **Пошук** слід обрати команду **Розширений пошук**. В результаті буде відкрито вікно **Пошук і замінування** (рис. 13.1). У даному вікні в полі **Знайти** необхідно ввести бажане слово. Якщо натиснути кнопку **Більше** та встановити прапорець **Лише слова повністю**, програма буде шукати лише окремі слова. У вікні **Пошук і замінування** у разі потреби можна встановити й інші параметри: врахування регістра, префікса, суфікса тощо.



Ознайомтеся з видами недрукованих символів

Якщо нам потрібно знайти усі входження слова, де воно починається з великої літери, у вікні розширеного пошуку слів слід установити прапорець **Враховувати регістр**.

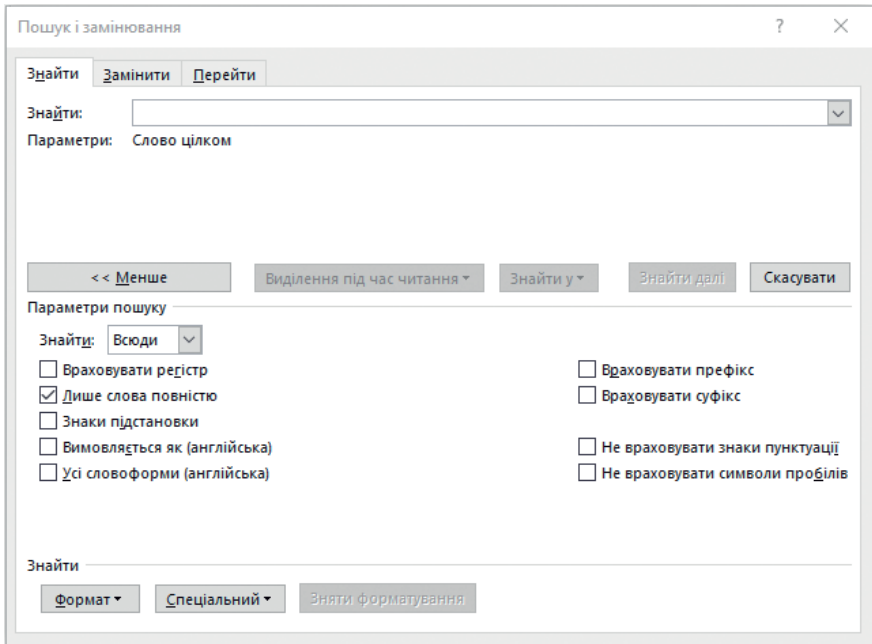


Рис. 13.1. Вікно пошуку і замінювання

При пошуку деяких фрагментів тексту зручно користуватися так званими *символами підстановки*.

Наприклад, нам потрібно знайти усі входження чисел. Для цього в полі пошуку слід ввести [0-9], що означатиме пошук будь-якої цифри від 0 до 9. Але для використання символів підстановки у вікні пошуку потрібно обов'язково встановити прапорець біля опції **Знаки підстановки**.

Символи підстановки можна вказувати і для пошуку слів, у яких відбувається чергування літер. Наприклад, якщо нам потрібно знайти усі входження слова **Київ** у різних відмінках. Тобто це може бути як слово **Київ**, так і слово **Києва**. Для цього в полі пошуку слід ввести наступний напис (**Ки[іє]в**).

У таблиці 13.1 наведено найбільш уживані символи підстановки.



Дізнайтеся більше про пошук фрагмента тексту за його форматом

Між іншим, якщо ви забули, які символи підстановки якій дії відповідають, підказку на ці символи можна отримати безпосередньо у вікні **Пошук і замінювання**. Для цього достатньо клацнути кнопку **Спеціальний**. В результаті ви побачите список символів підстановки.

Таблиця 13.1. Символи підстановки

Символ підстановки	Опис	Приклад
*	Будь-яка кількість будь-яких символів	На запит чуд* будуть знайдені слова чудес, чудово, чудових
?	Будь-який один символ	На запит ст?к будуть знайдені слова стук, стек, стік, стик
[]	Відповідає символам у дужках	На запит Ки[іє]в будуть знайдені слова, що містять Київ та Києв
-	Визначає діапазон символів однієї літери	На запит с[а-о]н будуть знайдені слова сон, сан, син , але не слон
!	Виключення символу з дужок	На запит ст[!е]к будуть знайдені слова стук, стік, стик , але не стек
#	Будь-який цифровий символ	На запит 1#3 будуть знайдені числа 123, 113, 143, 193, ...

13.2.

ЗАМІНА ТЕКСТОВИХ ФРАГМЕНТІВ

Під час роботи з документом часто трапляються ситуації, коли потрібно не лише відшукати всі входження фрагмента в текст, а й замінити кожне з них іншим фрагментом. Для цього використовують вкладку **Замінити** у вікні **Пошук і замінування** (рис. 13.1). Також цю вкладку можна викликати, натиснувши інструмент **Замінити** на вкладці **Основне**.

Ця вкладка містить такі самі елементи, як і вкладка **Знайти**, а також додаткове поле **Замінити на**, куди вводять текст, яким має бути замінений шуканий.

Досить часто можливості заміни фрагментів тексту використовують для видалення зайвих пробілів, примусових перенесень рядка тощо.

Розглянемо, як прибрати в документі зайві пробіли.

1. Викличте діалогове вікно **Пошук і замінування**.
2. У полі **Знайти** введіть два пробіли, а у полі **Замінити на** введіть один пробіл.
3. Натисніть кнопку **Замінити все**. В результаті з'явиться вікно з інформацією про кількість виконаних заміни. Натисніть **ОК**. Знову натисніть кнопку **Замінити все**. І так робіть доти, поки кількість заміни не стане рівною нулю.

Існує більш раціональний спосіб виконання процедури позбавлення тексту від пробілів. Розглянемо і його.

1. Установіть курсор у поле **Знайти:**.
2. Натисніть кнопку **Спеціальний**.
3. У списку, що з'явився, оберіть **Пустий простір**. У результаті в полі пошуку з'являться символи **w^**, це спеціальне позначення, з допомогою якого прибираються усі пробіли з документа за один крок пошуку.
4. Щоб на місці багатьох пробілів залишився лише один, у полі **Замінити на:** залишіть уведений пробіл. І знову натисніть кнопку **Замінити все**. В цьому випадку відразу всі зайві пробіли буде вилучено.

ВПРАВА 13

Завдання. У файлі **Вправа_13.docx** виконайте завдання, вказані нижче.

1. Видаліть зайві пробіли між словами.
2. Замініть установлені підряд два символи розриву рядка на символи абзацу.
3. Замініть поодинокі символи розриву рядка на символи пробілу.
4. Знайдіть у документі кількість входжень чисел, що містять значення року.
5. Як відомо, за правилами оформлення знаків пунктуації, знак дефісу позначається в документі у вигляді короткої риски, біля якої знаки пробілу не ставляться, а знак тире позначається довгою рискою та відокремлюється знаками пробілу. Відредагуйте документ згідно із зазначеними правилами, а саме: замініть для позначення тире маленькі риски на великі, а символи, що мають позначати дефіс, відобразіть у вигляді короткої риски.
6. Також відповідно до правила оформлення пунктуації пробіли не ставляться перед знаками пунктуації (точкою, комою, двокрапкою, знаком оклику, знаком питання), після дужки, що відкривається, та перед дужкою, що закривається. Видаліть зайві пробіли відповідно до даного правила.
7. Збережіть файл.

**Контрольні запитання та завдання**

1. Які способи редагування тексту ви знаєте?
2. Опишіть порядок пошуку фрагмента тексту.
3. Як здійснити пошук за цілими словами?
- 4*. Що таке символи підстановки? Яким є їхнє призначення в процесі пошуку та заміни фрагментів тексту?



Питання для роздумів

1. Чому після копіювання тексту, відформатованого в програмі Word, у програму Блокнот зникає форматування символів?
2. Для чого необхідно мати можливість відображення недрукованих символів?
- 3*. Як у тексті можна замінити розриви рядків на символи завершення абзацу?



Завдання для досліджень

- 1*. Як з допомогою символів підстановки створити запит для пошуку входжень у тексті будь-якого символу латинського алфавіту?
- 2*. Як з допомогою символів підстановки створити запит для пошуку усіх слів, що не містять латинських літер «x», «y» та «z»?

§14.

Використання стилів

14.1.

ПОНЯТТЯ СТИЛЮ, ТИПИ СТИЛІВ

Вивчаючи засоби і методи форматування тексту, ви використовували області **Шрифт** та **Абзац** на вкладці **Основне**, а також відповідні діалогові вікна. Параметри форматування при цьому призначалися вручну окремо для кожного абзацу та фрагмента тексту. Цей метод називається *безпосереднім форматуванням*. Основний його недолік очевидний: для зміни форматування потрібно кожного разу вручну задавати параметри об'єкта. Уникнути цього можна в разі форматування з *використанням стилів*. Що ж таке стиль?

Стиль — це іменованій набір параметрів форматування, які можна застосувати до об'єкта певного типу.

Оскільки кожний вид об'єктів має свій набір параметрів, для їхнього зберігання потрібні стилі різних типів. Зокрема, у програмі Word їх чотири:

- **стиль знака** — визначає лише параметри символів (шрифт, розмір, накреслення тощо);
- **стиль абзацу** — визначає параметри абзацу і параметри шрифту, що застосовуються до всього абзацу;

- **стиль таблиці** — зберігає інформацію про оформлення меж таблиці, кольори та різні параметри тексту, що в ній зберігається, про заливку клітинок, вирівнювання тексту в них;
- **стиль списку** — містить відомості про форматування маркованого або нумерованого списку.

14.2.

ПЕРЕГЛЯД ІНФОРМАЦІЇ ПРО СТИЛІ

Переглядати та застосовувати стилі, а також керувати ними можна за допомогою списку стилів, який можна викликати, клацаючи на значок відкриття діалогового вікна із розділу **Стилі** вкладки **Основне**. В результаті чого праворуч буде відображено вікно **Стилі** (рис. 14.1).

В основній частині вікна представлено список, що містить перелік усіх стилів документа, а також комбінації параметрів, застосованих до тексту під час безпосереднього форматування.

Зверніть увагу на позначки, встановлені біля назв стилів праворуч. Це може бути літера **a**, символ ¶ або символи ¶a. Що ж ці позначення означають? Неважко здогадатися, що літера **a** — це ознака того, що стиль використовується для форматування окремих символів, знак абзацу (¶) — для усього абзацу, а сукупність символів ¶a призначена для символу і для абзацу одночасно.

При встановленні курсору на будь-якому тексті документа у списку стилів прямокутником буде обмежено стиль обраного в тексті фрагмента. При наведенні на цей прямокутник з'являється стрілка. Натиснувши її, можна перейти в меню, яке містить команди для роботи зі стилем (зміна, виділення в тексті об'єктів з даним стилем, видалення тощо) (рис. 14.2).

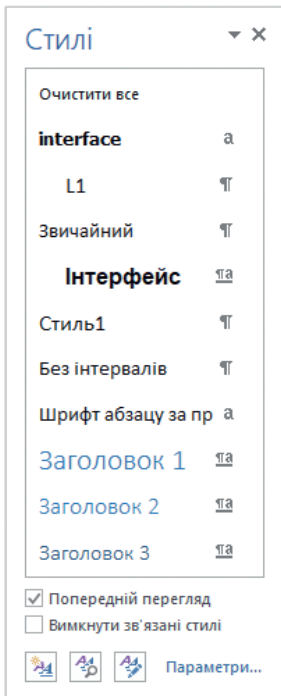


Рис. 14.1

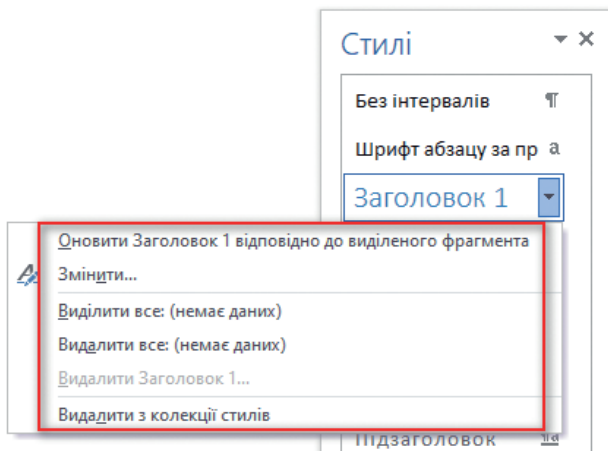


Рис. 14.2



Якщо обрати команду **Виділити все**, то програма виділить усі відформатовані з використанням цього стилю фрагменти, і ви зможете призначити для них інший стиль, вибравши його із представленого списку.

14.3.

ЗАСТОСУВАННЯ СТИЛІВ

Форматувати текст за допомогою стилів дуже просто. Для цього необхідно:

- 1) виділити фрагмент тексту або встановити курсор у потрібний абзац чи слово;
- 2) у вікні **Стилі** обрати необхідний стиль. Після чого вигляд тексту зміниться — він відповідатиме заданим у стилі параметрам.

Розглянемо способи застосування стилів.

Для застосування стилю **до фрагменту тексту** необхідно виділити цей фрагмент та обрати бажаний стиль. Після цього цей фрагмент буде відформатовано відповідно до обраного стилю.

Аби застосувати стиль **до окремого слова**, його необов'язково виділяти, достатньо на цьому слові встановити курсор та обрати бажаний стиль.

Аналогічно можна застосувати стиль **до абзацу**. Для цього слід установити курсор в абзац та обрати стиль, призначений для форматування абзацу. В результаті абзац буде відформатований відповідно до обраного стилю.


Використовуючи стильове оформлення, слід урахувати, що коли до фрагмента застосовано і стиль знака, і стиль абзацу, перевагу має перший із них. Поверх будь-якого стилю можна виконувати і безпосереднє форматування, воно переважатиме над стильовим оформленням.

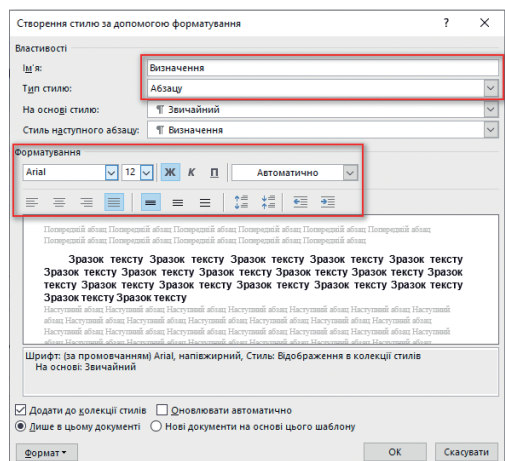
Якщо до тексту застосовано безпосереднє форматування, у багатьох випадках перед використанням стилів бажано його позбутися. Для очищення тексту від форматування та відміни усіх стилів необхідно виділити весь текст, натиснувши **Ctrl+A**, та обрати першу команду у вікні **Стилі: Очистити все**.

14.4.

СТВОРЕННЯ СТИЛІВ

Розглянемо, як створити новий стиль.

- 1) У вікні **Стилі** оберіть інструмент **Створити стиль** .
- 2) У діалоговому вікні **Створення стилю за допомогою форматування** введіть:
 - ім'я стилю в полі **Ім'я**;



- у списку **Тип стилю** оберіть об'єкт, до якого слід застосувати стиль (абзацу, символу тощо);
- укажіть необхідні параметри форматування шрифту та абзацу в області **Форматування** та натисніть **ОК**.

У результаті новий стиль з'явиться у переліку стилів, і його можна буде використовувати.

14.5.

ЗМІНА СТИЛЮ

Після застосування стилю може виявитися, що для документа не підходять установлені у ньому шрифт або, наприклад, параметри абзацу. У такому разі можна змінити стиль і тим самим переформатувати весь оформлений у відповідності з ним текст.

Для цього у вікні **Стилі** біля стилю, який необхідно змінити, слід клацнути на кнопку зі стрілкою та в меню, що з'явилося, обрати команду **Змінити**. Після чого відкриється таке саме вікно, як для створення стилю, і ви зможете змінити потрібні параметри.

Якщо для створення чи зміни шрифту недостатньо параметрів форматування, запропонованих у відповідному діалоговому вікні, клацніть у цьому вікні кнопку **Формат** та оберіть необхідну команду в меню цієї кнопки.



Дізнайтеся більше про збереження створених стилів у шаблоні

Так, після вибору команди **Абзац** відкривається діалогове вікно, де можна задати параметри абзацу, а при виборі команди **Шрифт** — відповідно вікно форматування шрифту.

ВПРАВА 14.1



Завдання. У даній вправі ви навчитеся створювати власні стилі та застосовувати їх до текстових фрагментів. У файлі **Вправа_14_1.docx** виконайте завдання, вказані нижче.

1. Створіть стилі для заголовка та числових даних. Для заголовка: ім'я — **Основний заголовок**, тип — **Абзац**, шрифт — **Times New Roman**, розмір — **14**, **напівжирний**. Для числових даних: ім'я — **Число**, тип — **Символу**, шрифт — **Consolas**, розмір — **11**, **зелений**, **напівжирний**, **курсив**.
 - a. Відкрийте вікно **Стилі** (на лінійці **Основне** в області **Стилі**).
 - b. Виділіть заголовок документа та у вікні **Стилі** оберіть інструмент (**Створити стиль**).
 - c. У діалоговому вікні, що відкриється, в полі **Ім'я** введіть — **Основний заголовок**.
 - d. У полі **Тип стилю** — **абзацу**.
 - e. У полі **Стиль наступного абзацу** — **звичайний**.



- f. В області **Форматування** встановіть необхідні параметри шрифту (шрифт — **Times New Roman**, розмір — **14**, **напівжирний**).
 - g. Натисніть **ОК** для підтвердження встановлених параметрів.
 - h. Аналогічно виконайте створення стилю для числових даних документа.
2. Змініть стилі підкресленого тексту документа та тексту, виділеного курсивом. Для підкресленого тексту встановіть такі параметри: шрифт — **Arial**, колір — **синій**, **напівжирний**, **курсив**. Для тексту, виділеного червоним кольором, установіть такі параметри: шрифт — **Times New Roman**, колір — **коричневий**, розмір шрифту — **12**, **напівжирний**, **курсив**.
- a. Клацніть на будь-якому підкресленому слові, після чого у вікні стилів активним стане стиль **виділений текст**.
 - b. Клацніть на стрілочці праворуч біля даного стилю та оберіть у випадаючому меню команду **Змінити**.
 - c. У діалоговому вікні, що відкриється, введіть необхідні параметри (шрифт — **Arial**, колір — **синій**, **напівжирний**, **курсив**).
 - d. Натисніть **ОК** для підтвердження встановлених параметрів.
 - e. Аналогічно виконайте зміну стилю для тексту, виділеного червоним кольором (цей стиль в даному документі має назву **Цікаві факти**).
3. Збережіть документ.



Виконайте вправу 14.2

**Контрольні запитання та завдання**

1. Що таке стиль? Які переваги надає застосування стилів?
2. Як переглянути інформацію про стилі?
3. До яких об'єктів документа Word можна застосувати стилі?
- 4*. Опишіть послідовність дій створення нового стилю.

**Питання для роздумів**

- 1*. Наведіть приклади використання стилів.

**Завдання для досліджень**

1. Визначте, чи можна при зміні параметрів стилю змінити його тип. Наприклад, стиль було застосовано до абзацу. Чи можна його застосувати до знака?
2. Прочитайте в довідці Word, як створити стиль списку, і створіть такий стиль.
- 3*. Визначте, що відбувається у випадку видалення стилю з документа.

Пройдіть
тест

§15.

Розділи. Колонтитули

15.1.

ВЕРСТКА СТОРІНОК

Програма Word дає змогу по-різному оформлювати окремі блоки тексту, наприклад, розташувати один фрагмент тексту в кількох колонках, один аркуш у документі зробити вертикальним, інший — горизонтальним. Це стає можливим завдяки тому, що документ можна поділити на кілька розділів (не слід плутати їх із розділами і підрозділами, на які текст поділяється за допомогою заголовків).

Розділ у MS Word — це частина документа, що має певні параметри форматування сторінки.

Новий розділ створюється у разі зміни таких параметрів, як кількість стовпців, розміри полів та орієнтація сторінки. За умовчанням у документі є лише один розділ. Для створення нового розділу потрібно встановити курсор у місце, де має закінчуватися наявний розділ, на вкладці **Макет** обрати команду **Розрив**, вибрати відповідну команду із випадаючого вікна в області **Розриви розділів**. У результаті розділ буде створено.

Розриви розділів і розриви сторінок можна побачити у режимі відображення недрукованих символів. Нагадаємо, що він встановлюється за допомогою інструмента **Відобразити всі знаки** ¶ (рис. 15.1).

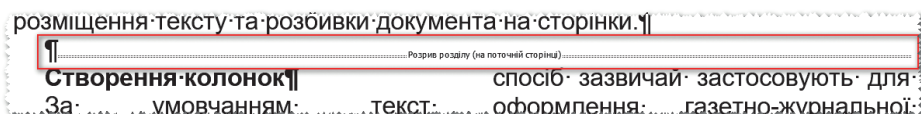


Рис. 15.1

Якщо текст не поміщається на одній сторінці, програма автоматично розбиває документ на сторінки. Проте у деяких випадках, наприклад, коли заголовок, що міститься в кінці сторінки, потрібно перенести на наступну сторінку або коли розділ документа має розпочинатися з нової сторінки, виникає потреба у примусовому переміщенні тексту. Робиться це шляхом примусового створення розриву сторінки. Щоб виконати розрив сторінки, необхідно встановити курсор у місце розриву сторінки і натиснути комбінацію клавіш **Ctrl+Enter**. Також це можна зробити, обравши на вкладці **Вставлення** інструмент **Розрив сторінки**.

Розрив сторінки, а також установлення інших розривів у документі можна виконати на вкладці **Макет**, обравши команду **Розриви**. Після чого у випадаючому вікні обрати необхідну команду.

Створення розриву сторінки і розриву розділу можна скасувати. Для цього потрібно спочатку активізувати режим відображення недрукованих символів, а потім установити курсор на початок лінії розриву і натиснути клавішу **Delete**. Після цього розділ матиме такі самі параметри сторінки і розташування тексту, що й попередній розділ.

15.2. СТВОРЕННЯ КОЛОНТИТУЛІВ

Ви маєте можливість на кожній сторінці документа, крім її номера, відобразити ще й деяку додаткову інформацію, наприклад, назву розділу, параграфа тощо. З цією метою використовуються колонтитули.

Отже, **колонтитул** — це розташований вверху чи внизу сторінки рядок, у якому зазвичай міститься назва документа, назва розділу чи, скажімо, номер сторінки. Колонтитул також може містити малюнок.

У документі Word можна створювати як нижні, так і верхні колонтитули. При цьому ви маєте змогу задати однакові колонтитули для всього документа або особливі колонтитули для певного розділу, першої сторінки розділу, парних та непарних сторінок.



15.3. АЛГОРИТМ СТВОРЕННЯ КОЛОНТИТУЛА

1. Оберіть вкладку **Вставлення**.
2. У групі **Колонтитули** оберіть інструмент **Верхній колонтитул** або **Нижній колонтитул**, у залежності від місця його розташування.
3. Оберіть, наприклад, **Верхній колонтитул**. У результаті має відкритися меню, в якому можна обрати зовнішній вигляд одного із вбудованих колонтитулів, а також команди **Змінити верхній колонтитул** та **Видалити верхній колонтитул**.
4. Оберіть перший вбудований тип. У результаті цього основний текст документа стане блідим і з'явиться поле для введення тексту колонтитула, а також відокремлюється пунктирною лінією ділянка верхнього колонтитула. Також у режимі роботи з колонтитулами стає активною вкладка **Конструктор** у розділі **Знаряддя для колонтитулів** (рис. 15.2)

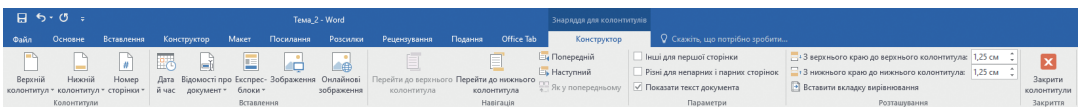


Рис. 15. 2. Вкладка **Конструктор** в розділі **Знаряддя для колонтитулів**

5. Уведіть текст верхнього колонтитула.
6. Для завершення роботи з колонтитулом оберіть інструмент **Закрити колонтитули** або двічі клацніть в області основного тексту документа.

У більшості документів колонтитул на першій сторінці не розташовують. Аби змінити створений колонтитул, відкрийте розділ **Знаряддя для колонтитулів**. Для цього можна обрати команду **Змінити верхній колонтитул** в меню інструмента **Верхній колонтитул**. Також цей розділ можна викликати, клацнувши двічі на колонтитулі.

Аби прибрати колонтитул з першої сторінки, необхідно встановити прапорець **Інші для першої сторінки**. В результаті з першої сторінки колонтитул буде прибрано, а на інших він залишиться.

Для переходу між колонтитулами використовуються інструменти з області **Навігація**. Для переходу між нижнім та верхнім колонтитулами використовуються інструменти **Перейти до нижнього колонтитула** та **Перейти до верхнього колонтитула**, а для переходу від колонтитула до колонтитула по сторінках — інструменти **Попередній** та **Наступний**.

Досить часто при роботі з документами в одному колонтитулі міститься більш загальна інформація, а в другому (нижньому) колонтитулі доцільно розташувати назви окремих розділів документа. В MS Word реалізовано можливість уведення власних колонтитулів для різних розділів документів. Як це можна реалізувати, розглянемо у вправі 15.


15.4.

НУМЕРАЦІЯ СТОРІНОК

Сторінки великих документів бажано нумерувати, аби не заплутатися у великій кількості сторінок однотипного тексту. Нумерацію сторінок устанавлюють аналогічно до створення колонтитулів.

1. На вкладці **Вставлення** оберіть в області **Колонтитули** кнопку **Номер сторінки**. В результаті має з'явитися випадаюче меню, в якому можна обрати спосіб розташування номера сторінки, встановити формат номерів сторінок у діалоговому вікні, а також видалити номери сторінок.
2. Оберіть перший варіант розташування номера сторінки. В результаті буде створено нумерацію, і документ отримає вигляд, як у випадку створення колонтитулів. Тут також буде відображено вкладку **Конструктор** розділу **Знаряддя для колонтитулів**. На цій вкладці можна задати, чи буде відображено номер на першій сторінці документа, встановлюючи прапорець **Інакше для першої сторінки**.

Розглянемо зміну параметрів нумерації сторінок.

Оберіть інструмент  (Номер сторінки).

В його меню скористайтеся командою **Формат номерів сторінок**.

В результаті відкривається вікно **Формат номера сторінки** (рис. 15.3), в якому у полі **Формат номерів** можна змінити формат номера, встановити початковий номер для нумерації тощо.

Видалення колонтитулів

Як створювати та змінювати колонтитули ви вже знаєте, залишилося навчитися їх видаляти. Вміст колонтитула, а також номери сторінок можна видалити вручну в режимі роботи з колонтитулами — у такий самий спосіб, як видаляється основний текст документа.

Також для видалення колонтитулів можна скористатися командами **Видалити верхній колонтитул** або **Видалити нижній колонтитул**, а для видалення номерів сторінок — командою **Видалити номери сторінок**.

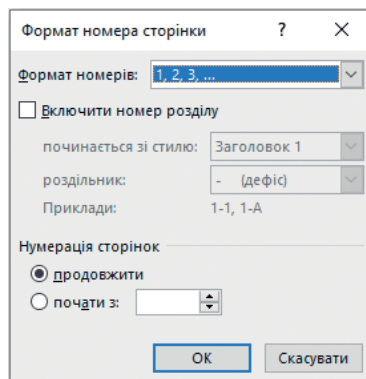


Рис. 15.3. Вікно форматування номера сторінки

ВПРАВА 15




Завдання. У даній вправі ви навчитеся створювати та редагувати колонтитули, номери сторінок та виконувати дії з розділами. У файлі **Вправа_15.docx** виконайте завдання, вказані нижче.

1. Створіть верхній колонтитул з написом «7 чудес Києва» відповідно до зразка.



7 чудес Києва

- а. На вкладці **Вставлення** оберіть інструмент **Верхній колонтитул** та оберіть перший зразок.
- б. Додайте зображення. Для цього на вкладці **Вставлення** оберіть інструмент **Зображення**. Після цього в діалоговому вікні оберіть файл **kyiv.png**.
- в. Зробіть відступ від зображення, натиснувши клавішу **Tab** необхідну кількість разів.
- г. Уведіть текст **7 чудес Києва**.
- ґ. Зробіть для колонтитула підкреслення. Для цього виділіть вміст верхнього колонтитула. Відкрийте меню інструмента  (Підкреслення) та оберіть команду **Інші підкреслення**. Встановіть необхідні параметри підкреслення.
- д. Закрийте панель для роботи з колонтитулами.

2. Приберіть колонтитул з першої сторінки.
 - а. Клацніть 2 рази в області верхнього колонтитула.
 - б. На вкладці **Знаряддя для колонтитулів** ▶ **Конструктор** установіть опцію **Інші для першої сторінки**.
 - в. Закрийте панель для роботи з колонтитулами.
3. Створіть розділи, починаючи з кожного заголовка другого рівня, розташовуючи кожний розділ з нової сторінки.
 - а. Установіть курсор перед заголовком **II. Володимирський собор**.
 - б. На вкладці **Макет** оберіть меню інструмента **Розриви**.
 - в. Оберіть команду **Наступна сторінка**. В результаті заголовок має переміститися на початок наступної сторінки.
 - г. Аналогічно створіть розриви і перед іншими заголовками другого рівня.
4. Відновіть верхні колонтитули для перших сторінок кожного розділу.
 - а. Увійдіть до верхнього колонтитула розділу, що містить інформацію про Києво-Могилянську академію.
 - б. На вкладці **Знаряддя для колонтитулів** ▶ **Конструктор** приберіть опцію **Інші для першої сторінки**.
 - в. Аналогічно відновіть верхні колонтитули для перших сторінок інших розділів.
5. Додайте нижній колонтитул, у якому ліворуч виведіть назву кожного розділу, а праворуч — номер сторінки.

Нижній колонтитул - Розділ 2-

Як у попередньому

Києво-Могилянська академія

2

- а. На вкладці **Вставлення** оберіть інструмент **Нижній колонтитул** та оберіть перший зразок.
 - б. Уведіть назву першого заголовка **Києво-Могилянська академія**.
 - в. Два рази натисніть клавішу **Tab**.
 - г. Для додавання нумерації сторінок оберіть інструмент **Номер сторінки** ▶ **Поточне положення** ▶ **Звичайні цифри**.
 - г. Виділіть нижній колонтитул.
 - д. Перейдіть на вкладку **Основне**.
 - е. В меню інструмента **Заливка** оберіть світло-синій колір. У результаті напис та номер сторінки будуть відображені на світло-синьому тлі.
 - є. Перейдіть до нижнього колонтитула наступного розділу.
 - ж. Зніміть активний режим з інструмента **Як у попередньому**.
 - з. Уведіть замість наявної назви назву другого розділу **Володимирський собор**.
 - и. Аналогічно встановіть нижні колонтитули для інших розділів.
6. Збережіть документ.



Контрольні запитання та завдання

1. Що таке розділ? За яких умов створюється новий розділ?
2. Що таке колонтитул? Які є види колонтитулів?
3. Опишіть послідовність дій для додавання номерів сторінок. Як вилучити номер на першій сторінці документа?
- 4*. Як вилучити створений колонтитул або номер сторінки?



Питання для роздумів

- 1*. Наведіть приклади використання колонтитулів.
- 2*. Як змінити початок нумерації сторінок у документі?



Завдання для досліджень

- 1*. Визначте, які об'єкти Word, окрім тексту та зображень, можна додати до колонтитулів.
- 2*. Чи можна зробити так, щоб у документі колонтитул був на всіх сторінках, окрім першої та другої?

§16.

Структура документа

16.1.

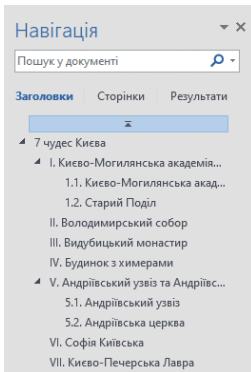
СТВОРЕННЯ СТРУКТУРИ ДОКУМЕНТА

Структуру великого документа доцільно формувати з використанням стилів. Насамперед йде мова про призначення заголовкам



Рис. 16.1. Структура заголовків у документі

і підзаголовкам стилів різного рівня. Заголовку верхнього рівня варто призначити стиль **Заголовок 1**, наступного — **Заголовок 2** і т. д. до стилю **Заголовок 6**. Основному тексту зазвичай надають стиль **Звичайний** (рис. 16.1).



Це дозволяє не лише змінювати форматування всіх заголовків вибраного стилю, а й користуватися перевагами області переходів і швидко виводити на екран потрібний фрагмент документа.

Відобразити цю панель можна, встановивши на вкладці **Подання** прапорець **Область переходів**. Після цього ліворуч відобразиться панель **Навігація** зі структурою заголовків.

Робота на цій панелі подібна до роботи з деревом папок: вибираючи заголовки на панелі, можна переходити до потрібних підрозділів документа. При виборі відповідного заголовка документа в області навігації ми відразу будемо потрапляти до потрібного розділу.

16.2.

АВТОМАТИЧНЕ СТВОРЕННЯ ЗМІСТУ

Якщо документ великий, для нього доцільно створити зміст. Програма Word формує його автоматично, але це можливо лише в разі, якщо для заголовків застосовано вбудовані стилі заголовків. Наприклад, як у розглянутому нами документі «7 чудес Києва» (рис. 16.1).

Алгоритм автоматичного створення змісту

1. Установіть курсор в те місце документа, де зміст має розміщуватися (зазвичай це початок або кінець документа).
2. Оберіть вкладку **Посилання**. На цій вкладці виділено окремий розділ **Зміст**, інструменти якого відповідають за створення та редагування змісту документа.
3. Для створення змісту оберіть інструмент **Зміст**. В результаті чого відкриється меню цього інструмента, в якому можна обрати один із запропонованих зразків змісту або встановити власні налаштування, обравши команду **Настроюваний зміст**.
4. Створіть зміст автоматично без додаткових налаштувань, обравши для цього перший зразок автоматичного змісту. В результаті буде створено зміст, що має чітко виважену структуру із вказуванням номерів сторінок розташування відповідних розділів. Для переходу до певного розділу достатньо клацнути на ньому, втримуючи клавішу **Ctrl**.

Звичайно, після створення змісту в основному документі можуть відбутися певні зміни, наприклад, може бути вилучено або додано



деякий текст. У цьому випадку могли змінитися номери сторінок, на яких розташовано заголовки розділів документа. Також могли змінитися і назви заголовків.



Дізнайтеся більше про створення настроюваного змісту

Для оновлення змісту оберіть на вкладці **Посилання** інструмент **Оновити таблицю**. В результаті відкривається діалогове вікно **Оновлення змісту**.

У цьому вікні можна обрати опції:

- **оновити лише номери сторінок** (ця опція використовується, якщо в документі додано або вилучено основний текст, а назви заголовків не змінювалися);
- **оновити цілком** (використовується у разі зміни заголовків, включених до змісту).



Дізнайтеся більше про додавання у документ покажчиків

ВПРАВА 16



Завдання. У даній вправі ви навчитеся створювати та оновлювати зміст та предметний покажчик. Для цього відкрийте файл із вправою з попереднього уроку або файл **Вправа_16.docx** та виконайте завдання, вказані нижче.

1. Вставте у документ нову сторінку після першої сторінки.
2. Створіть автоматичний зміст документа відповідно до зразка (рис. 16.2).

Зміст	
I. Києво-Могилянська академія і Старий Поділ.....	3
1.1. Києво-Могилянська академія.....	3
1.2. Старий Поділ.....	5
II. Володимирський собор.....	9
III. Видубицький монастир.....	11
IV. Будинок з химерами.....	13
V. Андріївський узвіз та Андріївська церква.....	15
5.1. Андріївський узвіз.....	15
5.2. Андріївська церква.....	15
VI. Софія Київська.....	17
VII. Києво-Печерська лавра.....	19
Предметний покажчик.....	20

Рис. 16.2.

- 3*. На останній сторінці створіть предметний покажчик відповідно до зразка, представленого у файлі **Вправа_16_зразок.pdf**.
- 4*. Заголовку предметного покажчика надайте стиль заголовка другого рівня.
- 5*. Оновіть зміст. В результаті у змісті має з'явитися посилання на предметний покажчик.
6. Збережіть файл.

**Контрольні запитання та завдання**

1. Що таке структура документа?
2. Яким є призначення панелі навігації?
3. Опишіть послідовність дій для створення автозмісту.
- 4*. За яких умов можна автоматично створити зміст у документі?
5. Що таке покажчики?
6. Опишіть послідовність дій для створення покажчиків.

**Завдання для досліджень**

- 1*. Ознайомтеся з призначенням інструментів групи **Виноски** на вкладці **Посилання**. З'ясуйте, як додавати виноски.
- 2*. Додайте до текстового документа кілька зображень та пронумеруйте їх з допомогою команди **Вставити назву** з групи **Підписи** на вкладці **Посилання**. Поясніть принцип нумерації зображень.

§17.**Гіперпосилання в текстових документах****17.1.****ПОНЯТТЯ ГІПЕРПОСИЛАННЯ**

Чи зустрічалися ви з гіперпосиланнями, працюючи на комп'ютері? Звичайно, зустрічалися, і не один раз. Переважно це трапляється, коли ви працюєте із сайтами в мережі Інтернет, переміщаючись від однієї сторінки до іншої. Саме гіперпосилання надають нам можливість перейти від однієї сторінки до іншої. Виявляється, гіперпосилання можна створювати і в текстовому процесорі Word.

Давайте визначимо, що ж таке взагалі гіперпосилання?

Гіперпосилання — це текст або зображення на сторінці, натиснення на яке викликає перехід на іншу сторінку чи іншу частину поточної сторінки.

Гіперпосилання може посилатися на місце в цьому ж документі, а також відкривати інші документи. Розглянемо обидва випадки.

17.2.

СТВОРЕННЯ ВНУТРІШНІХ ГІПЕРПОСИЛАНЬ

Гіперпосилання, що посилається на місце в цьому ж документі, називають **внутрішнім гіперпосиланням**.

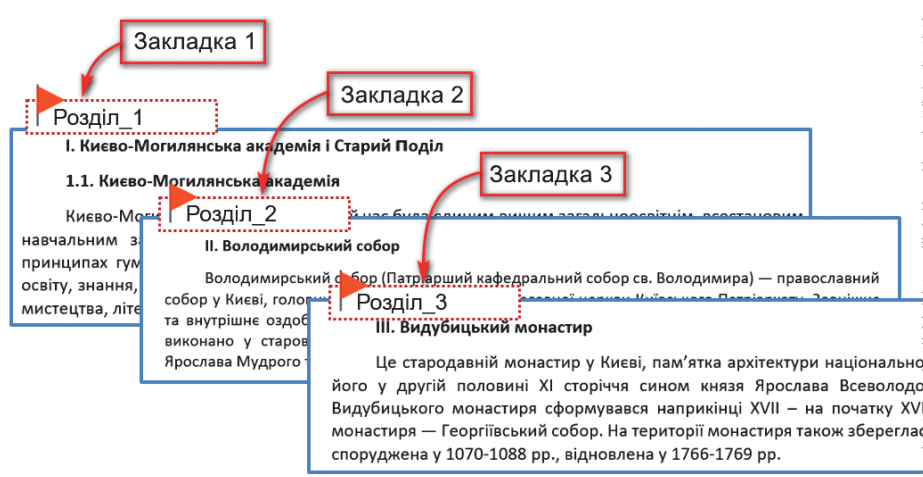
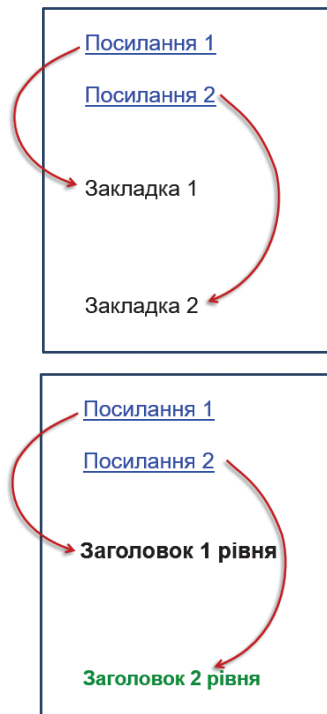
Для його реалізації необхідні два елементи: **посилання**, яке вказує, куди необхідно перейти, і **закладка**, яка визначає місце, у яке необхідно здійснити перехід. Роль закладок можуть також виконувати **заголовки**, яким надано стилі заголовків відповідних рівнів.

Розглянемо створення внутрішніх гіперпосилань з використанням закладок. Першим кроком буде створення закладок на те місце, куди будуть указувати гіперпосилання. В нашому випадку це будуть заголовки відповідних розділів.

При утворенні закладок слід надати їм імена. Щодо надання імен, то вони мають відповідати певним правилам:

- ім'я складається з літер, цифр та символів підкреслення;
- ім'я починається з літери і не може починатися цифрою або символом підкреслення;
- ім'я не може містити пробіли.

Відповідно до наведених правил надамо закладкам імена **Розділ_1**, **Розділ_2**, **Розділ_3** і т. д.



Розглянемо створення закладок в MS Word, виконавши практичне завдання.

ПРАКТИЧНЕ ЗАВДАННЯ 1

1. Відкрийте документ **7 чудес Києва.docx**.
2. Перемістіться на початок першого розділу.
3. Виділіть заголовок першого розділу (І. Києво-Могилянська академія і Старий Поділ).
4. На вкладці **Вставлення** в області **Посилання** оберіть інструмент **Закладка**.
5. У вікні, що з'явилося, введіть ім'я закладки **Розділ_1**.
6. Натисніть кнопку **Додати**. В результаті закладки буде створено.

Зауваження. Для відображення закладок у документі оберіть вкладку **Файл**, команду **Параметри** ▶ **Додатково** і в розділі **Відображення вмісту** документа встановіть прапорець біля опції **Відображати закладки**. Натисніть **ОК**. У результаті в документі закладки будуть обмежені сірими квадратними дужками.

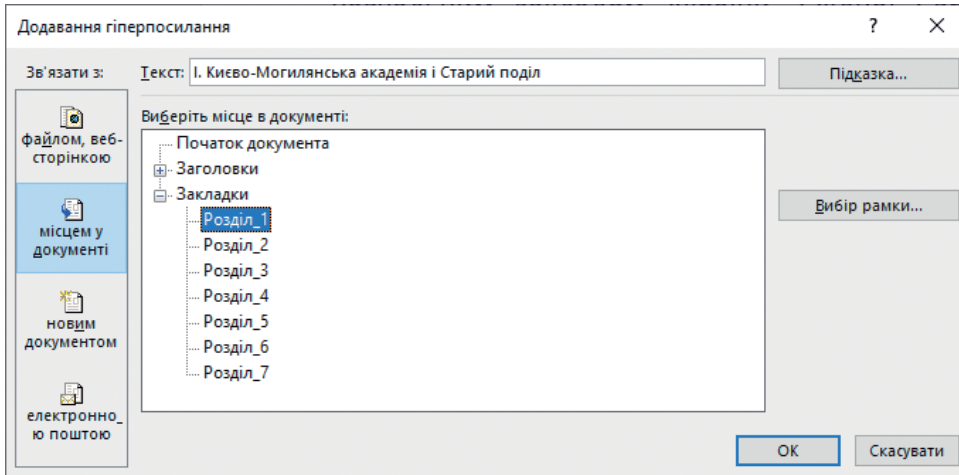
7. Аналогічно додайте закладки і для інших розділів.
Аби з будь-якого місця документа можна було перейти до бажаної закладки, необхідно обрати інструмент **Закладка**, вибрати у списку необхідну закладку та натиснути кнопку **Перейти**.
Після того, як закладки створено, наступним кроком буде створення гіперпосилань на відповідні закладки.
8. На другій сторінці документа створіть навігаційну панель відповідно до зразка.

Меню

№ 1	Києво-Могилянська академія і Старий Поділ
№ 2	Володимирський собор
№ 3	Видубицький монастир
№ 4	Будинок з химерами
№ 5	Андріївський узвіз та Андріївська церква
№ 6	Софія Київська
№ 7	Києво-Печерська лавра

Рис. 17.1. Зразок навігаційної панелі

9. Для створення гіперпосилання виділіть текст першого пункту навігаційної панелі (Києво-Могилянська академія і Старий Поділ).
10. На вкладці **Вставлення** в області **Посилання** оберіть інструмент **Гіперпосилання**.
11. У вікні, що відкриється, в області **Зв'язати з** оберіть **місцем у документі**.
12. Оберіть назву закладки, на яку необхідно зробити посилання. В нашому випадку це буде закладка **Розділ_1**. Натисніть **ОК**.



У результаті гіперпосилання буде створено. Цей факт також підтверджує те, що шрифт обраного пункту має стати синім та підкресленим. При наведенні на цей пункт має з'явитися повідомлення, що містить назву закладки та підказку про те, що для переміщення на закладку слід натиснути **Ctrl** і посилання.

13. Натисніть посилання для перевірки його роботи.

Якщо ви повернетеся до навігаційної панелі, то побачите, що колір посилання змінився на фіолетовий. Це означає, що за посиланням було здійснено перехід.

14. Аналогічно додайте посилання на інші розділи документа.

Іноді кольори посилань не завжди гармонійно відображаються відповідно до загального дизайну документа. При бажанні їх можна змінити. Для цього слід обрати вкладку **Конструктор**, інструмент **Кольори** та обрати команду **Настроїти кольори**. Після чого встановити бажані кольори **Гіперпосилання** та **Переглянуте гіперпосилання**.

Як ви, напевно, звернули увагу, при наведенні на посилання з'являється підказка з назвою закладки. При бажанні цю підказку можна змінити. Для цього в контекстному меню посилання необхідно обрати команду **Змінити гіперпосилання**. В результаті відкриється те саме вікно, як і при створенні посилання. Тут можна змінити об'єкт посилання, видалити посилання, а також установити або змінити підказку, що з'являється при наведенні на посилання. Натисніть у цьому вікні кнопку **Підказка** і введіть підказку з назвою розділу **Києво-Могилянська академія і Старий Поділ**, натисніть двічі **ОК**. Після наведення на пункт меню з'явиться змінена підказка.

Розглянутий щойно спосіб створення гіперпосилань використовується, коли в документі заголовкам не призначено вбудовані стилі заголовків.

У випадку структурування документа встановленням заголовкам вбудованих стилів заголовків, створення гіперпосилань значно спрощується. В цьому випадку не потрібно створювати закладки, а гіперпосилання можна створювати на заголовки. З цим способом ви познайомитеся, виконуючи вправу 17.

17.3.

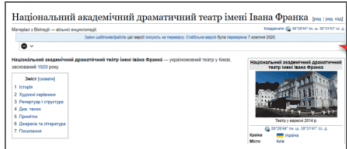


СТВОРЕННЯ ЗОВНІШНІХ ГІПЕРПОСИЛАНЬ

Як було зазначено раніше, крім внутрішніх гіперпосилань можна створювати і гіперпосилання для виклику окремих файлів, вебсторінок тощо. Такі гіперпосилання називають *зовнішніми*.

Розглянемо, як створити зовнішні гіперпосилання.

ПРАКТИЧНЕ ЗАВДАННЯ 2

1. Відкрийте файл **7 Чудес Києва.docx** та перейдіть до розділу **IV. Будинок з химерами**.

IV. Будинок з химерами

Це, мабуть, найвідоміша будівля у Києві. Будинок розташований у теперішньому центрі Києва, неподалік від Палацу президента України, а на початку ХХ століття все було інакше. Тоді в цій частині Києва були проведені роботи по впорядкуванню цієї доволі занедбаної території. На місці осушеного Козиного болота побудували театр Соловцова (нинішній [театр ім. І. Франка](#)), а ділянка землі на кручі, що височила над театром, продавалася на міському аукціоні за дуже низькою ціною, оскільки вона була тьє непридатна для будівництва. Тому рішення архітектора [Владислава Городецького](#) зводити там будинок для своєї родини викликало всезагальне здивування.

[Будинок з химерами](#) побудований у стилі модерн у 1902-1903 рр. За рік побудували сам будинок, і ще два роки було витрачено на зовнішнє та

2. Додайте гіперпосилання на текст **Владислава Городецького**, що має відкрити файл **Gorodeckiy.docx**.
 - a. Виділіть текст, на який будете робити посилання. (На рисунку вище це текст **Владислава Городецького**.)
 - b. На вкладці **Вставлення** в області **Посилання** оберіть інструмент **Гіперпосилання**.
 - c. За умовчанням у вікні встановлено **Зв'язати з: файлом, вебсторінкою**. Залишіть ці налаштування без змін.
 - d. Оберіть файл **Gorodeckiy.docx**, що має відкритися при виборі гіперпосилання (папка **Практичне завдання 2**).
 - e. Установіть підказку, що відповідає гіперпосиланню.

