

ВИДАВНИЦТВО  
**РАНОК**



Інтернет-  
підтримка

7

# ІНФОРМАТИКА



УДК 004:37.016(075.3)

I-74

**Авторський колектив:**

Олена Бондаренко, Василь Ластовецький,  
Олександр Пилипчук, Євген Шестопалов

**Рекомендовано Міністерством освіти і науки України**  
(наказ Міністерства освіти і науки України від 25.03.2020 № 449)

Видано за рахунок державних коштів. Продаж заборонено

I-74 **Інформатика** : підруч. для 7 кл. закл. загал. серед. освіти / [О. О. Бондаренко, В. В. Ластовецький, О. П. Пилипчук, Є. А. Шестопалов]. — Харків : Вид-во «Ранок», 2020. — 160 с. : іл.

ISBN 978-617-09-6266-9

УДК 004:37.016(075.3)



Інтернет-підтримка

ISBN 978-617-09-6266-9

© Бондаренко О. О., Ластовецький В. В.,  
Пилипчук О. П., Шестопалов Є. А., 2020  
© ТОВ Видавництво «Ранок», 2020

# ДОРОГІ СЕМИКЛАСНИКИ ТА СЕМИКЛАСНИЦІ!

Ви тримаєте в руках підручник з інформатики, призначений саме для вас, учнів і учениць 7 класу. У 6 класі ви ознайомилися з особливостями побудови й опрацювання растрових і векторних зображень, навчилися створювати презентації з елементами керування показом, розробляти їх сценарії і добирати стильове оформлення слайдів.

Ви продовжували вивчення основ мови програмування Python, опанували складання та виконання алгоритмів із розгалуженнями та повтореннями.

У цьому навчальному році на вас теж очікує чимало цікавого та корисного, а пропонований підручник буде вашим надійним помічником. Пригадаймо, як із ним працювати.

Підручник складається з трьох розділів. *Розділ 1* присвячений вивченню основ об'єктно-орієнтованого програмування засобами середовища програмування Python, *розділ 2* — знайомству з поштовими службами інтернету, етикетом електронного листування, онлайн-вими перекладачами і хмарними сервісами, *розділ 3* — опануванню табличного процесора — програми для опрацювання даних, поданих у вигляді електронних таблиць.

На початку кожного розділу вміщено матеріали за рубрикою «*Повторюємо*». Вони допоможуть згадати відомості, які ви вивчали з цієї теми в попередніх класах і які будуть необхідні для засвоєння нового матеріалу.

Розділ складається з параграфів і практичних робіт за темами розділу. Кожен параграф містить *теоретичні відомості, приклади практичного застосування знань, питання для самоперевірки, вправи*.

*Питання для самоперевірки* допоможуть з'ясувати, чи достатньою мірою ви опанували вивчений матеріал, а також підготуватися до виконання вправ і практичних робіт.

*Вправи* складаються із завдань теоретичного і практичного спрямування. Вправи з позначкою ► містять набір окремих завдань, а вправи з позначкою ►► — завдання з описом покрокового виконання. Виконуючи їх, ви здобудете необхідні практичні навички роботи за комп'ютером.

*Практичні роботи* дають вам змогу розвинути дослідницькі навички. Щоб успішно виконати кожну з них, ви маєте повторити матеріал, вивчений на попередніх уроках, і застосувати навички роботи з певним програмним забезпеченням.

Оцінити здобуті знання й уміння вам допоможе *комп'ютерне тестування*, яке можна пройти в онлайн-режимі з автоматичною перевіркою результату.

*Бажаємо натхнення та успіхів!*

У тексті підручника використано такі позначення:



Запам'ятайте



Розгляньте приклад



Ознайомтеся з цікавою інформацією



Знайдіть відповідь в інтернеті



Виконайте завдання за комп'ютером



Виконайте завдання підвищеної складності



Виконайте завдання в парах



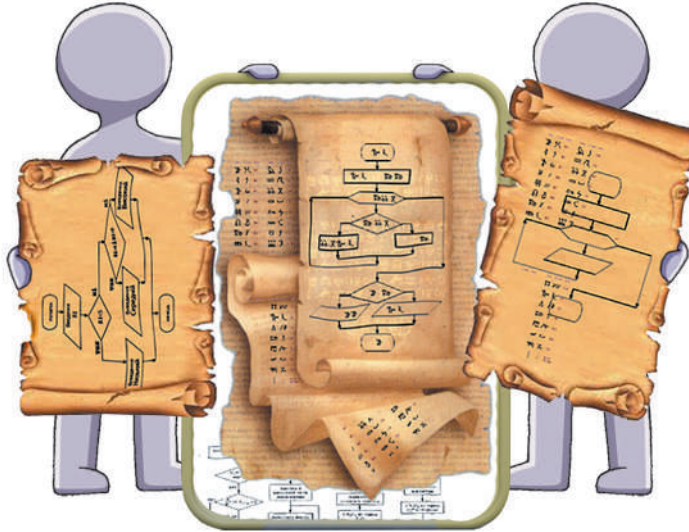
Дайте відповідь на питання для самоперевірки



Виконайте тестове завдання

# РОЗДІЛ 1

## АЛГОРИТМИ ТА ПРОГРАМИ



- § 1. Етапи розв'язування задачі за допомогою комп'ютера
- § 2. Величини. Алгоритми роботи з величинами
- § 3. Створення алгоритмів і програм із розгалуженнями та повтореннями
- § 4. Рядки як послідовності символів
- § 5. Опрацювання рядкових величин

Практична робота 1. Створення програм для опрацювання рядкових величин

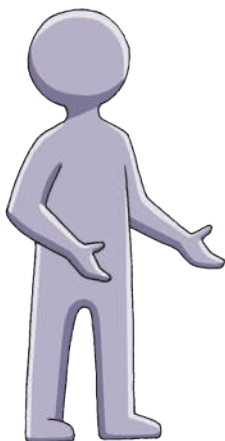
- § 6. Списки — змінювані послідовності
- § 7. Опрацювання елементів списку
- § 8. Функції та методи опрацювання списків
- § 9. Створення інтерфейсу користувача

Практична робота 2. Створення програм для опрацювання списків

- § 10. Графічні методи модуля tkinter
- § 11. Створення анімації
- § 12. Об'єктно-орієнтоване програмування

Практична робота 3. Моделювання руху об'єктів

## ПОВТОРЮЄМО



У попередніх класах ви ознайомилися з основами програмування мовою Python та прийомами роботи в середовищі програмування IDLE.

Ви знаєте, що таке *величини*, та вмієте складати прості алгоритми роботи з величинами *числових і рядкового типів*. Вам відомо, як реалізувати *базові алгоритмічні структури* мовою Python, ви вмієте створювати цікаві малюнки за допомогою черепашої графіки.

Ви ознайомилися з *принципами об'єктно-орієнтованого програмування* й знаєте, що в програмі описуються *об'єкти* та їхня взаємодія.

Ви навчилися створювати програми з *графічним інтерфейсом*, задавати властивості об'єктів графічного інтерфейсу та програмувати обробку подій для цих об'єктів. Ви навчилися розв'язувати складні задачі методом поділу на підзадачі, а також виконувати обчислення з використанням функцій користувача.

1. Які види вікон існують у середовищі IDLE?
2. Як відкрити вікно програми?
3. Які повідомлення виводяться у вікні консолі?
4. Який модуль містить набір класів компонентів для створення графічного інтерфейсу користувача?
5. Який алгоритм створення функцій у Python?



У цьому розділі ви продовжите вивчення мови програмування Python і навчитеся працювати з такими структурами даних, як рядки та списки. Ви опануєте графічні методи модуля tkinter і створите свої перші ігрові програми.



## § 1. Етапи розв'язування задачі за допомогою комп'ютера

За допомогою комп'ютера можна розв'язувати безліч різноманітних задач різної складності. Наприклад, підрахувати вартість спожитої електроенергії, перекласти текст на іншу мову, розрахувати орбіту супутника зв'язку тощо. Будь-яку задачу легше розв'язати, якщо існує детальний план. Як скласти план розв'язування задачі на комп'ютері?

### Етапи розв'язування задачі

Розглянемо таку ситуацію. Влітку учні й учениці сьомого класу плавали Дніпром на теплоході. Під час подорожі вони визначили час руху вниз по Дніпру, час повернення назад, а також дізналися значення середньої швидкості течії на цій ділянці. Потім діти зацікавилися, з якою швидкістю йшов би теплохід у стоячій воді, і для пошуку відповіді вирішили скласти комп'ютерну програму.

Розв'язання задачі за допомогою комп'ютера складається з певних етапів (рис. 1.1). Проаналізуємо їх докладніше.

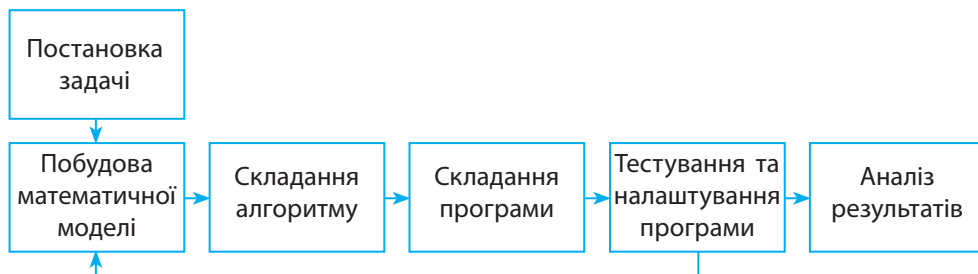


Рис. 1.1

#### ► I етап. Постановка задачі

На цьому етапі слід проаналізувати ситуацію, відкинути несуттєву для розв'язання інформацію, виділити початкові дані та очікувані результати.

Сформулюємо задачу: «Теплохід пройшов  $a$  годин за течією і  $b$  годин проти течії та повернувся в початкову точку. Знайти швидкість теплохода в стоячій воді, якщо швидкість течії становить  $v$  км/год».

## ► II етап. Побудова математичної моделі

На цьому етапі потрібно розгорнути умову задачі замінити математичною моделлю описаних об'єктів або процесів.



**Математична модель** — це система математичних співвідношень між величинами, необхідними для розв'язування задачі.

Для побудови математичної моделі потрібно з'ясувати, до якої предметної галузі належать об'єкти, згадані в умові задачі; визначити суттєві для розв'язуваної задачі властивості об'єктів; позначити змінні для зберігання значень величин; з'ясувати, як пов'язані результати із вхідними даними; записати рівняння та/або нерівності, що забезпечують розв'язання задачі.

**!** Помилки в математичній моделі — одна з основних причин неправильного розв'язування задачі.

Математичну модель зручно подавати в такій формі:

*Що дано?* <Перелік початкових даних>  
*Що треба знайти?* <Перелік потрібних результатів>  
*Який зв'язок між величинами?* <Математичні співвідношення, що зв'язують вхідні дані та результати>  
*Які є обмеження на дані?* <Умови допустимості початкових даних>

Запишемо математичну модель задачі:

*Що дано?*  $a$  — час руху за течією (год);  
 $b$  — час руху проти течії (год);  
 $v$  — швидкість течії (км/год).  
*Що треба знайти?*  $x$  — власна швидкість теплохода (км/год).  
*Який зв'язок між величинами?*  $(x+v) \cdot a = (x-v) \cdot b$ , звідки  $x = v \cdot \frac{b+a}{b-a}$ .  
*Які є обмеження на дані?*  $0 < a < b$ .

## ► III етап. Складання алгоритму

На цьому етапі потрібно скласти алгоритм розв'язування задачі.

**!** Від якості алгоритму залежать правильність результатів, ефективність використання часу та оперативної пам'яті комп'ютера.



Блок-схему алгоритму наведено на рис. 1.2. Алгоритм розв'язування задачі є лінійним.

#### ► IV етап. Складання програми

Реалізуємо алгоритм мовою Python:

```
a = float(input('a = ?'))
```

```
b = float(input('b = ?'))
```

```
v = float(input('v = ?'))
```

```
x = v*(b+a)/(b-a)
```

```
print('x = ', x)
```

#### ► V етап. Тестування та налаштування програми

На цьому етапі програму потрібно перевірити за допомогою тестів і виправити виявлені помилки.