- f. Натисніть **OK**.
 - g. Перевірте роботу гіперпосилання. Для цього наведіть на посилання, натиснувши **Ctrl** та посилання. В результаті має відкритися відповідний файл.
3. Аналогічно додайте гіперпосилання на текст **Будинок з химерами**, що має відкрити зображення з файлу **Building.jpg**.
 4. Додайте гіперпосилання на текст **театр ім. І.Франка**, що має відкрити вебсторінку у Вікіпедії.
 - a. Відкрийте українську Вікіпедію та знайдіть у ній статтю за ключовою фразою **театр імені Франка**.
 - b. Скопіюйте адресу посилання.
 - c. Перейдіть до документа та виділіть напис **театр ім. І.Франка**.
 - d. Оберіть інструмент **Гіперпосилання**.
 - e. У поле **Адреса:** вставте з буфера обміну скопійовану URL-адресу.
 - f. Додайте відповідну підказку.
 - g. Натисніть **OK**.
 - h. Перевірте роботу гіперпосилання.

17.4.

ВИЛУЧЕННЯ ГІПЕРПОСИЛАНЬ

Досить часто при копіюванні документів з мережі Інтернет скопійований текст містить гіперпосилання на різні джерела мережі (рис. 17.2). Але використання в документі гіперпосилань не завжди є доцільним. Отже, ви маєте знати, як вилучити непотрібні гіперпосилання.

Київ — [столиця](#) та найбільше місто [України](#), одне з найбільших і найстаріших міст [Європи](#). Розташований у середній течії [Дніпра](#), у північній [Наддніпрянщині](#). Політичний, соціально-економічний, транспортний та освітньо-науковий центр країни. Окрема [адміністративно-територіальна одиниця](#) в складі України й адміністративний центр [Київської області](#). Районний центр [Києво-Святошинського району](#). Адміністративно до складу [Київської області](#) не входить Місце розташування центральних органів влади України, [іноземних місій](#), штаб-квартир більшості підприємств і громадських об'єднань, що працюють в Україні.

Рис. 17.2. Приклад тексту, скопійованого з вебсторінки

Для вилучення гіперпосилання достатньо перейти в контекстне меню посилання та обрати команду **Видалити гіперпосилання**. В результаті посилання буде вилучено.

Проте це не дуже придатний спосіб, коли посилань дуже багато. Для вилучення великої кількості посилань необхідно спочатку виділити увесь документ, це можна зробити з допомогою комбінації клавіш **Ctrl+A**, після чого натиснути **Ctrl+Shift+F9**. У результаті всі посилання, що містив документ, буде вилучено.

ВПРАВА 17



Завдання. У даній вправі ви навчитеся створювати внутрішні гіперпосилання з використанням заголовків, а також створите зовнішні гіперпосилання. Для цього відкрийте файл із вправою з попереднього уроку або файл **Вправа_17.docx** та виконайте завдання, вказані нижче.

1. Видаліть на другій сторінці зміст.
2. Створіть навігаційну панель відповідно до зразка (рис. 17.1).
3. Створіть гіперпосилання від пунктів меню до відповідних розділів.
 - a. Виділіть напис **Києво-Могилянська академія і Старий Поділ**.
 - b. На вкладці **Вставлення** оберіть інструмент **Гіперпосилання**.
 - c. В області **Зв'язати з:** оберіть **Місцем у документі**.
 - d. Оберіть заголовок **Києво-Могилянська академія і Старий Поділ**.
 - e. Додайте підказку. Для цього натисніть кнопку **Підказка** та введіть підказку, що відповідає тексту гіперпосилання.
 - f. Двічі натисніть **ОК**.
 - g. Аналогічно створіть й інші гіперпосилання.
4. Перевірте роботу гіперпосилань, натискаючи на пункти меню з утриманням клавіші **Ctrl**.
5. Додайте закладку на меню.
 - a. На другій сторінці встановіть курсор у першому рядку.
 - b. На вкладці **Вставлення** оберіть інструмент **Закладка**.
 - c. Уведіть ім'я закладки: **Меню**.
 - d. Натисніть **Додати**.
6. Додайте гіперпосилання від назв розділів до меню.
 - a. На третій сторінці виділіть заголовок першого розділу.
 - b. На вкладці **Вставлення** оберіть інструмент **Гіперпосилання**.
 - c. В області **Зв'язати з:** оберіть **Місцем у документі**.
 - d. Оберіть закладку **Меню**.
 - e. Натисніть **ОК**.
 - f. Аналогічно створіть й інші гіперпосилання.
7. Перевірте роботу гіперпосилань, натискаючи на заголовки з утриманням клавіші **Ctrl**.
8. Створіть зовнішнє гіперпосилання на вебсторінку Вікіпедії на текстовий фрагмент «**Галшки Гулевичівни**».
9. Аналогічно створіть гіперпосилання з інших текстових фрагментів документа на сторінки Вікіпедії (принаймні по одному з кожного розділу).
10. Збережіть файл.
11. Зразок виконання вправи перегляньте у файлі **Вправа_17_зразок.pdf**.



Контрольні запитання та завдання

1. Що таке гіперпосилання?
2. Чим відрізняються внутрішні посилання від зовнішніх?
3. Опишіть послідовність дій для створення внутрішнього гіперпосилання.
- 4*. Для чого в документі використовують закладки?
5. Які існують види зовнішніх посилань?



Питання для роздумів

- 1*. Чи можна засобами MS Word створити вебсторінку?
- 2*. Яким є практичне призначення використання гіперпосилань у текстовому документі?



Завдання для досліджень

- 1*. Визначте, чи можна встановити посилання в колонтитулі.



Пройдіть
тест

§18.


Спільна робота з документом

З розповсюдженням Інтернету кардинально змінилося поняття робочого колективу. Люди, що живуть в різних країнах і працюють в різних організаціях, можуть брати участь у спільній розробці проєктів чи опрацюванні одних і тих же документів. Сьогодні вже нікого не здивуєш співробітниками, які зовсім ніколи не заглядають в офіс компанії, а виконують свою роботу, не покидаючи власного будинку. Інші — постійно переміщуються по світу і беруть участь у різних нарадах, семінарах, перебуваючи в дорозі.

Доволі часто співробітники, що працюють над одним проєктом, перебувають у різних будівлях, кабінетах тощо. У таких випадках для організації документообігу між учасниками проєкту необхідно організувати *віртуальний офіс*, де будуть зберігатися документи для спільного використання. Для цього в більшості випадків використовують хмарні технології. З цими технологіями



ми ви вже знайомі з минулого року. Отже, у разі надання доступу до документа з ним можуть працювати одночасно декілька осіб.

Для надання спільного доступу в MS Word необхідно обрати кнопку  **Спільний доступ** у верхньому правому куті програми.

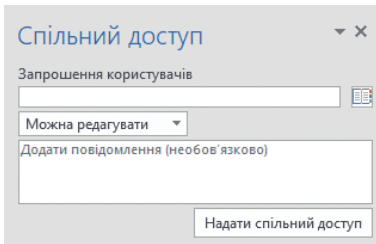


Рис. 18.1. Вікно надання спільного доступу



Дізнайтеся більше про спільну роботу в режимі офлайн

Після чого ви маєте зберегти документ у хмарі (OneDrive), натиснувши кнопку **Зберегти у хмарі**. Після чого надати спільний доступ до документа, заповнивши поля у вікні запрошення користувачів (рис. 18.1). Тут ви маєте ввести електронну адресу користувача та тип доступу (редагування чи перегляд).

Після натиснення кнопки **Надати спільний доступ** користувач одержить листа з посиланням на спільний документ і може розпочати роботу з ним.

При редагуванні файла в MS Word може з'явитися повідомлення про те, що користувачі, яким було надано спільний доступ, також працюють з цим файлом. Для спільної роботи над документом не існує особливого прийому. Ви редагуєте, як зазвичай, і якщо інші користувачі редагують документ, MS Word попереджає вас про їхню присутність і показує, в якому місці документа відбувається редагування іншим користувачем.

Після натиснення кнопки **Надати спільний доступ** користувач одержить листа з посиланням на спільний документ і може розпочати роботу з ним.

ВПРАВА 18



Завдання. Виконайте завдання вправи у парі. Ви маєте надати доступ до власного документа сусіду/сусідці та отримати посилання на документ, доступ до якого вам надано сусідом/сусідкою.

1. Відкрийте документ **Анкета.docx**.
2. Додайте у ньому власні три запитання.
3. Збережіть документ у хмарі з назвою **Вправа_18_Прізвище.docx**.
4. Надайте доступ до свого документа сусідові/сусідці для редагування та вчителю/вчительці для перегляду. Попередньо дізнайтесь адреси їхніх електронних скриньок.
5. Увійдіть до власної електронної скриньки та отримайте посилання на документ, який вам надіслано сусідом/сусідкою.
6. Почніть редагування документа сусіда/сусідки, відповідаючи на запитання анкети.
7. Також відслідкуйте зміни, які відбуваються у вашому документі (як здійснюється редагування документа сусідом/сусідкою).

**Контрольні запитання та завдання**

1. Які технології дозволяють організувати спільний доступ?
2. Опишіть послідовність дій надання спільного доступу до документа.

**Питання для роздумів**

- 1*. Укажіть платформи для організації спільного доступу до документа в режимі онлайн.
- 2*. Визначте, скільки осіб можуть редагувати одночасно документ.

§19.**Створення складного документа**

Протягом попередніх уроків ми розглядали різноманітні засоби програми MS Word. У цьому параграфі ми розглянемо послідовність дій щодо створення складного документа, які передбачають використання цих засобів.

При створенні, редагуванні та форматуванні документа великих розмірів із складним форматуванням бажано дотримуватися певної послідовності дій. Отже, далі пропонуємо орієнтовний алгоритм опрацювання складного текстового документа. В даному алгоритмі при роботі з різними документами деякі пункти можуть бути пропущені, місцями змінені або доповнені в залежності від початкового документа, який необхідно опрацювати і якого результату необхідно досягти.

Алгоритм створення складного документа.**1. Виконати налаштування параметрів сторінки.**

Відкрийте діалогове вікно **Параметри сторінки** або оберіть на вкладці **Макет** відповідний інструмент у групі **Параметри сторінки**.

2. Очистити текст документа від надлишкового форматування.

Для цього необхідно виділити весь текст та у вікні **Стилі** обрати команду **Очистити все**.

3. Позбутися від зайвих розривів рядків, абзаців, пробілів тощо.

Вилучення зайвих символів слід виконати у вікні **Пошук і заміна**, обравши відповідний інструмент на вкладці **Основне**.

4. Установити мову перевірки правопису.

Для цього на вкладці **Рецензування** слід обрати інструмент **Мова**. Після чого в меню інструмента обрати команду **Вибір мови перевірки**. У діалоговому вікні, що відкриється, слід обрати необхідну мову, наприклад, українську.

5. Перевірити правопис.

Перевірити документ на наявність помилок можна, обравши інструмент **Правопис і граматика** на вкладці **Рецензування**. Після чого у вікні, що з'явилося, діяти за вказівками майстра перевірки орфографії. Після того, як в документі прибрано зайві формати, символи помилки, можна приступити до його форматування.

6. Установити стиль для основного тексту та застосування його до всього документа.

Для цього у вікні **Стилі** слід обрати стиль **Звичайний** та виконати його налаштування.

7. Визначити структуру заголовків документа, надавши їм відповідні стилі.

Для цього слід виділити в документі заголовки та обрати відповідні стилі у вікні **Стилі**.

8. Виконати форматування окремих частин документа.

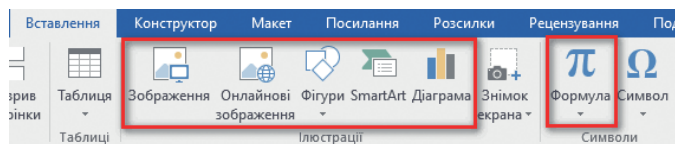
Це може бути виділення цитат, термінів, окремих абзаців тощо. Виконати це можна з використанням безпосереднього форматування, обравши необхідний інструмент в області **Шрифт** або **Абзац** на вкладці **Основне**. Також виконати форматування можна, надавши окремим частинам документа необхідні стилі.

9. Створити списки.

Наступним кроком, у разі необхідності, може бути створення списків. Нагадаємо, що створення та налагодження списків здійснюється з використанням інструментів **Маркери**, **Нумерація**, **Багаторівневий список** та команд меню відповідних інструментів.

10. Додати графічні об'єкти, формули тощо.

Далі, в разі необхідності, в документ додають графічні об'єкти, формули тощо. Для додавання зображень, автофігур, об'єктів SmartArt та діаграм використовують відповідні інструменти з області **Ілюстрації**, що розташована на вкладці **Вставлення**. Також на цій вкладці є інструмент **Формула**, з допомогою якого в документ додають формули.





11. Виконати форматування графічних об'єктів.

Наступний крок — форматування графічних об'єктів, яке здійснюється з використанням вкладки **Формат**, що з'являється після виділення графічного об'єкта. На цій вкладці ви можете змінити фігуру, її стиль, колір, надати різноманітні ефекти тощо.

12. Виконати оформлення титульної сторінки.

Наступним кроком, у разі необхідності, може бути оформлення титульної сторінки, що передбачає додавання, редагування та форматування тексту, зображень та інших об'єктів.

13. Додати колонтитули та нумерацію сторінок.

Це здійснюється на вкладці **Вставлення** з використанням інструментів **Верхній колонтитул**, **Нижній колонтитул** та **Номер сторінки**.

14. Додати зміст та предметний покажчик.

Ці дії виконуються на вкладці **Посилання** з використанням відповідних інструментів.

Виконання наступної практичної роботи спиратиметься на запропонований вище алгоритм.

§20.

Практична робота №1. Створення складного документа

У даній практичній роботі ви відпрацюєте вміння та навички роботи зі складним документом. Для цього відкрийте файл **ПР_1.docx** та виконайте кроки алгоритму, описані в §19, відповідно до вимог, указаних нижче. Зразок документа можна переглянути у файлі **ПР_1_зразок.pdf**.

1. Установіть усі поля сторінки по 1 см, а лівого поля — 2 см.
2. Очистіть текст документа від надлишкового форматування.
3. Позбавтеся від зайвих розривів рядків, абзаців, пробілів.
4. Установіть українську мову перевірки правопису.
5. Перевірте правопис.
6. Установіть стиль для основного тексту та застосуйте його до всього документа. *Параметри стилю*: гарнітура — Calibri, розмір — 14 пт., відступ абзацу — 1 см, міжрядковий інтервал — одинарний, вирівнювання — по ширині.
7. Установіть параметри стилів для заголовків.
Параметри заголовків 1-го рівня: гарнітура — Arial, розмір — 16 пт., накреслення — напівжирне, відступ абзацу — 1 см, колір — темно-синій.
Параметри заголовків 2-го рівня: гарнітура — Arial, розмір — 15 пт., накреслення — напівжирне, відступ абзацу — 1 см, колір — синій.

8. Виконайте форматування окремих частин документа.
 - a. Створіть власний стиль **Name** та застосуйте його до всіх імен, прізвищ та по батькові, що наявні в документі.
Параметри стилю: гарнітура — Arial, розмір — 13 пт., накреслення — напівжирне, колір — темно-червоний.
 - b. Створіть власний стиль **Number** та застосуйте його до всіх чисел, що наявні в документі. *Параметри стилю:* накреслення — напівжирне, підкреслене, колір — коричневий.
 - c. Створіть власний стиль **Title** та застосуйте його до всіх назв обчислювальних приладів. *Параметри стилю:* колір — темно-червоний.
 - d. Виконайте підписи зображень. Для підписів створіть стиль **Picture**. *Параметри стилю:* гарнітура — Times New Roman, розмір — 10 пт., колір — зелений.
9. Виконайте форматування зображень та титульної сторінки відповідно до файла-зразка.
10. Додайте колонтитули, нумерацію сторінок, зміст та предметний покажчик відповідно до файла-зразка.
11. Збережіть файл.



Розгадайте
електронний
кресворд



Висновки

Пошук та заміна фрагментів тексту здійснюється засобами команд **Пошук** та **Замінити**.

Стиль — це іменованій набір параметрів форматування, які можна застосувати до символів, абзаців, таблиць та списків.

Колонтитули на номери сторінок встановлюються з використанням інструментів **Верхній колонтитул**, **Нижній колонтитул** та **Номер сторінки**, розташовані на вкладці **Вставлення**.

Структуру великого документа створюють з використанням вбудованих стилів заголовків **Заголовок 1**, **Заголовок 2** і т. д. Основному тексту зазвичай надають стиль **Звичайний**.

У MS Word можна автоматично створювати зміст документа за умови, якщо для заголовків застосовано вбудовані стилі заголовків.

У MS Word є можливість створювати внутрішні та зовнішні гіперпосилання. Внутрішні гіперпосилання посилаються на місце в цьому ж документі, зовнішні дозволяють відкривати інші документи або вебсторінки.

При створенні складних документів слід дотримуватися певної послідовності дій, що умовно поділяються на такі частини: очищення документа від надлишкового форматування; редагування документа; форматування документа; додавання та форматування інших структурних елементів.

Розділ 3



Створення та публікація вебресурсів



§21.

Автоматизовані засоби для створення та публікації вебресурсів

21.1.

ОГЛЯД АВТОМАТИЗОВАНИХ ЗАСОБІВ

На сьогоднішній день школярів 8-го класу можна вже вважати професійними користувачами мережі Інтернет. Ви вправно вмiєте користуватися багатьма сервісами цієї мережі. Неодмінно знайомим для вас є сервіс World Wide Web (WWW), який дозволяє одержувати доступ до вебсторiнок з допомогою браузера на комп'ютері. Подальше вивчення ми присвятимо методам створення вебсторiнок.

Для створення вебсторiнок iснують принаймні два пiдходи. Сторiнки можна створювати з використанням сервісів **автоматизованого створення сайтiв** або засобами **мови гiпертекстової розмiтки та вебпрограмування**. Перший спiсiб використовують, коли вимоги до сайту є не високими. У разі специфічних умов до сайту вдаються до другого способу. Розглянемо перший спiсiб створення вебсторiнок.

Iснує багато онлайн-систем для автоматизованого створення сайтiв. Ось посилання лише на деякі з них: **wix.com**, **www.ucoz.ua**, **www.jimdo.com**, **www.hostinger.com.ua**. Усі вони надають широкі можливості для створення вебсайтiв з використанням як безкоштовних, так і платних складових.

Подiбні можливості щодо створення власного сайту має також додаток Google Sites. У цьому уроці ми розглянемо створення сайту саме з його допомогою.

21.2.

СТВОРЕННЯ ВЕБСТОРИНКИ

Створимо вебсторiнку деякої туристичної фiрми **BestTravel**.



Завантажте для ознайомлення зразок



1. Для створення вебсторінки попередньо завантажте на комп'ютер файли зображень для даного уроку.
2. Увійдіть до власного акаунта Google та відкрийте сторінку sites.google.com.
3. Для створення нового сайту натисніть значок плюс в області **Створити новий сайт**, обравши шаблон **Пустий**.

У результаті ви потрапите на сторінку створення та редагування сайту.

Спочатку розглянемо створення верхньої частини вебсторінки. У цьому додатку він має назву **Верхній колонтитул**. У результаті ми маємо одержати такий колонтитул, як на рисунку нижче.

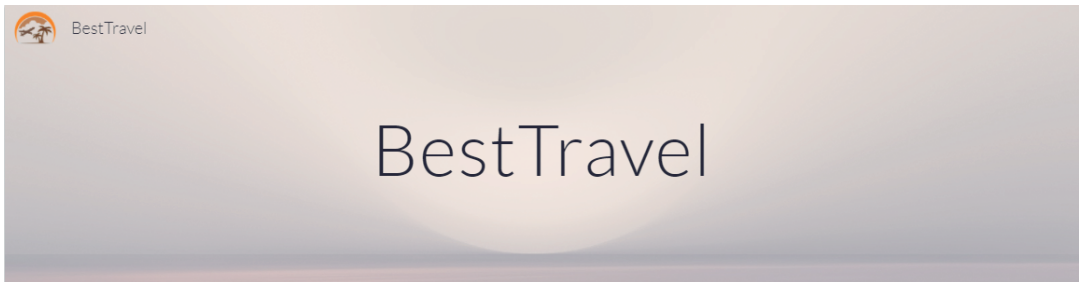


Рис. 21.1. Верхній колонтитул сайту

4. Уведіть заголовок сайту **BestTravel**. Для цього клацніть на область заголовка та введіть необхідний текст.
5. Додайте логотип.
 - a. Наведіть на верхній правий кут шаблону.
 - b. Оберіть посилання **Додати логотип** ▶ **Додати**.
 - c. Оберіть файл **logo.png** на комп'ютері.
 - d. Установіть прозорий колір фону.
6. Додайте назву сайту. Для цього клацніть на полі **Введіть назву сайту** та введіть **BestTravel**.
7. Додайте фон для верхньої частини сторінки.
 - a. Наведіть на верхню частину сторінки та оберіть команду **Змінити зображення**.
 - b. Оберіть команду **Вибрати зображення**.
 - c. Оберіть зображення відповідно до зразка (рис. 21.1)

Розглянемо, що ще можна зробити з верхньою частиною сторінки.



Команда **Скинути** дозволяє повернутися до попереднього зображення. Команда **Тип верхнього колонтитула** дає змогу змінити тип колонтитула.



8. Установіть тип верхнього колонтитула — Банер.

У нижньому правому куті верхнього колонтитула розташовано посилання **Покращення читабельності**. При вимкненні даного режиму фон набуде початкової яскравості. Увімкнення даного режиму робить фон світлішим, шрифт — темнішим.

9. Увімкніть режим покращення читабельності.

В режимі конструювання сайту ліворуч біля усіх складових є інструмент у вигляді кошика, натискання на який дозволяє вилучити ту чи іншу структурну частину.

Під колонтитулом додамо розділ **Гарячі тури** відповідно до зразка (рис. 21.2).

Гарячі тури




		
Готель "Gagarinn", Україна	Готель "Кемер", Туреччина	Готель "Порто-Крісто", Іспанія
01.09-10.09 - 5000 грн. за особу	03.09-12.09 - 15000 грн. за особу	05.09-15.09 - 17000 грн. за особу
Замовити	Замовити	Замовити

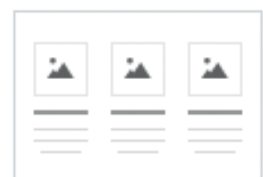
Рис. 21.2. Блок Гарячі тури

10. Розташуйте блок заголовка в даному розділі.

- a. Перейдіть на вкладку **Вставити**.
- b. Оберіть інструмент **Текстове поле**.
- c. Оберіть тип текстового поля **Назва** та встановіть вирівнювання по центру.
- d. Уведіть текст **Гарячі тури**.
- e. Установіть фон розділу **Виділення 1**.

11. Додайте блок з фотографіями та текстом.

- a. В області **Макети** оберіть наступний макет:
- b. Для додавання зображення натисніть знак плюс та команду **Додати**.
- c. Оберіть зображення **gagarinn.jpg** з файла на диску.



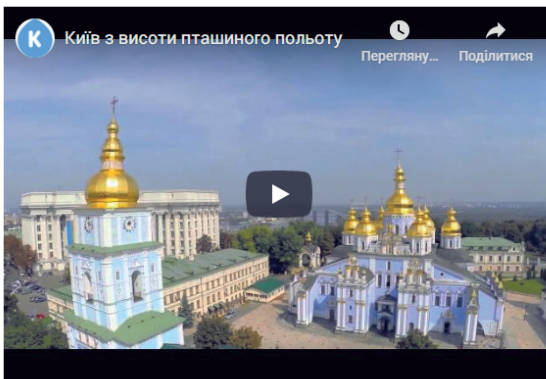


- d. Додайте необхідний текст у відповідні області.
 - e. Аналогічно додайте інші зображення та текст.
12. Додайте кнопки **Замовити**.
- a. Оберіть інструмент **Кнопка**.
 - b. Уведіть ім'я кнопки **Замовити**.
 - c. Як посилання на місце переходу по кнопці вкажіть довільний символ, оскільки зараз місце переходу за кнопкою ще не визначено.
 - d. Натисніть **Вставити**.

У результаті кнопку буде додано, але не після блоків з гарячими турами.

- e. Створіть ще дві копії кнопки та перенесіть їх, утримуючи ліву кнопку миші, в необхідні місця блоків з гарячими турами.

Нижче додамо розділ з відеофрагментом про Київ та його описом відповідно до зразка (рис. 21.3).



Київ - столиця України

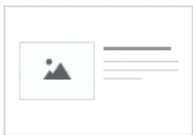
У цьому відео-ролику ви познайомитеся з деякими пам'ятками перлини туристичної України - містом Київ.

Київ - це воістину дивовижне місто. Воно багатогранне, цікаве, яскраве і неповторне.

Ось лише невеликий перелік **найпопулярніших місць Києва**:

- Софійський собор
- Києво-Печерська лавра
- Успенський собор
- Володимирська гірка
- Фунікулер
- Андріївський узвіз

Рис. 21.3. Блок з відео та його описом





13. На вкладці **Вставити** оберіть необхідний макет.
14. До першого блока додайте відео.
- a. Натисніть знак плюс у першому блоці та оберіть команду **YouTube**.
 - b. У відповідне вікно додайте посилання на відео:
<https://youtu.be/dvW1IVvo-Z0>.
 - c. Натисніть значок пошуку. В результаті необхідне відео має бути знайдено.
 - d. Натисніть на знайденому відео, а потім на кнопці **Вибрати**. Обране відео має з'явитися на вашому сайті.



15. Додайте в текстові блоки написи, як у зразку. Відшукайте в Інтернеті ще принаймні п'ять визначних місць Києва та додайте їх до списку.
16. Виконайте необхідне форматування текстових блоків (змінить у деяких фрагментів жирність шрифту та встановіть стиль списку для переліку визначних місць).
17. Для розташування сайту в мережі Інтернет його необхідно опублікувати. Для цього натисніть кнопку **Опублікувати**, введіть вебадресу. (Якщо така адреса вже існує, то система повідомить про це.) Натисніть кнопку **Опублікувати**.
18. Для перегляду сайту в режимі публікації оберіть в меню кнопки **Опублікувати** команду **Переглянути опублікований сайт**.

У режимі перегляду сайт уже стає інтерактивним. Тут спрацьовує посилання на кнопки. Звичайно, воно поки нікуди вас не приведе, адже ви не задавали для них конкретні дії. Також здійснюється завантаження відео.

Аби сайт був доступний не лише вам як автору, але й іншим користувачам мережі Інтернет, до нього необхідно надати доступ, а також отримати посилання, за яким цей сайт можна переглядати. Для цього призначені відповідні інструменти:

-  — надати доступ;
-  — скопіювати посилання.

19. Надайте доступ до сайту для загального доступу. Надішліть на пошту вашому вчителю посилання на створений сайт.



Контрольні запитання та завдання

1. Які існують підходи до створення сайтів?
2. Які сервіси для автоматизованого створення сайтів ви знаєте?



Питання для роздумів

1. Для чого існує багато різних сервісів автоматизованого створення сайтів?
2. Чому існують безкоштовні сервіси автоматизованого створення сайтів?



Завдання для досліджень

- 1*. Дослідіть, чи можна у сервісі Google Sites створити багатосторінковий сайт.
- 2*. Самостійно ознайомтеся з об'єктами, які можна ще додати до сайту в сервісі **Google Sites**. Визначте, коли вони можуть стати у нагоді.
- 3*. Дослідіть, яких, на вашу думку, можливостей зі створення вебсторінок не вистачає у сервісі Google Sites.
- 4**. Зареєструйтесь на одній із онлайн-ових систем автоматизованого створення сайтів та порівняйте її можливості із можливостями сервісу Google Sites.

§22.

Мова гіпертекстової розмітки

22.1.

ПОНЯТТЯ ПРО МОВУ HTML

У попередньому параграфі було вказано, що існують два підходи для створення вебсторінок. З першим, автоматизованим способом, ми вже познайомилися, розглянемо другий спосіб.

Мова гіпертекстової розмітки HTML розшифровується, як **Hyper Text Markup Language** — це мова розмітки гіпертексту для подальшого його відображення у вікні браузера. Являє собою сукупність тегів та правил їхнього застосування.

Тегом називається так званий код мови і являє собою службове слово, записане в кутових дужках `< >`. Наприклад, `<html>`.

Правила запису тегів

1. Теги записують латинськими літерами.
2. Регістр у тегах не розрізняється.
3. Існують парні та непарні теги. Парні теги містять відкриваючий і закриваючий теги. Закриваючий тег починається символом `</>` (слеш).

Наприклад, парні теги — `<html></html>`; непарні теги `
`, `<hr>`.

22.2.

СТРУКТУРА HTML-ДОКУМЕНТА

Для правильного відображення браузером вебсторінки слід дотримуватися певної структури html-документа (рис. 22.1):



- `<html>` — початок документа;
- `<head>` — заголовок документа;
- `<title></title>` — назва, що відображається у вкладці браузера.

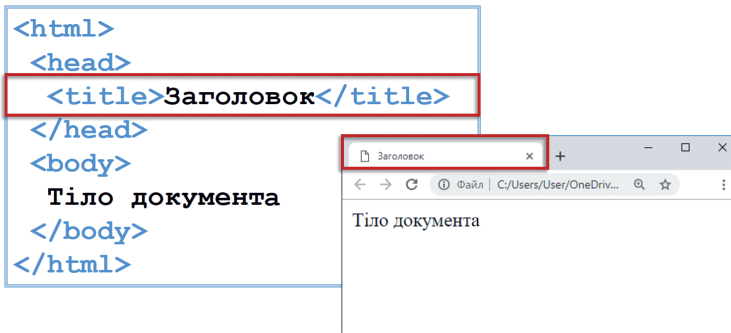


Рис. 22.1. Структура html-документа та відповідне відображення в браузері

- `</head>` — кінець заголовка документа;
- `<body></body>` — тіло документа, його основна частина. Тут міститься все те, що відображається на вебсторінці (рис. 22.2).

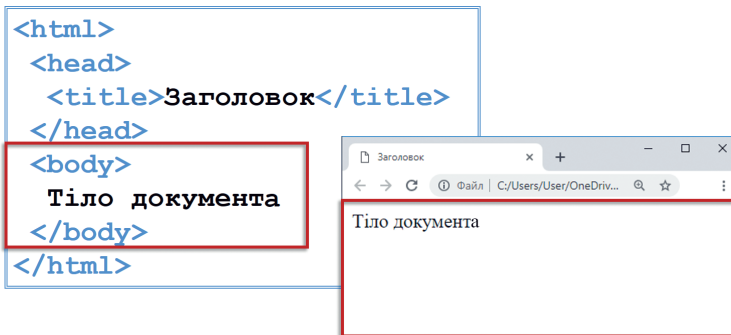


Рис. 22.2. Позначення тіла документа в html-кодi та у вікні браузера

- `</html>` — кінець документа.

22.3.

СТВОРЕННЯ ВЕБСТОРИНКИ

Для створення вебсторінок існує безліч різноманітних редакторів (Notepad++, Brackets, VS Code тощо). Проте в якості редактора html-коду підійде і звичайний блокнот. Подальші пояснення ми будемо здійснювати з використанням редактора NotePad++. Цей редактор є зручним у використанні, також його безкоштовно можна скачати з мережі Інтернет.



Алгоритм створення вебсторінки

1. Відкрити редактор коду для введення html-коду.
2. Увести html-код.

Це може бути найпростіший html-код з дотриманням загальної структури документа (рис. 22.3).

Аби браузер міг відтворити створений код, необхідно зберегти документ у форматі, який може сприймати браузер, а саме: у форматі html. Як правило перша сторінка сайту має назву **Index.html**.

3. Зберегти файл. Для цього слід обрати команду **Зберегти як...** та ввести ім'я **Index.html**.

```
1 <html>
2   <head>
3     <title>Перша спроба</title>
4   </head>
5   <body>
6     Моя перша вебсторінка
7   </body>
8 </html>
```

Рис. 22.3. Зразок відображення коду у вікні редактора після його збереження у форматі html

Також у результаті збереження файла його піктограма набуде вигляду, встановленого за умовчанням на вашому комп'ютері браузера.

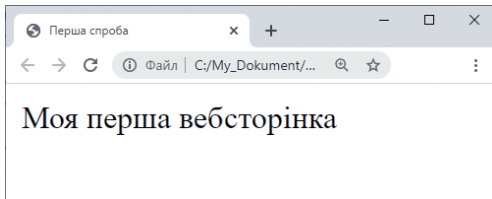


Рис. 22.4. Відображення вебсторінки у вікні браузера

Зауваження. Після збереження файла у вікні редактора коди документа будуть структуровані та підсвічені різними кольорами (рис. 22.3). Це дуже зручно, особливо при роботі з великими файлами. Отже, аби підсвічення кодів відбувалося під час уведення та редагування коду, можна файл зберегти до початку введення коду з форматом html.

4. Відкрити файл у вікні браузера (рис. 22.4).

У разі необхідності переглянути код вебсторінки, в контекстному меню вікна браузера оберіть команду **Переглянути джерело сторінки**. В результаті буде відображено html-код цієї сторінки.

ПРАКТИЧНЕ ЗАВДАННЯ 1



Завантажте для ознайомлення зразок готової вебсторінки



Під час ознайомлення з мовою HTML створимо вебсторінку BestTravel, подібну до тієї, яку ми робили в автоматичному режимі.

Розпочнемо створення сайту з найпростішого. Додамо до сторінки необхідний текст та збережемо файл під назвою **Index.html**.

1. Відкрийте редактор NotePad++.
2. Збережіть файл з назвою **Index.html**. Це потрібно для того, щоб редактор сприймав документ як вебсторінку і підсвічував кольором теги документа.



- Уведіть у файлі загальний код html-документа.

```
<html>
  <head>
    <title></title>
  </head>
  <body>
  </body>
</html>
```

- Додайте текст вебсторінки між тегами <body>...</body>, скопіювавши його з файла **заготовка.txt**.
- Між тегами <title>...</title> введіть заголовок вебсторінки **BestTravel**.
- Збережіть документ.
- Відкрийте документ у вікні браузера. В результаті вебсторінка має містити наданий текст. Але, звичайно, до одержання бажаного результату ще далеко.

22.4.

ТЕГИ ДЛЯ СТВОРЕННЯ СТРУКТУРИ ДОКУМЕНТА

Продовжимо знайомство з тегами. І далі розглянемо, як з допомогою тегів установити заголовки та абзаци документа. В мові HTML виділяють усього 6 рівнів заголовків, що позначаються відповідно тегами від <h1>...</h1> до <h6>...</h6>. У браузері заголовок першого рівня буде відображено найбільшим, а заголовок шостого рівня — найменшим. На рис. 22.5 показано відображення заголовків у вікні браузера та відповідний html-код.

Для встановлення абзців використовують тег <p>...</p>. Для налаштування параметрів розміщення тексту та структурування документа використовують непарні теги:
 — для розриву рядка; <hr> — для встановлення горизонтальної лінії.

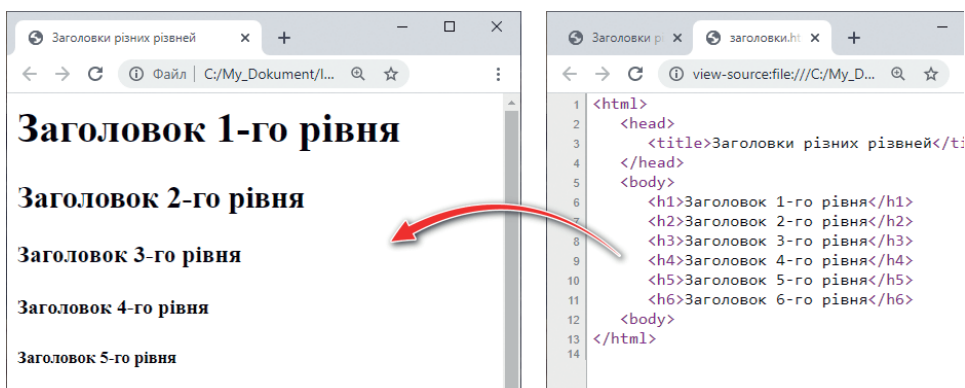


Рис. 22.5. Відображення заголовків у вікні браузера та відповідний html-код



ПРАКТИЧНЕ ЗАВДАННЯ 2



Додайте до вебсторінки BestTravel теги для структурування документа.

1. Перший рядок не є заголовком, отже, його позначте тегом `<p>`, уклавши, що він є окремим абзацом:

```
<p>BestTravel</p>
```

2. Наступний рядок, назва сайту, безперечно є заголовком, і він є найвищого рівня, отже, позначте його заголовком першого рівня:

```
<h1>BestTravel</h1>
```

3. Відповідно заголовок **Гарячі тури** позначте заголовком другого рівня.
4. Заголовки, що містять назву готеля та країну, позначте заголовками 3-го рівня.
5. Заголовок **Київ — столиця України** також позначте як заголовок 3-го рівня.
6. Усі інші частини тексту позначте тегами абзацу.
7. Перелік визначних місць Києва відокремте тегами розриву рядка, а саме,

```
<br>
```

```
<br>Софійський собор
```

```
<br>Києво-Печерська лавра
```

```
...
```

22.5.

АТРИБУТ ВИРІВНЮВАННЯ ТЕКСТУ

Для вирівнювання тексту по лівому та правому краю, по центру та по ширині використовують атрибут `align`. Він може приймати значення `left`, `right`, `center` та `justify` відповідно. Атрибути записують відразу після тегу, для якого його необхідно застосувати. Значення атрибута записують у лапках.

Наприклад, якщо необхідно вирівняти деякий заголовок першого рівня по центру, слід записати наступний код:

```
<h1 align="center"> Заголовок </h1>
```

Для вирівнювання абзацу по ширині код буде такий:

```
<p align="justify"> Текст абзацу </p>
```

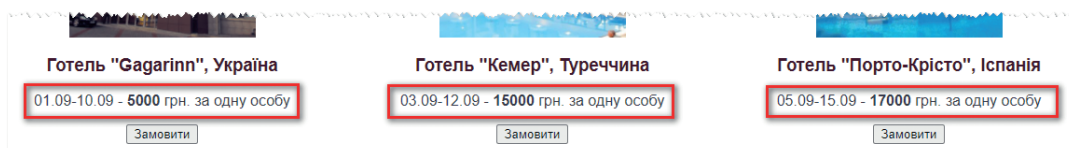
ПРАКТИЧНЕ ЗАВДАННЯ 3



1. Виконайте вирівнювання усіх заголовків у сайті BestTravel по центру.



- a. Для вирівнювання по центру першого заголовка додайте атрибут вирівнювання тексту, аби код набув такого виду:
`<h1 align="center"> BestTravel </h1>`.
 - b. Для вирівнювання інших заголовків атрибути додайте самостійно.
2. Додайте атрибути вирівнювання по центру абзаців у розділі опису гарячих турів.



3. Збережіть файл.
4. Виконайте оновлення документа у вікні браузера, натиснувши **F5**.

22.6.

ВСТАНОВЛЕННЯ КОЛЬОРУ ФОНУ ТА ШРИФТУ

За умовчанням сторінка має білий колір фону та чорний колір літер. Проте ці значення можна змінити. Для цього використовують атрибути для тега `<body>`.

Так, при встановленні чорного кольору фону та білих літер тег `<body>` буде мати вигляд:

```
<body bgcolor = "#000000" text = "#FFFFFF">
```

Тут `bgcolor` — атрибут для встановлення кольору фону, `text` — атрибут для встановлення кольору шрифту. `#000000` та `#FFFFFF` — значення цих атрибутів, які визначають чорний та білий кольори відповідно. Взагалі кольори можна встановлювати і звичними для нас словами англійської мови `black`, `white` тощо. Але перелік цих слів досить обмежений.

Для визначення коду бажаного кольору можна скористатися кольоровою онлайн-палітрою. Ось один із таких сайтів, що надають можливість визначити код кольору: <https://html-color-codes.info/>.

ПРАКТИЧНЕ ЗАВДАННЯ 4



1. Додайте до вебсторінки атрибути для встановлення чорного кольору фону та білого кольору літер.
 - a. У редакторі коду додайте атрибут кольору фону `bgcolor` та встановіть чорний колір.



- b. Поряд через пробіл додайте атрибут `text` та надайте йому значення білого кольору:
- ```
<body bgcolor = "#000000" text = "#FFFFFF">
```
- c. Збережіть внесені зміни та оновіть вебсторінку у вікні браузера. В результаті колір фону і літер має бути змінено.
2. Ще раз виконайте зміни в атрибутах тегу `<body>`, встановивши ті значення кольорів, що є реальними у сайті.
- a. Для фону поверніть білий колір. Для цього приберіть атрибут `bgcolor`.
- b. Для літер установіть темно-сірий колір літер, змінивши код чорного кольору на код темно-сірого кольору (`#333333`).



Виконайте вправу 22



### Контрольні запитання та завдання

1. Що таке тег? Які використовуються правила для запису тегів?
2. Укажіть основні теги, що має містити html-документ.
- 3\*. Опишіть алгоритм створення вебсторінки.
4. Які теги використовуються для створення заголовків; абзаців; розриву рядка; горизонтальної лінії?
5. Як виконати вирівнювання абзацу в html-документі?
6. Як змінити колір тексту та фону в html-документі?



### Питання для роздумів

1. Для чого використовується два різних теги для створення абзацу та розриву рядка?
2. Чому при створенні html-документа необхідно дотримуватися вкладеності тегів один в одній?



### Завдання для дослідження

- 1\*. Дослідіть можливості онлайн-редакторів html-коду, визначте їхні відмінності з редакторами, встановленими на комп'ютері.



Пройдіть  
тест



## §23.

## Форматування символів

### 23.1.

### ТЕГИ НАКРЕСЛЕННЯ

Для надання шрифту документа привабливого вигляду його піддають форматуванню. В мові HTML для цього також є спеціальні теги.

Для зміни накреслення шрифту використовують теги:

- `<b>...</b>` — напівжирний шрифт;
- `<i>...</i>` — курсив;
- `<u>...</u>` — підкреслення.

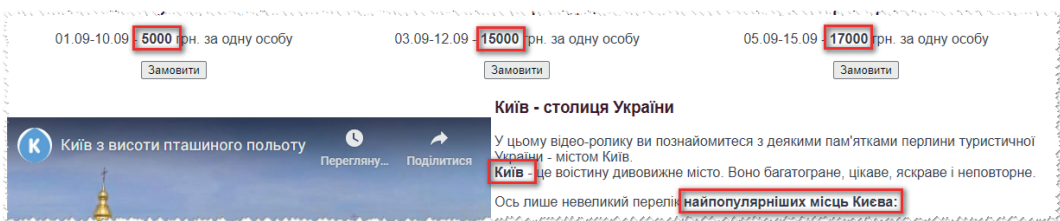
Існують випадки, коли до фрагментів тексту слід застосувати два або відразу три типи накреслення. Наприклад, зробити шрифт напівжирним курсивом або підкресленим курсивом. Для цього використовують обидва теги, наприклад: `<b><i>текст</i></b>`. У цьому випадку слід пам'ятати правило вкладених тегів, відповідно до якого їх потрібно закривати у зворотному до відкриття порядку. Тобто якщо спершу відкрити тег курсиву, а потім тег підкреслення, то спочатку слід закривати тег підкреслення, а потім тег курсиву.

### ПРАКТИЧНЕ ЗАВДАННЯ 1



Додайте в сайті BestTravel необхідні теги накреслення.

З рисунка видно, що у блоці **Гарячі тури** слід виділити напівжирним вартості турів, а в останньому блоці напівжирним шрифтом виділено окремі фрагменти тексту.



1. Додайте теги `<b>...</b>` для вартості турів, до фрагмента «Київ» та фрагмента «найпопулярніших місьць Києва:»
2. Збережіть файл та відкрийте у вікні браузера. В результаті позначені фрагменти мають стати напівжирними.



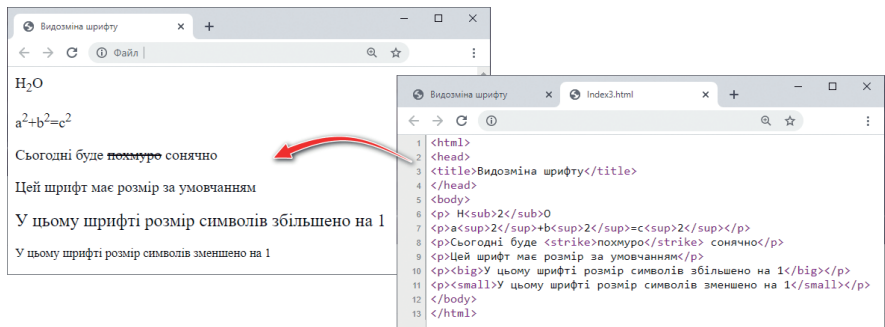
## 23.2.

## ТЕГИ ВИДОЗМИНИ ШРИФТУ

Розглянемо ще ряд тегів, які дозволять змінити зовнішній вигляд шрифту:

- `<sub></sub>` — підрядковий текст;
- `<sup></sup>` — надрядковий текст;
- `<strike></strike>` — перекреслення тексту;
- `<big></big>` — збільшення розмірів шрифту на 1;
- `<small></small>` — зменшення розмірів шрифту на 1.

На рисунку 23.1 наведено приклад html-коду з використанням цих тегів та відповідне відображення цього коду у вікні браузера.



**Рис. 23.1.**  
Приклад використання тегів видозміни шрифту

## 23.3.

## ЗМІНА ГАРНІТУРИ ШРИФТУ, КОЛЬОРУ ТА РОЗМІРУ

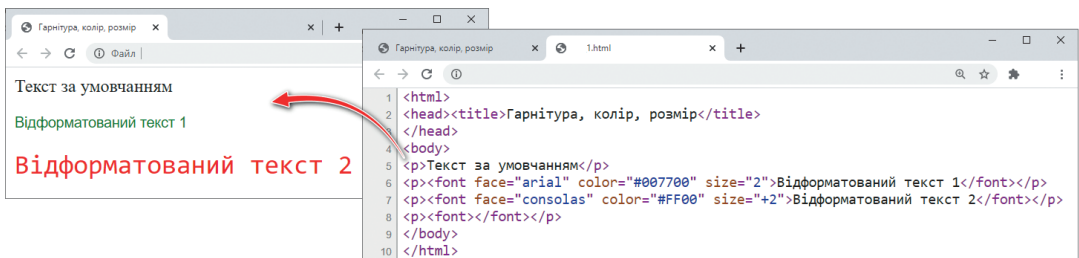
Розглянемо ще один тег, який дозволяє змінити такі параметри, як гарнітуру шрифту, колір та розмір. Усі ці параметри можна встановити за допомогою атрибутів тегу `<font>`.

Він має таку конструкцію:

```
текст
```

- `face` — гарнітура шрифту;
- `color` — колір шрифту;
- `size` — розмір шрифту.

На рисунку наведено деякі зразки кодів та відповідне їхнє відображення у вікні браузера.



**ПРАКТИЧНЕ ЗАВДАННЯ 2**

Змініть необхідні параметри шрифту в сайті BestTravel.

1. Установіть колір заголовків.
  - a. Додайте до першого заголовка тег `<font>`.
  - b. Додайте до тегу `<font>` атрибут `color` зі значенням `#420000`.
2. Змініть гарнітуру всього шрифту, встановивши значення **Arial**. Оскільки гарнітуру слід змінити для всієї сторінки, то атрибут `face` додайте для тегу `<body>`.
3. Збережіть файл та відкрийте у вікні браузера. В результаті позначені фрагменти мають змінити колір шрифту та гарнітуру.



Виконайте вправу 23

**Контрольні запитання та завдання**

1. Як змінити накреслення шрифту?
2. Як зробити шрифт курсивом та підкресленим; напівжирним та підкресленим; напівжирним та курсивом?
3. З допомогою яких тегів можна зменшити або збільшити шрифт?
4. Як змінити гарнітуру шрифту; колір шрифту; розмір шрифту?

**Питання для роздумів**

1. Чи можна для створення верхніх та нижніх індексів використовувати теги збільшення та зменшення шрифту?
2. Де на практиці можна використовувати теги видозміни шрифту? Наведіть приклади.
3. Що краще: використовувати для зміни розміру шрифту теги видозміни шрифту чи атрибут `size` для тегу `<font>`? Відповідь обґрунтуйте.

**Завдання для дослідження**

- 1\*. З допомогою яких html-кодів можна відформатувати текст наступним чином:
  - a)  $a^2 + b^2 = c^2$ ;
  - б)  $\text{Na}_2\text{O} + \text{H}_2\text{O} = 2\text{NaOH}$ ?



Пройдіть  
тест



## §24.

## Списки

Для представлення деякого переліку даних використовують списки. Розглянемо, як їх створити з використанням мови гіпертекстової розмітки. З використанням мови HTML можна створити два види списків: нумеровані та марковані. Нумеровані списки використовують, коли порядок елементів має значення. Коли порядок елементів не є принциповим, використовують марковані списки.

## 24.1.

## МАРКОВАНІ СПИСКИ

Для створення маркованого списку використовують тег `<ul></ul>`. Для цього тегу можливе використання атрибута `type`, який визначає тип маркера.