Тест — це набір спеціально дібраних вхідних даних і відповідних їм результатів. Тестування полягає в порівнянні очікуваних результатів з результатами, отриманими під час виконання програми. Причиною розбіжності результатів роботи програми з тестовими може бути, наприклад, недолік математичної моделі, помилка в алгоритмі та/або в програмі. Після тестування доводиться уточнювати та корегувати модель і перевіряти алгоритм.

Випробуємо програму на такому тесті:  $a = 4.0$ ,  $b = 4.48$ ,  $v = 1.5$ ,  $x = 26.5$ . Маємо  $x = 26.499999999999998 \approx 26.5$ , результат збігається з наведеним.

#### ► VI етап. Аналіз результатів

На цьому етапі слід запустити програму з потрібними вхідними даними і проаналізувати отримані результати.

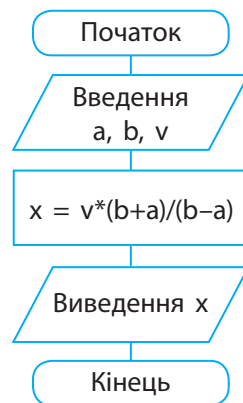


Рис. 1.2

### Питання для самоперевірки



1. Опишіть етапи розв'язування задачі за допомогою комп'ютера.
2. У чому полягає постановка задачі?
3. Що таке математична модель задачі?



4. Побудуйте математичну модель задачі.

*Задача.* У магазин привезли  $P$  кг яблук, а груш — у  $n$  разів більше. Скільки всього фруктів привезли в магазин?



5. Побудуйте математичну модель задачі.

*Задача.* Від продажу перших 100 газет продавець заробляє  $a$  грн, а від продажу кожної наступної — по  $x$  к. Визначте, скільки отримує продавець, якщо продано  $y$  газет ( $y > 100$ ).

### Вправа 1



- Прямокутник, довжини сторін  $a$  і  $b$  якого задовольняють умові  $\frac{a}{b} = \frac{b}{a-b}$ , називають «золотим» (рис. 1.3). Визначте за відомими довжинами сторін, чи є прямокутник «золотим».

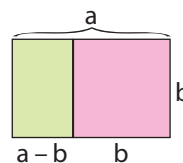


Рис. 1.3



- 1) Проаналізуйте постановку задачі та складіть математичну модель задачі.

*Що дано?*  $a, b$  — довжини сторін.

*Що треба знайти?* Відповідь Так або Ні.

*Який зв'язок між величинами?*

Якщо  $\frac{a}{b} = \frac{b}{a-b}$ , то вивести Так, інакше — вивести Ні.

*Які є обмеження на дані?*  $0 < b < a$ .

- 2) Накресліть блок-схему алгоритму розв'язування задачі.  
 3) У Python IDLE виберіть команду File → New File.  
 4) Запишіть оператори введення значень змінних  $a$  і  $b$ :  
 $a = \text{float}(\text{input}('a = ?'))$   
 $b = \text{float}(\text{input}('b = ?'))$
- 5) Запишіть оператор розгалуження для перевірки відповідності значень змінних  $a, b$  умові задачі:  
 $\text{if } \text{abs}(a/b - b/(a-b)) < 0.001: \text{print}(\text{'Так'})$   
 $\text{else: print}(\text{'Ні'})$
- Примітка.* Через особливості подання в пам'яті дробових чисел порівняння двох виразів за допомогою операції «==» не завжди дає очікуваний результат. Вважатимемо значення рівними, якщо вони відрізняються не більше ніж на деяку досить малу величину (*точність*). Тут 0,001. Функція  $\text{abs}$  повертає модуль аргумента.
- 6) Збережіть файл з іменем `Vprava1`. Запустіть програму, проаналізуйте результат її виконання у вікні консолі.  
 Випробуйте програму для різних значень змінних  $a$  і  $b$ :  
 а)  $a = 8.09, b = 5$ ; б)  $a = 10, b = 5$ .



### Комп'ютерне тестування

Виконайте тестове завдання 1 із автоматичною перевіркою результату.



## § 2. Величини. Алгоритми роботи з величинами

Комп'ютер під час роботи опрацьовує дані, які в програмі подають у вигляді величин. Так, під час моделювання руху автомобіля будуть використані такі величини, як швидкість руху, час, шлях. Розглянемо, як величини використовують у програмі.

### Основні характеристики величин

Окремий інформаційний об'єкт (число, символ, рядок тощо) називають **величиною**. Основними характеристиками величини є її вид, тип, назва та значення.

**Вид** величини визначає, чи можна змінювати значення цієї величини в програмі. Величина може бути *константою* або *змінною*.

*Константа* (від лат. *constans* — сталий) — незмінна величина.

Наприклад, під час обчислення площі круга його радіус може набувати різних значень, але величина  $\pi \approx 3,14$  є сталою.

*Змінна* — величина, значення якої може змінюватися в ході виконання програми. Змінним надають назви (ідентифікатори).



**Назва змінної** — це посилання на ділянку пам'яті, в якій зберігається її значення.

У 5 класі ви ознайомились із правилами, яких необхідно дотримуватися, вибираючи назви змінних.

**Значення** — це дані, що зберігає величина. Воно може змінюватися під час виконання програми.

Щоб **створити змінну** в Python, необхідно дати їй назву й присвоїти значення (рис. 2.1).



Рис. 2.1

1 Виконаємо дії зі змінними в інтерактивному режимі:

```

>>> day = 7           # Присвоєння значення змінній day
>>> k = 5*day         # Присвоєння значення змінній k
>>> k                 # Виведення значення змінної k
35
  
```

Під час створення змінної для збереження її значення виділяється ділянка в оперативній пам'яті. Інтерпретатор Python сам визначає, до якого типу належить створена змінна, і виділяє необхідний обсяг пам'яті.

*Тип величини* визначається множиною припустимих значень величини, операціями, які можна над нею виконувати, та обсягом пам'яті, необхідним для її збереження.

Змінні можуть зберігати дані різних типів: числа, рядки або екземпляри класів.

У 5 класі ви вже ознайомилися з простими типами даних — числовим, рядковим і логічним (рис. 2.2).

```
>>> a = 12      # Ціле число
>>> b = 3.14    # Дробове число
>>> c = 'Hello!' # Рядок
>>> print(a, b, c)
12 3.14 Hello!
```

Рис. 2.2

Щоб дізнатися, до якого типу належить змінна або константа, можна скористатися командою `type`.

```
>>> type(25)
<class 'int'>
```

## Множинне присвоювання в Python

У мові Python допустиме так зване множинне присвоювання, що дозволяє скоротити код. При цьому одне значення присвоюється кільком змінним одночасно. Таким чином можна створити кілька змінних, а потім у ході виконання програми змінити їхні значення.

Присвоїмо змінним *a*, *b*, *c* значення 0:

```
a = b = c = 0.
```

У результаті три змінні мають одне й те саме значення та зберігаються в тій самій ділянці пам'яті (рис. 2.3).

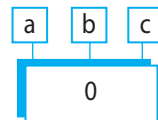


Рис. 2.3

Присвоїти значення кільком змінним можна в одному операторі.

Присвоїмо змінній *a* значення 43, змінній *b* — значення 'корови':

```
a, b = 43, 'корови'
```

Поміняємо місцями значення змінних *a* і *b* (рис. 2.4):

```
a = 3
b = 5
a, b = b, a
print(a, b)
```

# Буде надруковано 5 3

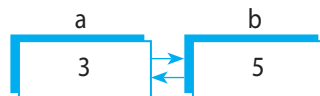


Рис. 2.4

## Алгоритмічна структура слідування

Під час конструювання алгоритмів використовуються три базові алгоритмічні структури: слідування, розгалуження, повторення. Ви ознайомилися з ними в 5 класі.



**Слідування** — це така організація дій в алгоритмі, за якої дії виконуються послідовно одна за одною (без пропусків або повторень) (рис. 2.5).

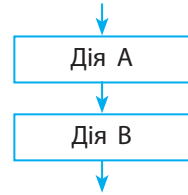


Рис. 2.5

Алгоритми, в яких використовується тільки структура слідування, називають лінійними. У програмах, які реалізують лінійні алгоритми, використовують команди введення даних, присвоювання та виведення.

## Вказівки введення та виведення даних

Функція `input()` призначена для введення даних із клавіатури. У дужках може бути записана підказка користувачеві, що саме потрібно ввести.

6 Уведемо з клавіатури значення змінної  $a$ :

```
a = input('Введіть значення a: ')

```

Функція `print()` потрібна, щоб вивести текст у вікно консолі. У списку виведення можуть бути константи, змінні, вирази. За замовчуванням після виведення всіх елементів списку курсор переводиться на наступний рядок. Якщо треба, щоб наступне виведення відбувалось у тому самому рядку, до списку виведення потрібно додати параметр `end = "` (замість `"` можна ввести текст-роздільник).

7 Виведемо значення змінних  $x = 5$  і  $y = 7$  та їх суми:

```
print('x = ', x, ', y = ', y, end = ', ') # Роздільник — кома і пропуск
print('сума дорівнює ', x+y)

```

Буде виведено:  $x = 5, y = 7$ , сума дорівнює 12.

## Перетворення типу даних

Функція `input()` повертає значення рядкового типу (`str`), а для виконання над введеними даними математичних операцій вони повинні мати числовий тип. У цьому та інших подібних випадках виникає необхідність змінити тип даних.

Для правильного опрацювання числа при введенні даних потрібно перетворити рядкове значення на числове, а при виведенні може знадобитися перетворити число на рядок. Ці операції називаються **перетворенням типів**.

Для **перетворення типів** призначені такі функції:

- `str()` — створює рядок із числа;
- `float()` — створює дробове число з рядка або цілого числа;
- `int()` — створює ціле число з рядка або дробового числа. Зверніть увагу: дробове число при цьому не округлюється, а лише відкидається дробова частина.

Проаналізуйте приклади застосування функцій перетворення типів для різних аргументів, наведені в таблиці:

Вираз	<code>int('25')</code>	<code>int(4.9)</code>	<code>int('4abc56')</code>	<code>str(25)</code>	<code>float(25)</code>
Результат виконання	25	4	Помилка	'25'	25.0

8

Складемо програму для розв'язання задачі.

*Задача.* Скільки контейнерів можна розмістити на вантажній платформі розмірами  $a \times b$  м, якщо сторона квадратної основи контейнера  $x$  м (рис. 2.6)? Сторони контейнерів мають бути паралельні до країв платформи.

Поміркуємо, скільки контейнерів можна поставити вздовж сторін платформи:  $k_1 = a : x$ ,  $k_2 = b : x$ .

Але число контейнерів має бути цілим, тобто  $k_1 = [a : x]$ ,  $k_2 = [b : x]$  (квадратні дужки позначають цілу частину числа).

Тоді загальну кількість знайдемо як добуток  $k_1$  і  $k_2$ .

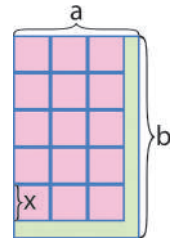


Рис. 2.6

Математична модель	Програма
Що дано? $a, b, x$ .	<code>a = float(input('a = ?'))</code>
Що треба знайти? $k$ — число контейнерів.	<code>b = float(input('b = ?'))</code>
Який зв'язок між величинами? $k = \left[ \frac{a}{x} \right] \cdot \left[ \frac{b}{x} \right]$ .	<code>x = float(input('x = ?'))</code>
Які є обмеження на дані? $a > 0, b > 0, x > 0$ .	<code>k = int(a/x)*int(b/x)</code> <code>print(k)</code>

Якщо  $a = 12.6$ ,  $b = 7.5$ ,  $x = 2.2$ , то отримаємо  $k = 15$ .

Отже, необхідно вміти визначати типи величин, потрібних для розв'язування поставленої задачі, й оперувати ними в програмі.

## Питання для самоперевірки



1. Поясніть, як виконується команда `input()`.
2. Назвіть константи та змінні у списку виведення:  
`print('a = ', a, 5, '3*b', 3*b)`
3. Знайдіть значення  $x$  після виконання послідовності присвоєвань:  
а)  $y = 2$ ;  $x = y$ ;      б)  $x = 8$ ;  $x = x + 2$ ;      в)  $x = 5$ ;  $x = -x$ .
4. Складіть програму для знаходження суми цифр трицифрового цілого десяткового числа. (Згадайте операції `//` (обчислення неповної частки від ділення), `%` (обчислення остачі від ділення)).
5. Складіть програму для розв'язування задачі.  
Потрібно обклеїти шпалерами стіну довжиною  $a$  м і шириною  $b$  м. Скільки коштуватимуть шпалери, якщо ціна одного рулону  $k$  грн? Довжина рулону шпалер 10 м, ширина — 80 см.



## Вправа 2



- Вартість поїздки на таксі тривалістю до 5 хв становить 20 грн, а вартість кожної наступної хвилини — 10 грн. Складіть програму для розрахунку вартості поїздки тривалістю  $t$  хв (для  $t \geq 5$ ).



- 1) Проаналізуйте задачу та складіть математичну модель.

*Що дано?*  $t$  — тривалість поїздки (хв).

*Що треба знайти?*  $c$  — вартість поїздки (грн).

*Який зв'язок між величинами?*  $c = 20 + 10(t - 5)$ .

*Які є обмеження на дані?*  $t \geq 5$ .

- 2) Запишіть оператор введення тривалості поїздки й присвоєння цього значення змінній  $t$ : `t = int(input('t = ?'))`
- 3) Запишіть оператор присвоєння значення змінній  $c$ .
- 4) Запишіть оператор виведення значення змінної  $c$ , додавши до списку виведення текстову константу «грн».
- 5) Збережіть файл з іменем `Vprava2`. Випробуйте програму.
- 6) Доповніть програму перевіркою правильності введеного значення ( $t \geq 5$ ). Випробуйте програму для різних значень  $t$ .



## Комп'ютерне тестування

Виконайте тестове завдання 2 із автоматичною перевіркою результату.





## § 3. Створення алгоритмів і програм із розгалуженнями та повтореннями

У мові Python, як і в інших мовах програмування, можна створювати програми з розгалуженнями та повтореннями. Тож пригадаймо відповідні оператори та розглянемо приклади їх використання.

### Алгоритмічна структура розгалуження

Якщо залежно від певних умов можливі декілька варіантів дій, при складанні алгоритму використовується алгоритмічна структура розгалуження.



**Розгалуження** — це алгоритмічна структура, в якій виконання заданої послідовності дій залежить від істинності певної умови (логічного виразу).

Розглянемо блок-схеми неповного (рис. 3.1) і повного (рис. 3.2) розгалужень, а також приклади запису відповідних умовних операторів.

Оператори, які належать до гілок `if` і `else`, записують із обов'язковим однаковим відступом вправо.

Неповне розгалуження	Повне розгалуження
<p>Рис. 3.1</p>	<p>Рис. 3.2</p>
<pre>if a&gt;b+c:     print('Трикутник не існує')     a = 0.8*a</pre>	<pre>if n%2 == 0:     print('Число парне') else:     print('Число непарне')</pre>

Як ви вже знаєте з 6 класу, якщо потрібно послідовно перевірити декілька умов, застосовують множинне розгалуження. Для цього до оператора розгалуження слід додати потрібну кількість гілок `elif` (від англ. *else if* — інше якщо), в кожній з яких перевіряється певна умова.

1 Створимо програму-довідник, яка за введеною назвою пори року повідомлятиме назви відповідних їй місяців.

```
W = input('Введіть назву пори року: ')
if W == 'весна': print('березень, квітень, травень')
elif W == 'літо': print('червень, липень, серпень')
elif W == 'осінь': print('вересень, жовтень, листопад')
elif W == 'зима': print('грудень, січень, лютий')
else: print('неправильні дані')
```

Під час виконання програми будь-який рядок, відмінний від української назви пори року, набраної малими літерами, оброблятиметься гілкою else — виводитиметься повідомлення «неправильні дані».

### Алгоритмічна структура повторення

Одним із засобів програмування є можливість багаторазового виконання деякого набору команд. Це дозволяє створювати програми для обробки великих сукупностей даних, побудови візерунків тощо.

Команди, що повторюються, разом зі службовими словами, які забезпечують керування цим процесом, називаються *циклом*.



**Повторення (цикл)** — це алгоритмічна структура, в якій певна послідовність дій може бути виконана кілька разів.

Як ви знаєте, повторювані під час виконання циклу дії називають **тілом циклу**, а кожне виконання тіла циклу — **ітерацією**.

У мові Python реалізовано два види операторів циклу: з умовою (рис. 3.3) і з параметром (рис. 3.4).

Цикл із умовою		Цикл із параметром	
<p style="text-align: center;"><b>Рис. 3.3</b></p>	<pre>while a&lt;0:     print(50*a)     a = a+10</pre>	<p style="text-align: center;"><b>Рис. 3.4</b></p>	<pre>for x in range(1, 100, 2):     y = x*2     print(x, y)</pre>

Як і в умовному операторі, команди тіла циклу слід записувати з однаковим відступом від початку рядка із заголовком циклу.

### ► Цикл із умовою

Під час виконання циклу з умовою (while) спочатку перевіряється умова. Якщо вона істинна, то виконується тіло циклу.

Якщо умова хибна, то виконання циклу припиняється.

Якщо при першій перевірці умова виявиться хибною, то тіло циклу не виконається жодного разу.

Якщо в ході виконання циклу умова залишається істинною, то цикл буде повторюватися нескінченно (кажуть, програма «зациклена»).

2 Знайти суму цифр числа  $n$ :

```
n = int(input('n = ?'))
```

```
s = 0
```

```
while n > 0:
```

```
    d = n % 10
```

```
    s = s + d
```

```
    n = n // 10
```

```
print('s =', s)
```

n	d	s
392	—	0
Початок циклу		
392 > 0	2	2
39 > 0	9	11
3 > 0	3	14
0	Цикл припинено	

### ► Цикл із параметром

Тіло циклу із параметром (for) виконується задану кількість разів. Діапазон значень параметра можна задати кількома способами.

- За допомогою функції range(start, stop, step). Ця функція повертає послідовність цілих чисел від start до stop із кроком step. Значення stop у послідовність не включається.

Якщо start = 0 або step = 1, то ці параметри можна не зазначати. Щоб вести відлік у зворотному порядку, step має бути від'ємним.

3 Надрукуємо парні двоцифрові числа у порядку спадання:

```
for i in range(98, 9, -2):
```

```
    print(i)
```

- За допомогою списку значень. Список береться у квадратні дужки; параметр почергово набуває всіх значень зі списку.

4 Виведемо квадрати чисел зі списку значень:

```
for i in [2, 5, 3, 12]:
```

```
    print(i**2)
```

У тілі циклу можна використовувати будь-які команди.

5 Складемо програму для розв'язання задачі. *Задача.* Знайти двоцифрове число, яке на 26 більше від добутку своїх цифр.

Нехай  $a$  — кількість десятків,  $b$  — кількість одиниць. Тоді двоцифрове число можна подати у вигляді  $10a + b$ .

За умовою число  $10a + b$  на 26 більше від добутку  $ab$ . Отже, визначимо зв'язок між величинами:

$$10a + b = ab + 26; 0 < a < 10; 0 \leq b < 10.$$

Складемо програму для пошуку розв'язків методом перебору. Запишемо оператори циклу для перебору всіх можливих комбінацій цифр  $a$  і  $b$  та перевіримо для кожної з них умову задачі.

```
for a in range(1, 10):
    for b in range(0, 10):
        if 10*a+b == a*b+26: print(10*a+b)
```

Задача має три розв'язки: числа 32, 56, 98.

6 Досконалыми давні греки називали числа, що дорівнюють сумі всіх дільників, окрім самого числа ( $6 = 1 + 2 + 3$ ). Складемо програму для перевірки, чи є введене число досконалим.

Загальний алгоритм такий:

- 1) ввести число  $k$ ;
- 2) обчислити суму дільників, крім самого числа  $k$ ;
- 3) якщо сума дільників дорівнює числу  $k$ , то повідомити, що  $k$  — досконале.

Щоб знайти суму дільників числа (див. коментарі до програми), можна перебрати цілі числа від 1 до половини  $k$ , щоразу перевіряючи, чи є число дільником  $k$ . Знайдені дільники слід додавати до змінної  $sum$ . Описаному алгоритму відповідає така програма:

```
k = int(input('k = ?'))
sum = 0
for i in range(1, k//2+1):
    if k%i == 0:
        sum = sum+i
if sum == k: print(k, '- досконале число')
else: print(k, 'не є досконалим числом')
```

Для перевірки програми скористайтесь тим, що перші 4 досконалі числа — 6, 28, 496 і 8128.

Використовуючи базові алгоритмічні структури, можна побудувати будь-який як завгодно складний алгоритм.

## Питання для самоперевірки



1. Дано фрагмент коду:

```
if a%10<3: b = a*2
elif a%10 <= 5: b = a+10
elif a%10 <= 7: b = a*10
else: b = a+1
```

Чому дорівнюватиме значення змінної  $b$  після виконання коду, якщо спочатку: а)  $a = 13$ ; б)  $a = 30$ ; в)  $a = 88$ ; г)  $a = 11$ ?

2. Запишіть оператор розгалуження для визначення за назвою дня тижня, робочий він, чи вихідний.  
3. Що буде надруковано після виконання фрагмента програми?

```
a) for i in range(10):
    if i%2 == 0:
        print(i)
б) for i in range(4):
    print(i*2)
в) x = 2
    while x <= 5:
        x = x+1
    print(x, end = ")
```



4. Запишіть оператори циклу для друку двоцифрових двійкових чисел: 00, 01, 10, 11.



5. Одна пальчикова батарейка може забруднити 400 л води. Об'єм води в озері Світязь становить 180 000 000 м<sup>3</sup>. Запишіть оператор циклу для визначення того, на який день забрудниться все озеро, якщо першого дня безвідповідально викинути 1 батарейку, а кожного наступного — удвічі більше, ніж попереднього.



6. Складіть програму для розв'язування задачі методом перебору. Придбали олівці по 6 грн і зошити по 5 грн. Вартість покупки склала 68 грн. Скільки купили олівців і скільки зошитів?

## Вправа 3



- Написати програму для пошуку розв'язку задачі методом перебору. Знайти можливий номер автомобіля, що порушив правила дорожнього руху, якщо відомо, що номер чотирицифровий, кратний 5 і 7, а сума його цифр становить  $K$  ( $4 \leq K \leq 36$ ).

- 1) Запрограмуйте повторення запиту на введення змінної  $k$ , поки користувач не введе число, яке відповідає умові:

```
k = 1
while k < 4 or k > 36:
    k = int(input("Введіть число від 4 до 36"))
```



- 2) Позначте цифри номера через  $a$ ,  $b$ ,  $c$  і  $d$ . Напишіть оператори циклу для перебору всіх можливих комбінацій значень  $a$ ,  $b$ ,  $c$  і  $d$ .

Для кожного з варіантів номера слід перевірити умову задачі і, якщо умова істинна, вивести число, що утворюється з цифр  $a$ ,  $b$ ,  $c$  і  $d$ :

```
if a+b+c+d == k and (1000*a+100*b+10*c+d)%35 == 0:
    print(1000*a+100*b+10*c+d)
```

- 3) Збережіть файл з іменем Vprava3.  
4) Випробуйте програму для: а)  $k = 25$ ; б)  $k = 12$ .

### Комп'ютерне тестування

Виконайте тестове завдання 3 із автоматичною перевіркою результату.



## § 4. Рядки як послідовності символів

Багато задач програмування пов'язано з опрацюванням текстових даних: задачі шифрування, перевірки тексту на відсутність плагіату, перекладу з однієї мови на іншу, ігри зі словами тощо.

Для опрацювання текстових даних у Python існує тип даних `str` (від англ. *string* — рядок).

**!** Значенням величини типу `str` є послідовність символів.

Рядкові константи в Python записують в одинарних або подвійних лапках. Початок і кінець рядка мають бути в лапках одного виду.

### Коди символів

Усі символи (літери, цифри, знаки пунктуації тощо) утворюють так звану таблицю символів. Кожний символ у таблиці має свій номер, який називають його **кодом**.

Код будь-якого символу можна визначити за допомогою функції `ord()`:

```
>>> ord('A')          # Велика англійська літера 'A'
65
```

Дізнатися, який символ відповідає певному коду, можна за допомогою функції `chr()`:

```
>>> chr(1040)        # Код великої української літери 'А'
'А'
```

- ! Великі й малі літери, літери англійського та українського алфавітів із однаковим написанням мають різні коди.