Даний атрибут може приймати значення:

- `disk` — зафарбовий круг;
- `square` — зафарбований квадрат;
- `circle` — коло.

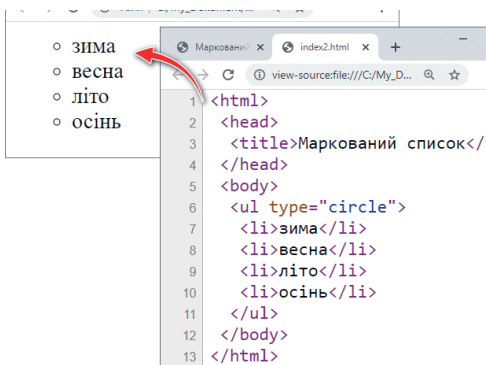
За умовчанням маркований список має тип `disk`.

Відокремлення елементів списку здійснюється з використанням тегу `<li></li>`.

Маркований список має таку загальну структуру:

```
<ul type = значення>
 Перший елемент списку
 Другий елемент списку
 ...
 Останній елемент списку

```



На рис. 24.1 наведено приклад html-коду для створення маркованого списку та відповідне його відображення у вікні браузера. З рисунка видно, що до даного списку застосовано тип маркерів `circle`.

Рис. 24.1. Приклад маркованого списку



### ПРАКТИЧНЕ ЗАВДАННЯ 1



Додайте в сайті BestTravel теги для утворення списку переліку визначних місць Києва. До цього ці написи було відокремлено тегом `<br>`, який просто розташовував їх один під одним.

1. Перед списком додайте тег `<ul>` та встановіть атрибут `type`, якому надайте значення `square`, адже маркерами списку мають бути зафарбовані прямокутники.
2. Установіть відповідний тег `</ul>` в кінці списку.
3. Замість тегу `<br>` перед кожним елементом списку додайте тег `<li>`, закрийте елемент списку тегом `</li>`.
4. Збережіть файл та перегляньте його у вікні браузера. В результаті ви маєте одержати список із квадратними маркерами.

### 24.2.

### НУМЕРОВАНІ СПИСКИ

Для створення нумерованого списку використовують тег `<ol>`. Для цього тегу використовують атрибути `type`, `start` та `reversed`.

Атрибут `type` визначає тип номера.

Даний атрибут може приймати значення:

- "A" або "a" — список буде мати відповідно великі або малі латинські літери (A, B, C, D і т. д. та a, b, c, d і т. д.);
- "I" або "i" — список матиме відповідно великі або малі римські цифри;
- "1" — звичайні арабські цифри. За умовчанням список буде використовувати саме арабські цифри.

Атрибут `start` використовують для визначення початкового номера списку. Атрибут `reversed` встановлює зворотну нумерацію для списку. Наприклад, 3, 2, 1. Для відокремлення елементів списку, як і у випадку маркованих списків, використовують тег `<li>`.

Нумерований список має таку загальну структуру:

```
<ol type = значення start = значення reversed>
 Перший елемент списку
 Другий елемент списку
 ...
 Останній елемент списку

```



Дізнайтеся про створення багаторівневих списків



Виконайте вправу 24



### Контрольні запитання та завдання

1. Які види списків можна створювати засобами мови HTML?
2. Укажіть атрибути та їхнє призначення для створення маркованого та нумерованого списків.
3. Наведіть приклади маркованих та нумерованих списків.



### Питання для роздумів

- 1\*. Як зробити у списку маркери одного кольору, а значення списку — іншого?



### Завдання для досліджень

- 1\*. Які атрибути необхідно встановити для створення нумерованого списку, що буде мати таку нумерацію: VII, VI, V, ...?
- 2\*. Чи можливо засобами мови HTML установити інші види маркерів, ніж ті, що були розглянуті в параграфі?



Пройдіть  
тест

## §25.

### Додавання зображень на вебсторінці

#### 25.1.

#### ФОН ВЕБСТОРИНКИ

Вебсторінка стає більш наочною та цікавою, якщо її доповнити зображеннями та графічними елементами. Для формування фону вебсторінки у вигляді зображення використовують атрибут `background` для тегу `<body>`.

```
<body background="URL"></body>
```

Тут URL — адреса файлу зображення.

Так з допомогою коду `<body background="fon.jpg"></body>` буде створено фон вебсторінки, заповнений зображенням (рис. 25.1). Тут `fon.jpg` — файл, що містить дане зображення. Зверніть увагу, що в даному випадку файл має бути розміщений у тій же папці, що і файл вебсторінки.



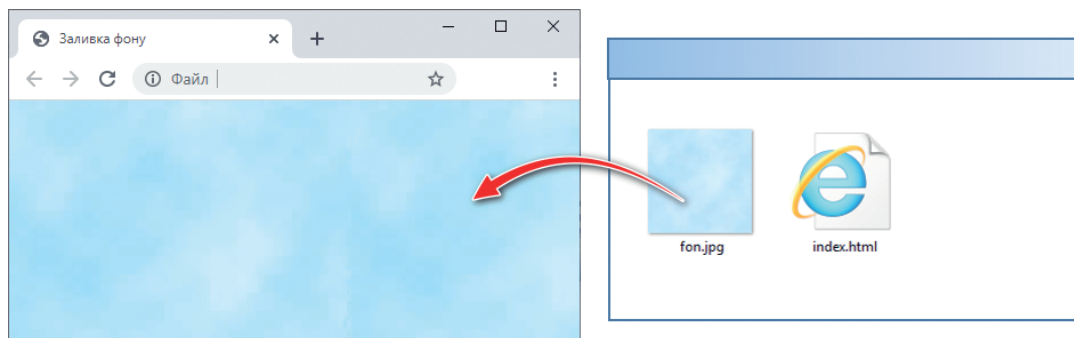


Рис. 25.1. Приклад вебсторінки, що в якості фону містить зображення

Якщо зображення міститься у деякій папці, наприклад, `img`, необхідно вказати повний шлях до файлу з урахуванням папки розташування. Наприклад, `img/1.jpg`.

### ПРАКТИЧНЕ ЗАВДАННЯ 1



Додайте фонове зображення до деякої вебсторінки.

1. Завантажте на комп'ютер папку `pz_1` для даного уроку. Ця папка має містити html-файл `index.html` та зображення фону `fon.png`.
2. Уведіть після тегу `<body>` атрибут `background="fon.png"`.
3. Збережіть файл та відкрийте його у вікні браузера. В результаті фон має бути заповнений зображенням `fon.png`.

## 25.2. ДОДАВАННЯ ЗОБРАЖЕНЬ

Для вставки зображень використовують тег `<img>`. Даний тег має обов'язковий атрибут `src`, значенням якого є адреса файлу зображення:

```

```

Так з допомогою коду `` буде додано зображення ноутбука (рис. 25.2). Тут `laptop.png` — файл, що містить дане зображення.

Зверніть увагу, що в даному випадку файл розміщений у тій же папці, що і файл вебсторінки. Якщо зображення буде розміщено у деякій папці, слід вказати повний шлях до файлу з урахуванням папки розташування.

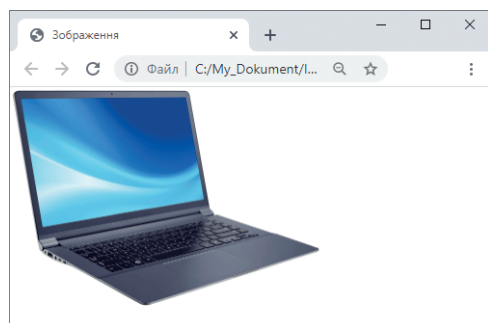


Рис. 25.2. Приклад вебсторінки з зображенням



### ПРАКТИЧНЕ ЗАВДАННЯ 2



Додамо необхідні зображення до сайту BestTravel. Тут мають бути зображення логотипу та готелів.

1. Розташуйте файли зображень у папці збереженого проєкту.
2. Уведіть тег додавання зображення логотипу:  
``.
3. Аналогічно введіть теги для додавання зображень готелів.
4. Збережіть файл та відкрийте його у вікні браузера. В результаті рисунки буде додано, але логотип буде мати дуже великі розміри.

### 25.3.

### РОЗМІРИ ЗОБРАЖЕНЬ

Для зміни розмірів зображень використовуються атрибути `width`, що визначає *ширину* зображення, та атрибут `height`, для встановлення *висоти* зображення.

Так, у коді `` зображення буде мати ширину 200 пікселів, а висоту 100 пікселів.

Але в цьому випадку розміри зображення не завжди є пропорційними. Якщо вказати лише один із розмірів, ширину або висоту, то другий вимір браузер обчислить самостійно, виходячи з пропорцій зображення.

Розмір зображення можна також задавати у відсотках. Це значення встановлюється відносно розмірів ширини та висоти сторінки браузера. Цей спосіб є зручним, коли необхідно, аби при зменшенні або збільшенні сторінки зображення пропорційно змінювало свій розмір. Наприклад, при використанні даного коду `>` ширина зображення буде мати 50% свого початкового розміру, тобто зменшена у два рази, відповідно його висота буде змінена пропорційно до зміни розмірів ширини.

### ПРАКТИЧНЕ ЗАВДАННЯ 3



Додайте атрибут ширини для логотипу сайту BestTravel.


1. Установіть значення ширини логотипу 50 пікселів, додавши атрибут `width="50"`. Значення висоти не вказуйте, аби зображення було зменшено пропорційно.
2. Збережіть html-код.
3. Відкрийте файл у вікні браузера. В результаті зображення логотипу має бути зменшено.

**25.4. ДОДАВАННЯ ВІДЕО. ПРАКТИЧНЕ ЗАВДАННЯ 4**

Додайте відео до вебсторінки.

Відео, яке нам необхідно додати, завантажено до сервісу YouTube.

Отже, ви маєте вставити посилання на дане відео.

1. Відкрийте сторінку за посиланням: <https://youtu.be/dvW1IVvo-Z0>.
2. Оберіть посилання **Поділитися**.
3. Клацніть на значок .

Вставити

У результаті на сайті буде сформовано готовий html-код з тегом `<iframe>` з різними атрибутами.

4. Скопіюйте в необхідне місце тег `<iframe>...</iframe>` до html-коду вашої сторінки.
5. Збережіть код та оновіть вікно браузера. В результаті відео буде розташоване на сторінці.



Виконайте вправи 25.1 та 25.2

**Контрольні запитання та завдання**

1. Як додати фонове зображення до вебсторінки?
2. Як додати до вебсторінки зображення?
3. В яких випадках у нагоді стане використання альтернативного тексту?
4. Які існують способи вирівнювання зображення відносно тексту?
5. Як установити відступи від зображень?
6. Як додати рамку до зображення?

**Питання для роздумів**

1. Де на практиці використовується виведення тексту при наведенні на зображення?
2. Для чого використовуються два способи встановлення розмірів зображень: у пікселях та у відсотках?

**Завдання для досліджень**

- 1\*. Чи можна при встановленні рамки зображення змінити її колір?
- 2\*. Чи можна засобами мови HTML вирівняти зображення по центру, аби текст обтікав його ліворуч та праворуч?
- 3\*\*. Дослідіть, як додати до вебсторінки відео із файла на диску.



## §26.

## Додавання гіперпосилань на вебсторінці

Для здійснення навігації в межах однієї сторінки або переходу на інші вебсторінки використовують гіперпосилання. Які існують види гіперпосилань і як їх додати на вебсторінку, розглянемо далі.

**Гіперпосилання** — це вказівка для браузера, до якого об'єкта у межах або поза межею html-документа він має звернутися.

## 26.1.

## АБСОЛЮТНІ ТА ВІДНОСНІ ПОСИЛАННЯ

Для додавання гіперпосилань використовують тег `<a>` з атрибутом `href`, значення якого є адресою посилання. Після атрибута `href` можна вказати посилання на будь-який інший сайт.

Якщо посилання містить повний шлях до сайту, його називають **абсолютним**. Такі посилання містять протокол, ім'я сервера та шлях.

```
 Текст посилання
```

протокол + ім'я сервера + шлях

Наприклад,



Якщо потрібно вказати посилання на деяку вебсторінку цього ж сайту, в атрибуті `href` достатньо вказати лише ім'я вебсторінки, якщо вона знаходиться в тій же папці, що і файл, з якого здійснюється посилання. Наприклад, `<a href="winter.html">Зима</a>`.

Якщо файл знаходиться в деякій папці, то вказують ще й назву папки. Наприклад, `<a href="season/winter.html">Зима</a>`.

Посилання, в якому вказано лише шлях до файлу відносно розташування поточної вебсторінки, називають **відносним**.



Дізнайтеся, як створити посилання для завантаження файла



## 26.2. ВНУТРІШНІ ПОСИЛАННЯ

Аби посилання здійснювалися в межах однієї сторінки використовують **внутрішні посилання**. Для реалізації такого посилання необхідні два елементи: **посилання**, яке вказує, куди необхідно перейти, і **мітка**, яка визначає місце переходу.

Наприклад, ми маємо реалізувати наступну схему навігації (рис. 26.1). При натисканні на текст **Посилання1** необхідно перейти у місце розташування мітки 1, а при натисканні на **Посилання2** — у місце розташування мітки 2.

Для реалізації такої навігації можна застосувати наступний html-код:

```
Посилання1
Посилання2
Мітка 1
Мітка 2
```

Тут Label1 та Label2 імена першої та другої міток відповідно, які задаються з допомогою атрибута name. Для здійснення переходу до відповідної мітки адреса посилання в атрибуті href містить значок решітки та ім'я мітки. Загальна структура коду для створення посилання буде мати вигляд:

```
<!--Посилання, яке вказує,
куди необхідно перейти:-->
 Текст посилання
<!--Опис мітки: -->
 Текст мітки
```



Рис. 26.1. Схема навігації на вебсторінці.



Дізнайтеся, як виконати посилання на зображення та додати кнопку



Виконайте практичне завдання 1, вправи 26.1 та 26.2



### Контрольні запитання та завдання

1. Що таке гіперпосилання?
2. У чому відмінність між абсолютним та відносним посиланнями?
3. Що таке внутрішні та зовнішні посилання?
4. Як установити посилання для завантаження файлу?
5. Як установити посилання на зображення?



### Питання для роздумів

1. Коли доцільно використовувати внутрішні посилання, а коли — зовнішні?



### Завдання для досліджень

- 1\*. Відкрийте деяку сторінку в українській Вікіпедії, визначте, які посилання є внутрішніми, а які — зовнішніми.
- 2\*. Дізнайтеся, що таке карти гіперпосилань. Яке їхнє практичне застосування?



Пройдіть  
тест

## §27.

## Розмітка таблиць засобами мови HTML

### 27.1.

### СТВОРЕННЯ ТАБЛИЦЬ

Для впорядкування розміщення деяких об'єктів на сторінці іноді використовують таблиці. Далі розглянемо основні теги та їхні атрибути для додавання та форматування таблиць.

Для додавання таблиці використовують тег `<table>`. Одним із основних його атрибутів є `border`, який визначає ширину контуру таблиці. Якщо цей атрибут відсутній, то контур таблиці буде невидимим.

З допомогою атрибута `align` визначають вирівнювання таблиці відносно сторінки:

- `left` — вирівнювання ліворуч;
- `right` — вирівнювання праворуч;
- `center` — розташування по центру.

Для формування рядків таблиці використовують тег `<tr>...</tr>`, а для утворення клітинок у вказаних рядках — тег `<td>...</td>`.

Так, для створення таблиці, що має два рядки та три стовпчики, використовують код, наведений на рис. 27.1. Тут кількість стовпчиків визначається кількістю клітинок у рядку, що встановлюються тегом `<td>`.

Як правило, таблиці ще містять заголовок. У html-коді він встановлюється тегом `<caption>...</caption>` з атрибутом `align`. Цей атрибут дозволяє визначати спосіб розташування заголовка: `top` — над таблицею; `bottom` — під таблицею.

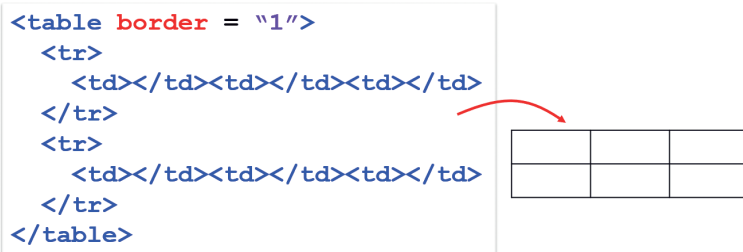


Рис. 27.1. Створення таблиці

Також у таблицях прийнято виділяти заголовки стовпчиків. Для цього замість тегів `<td>...</td>`, що позначають звичайні клітинки, використовують тег `<th>...</th>`. У вікні браузера текст у клітинках, що позначені в якості заголовків, буде відображено напівжирним шрифтом.

## 27.2.

### ВИРІВНЮВАННЯ ВМІСТУ КЛІТИНОК

Текст усередині клітинок можна вирівнювати як по горизонталі, так і по вертикалі. Горизонтальне вирівнювання в клітинці здійснюють з допомогою атрибута `align` у тегах `<td>` або `<th>`. Атрибут може приймати значення: `left`, `right`, `center` або `justify`. Відповідно текст у клітинці буде вирівняно по лівому або по правому краю, по центру або по ширині.

Для встановлення вертикального вирівнювання використовують атрибут `valign` із значеннями `top`, `middle` або `bottom`. У результаті текст буде вирівняно по верхньому краю, по центру або по нижньому краю клітинки.

Якщо необхідно застосувати вирівнювання тексту в клітинці до всього рядка атрибути `align` та `valign` застосовують для тегу `<tr>`.

## 27.3.

### ОБ'ЄДНАННЯ КЛІТИНОК

Кілька клітинок можуть бути об'єднані в одну як по горизонталі, так і по вертикалі. Для об'єднання клітинок у тегу `<td>` або `<th>` записують атрибути: `colspan` — для об'єднання стовпців у рядку; `rowspan` — для об'єднання рядків у стовпці. Так, при використанні html-коду, де використовується атрибут `colspan="3"`, ми отримаємо об'єднання трьох клітинок у рядку, а при використанні атрибута `rowspan="2"` буде об'єднано дві клітинки у стовпці (рис. 27.2).





Рис. 27.2. Приклади об'єднання клітинок

Для створення першої таблиці буде використано код:

```
<table border = "1">
 <tr>
 <td colspan="3"></td>
 </tr>
 <tr>
 <td></td><td></td><td></td>
 </tr>
</table>
```

Зверніть увагу, що в першому рядку наведеного вище коду тег `<td>` використовується лише один раз, на відміну від коду для створення другого рядка, де тег `<td>` використовується тричі. Це тому, що одна клітинка першого рядка буде розтягнута на три стовпчики.

Для створення другої таблиці код буде таким:

```
<table border = "1">
 <tr>
 <td rowspan="2"></td>
 <td></td><td></td>
 </tr>
 <tr>
 <td></td><td></td>
 </tr>
</table>
```

Тут у першому стовпці об'єднано два рядки, отже, у другому рядку ми вказуємо лише дві клітинки, оскільки перша буде розтягнута з першого рядка.



## 27.4. ВСТАНОВЛЕННЯ РОЗМІРІВ КЛІТИНОК

За умовчанням ширина клітинок змінюється в залежності від їхнього вмісту. Для зміни ширини і висоти всієї таблиці для тегу `<table>` використовують атрибути `width` та `height` відповідно. Якщо ж необхідно змінити розміри окремої клітинки, атрибути `width` та `height` додають до тегів `<td>` або `<th>`. Значення ширини та висоти вказують у пікселях без зазначення одиниць вимірювання.

Значення ширини можна вказати у відсотках відносно загальної ширини вікна браузера, в якому відображено вебсторінку.

### ПРАКТИЧНЕ ЗАВДАННЯ 1



Виконайте розмітку сайту BestTravel у вигляді таблиці.

Макет сайту можна умовно представити у вигляді двох таблиць. Перша буде містити заголовок та гарячі тури. Друга — відеоролик з текстовим описом (рис. 27.3).

The image shows a screenshot of the BestTravel website. It is divided into two main sections, each enclosed in a red-bordered box with a red arrow pointing to it. The first box, labeled 'Перша таблиця' (First table), contains the website header with the 'BestTravel' logo and a section titled 'Гарячі тури' (Hot deals). Below this, there are three columns, each featuring a hotel image, its name, location, price, and a 'Замовити' (Book) button. The second box, labeled 'Друга таблиця' (Second table), contains a video player on the left showing a view of St. Sophia's Cathedral in Kyiv, and a text block on the right titled 'Київ - столиця України' (Kyiv - capital of Ukraine) with a list of popular landmarks.

Рис. 27.3. Розмітка сайту BestTravel з використанням таблиць

Спочатку створимо першу таблицю.

1. Відкрийте тег `<table>` після тегу встановлення шрифту і закрийте перед тегом додавання відео.
2. Щоб добре орієнтуватися в макеті таблиці, встановіть атрибут `border="1"`. Згодом межі таблиці приберете.



3. Як видно з макета сайту, таблиця буде містити три рядки, отже, додайте їх з допомогою тегів `<tr>`.
  - a. Першим тегом `<tr>` обгорніть логотип і назву сайту.
  - b. Другим — заголовок розділу **Гарячі тури**.
  - c. Третім — пропозиції гарячих турів.
4. Як видно з макета сайту, дана таблиця матиме три стовпчики. Отже, перший та другий рядки — це є клітинка, утворена об'єднанням трьох клітинок по горизонталі. Для утворення цього ефекту додайте для заголовка сайту тег `<td>` та атрибут `colspan="3"`.
5. Аналогічний тег застосуйте до клітинки, що містить заголовок **Гарячі тури**.
6. Далі обітніть інформацію про тури тегами `<td>`.
7. Збережіть документ та відкрийте його у вікні браузера.  
В результаті буде створено таблицю відповідно до бажаного макета, але її слід допрацювати, використовуючи необхідні атрибути. Кольорове оформлення залишимо наостанок, зараз установимо параметри вирівнювання таблиці та її елементів.
8. Вирівняйте таблицю по центру. Для цього додайте атрибут `align` із значенням `center`.
9. Як правило, стандартна ширина макета сайту становить 1200 пікселів, отже, встановіть для таблиці ширину 1200 пікселів. Це зробіть з допомогою атрибута `width` із значенням 1200.
10. Збережіть таблицю та перегляньте у вікні браузера.
11. Збільшіть висоту клітинки із заголовком **Гарячі тури**. Для цього додайте до тегу `<td>` атрибут `height` із значенням 100.
12. Розташуйте заголовок по центру, застосувавши вертикальне вирівнювання. Для цього додайте до тегу `<td>` атрибут `valign` із значенням `middle`.
13. Вирівняйте вміст клітинок з інформацією про тури. Для цього додайте до тегів `<td>` атрибут `align` із значенням `center`.
14. Збережіть html-код та відкрийте сторінку у вікні браузера.  
В результаті усі елементи мають бути вирівняні відповідно до зразка.
15. Самостійно додайте теги та відповідні атрибути для створення другої таблиці.

#### 27.5.

#### ФОНОВЕ ОФОРМЛЕННЯ КЛІТИНОК

Для заповнення клітинок зображенням використовують атрибут `background`, значенням якого є шлях до файлу зображення:

```
<table background="URL"></table>
```

Для зафарбування клітинок використовують атрибут `bgcolor`, значенням якого є код кольору. Якщо фон необхідно встановити до всієї



таблиці, ці атрибути встановлюють до тегу `<table>`, якщо до рядка — то до тегу `<tr>`, до окремої клітинки — до тегу `<td>` або `<th>`.

## ПРАКТИЧНЕ ЗАВДАННЯ 2



У сайті BestTravel заголовок заповнено фоновим зображенням.

1. Завантажте файл фону **fon.png** з матеріалів до уроку та розташуйте у папці сайту.
2. Додайте до необхідного тегу `<td>` атрибут `background="fon.png"`.
3. До заголовку **Гарячі тури** застосуйте сіру заливку. Для цього додайте до відповідного тегу `<td>` атрибут `bgcolor="#EEEEEE"`.
4. Збережіть файл та відкрийте у вікні браузера. В результаті має змінитися відповідне фонове оформлення.
5. Завершіть роботу над сайтом, прибравши межі таблиці.
6. Збережіть файл та відкрийте у вікні браузера.

### 27.6.

## ОФОРМЛЕННЯ РАМОК ТАБЛИЦІ

Для встановлення зовнішньої рамки таблиці використовують атрибут `frame` тегу `<table>`. Атрибут `frame` може приймати значення:

- `box` — усі чотири сторони рамки;
- `above` — лише верхня межа рамки;
- `below` — лише нижня межа рамки;
- `lhs` — лише ліва межа рамки;
- `rhs` — лише права межа рамки;
- `hsides` — верхня і нижня межі рамки;
- `vsides` — ліва і права межі рамки;
- `void` — зовнішня рамка не відобразиться.

Відображення розділювальних ліній між стовпцями та рядками таблиці визначають за допомогою атрибута `rules` тегу `<table>`.

Атрибут `rules` може приймати значення:

- `all` — відображати всі вертикальні та горизонтальні лінії;
- `rows` — лише горизонтальні лінії між рядками;
- `cols` — лише вертикальні лінії між стовпцями;
- `none` — не показувати розділювальних ліній.

Колір рамки можна задати за допомогою атрибута `bordercolor`, значенням якого є код кольору.



Виконайте вправи 27.1, 27.2 та 27.3

**Контрольні запитання та завдання**

1. Як додати до вебсторінки таблицю? Які теги використовують для встановлення рядків; клітинок?
2. Як у таблиці додати рядок заголовка?
3. Як виконати об'єднання клітинок?
4. Як установити розміри клітинок?
5. Як установити фон для клітинки; усієї таблиці?
6. Які налаштування рамок таблиці можна встановити у мові HTML?

**Питання для роздумів**

1. Чому для створення стовпців таблиці немає окремого тегу?

**Завдання для досліджень**

- 1\*. Чи можна засобами мови HTML вирівняти зображення по центру, аби текст обтікав його ліворуч та праворуч?
- 2\*. Чи можна вставити одну таблицю в іншу?
- 3\*. Як засобами мови HTML виконати таку розмітку таблиці?


Пройдіть  
тест**§28.****Правила ергономічного розміщення відомостей на вебсторінці****28.1.****ПОНЯТТЯ ЕРГОНОМІКИ**

На минулих уроках ми розглянули принаймні два способи створення власних сайтів. Який з них кращий, обирати власне вам. Ваше завдання створити власний сайт, але перед тим, як приступити до цієї цікавої та творчої роботи, ми ознайомимося з правилами ергономічного розміщення відомостей на вебсторінці.



Створення сайту з урахуванням ергономіки може бути визначено, як здатність ефективно реагувати на потреби користувачів і забезпечувати їм комфорт при перегляді сторінки.

У загальному сенсі слова термін **ергономіка** — це використання наукових знань про людину (психології, фізіології, медицини тощо) з метою поліпшення умов роботи на робочому місці.

Ергономіка в цілому характеризується двома принципами:

- комфорт під час використання, який забезпечує зменшення фізичної та психологічної втоми;
- безпека, яка передбачає вибір відповідних рішень для захисту користувача.

## 28.2.

### КРИТЕРІЇ ЕРГОНОМІКИ

Розглянемо основні критерії ергономіки в процесі створення сайту.

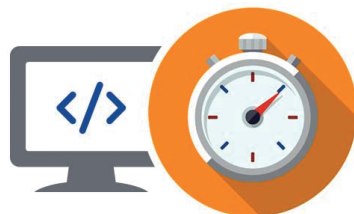
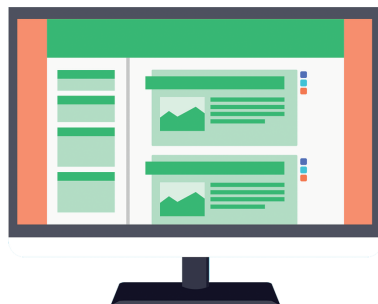
Одним із основних критеріїв є *лаконічність*, або простота. Простота означає, що на сторінці має бути якомога менше елементів дизайну. Завдяки цьому увагу фокусують на контенті. Легше привернути увагу до чогось важливого, коли її не відволікають елементами дизайну.

Не рекомендується використання анімованих зображень. Анімації повинні бути закріплені за важливими повідомленнями, оскільки вони привертають увагу користувача.

Ще один критерій — *чіткість*. Доведено, що друкований текст важче читати з монітора, ніж з паперу, що займає на 25% більше часу. Таким чином, текст повинен бути досить розмежований і структурований за параграфами і назвами для полегшення читання.

Інформація повинна мати такі дані, як ім'я автора, дату останнього оновлення і дату публікації. Погана ідея при створенні сайту на головній сторінці розміщувати посилання на сторінки, які ще не зроблені. Елементи, які використовуються для перегляду, повинні бути розташовані на тому ж місці на кожній сторінці й у разі можливості повинні мати ті ж самі форми.

Наступний критерій — *швидкість*. Час завантаження сторінки має бути якомога коротшим, бо ніхто не чекатиме більше 15 секунд завантаження. Цей фактор в основному залежить від розміру сторінки, зображень на ній і можливостей сервера. Гарною ідеєю є оптимізація розміру зображень, вибираючи для цього відповідний





розмір і кольори. Рекомендовано не завантажувати зображення більше 30–40 Кбайтів.

Ще один критерій — це *взаємодія*. Одним із способів реалізації взаємодії є гіпертекстові посилання. Вебдизайнери радять розміщувати достатню кількість посилань між різними сторінками. Проте занадто багато посилань може зробити читання важким.



Позитивним також є наявність форм для введення звернень до адміністраторів сайту.

Не менш важливим критерієм є *адаптивність*, яка характеризується здатністю вебсайту автоматично ставати персоналізованим без втручання користувача в залежності від пристрою, на якому цей сайт користувач має намір завантажити.

Ще один критерій — *доступність*, яка означає можливість усім користувачам мати доступ до вебсайту. Слід дотримуватися певних правил доступності, щоб дати доступ більшій кількості людей, незалежно від їхніх комп'ютерів або фізичної вади.

Бажано, щоб формати, які використовуються для створення сайту були прозорими, тобто їх можна було прочитати в текстовому режимі — таким чином, переважно використовувати HTML замість Flash. Крім того, зображення і анімація не повинні відволікати увагу від текстової інформації. Отже, графічні ілюстрації повинні бути використані тільки для візуального доповнення тексту. Замість зображень має бути присутній підпис (для цього в html-коді використовуйте атрибут alt). Це необхідно для того, щоб люди з вадами зору зрозуміли сенс зображення.

Що стосується кольорового оформлення, то кольори повинні бути підбрані таким чином, щоб навіть дальтоніки могли їх розрізнити. Між кольором фону і текстом повинен бути достатній контраст, щоб слабозорі люди могли його прочитати.

Розмір шрифту повинен бути адаптований таким чином, щоб користувачі могли збільшити розмір шрифту, якщо це необхідно. Крім того, шрифти не повинні бути дуже малі, оскільки при цьому втомлюються очі і це робить текст нерозбірливим. Для зручності читання рекомендується використовувати шрифт без зарубок.

І ще декілька порад щодо створення власного сайту.

На сайті рекомендується використовувати чіткий та виразний логотип. Це допоможе краще запам'ятати ваш сайт.





Ознакою гарного тону також вважається використання *нижнього колонтитула* — його ще називають «підвалом». Це невеликий блок унизу сторінки, в якому повторено навігаційне меню, розташовано копірайт, посилання на дизайнера сайту, контакти власника ресурсу.



### Контрольні запитання та завдання

1. Якими принципами характеризується ергономіка?
2. У чому полягає суть критеріїв ергономіки?
3. Які вимоги висуваються до кольорового оформлення сайтів?
4. Для чого на сайті використовується логотип?
5. Перелічіть можливі складові нижнього колонтитула сайту.



### Завдання для досліджень

- 1\*. Дослідіть, чи можна при створенні сайту засобами сервісу Google Sites налаштувати такі параметри, як кольорове оформлення, шрифтове оформлення, адаптивність сайту до різних пристроїв.

## §29.

### Практична робота №2. Створення власного сайту

Протягом вивчення даної теми нами було створено сайт BestTravel двома способами: з допомогою сервісу автоматизованого створення сайтів та з допомогою мови HTML. Якщо порівняти ці сайти, що другий сайт є менш досконалим. Але сайти створюють не лише з використанням мови HTML. Професійні вебдизайнери для оформлення сайтів використовують так звані CSS-стили, з якими ви познайомитеся згодом у старших класах. Зазначимо лише, що з їхньою допомогою ви зможете виконувати сайти не лише такого рівня як з допомогою вебдодатків, а значно потужніші. Мова HTML є основою щодо створення загальної структури сайту, отже, без неї не обходиться жоден професіональний вебдизайнер.

Дізнавшись про ергономічні вимоги до створення сайту, а також набувши навичок створення сайту, ви можете розпочати самостійну роботу щодо створення власного сайту. Його створення рекомендуємо виконувати, послідовно дотримуючись алгоритму, який ми використовували протягом усіх уроків з даної теми.



### ЗАВДАННЯ



1. Створіть власний сайт двома способами:
  - з використанням системи автоматизованого створення сайтів **sites.google.com**;
  - засобами мови HTML.
2. Для створення сайту оберіть одну із загальних тем.
  - Зовнішність
  - Hand Made
  - Романтика
  - Спорт
  - Сім'я
  - Гумор
  - Кулінарія
  - Колекціонування
3. Створений вами сайт має відповідати одному із запропонованих макетів, структуру яких виконайте з допомогою таблиць.

1			






Розгадайте електронний кросворд

4. На сайті мають бути присутні принаймні один список, заголовки різних рівнів, зображення та гіперпосилання.



### Висновки

Для автоматизованого створення сайтів існує багато сервісів. Великого поширення набуло використання сервісу **Google Sites**.

**Мова гіпертекстової розмітки HTML** — це мова розмітки гіпертексту для подальшого його відображення у вікні браузера. Являє собою сукупність тегів та правил їхнього застосування.

Засобами мови HTML здійснюється:

- структурування тексту;
- форматування тексту;
- створення списків;
- додавання зображень;
- додавання гіперпосилань;
- створення та форматування таблиць.

При створенні сайту слід дотримуватися **правил ергономічного розміщення** відомостей на вебсторінці.

## Розділ 4

# Опрацювання об'єктів мультимедіа



§30.

### Мультимедіа як ознака сучасного інформаційного світу

30.1.

#### ПОНЯТТЯ МУЛЬТИМЕДІА

Важко знайти поняття, яке б мало більше трактувань, аніж *мультимедіа*. Під цим терміном розуміють одночасне використання різних форм подання інформації, даних, що містять інформацію в різних формах (зазвичай у звуковій і візуальній), систем, що підтримують інтерактивне використання тексту, аудіо, відео й графіки тощо.

То що ж таке мультимедіа? Скористаємось означенням Європейської комісії з проблем упровадження і використання нових технологій.

**Мультимедіа** — це програмний продукт, що містить колекції зображень, текстів і даних, які супроводжуються звуком, відео, анімацією та іншими візуальними ефектами, оснащений інтерактивним інтерфейсом з елементами керування.

Отже, мультимедійний продукт містить такі основні складові: текст, аудіо (звук), зображення, анімацію, відео, інтерактивні елементи.

Слово «мультимедіа» вперше було використано в 1965 році для опису шоу, яке поєднало в собі живу рок-музику, кіно, експериментальні світлові ефекти і нетрадиційне мистецтво.

Далі впродовж 40 років цей термін означав різне: скажімо, наприкінці 1970-х років під ним розуміли презентації, одержані із зображень від кількох проєкторів, синхронізованих зі звуковою доріжкою.

І лише на межі XX і XXI століть це слово набуло сучасного значення.



## 30.2.

## ЗАСТОСУВАННЯ МУЛЬТИМЕДІА

Сьогодні можна говорити про те, що ми живемо в мультимедійному інформаційному середовищі.



Дізнайтеся більше про принципи анімації від Діснея та аніме



Мультимедійні продукти і технології так чи інакше використовуються практично в усіх галузях людської діяльності. Ось перелік лише деяких з них.

**Креативна економіка** (або економіка знань) об'єднує галузі, що базуються на інтелектуальній діяльності (рекламі, архітектурі, кінематографії, дизайні, моделюванні одягу, музейній діяльності тощо).

Фахівці цих галузей створюють мультимедійні продукти (кіно, рекламу, інтерактивні музеї та екскурсії) і застосовують засоби мультимедіа у своїй діяльності під час проєктування та моделювання.

У **комерційній діяльності** підприємства та підприємці застосовують мультимедійні презентації, рекламу, сайти з мультимедіа для просування і продажу своєї продукції.

В **індустрії розваг** мультимедіа застосовується для створення спецефектів у кіно та анімації, для створення вражаючих шоу (наприклад, лазерних) та інтерактивних ігор.

В **освіті** продуктами мультимедіа є комп'ютерні навчальні курси, електронні навчальні матеріали, підручники, енциклопедії тощо.

Широкого розповсюдження використання мультимедіа набуло в **науці та інженерії**. Без застосування комп'ютерного моделювання і симуляції

практично неможливі сьогодні наукові дослідження, проєктування і розробка промислової продукції.

Не минули мультимедійні технології і **медицину**. Віртуальні операції, моделі людського тіла та процесів, що в ньому відбуваються, широко застосовують під час навчання лікарів. А у повсякденній практиці мультимедійні технології використовуються для проведення різноманітних досліджень, операцій та інших медичних заходів.

## 30.3.

## СКЛАДОВІ МУЛЬТИМЕДІА

Як зазначалося, мультимедійний продукт містить такі основні складові: текст, аудіо (звук), зображення, анімацію, відео, інтерактивні елементи. Розглянемо їх детальніше.



**Текст** — це важлива складова таких мультимедійних продуктів, як електронні презентації, сайти, електронні підручники. Великий вибір шрифтів, різноманітні статичні й динамічні ефекти для оформлення написів дозволяють розробникам мультимедіа підвищити сприйняття тексту та досягти високих естетичних результатів під час його оформлення.

**Ілюстрації.** Знімки, малюнки, комп'ютерна графіка — усе це є сьогодні в розпорядженні тих, хто прагне створити ефектний мультимедійний продукт. Розвиток цифрової фотографії, графічних редакторів і мобільних пристроїв сприяють тому, що сьогодні майже кожний може створювати якісний ілюстративний матеріал.

**Анімація.** Під анімацією розуміють серію зображень, які відтворюються послідовно та імітують рух. Для її створення можна застосовувати програмні технології на зразок Adobe Flash, а також анімовані GIF-файли.



Дізнайтеся більше про створення та редагування gif-анімацій

**Аудіо** — невід'ємна частина багатьох мультимедійних продуктів. Наприклад, голос диктора в навчальних курсах, музичний супровід в іграх або підкасти (це цифрові аудіофайли, що поширюються через Інтернет і призначені для відтворення на медіапрогравачах). Аудіо записують за допомогою мікрофона або MIDI-інструментів, наприклад, MIDI-клавіатури, що імітує клавіатуру фортепіано.

**Відео.** На відміну від анімації, що складається з окремих зображень, які відтворюються послідовно, відео — це неперервний потік візуальної інформації, який розділяється на окремі кадри під час запису. Коли вони відтворюються з певною частотою (зазвичай 30 кадрів/с), око сприймає їх як безперервний потік візуальної інформації. Відео отримують за допомогою відеокамери, а розповсюджують за допомогою телебачення, кінофільмів (записаних на носіях) та через Інтернет.

**Інтерактивні елементи** створюються в програмний спосіб і забезпечують взаємодію з користувачем. Це — різноманітні елементи інтерфейсу користувача, гіперпосилання, керовані користувачем анімаційні об'єкти тощо.



Виконайте вправу 30



### Контрольні запитання та завдання

1. Що таке мультимедіа?
2. В яких галузях людської діяльності застосовують мультимедіа?
3. Опишіть основні складові мультимедіа. Наведіть приклади, що демонструють роль кожної складової.
4. В який спосіб і з яких джерел отримують анімацію, аудіо та відео?

**Завдання для досліджень**

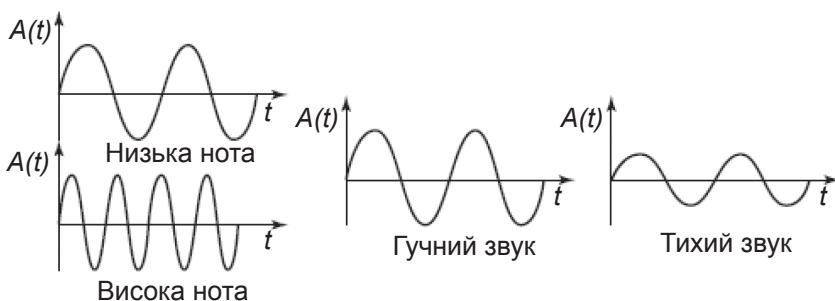
- 1\*. Вставте створений у вправі 30 анімований GIF-файл у документ Word. Який вигляд він матиме? Спробуйте пояснити, чому це відбувається.
- 2\*. З'ясуйте, які є онлайнні ресурси для створення анімованих GIF-файлів. Як вони працюють?

**§31.****Кодування аудіо- та відеоданих.  
Формати аудіо- та відеофайлів****31.1.****АУДІОДАНИ**

Щоб зафіксувати звуки довкола нас (голоси, музику, звуки природи тощо), застосовують техніку звукозапису. Загалом для цього потрібно мати:

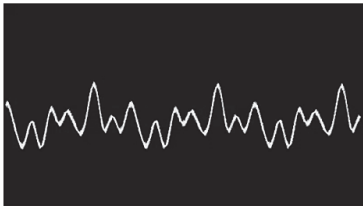
- 1) мікрофон, а саме, пристрій, що перетворює акустичні коливання на електричний сигнал, або звуковий синтезатор;
- 2) звукову карту — пристрій, що перетворює електричні коливання на послідовність цифр;
- 3) пристрій для запису даних на цифровий носій (жорсткий диск, флешкарту тощо).

Акустичні коливання або звукові хвилі характеризуються частотою і амплітудою коливань.

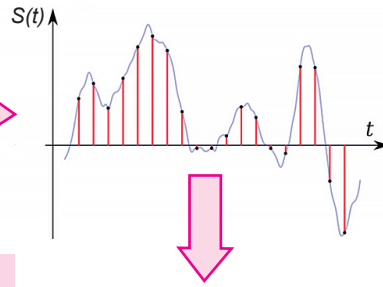


**Рис. 31.1.** Візуальне представлення акустичних коливань

Звукові сигнали, щоб зберегти в цифровому вигляді, дискретизують, тобто через певні проміжки часу вимірюють амплітуду сигналу і записують її величину у двійковій формі.



дискретизація

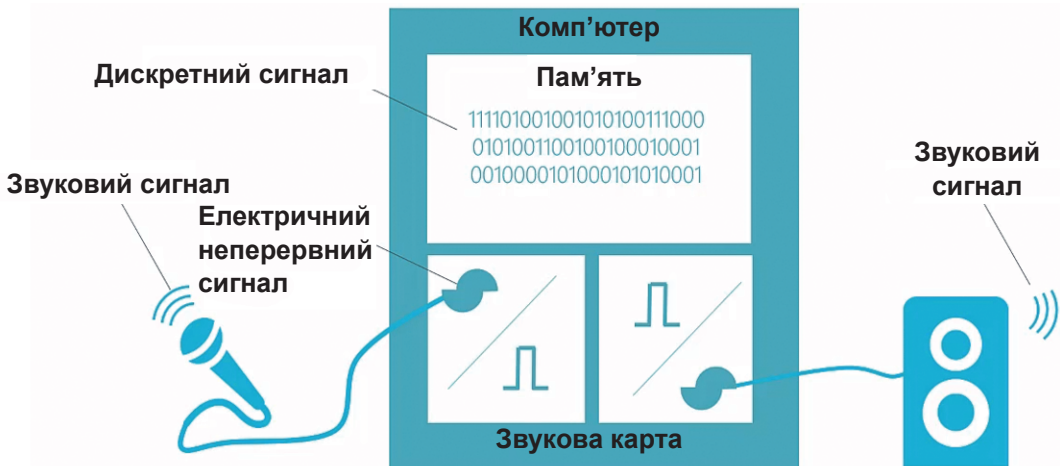


1	0	0	0	1	1	1	1
0	0	1	1	1	0	0	1
1	0	0	0	1	1	1	1
0	0	1	1	1	0	0	1

Подання в пам'яті комп'ютера

**Дискретизація** — це вимірювання значень звукової хвилі в певні моменти часу.

На схемі нижче представлено процес одержання, обробки та відтворення звукового сигналу з використанням комп'ютера.



Під час відтворення звуку дані потоком передаються на відповідний пристрій. Наприклад, якщо йдеться про звук у форматі CD-Audio, то за 1 с передається 1411,2 Кбітів. Кількість інформації, переданої за одиницю часу під час відтворення файлу, називають *бітровою частотою* (*бітрейтом*) і вимірюють у бітах за секунду. Чим більший бітрейт, тим ближчий звук до оригіналу. Обсяг звукового файлу в байтах дорівнює добутку бітрейту на тривалість звучання в секундах, поділеному на 8.

$$\text{Обсяг файлу} = (\text{бітрейт} \cdot \text{тривалість}) / 8$$





**Задача 1.** Бітова частота звукового файлу дорівнює 128 Кбіт/с, обсяг файлу становить 4 Мбайти. Визначте час тривалості звукового файлу, вказавши кількість повних хвилин і секунд.

*Розв'язання.* Обсяг файлу = (бітрейт · тривалість)/8

Тривалість = обсяг файлу/бітрейт · 8

4 Мбайти = 4048 Кбайтів

$4048 \text{ Кбайтів} / 128 \text{ Кбіт/с} \cdot 8 = 253 \text{ с} = 48 \text{ хв } 13 \text{ с.}$

Для зменшення обсягу звукових даних застосовують стиснення. Стиснення звуку з втратами зазвичай базується на тому, що люди не розрізняють або погано розрізняють негучні сигнали на фоні гучних.

Алгоритми стиснення видаляють негучні сигнали, завдяки чому вдається значно зменшити значення бітрейту. Сьогодні найчастіше використовують бітрейти величиною 256, 192 і 128 кбіт/с.

Для аматорських завдань із звукового оформлення кліпів можна записати голос за допомогою диктофона, вбудованого в смартфон. Також можна скористатися аудіозаписами з Інтернету.

## 31.2.

### ФОРМАТИ ФАЙЛІВ ДЛЯ ЗБЕРІГАННЯ ЗВУКУ

Оскільки для зберігання звукової інформації застосовують різні методи кодування та стиснення, існує багато звукових форматів файлів. Ознайомтеся з деякими з них.

**MP3-формат** забезпечує високий ступінь стиснення даних (в 11 разів вищий порівняно з форматом CD-Audio) і дає змогу створювати файли невеликого обсягу, які мають розширення **.mp3**. Використовує стиснення з втратами, підтримує стерео і потокове передавання. Сьогодні вважається найпоширенішим форматом зберігання аудіозаписів в Інтернеті.

**WMA-формат.** Це ліцензований формат, розроблений корпорацією Microsoft як альтернатива MP3. Номінально він забезпечує більш ефективне стиснення даних порівняно з конкурентом, однак загалом його переваги не є очевидними. Як послідовний прихильник дотримання авторських прав корпорація Microsoft забезпечує підтримку формату цифрової системи керування авторськими правами, що унеможливує прослуховування захищеного запису на будь-якому комп'ютері, крім того, з якого він був завантажений.

**Monkey's Audio.** Формат кодування аудіо без втрат. Файли мають розширення **.ape** (аудіодані) та **.apl** (метадані). Забезпечує дуже високий ступінь стискання аудіо, проте распаковування вимагає великих обчислювальних ресурсів.



**WAV-файли** містять цифровий ряд аудіоданих, який можна відтворити з використанням багатьох мультимедійних плеєрів і цифрових додатків Microsoft Windows. У порівнянні з форматом MP3 серед інших цифрових видів аудіостиснення даних WAV-файлами більше через його нестислий цифровий аудіоконтент.

Ми розглянули найбільш уживані аудіоформати, які можна представити у вигляді наступної схеми.



В якості прикладу ви можете побачити, як за обсягом відрізняються файли з однією й тією ж піснею у форматі **.wav**, що є форматом без втрати даних, та у форматі **.mp3**, що є форматом з маловідчутними на слух втратами.



### 31.3.

## ВІДЕОДАНИ

Щоб зафіксувати візуальну інформацію, можна застосовувати відеозапис — а саме, електронну технологію запису відеосигналу або цифрового потоку відеоданих на фізичний носій.

Звичайно, це можна робити і за допомогою комп'ютера. У такому разі йдеться про *захоплення відео* — процес перетворення відеосигналу із зовнішнього джерела на цифровий потік і запис його у відеофайл.

Зовнішнім джерелом можуть бути відеокамери, вебкамери, потокове відео з мережі, TV-тюнери, ресивери цифрового телебачення та інші пристрої.