## Нумерація символів у рядку

Кожний символ у рядку має унікальний порядковий номер — **індекс**. Нумерація символів починається з нуля. Розглянемо приклади роботи з рядками в інтерактивному режимі.

До конкретного символу в рядку можна звертатися за його індексом, зазначивши індекс у квадратних дужках.

1 Надрукуємо першу літеру рядка Червона калина:

```
>>> a = 'Червона калина'
>>> a[0]
'ч'
```

Можна починати відлік символів із кінця рядка. У цьому випадку індекси починаються з  $-1$  (останній символ).

2 Надрукуємо третю з кінця літеру рядка Червона калина:

```
>>> a[-3]
'и'
```

3 Присвоїмо змінній *b* символ рядка *a* з індексом 8:

```
>>> b = a[8]
>>> b
'к'
```

Функція `len()` повертає кількість символів, із яких утворено рядок.

**Синтаксис функції:**

```
k = len(s),
```

де *k* — змінна цілого типу для збереження значення, яке повертає функція; *s* — рядок, довжину якого потрібно визначити.



4 Визначимо довжину рядка Червона калина:

```
>>>len(a)
```

```
14
```

Пропуск теж є символом, тому довжина рядка дорівнює 14.

## Отримання зрізів

З рядка можна вибрати частину символів (зріз (підрядок)).

**Оператор добування зрізу з рядка:**

```
z = s[i: j: step],
```

де  $z$  — змінна, в яку зберігається підрядок рядка  $s$ ;  $i$  — індекс початку зрізу;  $j$  — індекс його закінчення (причому символ із номером  $j$  у зрізі не входить);  $step$  — крок, з яким вибираються символи.

Якщо символи вибираються підряд, параметр  $step$  можна опустити. Якщо відсутній індекс початку зрізу, то зріз береться від 0-го до  $(j-1)$ -го символу; за відсутності індексу закінчення зрізу, зріз береться від  $i$ -го символу до кінця рядка.

5 Скопіюємо з рядка  $a$  підрядок Червона:

```
>>> a = 'Червона калина'
```

```
>>> a[: 7]
```

```
'Червона'
```

6 Скопіюємо з рядка  $a$  підрядок калина:

```
>>> a[8:]
```

```
'калина'
```

7 Скопіюємо з рядка  $a$  символи з парними індексами:

```
>>> a[:: 2]
```

```
'Чроаклн'
```

**!** Рядки — незмінювані послідовності. Якщо спробувати змінити значення символу всередині рядка, виникне помилка.

## Перегляд символів у рядку

Перевірити, чи є певний рядок в іншому рядку, можна оператором `in`:

```
>>> print('y' in 'Python')
```

```
True
```

За допомогою циклу `for` можна вивести рядок по літерах:

```
>>> for c in a: print(c)
```

## Операції конкатенації і дублювання

Об'єднання двох або більше рядків в один рядок називають **конкатенацією**. Цю операцію позначають символом «+».

```
8 Об'єднаємо рядкові змінні a і b, додавши між ними пропуск.
>>> a = 'Богдан'
>>> b = 'Хмельницький'
>>> c = a + ' ' + b          # c = 'Богдан Хмельницький'
```

**Дублювання** — це повторення текстового фрагмента задану кількість разів. Цю операцію позначають символом «\*».

```
9 Повторимо символ «-» 20 разів.
>>> c = '-'*20              # c = '-----'
```

```
10 Потроїмо кожний символ рядка s.
s = '12345'
s1 = ""
for i in s:
    s1 = s1+i*3
print(s1)                   # Буде надруковано 111222333444555
```

## Створення вікна програми з написом і кнопкою

Пригадаємо основні кроки зі створення програми з графічним інтерфейсом із використанням модуля tkinter.

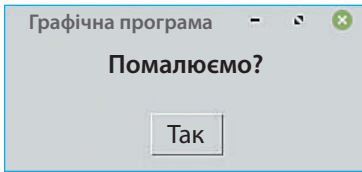
Побудова програми починається зі створення вікна. У вікно додаються інші елементи керування. Для деяких елементів керування створюються обробники подій.

Проаналізуємо програмний код створення вікна з написом і кнопкою.

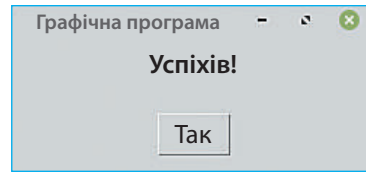
<code>from tkinter import*</code>	імпорт змісту модуля tkinter
<code>def btn_click():     lab.config(text = 'Успіхів!')</code>	опис методу btn_click — обробника події натискання на кнопку
<code>root = Tk()</code>	створення об'єкта root класу Tk() (вікна)
<code>root.title('Графічна програма')</code>	задання заголовка вікна
<code>root.geometry('250x100')</code>	задання розмірів вікна

lab = Label(root, text = 'Помалюємо?', font = 'Arial 14')	створення об'єкта lab класу Label (напису)
lab.place(x = 80, y = 1)	
btn = Button(root, text = 'Так', command = btn_click)	створення об'єкта btn класу Button (кнопки). При натисканні на кнопку ви- кликається метод btn_click
btn.place(x = 100, y = 50)	
root.mainloop()	запуск головного циклу обробки подій

Якщо зберегти й запустити цю програму, побачимо вікно програми (рис. 4.1, а). Після натискання на кнопку напис зміниться на 'Успіхів!' (рис. 4.1, б).



а



б

Рис. 4.1

### Питання для самоперевірки



1. Запишіть оператор, який дозволив отримати з рядка *s* такий рядок: а) *s1* = 'форма'; б) *s1* = 'інтика'.  
*s* = 'інформатика'
2. Запишіть оператори, що реалізують присвоєння змінній *b* таких зрізів рядка *a*: а) перші вісім символів; б) символи з індексами, кратними трьом.
3. Отримайте за допомогою зрізів рядки 'програміст' та 'зірка' з рядка 'Якщо ви можете запрограмувати комп'ютер, то ви зможете досягти своєї мрії. Дік Костоло'.
4. Розгадайте фразу, яку зашифровано в рядку *s1*. Який зріз із рядка *s1* потрібно добути для розшифрування фрази?  
*s1* = 'яа клеюгбмлщюй кіеннфгошрщмзастхиткуур'
5. Задано рядок символів. Поміняйте місцями першу половину рядка з другою. Приклади: 123абв → абв123; 1234абв → абв4123.
6. Опишіть основні кроки зі створення програми з графічним інтерфейсом.



## Вправа 4



► Скласти програму Рухомий рядок.

- 1) Запишіть команду завантаження модуля tkinter.
- 2) Створіть вікно програми з розмірами  $300 \times 100$  із заголовком Реклама (рис. 4.2).
- 3) До змінної `s` занесіть текст реклами, наприклад:  
`s = 'Вивчайте мову Python!'`

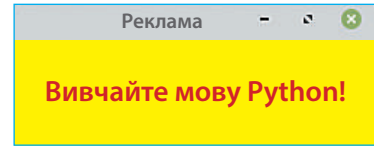


Рис. 4.2

- 4) Додайте до вікна віджет `lab` класу `Label`. Задайте значення атрибутів об'єкта `lab` на ваш розсуд, наприклад:  
`lab = Label(root, text = s, font = ('Comic Sans MS', 18, 'bold'), width = 20, height = 40, bg = 'yellow', fg = 'red')`  
`lab.pack()`



- 5) Для того щоб рядок рухався, створіть функцію `clock()`. При кожному виклику функції 0-й символ рядка `s` переміщується в кінець рядка й оновлюється властивість `text` віджета `lab`:  
`def clock():`

```

    global s                # Оголошення змінної s як глобальної
    s = s[1:]+s[0]         # 0-й символ рядка s переміщується
                           # в кінець рядка
    lab.config(text = s)   # Оновлення властивості text віджета lab
    root.after(200, clock) # Повторний виклик clock через 0,2 с

```

- 6) В основній програмі запишіть команду виклику функції `clock`, а в останньому рядку програмного коду — оператор запуску головного циклу обробки подій:

```

clock()
root.mainloop()

```

Збережіть код у файлі `Vprava4`.

Випробуйте програму.

Поекспериментуйте з нею, змінюючи рекламний текст, періодичність виклику функції `clock` тощо.

## Комп'ютерне тестування



Виконайте тестове завдання 4 із автоматичною перевіркою результату.



## § 5. Опрацювання рядкових величин

Ви вже знайомі з прийомами опрацювання текстів у текстовому редакторі, наприклад із пошуком і заміною символів. Мова Python має засоби для того, щоб запрограмувати подібні дії з рядковими величинами. Зважаючи на те, що всі змінні в Python є об'єктами, відповідні операції доступні у вигляді методів.

**Синтаксис виклику методу для рядкової змінної:**

<назва змінної>.<назва методу> ([<параметри>])

**!** Рядкові методи в Python не змінюють рядка, а утворюють новий.

Якщо потрібно далі в програмі використовувати результат виконання методу, його слід зберегти в новій змінній.

### Рядкові методи

Розглянемо деякі методи опрацювання рядкових величин і проаналізуємо результат застосування цих методів в інтерактивному режимі. У наведених далі прикладах методи викликаються для змінної `s`, яка містить рядок, який опрацьовується.

#### ► Пошук символів у рядку

Метод `find()` шукає задану групу символів у рядку `s` і повертає номер першого символу шуканого підрядка в рядку.

Якщо підрядок не знайдено, результат виклику методу дорівнює `-1`:  
`номер = s.find(subs)`,

де `номер` — змінна цілого типу, яка отримує значення індексу першого символу підрядка `subs` у рядку `s`; `subs` — шуканий підрядок.

**1** Знайдемо індекс першого символу підрядка 'га' у рядку `s`:

```
>>> s = 'Гуси, гуси, га-га-га'
>>> n = s.find('га')           # n = 12
```

#### ► Заміна символів у рядку

Метод `replace()` шукає й замінює шуканий підрядок у заданому рядку:

```
srez = s.replace(sfind, sins),
```

де `srez` — змінна для рядка, що повертає метод; `sfind` — підрядок, який потрібно замінити у рядку `s`; `sins` — рядок, яким буде замінений підрядок `sfind`.

Якщо `sins = ""` (порожній рядок), то метод `replace` просто вилучить знайдені підрядки.

2 Замінімо в рядку `s` усі підрядки `'cry'` на підрядки `'smile'` і результат запишемо в змінну `s1`:

```
>>> s = 'Why do You cry, Willy, why do You cry?'
>>> s1 = s.replace('cry', 'smile')
>>> s1
'Why do You smile, Willy, why do You smile?'
```

### ► Визначення кількості входжень підрядка

За допомогою методу `count()` можна визначити кількість входжень заданої групи символів у рядок:

```
kilk = s.count(subs),
```

де `kilk` — змінна цілого типу, яка отримує значення кількості входжень підрядка `subs` у рядок `s`; `subs` — шуканий підрядок.

3 Підрахуємо кількість повторень підрядка `'га'` в заданому рядку `s`. Результат присвоїмо змінній `k`:

```
>>> s = 'Гуси, гуси, га-га-га'
>>> k = s.count('га')           # k = 3
```

### ► Перевірка складу рядка

Іноді буває потрібно перевірити, чи складається рядок тільки з літер або тільки з цифр. Для цього в Python є спеціальні методи, які повертають логічне значення (`True` або `False`). А саме:

- `s.isdigit()` — повертає `True`, якщо рядок `s` містить тільки цифри;
- `s.isalpha()` — повертає `True`, якщо рядок `s` містить тільки літери.

4 Напишемо код, який пропонує користувачеві створити пароль, що складається тільки з літер:

```
parol = "                               Введіть пароль: 123secret
while not(parol.isalpha()):             Введіть пароль: secret
    parol = input('Введіть пароль: ')   Ваш пароль secret
print('Ваш пароль' + parol)           >>>
```

## Функції користувача

Методи можна порівняти з цеглинками, з яких ви складаєте свою програму. Що робити, якщо немає методу, який виконує потрібну в алгоритмі розв'язування задачі дію? Ми можемо самі створити таку «цеглинку» — описати функцію користувача.

З правилами опису й виклику функцій користувача ви ознайомились у 6 класі.

### Синтаксис опису функції:

```
def <назва функції>([<формальні параметри>]):
    <тіло функції>
```

Тіло функції являє собою послідовність операторів, які будуть виконані після викликання функції. Після того як функцію створено, її можна викликати з іншої функції або безпосередньо з програми.

Щоб викликати функцію, потрібно:

- 1) ввести назву функції й додати дужки;
- 2) у дужках перелічити аргументи функції — значення, які під час виклику будуть присвоєні формальним параметрам.

Якщо формальних параметрів немає, то після назви функції потрібно залишити порожні дужки.

- 5 Опишемо функцію `ins` для вставляння підрядка `subs` у заданий рядок `s` після  $n$ -го символу:

```
def ins(s, subs, n):
    snew = ""
    snew = s[0:n+1]+subs+s[n+1:len(s)]
    return snew
s = '1234'
subs = 'abc'
s = ins(s, subs, 2)
print(s)                                # Буде надруковано '123abc4'
```

- 6 Запишемо рядок `s` у зворотному порядку, вилучивши пропуски:

```
def rev(s):
    snew = s.replace(' ', '')           # Вилучаємо пропуски
    snew = snew[::-1]                   # Розвертаємо рядок
    return snew
s = '1 2 3 4'
print(rev(s))
```



- 7 Зашифруємо речення шляхом вставлення заданого рядка після кожної літери:

```
def shifr(s, c):
    snew = ""
    for i in range(len(s)):
        snew = snew+s[i]+c
    return snew
s = 'Секрет'
c = 'abc' # Рядок для вставлення
print(shifr(s, c)) # Буде надруковано 'Cabсеабскабсрабсеабстабс'
```

- 8 Задано текстовий рядок. Вилучимо з нього всі цифри.

```
def lit(s):
    snew = ""
    for c in s:
        if not(c.isdigit()):
            snew = snew+c
    return snew
s = 'd3fgh56as67'
s = lit(s)
print(s) # Буде надруковано 'dfghas'
```

Опрацювання рядкових величин лежить в основі програм, які призначено для шифрування та дешифрування текстів, автоматичного перекладу тощо.

### Питання для самоперевірки



- За допомогою яких засобів можна отримати підрядок заданого рядка? Поясніть на прикладі.
- Як можна вилучити підрядок із заданого рядка?
- Як запрограмувати вставлення підрядка в заданий рядок?
- Обчисліть значення функції:
 

а) <code>chr(90)</code> ;	б) <code>ord('z')-ord('y')</code> ;	в) <code>ord(chr(13))</code> .
---------------------------	-------------------------------------	--------------------------------
- Яких значень набудуть змінні внаслідок послідовного виконання операторів, якщо початкові значення  $a = \text{'ми'}$ ;  $b = \text{'вивчаємо'}$ ;  $c = \text{'Python'}$ .
 

а) $c = a+c+b$ ;	в) $b = c[12:]$ ;
б) $x = \text{len}(c)$ ;	г) $b = c[0:10:2]$ ;

г) `a = c.replace('и', '')`;

е) `x = c.count('в')`.

д) `x = c.find('Р')`;



6. Непорожній рядок, який однаково читається як зліва направо, так і справа наліво, називається паліндромом (наприклад, «козак з казок»). Напишіть програму для визначення, чи є уведений рядок паліндромом.

### Вправа 5



- У XIX ст. для пересилання важливих повідомлень використовувались шифри, в яких літери переставляються за певним правилом. Напишіть програму, що шифрує повідомлення, міняючи місцями кожні два знаки.

*Наприклад:* введення: Я чекатиму на тебе завтра

виведення: Яечакитумн аетєбз варт а

- 1) Запрограмуйте введення значення змінної `s`. Передбачте виведення підказки "Введіть текст для шифрування".
- 2) Створіть змінну `s1` для формування результату і запишіть у неї порожній рядок.
- 3) Число символів у рядку `s` має бути парним, тому запишіть оператор розгалуження для реалізації такої дії: якщо довжина рядка `s` є непарним числом, то додати до рядка пропуск:
 

```
if(len(s)%2 != 0):
    s = s+' '
```
- 4) Запишіть оператор циклу для перебору непарних індексів. В тілі циклу до рядка `s1` додаються символи `s[i+1]` і `s[i]`:
 

```
for i in range(0, len(s), 2):
    s1 = s1+s[i+1]+s[i]
```
- 5) Виведіть зашифроване повідомлення.
- 6) Збережіть програмний код у файлі з іменем `Vprava5`. Випробуйте програму. Переконайтесь, що повідомлення успішно розшифровується за допомогою цієї ж програми.

### Комп'ютерне тестування

Виконайте тестове завдання 5 із автоматичною перевіркою результату.





## Практична робота 1

### Створення програм для опрацювання рядкових величин

**Завдання:** скласти програму для розв'язування задачі.

*Задача.* Дано номер мобільного телефону й прізвище власника (власниці). Зашифрувати прізвище власника, замінивши символи їх кодами. Визначити PIN-код власника, що дорівнює сумі цифр у номері телефону. Вивести в зашифрованому вигляді прізвище власника та PIN-код (див. рисунок).

**Обладнання:** комп'ютер зі встановленим середовищем програмування Python.

#### Хід роботи

*Під час роботи з комп'ютером дотримуйтеся правил безпеки.*

У Python IDLE виберіть команду File → New File.

- ▶ 1. Запишіть команду завантаження модуля tkinter.
- ▶ 2. Створіть вікно програми розмірами 500 × 200 із заголовком Шифрування.
- ▶ 3. Створіть віджети lab1–lab4 класу Label. Задайте для них значення властивості text відповідно до рисунка.  
`lab1 = Label(root, text = 'Прізвище')`  
`lab1.place(x = 60, y = 10)`

- ▶ 4. Створіть віджети `entry1`, `entry2` класу `Entry` для введення прізвища та номера мобільного телефону:
 

```
s = "
entry1 = Entry(root, textvariable = s, width = 15)
entry1.place(x = 200, y = 10)
```
- ▶ 5. Створіть віджети `entry3`, `entry4` класу `Entry` для виведення зашифрованого прізвища власника та його PIN-коду.
- ▶ 6. Додайте до вікна віджет `b1` класу `Button`:
 

```
b1 = Button(root, text = 'Зашифрувати', command = b1_click)
b1.place(x = 100, y = 90)
```
- ▶ 7. Запишіть заголовок функції — обробника події для кнопки `b1`:
 

```
def b1_click():
```
- ▶ 8. У тілі функції — обробника події для кнопки візьміть значення з текстового поля `entry1`:
 

```
s = entry1.get()
```

 Створіть змінну `shifr` і запишіть у неї порожній рядок. У циклі `for` визначте код кожного символу рядка `s`, перетворіть числове значення коду на тип `str` і додайте до рядка `shifr`:
 

```
shifr = "
for i in range(len(s)):
    shifr = shifr+str(ord(s[i]))+''
```
- ▶ 9. Значення рядка `shifr` виведіть до текстового поля `entry3`:
 

```
entry3.delete(0, END)
entry3.insert(0, shifr)
```
- ▶ 10. У тілі функції прочитайте значення з текстового поля `entry2`:
 

```
number = entry2.get()
```

 Для обчислення суми цифр у рядку `number` кожний символ рядка `number[i]` перетворіть на тип `int` і додайте до значення суми `pin`:
 

```
pin = 0
for i in range(len(number)):
    pin = pin+int(number[i])
```

 Виведіть отримане значення суми `pin` до текстового поля `entry4`.
- ▶ 11. Останнім рядком програмного коду запишіть оператор запуску головного циклу обробки подій.
- ▶ 12. Збережіть програмний код з іменем `Практична1`. Випробуйте програму.

**Зробіть висновок** про результати роботи.

## § 6. Списки — змінювані послідовності

Ми вже розглянули прості типи даних — цілі числа, дробові числа, рядки, логічні значення. Величина простого типу містить одне значення. Але в житті ми постійно стикаємося з наборами значень, об'єднаних за певними ознаками, наприклад, список гравців футбольної команди, значення середньодобових температур за тиждень.

### Поняття списку

Для опрацювання сукупностей даних у програмуванні існують структуровані типи, один з яких — список. Величина структурованого типу містить множину значень зі спільною назвою.



**Список** (*list*) — це упорядкований набір об'єктів довільних типів (чисел, рядків, списків тощо), який можна змінювати.

У тексті програми список вкладають у квадратні дужки [], а його елементи відокремлюють комами.

Ви вже використовували списки кольорних констант під час малювання Черепашкою:

```
colors = ['red', 'yellow', 'green', 'blue']
```

Як бачимо, створити змінну типу список можна за допомогою оператора присвоєння.

**1** Створимо змінні типу список:

```
list1 = [] # Порожній список
list2 = [3.14, 9.8, 2.4, 1.0] # Список із дробових чисел (тип float)
list3 = ['робот', 'машина', 'ведмедик', 'курча'] # Список із рядків (тип str)
```

Список можна уявити як полицю, на якій розташовано деякі об'єкти (рис. 6.1).

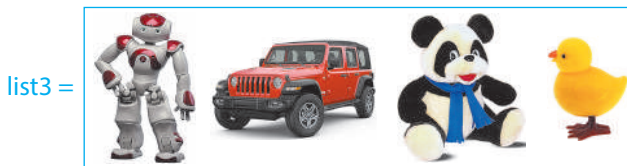


Рис. 6.1

Додати елемент до списку можна за допомогою методу `append()`. Доданий елемент займе місце в кінці списку.

2 Додамо до списку елемент 'равлик':

```
list3.append('равлик')
print(list3)
```

Буде виведено: ['робот', 'машина', 'ведмедик', 'курча', 'равлик'].

3 Створимо список із 10 чисел, уведених із клавіатури:

```
a = []
for i in range(10):
    a.append(int(input()))
print(a) # Буде надруковано список уведених чисел
```

! Додати елемент можна тільки до вже створеного списку (можливо, порожнього).

## Операція in

Перевірити, чи є в списку певний елемент, допоможе операція `in`. Синтаксис використання операції `in`:

<значення> `in` <послідовність>,

де послідовність — рядок, список, діапазон тощо, в якому виконується пошук значення.

Результат операції — значення логічного типу: `True`, якщо послідовність містить значення, або `False`, якщо не містить.

4 Визначимо, чи є в списку `letters` елемент 'a':

```
if 'a' in letters: print('Так')
else: print("Символа 'a' в списку немає")
```

Умова `'a' in letters` набуває значення `True`, якщо в списку є літера «а», або значення `False` — якщо немає.

## Нумерація елементів списку

Списки, як і рядки, є впорядкованими послідовностями, тому можна отримувати доступ до елементів списку за їхніми індексами, добувати зрізи тощо. Нумерація елементів починається з нуля (рис. 6.2).

	0	1	2	3	4	5
lst =	'a'	'b'	'c'	'd'	'e'	'f'
	lst[0] = 'a'					

Рис. 6.2

- 5 Дано список ['a', 'b', 'c', 'd', 'e', 'f']. Проаналізуємо результати виконання дій зі списком.

	Дія	Оператор	Результат виконання
1	Визначити довжину списку	>>> len(lst)	6
2	Вивести елемент списку з індексом 3	>>> lst[3]	'd'
3	Вивести перші 3 елементи списку	>>> lst[0:3]	['a', 'b', 'c']
4	Вивести елементи списку, починаючи з елемента з номером 3 і до кінця	>>> lst[3:]	['d', 'e', 'f']
5	Вивести зріз списку довжиною 1 елемент	>>> lst[3:4]	['d']

! Зріз списку теж є списком. У прикладі 5 (2) для отримання елемента списку вказано його індекс, тому отримано значення типу string. Але якщо вказати суміжні індекси через двокрапку (приклад 5 (5)), отримаємо зріз у вигляді списку з одного елемента.

На відміну від рядків, списки — це змінювані послідовності, тобто елементи списку можна змінювати, вилучати, додавати нові.