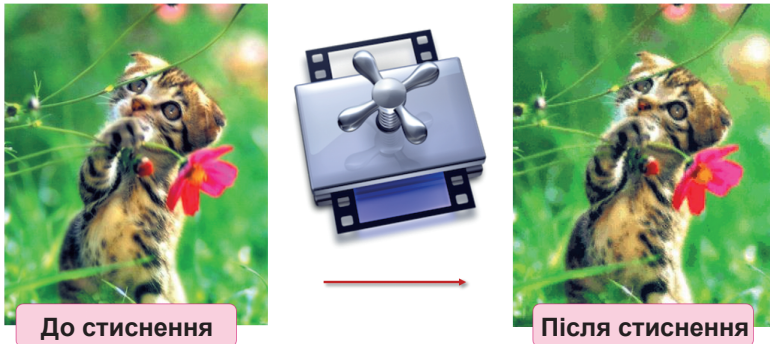
Зазначимо, що відеозображення телевізійної якості в «початковому» вигляді мають дуже великий обсяг даних. Проведемо прості розрахунки, виходячи з того, що відео — це потік нерухомих зображень-кадрів. Припустимо, що розмір статичного зображення становить  $(800 \times 600)$  пікселів, а колір кожного пікселя кодується двома байтами. Отже, одне зображення займатиме в пам'яті  $800 \times 600 \times 2 = 960\,000$  байтів  $\approx 1$  Мбайт. Це начебто й небагато: на флешці, що має 8 Гбайтів, поміститься понад 8000 таких зображень.

Але якщо йдеться про фільм, стандартна швидкість відтворення якого становить 24 кадри за секунду, то в нестисненому вигляді на



флешку можна буде записати лише фрагмент тривалістю близько 340 с, а це менше 6 хвилин. Ось чому відеозображення неодмінно потребують стиснення (компресії).

Для стиснення відеоданих застосовують алгоритми стиснення з втратами, коли початкові дані можна відновити лише з викривленнями, які можуть бути непомітними для людини.



Алгоритми стиснення відео ґрунтуються на відповідних алгоритмах для статичних зображень, використовуючи їх у різних поєднаннях і послідовностях. Але є і додаткові методи компресії. Наприклад, кожний наступний кадр можна зберігати не повністю, а лише його відмінності від попереднього.

### 31.4.

## ФОРМАТИ ФАЙЛІВ ДЛЯ ЗБЕРІГАННЯ ВІДЕО

Як і аудіо, відео зберігається у файлах різноманітних форматів. Розглянемо найпоширеніші з них.

**WMV** (Windows Media Video). Формат розроблений корпорацією Microsoft для програвача Windows Media. Відеофайли, збережені в цьому форматі, мають розширення **.wmv**.

Формат **MPEG**. Створений групою MPEG ще на початку 1990-х років. Файли цього формату мають розширення **.dat**, **.mpg**, **.mpe**, **.mpeg**, **.mpl**, **.mp2** або **.mp4**. Група MPEG розробила три алгоритми стиснення відеоданих. MPEG-1, який був створений першим, забезпечував середню якість зображення і не дуже високий ступінь стиснення. MPEG-2 давав відчутно кращу якість відеозображень. MPEG-4 забезпечує ще вищу якість і більший ступінь стиснення порівняно зі своїми попередниками, підтримує додаткові можливості (захист від несанкціонованого копіювання, використання інтерактивних елементів тощо) і потокове передавання відео.

Також широкого поширення набув формат **.avi**, відомий як Audio Video Interleave. Файли цього формату містять стиснутий мультимедійний вміст, що складається з аудіо- та відеоданих.



### Контрольні запитання та завдання

1. Де можна взяти аудіо для аматорського проєкту?
2. Що таке захоплення відео? З яких пристроїв можна одержати відеопотік?
3. Назвіть найпоширеніші формати для зберігання аудіо- та відео-файлів.



### Питання для роздумів

- 1\*. Чому під час редагування звуку використовують файли формату WAV, а не MP3?



### Завдання для досліджень

- 1\*\*. З'ясуйте, що таке відео з високою чіткістю. Що означають написи 720p, 1080p, 1080i, 4K?

## §32.

# Програмне забезпечення для опрацювання об'єктів мультимедіа. Засоби перетворення аудіо- й відеоформатів

## 32.1.

### КОДЕКИ

Алгоритми стиснення та кодування мультимедійних даних реалізують у спеціальних програмах, що називаються *кодеками*. Інколи кодеками називають також пристрої, здатні перетворювати аналоговий сигнал на цифровий і навпаки. Зокрема, кодеком є звукова карта.

**Кодек** (англ. *codec*, від *coder/decoder* — кодування/декодування) — програма, що перетворює потік даних або сигнали для їхнього подальшого використання. Кодеки можуть як кодувати, так і розкодувати мультимедійні дані.

Звукові та візуальні дані потребують різних методів стиснення, а тому для них розроблені окремі кодеки. Кожен кодек використовує свої алгоритми стиснення і зазвичай не працює з даними від іншого кодека. Кодеки взаємодіють з певними прикладними програмами, зокрема з медіапрогравами, і допомагають їм відтворювати медіадані. Кодеки вбудовано в медіапрограми, їх можна встановити як окремі програми





(наприклад, FLAC, LAME, XviD та DivX,) або як пакети, найбільшим і найпопулярнішим з яких сьогодні вважається K-Lite.

### 32.2.

## МЕДІАКОНТЕЙНЕРИ

Для того, щоб створити мультимедійний об'єкт, часто недостатньо зберегти дані в одному з форматів, що забезпечують стиснення і кодування даних, оскільки вони не надають можливість синхронізувати (узгодити в часі) відео- та аудіопотоки і додати до них супровідний текст або меню. Такі можливості є у *форматів-медіаконтейнерів*.

**Медіаконтейнер** — формат, що дає змогу розміщувати в одному файлі мультимедійні дані різних типів і синхронізувати звук, відеозображення і текстову інформацію.

Розглянемо кілька найпоширеніших форматів-медіаконтейнерів.

**WAV (Waveform Audio Format).** Медіаконтейнер для зберігання нестисненого звуку. Створений корпорацією Microsoft і прийнятий як стандарт для звукового супроводу роботи системи Windows і комп'ютерних ігор. Звук у форматі WAV зберігається без втрати якості, але відсутність стиснення призводить до того, що обсяги wav-файлів дуже великі. Через цей та деякі інші недоліки формат WAV не набув популярності серед користувачів, проте його і зараз широко використовують під час редагування звуку та на переносних аудіопристроях, зокрема на цифрових диктофонах.

**AVI (Audio and Video Interleaved).** Надає можливість об'єднувати нестиснені або закодовані різноманітними кодеками аудіо- та відеодані. Донедавна AVI був лідером серед форматів для платформи Windows, однак через певні обмеження, зокрема на обсяг файла (не більше 4 Гбайти), його популярність зменшилася.

**MOV (QuickTime Movie).** Медіаконтейнер, розроблений корпорацією Apple Computer спеціально для мультимедійного програвача QuickTime, який можуть відтворювати й інші програвачі. Як і формат AVI, дозволяє поєднувати аудіо- та відеопотоки, закодовані в різний спосіб.

### 32.3.

## КОНВЕРТУВАННЯ ВІДЕО ТА АУДІО

Якщо вихідний формат відео- або аудіофайла не підходить для обробки і відтворення, можна конвертувати файл в інший формат. Для цього є досить багато різноманітних безкоштовних програм, зокрема й онлайн-ових.



Отже, **конвертер** — це програма, що перетворює один формат файлу в інший.

Конвертування може здійснюватися як у певний формат, так і у формат певного пристрою (ігрова консоль, смартфон тощо). Програми-конвертори також часто мають функції з монтажу, об'єднання відео, створення ефектів і субтитрів. Огляди та переліки таких програм легко знайти в Інтернеті.



Дізнайтеся більше про конвертування відеофайлів

#### 32.4.

### МУЛЬТИМЕДІЙНІ ПРОГРАВАЧІ

Для відтворення (прослуховування чи перегляду) аудіо- та відеофайлів використовують спеціальні програми — програвачі (їх ще називають *плеєрами*), переважну більшість яких можна безкоштовно завантажити з Інтернету. Перелічимо найпопулярніші програвачі, які підтримують більшість поширених форматів мультимедійних файлів.

**VLC Media Player.** Потужний програвач, що відворює більшість аудіо- та відеоформатів, зокрема MPEG-1, MPEG-2, MPEG-4, DivX, DVD, VCD тощо.

**Media Player Classic Home Cinema.** Безкоштовний програвач з простим і зручним інтерфейсом. Підтримує дуже широкий спектр аудіо- та відеоформатів. Якщо файл не вдається відтворити за допомогою інших програм, варто спробувати зробити це в Media Player Classic.

**GOM Player.** Безкоштовний медіаплеєр для Windows. Може програвати більшість медіафайлів без сторонніх кодеків, а також пошкоджені та недозавантажені файли. Підтримує знімок кадру, запис звуку, субтитри.

**Програвач Windows Media (Windows Media Player).** Входить до складу операційної системи Windows, відтворює аудіо- та відеофайли більшості популярних форматів.



Рис. 32.1. Інтерфейс медіапрогравача



#### Контрольні запитання та завдання

1. Що таке кодек? Яке його призначення?
2. Які дані можуть містити медіаконтейнери?
3. Які програвачі мультимедіа є найпопулярнішими?



**Питання для роздумів**

- 1\*. Чи можна, на вашу думку, створити один кодек для звуку та відео?
- 2\*. Яку проблему, що виникає в разі окремого зберігання аудіо- та відеоряду, можна вирішити з допомогою медіаконтейнерів?

**Завдання для досліджень**

- 1\*. Знайдіть в Інтернеті інформацію про формати-медіаконтейнери ASF, MP4, RealMedia, 3gp. Які вони мають характерні особливості?
- 2\*\*. Які є можливості щодо конвертування аудіо- та відеоонлайн?

Пройдіть  
тест**§33.****Відеозахоплення****33.1.****ЗНАЙОМСТВО З ПРОГРАМОЮ FASTSTONE CAPTURE**

Однією з програм, що надає можливості захоплення відео є програма **FastStone Capture**. Вона є безкоштовною і не дивлячись на свій незначний обсяг, що займає на жорсткому диску, після встановлення має потужні інструменти для захоплення, редагування та анотування відео. Далі розглянемо використання версії 9.2 цієї програми.

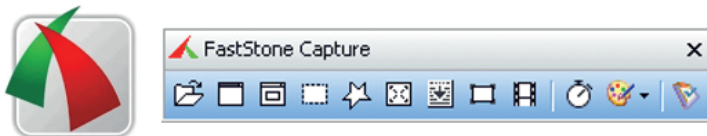


Рис. 33.1. Інтерфейс програми FastStone Capture

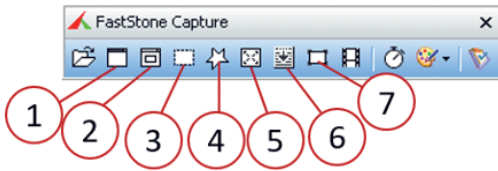
**33.2.****СТВОРЕННЯ ТА РЕДАГУВАННЯ СКРІНШОТІВ**

Однією із можливостей програми є створення скріншотів різних областей екрана та їхнє наступне редагування.

**Скріншот** — це копія зображення екрана.

Більшість інструментів панелі FastStone Capture надають можливість вибору різних варіантів створення скріншотів (рис. 33.2): захоплення активного вікна, обраного вікна або об'єкта, прямокутної або довільної області, усього екрана, вікна з прокручуванням та фіксованої області.





**Рис. 33.2.** Інструменти створення скріншотів: (1) — захоплення активного вікна, (2) - обраного вікна, (3) — прямокутної області, (4) — довільної області, (5) — усього екрана, (6) — вікна з прокручуванням, (7) — фіксованої області

Після створення скріншоту одним із можливих способів ви автоматично потрапляєте у вікно, що дозволяє відредагувати зображення та зберегти його. Вбудований графічний редактор має високі можливості, аби зробити ваше зображення більш інформативним.



Дізнайтеся більше про створення та редагування скріншотів

### 33.3.

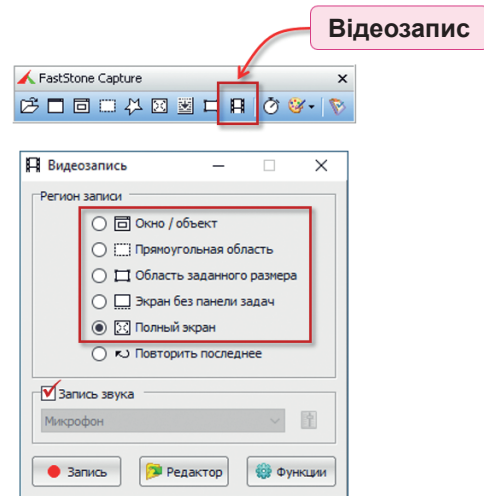
## ВІДЕОЗАХОПЛЕННЯ

Ще однією можливістю програми FastStone є захоплення відео, що дозволяє створювати навчальні відеопрезентації. Відеозахоплення здійснюється за допомогою інструмента **Відеозапис** (**Відеозапис**). Після його вибору з'являється вікно, що надає можливість вибору області відеозахоплення.

Також при відеозахопленні можливий запис звукового супроводу через мікрофон або динаміки. Для цього необхідно встановити прапорець біля опції **Запис звуку** (**Запис звуку**).

Перед здійсненням відеозахоплення доцільно здійснити ще деякі налаштування: вказати частоту кадрів, якість відеозахоплення, налаштувати захоплення вказівника миші тощо. Вікно налаштувань відкривається кнопкою **Функції** (**Функції**).

Розглянемо створення та редагування відеозахоплення, виконавши практичні завдання.



**Рис. 33.3.** Інтерфейс вікна **Відеозапис**

### ПРАКТИЧНЕ ЗАВДАННЯ 1




Створити відеозахоплення та його редагування. Подія, що має відбуватися при відеозахопленні — це вставлення та форматування графічного об'єкта в програмі MS Word.



Перед тим, як здійснювати відеозахоплення, необхідно продумати алгоритм дій, що мають відбуватися під час запису відео, після чого розпочати запис відео.

### Запис відео.

1. Відкрийте програму MS Word.
2. Запустіть програму FastStone Capture.
3. Оберіть інструмент  **Видеозапись (Відеозапис)**.
4. Оберіть регіон запису **Окно / объект (Вікно / об'єкт)**.
5. Для встановлення налаштувань відеозапису натисніть кнопку **Функции (Функції)**.
6. Установіть частоту кадрів в секунду – 30.
7. Установіть опцію **Захват указателя мыши (Захоплення вказівника миші)**.
8. Зробіть видимими клацання мишею, встановивши режим **Щелчки мыши (Клацання миші)**.
9. Оберіть фігуру у вигляді зірочки розміром 7 пікселів.
10. Натисніть **Ок**.
11. Натисніть кнопку **Запис**.

Гарною манерою при створенні відео є додавання його заголовка.

12. Установіть опцію **Вставить заголовок в начало видео (Вставити заголовок у початок відео)**.
13. У вкладці **Заголовок** напишіть заголовок «Додавання графічного об'єкта в MS Word». Установіть синій колір тексту заголовка та білий колір фону.
14. Перейдіть на вкладку **Описание (Опис)** та додайте опис до заголовка за власним бажанням.
15. Натисніть кнопку **Старт**. З цього моменту починається запис відеозахоплення.
16. Тепер виконайте необхідні дії у вікні MS Word.

У програмі MS Word оберіть вкладку **Вставлення**.

- Оберіть інструмент **Фігури**.
- Оберіть інструмент **Прямокутник** та побудуйте його.
- Зробіть у записі невелику паузу, тобто певний час не виконуйте ніяких дій. Ця пауза вам потім знадобиться для пояснення призначення маркерів, що з'явилися навколо прямокутника.
- Виконайте зміну розмірів прямокутника.
- Виконайте його поворот.
- На вкладці **Формат** застосуйте до прямокутника стиль.
- Змініть колір заливки, обравши інструмент **Заливка фігури**.



- Змініть колір контуру прямокутника, обравши інструмент **Контур фігури**.
- Змініть розмір контуру прямокутника.
- Застосуйте заготовку, обравши інструмент **Ефекти для фігур**, а потім інструмент **Заготовка**.

17. Зупиніть запис, натиснувши **Ctrl+F10**.

18. Натисніть **Зберегти**. Після чого збережений файл буде запущений для перегляду.

### 33.4.

## РЕДАГУВАННЯ ВІДЕО

Звичайно, відзняте відео можна редагувати. Для цього слід натиснути кнопку **Редактор**. У результаті має відкритися вікно редактора відео (рис. 33.4).

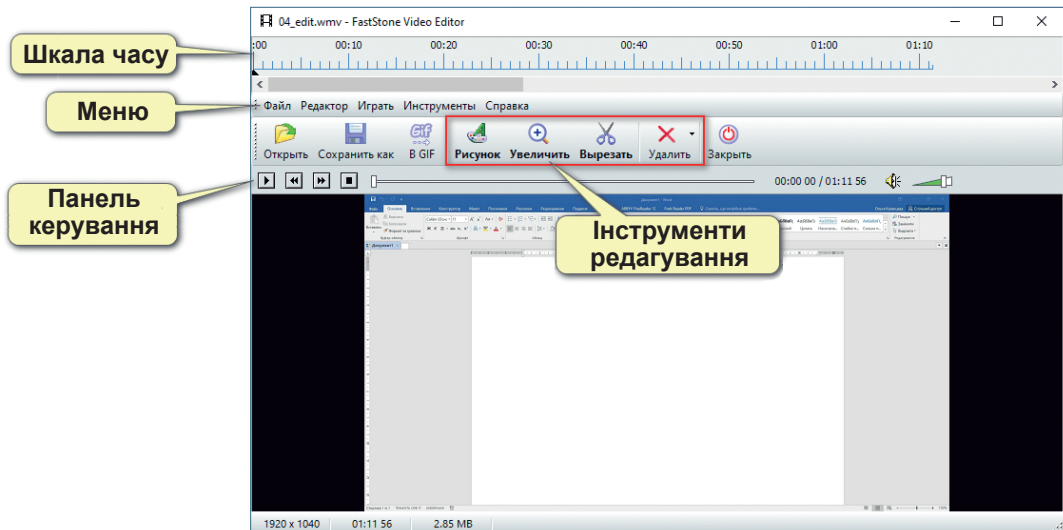


Рис. 33.4. Вікно редактора відео

### Відео ефект масштабування

Одна з цікавих можливостей редагування — це *відео ефект масштабування*, що дозволяє здійснити наближення збільшеного фрагмента відео.

## ПРАКТИЧНЕ ЗАВДАННЯ 2




Додайте до відео, відзнятого в попередньому завданні, відео ефект масштабування.

1. Оберіть фрагмент, до якого доцільно буде застосувати масштабування.



2. Перемістіть повзунок у необхідне місце на шкалі часу.

3. Оберіть інструмент  **Увеличить (Збільшити)**.



Ознайомтеся з роботою програми Кіностудія Windows

4. Виділіть бажану область збільшення. Натисніть **ОК**. В результаті на шкалі часу з'явиться жовта смуга, що визначає час застосування ефекту збільшення. Розмір цієї смуги можна збільшити або зменшити в залежності від бажаного результату.

### Додавання графічних об'єктів


Ще одна можливість програми FastStone — додавання виносок, зображень тощо.

### ПРАКТИЧНЕ ЗАВДАННЯ 3

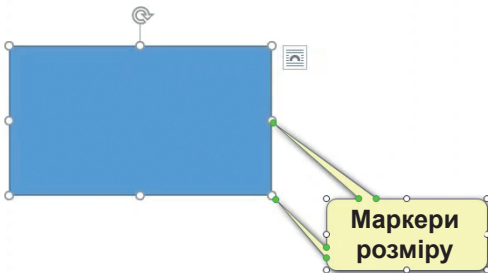


Додайте виноски для пояснення маркерів прямокутника, що з'являються після його додавання в програмі MS Word.

1. Установіть курсор в необхідне місце на смузі часу (це має бути момент запису, в якому здійснюється демонстрація маркерів, що з'являються навколо побудованого в MS Word прямокутника).

2. Для виклику графічного редактора оберіть інструмент  **Рисунок**.

У результаті має відкритися редактор зображень..



3. Оберіть інструмент **Рамка с текстом (Рамка з текстом)**.

4. Створіть рамку тексту біля прямокутника та вкажіть на маркери розміру відповідними елементами виноски.

5. Змініть текст, обравши в контекстному меню команду **Редактировать текст (Редагувати текст)**. Уведіть текст.

6. Збережіть внесені зміни, натиснувши **ОК**. В результаті на шкалі часу має відобразитися зелена смуга, що визначає час дії відображення створеної виноски.

Як і у випадку з попереднім ефектом, змінюючи довжину смуги, можна змінити час відображення ефекту рисунка. Для зміни даного ефекту слід клацнути на смузі правою кнопкою миші і в контекстному меню обрати команду **Змінити**.




## Вирізання зайвих фрагментів

Що ж робити, якщо відзнято зайвий матеріал? Звичайно, непотрібні частини можна вирізати. Для цього використовується інструмент **Вырезать (Вирізати)**. Розглянемо, як це можна зробити.

### ПРАКТИЧНЕ ЗАВДАННЯ 4



1. Установіть курсор у необхідне положення на часовій шкалі.
2. Оберіть інструмент  **Вырезать (Вирізати)**.

У результаті програма запропонує вирізати фрагмент довжиною 5 секунд.

3. Нічого не змінюючи, натисніть **ОК**.

Для цього ефекту на шкалі часу з'являється смугастий прямокутник, розмір і місце положення якого в разі необхідності можна змінити. Виділена для вирізання частина файла буде видалена лише після збереження файла.



Виконайте вправи  
33.1–33.4

4. Для збереження файла натисніть кнопку **Зберегти як**.



### Контрольні запитання та завдання

1. Які можливості надає програма FastStone?
2. Що таке скріншот? Які види скріншотів можна виконати у програмі FastStone?
3. Який відеоефект можна додати у програмі FastStone?
4. Які дії редагування відео надає програма FastStone?



### Питання для роздумів

- 1\*. На вашу думку, яких відеоефектів не вистачає в редакторі FastStone?



### Завдання для досліджень

- 1\*. Дізнайтеся, чи існують онлайн-редактори відео.



Пройдіть  
тест



## §34.

## Розміщення відео- та аудіофайлів в Інтернеті

## 34.1.

## ОГЛЯД ХОСТИНГІВ



Сьогодні, в еру бурхливого розвитку платформ хостингу контенту, соціальних мереж і мобільних пристроїв, розміщення фотографій, аудіо- та відео в Інтернеті стало повсякденною справою багатьох користувачів. Далі розглянемо, як власне можна розмістити відео- та аудіофайли в мережі Інтернет.

Досі ми розглядали технології мультимедіа, які передбачають, що мультимедійні дані зберігаються локально, редагуються і відтворюються за допомогою програм, установлених на комп'ютері. Проте мультимедіа давно є невід'ємною частиною вебпростору.

У мережі існує багато хостингів для завантажень фотографій та відео. Щодо завантаження, збереження та впорядкування фотографій найбільшої популярності набула служба <https://photos.google.com>, яка може автоматично копіювати знімки з вашого смартфона на інші ваші пристрої. Також вона дозволяє в різноманітний спосіб упорядковувати колекції знімків.

Звичайно, обсяг наданого дискового простору в онлайні не безмежний, і періодично доводиться переміщувати впорядковані колекції знімків до основної колекції.

Одним із поширених продуктів мультимедіа є відео, що зберігається на так званих відеохостингах.

Відеохостинги надають можливість:

- здійснювати пошук відео;
- передивлятися відео;
- завантажувати власне відео;
- створювати відеотеку;
- підписуватися на оновлення відео;
- переглядати історію переглядів.

І всі ці можливості можна використовувати не тільки задля власного задоволення, а й у комерційних цілях для отримання прибутку.

Щодо завантаження та зберігання аудіо- та відеофайлів найбільшої популярності набув **відеохостинг YouTube**.



YouTube був створений трьома колишніми працівниками компанії PayPal Чадом Герлі, Стівеном Чені та Джаведом Карімом у 2005 році. У 2006 році YouTube продали компанії Google, і зараз він діє як їхній дочірній підрозділ.

Сайт надає користувачам можливість здійснювати пошук, переглядати, оцінювати, коментувати та ділитися відеозаписами. Сьогодні він є одним з перших за кількістю відвідувачів. YouTube має досить розвинену довідкову систему, яка допомагає ефективно користуватися цією платформою.



### 34.2.

## ПОШУК ТА ВІДТВОРЕННЯ ВІДЕО

Найпростішими функціями YouTube є пошук і відтворення відео. Розглянемо, як це робити після завантаження самого відеохостингу за адресою: [www.youtube.com](http://www.youtube.com).

Знайдемо, наприклад, відео про штучний інтелект, що є одним із сучасних на сьогоднішній день напрямків розвитку інформаційних технологій.

1. У рядку пошуку введіть ключову фразу «штучний інтелект» та натисніть кнопку пошуку.
2. Аби звузити коло пошуку, в YouTube передбачена фільтрація пошуку. Щоб скористатися можливістю використання фільтра, натисніть кнопку **Фільтрувати**. В результаті буде запропоновано можливі способи фільтрації: за датою завантаження; за типом; за тривалістю та за характеристиками.
3. Установіть фільтр на час відтворення, обравши значення **Короткі (менше 4 хвилин)**. В результаті буде виведено лише відеофайли про штучний інтелект, що мають тривалість менше ніж 4 хвилини. Час відтворення можна побачити у нижньому правому куті заставок відео.
4. Оберіть, наприклад, друге відео.

Оскільки YouTube — комерційний проєкт, часто відео починається з рекламного ролика, який можна переглянути повністю або через деякий час після початку вимкнути, клацнувши відповідне посилання.

5. Пропустіть рекламу. Після реклами має розпочатися відтворення бажаного ролика.

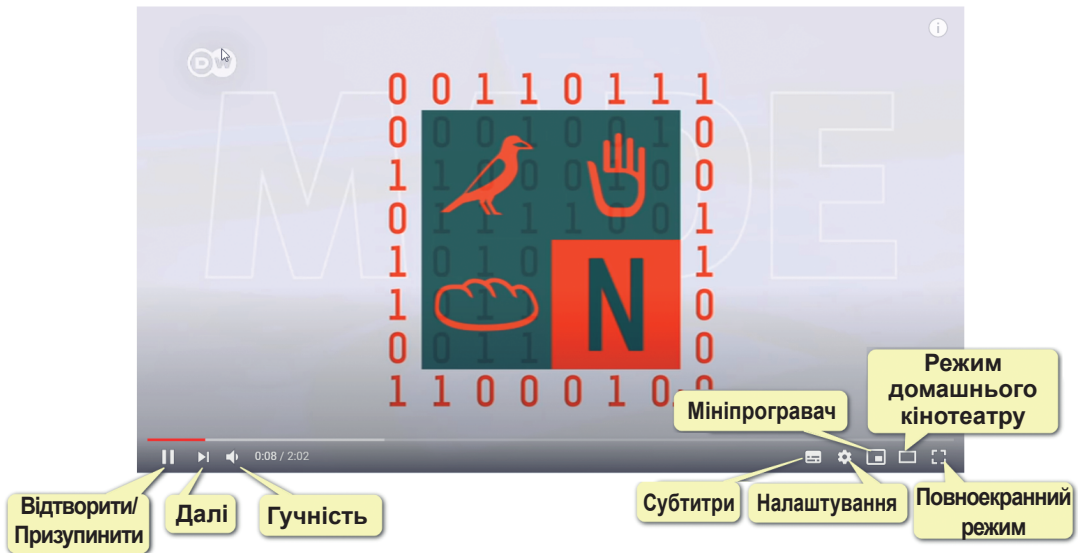
Ролик відображається у відеоплеєрі, який має інтуїтивно зрозумілі кнопки керування.

Для здійснення налаштувань відтворення відео призначена кнопка **Налаштування**. У вікні налаштувань є можливість збільшення або зменшення швидкості відтворення, увімкнення відтворення субти-





трів (субтитри — це супроводження відео, крім звуку, ще й написами), встановлення якості відтворення (найбільше значення відповідає найкращій якості).

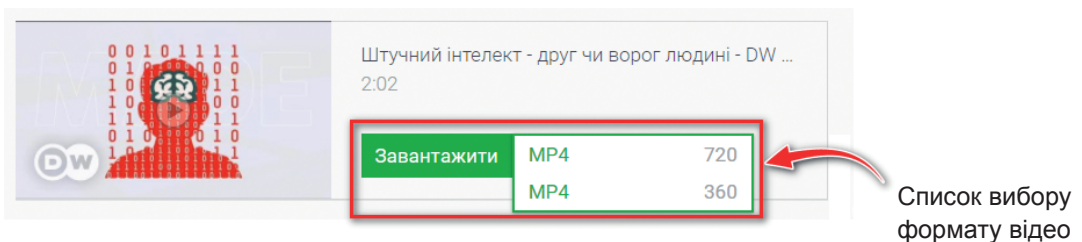


### 34.3.

### ЗАВАНТАЖЕННЯ ВІДЕО З YOUTUBE

Розглянемо, як завантажити відео з YouTube.

1. Відкрийте на YouTube сторінку з відео, яке вам потрібно завантажити.
2. Додайте символи **ss** перед словом **youtube** в адресі ресурсу та натисніть **Enter**. У результаті ви потрапите на сайт [savefrom.net](http://savefrom.net), саме з якого і має здійснитися завантаження обраного відео.
3. На сайті [savefrom.net](http://savefrom.net) оберіть у списку бажаний формат відео.



Наведені у списку числа визначають кількість пікселів по висоті відеокадру. Формат із більшим числовим значенням буде мати більшу якість, але й важитиме більше. Після вибору формату буде здійснено завантаження на ваш комп'ютер.



4. Відкрийте скачаний файл з допомогою будь-якого відеопрограва-  
вача, встановленого на вашому комп'ютері.

### 34.4.

## СТВОРЕННЯ ТА ОФОРМЛЕННЯ ВЛАСНОГО КАНАЛУ

Звичайно, через відеохостинг YouTube можна поширювати і власне відео. Зверніть увагу на основні правила спільноти та дотримуйтеся їх.

Заборонено завантажувати відео, що містить:

- шкідливий або небезпечний вміст;
- вміст, який пропагує ненависть;
- насильницький вміст;
- агресивні дії та погрози;
- спам або шахрайство,

а також відео, що порушує авторські права.

У заборонених відео працівники YouTube часто вимикають звук, а потім узагалі видаляють їх із сайту.

Завантаження власного відео розпочинається зі створення власного каналу на відеохостингу. Оскільки YouTube — це відеохостинг компанії Google, то для створення власного каналу потрібно мати обліковий запис Google.

Далі розглянемо послідовність дій щодо створення власного каналу на відеохостингу.

1. У вікні сайту YouTube увійдіть до власного облікового запису Google.
2. Відкрийте меню облікового запису, натиснувши піктограму, що визначає власника.
3. Оберіть команду **Створити канал**.
4. У вікні, що з'явилося, оберіть кнопку **Почати**.
5. Оберіть бажаний вид каналу.

У наступному вікні можна здійснити додаткові налаштування каналу. Виконайте налаштування або натисніть **Налаштувати пізніше**.

У результаті канал буде створено. Для подальшої зручної та приємної роботи можна здійснити додаткові налаштування.

### 34.5.

## ЗАВАНТАЖЕННЯ ВЛАСНОГО ВІДЕО

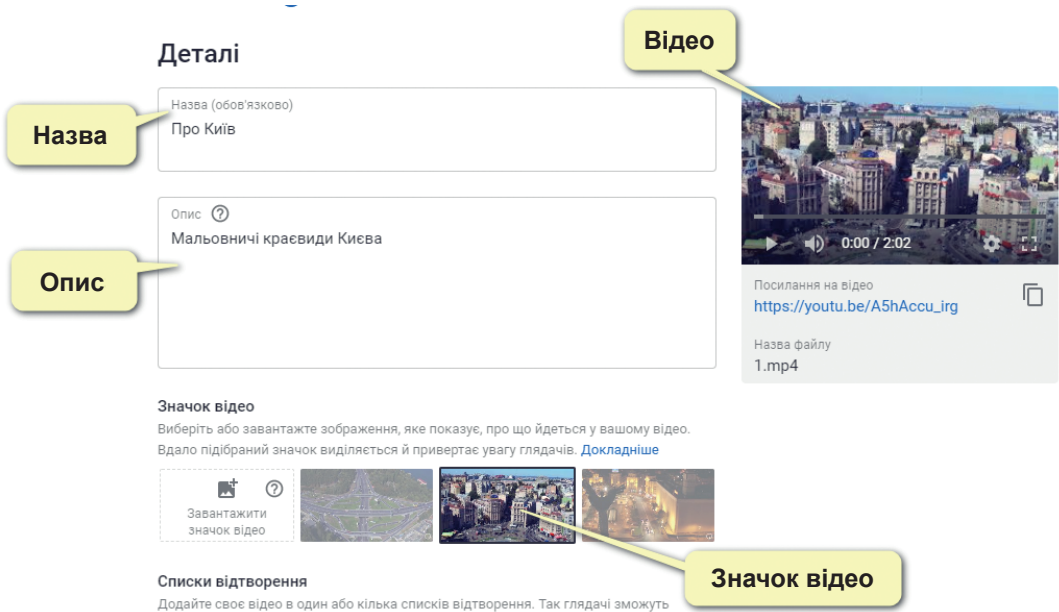
Далі розглянемо, як завантажити відео на свій канал.

1. Натисніть кнопку **Створити** та оберіть команду **Завантажити відео**.
2. Оберіть інструмент завантаження відео або кнопку **Вибрати файл**.
3. У діалоговому вікні оберіть власне відео з комп'ютера. В результаті відео почне завантажуватися (час завантаження залежить від обсягу відео).



- У вікні, що відкриється (рис. 34.1), введть назву відео та опис до відео.
- Оберіть в області ескізів зображення, що якнайбільше підходить до початкового ескізу вашого відео.
- Наступний параметр, який необхідно встановити, — це доступ до даного відео дитячої аудиторії. Тут можна встановити два види доступу:

**Так, цей контент призначено для дітей.** Буде доступний для спільного перегляду, але для відео буде відключено коментарі.



**Рис. 34.1.** Початкові налаштування завантаженого відео

**Ні, цей контент не призначено для дітей.** Цей параметр містить вибір однієї з двох опцій:

- **Так, показувати лише глядачам від 18 років** (цей контент для перегляду дітям буде недоступний);
  - **Ні, відео підходить для глядачів до 18 років** (цей контент можна переглядати усім користувачам та коментувати його).
- Пропустіть вікно, що надає можливість додавати деякі елементи до відео, і ще раз натисніть **Далі**.
  - У наступному вікні необхідно встановити рівні доступу до відео: **Приватне, Не для всіх та Для всіх**). Також можна запланувати демонстрацію відео у вказаний час. Установіть значення **Для всіх**.
  - Для публікації відео натисніть **Опублікувати**.



Після публікації відео на нього буде створено посилання, яке можна вставити на власний сайт або надіслати знайомим електронною поштою. Для цього його достатньо скопіювати до буфера обміну та вставити до листа або в редакторі сайту. Також цим відео можна поділитися в соціальній мережі, обравши піктограму бажаної соціальної мережі.

10. Для доступу до здійснення додаткових налаштувань або одержання статистики з відвідування відео наведіть на нього. В результаті біля відео будуть розташовані необхідні інструменти (рис. 34.2). Самостійно ознайомтеся з їхніми можливостями.

Для відтворення завантаженого відео достатньо клацнути на його ескізі.

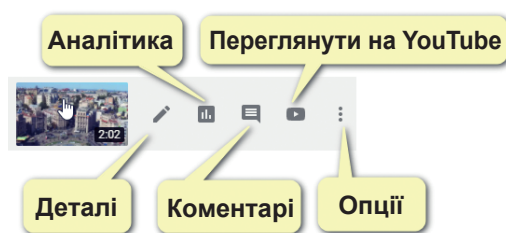


Рис. 34.2. Інструменти для налаштування завантаженого відео

#### ВПРАВА 34



Завантажити на YouTube відео, створене під час виконання практичних завдань із §33.

1. Відкрийте сторінку YouTube в браузері на комп'ютері.
2. Увійдіть у свій акаунт.
3. Натисніть кнопку **Створити** та оберіть команду **Завантажити відео**.
4. Натисніть кнопку **Вибрати файли**.
5. У діалоговому вікні оберіть власне відео з комп'ютера.
6. Уведіть назву та опис до відео.
7. Оберіть в області ескізів зображення, що якнайбільше підходить до початкового ескізу вашого відео.
8. Установіть доступ до даного відео дитячої аудиторії, обравши опцію **Ні, цей контент не призначено для дітей**, а у списку **Вікове обмеження** встановіть опцію **Ні, відео підходить для глядачів до 18 років**. Натисніть двічі кнопку **Далі**.
9. У списку **Зберегти або Опублікувати** встановіть спільний доступ, обравши опцію **Для всіх**.
10. Для публікації відео натисніть **Опублікувати**. В результаті має з'явитися вікно, в якому можна обрати спосіб поширення власного відео.
11. Скопіюйте посилання вашого відео та надішліть на електронну адресу вчителя для перевірки.
12. Натисніть кнопку **Закрити**. Опубліковане відео буде відображено у списку відео вашого каналу.



### Контрольні запитання та завдання

1. Які служби розміщення відеоконтенту ви знаєте? Чи користувалися ви ними? Якщо так, опишіть роботу з цими службами.
2. Як ви впорядковуєте свою колекцію знімків?
3. Опишіть можливості відеохостингу YouTube.
4. Як здійснити завантаження відео з YouTube?



### Питання для роздумів

- 1\*. В якому напрямку, на вашу думку, розвиватимуться платформи відеохостингу?
2. Яких функцій, на ваш погляд, не вистачає відеохостингу YouTube?



### Завдання для досліджень

1. Дізнайтеся, як підписатися на канал в YouTube та як відписатися від нього.
- 2\*. Які можливості редагування відео містить вбудований редактор YouTube?
- 3\*. Знайдіть інформацію про відеохостинг Vimeo.
- 4\*. Як, завантажуючи власні відео на YouTube, можна заробляти гроші?



Пройдіть  
тест



### Висновки

**Мультимедіа** — це програмний продукт, що містить колекції зображень, текстів, а також даних, які супроводжуються звуком, відео, анімацією та іншими візуальними ефектами, оснащений інтерактивним інтерфейсом з елементами керування.

**Захоплення відео** — процес перетворення відеосигналу із зовнішнього джерела на цифровий потік і запис його у відеофайл з метою обробки, зберігання та відтворення.

**Конвертер** — це програма, що перетворює файл з одного формату в інший.

Для здійснення захоплення відео зручно використовувати програму **FastStone Capture**.

Найбільш поширеним хостингом для збереження зображень та фотографій є <https://photos.google.com>.

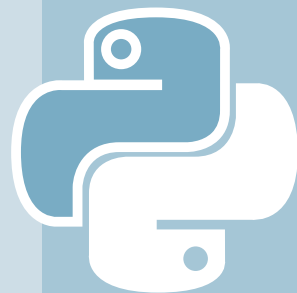
Для завантаження та зберігання аудіо- та відеофайлів найбільшої популярності набув відеохостинг **YouTube**.



Розгадайте електронний кросворд

# Розділ 5

## Алгоритми та програми



### §35.

## Поняття мови програмування

### 35.1.

## ПОНЯТТЯ МОВИ ПРОГРАМУВАННЯ

У даній темі ми розпочинаємо вивчення основ програмування. Вивчаючи мову програмування, ви зможете в майбутньому навчитися складати програми, що прискорюють обробку інформації в будь-яких галузях, і, в подальшому, оволодіти професією програміста/програмістки.



Ознайомтеся з історичними відомостями про мови програмування

Обробляючи інформацію за допомогою програмного засобу, користувач може самостійно виконати послідовність дій, а може записати цю послідовність дій на мові, «зрозумілій» комп'ютеру, та надати вказівку щодо її виконання. Таку мову ще називають *мовою програмування*.

**Мова програмування** — формалізована мова для запису алгоритмів, призначених для виконання за допомогою комп'ютера.

Мова програмування (як і будь-яка інша мова) має три складові: алфавіт, синтаксис і семантику.

*Алфавіт* — фіксований для даної мови набір символів (літери, цифри, спеціальні символи).

*Синтаксис* — набір правил, що встановлюють, які комбінації символів є осмисленими реченнями на цій мові.

*Семантика* — визначає смислове значення речень у мові.



Дізнайтеся більше про мови програмування та їхню класифікацію



## 35.2.

## СИСТЕМА ПРОГРАМУВАННЯ

Для створення програми, «зрозумілої» комп'ютеру, потрібний комплекс програмних засобів, який називають *системою програмування*.

**Система програмування** — система для створення та редагування програм певною мовою програмування.

Основні програми, що мають входити в систему програмування:

- *текстовий редактор* (для написання програми);
- *транслятор* (для перетворення програми в машинний код);
- *засоби автоматизації* (вбудовані команди, бібліотеки тощо);
- *налагоджувач* (для перевірки та виправлення помилок програми).

З поняттям текстового редактора ви вже ознайомлені. Розглянемо, що таке транслятор та які види трансляторів можуть бути використані в системі програмування.

Оскільки комп'ютер має можливість обробляти інформацію, подану лише у двійковому коді, то має існувати спеціальна програма, що перекладає зрозумілі для людини символи в символи, зрозумілі машинам.

**Транслятор** — це програмний засіб, що здійснює перетворення програми з однієї мови програмування в програму на іншій мові програмування.

Транслятори бувають двох видів: інтерпретатори та компілятори.







*Інтерпретатор* перетворює та виконує програму по рядках, а *компілятор* читає програму цілком і перекладає її в машинний код, який потім і виконується.

**35.3.****СУЧАСНІ МОВИ ПРОГРАМУВАННЯ**

Ми розібралися з вами, що таке мова програмування. Насправді мов програмування існує дуже багато. До того ж вони мають різне призначення.

Найбільшої популярності на сьогоднішній день набули такі мови програмування, як JavaScript, PHP, C++, Python. Велику популярність мають також мови C#, Go, Swift, Rust, Kotlin тощо.



Дізнайтеся більше про призначення сучасних мов програмування

У цьому уроці ми розпочали ознайомлення із найцікавішим, але і з найважчим розділом інформатики — програмуванням. На наступних уроках ви будете вивчати мову програмування Python, що є сучасною мовою програмування високого рівня.

**Контрольні запитання та завдання**

1. Що таке мова програмування?
2. Що таке система програмування? Назвіть її складові.
3. Що таке транслятор? Назвіть види трансляторів.
4. Які вам відомі сучасні мови програмування? Яке їхнє основне призначення?

**Питання для роздумів**

- 1\*. Що робить програміст; тестувальник?
- 2\*. Чому існує така велика кількість мов програмування?

**Завдання для досліджень**

- 1\*. Підготуйте повідомлення про будь-яку мову програмування (засновник мови, з якою метою створювалася мова, яке має призначення тощо).



## §36.

## Знайомство з середовищем програмування

## 36.1.

## ЗНАЙОМСТВО З ІНТЕРАКТИВНИМ РЕЖИМОМ

Розпочнемо знайомство з однією із найбільш популярних мов програмування — мовою Python. Існує багато середовищ, які підтримують програмування на мові Python. Найбільш простим є **Python IDLE**. Скачати його можна безкоштовно на сайті [www.python.org](http://www.python.org).

Також створювати, редагувати та виконувати програми можна в режимі онлайн. Ось одне із посилань для використання такого способу:

<https://www.programiz.com/python-programming/online-compiler>.

Після завантаження та встановлення Python на комп'ютері в меню **Пуск** буде створено посилання на папку з файлами. Для запуску Python слід обрати **IDLE Python**.

Після запуску програми буде виведено наступне вікно.

```
Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40)
[MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more
information.
>>> |
```

Три кутові дужки свідчать про запрошення до введення команд програми.

Якщо у даному середовищі ввести, наприклад,  $3+5$  та натиснути **Enter**, тоді нижче буде виведено результат обчислення, а саме, 8.

```
>>> 3+5
8
```

Самостійно спробуйте ввести й інші приклади, використовуючи арифметичні операції віднімання (-), множення (\*), ділення (/).

Якщо замість числових виразів написати деякий текст, наприклад, «Привіт!» та натиснути **Enter**, система повідомить про помилку. Адже для виведення тексту необхідно скористатися спеціальною командою `print`. Для виведення напису «Привіт!» програмний код матиме вигляд:

```
print("Привіт!")
```



Після введення команди слід натиснути **Enter**. У результаті буде виведено слово «Привіт!».

Для виведення деякого числа, наприклад, 5 також можна скористатися командою `print`, записавши число в дужках, але вже без лапок. Після натиснення **Enter** буде виведено число 5.

```
>>> print("Привіт!")
Привіт!
>>> print(5)
5
```

Розглянемо детальніше синтаксис команди `print`. Спочатку записується службове слово `print`. Далі в дужках записують те, що треба вивести на друк. Зауважимо, що текст записують у лапках, а числа — без лапок. Якщо необхідно вивести у рядок декілька об'єктів, їх відокремлюють комами. Також зверніть увагу, що службове слово `print` записують маленькими латинськими літерами.

```
print(об'єкт 1, об'єкт 2, ...)
```

Так, для виведення прикладу «3+5=» та результату обчислення 8 слід скористатися командою: `print("3 + 5 =", 3+5)`.

У результаті програма виводить приклад, а після знака «дорівнює» — результат виконання.

```
>>> print("3+5=", 3+5)
3+5= 8
```

Розглянемо випадок, коли нам необхідно вивести, наприклад, таку фразу:

```
Книга "Казки" коштує 100 грн.
```

У цьому випадку всю фразу необхідно обмежити не лапками, а апострофами. Команда для виведення такої фрази буде мати вигляд:

```
print('Книга "Казки" коштує 100 грн.')
```

Отже, написи в команді `print` можна обмежувати лапками або апострофами. Відповідно команди `print('Привіт!')` та `print("Привіт!")` для середовища Python є однаковими.

## 36.2.

### СТВОРЕННЯ ТА ВИКОНАННЯ ФАЙЛІВ ПРОГРАМНОГО КОДУ

Інтерактивний режим IDLE не є основним. Як правило, ми будемо створювати програми, які складаються з декількох команд, зберігати їх, а потім уже запускати на виконання.



Для створення нового документа слід обрати в меню **File** команду **New File**. В результаті буде створено новий документ. Після цього у його вікні слід увести програмний код, наприклад,

```
print("5+4=",5+4)
```

Для запуску програми на виконання в меню **Run** слід обрати команду **Run Modul** або натиснути клавішу **F5**. У результаті має з'явитися діалогове вікно, в якому необхідно підтвердити або скасувати збереження файлу. Після збереження файлу відбувається виконання програми у вікні інтерактивного режиму. За умовчанням файл зберігається у тій папці, до якої встановлено сам Python і має розширення **.py**.

Відкрити створений нами файл для подальшого редагування або виконання можна таким чином:

### 1 спосіб.

- 1) У вікні **IDLE Python** в меню **File** обрати команду **Open**.
- 2) Обрати місце знаходження файлу.
- 3) Натиснути **Відкрити**.

### 2 спосіб.

- 1) Відкрити папку з файлом.
- 2) У контекстному меню файлу обрати команду **Edit With IDLE** ▶ **Edit With IDLE**.

У результаті файл буде відкрито для редагування.

## ВПРАВА 36.1

1 x 5 = 5  
2 x 5 = 10  
3 x 5 = 15  
4 x 5 = 20  
5 x 5 = 25  
6 x 5 = 30  
7 x 5 = 35  
8 x 5 = 40  
9 x 5 = 45



**Завдання.** Створити програму виведення таблиці множення на 5 у наступному вигляді.

1. Створіть новий файл.
2. Уведіть команду `print("1 x 5 =",1*5)`.
3. Скопіюйте її ще 8 разів.
4. Змініть цифру 1 на 2, 3, і т. д. до 9.
5. Запустіть та збережіть програму. У результаті має бути виведено таблицю множення на 5 відповідно до зразка.

## ВПРАВА 36.2



**Завдання.** Із використанням середовища Python IDLE створіть програму виведення умови та розв'язання задачі у наступному вигляді.

**ЗАДАЧА**

В Андрія 32 журнали "Пізнайко" і 44 журнали "Барвінок", а в Юри – на 23 журнали менше, ніж є усього журналів в Андрія. Скільки є журналів у Юри?

**Розв'язання**

- 1)  $32+44 = 76$  (ж.) – всього у Андрія;
- 2)  $76-23 = 53$  (ж.)

Відповідь: у Юри є 53 журнали.

**Зауваження.**

1. Числа, що є результатом обчислення виразів, мають бути обчислені засобами мови Python.
2. Дотримуйтесь усіх відступів та інтервалів.
3. Відповідні рядки повинні мати лише ті літери, цифри та символи, що представлені у зразку.

**Контрольні запитання та завдання**

1. Де можна створювати та редагувати програми на мові Python?
2. Яка команда призначена для виведення даних?
3. Як вивести текстову інформацію; як вивести числа?
4. Як створити, зберегти та запустити програму на виконання?