- 6 Змінимо значення 1-го елемента списку mylist на "м'яч" (рис. 6.3). (У слові «м'яч» є апостроф, тому слово потрібно взяти в подвійні лапки.)

```
>>> mylist = ['робот', 'машина', 'ведмедик', 'курча']
>>> mylist[1] = "м'яч"
>>> mylist
['робот', 'м'яч', 'ведмедик', 'курча']
```

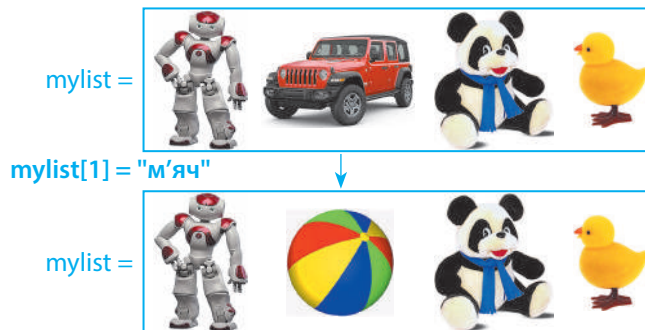


Рис. 6.3



7 Замінімо перші три елементи списку нулями:

```
>>> lst = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
>>> lst[0:3] = [0, 0, 0]
>>> lst
[0, 0, 0, 4, 5, 6, 7, 8, 9, 10]
```

## Об'єднання й повторення списків

Як і над рядками, над списками можна виконувати операції об'єднання (+) та повторення (\*).

8 Подвоїмо список [1, -2, 5]:

```
>>> a = [1, -2, 5]
>>> a = a*2
>>> a
[1, -2, 5, 1, -2, 5]
```

9 Об'єднаємо списки [1, -2, 5] і [2, 3, 0]:

```
>>> c = [1, -2, 5]+[2, 3, 0]
>>> c
[1, -2, 5, 2, 3, 0]
```

Списки дуже корисні в програмах, у яких потрібно працювати з сукупностями даних (наприклад, таблиця результатів спортивних змагань). Як бачимо, використання списків дозволяє переглянути в циклі весь набір елементів, опрацювати всі елементи за єдиним алгоритмом.

## Питання для самоперевірки



1. Як створити порожній список?
2. Як додати елемент до списку?
3. Як отримати частину заданого списку? Поясніть на прикладах.



4. Дано список  $a = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$ . Чому дорівнюють значення виразів: а)  $\text{len}(a)$ ; б)  $a[1] + a[3]$ ?
5. Дано список  $a = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$ . Яких значень набувають елементи списку  $b$ , якщо:

а)  $b = a[0:3]$ ;      в)  $b = a$ ;      г)  $b = a[2:3]+[1, -2, 0]$ ;  
 б)  $b = a[6:]$ ;      г)  $b = [-1, 0, 1]$ ;      д)  $b = [-1, -2, 5]+[2, 3, 0]$ .



6. Заповніть список  $a$  10 цілими випадковими числами ( $0 \leq a_i < 10$ ). Перевірте, чи є у списку число 0. Створіть список  $b$  з елементів списку  $a$ , що мають парні індекси.

## Вправа 6



- Скласти ігрову програму Відгадай столицю. Занести назви країн і їхніх столиць до двох списків. Програма має по чергово запитувати назву столиці і перевіряти правильність відповіді, а після трьох помилок припиняти роботу, повідомляючи результат.

У Python IDLE виберіть команду File → New File.

- 1) Створіть список назв країн і список їхніх столиць:  
`kr = ['Франція', 'ОАЕ', 'Нідерланди', 'Туреччина', 'Швейцарія']`  
`st = ['Париж', 'Абу-Дабі', 'Амстердам', 'Анкара', 'Берн']`  
 Створіть змінну-лічильник помилок:  $k = 0$ . Надайте значення 0 змінній  $i$  для збереження індексу поточного елемента.
- 2) Запитання пропонуються доти, поки не закінчився список країн ( $i < 5$ ) і користувач не припустився третьої помилки ( $k < 3$ ).

```
while k < 3 and i < len(kr):
    print('Країна: ', kr[i])
    s = input('Назвіть столицю країни: ')
    if s == st[i]: print('Правильно!')
    else:
        k = k + 1
        print('Помилка!')
    i = i + 1
```

- 3) Напишіть оператор розгалуження, в якому залежно від значення  $k$  виводиться відповідне повідомлення:  
`if(k == 0): print('Вітаю! Ви – знавець географії!')`  
`elif(k == 1): print('Непогано! Ви можете вирушати в подорож!')`  
`elif(k == 2): print('Почитайте підручник географії!')`  
`else: print('Вам краще сидіти дома!')`



- 4) Додайте до списків назви ще декількох країн і столиць. Внесіть до програмного коду відповідні зміни.
- 5) Збережіть програму у файлі з іменем Vprava7. Випробуйте програму, проаналізуйте результати її роботи.

## Комп'ютерне тестування

Виконайте тестове завдання 6 із автоматичною перевіркою результату.



## § 7. Опрацювання елементів списку

Уявіть, що вам доручили проаналізувати дані про зріст учнів класу. Визначення найбільшого і найменшого зросту, середнього значення зросту — потребує опрацювання сукупностей значень, тобто списків.

Розглянемо основні типи задач опрацювання елементів списку:

- задачі на зміну значень елементів списку;
- задачі на пошук елемента із заданою властивістю;
- задачі на знаходження суми (добутку) елементів списку.

### Перебір елементів списку

Опрацювання елементів списку виконується в циклі, де параметр циклу послідовно набуває значень елементів списку або їх індексів. Розглянемо це на прикладі виведення списку на екран:

1	Послідовно звертатися до елементів за їх індексами. У такому випадку потрібно знати кількість елементів у списку (функція <code>len()</code> )	<code>a = [2, 5, 14, 62, -5]</code> <code>for i in range(len(a)):</code> <code>print(a[i])</code>
2	Перебрати всі наявні в списку елементи	<code>for item in a:</code> <code>print(item)</code>

Між цими способами є суттєва відмінність: лише звернення за індексом дозволяє *змінювати* значення елементів списку (див. далі).

### Опрацювання елементів списку

Розглянемо основні типи задач опрацювання списків.

#### ► Заміна значень усіх елементів списку

*Алгоритм розв'язування задачі:* послідовно перебрати всі елементи списку й значення чергового елемента змінити за заданим правилом.

- 1 Подвоїмо значення елементів списку `a`:  
`for i in range(len(a)): a[i] = a[i]*2`

#### ► Заміна значень елементів списку з перевіркою умови

*Алгоритм розв'язування задачі:* послідовно перебрати всі елементи списку; якщо черговий елемент або його індекс відповідає деякій умові, його значення змінити за певним правилом.

2 Поділимо на 2 парні елементи списку  $a = [2, 5, 14, 62, -5]$ :

```
for i in range(len(a)):
    if a[i]%2 == 0:
        a[i] = a[i]/2
print(a) # Буде надруковано [1, 5, 7, 31, -5]
```

3 Замінімо нулем елементи списку  $a = [2, 5, 14, 62, -5]$ , що мають парні індекси:

```
for i in range(len(a)):
    if i%2 == 0:
        a[i] = 0
print(a) # Буде надруковано [0, 5, 0, 62, 0]
```

### ► Визначення кількості елементів списку із заданою властивістю

*Алгоритм розв'язування задачі:* присвоїти змінній-лічильнику значення 0; перебрати всі елементи списку; якщо черговий елемент відповідає деякій умові, то значення лічильника збільшити на 1.

4 Визначимо кількість додатних елементів списку  $a = [2, 5, 14, 62, -5]$ :

```
k = 0; # Лічильник елементів із заданою властивістю
for item in a:
    if item > 0:
        k = k + 1
print('k = ', k) # Буде надруковано k = 4
```

### ► Знаходження суми (добутку) елементів списку

*Алгоритм розв'язування задачі:* зробити початкове значення суми таким, що дорівнює 0 (або значення добутку дорівнює 1); перебрати всі елементи списку і значення кожного елемента додати до значення суми (або помножити на нього значення добутку).

5 Знайдемо добуток всіх елементів списку  $a = [2, 5, 1, 6, -5]$ :

```
p = 1; # Змінна для збереження значення добутку
for item in a:
    p = p * item
print('p = ', p) # Буде надруковано p = -300
```

### ► Знаходження суми (добутку) елементів із заданою властивістю

*Алгоритм розв'язування задачі:* зробити початкове значення суми таким, що дорівнює 0 (або значення добутку дорівнює 1); перебрати

всі елементи списку; якщо черговий елемент відповідає деякій умові, додати його значення до значення суми (або помножити на нього значення добутку).

7 Знайдемо суму додатних елементів списку  $a = [2, -3, 1, 6, -5]$ :

```
suma = 0;           # Змінна для збереження значення суми
for item in a:
    if item > 0: suma = suma + item
print('suma = ', suma)  # Буде надруковано suma = 9
```

Деякі типові алгоритми опрацювання списків реалізовані у вигляді методів класу `list` (про це далі).

### Питання для самоперевірки



1. Поясніть відмінності між двома способами перебору елементів списку:

а) `for i in range(len(a)): print(a[i])`

б) `for i in a: print(i)`

2. Поясніть алгоритм заміни значень елементів списку за певним правилом.



3. Поясніть алгоритм визначення кількості елементів списку із заданою властивістю.

4. Поясніть алгоритм обчислення суми елементів списку.

5. Дано список  $a = [2, -7, -1, 8, 4]$ . Що буде надруковано після виконання наведених фрагментів коду?

а) `for i in range(5):`

`a[i] = a[i]*2`

`print(a)`

б) `k = 0`

`for item in a:`

`if(item > 0): k = k+1`

`print(k)`

в) `for i in range(5):`

`if(a[i]<0): a[i] = 0`

`print(a)`

г) `m = a[0]`

`for i in range(1, 5):`

`if(a[i]<m): m = a[i]`

`print(m)`



6. На складі зберігається 20 видів товарів. До списку  $a$  занесено кількість одиниць кожного товару, до списку  $b$  — ціни цих товарів. Обчисліть загальну вартість товарів на складі.

## Вправа 7



► Розробити програму опрацювання списку значень середньодобової температури за один тиждень місяця. Визначити, скільки разів за тиждень значення температури змінювало знак. У Python IDLE виберіть команду File → New File.

- 1) Завантажте модуль для генерації випадкових чисел:  
`from random import*`
- 2) Створіть порожній список `t`.
- 3) Заповніть список випадковими значеннями в діапазоні від `-10` до `+5`. Виведіть утворений список у консоль.
- 4) Створіть змінну `k` для збереження кількості змін знака температури.
- 5) Запишіть оператор циклу для перегляду елементів списку з індексами від `0` до `5`.

Якщо добуток поточного і наступного елементів від'ємний, потрібно вивести індекси цих елементів і збільшити значення лічильника `k`:

```
k = 0
```

```
for i in range(6):
```

```
    if(t[i]*t[i+1]<0):
```

```
        print(i, ' ', i+1)
```

```
        k += 1
```

- 6) Запишіть оператор для виведення значення `k`. Збережіть програму у файлі з іменем `Vprava6`.

Випробуйте програму. Внесіть зміни до програми, щоб окремо підраховувалось число днів, коли середньодобова температура була рівна `0`.



## Комп'ютерне тестування

Виконайте тестове завдання 7 із автоматичною перевіркою результату.



## § 8. Функції та методи опрацювання списків

Для розв'язування задач опрацювання сукупностей даних Python має вбудовані функції і методи списків, застосування яких допомагає швидше розв'язувати складні завдання. Ви знайомі з методом `append()`.