**Питання для роздумів**

- 1\*. Для чого в інтерактивному режимі реалізована можливість виконання окремих команд?
- 2\*. Чи можна при створенні різних програм використовувати лише інтерактивний режим?
- 3\*. Навіщо при використанні команди `print` для виведення тексту реалізована можливість обмеження тексту як лапками, так і апострофами?

**Завдання для досліджень**

- 1\*. Які переваги та недоліки створення та редагування програм на мові Python в режимах офлайн та онлайн?



Пройдіть  
тест



## §37.

## Типи даних у програмуванні. Присвоєння значень величинам

## 37.1.

### АЛФАВІТ МОВИ ПРОГРАМУВАННЯ

Вам уже відомо, що однією зі складових мови програмування є алфавіт. До алфавіту мови програмування Python входять:

- **латинські** літери від *a* до *z* та від *A* до *Z*;
- **цифри** від 0 до 9;
- **спеціальні** символи (+, -, \*, / тощо);
- **зарезервовані** (службові слова): *if*, *for*, *def*, *print* тощо.

## 37.2.

### ПОНЯТТЯ ВЕЛИЧИН

Об'єкти, над якими виконуються певні операції в середовищі Python, прийнято називати *величинами*. Величини характеризуються іменем, типом і значенням.

Для запису імен величин використовують правила:

- **першим** символом в імені має бути літера або символ підкреслення;
- **ім'я** може складатися з букв, цифр та символу підкреслення «\_»;
- **ім'я** має бути унікальним і не збігатися з ключовими словами — такими як **if**, **print**, **for** тощо;
- **ім'я** не може містити пробілів;
- **однакові** імена в різних регістрах сприймаються програмою, як різні імена (тобто, імена величин **A1** та **a1** є різними іменами).

У редакторі IDLE правильні імена змінних відображені чорним кольором. Якщо ім'я змінної є іншого кольору, його потрібно змінити.

Величини можуть бути різних типів. Залежно від того, який тип має величина, для неї виділяється певний обсяг у байтах на ділянці пам'яті. У таблиці 37.1. представлено основні типи величин, що використовуються при створенні програм.

Таблиця 37.1. Типи величин

Тип	Опис	Приклади
int	Цілі числа	5, 327, 0, -15
float	Дійсні числа	1.35, -3.14, 0.0125
str	Рядки	"hello", "0556", 'Привіт', '8-A'

Зверніть увагу, що для розділення цілої та дробової частин у Python використовується крапка, а не кома.

Зауважимо, що в більшості мов програмування тип величин необхідно оголошувати перед їхнім використанням. У мові Python цього



робити не потрібно, оскільки у ній використовується динамічна типізація змінних. Це означає, що у процесі присвоєння змінним значень їхній тип визначається автоматично.

Величини, які під час виконання програми можуть набувати різних значень, називають **змінними**. Надання змінній деякого значення називається **командою присвоєння**.

У мові Python присвоєння позначають знаком «дорівнює» і змінні створюються при першому присвоєнні їм значень. Загальний вигляд команди присвоєння є таким:

```
змінна=значення_або_вираз
```

При присвоєнні ліва частина завжди є іменем змінної, а права — значенням або виразом. Змінні корисні тим, що вони зберігають дані, які потім можна використати у програмі.

Розглянемо приклади надання значень змінним.

`x=5` — змінна `x` одержує значення **5** типу **int**.

`y=5.5` — змінна `x` одержує значення **5.5** типу **float**.

`d=-d` — значення змінної `d` буде замінено на протилежне за знаком значення.

`x=x+1` — поточне значення змінної `x` буде збільшено на 1.

`q="Hello"` — змінна `q` одержує значення **Hello** типу **str**.

`s='505'` — змінна `s` одержує значення **505** типу **str**.

В одному рядку можна присвоїти однакові значення кільком змінним, наприклад, `a=b=c=5`.

Також одним оператором присвоєння можна присвоїти різні значення. В цьому випадку змінні та їхні значення відокремлюють комами, наприклад: `a,b,c=1,2,3`.

Якщо тип об'єкта, який програма самостійно визначила, не влаштовує програміста, можна скористатися функціями перетворення одного типу даних на інший, що представлені в таблиці 37.2.

**Таблиця 37.2.** Функції перетворення типів об'єкта

Функція	Призначення	Приклад
<code>int(об'єкт)</code>	перетворення об'єкта на ціле число	<code>int(5.7)=5</code> <code>int("5")=5</code>
<code>float(об'єкт)</code>	перетворення об'єкта на дійсне число	<code>float(5)=5.0</code> <code>float("5.7")=5.7</code>





## 37.3.

## УВЕДЕННЯ ЗНАЧЕНЬ КОРИСТУВАЧЕМ

Ми розглянули з вами присвоєння значень змінним безпосередньо з допомогою програмного коду, але значення змінним може надавати і користувач програми. Зробити це можна з допомогою функції `input()`. Загальний вигляд надання змінній значення, введеного користувачем, є таким:

```
змінна=input(запрошення)
```

Наприклад, для надання змінній `a` деякого числа можна скористатися наступною командою: `a=input("Уведіть число, a=")`.

Тут `a` — це змінна, а в лапках записано запрошення.

Після виконання цього програмного коду у вікні IDLE з'явиться запрошення, і програма буде очікувати введення значення користувачем.

```
Уведіть число, a=
```

Розглянемо приклад програмного коду, з допомогою якого введене користувачем число збільшувалося б у 3 рази.

1. У вікні IDLE створіть новий файл та введіть код:

```
a=input("Уведіть число, a=")
print("Число, збільшене у 3 рази =", a*3)
```

2. Запустіть файл на виконання. В результаті має з'явитися запрошення для введення числа.

3. Уведіть, наприклад, число 7.

4. Натисніть **Enter**. В результаті як відповідь ви маєте одержати 777.

Чому 777, а не 21? Це відбувається тому, що програма не додає числа, а записує поряд три рази. Чому так відбувається?

Уведене користувачем значення за умовчанням сприймається програмою як об'єкт текстового типу. Аби програма сприймала значення як число слід скористатися функцією `int` або `float`, що перетворює текстовий тип на числовий. У даному випадку більше підійде функція `int`. Отже, код, який ми маємо ввести, буде мати наступний вигляд:

```
a=int(input("Уведіть число, a="))
```

5. Змініть програмний код, додавши функцію `int`.

6. Запустіть програму.

7. Після запрошення програми введіть число 7.

8. Натисніть **Enter**. В результаті ви маєте одержати число 21.



## ВПРАВА 37



**Завдання.** Створіть програму додавання двох чисел, уведених користувачем.

1. Створіть новий файл в середовищі Python IDLE.
2. Введіть команду для введення першого числа:  
`a=int(input("Перше число ="))`
3. Скопіюйте введену команду нижче та змініть її відповідним чином:  
`b=int(input("Друге число ="))`
4. Уведіть команду обчислення суми двох чисел:  
`s=a+b`
5. Додайте команду виведення результату обчислень:  
`print("s=",s)`
6. Збережіть програму з іменем **Вправа\_37.py**.
7. Запустіть програму та перевірте правильність її виконання.



## Контрольні запитання та завдання

1. Що таке величина? Чим вона характеризується?
2. Які правила запису імен величин?
3. Назвіть основні типи величин.
4. Що таке змінна?
5. Опишіть порядок виконання оператора присвоєння. Наведіть приклади команд присвоєння.
6. За допомогою яких функцій можна переводити змінні одного типу на інший?
7. З допомогою якої функції можна ввести значення змінної користувачем? Який синтаксис цієї функції?



## Питання для роздумів

- 1\*. Чи обмежена кількість змінних у програмі?
- 2\*. Запишіть оператор присвоєння для збільшення змінної  $a$  у 2 рази; зменшення змінної  $b$  у 2 рази.



## Завдання для досліджень

- 1\*. Визначте, які ще типи змінних використовують у середовищі програмування. З'ясуйте їхні допустимі значення.
- 2\*. Що відбуватиметься, якщо функцію `input` використати без запитання?



Пройдіть  
тест



## §38.

## Використання арифметичних операцій

## 38.1.

## АРИФМЕТИЧНІ ОПЕРАЦІЇ

Часто при обробці даних використовують числові величини цілого та дійсного типів. Щоб описати дії з числовими даними, використовують арифметичні вирази. Вони дають змогу використовувати різноманітні математичні формули. Користувач самостійно створює арифметичні вирази, комбінуючи числа, змінні, функції та поєднуючи їх із символами арифметичних операцій. Арифметичні операції та їхній пріоритет представлені в таблиці 38.1.

Таблиця 38.1. Арифметичні операції

Знак операції	Операція	Приклад	Пріоритет операції
+	Додавання	$2 + 3 = 5$	3
-	Віднімання	$3 - 2 = 1$	3
*	Множення	$2 * 3 = 6$	2
/	Ділення	$6 / 2 = 3$	2
//	Ділення без остачі	$15 // 10 = 1$	2
%	Остача від ділення	$15 \% 10 = 5$	2
**	Піднесення до степеня	$3 ** 2 = 9$	1

Вирази, що використовують арифметичні операції, записують в один рядок, розставляючи дужки з урахуванням пріоритету виконання операцій. Наприклад, для запису в мові Python виразу

$$a = \frac{25 + 8}{46 - 12} + 3^2, \text{ слід скористатися таким програмним кодом:}$$

$$a = (25 + 8) / (46 - 12) + 3 ** 2.$$

Зверніть увагу, якщо в арифметичній операції беруть участь цілі числа, то результатом обчислення буде завжди ціле число. Виключенням тут є лише звичайна дія ділення. Результатом дії ділення буде завжди дійсне число. Так, наприклад,  $4/2 = 2.0$ .

Якщо ж при виконанні арифметичної операції одне число є цілим, а друге — дійсним, то спочатку ціле число буде перетворено на дійсний тип, а потім буде здійснюватися операція над дійсними числами. Звичайно, результат у цьому випадку буде дійсним числом.

Наприклад,  $2 + 3.0 = 5.0$ ,  $7.0 + 5 = 12.0$ .



Дізнайтеся більше про спрощення арифметичних виразів



## 38.2. РОЗВ'ЯЗУВАННЯ ЗАДАЧ НА ОБЧИСЛЕННЯ

Розглянемо використання арифметичних виразів, розв'язавши таку задачу на обчислення.

**Задача.** Дано сторони прямокутника  $a$  і  $b$ . Обчислити площу та периметр прямокутника.

Створюючи програми, які потребують обчислень, бажано дотримуватися певної послідовності дій, що буде представлена при виконанні цього завдання. В залежності від складності розв'язання задачі деякі дії можна пропускати.

### I. Визначимо вхідні та вихідні дані.

*Вхідними* даними є значення змінних  $a$  та  $b$ , що є сторонами прямокутника.

*Вихідними* — значення змінних  $S$  та  $P$ , які позначають площу та периметр прямокутника.

**II. Визначимо математичну модель розв'язання задачі** (в даному випадку математична модель — це формули, за якими дана задача має бути розв'язана).

$S = a \cdot b$  — обчислення площі.

$P = 2(a + b)$  — обчислення периметра.

### III. Створення програмного коду.

1. Уведіть команди для введення значень сторін:

```
a=int(input("a="))
b=int(input("b="))
```

2. Уведіть формули для обчислень:

```
S=a*b
P=2*(a+b)
```

3. Додайте коди виведення результату:

```
print("S=",S)
print("P=",P)
```

4. Запустіть програму.

5. Переконайтеся у правильності її виконання. Для цього введіть значення першої сторони — 5, а другої сторони — 6. В результаті ви маєте одержати результат:  $S = 30$ ,  $P = 22$ . Звичайно, для перевірки працездатності більш складних програм одного набору значень не достатньо.

Процес перевірки правильності виконання програми за різних умов і значень називають *тестуванням програми*.

Зверніть увагу, що в більшості випадків код програми з використанням змінних складається з трьох частин:



- **уведення** даних;
- **обробка** даних;
- **виведення** результату.

В залежності від умови задачі деякі частини можуть бути пропущені.

```
Уведення даних (a=int(input("a="))
 b=int(input("b="))
Обробка даних (S=a*b
 P=2*(a+b)
Виведення результату (print("S=",S)
 print("P=",P)
```

### ВПРАВА 38.1



**Завдання.** Створіть програму обчислення середнього арифметичного трьох уведених користувачем чисел.

(Формула для обчислення:  $s = \frac{a + b + c}{3}$ .)

### ВПРАВА 38.2



**Завдання.** Створіть програму обчислення об'єму та площі поверхні куба, якщо його ребро дорівнює  $a$ .

(Формули для обчислення:  $V = a^3$ ,  $S = 6a^2$ .)

### ВПРАВА 38.3



**Завдання.** Створіть програму розв'язання задачі: «Дівчинка збирала  $k$  кошиків винограду вагою  $m$  кг  $n$  г у кошику та  $a$  кошиків вагою  $b$  кг  $c$  г у кошику. Обчисліть вагу винограду, що збирала дівчинка».

Результат виведіть у наступному вигляді:

Дівчинка збирала 2 кг 230 г винограду.



### Контрольні запитання та завдання

1. Які арифметичні операції використовуються у Python? Вкажіть їхні пріоритети.
2. Які формули використовують для спрощення арифметичних операцій?
3. Що таке тестування програми?
4. Яких основних етапів слід дотримуватися при створенні програм на обчислення?
5. З яких основних частин складається програма на обчислення? Чи завжди програма має містити усі ці частини?

**Питання для роздумів**

- 1\*. Який буде тип результату обчислення виразів:  $9.0/3.0$ ;  $15/3$ ;  $2.5*2$ ;  $7*5$ ?
- 2\*. Як з допомогою арифметичних операцій знайти першу цифру двоцифрового числа; останню цифру будь-якого числа?

**Завдання для досліджень**

- 1\*. Ураховуючи пріоритет арифметичних операцій, розставте дужки у виразі так, щоб у результаті отримати 10.  
 $2**2+3-10+1*2$ .

**§39****Розв'язування задач з арифметичними операціями****39.1.****ЗНАХОДЖЕННЯ ЦИФР ЧИСЛА**

У програмуванні досить часто розв'язують задачі, в яких необхідно знайти цифри числа. Для цих дій використовують арифметичні операції знаходження цілої частини від ділення ( $//$ ) та остачі від ділення ( $\%$ ).

Використовуючі ці арифметичні операції, можна вивести формули, що дозволять визначити цифру числа будь-якої розрядності.

У таблиці нижче наведено формули для знаходження різних цифр чисел.

**Таблиця 39.1.** Знаходження цифр цілого числа  $a$

Необхідно знайти	Вираз	Приклади
Остання цифра числа	$a\%10$	$18\%10=8$
Знаходження першої цифри довільного числа	$a//10^{n-1}$ , де $n$ — розрядність числа	$123456//10^{5}=1$
Знаходження другої цифри трицифрового числа	$a//10\%10$	$736//10=73 \rightarrow 73\%10=3$
Знаходження другої цифри чотирицифрового числа	$a//100\%10$	$7362//100=73 \rightarrow 73\%10=3$
Знаходження третьої цифри чотирицифрового числа	$a//10\%10$	$7362//10=736 \rightarrow 736\%10=6$



## ВПРАВА 39.1



**Завдання.** Створити програму обчислення суми квадратів цифр цілого трицифрового числа.

Так, для даного числа 125 —  $1^2 + 2^2 + 5^2 = 30$ .

**I. Визначення вхідних та вихідних даних.**

*Вхідними даними* є трицифрове число, позначимо його змінною  $a$ .

*Вихідними даними* буде значення суми квадратів цифр числа, позначимо його змінною  $S$ . Для знаходження цифр числа скористаємося проміжними змінними  $a_1$ ,  $a_2$ ,  $a_3$ , що відповідають першій, другій та третій цифрам відповідно.

**II. Визначення математичної моделі розв'язання задачі.**

$a_1 = a // 100$  — перша цифра числа;

$a_2 = a // 10 \% 10$  — друга цифра числа;

$a_3 = a \% 10$  — третя цифра числа;

$S = a_1^2 + a_2^2 + a_3^2$ .

**III. Створення програмного коду.**

За даною математичною моделлю створимо програмний код:

```
#Введення даних
a=int(input("Уведіть трицифрове число, a="))
a1=a//100 #Перша цифра числа
a2=a//10%10 #Друга цифра числа
a3=a%10 #Третя цифра числа
#Обчислення суми квадратів цифр числа
S=a1**2+a2**2+a3**2
#Виведення результату
print("Сума квадратів числа, S=",S)
```

До речі, зверніть увагу на текст, виділений у програмі червоним. Таким чином у коді позначають *коментар*.

**Коментар** — це рядок, який не виконується програмою, в Python він починається символом решітки. Може бути записаний як в окремому рядку, так і наприкінці рядка коду. Коментарі використовують для пояснення коду програми або для вилучення з алгоритму виконання програми деякого незавершеного коду.

Уведіть програмний код та запустіть створену програму на виконання. Введіть число 241. В результаті ви маєте одержати 21. Оскільки,  $2^2 + 4^2 + 1^2 = 4 + 16 + 1 = 21$ .



Виконайте вправи  
39.2–39.6



**Контрольні запитання та завдання**

1. Як знайти останню цифру числа?
2. Як знайти першу цифру будь-якого числа?
3. Що таке коментар? Коли його використовують?

**Питання для роздумів**

- 1\*. Для розв'язування яких задач можна використати арифметичні операції цілочисельного ділення?

**Завдання для досліджень**

- 1\*. Укажіть вирази, за допомогою яких можна визначити цифри п'ятицифрового числа; шестицифрового числа.
- 2\*. Визначте загальну формулу для знаходження другої цифри довільного числа.

Пройдіть  
тест**§40.****Стандартні математичні функції****40.1.****ВБУДОВАНІ ФУНКЦІЇ**

Для виконання більш складних обчислень у математиці використовують різноманітні функції. У мові програмування Python також є власні математичні функції. З можливостями їхнього використання ми познайомимося далі.

Для роботи з числами можна використовувати функції, що є вбудованими до мови Python. З деякими з них ви вже знайомі (`int`, `float` тощо). Розглянемо ще декілька найбільш уживаних функцій, що представлені у таблиці 40.1.

**Таблиця 40.1.** Найбільш уживані вбудовані математичні функції

Функція	Пояснення	Приклади
<code>round(x, n)</code>	Округлення числа, де $x$ — число, $n$ — кількість цифр дробової частини	<code>round(5.57, 1)=5.6</code> <code>round(3.234, 2)=3.23</code>
<code>abs(x)</code>	Модуль числа $x$	<code>abs(-5)=5</code>
<code>max(x1, x2, x3, ...)</code>	Знаходження максимального значення	<code>max(3, 5, 2)=5</code>
<code>min(x1, x2, x3, ...)</code>	Знаходження мінімального значення	<code>max(3, 5, 2)=2</code>



Зверніть увагу, що аргументи функцій завжди беруть у дужки. Аргументом може бути *число, змінна, арифметичний вираз або інша функція*.

## 40.2.

## ВИКОРИСТАННЯ МОДУЛЯ MATH

Крім вбудованих функцій, у Python використовують математичні функції, що входять до модуля **math**. Найбільш уживані математичні функції представлені в таблиці 40.2.

Таблиця 40.2. Найбільш уживані функції модуля *math*

Функція	Математичний запис	Пояснення	Приклади
sqrt(x)	$\sqrt{x}$	Квадратний корінь	sqrt(25) = 5
ceil(x)		Найближче більше ціле число	ceil(1.1)=2
floor(x)		Найближче менше ціле число	floor(1.1)=1
sin(x)	$\sin x$	Синус числа	sin(1.3)=0.96
cos(x)	$\cos x$	Косинус числа	cos(3.14)=0.99
tan(x)	$\operatorname{tg} x$	Тангенс числа	tan(1.5)=14.1

Для роботи з цими функціями необхідно імпортувати модуль у програму за допомогою команди: `import math`

Після цього звернення до функції слід записувати з префіксом **math**. Ось таким буде код обчислення кореня від числа 25:

```
>>>import math
>>>math.sqrt(25)
5.0
```

Якщо математичних функцій у програмі багато, можна записати команду імпорту в наступному вигляді: `from math import*`.

У цьому випадку префікс **math** перед математичними функціями записувати не потрібно. Отже, програмний код обчислення квадратного кореня від 25 буде мати вигляд:

```
>>>from math import*
>>>sqrt(25)
5.0
```

## ВПРАВА 40.1



**Завдання.** Дано цілі значення  $a$ ,  $b$ ,  $c$ . Обчислити значення  $x$  та  $y$  за наведеними формулами:

$$\text{а) } x = \frac{\sqrt{a+1} - |c|}{b^2 + 1};$$

$$\text{б) } y = \sqrt{abc + \frac{a}{b}} + |a|.$$



З умови зрозуміло, що в даному завданні вхідними даними будуть значення змінних  $a$ ,  $b$  і  $c$ , вихідними — значення змінних  $x$  та  $y$ .

Уведемо програмний код.

1. Оскільки у формулах використовуються математичні формули із модуля `math`, імпортуйте його до програми.

```
from math import*
```

2. Уведіть значення змінних  $a$ ,  $b$  і  $c$ .

```
a=int(input("a="))
b=int(input("b="))
c=int(input("c="))
```

3. Уведіть формулу для знаходження змінної  $x$ .

```
x=(sqrt(a+1)-abs(c))/(b**2+1)
```

4. Уведіть формулу для знаходження  $y$ .

```
y=sqrt(a*b*c+a/b)+abs(a)
```

5. Уведіть коди для виведення результату.

```
print("x=",x)
print("y=",y)
```

6. Перевірте правильність програмного коду. Для  $a = 1$ ,  $b = 1$  та  $c = 1$  ви маєте одержати приблизно такі значення  $x = 0,2071067$ ,  $y = 2,4142135$ .

7. Додайте до результату функції округлення до трьох знаків після коми. Ось таким має бути команда виведення змінної  $x$ .

```
print("x=",round(x,3))
```

8. Для виведення значення змінної  $y$  змініть код самостійно.

9. Перевірте виконання програми після внесення змін. У результаті ви маєте одержати  $x = 0,207$ ,  $y = 2,414$ .

## ВПРАВА 40.2



**Завдання.** Дано цілі значення  $a$ ,  $b$ ,  $c$ . Обчислити значення  $x$  та  $y$  за наведеними формулами:

$$\text{а) } x = \frac{\sqrt{|a+1|} - \sqrt{|b-2|}}{3 + \frac{c^3}{2} + \frac{b^2}{4}}; \quad \text{б) } y = \sqrt{\frac{abc}{3} + |a-b|}.$$

**Контрольні запитання та завдання**

1. Який модуль використовується для застосування математичних функцій? Укажіть способи імпортування цього модуля.
2. Як виконати округлення числа до одного знака після десяткової коми?
3. За допомогою якої функції можна знайти квадратний корінь від числа?

**Питання для роздумів**

- 1\*. Який тип буде мати результат виконання функції `abs(x); sqrt(x)`?
- 2\*. Чим відрізняється використання функції `int()` від `floor()`?

**Завдання для досліджень**

- 1\*. Які математичні функції ще використовуються в мові програмування Python?



Пройдіть тест

**§41.****Практична робота №3. Реалізація лінійних алгоритмів в середовищі програмування****ЗАДАЧА 1**

**Завдання.** Створити програму розв'язання задачі: «Автобус, що рухався зі швидкістю  $a$  км/год, доїхав до міста за  $b$  хвилин. З якою швидкістю повинен рухатися велосипедист, щоб проїхати ту саму відстань до міста за  $c$  годин і  $d$  хвилин?». Відповідь надати в км/год.

- 1) Визначимо вхідні та вихідні дані.  
*Вхідні дані:*  $a$  — швидкість автобуса,  $b$  — час руху автобуса,  $c$  і  $d$  — кількість годин та хвилин — час руху велосипедиста.  
*Вихідним даним* буде значення змінної  $V_v$  — швидкість велосипедиста.
- 2) Складемо математичну модель розв'язання задачі.  
 Для зручності дані представимо у вигляді таблиці.

	Швидкість	Час	Відстань
Автобус	$a$ (км/год)	$b$ (хв)	
Велосипедист	?	$c$ (год) $d$ (хв)	



Визначимо формули для розв'язання задачі:

$T_a = b/60$  — час руху автобуса у годинах;

$S_a = a * T_a$  — відстань, пройдена автобусом;

$T_v = c + d/60$  — час руху велосипедиста у годинах;

$V_v = S_a / T_v$  — швидкість велосипедиста.

В останній формулі, оскільки відстані, пройдені автобусом та велосипедистом, є рівними, в якості відстані велосипедиста підставляємо відстань автобуса.

Далі ви маєте самостійно розробити програмний код, використовуючи створену математичну модель.

## ЗАДАЧА 2



**Завдання.** Створити програму обчислення площі прямокутного трикутника за його катетом та гіпотенузою.

1) Визначимо вхідні та вихідні дані.

*Вхідні дані:*  $a$  — катет,  $c$  — гіпотенуза прямокутного трикутника.

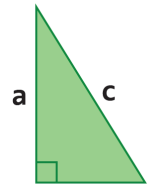
*Вихідне дане:*  $S$  — площа прямокутного трикутника.

2) Створимо математичну модель:

$S = \frac{ab}{2}$  — площа прямокутного трикутника;

$b = \sqrt{c^2 - a^2}$  — другий катет.

Далі ви маєте самостійно розробити програмний код, використовуючи створену математичну модель, зверніть увагу на правильну послідовність використання формул при обчисленні.



Виконайте задачі 3 та 4

## §42.

## Логічні вирази та змінні й операції над ними

### 42.1.

### ПРОСТІ ЛОГІЧНІ ВИРАЗИ

У попередніх класах, вивчаючи Scratch, ви познайомилися з операторами розгалуження (умовними операторами). Їх використовують, коли виконання тієї чи іншої дії залежить від виконання певної умови. Умова в таких операторах у середовищі програмування утворюється з допомогою логічних виразів та змінних. Далі познайомимося з різними видами логічних виразів.

Вибір однієї з двох послідовностей дій залежно від виконання або невиконання деякої умови називають **розгалуженням**.



Алгоритм, що містить одну або кілька умов і, відповідно, дві та більше гілок, називають *алгоритмом з розгалуженням*.

Умова в команді розгалуження формується у вигляді логічного виразу, результатом обчислення якого є значення true (істинний) або false (хибний).

*Простий логічний вираз* — це оператор порівняння, що містить два арифметичні вирази, з'єднані знаком операції. Можливі знаки операцій наведено у таблиці 42.1.

Приклади простих логічних виразів та їхніх значень наведено у таблиці 42.2.

**Таблиця 42.1.** Операції порівняння **Таблиця 42.2.** Приклади простих логічних виразів

Знак операції	Операція	$a = 2, b = 3$	
		Логічний вираз	Значення
==	Дорівнює	$a == b$	False
!=	Не дорівнює	$a + b == 5$	True
>	Більше	$a >= b$	False
<	Менше	$a != b$	True
>=	Більше або дорівнює	$a + b != b - a$	True
<=	Менше або дорівнює		

## 42.2.

### СКЛАДЕНІ ЛОГІЧНІ ВИРАЗИ

Вирази, з'єднані між собою логічними операціями not (не), and (і) та or (або), є *складеними логічними виразами*. Назви цих операцій та пріоритет їхнього виконання наведено у таблиці 42.3.

**Таблиця 42.3.** Логічні операції

Оператор	Назва	Пріоритет
not	Логічне заперечення	1
and	Логічне множення	2
or	Логічне додавання	3

Для визначення істинності або хибності складених логічних виразів використовують так звані таблиці істинності. Розглянемо їх більш детально.

Операція not, її ще називають *логічне заперечення*, при застосуванні до виразу  $x$  дає *хибний* результат, якщо вираз  $x$  є істинним, і навпаки, результат буде *істинним*, якщо вираз  $x$  є хибним (таблиця 42.4.)

Операція and, яку називають *логічне множення* або *кон'юнкція*, що застосовується до виразів  $x$  та  $y$ , дає *істинний* результат, якщо обидва вирази є істинними, в інших випадках результат буде *хибним* (таблиця 42.5).



**Таблиця 42.4.** Таблиця істинності операції not

x	not x
true	false
false	true

**Таблиця 42.5.** Таблиця істинності операції and

x	y	x and y
true	true	true
true	false	false
false	true	false
false	false	false

Операція or, яку називають *логічне додавання* або *диз'юнкція*, що застосовується до виразів  $x$  та  $y$ , дає *істинний* результат, якщо принаймні один вираз є істинним, якщо обидва вирази є хибними, результат буде *хибним* (таблиця 42.6).

Приклади складених логічних виразів та їхніх значень наведено у таблиці 42.7.

**Таблиця 42.6.** Таблиця істинності операції or

x	y	x or y
true	true	true
true	false	true
false	true	true
false	false	false

**Таблиця 42.7.** Приклади складених логічних виразів

$a = 2, b = 3$	
Логічний вираз	Значення
not $a == b$	True
$a + b == 5$ or $a > 5$	True
$a <= b$ and $b == 3$	True
$a >= b$ and $b == 3$	False
$a > b$ or $a == 7$	False

У математиці досить часто використовують подвійні нерівності, наприклад,  $1 < x \leq 5$ . На відміну від більшості мов програмування Python підтримує використання подвійних умов. Отже, наведений приклад мовою Python буде записано у вигляді наступного коду:  $1 < x \leq 5$ .

### 42.3.

## ВЕЛИЧИНИ ЛОГІЧНОГО ТИПУ

Значення логічних виразів можна присвоювати логічним величинам. Розглянемо приклади, представлені у таблиці 42.8.

**Таблиця 42.8.** Приклади використання величин логічного типу

№	Присвоєння значень логічних величин	Значення логічної величини
1	$x=y<5$	true або false
2	$a=7!=8$	true
3	$z=2+3==6$	false

У першому прикладі змінній  $x$  буде присвоєно вираз  $y < 5$ , що має значення true або false, в залежності від значення змінної  $y$ .





У наступному прикладі змінній  $a$  присвоюється логічний вираз  $7 != 8$ , значення цієї змінної буде `true`.

У третьому прикладі  $z$  присвоюється логічний вираз  $2+3=6$ , значення цієї змінної буде хибним.

Змінні  $x$ ,  $a$  та  $z$  є логічними величинами. В Python вони мають тип `bool` і, як видно з прикладів, можуть приймати значення `true` або `false`.



### Контрольні запитання та завдання

1. Що таке алгоритм з розгалуженням?
2. Що таке простий логічний вираз? Наведіть приклади.
3. Що таке складений логічний вираз? Наведіть приклади.
4. Поясніть принципи виконання операцій логічного заперечення, множення, додавання.
5. Який пріоритет логічних операцій?
6. Який тип мають логічні величини?



### Питання для роздумів

- 1\*. Яке значення отримає логічний вираз  $a > 8$ , якщо  $a = 5$ ;  $a = 10$ ?
- 2\*. Яке значення отримає логічний вираз  $a > 4$  and  $a + b <= 4$ , якщо  $a = 5$ ,  $b = -2$ ;  $a = 10$ ,  $b = 2$ ?
- 3\*. Яке значення отримає логічний вираз  $\text{not}(\text{not}(a = 5))$ , якщо  $a = 5$ ;  $a = 0$ ?
- 4\*\*. Яке значення логічного виразу  $a$  or  $\text{not}(a)$ ?



### Завдання для досліджень

- 1\*. Як потрібно розставити дужки в логічному виразі  $a$  Or  $b$  And  $a$  Or  $c$  And  $b$ , щоб його значення було істинним, якщо  $a = \text{False}$ ,  $b = \text{True}$ ,  $c = \text{True}$ ?



Пройдіть  
тест



## §43.

## Умовні оператори

### 43.1.

### ПОВНА ФОРМА РОЗГАЛУЖЕННЯ

У попередньому уроці ми розглянули види логічних виразів. Зараз познайомимося з їхнім використанням в умовних операторах.

У мові програмування Python використовують повну та скорочену форми умовного оператора.

Так, **повна форма** умовного оператора у Python має вигляд, представлений на рис. 43.1. Також на рисунку представлено відповідну їй блок-схему.

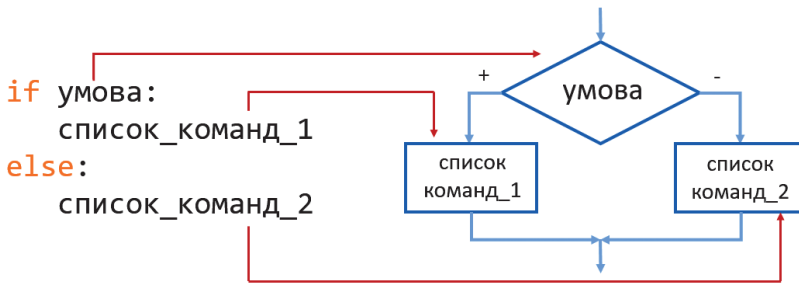


Рис. 43.1. Повна форма розгалуження

Отже, коли умова є істинною, то виконується **список\_команд\_1**, у протилежному випадку буде виконано **список\_команд\_2**.

Зверніть увагу на синтаксис та загальну конструкцію програмного коду. Перед умовою встановлено службове слово **if**, що перекладається як «якщо», після умови встановлюється знак двокрапки, нижче з відступом праворуч записуються команди, що мають бути виконані у разі виконання умови (кожна нова команда буде записана з нового рядка). Нижче, вже без відступу, встановлено службове слово **else** з двокрапкою (*else* у перекладі означає «інакше») і на наступному рядку знову з відступом праворуч — перелік команд у разі невиконання умови.

Розглянемо використання умовного оператора при розв’язанні наступної задачі.

#### ВПРАВА 43.1



**Завдання.** У Миколи є  $s$  гривень. Визначити, чи вистачить йому цих грошей для покупки трьох олівців, які коштують по  $n$  гривень. Вивести повідомлення «вистачить» або «не вистачить».

Блок-схема розв'язання даної задачі представлена на рис. 43.2.



Рис. 43.2

Відповідно до блок-схеми спочатку слід увести вхідні дані:  $s$  — суму грошей, яку має Микола, та  $n$  — вартість одного олівця. Далі слід перевірити, чи буде сума грошей, що є у Миколи, більшою або рівною вартості трьох олівців, а саме  $s \geq 3n$ . Якщо умова буде істинною, то буде виведено «вистачить», у протилежному випадку буде виведено «не вистачить».

Відповідно до даної блок-схеми створимо програмний код.

### І спосіб.

1. Спочатку введіть вхідні дані.

```
s=int(input("s="))
n=int(input("n="))
```

2. Далі додайте умовний оператор та логічний вираз.

```
if s>=3*n:
```

3. Для переходу на наступний рядок натисніть **Enter**. У результаті в середовищі автоматично має утворитися відступ.

4. Уведіть код для виведення напису «вистачить».

```
print("вистачить")
```

5. Перейдіть на наступний рядок, приберіть відступ та запишіть службове слово.

```
else:
```

6. Здійсніть перехід на наступний рядок та запишіть команду виведення напису «не вистачить».

```
print("не вистачить")
```



7. Запустіть програму на виконання.
8. Уведіть значення змінної  $s = 10$ ,  $n = 3$ . При цих значеннях Миколі грошей має вистачити.
9. Запустіть програму ще раз. Уведіть  $s = 10$ ,  $n = 5$ . Тепер у Миколи грошей повинно не вистачати.  
Як правило, одну й ту саму задачу можна розв'язати кількома способами. Ця задача не виняток, розглянемо й інші способи.

## II спосіб.

Так, у задачі ми можемо змінити логічний вираз в умові на протилежний, а саме,  $s < 3n$ . Звичайно, в цьому випадку слід поміняти місцями команди виведення повідомлень.

У коді задачі відповідно також буде змінено умову, а також повідомлення у разі виконання або невиконання умови. Отже, оператор розгалуження для цього способу буде мати вигляд:

```
if s < 3*n:
 print("не вистачить")
else:
 print("вистачить")
```

## III спосіб.

Ще один спосіб розглянемо з використанням логічних величин. Після команд введення даних код буде мати вигляд:

```
a = s >= 3*n
if a:
 print("вистачить")
else:
 print("не вистачить")
```

У даному коді перед застосуванням умовного оператора змінній  $a$  присвоюється значення логічного виразу. Після цього замість умови в умовному операторі достатньо записати лише змінну  $a$ . Якщо ця змінна приймає значення `true`, то виконується перша команда, в протилежному випадку — команда після службового слова `else`.

### 43.2.

## СКОРОЧЕНА ФОРМА РОЗГАЛУЖЕННЯ

Крім повної форми умовного оператора, існує скорочена форма (рис. 43.3). На блок-схемі видно, що команди будуть виконуватися лише у разі виконання умови, в протилежному випадку жодних дій виконано не буде.

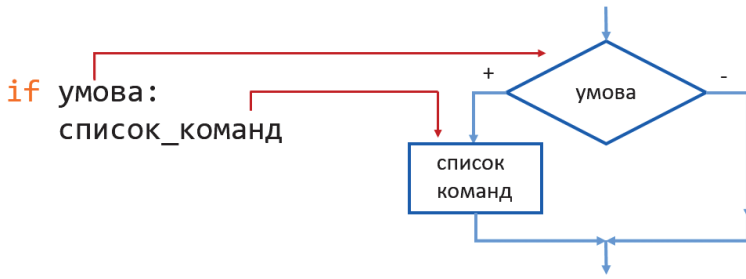


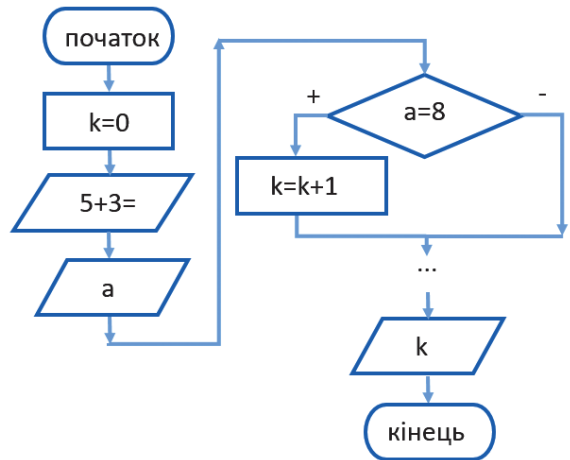
Рис. 43.3

Розглянемо приклад використання скороченої форми умовного оператора.

**ВПРАВА 43.2**



**Завдання.** Створити програму, що буде виводити послідовно три арифметичні вирази, а користувач має надавати на них відповідь. У результаті програма має підрахувати кількість правильних відповідей.



Алгоритм даної програми представлено у вигляді блок-схеми.

Кількість балів буде зберігатися у змінній  $k$ . Отже, на початку програми ця змінна має значення  $0$ . Далі на екран буде виведено перший вираз, і програма буде очікувати на відповідь. Значення, уведене користувачем, буде присвоєно змінній  $a$ . Після одержання відповіді здійснюється перевірка на правильність відповіді. Якщо відповідь є правильною, то значення змінної  $k$  збільшується на  $1$ . Так це має повторюватися три рази. Після чого буде виведено значення змінної  $k$ .

Відповідно до даної блок-схеми створимо програмний код.

1. Надайте змінній  $k$  значення  $0$ :  
 $k=0$



2. Уведіть код виведення завдання та очікування на відповідь:
 

```
a=int(input("5+3="))
```
3. Уведіть умовний оператор: якщо введена користувачем відповідь буде дорівнювати 8, то збільшити значення змінної  $k$  на 1:
 

```
if a==8:
 k=k+1
```
4. Запишіть код виведення значення змінної  $k$ :
 

```
print("Кількість правильних відповідей ", k).
```
5. Запустіть програму і введіть правильну відповідь. Результат має бути 1.
6. Ще раз запустіть програму та введіть неправильну відповідь. Результат має бути 0.  
Відповідно до умови задачі ми маємо перевірити правильність виконання трьох виразів.
7. Скопіюйте два рази коди виведення завдання та перевірки правильності відповіді.
8. Замініть запитання та відповіді (2 запитання:  $5*3=$ , 3 запитання:  $5-3=$ ).
9. Запустіть програму та перевірте, як вона буде працювати, якщо ввести усі відповіді правильно, дві відповіді правильно, одну або жодної правильної відповіді.



Виконайте  
вправу 43.3



### Контрольні запитання та завдання

1. Опишіть синтаксис повної форми розгалуження.
2. У чому відмінність повної форми розгалуження від скороченої?



### Питання для роздумів

- 1\*. В яких випадках доцільно використовувати логічні змінні в програмному коді?
- 2\*. Чим відрізняються програмні коди а) і б)? Який результат одержить змінна  $a$  в результаті виконання коду а); коду б)?
 

<p>а) <math>a=1</math></p> <pre>if a&gt;3:     a=a+2     a=a*2</pre>	<p>б) <math>a=1</math></p> <pre>if a&gt;3:     a=a+2     a=a*2</pre>
----------------------------------------------------------------------	----------------------------------------------------------------------



## Завдання для досліджень

1\*. Визначте значення змінної  $a$  в результаті виконання програмного коду:

```
a=5
if a!=0:
 a=a+3
else:
 a=a-3
```

2\*. Визначте значення змінної  $x$  після виконання програмного коду:

```
a = 5
b = 6
if a+b%10=1:
 x=b-a
else:
 x=b+a
```

## §44.

## Вкладені умови. Множинне розгалуження

## 44.1.

## ВКЛАДЕНІ УМОВНІ ОПЕРАТОРИ

Раніше ми познайомилися з різними формами умовного оператора. Розглянемо випадки, коли у програмах необхідно використати більше ніж одну умову. Це дозволяє здійснити використання вкладених конструкцій умовних операторів.

Отже, *вкладені умовні оператори* — це умовні оператори, що входять до складу інших умовних операторів.

Так, на рис. 44.1 наведено блок-схему одного із можливих випадків вкладеного умовного оператора.

У даному випадку вкладення використовується на гілці у разі невиконання умови. Також воно може бути на гілці після виконання умови або на обох гілках. До того ж у вкладений алгоритм можна також ще додавати умовні оператори. Такій конструкції алгоритму буде відповідати наступний код у Python.

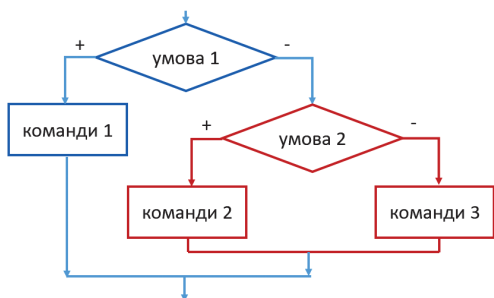


Рис. 44.1





```

if умова_1:
 команди_1
else:
 if умова_2:
 команди_2
 else:
 команди_3

```

Зверніть увагу, як у даному коді створено відступи. Вкладений умовний оператор після гілки **else** розташований після відступу праворуч, відповідно його команди розташовані з відступом від службових слів **if** та **else** даного оператора. Розглянемо використання вкладених розгалужень при виконанні вправ.

### ВПРАВА 44.1



**Завдання.** Дано значення кута. Визначити, яким він буде: прямим, тупим чи гострим.

Тобто, якщо кут дорівнює  $90^\circ$ , то він буде прямим, більше  $90^\circ$  — тупим, менше  $90^\circ$  — відповідно гострим. Уважаємо, що користувач має ввести значення від 0 до 180 градусів.

При створенні проекту будемо керуватися блок-схемою, представленою на рис. 44.2.

Відповідно до блок-схеми спочатку необхідно ввести значення змінної  $a$ , що відповідає значенню кута. Наступний крок алгоритму — це перевірка  $a = 90$ . Якщо ця умова виконується, то буде виведено повідомлення, що кут є «прямий», у протилежному випадку розглядається умова  $a < 90$ , якщо умова виконується, буде виведено повідомлення, що кут «гострий», у протилежному випадку — повідомлення, що кут є «тупий».

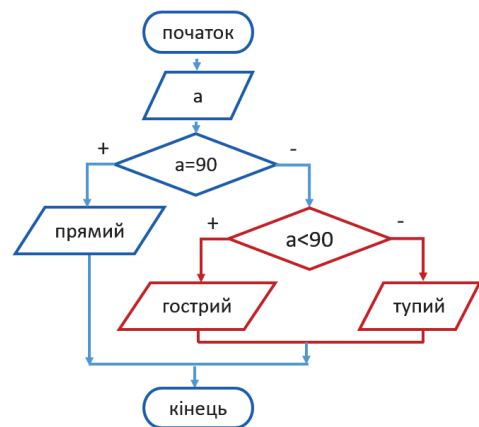


Рис. 44.2

За даною блок-схемою створимо програмний код.

1. Уведіть код введення значення змінної  $a$ .

```
a=int(input("a="))
```

2. Уведіть умовний оператор.

```

if a==90:
 print("прямий")

```



```
else:
 if a<90
 print("гострий")
 else:
 print("тупий")
```

3. Перевірте роботу програми за умови введення різних значень змінної  $a$ . Так, при  $a = 60$  ви маєте одержати повідомлення «гострий», при  $a = 90$  — «прямий», при  $a = 100$  — «тупий».

## 44.2.

### МНОЖИННЕ РОЗГАЛУЖЕННЯ

У математиці, як вам відомо, крім уже розглянутих, є й інші різновиди кутів. Так, кут, величина якого дорівнює 180 градусів, називається розгорнутим. Кут, більший за 180 градусів, але менший від 360 градусів, називають неопуклим, а кут, який дорівнює 360 градусів, називають повним.

Якщо у попередній задачі врахувати ще й ці умови, то ми одержимо досить складну конструкцію програмного коду, що буде містити принаймні ще три вкладення.

Для розв'язання подібних задач краще підійде *множинне розгалуження*, що у мові Python має таку загальну структуру:

```
if умова_1:
 команди_1
elif умова_2:
 команди_2
...
elif умова_n:
 команди_n
else:
 команди_n+1
```

Множинне розгалуження використовують при врахуванні більше ніж однієї умови.

#### ВПРАВА 44.2



**Завдання.** Вивести назву кута в залежності від уведених користувачем значень змінної  $a$ .

Види кутів від 0 та до 360 градусів:

- якщо  $0 < a < 90$  — кут гострий;
- якщо  $a = 90$  — кут прямий;
- якщо  $90 < a < 180$  — кут тупий;
- якщо  $a = 180$  — кут розгорнутий;



- якщо  $180 < a < 360$  — кут неопуклий;
- якщо  $a = 360$  — кут повний;
- інакше — кут не визначений.

1. Уведіть код введення значення змінної  $a$ :

```
a=int(input("a="))
```

2. Уведіть коди умовного оператора. Нижче наведено його початок та кінець. Частина, якої не вистачає, заповніть самостійно, враховуючи вказані у праві умови.

```
if 0<a<90:
 print("кут гострий")
elif a==90:
 print("кут прямий")
...
else:
 print("кут не визначений")
```

3. Перевірте роботу даної програми для різних значень змінної  $a$ . Так, при введенні значення 180, ви маєте одержати повідомлення, що кут є розгорнутим, для 270 — «неопуклий», для 400 — програма має повідомити, що кут не визначений.



Виконайте вправу 44.3



### Контрольні запитання та завдання

1. Для чого використовують вкладені розгалуження?
2. Для чого використовують множинне розгалуження?



### Питання для роздумів

- 1\*. Чим відрізняється вкладене розгалуження від множинного?
- 2\*. Чи можна було у вправі 44.1 при створенні програми скористатися множинним розгалуженням?



### Завдання для досліджень

- 1\*. Який вид розгалуження (вкладене чи множинне) буде більш раціональним при створенні програми для розв'язання задачі? Доведіть це, створивши відповідні програми.
  - а) Дано номер місяця, визначити, до якої пори року він належить.
  - б) Дано три додатні числа, визначити, чи можна побудувати трикутник з такими довжинами сторін.



Пройдіть  
тест



## §45.

## Налагодження та тестування програми

## 45.1.

## ВИДИ РЕДАКТОРІВ КОДУ

Ми вже навчилися з вами виконувати найпростіші програми у середовищі Python IDLE. Для невеликих програм його можливостей більш ніж достатньо, але з кожним уроком наші програми стають більш складними, в цьому випадку виникає необхідність перевірки працездатності програми на кожному її кроці. Далі розглянемо, як виконати налагодження та тестування комп'ютерних програм. Існують редактори коду, які є більш зручними для введення та редагування програм, серед яких Eclipse, PyCharm, Atom, Visual Studio Code тощо.

Основними їхніми перевагами над Python IDLE є автодоповнення команд програмного коду та можливість покрокового виконання програми. Професійні програмісти користуються переважно саме такими редакторами коду.

Одним із таких редакторів є також Visual Studio, який є абсолютно безкоштовним, зручним у користуванні та налагодженні. Безкоштовно його можна скачати на офіційному сайті Microsoft. У своїх поясненнях далі ми використовуватимемо російськомовну версію Visual Studio 2019.

Розглянемо послідовність кроків створення та збереження програми для розв'язання наступної задачі в середовищі Visual Studio, а вже потім на її прикладі виконаємо налагодження та тестування програм.

## ПРАКТИЧНЕ ЗАВДАННЯ



**Завдання.** Створити програму розв'язання задачі: «Дано чотирицифрове число. Визначити, чи є це число паліндромом». Результат вивести у вигляді повідомлення «паліндром» або «не паліндром».