### Методи списків

#### Синтаксис виклику методів:

<назва об'єкта>.<назва методу>([параметри])

Методи списків змінюють список, тому результат не потрібно зберігати в іншу змінну. Розглянемо таблицю (тут у першій колонці список, для якого викликається метод, позначено `list`, а в третій наведено приклади для списку `a`, причому спочатку `a = [1, 5, 7, 5, 31, -5]`):

Метод	Призначення	Приклад
<code>list.insert(i, x)</code>	Вставляє на <i>i</i> -те місце елемент зі значенням <i>x</i>	<pre>&gt;&gt;&gt; a.insert(2, 78) &gt;&gt;&gt; a [1, 5, 78, 7, 5, 31, -5]</pre>
<code>list.pop(i)</code>	Видаляє зі списку елемент з індексом <i>i</i> і повертає його значення.	<pre>&gt;&gt;&gt; a.pop(4) 31 &gt;&gt;&gt; a [1, 5, 7, 5, -5]</pre>
<code>list.remove(x)</code>	Видаляє зі списку перший елемент зі значенням <i>x</i>	<pre>&gt;&gt;&gt; a.remove(5) &gt;&gt;&gt; a [1, 7, 5, 31, -5]</pre>
<code>list.index(x)</code>	Повертає індекс першого елемента зі значенням <i>x</i>	<pre>&gt;&gt;&gt; a.index(7) 2</pre>
<code>list.count(x)</code>	Повертає кількість елементів зі значенням <i>x</i>	<pre>&gt;&gt;&gt; a.count(1) 3</pre>
<code>list.sort()</code> <code>list.sort(reverse = True)</code>	Упорядковує список за зростанням (або за спаданням, якщо параметр <code>reverse = True</code> )	<pre>&gt;&gt;&gt; a.sort() &gt;&gt;&gt; a [-5, 1, 5, 5, 7, 31]</pre>
<code>list.reverse()</code>	Переставляє елементи списку у зворотному порядку	<pre>&gt;&gt;&gt; a.reverse() &gt;&gt;&gt; a [-5, 31, 5, 7, 5, 1]</pre>

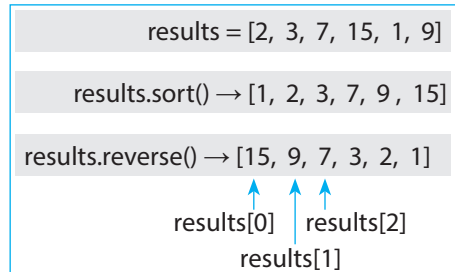
1 Видалимо зі списку `letters` всі елементи 'a':

```
letters = ['a', 'd', 'a', 'a', 'b']
while 'a' in letters:
    letters.remove('a')
print(letters)
```

Буде виведено: ['d', 'b'].

2 Надрукуємо три найбільших елементи списку `results`:

```
results = [2, 3, 7, 15, 1, 9]
results.sort()
results.reverse()
for i in range(3):
    print(results[i])
```



Списки зручно використовувати для опрацювання даних інших типів, наприклад рядків. Деякі призначені для цього функції та методи наведено в таблиці:

Метод	Призначення	Приклад
<code>s.split</code> (символ)	Розбиття рядка <code>s</code> на список рядків за заданим символом-роздільником	<pre>&gt;&gt;&gt; s = 'мама мила раму' &gt;&gt;&gt; a = s.split(' ') &gt;&gt;&gt; a ['мама', 'мила', 'раму']</pre>
<code>list</code> (рядок)	Розбиття рядка на список символів	<pre>&gt;&gt;&gt; b = list(s) &gt;&gt;&gt; b ['м', 'а', 'м', 'а', ' ', 'м', 'и', 'л', 'а', ' ', 'р', 'а', 'м', 'у']</pre>
<code>c.join</code> (список)	Об'єднання елементів списку в рядок із роздільником <code>c</code> між елементами	<pre>&gt;&gt;&gt; a = ['q', 'w', 'e', 'r', 'tttt', 'y'] &gt;&gt;&gt; ' '.join(a) 'q-w-e-r-tttt-y'</pre>

3 Запишемо рядок `c` у зворотному порядку:

```
c = 'мама мила раму'
a = list(c)           # Створення списку a з символів рядка c
a.reverse()          # Перестановка елементів списку a у зворотному порядку
c1 = ''.join(a)      # Об'єднання списку в рядок із порожнім роздільником
print(c1)            # Надруковано 'умар алим амам'
```



## Функції для роботи зі списками

Функції для роботи зі списками, на відміну від методів, не змінюють сам список, а повертають певне значення. З функціями `len()` і `list()` ви вже знайомі. Деякі корисні функції описано в таблиці:

Функція	Призначення	Приклад
<code>max(list)</code>	Повертає найбільше значення елемента	<pre>&gt;&gt;&gt; a = [1, 5, 7, 31, -5] &gt;&gt;&gt; max(a) 31</pre>
<code>min(list)</code>	Повертає найменше значення елемента	<pre>&gt;&gt;&gt; min(a) -5</pre>
<code>sum(list)</code>	Повертає значення суми елементів	<pre>&gt;&gt;&gt; sum(a) 39</pre>

4 Знайдемо індекс найбільшого елемента у списку  $a = [1, 5, 7, 31, -5]$ :

```
>>> a = [1, 5, 7, 31, -5]
>>> n = a.index(max(a))           # n = 3
```

Видалити будь-який об'єкт, зокрема й елемент списку, дозволяє оператор `del`:

```
>>> a = ['зима', 'весна', 'літо', 'осінь']
>>> del(a[2])
>>> a
['зима', 'весна', 'осінь']
```

Комбінуючи функції та методи роботи зі списками, ми можемо, як із цеглинок, скласти алгоритм розв'язування складної задачі.

### Питання для самоперевірки



1. Назвіть два способи видалення елемента зі списку.
2. Назвіть два способи перевірити, чи є в списку певний елемент.
3. Наведіть приклад завдання з реального життя, для розв'язання якого необхідно використовувати сортування.



4. Дано список  $a = [2, -7, -1, 8, 2, 4]$ . Що буде надруковано після виконання наведених фрагментів програми?

a) <code>m = min(a)</code>	б) <code>a.sort()</code>
<code>n = a.index(m)</code>	<code>print(a)</code>
<code>del(a[n])</code>	<code>s = sum(a[1:5])</code>
<code>print(a)</code>	<code>print(s)</code>

```

в) a.sort(reverse = True)
   for i in range(3):
       print(a[i])
г) k = a.count(2)
   print(k)
   
```

```

г) n = a.index(max(a))
   c = a[0]
   a[0] = a[n]
   a[n] = c
   print(a)
   
```

5. Складіть програму для заміни найбільшого елемента списку з 10 чисел нулем.
6. У фігурному катанні загальна оцінка за виступ обчислюється за такими правилами: відкидаються найвища і найнижча оцінки, а з решти обчислюється середнє арифметичне. Складіть програму для визначення оцінки за цими правилами.



### Вправа 8



- Скласти програму опрацювання списку. Дано список оцінок, отриманих учнями за проходження тесту.

Складіть програму для визначення:

- середнього арифметичного оцінок;
  - кількості учнів, які отримали оцінку, вищу за середній бал. У Python IDLE виберіть команду File → New File.
- Створіть список оцінок:  
`a = [11, 8, 6, 8, 12, 9, 7, 6]`
  - Обчисліть  $s$  — суму елементів списку  $a$ .
  - Обчисліть середнє арифметичне елементів списку  $a$ , збережіть це значення в змінній  $av$ .
  - Додайте оператор виведення результату, округленого до цілого за допомогою функції `round()`:  
`print (round(av))`
- 5) Запишіть програмний код для визначення кількості елементів списку, більших за значення середнього арифметичного.
- 6) Збережіть програму у файлі з іменем `Vprava8`. Випробуйте програму, проаналізуйте результати її роботи. Доповніть програму командами для введення списку з клавіатури й випробуйте її.



### Комп'ютерне тестування

Виконайте тестове завдання 8 із автоматичною перевіркою результату.



## § 9. Створення інтерфейсу користувача

Ви вже маєте досвід написання програм із графічним інтерфейсом із використанням бібліотеки `tkinter` і знаєте, що створення графічних компонентів (віджетів) потребує достатньо об'ємного коду. Значно швидше створити зручний графічний інтерфейс із використанням вікон виведення повідомлень, кнопочових меню, діалогових вікон та зображень допоможе бібліотека `EasyGUI`.

Для доступу до функцій бібліотеки `EasyGUI` її слід встановити на комп'ютер із офіційного сайту [easygui.sourceforge.net](http://easygui.sourceforge.net)

### Перші кроки з EasyGUI

Перед викликом функцій із бібліотеки `EasyGUI` на початку програмного коду слід імпортувати модуль `easygui`:

```
from easygui import*
```

#### ► Вікно повідомлень

Функція `msgbox()` модуля `easygui` створює діалогове вікно з повідомленням і єдиною кнопкою.

#### Формат функції:

```
msgbox(message[, title, button, image])
```

де `message` — текст повідомлення; `title` — заголовок діалогового вікна, `button` — напис на кнопці, `image` — ім'я графічного файлу (у форматі `.gif`) для додавання зображення.

- 1 Створимо діалогове вікно для привітання (рис. 9.1).  

```
mmsgbox('Вітаю!', 'Привітання', 'Привіт!', 'python.gif')
```

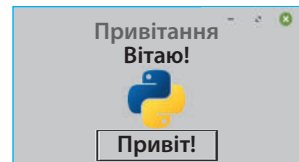


Рис. 9.1

Обов'язковим є тільки параметр `message`.

Якщо потрібно пропустити певний параметр, то значення решти параметрів задавайте за допомогою присвоєнь `параметр = значення`.

- 2 Створимо вітальне діалогове вікно з малюнком і кнопкою ОК, не вказуючи заголовка.  

```
mmsgbox('Вітаю!', image = 'python.gif')
```

## Опрацювання дій користувача

Дослідимо, як віджети з бібліотеки EasyGUI реагують на дії користувача. Введемо у вікні IDLE команди:

```
>>> from easygui import*
>>> messagebox("Вітаю!")
```

і отримаємо реакцію: 'OK'.

Рядком 'OK' інтерпретатор Python повідомляє, що користувач натиснув кнопку ОК.

Функції модуля `easygui` завжди підтримують зворотний зв'язок, інформуючи про дії користувача — яку кнопку клацнули, які дані уведено тощо. Інформацію про дії користувача можна зберегти в змінній для подальшого опрацювання.

Збережемо інформацію у змінній `my_click`:

```
>>> my_click = messagebox("Вітаю!")
```

Клацнемо кнопку ОК, щоб закрити діалогове вікно, й у вікні IDLE запитаємо значення змінної `my_click`:

```
>>> my_click
'OK'
```

Ми переконалися, що дії користувача — клацання кнопки ОК — зафіксовані у змінній `my_click`. Це дозволяє програмувати реакцію на дії користувача під час використання складніших функцій із бібліотеки EasyGUI (див. далі).

## Діалогове вікно з набором кнопок

Функція `buttonbox()` модуля `easygui` створює діалогове вікно з повідомленням і набором кнопок.

**Формат функції:**

```
buttonbox(message, title, choices)
```

де: `message` — текст повідомлення, `title` — заголовок діалогового вікна, `choices` — список заголовків кнопок.

**3** Створимо програму для відповіді на загадку:

```
from easygui import*
s = 'Фарбоване коромисло через річку зависло. Що це? '
puzzle = buttonbox(s, 'Загадка', ['Міст', 'Веселка', 'Електродроти'])
if (puzzle == 'Веселка'): messagebox ('Молодець! Це ' + puzzle, 'Відповідь')
else: messagebox ('Hi, це не ' + puzzle, 'Відповідь')
```

Після запуску програми побачимо вікно з трьома кнопками (рис. 9.2), після натискання однієї з них — вікно з коментарем.

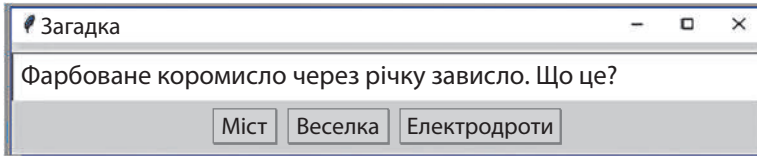


Рис. 9.2

Проаналізуємо хід виконання програми. Замість уведення значення змінної `puzzle` за допомогою оператора `input()` програма пропонує клацнути кнопку. Це клацання сприймається як введення даних. Текст заголовка кнопки (значення типу `string`) зберігається в змінній `puzzle`.

## Вікно `choicebox`

Коли потрібно зробити вибір із двох-трьох варіантів, зручно використовувати функцію `buttonbox()`. Але якщо варіантів більше, вікно буде перевантажене кнопками, тому краще скористатись функцією `choicebox()`. Вікно, що створене цією функцією, містить список варіантів. Користувач може вибрати один із них подвійним клацанням або клацнувши його у списку і натиснувши кнопку ОК.

### Формат функції `choicebox()`:

`choicebox (message, title, choices)`

де `message` — текст повідомлення; `title` — заголовок діалогового вікна, `choices` — список варіантів.