Число називають *паліндромом*, якщо воно однаково записується як зліва направо, так і справа наліво. Наприклад, числа 5445, 7777 — паліндроми, а числа 1122 та 1231 — не паліндроми.

Алгоритм розв'язання цієї задачі може бути наступним.

1. Увести чотирицифрове число (наприклад, 2332).
2. Визначити цифри числа (для нашого прикладу це будуть цифри 2, 3, 3, 2).
3. Перевірити рівність першої та четвертої, а також другої та третьої цифр (2=2 та 3=3?).



4. Якщо логічні вирази є істинними, то вивести повідомлення «паліндром», якщо принаймні один із виразів є хибним — вивести повідомлення «не паліндром». (У нашому прикладі буде виведено повідомлення «паліндром».)

Переходимо до введення програмного коду в середовищі Visual Studio.

1. Запустіть програму середовища. Для цього увійдіть до меню **Пуск**, оберіть програму Visual Studio 2019.
2. У вікні, що з'явилось, оберіть **Создание проекта (Створення проєкту)**.
3. У наступному вікні оберіть **Приложение Python (Додаток Python)** та натисніть **Далее (Далі)**.
4. У наступному вікні введіть ім'я проєкту (в середовищі Visual Studio створені програми називаються проєктами, оскільки крім файлу з розширенням **.py** утворюються ще деякі додаткові файли та папки). Надайте проєкту назву **Паліндром**, також оберіть папку збереження проєкту. Натискаємо кнопку **Создать (Створити)**. В результаті ви потрапите у вікно середовища Visual Studio, в якому у верхньому лівому куті відображається вікно введення коду.

З іншими елементами середовища познайомимось згодом. Отже, введемо програмний код.

5. У першому рядку введіть код для введення чотирицифрового числа.

```
a=int(input("Уведіть чотирицифрове число "))
```

Під час уведення коду, щоб уникнути помилок, намагайтеся використовувати автозавершення. Так, після початку введення службового слова у списку, що з'являється, перемістіться на необхідне слово та натисніть **Enter**. У результаті обране слово має бути застосовано.

6. У наступних рядках уведіть команди знаходження цифр числа.

```
a1=a//1000
a2=a//100%100
a3=a//10%10
a4=a%10
```

Якщо ви уважно подивитесь на команди коду, то помітите, що в одній із формул була допущена помилка. Це було зроблено спеціально, аби згодом при перевірці програми ми її знайшли.

7. Далі застосуйте умовний оператор, в якому сформуєте складений логічний вираз, перевіряючи рівність першої та четвертої цифр, а також другої та третьої цифр.

```
if a1==a4 and a2==a3:
```



8. Допишіть команди умовного оператора:

```
print("Паліндром")
else:
 print("Не паліндром")
```

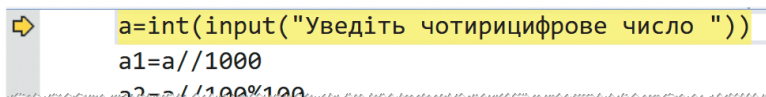
9. Збережіть програму. Для цього у меню **Файл** оберіть команду **Сохранить все (Зберегти все)**. Програму буде збережено в тій папці, яку було вказано при створенні проєкту.
10. Закрийте вікно середовища Visual Studio.
11. Для відкриття проєкту зайдіть до папки, в якій збережено проєкт, та запустіть файл **Паліндром.sln**. У результаті програму буде завантажено у середовищі Visual Studio.
12. Запустіть програму на виконання. Як і в середовищі IDLE, для запуску програми можна натиснути клавішу **F5** або кнопку **Пуск**. Виконання програми буде відбуватися у вікні консолі.
13. Уведіть число 2332, яке є паліндромом, і перевірте, чи правильно спрацює програма. Програма має вивести повідомлення, що число не є паліндромом. (Адже, як було зазначено, програма має помилки, отже, і результат має виводити неправильний.)
14. Для завершення роботи у вікні консолі натисніть довільну клавішу.

### 45.2.

### ПОКРОКОВЕ ВИКОНАННЯ ПРОГРАМИ

Ми бачимо, що синтаксичних помилок у програмі немає, напевно є помилка у логіці програми або в деякій формулі. Для налагодження коду програми у Visual Studio передбачено її покрокове виконання з перевіркою значень змінних на кожному кроці.

15. Запустіть покрокове виконання програми. Для цього натисніть клавішу **F11**. Після чого перший рядок програмного коду має підсвітитися жовтим кольором і навпроти нього повинна бути розташована жовта стрілка.



```
a=int(input("Уведіть чотирицифрове число "))
a1=a//1000
a2=a//100%100
```

16. Ще раз натисніть клавішу **F11**. У результаті буде здійснено виконання програмного коду підсвіченого рядка. В нашому випадку це команда введення змінної  $a$ , отже, відповідно має стати активним вікно консолі, в якому необхідно ввести чотирицифрове число.
17. Уведіть число 2332 та натисніть **Enter**. В результаті у програмі здійснюється перехід до наступного рядка, і він підсвічується також жовтим кольором.



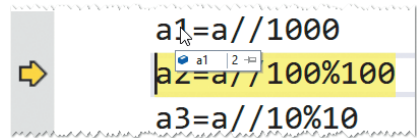
18. Натисніть ще раз **F11**, аби програма виконала знаходження першої цифри числа.

У будь-який момент покрокового виконання програми користувач може переглянути поточне значення змінної в програмному кодї, навівши на неї курсор миші.

19. Наведіть курсор на змінну  $a$ . В результаті ви побачите, що вона дорівнює введеному чотирицифровому числу 2332.

20. Наведіть на  $a1$  та переконайтеся, що значення першої цифри чотирицифрового числа було знайдено правильно і воно дорівнює 2.

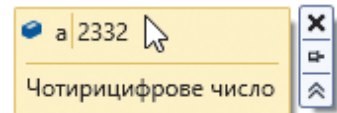
Проте кожний раз наводити курсор на змінну не дуже зручно. Щоб вирішити цю проблему, в середовищі передбачено закріплення бажаних змінних таким чином, аби їхнє значення було постійно відображено при покроковому виконанні програми.



21. Наведіть курсор на змінну  $a$  та натисніть значок у вигляді канцелярської кнопки. Щоб віконечко зі змінною не було таким великим, натисніть на значок канцелярської кнопки ще раз.

До того ж у цьому режимі можна ввести коментар до змінної. Це зручно, коли потрібно вказати її призначення.

22. Натисніть подвійну стрілочку та введіть коментар: «чотирицифрове число». Закріплену змінну можете перетягнути у бажане місце програмного коду.



23. Аналогічно закріпіть змінну  $a1$ .

24. Продовжіть покрокове виконання коду, натиснувши **F11**.

25. Перевірте значення змінної  $a2$ . Воно буде дорівнювати 23, а мало дорівнювати 3. Отже, напевно, помилка в обчисленні значення цієї змінної. Закріпіть її, як і попередні змінні.

26. Звичайно, цей код потрібно виправити, а для цього слід зупинити програму. В покроковому виконанні це можна зробити у будь-який момент. Для цього натисніть кнопку **Остановить отладку (Зупинити налагодження)** або **Shift+F5**.

27. Виправте програмний код наступним чином:  $a2=a//100%10$ .

28. Запустіть ще раз програмний код у покроковому режимі.

29. Натисніть два рази **F11**, уведіть у вікні консолі число 2332.



Тепер ви можете відслідковувати у створених вами віконечках значення змінних. Кожне нове поточне значення підсвічується червоним кольором, старі значення — блідо-сірим, нові — чорним.

30. Переконайтеся, що тепер значення другої цифри знайдено правильно, воно має дорівнювати 3.
31. Виконайте програму покроково далі, закріплюючи змінні. В результаті ви маєте переконатися, що інші цифри було знайдено правильно, і в результаті виконання програми одержати повідомлення, що число є паліндромом.

Якщо програма виконується покроково і на певному кроці необхідно продовжити її виконання без зупинки на кожному кроці, достатньо натиснути комбінацію клавіш **Shift+F11**.

Якщо код програми громіздкий, і покрокова перевірка всього коду недоцільна, можна встановити покажчик зупинки, починаючи з якого буде здійснено покрокове виконання коду.

```
a1=a//1000
a2=a//100%100
a3=a//10%10
```

Для цього потрібно клацнути навпроти бажаного коду, в результаті біля нього з'явиться червоний кружечок.

Запуск програми у цьому випадку здійснюється у звичайному режимі за допомогою кнопки **Пуск** або

- F5. Після чого код буде виконано до виділеного рядка і вже потім, натискаючи послідовно **F11**, ми зможемо покроково виконати бажаний фрагмент програми.

Щоб зняти покажчик зупинки, треба натиснути на червоний кружечок, який встановлено навпроти відповідного рядка коду.

### 45.3.

## ПОНЯТТЯ ТЕСТУВАННЯ ТА НАЛАГОДЖЕННЯ ПРОГРАМИ

Усі дії, що ми виконували до цього часу, є налагодженням та тестуванням програми.

**Тестування програми** — це перевірка відповідності між реальною і очікуваною поведінкою програми, що здійснюється на кінцевому наборі тестів, обраних певним чином.

Як правило, у тестах намагаються перевірити можливі варіанти виконання програми при тих чи інших значеннях. Тести в собі мають набір вхідних та вихідних даних. Якщо програма проходить перевірку на всіх тестах, то вважається, що вона створена правильно.

Так, наприклад, для нашої програми доцільно застосувати тести, наведені в таблиці 45.1.



Таблиця 45.1. Тести для перевірки програми

Вхідні дані	Результат	Пояснення варіанта вхідних даних
1111	паліндром	Усі однакові цифри
3443	паліндром	Однакові перша та остання, друга та третя цифри
1332	не паліндром	Різними є перша та четверта цифри
1321	не паліндром	Різними є друга та третя цифри
1234	не паліндром	Усі цифри різні

При виникненні помилки під час тестування програміст виконує налаштування програми, а саме, здійснює пошук програмного коду, в якому допущено помилку, та виправляє його. Як ми вже переконалися на практиці, знайти помилку в програмному коді зручніше всього під час покрокового виконання програми та перевірки значень змінних на кожному кроці програми.



### Контрольні запитання та завдання

1. Як запустити покрокове виконання програми?
2. Як закріпити змінну під час покрокового виконання програми?
3. Як виконати покрокове виконання частини програми?



### Питання для роздумів

- 1\*. Для чого використовують покрокове виконання програми?



### Завдання для досліджень

- 1\*. Як буде здійснюватися покрокове виконання програми, якщо встановити декілька покажчиків зупинки.
- 2\*\*. З'ясуйте призначення команд у меню **Отладка (Налаштування)**.

## §46.

## Задачі на знаходження найбільшого та найменшого серед кількох значень

## ЗАДАЧА 1

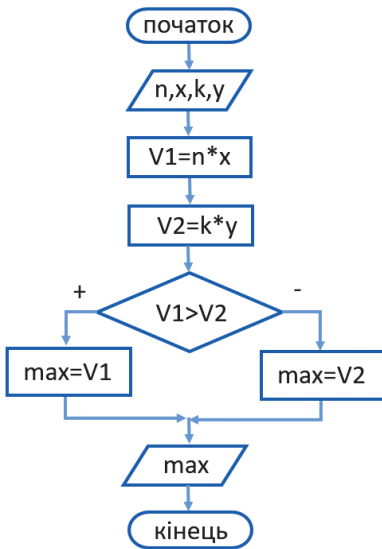


**Завдання.** Олена купила  $n$  зошитів по  $x$  гривень, а Ірина —  $k$  олівців по  $y$  гривень. Вивести вартість дорожчої покупки.

Для розв'язання даної задачі скористаємося наступною блок-схемою.

Тут  $V1$  та  $V2$  — вартості покупок Олени та Ірини відповідно. Після порівняння вартості покупок більше значення присвоюється змінній  $max$ .

На основі даної блок-схеми введемо програмний код.



1. Уведіть коди введення вхідних даних.

```
n=int(input("Кількість зошитів "))
x=int(input("Ціна одного зошита "))
k=int(input("Кількість олівців "))
y=int(input("Ціна одного олівця "))
```

2. Уведіть формули обчислення вартості.

```
V1=n*x #вартість зошитів
V2=k*y #вартість олівців
```

3. Уведіть умовний оператор, що визначає більше з двох значень.

```
if V1>V2:
 max=V1
else:
 max=V2
```

4. В останньому рядку запишіть команду виведення результату.

```
print("Вартість дорожчої покупки становить",max," грн")
```

5. Запустіть програму та перевірте правильність її виконання.

Вдосконалимо програму таким чином, щоб вона виводила повідомлення про те, вартість цієї покупки є більшою. Наприклад, «Більшою є вартість покупки, яку здійснила Олена. Вартість покупки 5 грн».

6. Після команди  $max=V1$  додайте команду, в якій змінній  $max\_name$  надайте значення "Олена":  $max\_name="Олена"$ .

7. Після команди  $max=V2$  додайте команду, в якій змінній  $max\_name$  надайте значення "Ірина":  $max\_name="Ірина"$ .



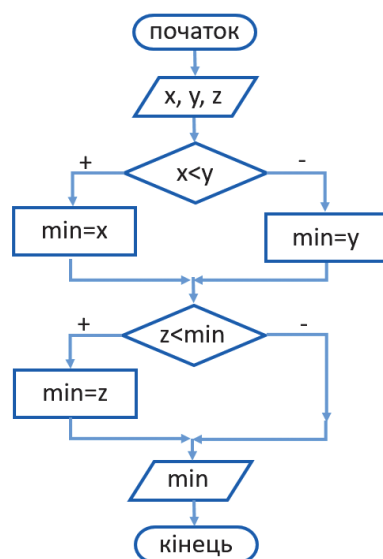
8. Змініть команду виведення результату:
- ```
print("Дорожчу покупку здійснила ",max_name,
      " Вартість покупки становить ",max," грн")
```
9. Запустіть програму та перевірте правильність її виконання.
10. Ще раз запустіть програму та введіть значення: $n=2$, $x=3$, $k=3$, $y=2$. В результаті ви маєте одержати повідомлення: «Дорожчу покупку здійснила Ірина. Вартість покупки становить 6 грн».
Вдосконалимо програму таким чином, аби при однакових значеннях було виведено повідомлення про це. Тут необхідно скористатися вкладеним розгалуженням.
11. Додайте умову, якщо $V1$ дорівнює $V2$, то вивести повідомлення, що вартості покупок є рівними і дорівнюють $V1$ грн. Весь умовний оператор, що був до цього, перемістіть до гілки **else**:
- ```
if V1==V2:
 print("Вартості покупок є однаковими і дорівнюють ",V1," грн")
else:
 if V1>V2:
 max=V1
 max_name="Олена"
 else:
 max=V2
 max_name="Ірина"
 print("Дорожчу покупку здійснила ",max_name,
 " Вартість покупки становить ",max," грн")
```
12. Запустіть програму та перевірте правильність її виконання.

## ЗАДАЧА 2



**Завдання.** Ігор, Андрій та Микола змагаються на велосипедах. Ігор подолав шлях за  $x$  секунд, Андрій — за  $y$  секунд, а Микола — на 5 секунд швидше ніж Андрій. Визначити, хто приїхав перший і за який час він подолав необхідну відстань.

Вхідними даними задачі є значення  $x$  — час Ігоря та  $y$  — час Андрія. За цими даними можна також визначити час руху Миколи. Він становить  $z = y - 5$ . Отже, ми маємо знайти найменше значення з трьох даних. Для знаходження мінімального значення із трьох даних скористаємося блок-схемою.





Спочатку вводимо вхідні дані  $x$ ,  $y$  та  $z$ . На наступному кроці порівнюємо спочатку значення двох величин  $x$  та  $y$ . Менше значення присвоюємо змінній  $min$ . На наступному кроці перевіряємо, чи буде значення змінної  $z$  менше від  $min$ . Якщо це так, то  $min$  одержить значення змінної  $z$ , в протилежному випадку нічого не виконуємо, адже  $min$  уже одержало менше із інших двох значень. На останньому кроці виводимо значення змінної  $min$  — це і буде мінімальне значення з трьох величин. Застосуємо ідею даного алгоритму до розв'язання задачі в середовищі програмування.

1. Уведіть коди для введення вхідних даних.

```
x=int(input("Час Ігора ="))
y=int(input("Час Андрія ="))
```

2. Уведіть код знаходження значення змінної  $z$  — часу, який витратив Микола.

```
z=y-5
```

3. Уведіть умовний оператор порівняння значення змінних  $x$  та  $y$ .

```
if x<y:
 min=x
 min_name="Ігор"
else:
 min=y
 min_name="Андрій"
```

4. Далі знову використайте умовний оператор, порівнюючи значення змінних  $min$  та  $z$ .

```
if z<min:
 min=z
 min_name="Микола"
```

5. Додайте команду виведення результату:

```
print("Швидше за всіх приїхав ", min_name,
 " він їхав ", min, " с.")
```

6. Запустіть програму та перевірте правильність її виконання. Введіть  $x = 10$ ,  $y = 17$ . В результаті маєте одержати, що першим подолав відстань Ігор. Спробуйте самостійно підібрати значення, при яких першим мав приїхати Микола; першим мав приїхати Андрій.

Удосконалимо задачу таким чином, щоб у ній також було виведено інформацію про того, хто подолав дистанцію останнім. Звичайно, для цього можна було б дописати у програмі нижче коди для знаходження більшого з трьох значень  $x$ ,  $y$  та  $z$ , але ми розглянемо більш раціональний спосіб.



7. Додайте до умовного оператора порівняння змінних  $x$  та  $y$  команди визначення максимального значення.

```
if x < y:
 min = x
 max = y
 min_name = "Ігор"
 max_name = "Андрій"
else:
 min = y
 max = x
 min_name = "Андрій"
 max_name = "Ігор"
```

8. Додайте команди порівняння максимального значення зі значенням часу Миколи в окремому умовному операторі:

```
if z > max:
 max = z
 max_name = "Микола"
```

9. Додайте команду виведення максимального значення.

```
print("Останнім приїхав ", max_name,
 " він їхав ", max, " с.")
```

10. Запустіть програму та перевірте правильність її виконання.

#### ВПРАВА 46.1



**Завдання.** Створіть програму розв'язання задачі: «Двоє туристів вирушили у подорож. Перший пройшов відстань  $a$  км за  $x$  годин, а другий — відстань  $b$  км за  $y$  годин. Визначте, який турист рухався швидше та вкажіть значення цієї швидкості».



Познайомтеся з розв'язанням ще однієї задачі

#### ВПРАВА 46.2



**Завдання.** Створіть програму розв'язання задачі: «Дано трицифрове число. Визначити найбільшу та найменшу цифри числа».

**§47.****Оператор циклу з лічильником****47.1.****ОСНОВНІ ПОНЯТТЯ**

Вивчення мови Python продовжимо, опановуючи оператори циклів. Розпочнемо ознайомлення з оператором циклу з лічильником. З сьомого класу ви пам'ятаєте, що багаторазове виконання тієї самої послідовності команд доти, доки виконується визначена наперед умова, називають **циклом** (повторенням).

Алгоритм, що містить цикли, називають **алгоритмом з повторенням**.

Майже всі мови програмування містять оператори циклів. Не є винятком і Python. Однією з властивостей комп'ютера є його здатність багато разів виконувати ті самі дії. З повторюваними обчисленнями програмісти стикаються повсякчас. Є цикли, в яких кількість повторювань відома заздалегідь, і є цикли, де кількість повторень залежить від виконання певної умови.

Цикли, в яких кількість виконуваних повторювань відома заздалегідь, називають **циклами з лічильником** або **циклами з параметром**, тому що для організації повторень використовують змінну, яку називають *лічильник* або *параметр*.

Програмний код такого циклу має загальний вигляд:

```
for лічильник in діапазон:
 тіло_циклу
```

Де: лічильник — це числова змінна; діапазон — інтервал значень лічильника циклу.

Зверніть увагу, що після оператора циклу слід поставити знак двокрапки, а тіло циклу необхідно записувати з відступом від оператора циклу.

**47.2.****ФУНКЦІЯ RANGE**

Для визначення діапазону параметрів циклу в Python використовується функція **range**.

Функція має такий загальний вигляд:

```
range([початок], кінець, [крок])
```

Де: початок — початкове значення лічильника циклу; кінець — кінцеве значення лічильника циклу, але воно є не включеним до діапазону значень, отже, реально це буде число на 1 менше, ніж указано (зауважимо, що аргументи функції **range** мають бути цілими числами);





крок — значення, на яке змінюється лічильник циклу після виконання тіла циклу.

Параметри, розташовані у квадратних дужках, є необов'язковими. Якщо значення початок пропущено, то за умовчанням воно дорівнює 0. Якщо пропущено значення крок, то крок дорівнює +1.

На рис. 47.1 наведено блок-схему виконання циклу з параметром, а нижче наведено відповідний програмний код з використанням функції `range`.

```
for i in range(n1, n2, k):
 тіло циклу
```

Розглянемо загальний алгоритм виконання програмного коду.

1. Лічильнику присвоюють початкове значення  $n1$ .
2. Якщо воно менше від кінцевого значення  $n2$ , то здійснюється перехід до виконання операторів тіла циклу.
3. Після виконання операторів тіла циклу змінюється значення лічильника на значення кроку  $k$ , отже, лічильник одержує значення  $n1 + k$ .
4. Якщо одержане значення лічильника  $n1 + k$  менше від кінцевого значення  $n2$ , програма знову переходить до виконання операторів тіла циклу. Такі порівняння виконуються доти, доки значення лічильника менше від кінцевого значення.
5. Коли значення лічильника стає більшим за кінцеве значення, здійснюється перехід до команд, що знаходяться після тіла циклу.

Якщо початкове значення лічильника циклу є більшим за кінцеве, то тіло циклу жодного разу не виконається. Винятком є ситуація, коли значення кроку від'ємне. Проте в цьому випадку і алгоритм виконання циклу буде здійснюватися з урахуванням такого значення кроку.



Ознайомтеся з прикладами використання функції `range`

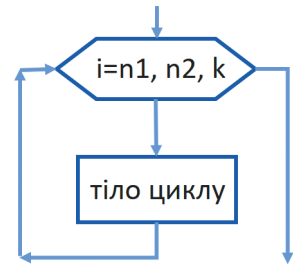


Рис. 47.1

### ВПРАВА 47.1



**Завдання.** Вивести  $n$  перших натуральних чисел у прямому та зворотному порядку.

**Дано:**  $n$  — кількість чисел.

Якщо  $n = 5$ , то буде виведено у стовпчик числа 1, 2, 3, 4, 5 та 5, 4, 3, 2, 1 (рис. 47.2).

1	5
2	4
3	3
4	2
5	1

Рис. 47.2

Уведемо програмний код.

1. Додайте команду введення значення змінної  $n$ , що визначає кількість чисел.

```
n=int(input("n="))
```

2. Уведіть оператор циклу з лічильником для виведення чисел від 1 до  $n$ .

```
for i in range(n):
 print(i+1)
```

Зверніть увагу на параметр команди **print**. Він приймає значення  $i + 1$ , оскільки лічильник змінюється від 0 до  $n - 1$ , отже, при додаванні до лічильника одиниці одержимо бажані значення від 1 до  $n$ .

3. Уведіть оператор циклу з лічильником для виведення чисел від  $n$  до 1.

```
for i in range(n,0,-1):
 print(i)
```

Тут для функції **range** встановлено початкове значення  $n$ , кінцеве — 0, а крок  $-1$ . Отже, значення лічильника буде змінюватися від  $n$  до 1. Відповідно для виведення чисел від  $n$  до 1 використовується команда `print(i)`.

4. Запустіть програму та перевірте правильність її виконання.

#### ВПРАВА 47.2



**Завдання.** Змінити попередню програму таким чином, аби вона виводила послідовність з перших  $n$  непарних чисел.

Так, якщо  $n = 6$ , то це будуть числа 1, 3, 5, 7, 9, 11, а також ці ж самі числа тільки у зворотному порядку.

1. Створіть копію файлу попередньої вправи.
2. Виконайте зміни у команді виведення значень від 1 до  $n$ .

```
print((i+1)*2-1)
```

3. Аналогічні зміни виконайте у команді виведення непарних чисел у зворотному порядку.

```
print(i*2-1)
```

4. Перевірте правильність програмного коду.



### Контрольні запитання та завдання

1. Що таке цикл у програмуванні?
2. Що таке алгоритм з повторенням?
3. Які цикли називають циклом з лічильником (параметром)?
4. Для чого використовується функція `range`?
- 5\*. Опишіть послідовність дій виконання циклу з лічильником з використанням функції `range`.



### Питання для роздумів

1. Наведіть приклади використання циклів у повсякденному житті.
- 2\*. Скільки разів буде виконуватися тіло циклу після заданих команд:
  - a) `for i in range(1,10,2):`
  - б) `for i in range(1,10,-2):`
  - в) `for i in range(15,1,-3):`
  - г) `for i in range(15,20):`
  - г) `for i in range(20,15)?`



### Завдання для досліджень

- 1\*. Які значення змінної `i` буде виведено в результаті виконання програми:

```
for i in range(2,10,2):
 print(i)?
```
- 2\*. Якого значення набуде змінна `x` у результаті виконання послідовності операторів:

```
x=0.5
for i in range(3):
 x=x*2
 x=x+1?
```



## §48.

## Використання циклу з параметром

## 48.1.

ЗНАХОДЖЕННЯ СУМИ, ДОБУТКУ, КІЛЬКОСТІ ТА  
СЕРЕДНЬОГО АРИФМЕТИЧНОГО ЧИСЛА

Досить часто, розв'язуючи різноманітні задачі, виникає потреба визначити суму, добуток, кількість або середнє арифметичне послідовності деяких чисел. Загальний вигляд програмних кодів для виконання цих дій представлено у таблиці 48.1.

Таблиця 48.1. Знаходження суми, добутку та кількості.

Знаходження суми	Знаходження добутку	Знаходження кількості
#Початкове значення $s=0$ for i in range(): #Обчислення суми $s=s+a$	#Початкове значення $p=1$ for i in range(): #Обчислення добутку $p=p*a$	#Початкове значення $k=0$ for i in range(): #Обчислення кількості $k=k+1$

У розглянутих кодах перший оператор присвоєння — це надання змінній початкового значення. Він виконується один раз перед циклом. Другий оператор виконується декілька разів у циклі. Тут замість  $a$  може бути будь-яке число, змінна або вираз.

Таблиця 48.2

крок	$i$	$s$
1	2	$0+2=2$
2	4	$2+4=6$
3	6	$6+6=12$
4	8	$12+8=20$
5	10	$20+10=30$

Наприклад, знайдемо суму парних чисел на проміжку від 2 до 10.

Для цього скористаємося таким програмним кодом:

```
s=0
for i in range(2,11,2):
 s=s+i
```

Отже, спочатку  $s=0$ . Далі у таблиці 48.2 представлено значення змінних  $i$  та  $s$  на кожному кроці виконання оператора циклу.

Отже, в результаті виконання даного коду  $s = 30$ .

## Знаходження середнього арифметичного

Для знаходження середнього арифметичного елементів послідовності використовують конструкцію:

```
if k!=0:
 sa=s/k,
```

де  $s$  — сума значень, а  $k$  — їхня кількість. Оскільки ділення на нуль неможливе, перед застосуванням формули здійснюється перевірка  $k \neq 0$ .



## ВПРАВА 48.1



**Завдання.** Знайти суму, кількість та середнє арифметичне всіх натуральних двоцифрових чисел, у яких перша цифра менша від другої. Вивести усі числа, що задовольняють умову.

### I. Визначення вхідних та вихідних даних.

*Вхідних даних* у задачі немає, оскільки це діапазон усіх двоцифрових чисел, а саме, чисел від 10 до 99. *Вихідними даними* будуть:  $s$ ,  $k$  та  $Sa$  — сума, кількість та середнє арифметичне, а також послідовність чисел, що задовольняє умову. *Проміжними змінними є*:  $i$  — лічильник,  $x_1$ ,  $x_2$  — перша та друга цифри числа.

### II. Уведення програмного коду

1. Уведіть команди встановлення початкових значень суми та кількості.

```
s=0
k=0
```

2. Уведіть оператор циклу з лічильником.

```
for i in range(10,100):
```

Для функції **range** початкове значення встановлюємо 10 (перше двоцифрове число), кінцеве 100 (на 1 більше ніж найбільше двоцифрове число).

3. У тілі циклу введіть команди для знаходження першої та другої цифр числа.

```
#знаходження першої цифри числа
x1=i//10
#знаходження другої цифри числа
x2=i%10
```

4. Також у тілі циклу запишіть умовний оператор для перевірки, чи є перша цифра менша від другої. У разі виконання умови виведіть це число, знайдіть суму чисел та кількість.

```
#Перевірка умови, чи є перша цифра числа менша від другої
if x1<x2:
 #Виведення числа, що задовольняє умову
 print(i)
 #Знаходження суми
 S=S+i
 #Знаходження кількості
 k=k+1
```

5. За межами тіла циклу введіть код знаходження середнього арифметичного.

```
if k!=0:
 Sa=S/k
```



6. Уведіть команди виведення значень суми, кількості та середнього арифметичного.

```
print("S=", S)
print("k=", k)
print("Sa=", Sa)
```

7. Запустіть програму та перевірте правильність її виконання.

## 48.2.

## СТВОРЕННЯ ТАБЛИЦЬ ЗНАЧЕНЬ ФУНКЦІЙ

Розглянемо ще один вид задач, для розв'язання яких використовують цикли з лічильником. Це задачі на створення таблиць значень функції. У задачах на створення таблиць значень функції потрібно обчислити значення функції на якомусь інтервалі із заданим постійним кроком.

*Крок* — це різниця між значеннями аргументу, для яких обчислюється значення функції.

## ВПРАВА 48.2



**Завдання.** Знайти значення функції  $y = x^2$  на інтервалі  $[-20; 20]$  з кроком 2.

Для даного прикладу ми одержимо таблицю 48.3, у якій кожному значенню  $x$  із заданого проміжку відповідає значення функції  $y$ .

Таблиця 48.3

x	y
-20	$(-20)^2=400$
-18	$(-18)^2=324$
-16	$(-16)^2=256$
...	...
16	$16^2=256$
18	$18^2=324$
20	$20^2=400$

## I. Визначення вхідних та вихідних даних.

*Вхідних даних* тут не буде — вони визначатимуться проміжком інтервалу змінної  $x$ . *Вихідні дані:*  $y$  — значення функції при різних значеннях  $x$ . *Проміжна змінна:*  $x$  — аргумент функції.

## II. Уведення програмного коду

1. Уведіть перший рядок програми. Це оператор циклу з лічильником, змінною тут є  $x$ , діапазон задайте з допомогою функції **range**, де мінімальне значення є  $-20$ , максимальне  $21$ , крок —  $2$ .

```
for x in range(-20, 21, 2):
```

2. У тілі циклу запишіть команду, що відповідає формулі функції:

```
y=x**2
```

3. Також у тілі циклу запишіть команду виведення значення змінних  $x$  та  $y$ .

```
print("x=", x, "y=", y)
```



Подумайте, чи можна було б в якості максимального значення встановити якесь інше число, при якому будуть виконуватися задані в умові задачі дії.

4. Запустіть програму та перевірте правильність її виконання. В результаті ви маєте одержати таблицю значень змінних  $x$  у діапазоні від  $-20$  до  $20$  та відповідні їм значення змінної  $y$ , що дорівнюють квадратам змінної  $x$ .

### ВПРАВА 48.3



**Завдання.** Створіть програми розв'язання задач:

- Знайти середнє арифметичне всіх натуральних двоцифрових чисел, кратних 13.
- Знайти суму всіх натуральних двоцифрових чисел, що закінчуються на 3.
- Знайти кількість усіх натуральних двоцифрових чисел, які складаються з непарних цифр.
- Знайти значення функції  $y = \sqrt{2x^2 + 1}$  на інтервалі  $[-15; 15]$  з кроком 3.



### Контрольні запитання та завдання

- Як виконати додавання (множення) послідовності чисел?
- Як знайти кількість чисел послідовності?
- Як знайти середнє арифметичне послідовності?



### Питання для роздумів

- Чому для знаходження добутку послідовностей чисел початкове значення встановлюють 1, а для знаходження суми — значення 0?



### Завдання для досліджень

- 1\*. Для цілих значень змінних  $a$ ,  $n$  та  $x$  створіть програмний код знаходження значення виразів, використовуючи цикл з лічильником:

- $s = 2 + 2^2 + \dots + 2^n$ ;
- $p = (a + x)(a^2 + 2x) \dots (a^n + nx)$ .



Пройдіть  
тест



## §49.

## Цикли з умовою

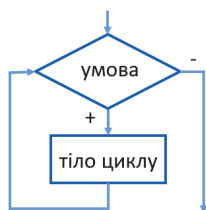


Рис. 49.1

При виконанні циклів кількість кроків у них не завжди є відомою, а залежить від певної умови. Саме для таких випадків використовують цикли з умовами про які піде мова далі.

Цикл, тіло якого виконується доти, доки виконується умова, називають **циклом з умовою**.

Так, на рис. 49.1 представлена блок-схема такого виду циклу, а нижче наведено відповідний код мовою Python:

```
while умова:
 тіло циклу
```

Відповідно до блок-схеми спочатку перевіряється умова. Якщо вона є істинною, виконуються оператори тіла циклу, після чого весь процес повторюється, доки виконується умова. Якщо спочатку умова не виконується, то тіло циклу жодного разу виконано не буде.

## ПРАКТИЧНЕ ЗАВДАННЯ 1



**Завдання.** Розглянемо використання циклу з умовою на прикладі обчислення значення виразу:  $s = 2 + 4 + 6 + \dots$  ( $n$  разів).

Визначимо вхідні та вихідні дані, а також проміжні змінні.

*Вхідні дані:*  $n$  — кількість доданків.

*Вихідні дані:*  $s$  — значення виразу.

*Проміжні змінні:*  $i$  — лічильник.

Дане завдання можна виконати з використанням як циклу з лічильником, так і циклу з умовою. Для порівняння наведемо приклади обох способів розв'язання задачі з використанням різних видів циклів.

Використання циклу з лічильником	Використання циклу з умовою
<pre>n=int(input("n=")) s=0 for i in range(1,n+1):     s=s+2*i print("s=",s)</pre>	<pre>n=int(input("n=")) s=0 i=1 while i&lt;=n:     s=s+2*i     i=i+1 print("s=",s)</pre>

Уведіть коди обох програм. Запустіть послідовно кожен на виконання. Спробуйте ввести однакові значення  $n$  та переконатися, що програмні коди будуть виводити однакові результати.



## ВПРАВА 49.1



**Завдання.** Створити програму визначення кількості та суми цифр даного числа.

Так, наприклад, якщо дано число 12456, то воно має 5 цифр, а сума його цифр дорівнює  $1 + 2 + 4 + 5 + 6 = 18$ .

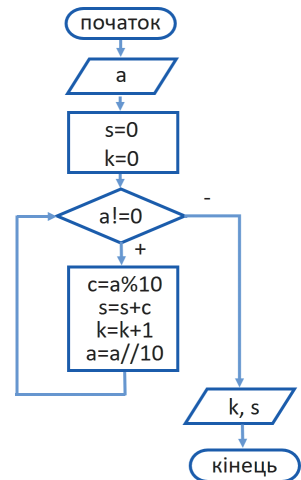
## I. Визначення вхідних та вихідних даних.

*Вхідні дані:*  $a$  — дане число; *вихідні дані:*  $k$  — кількість цифр числа,  $s$  — сума цифр числа.

## II. Алгоритм розв'язання задачі.

Алгоритм розв'язання задачі представлено у вигляді блок-схеми.

Розглянемо його покрокове виконання на прикладі числа 324 у таблиці 49.1.



Таблиця 49.1

Кроки програми	$a \neq 0$	$c$	$s$	$k$	$a$
Уведення числа та при- своєння початкових даних			0	0	324
1-й крок циклу	true	$324\%10=4$	$0+4=4$	$0+1=1$	$324//10=32$
2-й крок циклу	true	$32\%10=2$	$4+2=6$	$1+1=2$	$32//10=2$
3-й крок циклу	true	$3\%10=3$	$6+3=9$	$2+1=3$	$3//10=0$
4-й крок циклу	false				
Виведення результату			9	3	

## III. Відповідно до розглянутої блок-схеми введемо програмний код.

1. Уведіть код уведення числа та встановіть початкові значення змінних  $k$  та  $s$ .

```

a=int(input("a="))
k=0
s=0

```

2. Додайте оператор циклу з умовою:

```
while a!=0:
```

3. Уведіть тіло циклу:

```

c=a%10 #визначення останньої цифри числа
s=s+c #знаходження суми цифр числа
k=k+1 #знаходження кількості цифр числа
a=a//10 #відокремлення останньої цифри числа

```

4. Виведіть результат після виконання циклу:

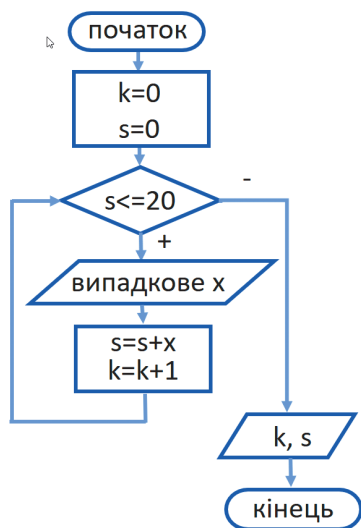
```
print("k=",k,"s=",s)
```

5. Запустіть програму та перевірте правильність її виконання.

**ВПРАВА 49.2**



**Завдання.** Сергій підкидає гральний кубик, на якому може випасти від 1 до 6 очок. Одержані значення він додає. Коли сума стане більшою, ніж 20, він закінчить підкидати кубик. Вивести, скільки йому знадобиться підкидань кубика та яку суму він одержить.



**I. Визначення вхідних та вихідних даних.**

*Вхідні дані:*  $x$  — випадкові числа від 1 до 6.

*Вихідні дані:*  $s$  — сума,  $k$  — кількість підкидань.

**II. Алгоритм розв’язання задачі.**

Розв’язання задачі виконаємо відповідно до наведеної блок-схеми.

Для лічильника  $k$  та суми  $s$  укажемо початкові значення, рівні нулю. Далі вказуємо умову, що дії будуть виконуватися до тих пір, поки  $s \leq 20$ . У тілі циклу вводимо випадкове значення змінної  $x$ . Збільшуємо суму  $s$  на одержане значення  $x$ , а значення лічильника  $k$  збільшуємо на 1. Ці дії будуть виконуватися до тих пір, поки буде виконуватися умова циклу. Як тільки цикл буде завершено, виводимо значення  $k$  та  $s$ .

**Визначення випадкових чисел**

Оскільки в нашій програмі використовується введення випадкового числа, то перед тим, як увести програмний код розв’язання задачі, розглянемо, як уводити випадкові числа у Python. Уведення випадкових чисел здійснюється з використанням однієї із функцій модуля **random**. У таблиці 49.2 наведено найчастіше уживані з них.

*Таблиця 49.2. Функції модуля random*

Функція	Призначення	Приклад та можливі значення
random()	Дійсне випадкове число від 0 до 1	random() — 0.1, 0.55
uniform(a, b)	Дійсне випадкове число від a до b	uniform(1, 5) — 1.5, 4.99
randint(a, b)	Ціле випадкове число від a до b	randint(1, 5) — 1, 2, 4, 5

Неважко здогадатися, що для одержання випадкового числа у задачі, де потрібно отримати цілу кількість очок в діапазоні від 1 до 6, доцільно скористатися саме функцією **randint()**.



III. Відповідно до розглянутої блок-схеми введемо програмний код.

1. Як уже було зазначено, функції введення випадкових чисел використовуються в модулі `random`, отже, ми його маємо імпортувати. Для цього введіть код:

```
from random import*
```

2. Установіть початкові значення  $k$  і  $s$  рівні нулю:

```
k=0
```

```
s=0
```

3. Уведіть команду оператора циклу з умовою:

```
while s<=20:
```

4. У тілі циклу надайте змінній  $x$  ціле випадкове число у проміжку від 1 до 6:

```
x=randint(1,6)
```

5. Виведіть це значення, використовуючи команду `print(x)`.

6. Уведіть команди збільшення значення змінної  $s$  на  $x$  та значення змінної  $k$  на 1.

7. За межами тіла циклу виведіть значення кількості та суми.

```
print("k=",k,"s=",s)
```

8. Запустіть програму та перевірте правильність її виконання. В результаті ви побачите виведені значення випадкових чисел від 1 до 6, а також їхні кількість та суму.



Виконайте  
вправу 49.3



### Контрольні запитання та завдання

1. Що таке цикл з умовою?
2. Опишіть послідовність дій виконання циклу з умовою.



### Питання для роздумів

- 1\*. Чи завжди замість циклу з умовою можна скористатися циклом з лічильником?
- 2\*. Опишіть та порівняйте принцип виконання циклів з лічильником та циклів з умовою.



### Завдання для досліджень

- 1\*. Для цілих значень змінних  $a$ ,  $n$  та  $x$  створіть програмний код знаходження значення виразів, використовуючи цикл з умовою:
  - а)  $s = 2 + 2^2 + \dots + 2^n$ ;
  - б)  $p = (a + x)(a^2 + 2x) \dots (a^n + nx)$ .



Пройдіть  
тест

## §50.

## Вкладені цикли

Цикли, так само, як і умовні оператори, можуть бути вкладеними. У цьому випадку розрізняють зовнішній і внутрішній цикли.

Так, на рис. 50.1 наведено приклад програмного коду із вкладеними циклами з умовою. Звичайно, використовувати вкладення можна як для циклів з умовою, так і для циклів з лічильником.

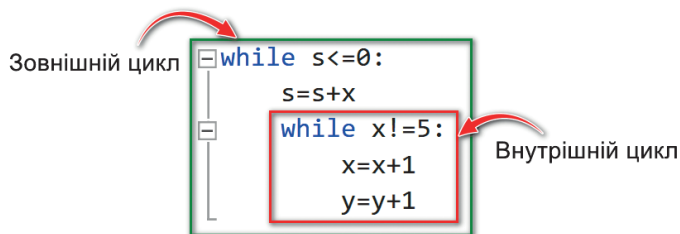


Рис. 50.1

Розглянемо вкладені цикли, виконуючи практичні вправи.

## ВПРАВА 50.1



**Завдання.** Створити програму виведення таблиці множення від 1 до 9. На рис. 50.2 наведено фрагмент виразів, які необхідно вивести.



Рис. 50.2

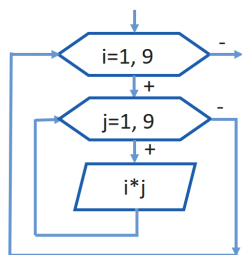


Рис. 50.3

## I. Визначення вхідних та вихідних даних.

У програмі скористаємося лише *проміжними змінними*, де змінна  $i$  буде відповідати першому множнику, а змінна  $j$  — другому множнику в дії множення.

## II. Алгоритм розв'язання задачі.

Блок-схема розв'язання даної задачі представлена на рис. 50.3.

Змінна  $i$  буде змінюватися від 1 до 9 у зовнішньому циклі, відповідно змінна  $j$  від 1 до 9 — у внутрішньому. Після одержання значень змінних здійснюється їхнє множення та виведення результату.

## III. Створення програмного коду.

## 1. Запишіть оператор зовнішнього циклу:

```
for i in range(1,10):
```

## 2. У тілі зовнішнього циклу запишіть оператор внутрішнього циклу:

```
for j in range(1,10):
```



3. У тілі внутрішнього циклу запишіть команду виведення виразу:
- ```
print(i, "", j, "=", i*j)
```
4. Виконайте роботу програми в покроковому режимі.
- Натисніть **F11**. В результаті має здійснитися перехід до першої команди.
 - Відкрийте вікно відображення значень змінних:
Отладка (Налагодження) ▶ Окна (Вікна) ▶ Локальные (Локальні).
 - Ще раз натисніть **F11**. В результаті після виконання першої команди змінна i має одержати значення 1.
 - Натисніть **F11**. В результаті змінна j має одержати значення 1. У вікні виведення має з'явитися вираз $1*1=1$.
 - Натисніть **F11**. В результаті знову має здійснитися перехід до оператора внутрішнього циклу, після його виконання змінна i має залишитися незмінною, а значення змінної j має збільшитися на 1, відповідно має бути виведено вираз $1*2=2$.
 - Продовжуйте натискати **F11**. В результаті ви побачите, що значення змінної j збільшується на 1. Після досягнення змінною j значення 9, виводиться вираз $1*9=9$ і здійснюється перехід до оператора внутрішнього циклу.
 - Оскільки j досягло максимально можливого значення, то після наступного натискання **F11** здійснюється перехід до оператора зовнішнього циклу. Після його виконання змінна i одержує значення 2, відповідно після виконання оператора внутрішнього циклу змінна j отримує знову значення 1. Наступні кроки виконання програми будуть призводити до виконання усіх ітерацій внутрішнього циклу, після чого буде здійснено перехід до зовнішнього циклу.
 - Натисніть **Shift+F11**, в результаті буде виконано останні кроки програми. І у вікні виведення ви отримаєте всю таблицю множення від 1 до 9.
5. Самостійно спробуйте створити програму для цього ж завдання, скориставшись:
- зовнішнім та внутрішнім циклами з умовою;
 - зовнішнім циклом з умовою, внутрішнім — з лічильником;
 - зовнішнім циклом з лічильником, внутрішнім — з умовою.

ВПРАВА 50.2



Завдання. Створити програму, що виводить трикутник із чисел від 1 до n .

На рис. 50.4 представлено зовнішній вигляд числового трикутника для $n = 7$.

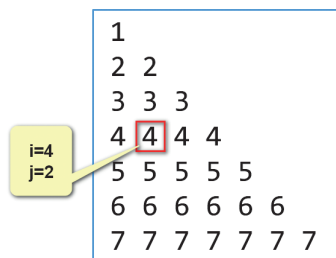


Рис. 50.4

I. Визначення вхідних та вихідних даних.

Вхідні дані: n — кількість рядків числового трикутника.

Проміжні змінні: i — номер поточного рядка, j — порядковий номер числа у рядку. Так, для позначеного у трикутнику числа номер i має значення 4, номер j — 2.

II. Створення програмного коду.

1. У першому рядку запишіть команду введення значення змінної n .

```
n=int(input("n="))
```

2. Запишіть команду зовнішнього циклу. Оскільки усього має бути n рядків, установіть для i мінімальне значення 1, а максимальне — $n + 1$.

```
for i in range(1,n+1):
```

3. У внутрішньому циклі буде здійснюватися виведення числа у рядок стільки разів, яке значення має змінна i . Отже, лічильник j буде змінюватися від 1 до $i + 1$. Уведіть команду внутрішнього циклу:

```
for j in range(1,i+1):
```

4. У тілі внутрішнього циклу запишіть команду виведення числа з допомогою команди `print`.

```
print(i,end=" ")
```

Зверніть увагу на запис цієї команди. Тут i — це число, що необхідно вивести. Щоб після команди `print` виведення чисел продовжувалося у цьому ж рядку, тут застосовано параметр `end=" "`. Це означає, що після виведення значення змінної i буде встановлено пробіл і всі записи будуть відбуватися в одному рядку.

5. Але після виведення ряду чисел необхідно здійснити перехід на новий рядок. Для цього, уже в тілі зовнішнього циклу, додайте команду:

```
print()
```



Виконайте праву 50.3

Отже, після виконання внутрішнього циклу, що виводить текст у рядок, буде здійснюватися перехід на новий рядок. Ці дії будуть відбуватися доти, доки не буде виведено n рядків чисел.

6. Запустіть програму та перевірте правильність її виконання.

**Контрольні запитання та завдання**

1. Що таке вкладені цикли? Наведіть приклади.
2. Які можуть існувати види вкладених циклів?
3. Що таке внутрішній цикл; зовнішній цикл?



Питання для роздумів

- 1*. Скільки вкладень можуть містити цикли?
- 2*. Яке значення одержить змінна a в результаті виконання фрагмента програми:

```
a=1
while a<10:
    for i in range(3):
        a=a+i*2?
```

§51.

Основні поняття об'єктно-орієнтованого програмування

51.1.

ОБ'ЄКТИ. ВЛАСТИВОСТІ Й МЕТОДИ ОБ'ЄКТІВ

Для створення різних прикладних програм, що мають графічний інтерфейс, використовують засоби **об'єктно-орієнтованого програмування**. Основною одиницею в об'єктно-орієнтованому програмуванні є програмний об'єкт, який поєднує властивості та дії, які з ним можна виконувати.

Одним із яскравих прикладів програм, створених засобами об'єктно-орієнтованого програмування, є MS Paint. У цьому додатку об'єктами є вікно, кнопки, вкладки, палітра, робоче поле тощо (рис. 51.1).

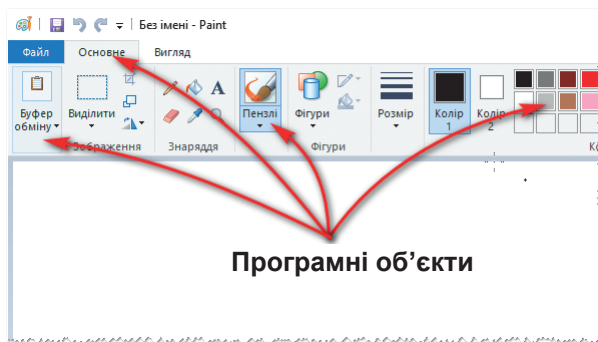


Рис. 51.1. Приклади програмних об'єктів у вікні Paint

Програмні об'єкти мають певні властивості. Так, кнопки мають такі властивості: розмір, назва, шрифт, колір, місце розташування на панелі інструментів, активність тощо.

Крім властивостей, об'єкт також може виконувати певні дії, що називаються *методами*. Скажімо, одна кнопка дозволяє намалювати

лінію, інша — виконати виділення певного фрагмента. Отже, *методи* визначають дії, які можна виконати з тим чи іншим об'єктом.

51.2.

ГРАФІЧНИЙ ІНТЕРФЕЙС КОРИСТУВАЧА

Працюючи за комп'ютером, ми використовуємо екранні вікна, текстові поля, кнопки, меню, вкладки тощо. Ці елементи називають елементами керування. Сукупність елементів керування, з якими взаємодіє користувач, називають *графічним інтерфейсом користувача*.

Отже, **графічний інтерфейс користувача** — це візуалізований засіб взаємодії програми з користувачем.

Елементи керування є об'єктами, що характеризуються властивостями і методами. До елементів керування, які використовуються для підтримки діалогу з користувачем при виконанні програми, належать меню, вкладки, текстові поля, командні кнопки, перемикачі, поля вибору, списки, смуги прокручування тощо. Більшість елементів інтерфейсу Windows призначені для організації введення даних користувачем та виведення результатів роботи програми на екран, папір або у файл.

Так, на рис. 51.2 на прикладі діалогового вікна **Шрифт** показано використання деяких основних елементів керування, таких, як вкладки, текстові поля, списки, прапорці, написи, кнопки.

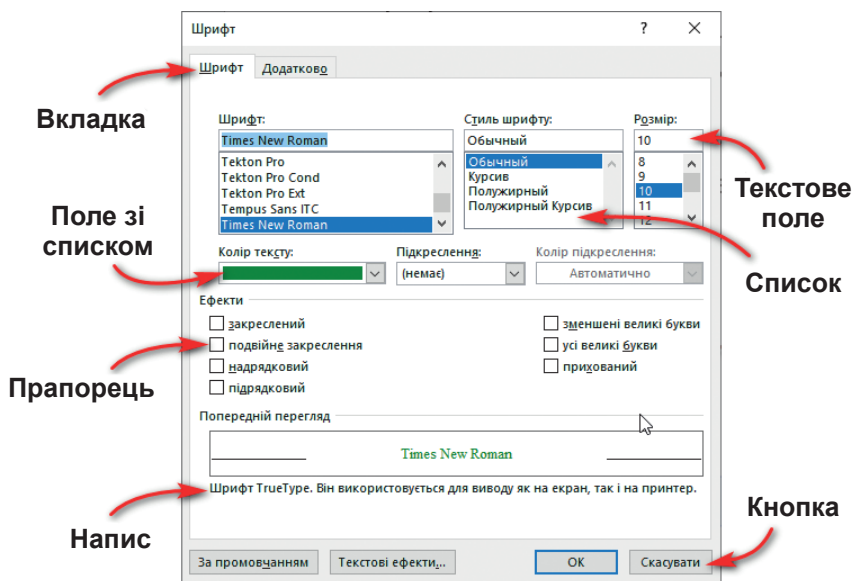


Рис. 51.2. Елементи керування у вікні **Шрифт**



51.3. ПОДІЯ, ОБРОБНИК ПОДІЇ

Для кожного з елементів керування користувач може передбачити деяку подію. Подією програми може бути натиснення певної клавіші на клавіатурі, будь-яка дія миші (клацання або подвійне клацання кнопкою, перетягування об'єкта, наведення курсору на об'єкт тощо). Для кожної події користувач може передбачити різні способи реагування на них, що вимагає вже написання програмного коду за допомогою мови програмування.

Отже, **подія** — це характеристика об'єкта, яка вказує, на яку дію користувача може реагувати об'єкт.

З будь-якою подією можна пов'язати певні операції оброблення даних. Для цього пишуть підпрограму мовою програмування. Для кожної події, яка обробляється, складається окрема підпрограма (програмний код).

Отже, **обробник події** — це підпрограма, яка виконується, як реакція на подію.

51.4. ГРАФІЧНЕ ВІКНО

З допомогою мови програмування Python також можна створювати графічні програми. Для цього в Python за замовчуванням застосовується спеціальний набір компонентів, який називається **tkinter**.

Tkinter доступний у вигляді окремого вбудованого модуля, який містить усі необхідні графічні компоненти — кнопки, текстові поля, написи тощо.

Створення графічного інтерфейсу в Python розпочинають зі створення графічного вікна.

Після чого у ньому додають усі інші складові графічного інтерфейсу (рис. 51.3).

Для створення найпростішого вікна необхідно ввести такий код:

```
from tkinter import*
window=Tk()
```

Тут перший рядок — імпорт модуля **tkinter**. У другому рядку створюється графічне вікно, для чого застосовується конструктор **Tk()**, який визначений у модулі **tkinter**. Створене вікно присвоюється змінній **Window**. Саме через цю змінну ми зможемо керувати атрибутами вікна.

Зверніть увагу, що назва конструктора записується з великої літери.

Після запуску даної програми виконання в середовищі Python IDLE ми одержимо порожнє вікно.

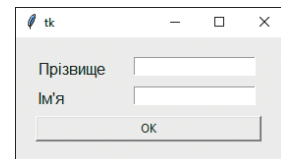


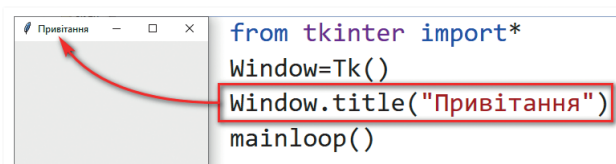
Рис. 51.3. Приклад графічного вікна з елементами керування

Якщо цю ж програму запустити у вікні редактора Visual Studio, нічого не відбудеться. Аби вікно відображалось і при роботі в цьому середовищі, необхідно дописати ще одну команду: `mainloop()`. Ця команда завантажує цикл обробки подій вікна для взаємодії з користувачем. В Python IDLE цей цикл спрацьовує автоматично. Тепер при запуску цієї програми у Visual Studio також буде відображено порожнє вікно.

Для вікна, що було створено, застосовано параметри за умовчанням. Змінимо деякі його налаштування.

Так, з допомогою методу `title()` можна змінити заголовок вікна. Щоб замінити заголовок «tk» на «Привітання», необхідно додати такий код:

```
Window.title("Привітання").
```



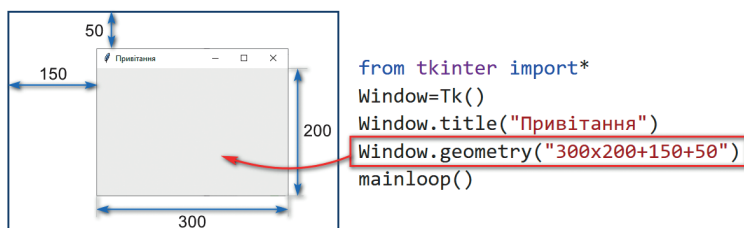
Для вікна також можна встановити його розміри ширини та висоти, що визначаються в пікселях з допомогою методу `geometry()`. Так, для встановлення розмірів вікна 300 пікселів по ширині та 200 пікселів по висоті необхідно записати такий код:

```
Window.geometry("300x200")
```

На першому місці вказується значення ширини, на другому — висоти.

За умовчанням вікно програми розташовано у верхньому лівому куті екрана. Це можна змінити, вказавши у методі `geometry()` необхідні значення відступу зліва та зверху. Так, при встановленні відступу зліва 150 пікселів, а зверху — 50 пікселів, програмний код буде мати вигляд:

```
Window.geometry("300x200+150+50").
```



Отже, загальна структура методу `geometry()` має вигляд:

```
geometry("ширинаxвисота+відступ_зліва+відступ_зверху")
```



Після створення графічного вікна на ньому можна додавати елементи керування. Саме про це піде мова в наступному параграфі.

ВПРАВА 51



Завдання. Створіть графічне вікно із заголовком «Анкета» з розмірами 400 пікселів по ширині і 250 пікселів по висоті. Доберіть параметри відступів зліва та зверху таким чином, аби вікно було розташоване приблизно по центру екрана.



Контрольні запитання та завдання

1. Що таке програмний об'єкт? Назвіть приклади об'єктів у Windows-програмах.
2. Наведіть приклади властивостей об'єкта «кнопка».
3. Що таке метод? Наведіть приклади методів.
4. Що таке графічний інтерфейс користувача?
5. Що таке елементи керування? Наведіть приклади.
6. Що таке подія? Наведіть приклади подій.



Питання для роздумів

- 1*. Розгляньте об'єкт «мобільний телефон». Укажіть, які він має властивості та значення цих властивостей.
- 2*. Опишіть у вигляді набору властивостей об'єкт «учень».



Завдання для досліджень

- 1*. У текстовому редакторі MS Word визначте елементи керування діалогового вікна «Абзац».
- 2*. Визначте, чи можна для методу **geometry** встановити лише відступи графічного вікна від екрана.

§52.**Властивості та методи елементів керування****52.1.****ДОДАВАННЯ НАПISУ**

Як уже було зазначено раніше, до графічного вікна можна додавати різноманітні елементи керування. Одним із найбільш розповсюджених є напис. Розташуємо на графічному вікні напис «Привіт!».

Для цього до коду, створеного в попередньому уроці, необхідно додати дві команди:

```
Lbl=Label(text="Привіт!")  
Lbl.pack()
```

У першому рядку з допомогою конструктора **Label** створюється напис, що має значення «Привіт!». Цей напис присвоюється змінній **Lbl**. У цьому коді **text** — це властивість об'єкта «напис». «Привіт!» — значення властивості **text**.

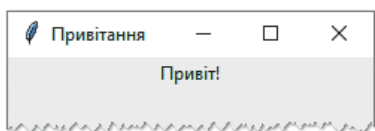


Рис. 52.1

У другому рядку з допомогою методу **pack()** напис розташовується на графічному вікні.

Після виконання даного коду в графічному вікні з'явиться відповідний напис (рис. 52.1).

Додамо до цього напису фразу «Як справи?», змінивши код наступним чином:

```
Lbl=Label(text="Привіт! Як справи?")
```

У результаті після запуску програми фразу буде змінено. А що ж робити, якщо цю фразу необхідно додати знизу під попередньою фразою? Звичайно, можна створити новий об'єкт **Label**, але для цього знадобиться записати ще два рядки коду. Скористаємось більш раціональним способом. Для перенесення тексту на новий рядок достатньо записати комбінацію символів **\n**. Отже, код буде мати наступний вигляд:

```
Lbl=Label(text="Привіт!\nЯк справи?")
```

Звичайно, **text** — не єдина властивість, яку можна встановити для напису. Зробимо, наприклад, у написі літери більшого розміру та червоного кольору. Для цього через кому додамо властивість **font=16**, що визначає розмір шрифту та **fg="red"**, що визначає колір шрифту:

```
Lbl=Label(text="Привіт!\nЯк справи?", font=16, fg="red")
```

У результаті виконання програми властивості шрифту буде змінено (рис. 52.2).



Отже, загальна структура коду для створення елемента керування має вигляд:

```
змінна=конструктор(властивість_1=значення,
                    властивість_2=значення,
                    властивість_3=значення)
```

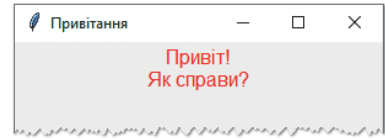


Рис. 52.2

Якщо властивостей небагато, їх записують в один рядок, якщо багато — їх зручніше розташувати у декілька рядків, виконуючи перенесення на наступний рядок після коми.

Для розташування елемента керування у вікні використовується метод `pack()`. Загальний вигляд коду для виклику даного методу має таку структуру:

```
змінна.pack()
```

52.2.

ДОДАВАННЯ ТЕКСТОВОГО ПОЛЯ ТА КНОПКИ

Розглянемо додавання інших елементів керування до графічного вікна на прикладі вікна авторизації, що містить два написи, два текстові поля та кнопку (рис. 52.3).

1. Для створення даного вікна введіть наступний код:

```
from tkinter import*
Windows=Tk()
Windows.geometry("300x150+800+300")
Windows.title("Авторизація")
```

2. Уведіть коди додавання напису «Логін».

```
Login_Lbl=Label(text="Логін", font="Arial 16", fg="blue")
Login_Lbl.pack()
```

Наступний елемент — це текстове поле, в якому потрібно буде вводити логін. Для створення текстового поля використовується конструктор **Entry**.

3. Для розташування у графічному вікні текстового поля додайте такий код:

```
Login_Ent=Entry()
Login_Ent.pack()
```

4. Запустіть програму і переконайтеся, що текстове поле розташоване під написом. Запишіть у ньому деякий текст, наприклад, **user**.

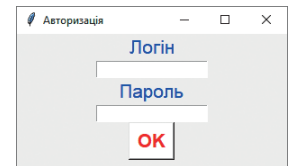


Рис. 52.3

5. Аналогічно додайте напис і текстове поле з паролем.

```
Pass_Lbl=Label(text="Пароль", font="Arial 16", fg="blue")  
Pass_Lbl.pack()  
Pass_Ent=Entry()  
Pass_Ent.pack()
```
6. Запустіть програму. В результаті у вікні мають бути розташовані один під одним два написи та два текстові поля.
7. Уведіть логін **user**, пароль — **1111**.
8. Як правило, у полі, де вводиться пароль, введені символи заміняють зірочками або іншими однаковими символами. Зупиніть роботу програми та додайте для об'єкта **Pass_Ent** властивість `show="*"`.
9. Ще раз запустіть програму та спробуйте ввести пароль. У результаті всі символи буде замінено зірочками.

Нам залишилося знизу ще додати кнопку. Для кнопок використовують конструктор **Button**.

10. Уведіть код додавання до вікна кнопки:

```
Btn=Button(text="OK", font=("arial",16,"bold"),  
           bg="white", fg="red")  
Btn.pack()
```

Тут **Btn** — назва об'єкта, **OK** — напис, що потрібно розташувати на кнопці, **arial** — назва шрифту, **16** — розмір, **bold** — напівжирний шрифт, **bg** — колір фону кнопки, **fg** — колір шрифту кнопки.

11. Запустіть програму, переконайтеся в тому, що необхідні елементи керування додано відповідно до зразка.

ВПРАВА 52.1



Завдання. Створіть вікно програми авторизації відповідно до зразка (рис. 52.3).

52.3.

ПОЗИЦІЮВАННЯ ЕЛЕМЕНТІВ КЕРУВАННЯ

У попередньому прикладі ми розташували елементи керування один над одним, але в більшості додатків об'єкти можуть бути розташовані в різних місцях вікна. Розглянемо, що треба змінити у програмному коді, аби вікно набуло вигляду, як на рис. 52.4.

Для розташування елемента керування у визначеному місці замість методу **pack()** використовують метод **place()**, для якого можна встановити параметри

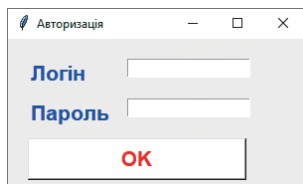


Рис. 52.4



координати розміщення об'єкта, розміри об'єкта тощо. Загальна структура даного методу має вигляд:

```
place(x,y,width,height),
```

де x , y — координати верхнього лівого кута об'єкта, $width$ — ширина, $height$ — висота об'єкта.

Отже, щоб розташувати елементи керування у вікні, необхідно спочатку визначити координати верхнього лівого кута кожного об'єкта, а також, у разі необхідності, їхні розміри. Так, на рис. 52.5 наведено приклад застосування методу `place()` для розташування у вікні кнопки з допомогою коду:

```
Btn.place(x=50, y=100,width=220,height=40)
```

Розглянемо, які значення координат необхідно використати для елементів вікна **Авторизація** (рис. 52.6). Зверніть увагу, що відлік координатної площини вікна починається з верхнього лівого кута. Значення осі x збільшуються з лівого краю до правого. Значення по осі y визначаються зверху вниз.

Отже, для виведення на екрані напису **Логін** необхідно замінити код `Login_Lbl.pack()` на `Login_Lbl.place(x=20, y=20)`, де x та y — координати верхнього лівого кута напису `Login_Lbl`.

Аналогічно необхідно замінити коди розташування у вікні й інших елементів керування.

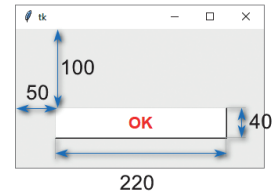


Рис. 52.5

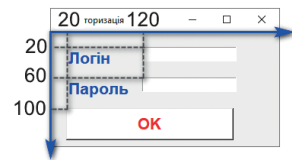


Рис. 52.6

ВПРАВА 52.2



Завдання. Розташуйте елементи керування у вікні **Авторизація** відповідно до зразка (рис. 52.4).

1. Відкрийте програму із вправи 52.1.
2. Для напису **Логін** замініть код `Login_Lbl.pack()` на `Login_Lbl.place(x=20, y=20)`.
3. Аналогічно замініть коди розташування у вікні й інших елементів керування, обираючи в якості необхідних координат значення, зазначені на рис. 52.6.
4. Запустіть програму на виконання. В результаті елементи мають змінити своє розташування у вікні.

52.4.

ВЛАСТИВОСТІ ЕЛЕМЕНТІВ КЕРУВАННЯ

Ми розглянули, як створити вікно авторизації, використовуючи різні властивості елементів керування. Проте цих властивостей знач-



но більше. Так, у таблиці 52.1 наведено основні властивості для напису, тобто об'єкта, створеного з допомогою конструктора **Label**. Далі деякі з них розглянемо більш детально.

Таблиця 52.1. Властивості напису

| Властивість | Опис | Значення |
|----------------|---|--|
| font | шрифт тексту | назва, розмір, накреслення (bold, italic, underline) |
| bg / fg | колір фону / колір шрифту | константа або код |
| bd | товщина контуру | константа або код |
| justify | вирівнювання | LEFT, CENTER, RIGHT |
| padx | відступ від межі елемента до тексту зліва та справа | значення в пікселях |
| pady | відступ від межі елемента до тексту знизу та зверху | значення в пікселях |
| relief | тип межі | SUNKEN, RAISED, GROOVE, RIDGE |

Властивість шрифту

Більш складною за своєю структурою є властивість **font**. Вона може містити значення назви шрифту, його розмір та накреслення.

Наприклад,

`font="arial"` — встановлення назви шрифту;

`font="arial 24"` — встановлення назви та розміру;

`font="arial 24 bold"` — встановлення назви, розміру та накреслення.

З останнього прикладу видно, що властивості шрифту можна записувати в лапках через пробіл. Проте цей спосіб не підійде, коли назва шрифту має більше ніж одне слово. В цьому випадку можна скористатися другим способом, записуючи параметри властивості в дужках, відокремлюючи комою. Наприклад,

```
font=("comic sans ms", 24, "bold", "italic", "underline")
```

У цьому прикладі `bold`, `italic`, `underline` — це накреслення, що приймає значення напівжирний, курсив та підкреслений відповідно.

Властивості кольору

Властивості **bg** (**background**) та **fg** (**foreground**) установлюють колір фону та шрифту.

При встановленні значення властивості кольору, крім англійських слів, що визначають колір (`red`, `blue` тощо), можна також використовувати код кольору у форматі RGB шістнадцяткового подання. Так, наприклад, команди `bg="red"`, `bg="#F00"`, `bg="#FF0000"` будуть указу-



вати на однакове значення властивості заливки фону червоним кольором.

У разі необхідності, коди кольорів можна легко знайти в Інтернеті у палітрах кольорів, наприклад, на сайті: html-color-codes.info.

Ми розглянули основні властивості елемента керування «напис». Більшість розглянутих властивостей містять також й інші елементи керування.



Дізнайтеся більше про властивості вирівнювання та типи меж



Дізнайтеся більше про особливі властивості кнопки



Контрольні запитання та завдання

1. Як додати до графічного вікна такі елементи, як напис, текстове поле, кнопку?
2. Як в одному об'єкті *напис* розташувати текст у декілька рядків?
3. Чим відрізняються методи **pack** та **place**?
4. Яка властивість використовується для зміни шрифту? Опишіть її синтаксис.
5. Як установити колір фону та шрифту елемента керування?
6. Для чого використовується властивість **state**?



Питання для роздумів

- 1*. Які елементи керування, розглянуті в цьому параграфі, ви використовували раніше у Windows-програмах? Наведіть приклади.
- 2*. Чим властивість об'єкта відрізняється від методу?



Завдання для досліджень

- 1*. Знайдіть, які ще властивості використовуються для елементів напису, текстового поля та кнопки.



Пройдіть
тест

§53.

Обробники подій, пов'язані з елементами керування

53.1.

ПОДІЯ НАТИСКАННЯ НА КНОПКУ

Ми вже навчилися до графічного вікна додавати елементи керування, проте — це лише графічний інтерфейс програми. У справжніх програмах елементи керування мають взаємодіяти з користувачем, реагувати на його дії. Це можна зробити з використанням обробників подій, про які й піде мова далі.

Як було зазначено раніше, **обробник події** — це підпрограма, яка виконується, як реакція на подію. Подією може бути натиснення миші на об'єкт (кнопку, напис, зображення тощо), наведення миші, подвійне клацання на об'єкт тощо. Однією із найбільш розповсюджених подій є подія натискання на кнопку. Розглянемо, як запрограмувати дію кнопок, виконавши вправи 53.1 та 53.2.

ВПРАВА 53.1



Завдання. У вікні розташуйте 4 кнопки, що відповідають порам року **Зима, Весна, Літо** та **Осінь**. Ліворуч від кнопок розташуйте об'єкт **напис** сірого кольору з увігнутим рельєфом. При натисненні на відповідну кнопку в об'єкті **напис** мають з'являтися назви місяців, що відповідають натисненій порі року.

1. Уведіть команди імпорту модуля **tkinter** та створення графічного вікна із заголовком **Пори року** і з розмірами 300 на 250.

```
from tkinter import*
#створення вікна
Window=Tk()
Window.title("Пори року")
Window.geometry("300x250")
```

2. Додайте команди створення кнопок **Зима, Весна, Літо** та **Осінь** з назвами **winter, spring, summer** та **autumn** відповідно. Встановіть розмір шрифту 16 пунктів, а колір фону відповідно до зразка (рис. 53.1).

```
#кнопки
winter = Button(text="Зима",font=16,bg="#FFFFFF")
winter.place(x=20, y=20, width=100)
spring = Button(text="Весна",font=16,bg="#FFA500")
spring.place(x=20, y=70, width=100)
summer = Button(text="Літо",font=16,bg="#00FFFF")
summer.place(x=20, y=120, width=100)
autumn = Button(text="Осінь",font=16,bg="#00FF7F")
autumn.place(x=20, y=170, width=100)
```



- Додайте команди створення порожнього напису, що має назву **Lbl**. Для літер, що мають відображатися у цьому написі, встановіть розмір 16 пунктів, сірий колір фону, увігнутий рельєф напису.

#напис

```
Lbl= Label(text="", font=16, bg="#AAAAAA",
relief=SUNKEN)
Lbl.place(x=140, y=10, width=100, height=200)
```

- Додайте команду `mainloop()`.
- Запустіть програму на виконання та переконайтеся, що елементи керування створено.

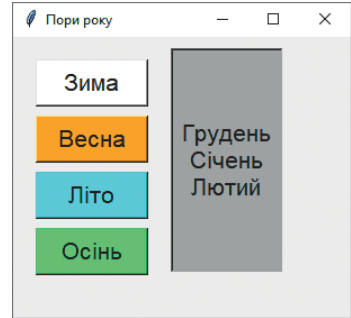


Рис. 53.1

53.2.

ПОДІЯ НАТИСКАННЯ КНОПКИ

Розглянемо створення кодів обробника події на прикладі двох об'єктів: кнопки **Зима**, на яку необхідно натиснути (вона має ім'я **winter**) та напису **Lbl**, в якому відображаються назви місяців (рис. 53.2).

Для зміни тексту в написі ми маємо створити функцію, яка виконає зміну властивості **text** у написі **Lbl**. Для цього можна ввести наступний код:

```
def winter_click():
    lbl['text'] = "Грудень\nСічень\nЛютий"
```

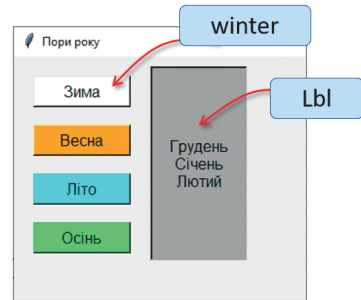


Рис. 53.2

Тут **def** — службове слово, що вказує на створення функції, **winter_click** — назва функції, **Lbl** — назва напису, 'text' — назва властивості. Напис у лапках — значення, що надається властивості. Символи `\n` застосовано для того, щоб назви місяців розпочиналися з нового рядка.

Зверніть увагу, що подія, яка має відбуватися у функції, записується з відступом.

Аби вказані у функції дії були виконані, її треба викликати. Здійснюється це в конструкторі кнопки, на яку буде натиснуто.

```
winter = Button(text="Зима", font=16, bg="#FFFFFF",
                command=winter_click)
```

З даної команди видно, що в конструкторі кнопки **winter** дописано команду виклику функції `command=winter_click`, де **winter_click** — це назва функції обробника події.

Зверніть увагу, що коди функції необхідно записати до того, як її викликають.

На рис. 53.3 представлено загальні конструкції щойно розглянутих команд — запам'ятайте їх!

Загальна структура функції обробника події:

```
def ім'я_функції():  
    події
```

Зміна властивості об'єкта:

```
ім'я_об'єкта['властивість'] = значення_властивості
```

Виклик події натискання кнопки:

```
command=ім'я_функції
```

Рис. 53.3

ВПРАВА 53.2



Завдання. Відкрийте файл, створений у вправі 53.1, та створіть обробники подій натискання кнопок.

1. Створіть код обробника події натискання на кнопку **Зима**.
2. В конструкторі кнопки **Зима** додайте команду виклику обробника події.
3. Запустіть програму на виконання та натисніть кнопку **Зима**. В результаті у написі мають з'явитися назви зимових місяців.
4. Самостійно додайте функції обробника події натискання на інші кнопки. Також додайте команди виклику даних функцій у відповідних конструкторах.

53.3.

ПРОГРАМУВАННЯ РІЗНИХ ПОДІЙ

Натискання миші на кнопки — це не єдина подія, що може відбуватися з об'єктами. Розглянемо програмування інших подій на прикладі програми «Словник», у якій події будуть відбуватися з написами.

ВПРАВА 53.3



Завдання. У графічному вікні розташуйте написи у два стовпчики відповідно до зразка (рис. 53.4). При запуску програми має завантажуватися вікно, що містить у першому стовпці слова на українській мові, у другому стовпці — слова приховані. При натисканні на українське слово має з'являтися його переклад англійською мовою. При подвійному натисканні на українське слово, англійське слово має зникати.

1. Уведіть команди імпорту модуля **tkinter** та створення графічного вікна із заголовком **Словник** та розмірами 300 на 200.



```
from tkinter import*
Window=Tk()
Window.title("Словник")
Window.geometry("300x200")
```

- Уведіть команди створення написів, що містять українські слова. Встановіть для них відповідні написи, розмір шрифту 16, колір — світло-зелений.

```
#українські слова
Ua1 = Label(text="Кіт",font=16,bg="#00FF7F")
Ua1.place(x=20, y=20, width=100)
Ua2 = Label(text="Собака",font=16,bg="#00FF7F")
Ua2.place(x=20, y=60, width=100)
Ua3 = Label(text="Лисиця",font=16,bg="#00FF7F")
Ua3.place(x=20, y=100, width=100)
```

- Самостійно додайте команди створення порожніх написів для розташування слів англійською мовою. Встановіть для них розмір шрифту 16, колір фону — оранжевий, увігнутий рельєф контуру.
- Додайте команду `mainloop()`.
- Запустіть програму на виконання та переконайтеся, що елементи керування створено.

Розглянемо створення кодів на прикладі двох об'єктів: напису `Кіт`, на який необхідно буде здійснювати натискання, що має назву `Ua1`, та напису `En1`, у якому має з'явитися слово `Cat` (рис. 53.5).

Для зміни тексту в написі ми маємо створити функцію, яка виконає зміну властивості `text` у написі `En1`. Для цього можна ввести такий код:

```
def En1_show(event):
    En1['text']="Cat"
```

Як вам уже відомо, `def` — службове слово, що вказує на створення функції, `En1_show` — назва функції. На відміну від обробника події, що ми створювали для натискання кнопки, тут для функції з'явився аргумент `even`, що є об'єктом події. Для виклику даної функції використовується метод `bind()`, який дозволяє викликати обробника події для будь-якого елемента керування. Програмний код виклику події буде таким:

```
Ua1.bind('<Button-1>', En1_show)
```

`Ua1` — це об'єкт, з яким відбувається дія, `bind` — метод виклику події, `<Button 1>` — це подія, в даному випадку вона визначає натискання лівої кнопки миші, `En1_show` — ім'я обробника події.

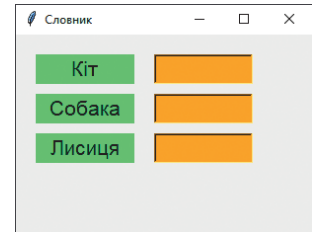


Рис. 53.4

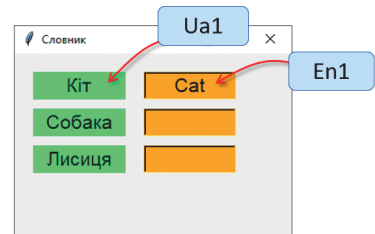


Рис. 53.5

ВПРАВА 53.4



Завдання. Відкрийте файл, створений у вправі 53.3, та створіть обробники подій наведення на написи з українськими словами; подвійного клацання на написи з українськими словами.

1. Створіть код обробника події натискання на слово **Кіт**.
2. Додайте команду виклику події.
3. Запустіть програму на виконання та натисніть слово **Кіт**. У результаті навпроти має з'явитися напис **Cat**.
4. Додайте обробники подій наведення на інші написи з українськими словами. Також додайте команди їхнього виклику.

Для виконання події подвійного натискання на напис **Ua1** створимо функцію:

```
def En1_hide(event):  
    En1['text']=""
```

У даному випадку функція має назву **En1_hide**, в результаті виконання події значення тексту в написі **En1** має стати порожнім.

Для виклику даного обробника події скористаємося знову методом `bind`. Отже, уся команда матиме вигляд:

```
Ua1.bind('<Double-Button-1>', En1_hide)
```

Тут `'<Double-Button-1>'` — подія подвійного клацання миші. Отже, даний код означає, що при виконанні події подвійного клацання миші на об'єкті **Ua1** буде виконано обробник події **En1_hide**, а саме, прибирання тексту з напису.

5. Додайте код обробника події подвійного натискання на слово **Кіт**.
6. Додайте команду виклику цієї події.
7. Запустіть програму на виконання та натисніть слово **Кіт**. В результаті навпроти має з'явитися напис **Cat**. Клацніть на слові **Кіт** два рази — слово **Cat** має зникнути.
8. Самостійно додайте коди для виконання відповідних дій з іншими словами.

На рис. 53.6 представлено загальні конструкції щойно розглянутих команд — запам'ятайте їх!

У двох попередніх вправах ми розглянули використання двох подій — натискання та подвійне натискання на об'єкт мишею.

В таблиці 53.1 на с. 225 наведено ще деякі можливі події, що відбуваються з допомогою миші.



Загальна структура функції обробника події:

```
def ім'я_функції(event):
    події
```

bind – метод виклику обробника події

Виклик обробника події:

```
ім'я_об'єкта.bind(подія, ім'я_функції)
```

Синтаксис події:

```
'<Подія>' або "<Подія>"
```

Рис. 53.6

Таблиця 53.1. Події, що відбуваються з використанням миші

| Подія | Опис |
|-------------------|---------------------------------------|
| <Button-1> | натискання лівої кнопки миші |
| <Button-2> | натискання середньої кнопки миші |
| <Button-3> | натискання правої кнопки миші |
| <Double-Button-1> | подвійне натискання лівої кнопки миші |
| <Motion> | рух миші |



Контрольні запитання та завдання

1. Що необхідно для створення обробника події натискання кнопки?
2. Як викликати обробник події натискання кнопки?
3. Які обробники подій з використанням миші можна запрограмувати?
4. Як викликати обробник події з використанням миші?



Питання для роздумів

- 1*. Чим відрізняється програмування обробника події натискання кнопки з обробниками виконання дій з іншими елементами керування?
- 2*. Як з допомогою обробника події можна змінити властивості кольору фону; шрифту?



Завдання для досліджень

- 1*. Знайдіть інформацію про створення обробників подій натискання клавіш на клавіатурі.

Пройдіть
тест

§54.

Уведення даних

54.1.

УВЕДЕННЯ ДАНИХ ІЗ ТЕКСТОВОГО ПОЛЯ

Щоб у програмі обробити якусь інформацію, її спочатку потрібно ввести — для цього використовують елементи керування, про які піде мова далі.

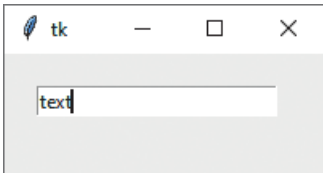


Рис. 54.1

Найчастіше вживаним елементом введення даних є текстове поле, що утворюється за допомогою конструктора `Entry`. Для отримання значення, яке містить текстове поле, використовують метод `get()`. Так, для випадку, представленого на рис. 54.1 після виконання команди `x=ent.get()` змінна `x` одержить значення `text`.

Розглянемо використання даного методу, виконавши практичну вправу.

ВПРАВА 54.1



Завдання. Користувач має ввести у текстовому полі, скільки йому років. Після введення числа та натиснення кнопки **ОК** у вікні консолі має з'явитися повідомлення у форматі «Тобі 12 років», де 12 — число, введене користувачем.

1. Створіть інтерфейс програми, використовуючи наступний код (рис. 54.2):

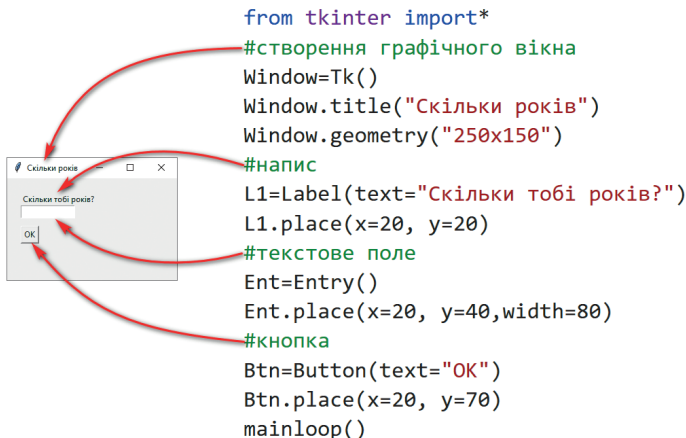


Рис. 54.2



2. Додайте до програмного коду обробник події натискання на кнопку.

```
def btn_click():
    x=int(Ent.get())
    print("Тобі",Ent.get(),"років.")
```

У першій команді змінній *x* надається значення текстового поля **Ent** з допомогою методу **get**. Причому, оскільки значення змінної *x* має бути цілим числом, до команди **Ent.get** застосовано функцію **int**. У другому рядку виводиться повідомлення з допомогою команди **print**. Спочатку виводиться слово «Тобі», до нього дописується одержане із текстового поля **ent** значення змінної *x*, далі дописується слово «років».

3. У конструкторі кнопки додайте команду виклику обробника події.

```
Btn=Button(text="OK",command=btn_click)
```

4. Перевірте, як буде працювати створена програма. Запустіть програму. У текстовому полі введіть 14. Натисніть кнопку. В результаті у вікні консолі має з'явитися повідомлення: Тобі 14 років.

Звичайно, якщо ви працюєте у графічному режимі, то використання консолі не є зручним. Змінимо код таким чином, аби текст було виведено до напису у графічному вікні.

5. Додайте команди створення порожнього напису **L2**.

```
L2=Label()
L2.place(x=20, y=110)
```

6. У коді обробника події натискання кнопки замініть команду виведення напису наступним кодом:

```
L2['text']="Тобі",Ent.get()," років."
```

Тут **L2** — елемент керування, до якого буде виведено напис, **text** — властивість виведення тексту. Після знака «дорівнює» записано той же код, що і був у команді **print**.

7. Запустіть програму на виконання. Введіть 14, натисніть кнопку. В результаті повідомлення має з'явитися у написі графічного вікна.

Удосконалимо обробник події таким чином, аби слово «рік» відмінювалося в залежності від введеного числа. Так, якщо число закінчується на 1, але не на 11, має виводитися «рік», якщо на 2 або 3 або 4, але не на 12, 13, 14, має виводитися «роки», в інших випадках — «років».

8. Для врахування зазначених вище умов уведіть такий код:

```
if x%10==1 and x%100!=11:
    L2['text']="Тобі",Ent.get()," рік."
```

```
elif (x%10==2 or x%10==3 or x%10==4) and x%100!=12 and x%100!=13
    and x%100!=14:
    L2['text']="Тобі",Ent.get()," роки."
else:
    L2['text']="Тобі",Ent.get()," років."
```

У першій умові даного коду буде знайдено останню цифру числа та перевірено, чи дорівнює вона одиниці, також знайдено останні дві цифри числа та перевірено, що вони не дорівнюють 11. Якщо обидві ці умови виконуються, останнє слово буде виведено «рік».

У другій умові також буде здійснено пошук останньої цифри числа і перевірено, що вона дорівнює 2 або 3, або 4. Також буде знайдено останні дві цифри числа і перевірено, що вони не дорівнюють ні 12, ні 13, а ні 14. Якщо ця складена умова виконується, то останнє слово буде виведено «роки».

В останньому рядку програми буде виведено повідомлення для випадку, якщо жодна із попередніх умов не виконується.

9. Запустіть програму на виконання та перевірте її роботу при врахуванні різних умов.

Знайомство з іншими можливостями використання текстових полів розглянемо, виконавши завдання наступної вправи.

ВПРАВА 54.2



Завдання. Створити програму, що містить вікно реєстрації учасників змагань відповідно до наведеного зразка (рис. 54.3). Після введення даних у текстові поля та натиснення

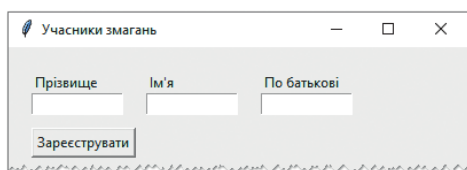


Рис. 54.3

кнопки **Зареєструвати** дані записуються до текстового напису, розташованого під кнопкою, а текстові поля очищуються. При подальшому введенні даних і натисненні кнопки **Зареєструвати** дані розташовуються під попереднім записом.

1. Уведіть команди імпорту модуля **tkinter** та створення графічного вікна із заголовком **Учасники змагань** та з розмірами 300 на 400.
2. Додайте команди створення написів **Прізвище**, **Ім'я** та **По батькові**, надайте їм імена **Pr**, **Im**, **Pb** відповідно.
3. Додайте команди створення текстових полів, що мають імена **Pr_Ent**, **Im_Ent**, **Pb_Ent**.
4. Додайте команди створення кнопки **Зареєструвати**, що має ім'я **Btn**.



5. Додайте команди виведення порожнього напису для додавання реєстраційних даних. Розташуйте цей напис під кнопкою **Зареєструвати**.

```
#напис реєстрації
L=Label(text="",justify=LEFT)
L.place(x=20,y=120)
```

6. Уведіть код обробника події натискання кнопки.

```
def btn_click():
    txt=Pr_ent.get()+" "+Im_ent.get()+" "+Pb_ent.get()
    L['text']=txt
```

З даного коду видно, що перший рядок обробника події з'єднує через пробіл прізвище, ім'я та по батькові, введені до відповідних текстових полів. У другому рядку одержане значення присвоюється напису, розташованому під кнопкою.

7. Запустіть програму на виконання. Введіть **Петренко Сергій Степанович**. Натисніть **Зареєструвати**. В результаті під кнопкою має з'явитися введений напис.

Далі розглянемо, щоб при натисненні кнопки **Зареєструвати** також було видалено введені написи.

Для вилучення написів із текстового поля використовують метод `delete(first,last)`. Цей метод видаляє символ за вказаним індексом **first**. При цьому нумерація символів починається з нуля. Параметр **last** є необов'язковим і вказує на номер індексу, до якого здійснюється видалення. Якщо в якості другого параметра вказати значення **END**, видалення здійснюється до кінця тексту. Отже, для видалення всього тексту із деякого текстового поля **Ent** необхідно скористатися командою: `Ent.delete(0,END)`.

8. Додайте до обробника події натискання кнопки команди вилучення тексту з кожного текстового поля:

```
Pr_ent.delete(0,END)
Im_ent.delete(0,END)
Pb_ent.delete(0,END)
```

У даних кодах на першому місці вказано імена текстових полів, далі, через крапку, — метод `delete` з відповідними параметрами.

9. Ще раз запустіть програму на виконання. Введіть **Петренко Сергій Степанович**, натисніть **Зареєструвати**, тепер написи з текстових полів мають зникнути. Введіть до них інший напис: **Антонюк Оксана Михайлівна**. Натисніть **Зареєструвати**. В результаті попередній напис буде замінено на новий.

Проте відповідно до завдання вправи потрібно, аби нові дані не замінювали попередні, а розташовувалися нижче. Отже, текст має дописуватися до попереднього з нового рядка.

10. Змініть програмний код у формуванні значення змінної `txt`. Для цього додайте після знака «дорівнює» `txt+`, що означає додати до наявного вже значення змінної `txt` те, що вказано після знака «плюс». В кінці команди додайте символи `"\n"`, які означають перехід на наступний рядок:

```
txt=txt+Pr_ent.get()+" "+Im_ent.get()+" "+Pb_ent.get()+"\n"
```

11. Ще раз запустіть програму. Введіть прізвище, ім'я та по батькові, натисніть **Зареєструвати**. В даному випадку у вікні консолі має з'явитися помилка, пов'язана зі змінною `txt`.
12. Виправте помилку. Для цього необхідно надати початкове значення змінній `txt` перед обробником події: `txt=""`. Також потрібно у другому рядку функції обробника події записати команду `global txt`. Ця команда вказує, що змінна `txt` є глобальною, тобто використовується не лише в рамках даної функції, а й у тілі програми.
13. Ще раз запустіть програму на виконання. Введіть **Петренко Сергій Степанович**. Натисніть **Зареєструвати**. Введіть ще дані: **Адаменко Оксана Михайлівна**, знову натисніть **Зареєструвати**. Тепер другий уведений напис має з'явитися під попереднім, і написи з текстових полів вилучено.

54.2.

ВІКНО ВВЕДЕННЯ ДАНИХ

Уведення даних також можна організувати з допомогою діалогових вікон. Для реалізації попереднього прикладу дані можна вводити не з допомогою текстових полів, а з використанням наступних діалогових вікон, що будуть з'являтися один за одним (рис. 54.4).

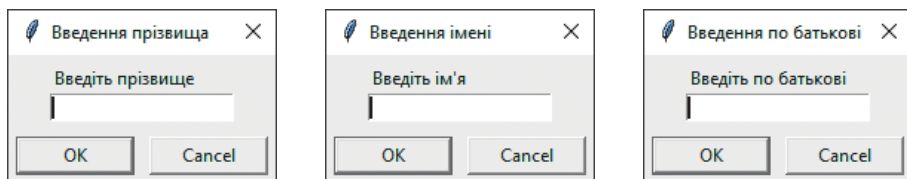


Рис. 54.4

У цьому випадку у графічному вікні вже не буде текстових полів та відповідних до них написів. Після натиснення кнопки **Зареєструвати** має з'явитися перше вікно, в якому необхідно ввести прізвище, після введення прізвища і натиснення кнопки **ОК** має з'явитися вікно для введення імені, а після цього — вікно для введення по батькові. Після введення даних у третьому вікні і натиснення кнопки **ОК** уве-



дені дані, як і раніше, мають розташовуватися у написі під кнопкою **Зареєструвати**.

Для створення діалогових вікон уведення даних використовують такий синтаксис:

```
змінна=askstring(заголовок,запрошення)
```

Так, наприклад, для створення вікна, в якому необхідно ввести прізвище (рис. 54.5), можна застосувати такий програмний код:

```
pr=askstring("Введення прізвища","Введіть прізвище")
```

Після введення деякого тексту до текстового поля і натискання кнопки **ОК** уведене значення буде надано змінній **pr**.

Для використання діалогових вікон уведення даних необхідно імпортувати модуль **simpledialog**, що входить до складу **tkinter**. Це можна виконати з допомогою наступного коду:

```
from simpledialog.tkinter import*
```

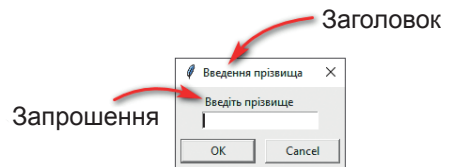


Рис. 54.5

ВПРАВА 54.3



Завдання. Створіть копію програми із вправи 54.2 та виконайте в ній зміни таким чином, аби введення даних для реєстрації було виконано з допомогою вікон уведення.

1. Вилучіть із програми команди створення написів та текстових полів, оскільки у новій версії програми вони відсутні.
2. Також вилучіть команди очищення текстових полів у коді обробника події кнопки.
3. У другому рядку програми запишіть код імпорту модуля: **simpledialog.tkinter**.
4. Додайте команди, що виводять діалогові вікна для введення прізвища, імені та по батькові. Значення, що будуть уведені до даних вікон, надайте змінним **pr**, **im** та **pb** відповідно (рис. 54.4).
5. Змініть код присвоєння змінній **txt** наступним чином:


```
txt=txt+pr+" "+im+" "+pb+"\n"
```
6. Запустіть програму та перевірте правильність її виконання.

**Контрольні запитання та завдання**

1. Як змінній надати значення тексту із текстового поля?
2. Як видалити текст із текстового поля?
3. Як створити вікно введення даних?
4. Які об'єкти графічного інтерфейсу можна використовувати для введення даних?

**Питання для роздумів**

- 1*. Що відбувається у разі натиснення кнопки **Cancel** у вікні введення даних?
- 2*. В чому переваги та недоліки використання вікон введення даних і текстових полів?

**Завдання для досліджень**

- 1*. Чи є обов'язковими обидва параметри в `askstring(заголовок, запрошення)`?
- 2*. Як до текстового поля додати деякий текст?

Пройдіть
тест**§55.****Використання вікон повідомлень**

Зручним засобом представлення даних є вікна повідомлень, виведення яких здійснюється при виконанні програмного коду. Саме з тим, як закодувати вікна повідомлень, ми і познайомимося далі.

Вікна повідомлень бувають різних типів. Тип вікна залежить від значка, що в ньому розташований, та наявних у ньому кнопок. Це можуть бути вікна, що містять запитання, вікна, що повідомляють про помилку, а також інформаційні вікна, що позначаються відповідними піктограмами (рис. 55.1).

Програмний код для виведення вікна повідомлення має таку загальну конструкцію:

```
змінна=функція_вікна_повідомлення(заголовок,повідомлення)
```

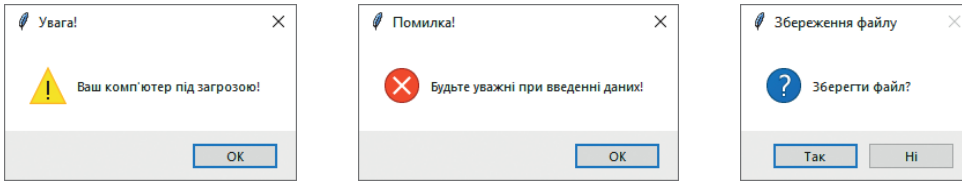


Рис. 55.1. Види вікон повідомлень

Так, наприклад, для виведення вікна, представленого на рис. 55.2, буде використано такий код: `x=showinfo("Вітання", "Ви зареєстровані!")`.

Для виведення вікон повідомлень, використовують модуль **messagebox** вбудований до **tkinter**. Отже, для використання в програмі таких вікон ми маємо спочатку імпортувати цей модуль. Це можна виконати з допомогою наступного коду:

```
from tkinter.messagebox import*
```

У таблиці 55.1 наведено можливі види вікон повідомлень.

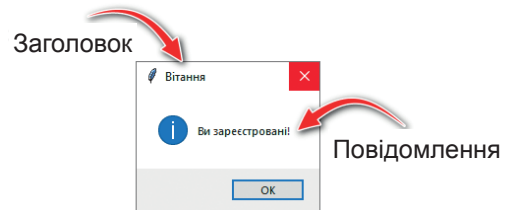


Рис. 55.2. Інформаційне вікно повідомлення

Таблиця 55.1. Види вікон повідомлень

| Функція | Вид | Значок | Кнопки | Значення результату |
|-----------------------------|--------------------|--------|----------------------|---------------------|
| <code>askyesno</code> | вікна запитань | ? | Так, Ні | True, False |
| <code>askquestion</code> | | | Так, Ні | yes, no |
| <code>askokcancel</code> | | | ОК, Скасувати | True, False |
| <code>askyesnocancel</code> | | | Так, Ні, Скасувати | True, False, None |
| <code>askretrycancel</code> | | | Повторити, Скасувати | True, False |
| <code>showwarning</code> | інформаційні вікна | ! | ОК | ok |
| <code>showinfo</code> | | i | ОК | ok |
| <code>showerror</code> | вікна помилок | X | ОК | ok |

З таблиці видно, що вид вікна залежить від вибору функції. Натиснення тієї чи іншої кнопки, що буде відображена у вікні, призводить до надання змінній відповідного значення, наведеного в останньому стовпчику. Більш детально використання вікон повідомлень розглянемо, виконавши завдання вправи.

ВПРАВА 55.1



Завдання. Створити тренажер виконання дії додавання з двоцифровими числами. У вікні випадковим чином має з'являтися приклад, відповідь на який необхідно ввести у текстове поле. Після натискання кнопки **Перевірити** має з'явитися повідомлення про правильність виконання прикладу. Кнопка **Наступний приклад** має вивести інший приклад. Завдання, що містять математичні вирази, мають генеруватися випадковим чином. Також у написах вивести номер прикладу, кількість правильних відповідей та кількість помилок.

Створення графічного інтерфейсу

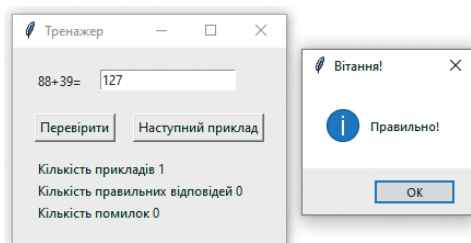


Рис. 55.3

Створіть графічний інтерфейс вікна програми відповідно до зразка (рис. 55.3).

1. Уведіть команди імпорту модулів **tkinter** та **tkinter.messagebox**.

```
from tkinter import*  
from tkinter.messagebox import*
```
2. Додайте команди створення графічного вікна із заголовком **Тренажер** та з розмірами 250x180.

3. Додайте команди для виведення порожнього напису **Lbl**, у якому буде відображено приклад, та текстового поля **Ent** для введення результату обчислення.
4. Додайте команди розташування у вікні кнопок **Перевірити** та **Наступний приклад**, надайте їм імена **btn1** та **btn2** відповідно.
5. Додайте команди створення написів, у яких буде виводитися статистика виконання обчислень, а саме, кількість прикладів, кількість правильних відповідей та кількість помилок.

```
#статистика  
Lbl_k=Label(text="Кількість прикладів 1")  
Lbl_k.place(x=20, y=100)  
Lbl_k_ok=Label(text="Кількість правильних відповідей 0")  
Lbl_k_ok.place(x=20, y=120)  
Lbl_k_err=Label(text="Кількість помилок 0")  
Lbl_k_err.place(x=20, y=140)
```

З даного коду видно, що в момент завантаження графічного вікна кількість прикладів буде 1, оскільки при завантаженні вікна відразу буде виведено перший приклад. А кількість правильних та неправильних відповідей на початку буде дорівнювати 0.



Створення прикладу

- Виконайте зміни у блоці формування прикладу. Для цього змінній **a** надайте значення першого доданка, що є цілим випадковим числом у проміжку від 10 до 99. Аналогічно змінній **b** надайте значення другого доданка. У написі **Lb1** сформууйте приклад, що складається з цих двох доданків.

```
#приклад
a=randint(10,99)
b=randint(10,99)
Lb1=Label(text=str(a)+" "+str(b)+"=")
...
```

- Оскільки у програмі використовується функція створення випадкових чисел, додайте на початку програми команду імпорту модуля **random**.

Обробник події кнопки **Перевірити**

- Сформууйте обробник події натискання кнопки **Перевірити**. Для цього введіть такий код:

```
def btn1_click():
    if Ent.get()==str(a+b):
        answer = showinfo("Вітання!", "Правильно!")
    else:
        answer=showerror("Помилка!", "Неправильно! Спробуй ще раз!")
```

Перший рядок обробника події перевіряє, чи дорівнює значення, введене до текстового поля **Ent**, сумі значень змінних **a** і **b**. Функція **str** переводить числове значення в текстове. Якщо умова виконується, то буде виведено повідомлення в інформаційному вікні **showinfo**. Якщо умова не виконується, буде виведено вікно помилки **showerror**.

- Запустіть програму. Введіть неправильну відповідь та натисніть **Перевірити**. В результаті має з'явитися вікно з повідомленням про помилку. Введіть правильну відповідь. Натисніть **Перевірити** — тепер має з'явитися вікно з повідомленням про правильне розв'язання.

Обробник події кнопки **Наступний приклад**

- Для обробника події натискання на кнопку **Наступний приклад** сформууйте наступний код:

```
def btn2_click():
    global a,b,k
    a=randint(10,99)
    b=randint(10,99)
    Lb1=Label(text=str(a)+" "+str(b)+"=")
    Lb1.place(x="20", y="20")
    Ent.delete(0,END)
```



По суті, він дублює фрагмент програми початкового створення прикладу при запуску програми. Тут аналогічно утворюються змінні **a** і **b** та формується текст у написі **Lbl**. Також після натискання кнопки **Наступний приклад** буде здійснено очищення текстового поля. Це реалізовано з допомогою методу **delete**. Оскільки сума значень змінних **a** і **b** перевіряється у тілі іншої функції, їх оголошено в якості глобальних змінних у першому рядку обробника події.

11. Запустіть програму та перевірте, чи будуть змінюватися приклади, натискаючи кнопку **Наступний приклад**.

Створення статистики обчислень

Удосконалимо програму таким чином, аби у ній здійснювався підрахунок кількості розглянутих прикладів, кількості правильних відповідей та кількості помилок. Для визначення цих значень скористаємося змінними **k** — кількість прикладів, **k_ok** — кількість правильних відповідей та **k_err** — кількість помилок.

12. У тілі програми встановіть початкові значення змінних:

```
k=1
k_err=0
k_ok=0
```

Оскільки перший приклад формується відразу при запуску програми, то змінна **k=1**.

13. Додайте у тілі обробника кнопки **Наступний приклад** коди обчислення кількості прикладів, а саме, значення змінної **k**:

```
def btn2_click():
    global a,b,k
    ...
    #формування та виведення значення k
    Lbl_k['text']="Кількість прикладів "+str(k)
    #збільшення k на 1 після виведення нового прикладу
    k=k+1
```

14. Коди обчислення кількості правильних та неправильних відповідей додайте у тілі обробника кнопки **Перевірити**.

```
def btn1_click():
    global k_err, k_ok
    if Ent.get()==str(a+b):
        answer = showwarning("Вітання!", "Правильно!")
        k_ok=k_ok+1
        Lbl_k_ok['text']="Кількість правильних відповідей "+str(k_ok)
    else:
        answer=showerror("Помилка!","Неправильно! Спробуйте ще раз!")
```




```
k_err=k_err+1
lbl_k_err['text']="Кількість помилок "+str(k_err)
```

У коді здійснюється оголошення змінних `k_ok` та `k_err` як глобальних. Якщо відповідь є правильною, буде здійснюватися збільшення значення змінної `k_ok` та зміна цього значення при виведенні його у графічному вікні, у випадку невиконання умови збільшується значення `k_err`, а також формується напис щодо виведення цього значення.

15. Запустіть програму та перевірте, як буде підраховуватися кількість прикладів.



Контрольні запитання та завдання

1. Які бувають вікна повідомлень?
2. Як вивести вікно повідомлення?
3. Які елементи має вікно повідомлення?



Питання для роздумів

- 1*. Для чого використовують вікна повідомлень у програмах?
- 2*. Для чого у вікнах повідомлень використовують різні кнопки?



Пройдіть
тест

§56.

Використання перемикачів та прапорців

56.1.

ВИКОРИСТАННЯ ПЕРЕМИКАЧІВ

Для взаємодії з прикладною програмою користувачеві можна не лише надати можливість уводити інформацію з клавіатури, а й обрати певні варіанти із запропонованих, використовуючи перемикачі або прапорці. Далі мова піде про використання перемикачів.

Перемикачі використовують, коли користувач має обрати лише один варіант із набору запропонованих. Так, ці елементи керування доцільно використовувати для створення тестів, опитувань тощо.

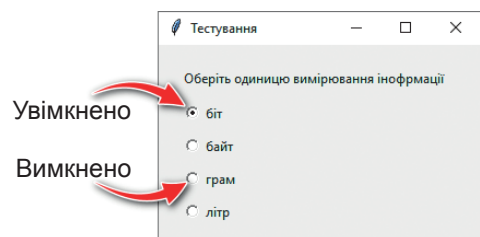


Рис. 56.1

Перемикачі створюють за допомогою конструктора **Radiobutton**. Ці елементи керування можуть перебувати в одному із двох станів: увімкнено або вимкнено (рис. 56.1). Загальний вигляд коду для створення перемикача буде таким:

```
змінна=Radiobutton(властивості)
```

Для виведення перемикача у графічному вікні, як і для інших елементів керування, використовують методи **pack** або **place**.

Використання перемикачів розглянемо, виконавши вправу.

ВПРАВА 56.1



Завдання. Створити програму, в якій додати перемикачі з назвами трьох найбільших міст України. Після вибору міста у верхньому написі має з'являтися назва міста, а у нижньому — інформація про рік його заснування, кількість населення та площу. Інформацію про міста можна знайти в мережі Інтернет або завантажити з файла до даного уроку.

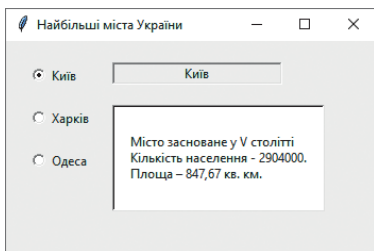


Рис. 56.2

```
r3=Radiobutton(text="Одеса")  
r3.place(x=20, y=100)
```

1. Створіть графічне вікно програми, що містить заголовок **Найбільші міста України** та має розміри 400x200.

2. Додайте команди розташування у вікні перемикачів:

```
#Перемикачі  
r1=Radiobutton(text="Київ")  
r1.place(x=20, y=20)  
r2=Radiobutton(text="Харків")  
r2.place(x=20, y=60)
```

Як видно з програмного коду, створення кожного перемикача складається з двох команд. У першій команді змінній присвоюємо конструктор **Radiobutton** із властивістю **text** та відповідним значенням тексту. У другій команді, як і для створення інших елементів керування, — метод розташування перемикача в графічному вікні.

3. Додайте команду `mainloop()`.

4. Запустіть програму. В результаті у вікні мають з'явитися перемикачі, які знаходяться у положенні **увімкнено**. Спробуйте змінити стан перемикача.

5. Аби перемикачі реагували на дію користувача, додайте команди, виділені чорним:



```
#Перемикачі
var=IntVar()
r1=Radiobutton(text="Київ",variable=var,value=1)
r1.place(x=20, y=20)
r2=Radiobutton(text="Харків",variable=var,value=2)
r2.place(x=20, y=60)
r3=Radiobutton(text="Одеса",variable=var,value=3)
r3.place(x=20, y=100)
```

У даному коді перша команда створює цілочисельну змінну, з якою будуть пов'язані перемикачі. У ній функція **IntVar()** визначає, що змінна буде цілого типу. Властивість **variable** пов'язує перемикач зі створеною змінною **var**. А властивість **value** встановлює значення змінної **var** у залежності від вибраного користувачем перемикача. Отже, для нашого випадку це буде значення 1, 2 або 3.

6. Запустіть програму ще раз. Тепер усі перемикачі мають бути вимкнені. Оберіть послідовно кожний із них.
7. Додайте до програми команди створення написів, відповідно до зразка (рис. 56.2). В обох написах створіть рельєф з допомогою властивості `relief=SUNKEN`. Для другого напису додайте білий колір фону та вирівнювання по лівому краю.

```
#Написи
Lb11=Label(relief=SUNKEN)
Lb11.place(x=100, y=20,width=160)
Lb12=Label(relief=SUNKEN, bg="white",justify=LEFT)
Lb12.place(x=100, y=60,width=200,height=100)
```

Створимо обробник події натискання на перемикач. Оскільки усі перемикачі прив'язані до однієї змінної, фактично вони утворюють групу перемикачів. Отже, для групи перемикачів має бути створена одна функція, що є обробником події натискання на перемикач із групи.

8. Для обробника події введіть такий код:

```
#вибір перемикача
def btn_click():
    if var.get()==1:
        Lb11['text']="Київ"
        Lb12['text']="Місто засноване у V столітті\n
        Кількість населення – 2904000.\nПлоща – 847,67 кв. км."
    if var.get()==2:
        Lb11['text']="Харків"
        Lb12['text']="Місто засноване у 1654 р.\n
        Кількість населення – 1444000.\nПлоща – 350 кв. км."
```



```
if var.get()==3:  
    Lb11['text']="Одеса"  
    Lb12['text']="Місто засноване у 1415 р.\n  
    Кількість населення – 1012000.\nПлоща – 162642 кв. км."
```

Як бачимо, в даному коді є три умови, що перевіряють значення, одержане змінною **var** з використанням методу **get**. Якщо змінна дорівнює **1**, виводиться інформація про Київ, якщо **2** — про Харків, якщо **3** — про Одесу. Зверніть увагу, що тут коди розташовані у два рядки, оскільки в один рядок вони не поміщаються. В редакторі коду ці команди слід розташовувати в одному рядку.

9. Для виконання обробника події додайте команди його виклику `command=btn_click` для першого перемикача.

```
r1=Radiobutton(text="Київ",variable=var, value=1,command=btn_click)
```



Дізнайтеся більше про використання у перемикачів властивості `indicator`

10. Аналогічно додайте цю команду і для інших перемикачів.

11. Запустіть програму та перевірте правильність її виконання.

Отже, для створення групи із 3-х перемикачів можна скористатися такою загальною конструкцією:

```
змінна=IntVar()  
змінна1=Radiobutton(text, variable=змінна, value=1)  
змінна1.pack()  
змінна2=Radiobutton(text, variable=змінна, value=2)  
змінна2.pack()  
змінна3=Radiobutton(text, variable=змінна, value=3)  
змінна3.pack()
```

Спочатку слід оголосити цілочисельну змінну, з якою будуть пов'язані усі перемикачі, далі створити перемикач з допомогою конструктора **Radiobutton**, у якому вказати властивість тексту **text**, властивість **variable**, що пов'язує перемикач зі змінною, та властивість **value**, що визначає значення змінної у разі вибору даного перемикача користувачем. У наступному рядку — метод розташування перемикача у графічному вікні. Аналогічно створюються й інші перемикачі групи.

56.2.

ВИКОРИСТАННЯ ПРАПОРЦІВ

Для взаємодії з прикладною програмою, крім розглянутих раніше перемикачів, використовують прапорці. Вони стануть у нагоді, коли користувачу потрібно надати можливість обирати одне або кілька значень із набору запропонованих. Прапорці, як і перемикачі, мають два стани:



увімкнено та вимкнено (рис. 56.3). Для створення прапорця використовують команду виду:

```
змінна=Checkbutton(властивості)
```

Для виведення прапорця у графічному вікні, як і для інших елементів керування, використовують методи **pack** або **place**. Використання прапорців розглянемо, виконавши вправу.

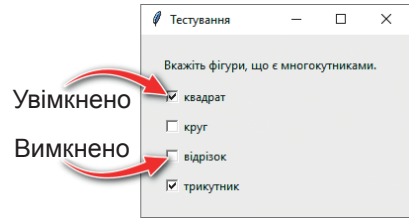


Рис. 56.3

ВПРАВА 56.2



Завдання. Створити програму формування меню, в якому користувач може обрати бажані страви, клацаючи прапорці біля їхніх назв (рис. 56.4). Після натискання на кнопку **Очистити** усі прапорці стають порожніми (вимкненими). Після натискання кнопки **Сформувати замовлення** з'являється вікно повідомлення з переліком обраних страв, і прапорці стають неактивними. При натисканні на кнопку **Редагувати замовлення** прапорці знову стають активними.

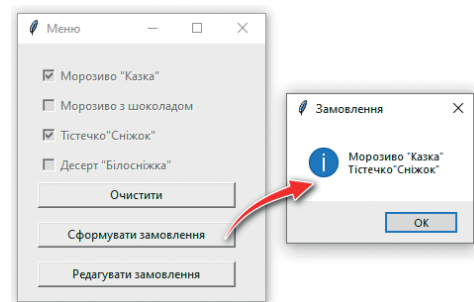


Рис. 56.4

Створення графічного вікна

1. Уведіть команди створення графічного вікна, що має заголовок **Меню** та розміри 250x260.

Створення прапорців

2. Додайте команди створення прапорців.

```
#Прапорці
d1=Checkbutton(text='Морозиво "Казка"')
d1.place(x=20, y=20)
d2=Checkbutton(text="Морозиво з шоколадом")
d2.place(x=20, y=50)
d3=Checkbutton(text='Тістечко"Сніжок"')
d3.place(x=20, y=80)
d4=Checkbutton(text='Десерт "Білосніжка"',variable=d4_var)
d4.place(x=20, y=110)
```

З програмного коду видно, що створення кожного прапорця складається з двох команд. У першій команді змінній присвоюємо конструктор **Checkbutton** із властивістю **text**. Друга команда містить метод розташування прапорця у вікні.



3. Запустіть програму на виконання. В результаті у вікні буде відображено прапорці, в яких можна встановлювати бажане положення.

Створення кнопок

4. Уведіть команди розташування у вікні кнопок **Очистити**, **Сформувати замовлення** та **Редагувати замовлення** відповідно до зразка (рис. 56.4). Надайте їм імена **clear**, **create** та **edit** відповідно.

Створення обробника кнопки **Очистити**

5. Створіть обробник події кнопки **Очистити**:

```
def clear_click():  
    d1.deselect()  
    d2.deselect()  
    d3.deselect()  
    d4.deselect()
```

Для очищення прапорця використовують метод **deselect**. В даному коді цей метод застосовано до кожного прапорця.

6. У конструкторі кнопки **Очистити** додайте команду виклику обробника події.
7. Запустіть програму. Оберіть деякі прапорці та натисніть кнопку **Очистити**. В результаті усі прапорці буде прибрано.

Створення обробника події кнопки **Сформувати замовлення**

В результаті натискання кнопки **Сформувати замовлення** має бути виведено повідомлення з переліком страв, а усі прапорці мають стати неактивними.

Аби програма могла запам'ятати положення прапорця, встановлене користувачем, прапорці, як і перемикачі, пов'язують зі змінними. Проте на відміну від перемикачів, де уся група була пов'язана з однією змінною, кожний прапорець пов'язують зі своєю змінною, яка може приймати значення 1, якщо прапорець увімкнено, і значення 0, при вимкненому прапорцеві.

8. У тілі програми створіть змінні для прапорців. Для цього введіть такий код:

```
d1_var=IntVar()  
d2_var=IntVar()  
d3_var=IntVar()  
d4_var=IntVar()
```

9. Додайте до кодів прапорців властивості, що пов'яжуть їх зі створеними змінними. Для цього, як і для перемикачів, використовують властивість **variable**.

```
d1=Checkbutton(text='Морозиво "Казка"',variable=d1_var)  
d2=Checkbutton(text="Морозиво з шоколадом",variable=d2_var)  
d3=Checkbutton(text='Тістечко"Сніжок"',variable=d3_var)  
d4=Checkbutton(text='Десерт "Білосніжка"',variable=d4_var)
```



10. Уведіть код обробника події кнопки **Сформувати замовлення**.

```
def create_click():
    txt=""
    if d1_var.get()==1:
        txt=txt+'Морозиво "Казка"\n'
    if d2_var.get()==1:
        txt=txt+'Морозиво з шоколадом\n'
    if d3_var.get()==1:
        txt=txt+'Тістечко"Сніжок"\n'
    if d4_var.get()==1:
        txt=txt+'Десерт "Білосніжка"\n'
    result=showinfo("Замовлення",txt)
```

Тут у першому рядку створено порожню змінну `txt`, у яку будуть дописуватися обрані з допомогою прапорців страви. У першому умовному операторі за допомогою методу `get` перевіряється значення змінної, що пов'язана з першим прапорцем. Якщо вона дорівнює 1 (тобто прапорець є увімкненим), то до змінної `txt` дописується назва першої страви і з допомогою символів `\n` здійснюється перехід на наступний рядок. Аналогічно сформовано умовні оператори і для інших прапорців.

В останньому рядку розташовано команду виведення інформаційного вікна повідомлення, де `showinfo` — тип інформаційного вікна, **Замовлення** — заголовок, `txt` — змінна, в якій сформовано повідомлення, яке необхідно вивести.

11. Для виведення вікна повідомлення імпортуйте вбудований до `tkinter` модуль `messagebox`:

```
from tkinter.messagebox import*
```

12. Для кнопки **Сформувати замовлення** додайте команду виклику обробника події.

13. Запустіть програму на виконання, оберіть страви. Натисніть кнопку **Сформувати замовлення**. В результаті має з'явитися вікно повідомлення з обраними стравами.

Крім виведення вікна повідомлення, до даного обробника події необхідно додати команди, аби прапорці ставали неактивними. За значення активності у прапорців відповідає властивість `state`. За умовчанням це значення дорівнює `NORMAL`, тобто елемент є активним; якщо значення дорівнює `DISABLED` — елемент буде неактивним. Оскільки в нашому випадку прапорці мають стати неактивними, то для цієї властивості необхідно надати значення `DISABLED`.



14. Додайте до обробника події такий код:

```
d1['state']=DISABLED
d2['state']=DISABLED
d3['state']=DISABLED
d4['state']=DISABLED
```

Подумайте, куди необхідно вставити цей код, аби прапорці ставали неактивними одночасно з появою вікна повідомлення.

15. Запустіть програму, оберіть деякі страви та натисніть кнопку **Сформувати замовлення**. В результаті має з'явитися вікно повідомлення з обраними стравами і прапорці мають стати неактивними.

Створення обробника події кнопки **Редагувати замовлення**

При натисненні кнопки **Редагувати замовлення** усі прапорці знову мають стати активними. Неважко здогадатися, що для цього необхідно повернути в початкове положення властивість **state** для прапорців.

16. Уведіть код обробника події кнопки **Редагувати замовлення**:

```
def edit_click():
    d1['state']=NORMAL
    d2['state']=NORMAL
    d3['state']=NORMAL
    d4['state']=NORMAL
```

17. Для кнопки **Редагувати замовлення** додайте команду виклику обробника події.

18. Запустіть програму та перевірте правильність роботи усіх кнопок.

Отже, для створення групи із 3-х прапорців можна скористатися такою загальною конструкцією:

```
змінна1=IntVar()
змінна2=IntVar()
змінна3=IntVar()
змінна4=Checkbutton(text, variable=змінна1)
змінна4.pack()
змінна5=Checkbutton(text, variable=змінна2)
змінна5.pack()
змінна6=Checkbutton(text, variable=змінна3)
змінна6.pack()
```



Контрольні запитання та завдання

1. Для чого використовують перемикачі та прапорці?
2. Які конструктори використовують для створення перемикачів та прапорців?
3. Як створити групу перемикачів; прапорців?
4. Які властивості мають перемикачі та прапорці?



Питання для роздумів

- 1*. Які відмінності у використанні перемикачів та прапорців?
- 2*. Наведіть приклади використання перемикачів (прапорців).



Завдання для досліджень

- 1*. Наведіть приклади використання перемикачів (прапорців) у текстовому редакторі Word.
- 2*. Дослідіть призначення властивості `indicator` для прапорців.



Пройдіть
тест

§57.

Використання списків

Якщо користувачеві потрібно надати можливість обирати один або кілька елементів з певного переліку, застосовують такий елемент інтерфейсу, як список. У вигляді списків можна подавати назви річок, прізвища, розміри шрифтів, оцінки учнів тощо. Отже, *список* — це елемент керування, що складається з текстових рядків, які може обирати користувач. Списки створюють за допомогою конструктора `Listbox`.

Загальний вигляд коду для створення списку буде таким:

```
змінна=Listbox(властивості)
```

Кількість елементів, що буде відображено у списку, встановлюється за допомогою властивості `height`. Так, для створення списку, наведеного на рис. 57.1, `height=7`.

Для виведення списку у графічному вікні, як і для інших елементів керування, використовують методи `pack` або `place`.

Дії, які можна виконувати зі списками, розглянемо, виконавши вправу.

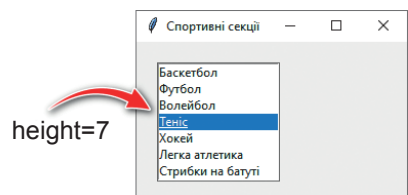


Рис. 57.1

ВПРАВА 57



Завдання. Створити програму, що містить два списки та чотири кнопки відповідно до зразка (рис. 57.2). При натисненні на верхню кнопку всі елементи з першого списку переміщуються до другого. При натисненні на нижню кнопку всі елементи із другого списку повертаються до

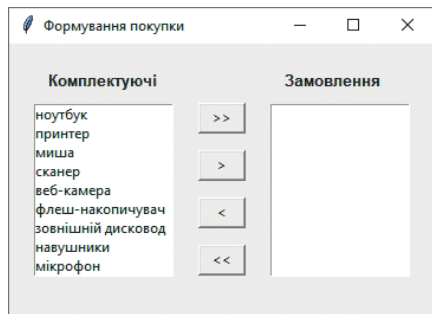


Рис. 57.2

першого. При натисненні на другу кнопку виділений елемент із першого списку переміщується до другого списку, а при натисненні на третю кнопку виділений елемент другого списку переміщується до першого списку.

Створення графічного вікна та написів

1. Уведіть команди створення графічного вікна, що має заголовок **Формування покупки** та розміри 360x230.
2. Додайте команди створення написів **Комплектуючі** та **Замовлення**, що мають бути розташовані над списками.

Створення списків

3. Уведіть команди створення списків:

```
#списки
Lbox1=Listbox(height=9)
Lbox1.place(x=20,y=50, width=120)
Lbox2=Listbox(height=9)
Lbox2.place(x=220,y=50, width=120)
```

Як видно з програмного коду, створення кожного списку складається з двох команд. У першій команді змінній присвоюємо конструктор **Listbox** з властивістю `height=9`. У другій команді, як і для створення інших елементів керування, — метод розташування списку в графічному вікні.

4. Додайте команду `mainloop()`.
5. Запустіть програму та перевірте правильність створення вікна та елементів керування.

Додавання елементів до списку

Для додавання елементів до списку використовують метод `insert(індекс, елемент)`.

Тут **індекс** — номер позиції, на яку необхідно вставити **елемент**. Зверніть увагу на те, що нумерація елементів списку починається з нуля. Якщо елемент необхідно додати в кінець списку, то замість параметра **індекс** потрібно записати значення **END**.

6. Уведіть команду додавання до створеного списку елемента «ноутбук»:
`Lbox1.insert(END, "ноутбук")`
7. Інші команди додавання елементів списку створіть аналогічно.



- Запустіть програму та переконайтеся, що елементи з назвами товарів розташовано у списку.

Створення кнопок та обробників події першої та четвертої кнопок

- Додайте до інтерфейсу програми кнопки >>, >, <, << з іменами **Btn1**, **Btn2**, **Btn3** та **Btn4** відповідно.

Розглянемо створення обробника події першої кнопки. Натиснення на неї має переміщувати всі елементи з першого списку до другого.

Алгоритм дії обробника може бути таким. Беремо по черзі кожен елемент списку 1 і вставляємо його до списку 2. Після закінчення цього процесу очищуємо список 1. Отже, з даного алгоритму зрозуміло, що перенесення елементів зі списку 1 до списку 2 має відбуватися у циклі. Кількість ітерацій даного списку буде дорівнювати кількості елементів, що має список на момент натискання кнопки.

Для визначення розмірності списку використовують метод **size**. Отже, розмір першого списку буде знайдено з допомогою коду `Lbox1.size()`.

- Уведіть код обробника першої кнопки, що здійснює копіювання елементів з першого списку до другого.

```
def btn1_click():
    #цикл від першого до останнього елемента списку
    for i in range(Lbox1.size()):
        #змінній надається значення елемента списку під номером i
        item=Lbox1.get(i)
        #значення item розміщується в кінці списку Lbox2
        Lbox2.insert(END,item)
```

Також після копіювання елементів зі списку 1 до списку 2 ми маємо очистити перший список. Для очищення списку використовують метод **delete**. В нашому випадку — це команда `Lbox1.delete(0,END)`, що означає вилучення усіх елементів від першого до останнього.

- Додайте до обробника події першої кнопки команду очищення списку за межами тіла циклу.
- Додайте команду виклику обробника події у коді першої кнопки.
- Запустіть програму та перевірте, як буде працювати перша кнопка.
- Аналогічно створіть команди обробника 4-ї кнопки та перевірте її роботу після запуску програми.

Обробники події другої та третьої кнопок

Після виділення елемента зі списку 1 та натиснення на другу кнопку цей елемент має переміщуватися до списку 2 знизу.

Для визначення номера виділеного елемента використовується метод **curselection**.



Так, для прикладу, наведеного на рис. 57.1, в результаті виконання команди `n=Lbox.curselection()` змінна `n` набуде значення 3.

15. Уведіть обробник для другої кнопки:

```
def Btn2_click():
    n_item=Lbox1.curselection() #присвоєння номера виділеного списку
    item=Lbox1.get(n_item) #визначення значення елемента за номером
    Lbox2.insert(END,item) #додавання елемента в кінець 2-го списку
    Lbox1.delete(n_item) #вилучення елемента з 1-го списку
```

16. У кодї створення другої кнопки додайте команду виклику обробника події.

17. Запустіть програму на виконання. Оберіть елемент списку, натисніть другу кнопку, в результаті цей елемент має з'явитись у другому списку, а з першого списку він буде вилучений. Виконайте ці ж дії і з іншими елементами першого списку.

18. Аналогічно створіть обробник події для третьої кнопки.

19. Запустіть програму та перевірте роботу кнопок.

Виконуючи завдання даної вправи, ми познайомилися з методами роботи зі списками. Усі вони наведені у таблиці 57.1.



Дізнайтеся більше про додавання до списків смуги прокручування

Таблиця 57.1. Методи роботи зі списками

| Метод | Опис |
|--------------------------------------|--|
| <code>insert(індекс, елемент)</code> | вставлення елемента до вказаної позиції |
| <code>size()</code> | визначення кількості елементів списку |
| <code>curselection()</code> | визначення номера виділеного елемента |
| <code>delete(початок, кінець)</code> | видалення елементів зі списку |
| <code>get(індекс)</code> | визначення значення елемента за його номером |



Контрольні запитання та завдання

1. Для чого використовують списки?
2. З допомогою якого конструктора створюються списки?
3. Як додати елементи до списку?
4. Які методи використовують для роботи зі списками? Для чого вони використовуються?



Питання для роздумів

- 1*. Для чого у списках використовується смуга прокручування?
- 2*. Наведіть приклади використання різних видів списків.



Завдання для досліджень

- 1*. Наведіть приклади використання списків у табличному процесорі MS Excel.
- 2*. Відшукайте інформацію про те, як до графічного вікна додати поле зі списком.



Пройдіть
тест

§58.

Додавання зображень

Для надання інтерфейсу програми привабливості до неї можна додавати зображення із зовнішніх файлів.

58.1.

СТВОРЕННЯ ПОЛОТНА

Для додавання файлу зображення у графічному вікні необхідно створити полотно для малювання **Canvas**. Загальна структура коду для створення полотна має вигляд:

```
змінна=Canvas(властивості)
```

В якості властивостей указують ім'я змінної, яка приймає значення графічного вікна, можна також вказати ширину та висоту полотна. Для розташування полотна у графічному вікні, як і для інших елементів керування, використовують методи **pack** або **place**.

Розглянемо приклад коду створення полотна:

```
from tkinter import*
Window=Tk()
Window.geometry("140x140")
canvas=Canvas(Window)
canvas.place(x=20, y=20, width=100, height=80)
```

З допомогою цього коду буде створено графічне вікно, в якому розташовується полотно **Canvas**, розміщене відносно графічного вікна, починаючи з координат (20, 20) шириною 100 пікселів та висотою 80 пікселів (рис. 58.1).

Звичайно, при запуску даної програми на виконання ми не побачимо ніяких змін у графічному вікні. Зміни будуть видні тоді, коли в цьому полотні буде розміщено деяке зображення.

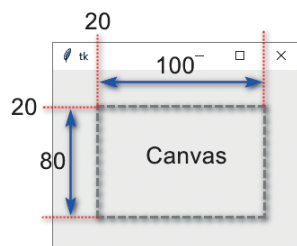


Рис. 58.1

58.2.

ДОДАВАННЯ ЗОБРАЖЕННЯ

Для завантаження до полотна зображення файла використовують конструктор **PhotoImage**. Загальний синтаксис коду буде таким:

```
змінна=PhotoImage(властивості)
```

Однією з основних властивостей даного конструктора є **file**, що містить адресу файла, який необхідно завантажити. Якщо файл міститься у тій же папці, що і файл програми, вказують лише ім'я файла, якщо в іншій папці — слід указати шлях до цієї папки. Наприклад,

```
img=PhotoImage(file='flower.png')
```

Але для розташування цього файла у графічному вікні необхідно є ще функція `canvas.create_image`. Наприклад,

```
canvas.create_image(0,0,anchor=NW,image=img)
```

Тут **canvas.create_image** — функція виведення зображення;

0, 0 — координати зображення по осі *x* та *y* відносно полотна Canvas;

image=img — вказує ім'я змінної, до якої завантажено зображення;

anchor — параметр, що визначає, до якої точки полотна прив'язується зображення. Значення **NW** — скорочено North West, що в перекладі означає «північний захід», вказує на прив'язку зображення до верхнього лівого кута полотна.

Звичайно, атрибут **anchor** може набувати й інших значень. На рис. 58.2 представлено його можливі значення та відповідні точки прив'язки.

За умовчанням **anchor** має значення **CENTER**, отже, якщо цей атрибут у команді пропущений — це означає прив'язку до центра.

Більш детально додавання зображень у графічному вікні розглянемо, виконавши вправу.

| | | |
|----|--------|----|
| NW | N | NE |
| W | CENTER | E |
| SW | S | SE |

Рис. 58.2

ВПРАВА 58



Завдання. Створити програму, в якій при завантаженні відображаються назви тварин. При наведенні на назву тварини з'являється її зображення. При клацанні на назву тварини — виводиться вікно повідомлення з інформацією про цю тварину. Також дане вікно програми має містити власну піктограму, розміщену в рядку заголовка (рис. 58.3).



Створення інтерфейсу програми

1. Створіть графічне вікно із заголовком **Зоопарк** та з розмірами 350x240.
2. Додайте команди створення написів, що мають містити відповідні назви тварин, розмір шрифту — 16 пт. та зелений колір фону (рис. 58.3).

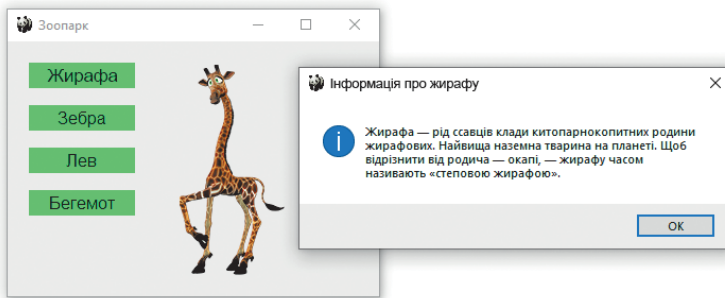


Рис. 58.3

3. Додайте команду `mainloop()`.

Додавання зображень

4. Розташуйте необхідні для програми файли зображень у папці з програмою (це мають бути зображення панди, жирафи, зебри, лева та бегемота). Їх ви можете завантажити з папки до даного уроку.
5. Додайте команду: `Window.iconbitmap('panda.ico')`. Тут **iconbitmap** — метод установлення іконки у рядку заголовка, **panda.ico** — ім'я файла зображення.
6. Додайте команди створення графічного полотна.
`canvas=Canvas(Window)`
`canvas.place(x=160,y=20,width=190,height=220)`
7. Створіть обробник події, що має відбуватися в разі наведення на напис **L1**, що має значення **жирафа**:

```
def L1_mot(event):
    global img
    img=PhotoImage(file='giraffe.png')
    canvas.create_image(0,0,anchor=NW,image=img)
```

Перший рядок — заголовок функції. Оскільки подія відбувається не при натисненні на деяку кнопку, в якості аргумента функції використовується службове слово **event** — подія. У другому рядку оголошено в якості глобальної змінної **img**, що буде мати значення зображення. У третьому рядку створено зображення з допомогою конструктора **PhotoImage**. В якості аргументу вказано ім'я файла 'giraffe.png'. В останньому рядку вказано місце розташування зображення на полотні.



8. Для виклику обробника події наведення на напис **L1** додайте наступну команду:

```
L1.bind('<Motion>',L1_mot)
```

Тут **L1** — напис, на який необхідно навести курсор миші; **Motion** — подія наведення курсору миші; **L1_mot** — ім'я обробника події.

9. Запустіть програму. Наведіть курсор на напис **Жирафа**. В результаті має з'явитися зображення жирафи.
10. Самостійно додайте обробники подій для наведення курсору на інші написи. А також додайте команди виклику цих обробників подій.
11. Уведіть обробник події клацання на написі, що має призвести до появи вікна повідомлення з інформацією про жирафу:

```
def L1_click(event):  
    info=showinfo("Інформація про жирафу",  
                  'Жирафа – рід ссавців класу китопарнокопитних  
                  родини жирафових. Найвища наземна тварина на  
                  планеті. Щоб відрізнити від родича – окапі, –  
                  жирафу часом називають "степовою жирафою".')
```

12. Для виклику даного обробника події додайте команду:

```
L1.bind('<Button-1>',L1_click)
```

13. Запустіть програму. Клацніть на написі. В результаті має з'явитися вікно повідомлення з інформацією про жирафу.
14. Самостійно додайте обробники подій клацання курсору на інші написи. А також додайте команди виклику цих обробників подій (інформацію про інших тварин відшукайте в мережі Інтернет).



Контрольні запитання та завдання

1. Як змінити піктограму заголовка вікна?
2. Що необхідно для розташування зображення у графічному вікні?
3. Що таке полотно?
4. Як завантажити зображення до полотна?
5. Для чого використовується параметр **anchor**?
6. Де мають розташовуватися файли зображень, що відображаються у графічному вікні?



Питання для роздумів

- 1*. Чи можна при додаванні зображень не вказувати параметр `anchor`?



Завдання для досліджень

- 1*. Дослідіть, які типи файлів зображень можна додавати до графічного вікна.
- 2*. Відшукайте інформацію про те, як до графічного вікна додати графічні примітиви (відрізки, кола, прямокутники тощо).



Пройдіть
тест



Висновки

Мова програмування — формалізована мова для запису алгоритмів, призначених для виконання за допомогою комп'ютера.

Транслятор — це програмний засіб, що здійснює перетворення програми з однієї мови програмування в програму на іншій мові програмування.

Величинами називають об'єкти, над якими виконуються певні операції. Величини характеризуються іменем, типом і значенням.

Існують такі найбільш уживані типи величин: `int` — цілі числа; `float` — дійсні числа; `str` — рядки.

Величини, які під час виконання програми можуть набувати різних значень, називають **змінними**.

Надання змінній деякого значення називається **командою присвоєння**.

У мові Python для використання деяких функцій необхідно імпортувати їхні модулі. Модуль `math` містить математичні функції, модуль `random` містить функції створення випадкових чисел, модуль `tkinter` використовують для створення графічних вікон.

Тестування програми — це перевірка відповідності між реальною і очікуваною поведінками програми, що здійснюється на кінцевому наборі тестів, обраних певним чином.

Основною одиницею в об'єктно-орієнтованому програмуванні є **програмний об'єкт**, який поєднує *властивості* та *методи*.

Метод описує дії, які можна виконувати з об'єктом.

Елементи керування — це об'єкти, за допомогою яких створюється графічний інтерфейс.



Розгадайте електронний кросворд

ЗМІСТ

РОЗДІЛ 1. Кодування даних та апаратне забезпечення 5

| | | |
|------|---|----|
| §1. | Роль і місце кодування в інформаційних процесах | 5 |
| §2. | Системи числення | 8 |
| §3. | Кодування чисел | 11 |
| §4. | Кодування символів..... | 14 |
| §5. | Вимірювання інформації..... | 19 |
| §6. | Кодування графічних даних..... | 24 |
| §7. | Апаратне забезпечення..... | 32 |
| §8. | Пристрої введення..... | 37 |
| §9. | Пристрої виведення та мультимедійні пристрої..... | 41 |
| §10. | Технічні характеристики основних складових персонального комп'ютера | 44 |
| §11. | Історія обчислювальних та комп'ютерних пристроїв | 49 |
| §12. | Види сучасних комп'ютерів та їхнє застосування | 58 |

РОЗДІЛ 2. Опрацювання текстових даних.....63

| | | |
|------|--|----|
| §13. | Пошук та заміна фрагментів тексту..... | 63 |
| §14. | Використання стилів..... | 67 |
| §15. | Розділи. Колонтитули | 72 |
| §16. | Структура документа | 77 |
| §17. | Гіперпосилання в текстових документах..... | 80 |
| §18. | Спільна робота з документом..... | 87 |
| §19. | Створення складного документа | 89 |
| §20. | Практична робота №1. Створення складного документа | 91 |

РОЗДІЛ 3. Створення та публікація вебресурсів93

| | | |
|------|--|-----|
| §21. | Автоматизовані засоби для створення та публікації вебресурсів..... | 93 |
| §22. | Мова гіпертекстової розмітки..... | 98 |
| §23. | Форматування символів | 105 |
| §24. | Списки..... | 108 |
| §25. | Додавання зображень на вебсторінці | 110 |
| §26. | Додавання гіперпосилань на вебсторінці | 114 |
| §27. | Розмітка таблиць засобами мови HTML..... | 116 |
| §28. | Правила ергономічного розміщення відомостей на вебсторінці..... | 122 |
| §29. | Практична робота №2. Створення власного сайту | 125 |

РОЗДІЛ 4. Опрацювання об'єктів мультимедіа 127

| | | |
|------|--|-----|
| §30. | Мультимедіа як ознака сучасного інформаційного світу..... | 127 |
| §31. | Кодування аудіо- та відеоданих. Формати аудіо- та відеофайлів | 130 |
| §32. | Програмне забезпечення для опрацювання об'єктів мультимедіа.
Засоби перетворення аудіо- й відеоформатів | 135 |
| §33. | Відеозахоплення | 138 |
| §34. | Розміщення відео- та аудіофайлів в Інтернеті | 144 |

РОЗДІЛ 5. Алгоритми та програми..... 151

| | | |
|------|---|-----|
| §35. | Поняття мови програмування | 151 |
| §36. | Знайомство з середовищем програмування | 154 |
| §37. | Типи даних у програмуванні. Присвоєння значень величинам | 158 |
| §38. | Використання арифметичних операцій | 162 |
| §39. | Розв'язування задач з арифметичними операціями | 165 |
| §40. | Стандартні математичні функції..... | 167 |
| §41. | Практична робота №3. Реалізація лінійних алгоритмів в середовищі
програмування | 170 |
| §42. | Логічні вирази та змінні й операції над ними..... | 171 |
| §43. | Умовні оператори..... | 175 |
| §44. | Вкладені умови. Множинне розгалуження..... | 180 |
| §45. | Налагодження та тестування програми..... | 184 |
| §46. | Задачі на знаходження найбільшого та найменшого серед кількох значень | 190 |
| §47. | Оператор циклу з лічильником..... | 194 |
| §48. | Використання циклу з параметром..... | 198 |
| §49. | Цикли з умовою | 202 |
| §50. | Вкладені цикли..... | 206 |
| §51. | Основні поняття об'єктно-орієнтованого програмування..... | 209 |
| §52. | Властивості та методи елементів керування | 214 |
| §53. | Обробники подій, пов'язані з елементами керування | 220 |
| §54. | Уведення даних..... | 226 |
| §55. | Використання вікон повідомлень | 232 |
| §56. | Використання перемикачів та прапорців..... | 237 |
| §57. | Використання списків..... | 245 |
| §58. | Додавання зображень..... | 249 |

Відомості про стан підручника

| № | Прізвище та ім'я учня/учениці | Навчальний рік | Стан підручника | |
|---|-------------------------------|----------------|-----------------|--------------|
| | | | на початку року | в кінці року |
| 1 | | | | |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |
| 5 | | | | |

Навчальне видання

**КАЗАНЦЕВА Ольга Павлівна
СТЕЦЕНКО Ірина Володимирівна**

ІНФОРМАТИКА

**Підручник для 8 класу
закладів загальної середньої освіти**

*Рекомендовано
Міністерством освіти і науки України*

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

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

Головний редактор *Богдан Будний*
Редактор *Володимир Дячун*
Обкладинка *Ростислава Крамара*
Комп'ютерна верстка *Андрія Кравчука*
Художній редактор *Ростислав Крамар*
Технічний редактор *Неля Домарецька*

Підписано до друку 28.04.2021. Формат 70×100/16. Папір офсетний.
Гарнітура CentSchbook Win95BT. Друк офсетний. Умовн. друк арк. 20,80.
Умовн. фарбо-відб. 83,20. Обл.-вид. арк. 12,48. Тираж 18 305 пр. Зам. 604/04.

Видавництво «Навчальна книга – Богдан»
Свідоцтво про внесення суб'єкта видавничої справи до Державного
реєстру видавців, виготівників і розповсюджувачів видавничої продукції
ДК № 4221 від 07.12.2011 р.

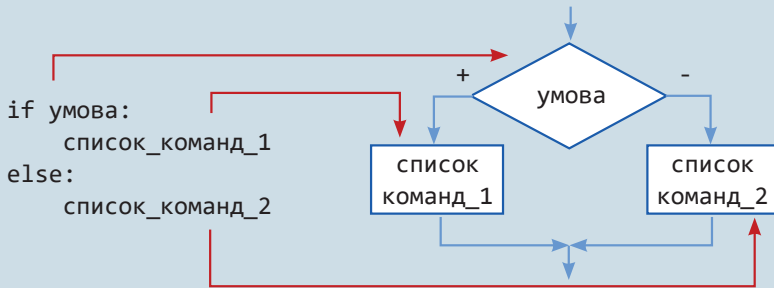
Навчальна книга – Богдан, просп. С. Бандери, 34а, м. Тернопіль, 46002

Надруковано на ПП «Юнісофт», вул. Морозова, 136, м. Харків, 61036
Свідоцтво серія ДК №3461 від 14.04.2009 р.

АЛГОРИТМИ ТА ПРОГРАМИ



ПРОГРАМНИЙ КОД ТА БЛОК-СХЕМА ПОВНОЇ ФОРМИ РОЗГАЛУЖЕННЯ



ЦИКЛИ З ЛІЧИЛЬНИКОМ ТА З УМОВОЮ

| Цикл з лічильником | Цикл з умовою |
|--|--|
| | |
| <pre>for лічильник in range(n1,n2,k): тіло циклу</pre> | <pre>while умова: тіло_циклу</pre> |

КОНСТРУКТОРИ ЕЛЕМЕНТІВ КЕРУВАННЯ

| | |
|-------------|---------------|
| Label | напис |
| Entry | текстове поле |
| Button | кнопка |
| Radiobutton | перемикач |
| Checkbutton | прапорець |
| Listbox | список |
| Canvas | полотно |
| PhotoImage | зображення |

МЕТОДИ МОДУЛЯ TKINTER

| | |
|----------|--|
| title | Заголовок графічного вікна |
| geometry | Розмір та положення графічного вікна |
| pack | Розташування об'єкта у вікні |
| place | Розташування об'єкта за вказаними координатами |
| get | Одержання значення з елемента керування |
| delete | Вилучення значення з елемента керування |
| insert | Додавання значення до елемента керування |

ВЛАСТИВОСТІ ЕЛЕМЕНТІВ КЕРУВАННЯ

| Властивість | Опис | Значення |
|-------------|---|--|
| font | шрифт тексту | назва, розмір, накреслення (bold, italic, underline) |
| bg / fg | колір фону / колір шрифту | константа або код |
| bd | товщина контуру | константа або код |
| justify | вирівнювання | LEFT, CENTER, RIGHT |
| padx | відступ від межі елемента до тексту зліва та справа | значення в пікселях |
| pady | відступ від межі елемента до тексту знизу та зверху | значення в пікселях |
| relief | тип межі | SUNKEN, RAISED, GROOVE, RIDGE |

ОБРОБНИК ПОДІЇ НАТИСКАННЯ КНОПКИ

Структура функції обробника події:

```
def ім'я_функції():  
    подія
```

Виклик обробника події:

```
command=ім'я_функції
```

ОБРОБНИК БУДЬ-ЯКОЇ ПОДІЇ

Структура функції обробника події:

```
def ім'я_функції(event):  
    подія
```

Виклик обробника події:

```
ім'я_об'єкту.bind(подія, ім'я_функції)
```