- 4 Виберемо персонажа для гри за допомогою діалогового вікна зі списком варіантів. Присвоїмо змінній вибраний елемент списку:

```
from easygui import*
msg = "Виберіть персонажа гри"
title = "Вибір героя"
choices = ["Птах", "Заєць", "Злюка", "Кажан"] # Список варіантів
choice = choicebox(msg, title, choices) # Присвоєння змінній choice
if (choice == "Птах"): image = "1.gif" # вибраного елемента списку
elif (choice == "Заєць"): image = "2.gif"
elif (choice == "Кажан"): image = "3.gif"
else: image = "4.gif"
msgbox ("Ваш герой " + choice, "Ваш вибір", "ОК", image)
```

Функція `choicebox()` виводить діалогове вікно зі списком (рис. 9.4). Після натискання кнопки ОК змінна `choice` набуває значення, вибраного в списку, яке й виводиться функцією `msgbox()` (рис. 9.5).

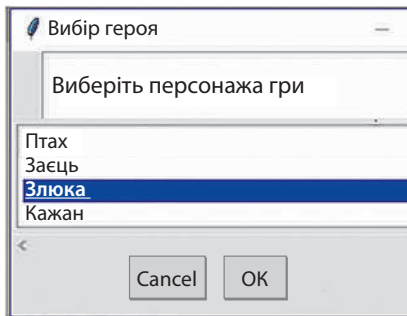


Рис. 9.4

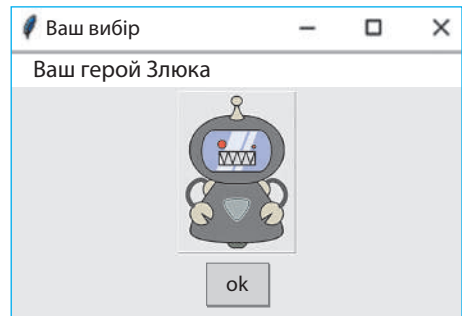


Рис. 9.5

## Введення даних

Розглянуті функції дозволяють користувачеві вибирати з обмеженої кількості запропонованих варіантів. А як надати можливість вводити власний варіант відповіді? Для цього в бібліотеці EasyGUI є функція `enterbox`, яка виводить вікно введення.

**Синтаксис виклику функції:**

змінна = `enterbox(message, title, default)`

де `message` — текст підказки, `title` — заголовок діалогового вікна, `default` — варіант, що пропонується за замовчуванням.

- 5 Створимо діалогове вікно без заголовка для введення відповіді, запропонуємо за замовчуванням варіант Ескімо (рис. 9.6).  
`favor = enterbox('Твоє улюблене морозиво?', default = 'Ескімо')`

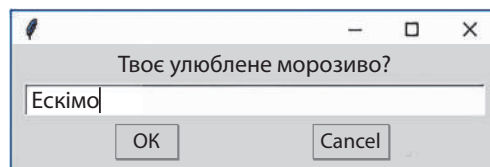


Рис. 9.6

При запуску програми ми побачимо, що в полі введення вже є варіант Ескімо. Замість нього можна ввести іншу відповідь, але, якщо нас влаштовує варіант Ескімо, ми просто клацнемо кнопку ОК.

Значення, отримане за допомогою функції `enterbox`, належить до рядкового типу. Якщо потрібно опрацювати числове значення, отриманий рядок слід перетворити на число за допомогою відомих вам функцій `int()` або `float()`.

Як бачимо, можливостей бібліотеки `EasyGUI` достатньо, щоб зробити, наприклад, програму для перевірки знань з інформатики.

### Питання для самоперевірки



1. Як з модуля `easygui` викликати вікно з повідомленням і єдиною кнопкою?
2. Як виглядає діалогове вікно, створене з такими параметрами: `choice = buttonbox('Виберіть рівень', 'Гра', ['1', '2', '3'])`?
3. Як виглядає діалогове вікно, створене з такими параметрами: `choice = choicebox('Скільки гравців?', 'Гра', ['1', '2', '3'])`?
4. Як засобами модуля `easygui` ввести рядкові дані?
5. Як засобами модуля `easygui` ввести ціле число?
6. Поясніть, як у програмі з прикладу 4 вибирається правильний малюнок.



### Вправа 9



- Скласти програму Калькулятор для обчислення результату виконання арифметичних дій над двома введеними числами. У Python IDLE виберіть команду `File` → `New File`.
- 1) Запишіть оператор для завантаження модуля `easygui`:  
`from easygui import*`
  - 2) Уведення чисел і обчислення виразу відбуватимуться в циклі `while(rep)`, тобто поки значення змінної `rep` дорівнює `True`. Задайте початкове значення змінної `rep`:  
`rep = True`
  - 3) У тілі циклу `while(rep)`: введіть два числа за допомогою діалогового вікна `enterbox()` (рис. 9.7):  
`while(rep):`  
    `n1 = float(enterbox('Введіть перше число: '))`  
    `n2 = float(enterbox('Введіть друге число: '))`
  - 4) Реалізуйте вибір арифметичної дії за допомогою відповідних кнопок діалогового вікна `buttonbox()` (рис. 9.8).

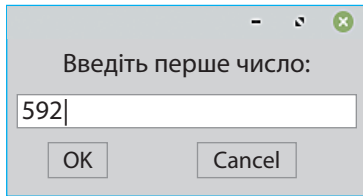


Рис. 9.7

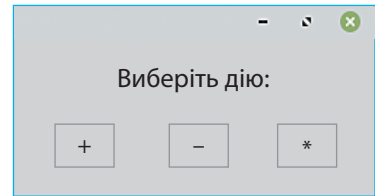


Рис. 9.8

Вибраний елемент списку присвойте змінній operation:

```
ch = ['+', '-', '*']
```

```
operation = buttonbox('Виберіть дію:', choices = ch)
```

Проаналізуйте значення змінної operation і організуйте виконання потрібної арифметичної дії в операторі if ... elif ... else і виведення результату (рис. 9.9):

```
if operation == '+':
```

```
    msgbox('Результат ' + str(n1+n2))
```

```
elif operation == '-':
```

```
    msgbox('Результат ' + str(n1-n2))
```

```
else:
```

```
    msgbox('Результат ' + str(n1*n2))
```

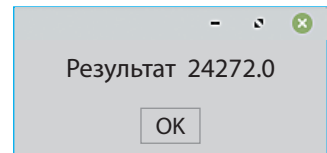


Рис. 9.9

- 5) Щоб запитати в користувача, чи слід повторити обчислення, створіть діалогове вікно з кнопками Так, Ні; вибраний елемент списку присвойте змінній povtor:

```
povtor = buttonbox('Ще раз? ', choices = ['Так', 'Ні'])
```

Якщо змінна povtor набуває значення Ні, змінна rep отримує значення False, і цикл while(rep): припиняє роботу:

```
if povtor == 'Ні':
```

```
    rep = False
```



- 6) Додайте до списку заголовків кнопок діалогового вікна для вибору дії кнопки «//», «%» і запрограмуйте відповідні обчислення. Запропонуйте власне розширення можливостей калькулятора і реалізуйте його. Збережіть програму в файлі з іменем Vprava9.

## Комп'ютерне тестування

Виконайте тестове завдання 9 із автоматичною перевіркою результату.







## Практична робота 2

### Створення програм для опрацювання списків

**Завдання:** скласти програму для розв'язування задачі.

*Задача.* На автобазі 10 вантажних автомобілів «КрАЗ». Список [12, 8, 13, 9, 14, 10, 15, 16, 11, 18] містить значення маси вантажів (у тоннах), перевезених кожним автомобілем за добу.

Визначити: а) загальну масу вантажу, перевезеного всіма автомобілями; б) номер у списку автомобіля, що перевіз вантаж найбільшої маси; в) кількість автомобілів, що порушили обмеження (15 т) на масу вантажу.

**Обладнання:** комп'ютер із середовищем програмування Python.

#### Хід роботи

*Під час роботи з комп'ютером дотримуйтеся правил безпеки.*

У Python IDLE виберіть команду File → New File.

- ▶ 1. Створіть змінну  $v$  типу список і надайте цій змінній значення, що вказані в умові задачі.
- ▶ 2. Запишіть оператор присвоєння змінній  $s$  значення суми елементів списку  $v$  (скористайтеся функцією `sum(list)`). Запишіть оператор для виведення значення змінної  $s$ .
- ▶ 3. Запишіть оператор присвоєння змінній  $m$  значення найбільшого елемента списку (скористайтеся функцією `max(list)`). Запишіть оператор для виведення значення змінної  $m$ .
- ▶ 4. Запишіть оператор присвоєння змінній  $n$  індексу найбільшого елемента списку (скористайтеся методом `list.index()`). Запишіть оператор для виведення значення змінної  $n$ .
- ▶ 5. Для запобігання пошкодженню дорожнього покриття маса вантажу, що перевозить автомобіль, не має перевищувати 15 тонн. Скористайтеся алгоритмом визначення кількості елементів із заданою властивістю (див. § 7), щоб з'ясувати, скільки автомобілів перевищили норму.
- ▶ 6. Збережіть файл з іменем Практична2.

**Зробіть висновок:** як опрацьовувати числові дані, що зберігаються у списках.

## § 10. Графічні методи модуля tkinter

У мові Python існують різні способи роботи з графікою. Так, ви вже знайомі з черепашачою графікою, що реалізується засобами модуля `turtle` і дає змогу створювати яскраві зображення в окремому графічному вікні.

Графічні методи модуля `tinker` дозволяють виводити на екран зображення з файлів (наприклад, фотографії), а також створювати малюнки в програмі з окремих геометричних фігур.

Розглянемо, як використовуються засоби модуля `tinker`.

### Полотно для малювання

У `tinker` зображення створюється в межах полотна — об'єкта класу `Canvas`, який теж входить до модуля `tinker`. Тому, перш ніж записувати оператори для малювання, потрібно створити полотно.

**Синтаксис створення об'єкта класу `Canvas`:**

```
<змінна> = Canvas(<батьківський віджет>, width = <ширина>,
height = <висота>)
```

де `width` — ширина полотна; `height` — його висота, виражені в пікселях.

**1** Створимо полотно розміром  $500 \times 500$  пікселів:

```
from tkinter import*
root = Tk()
canvas = Canvas(root, width = 500, height = 500)
canvas.pack()
```

Для задавання положення точок на полотні використовують координати. Будь-яка точка може бути задана парою чисел  $(X, Y)$ , де  $X$  — відстань від точки до лівого краю полотна,  $Y$  — відстань від точки до верхнього краю полотна (рис. 10.1).

Колір полотна — світло-сірий, за бажання можна змінити фон для майбутнього малюнка за допомогою оператора

```
canvas.config(bg = 'blue') # blue замінить назвою потрібного кольору
```

Об'єкт `Canvas` має низку методів, призначених для малювання точок, ліній, прямокутників, кіл, дуг, виведення тексту.

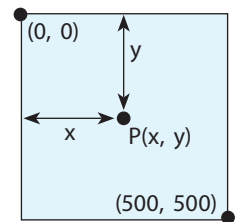


Рис. 10.1

## Методи для малювання фігур

Розглянемо використання графічних методів полотна Canvas.

### ▶ Відрізок

Метод `create_line` креслить відрізок від точки з координатами  $(x_1, y_1)$  до точки з координатами  $(x_2, y_2)$ :

```
canvas.create_line(x1, y1, x2, y2)
```

2 Намалюємо лінію від верхнього лівого кута полотна до правого нижнього кута:

```
canvas.create_line(0, 0, 500, 500)
```

### ▶ Прямокутник

Метод `create_rectangle` малює незаповнений прямокутник, у якого протилежні кути збігаються з точками  $(x_1, y_1)$  і  $(x_2, y_2)$ . Сторони прямокутника паралельні сторонам екрана:

```
canvas.create_rectangle(x1, y1, x2, y2)
```

3 Намалюємо квадрат зі стороною завдовжки 100 пікселів:

```
canvas.create_rectangle(220, 30, 319, 129)
```

### ▶ Еліпс

Метод `canvas.create_oval` малює еліпс, вписаний у прямокутник, який задано координатами двох протилежних вершин  $(x_1, y_1)$  і  $(x_2, y_2)$  (рис. 10.2):

```
canvas.create_oval(x1, y1, x2, y2)
```

4 Намалюємо коло діаметром 100 пікселів:

```
canvas.create_oval(220, 30, 319, 129);
```

### ▶ Багатокутник

Метод `canvas.create_polygon` малює багатокутник за координатами його вершин (рис. 10.3):

```
canvas.create_polygon(x1, y1, x2, y2, ..., xn, yn)
```

5 Накреслити трикутник:

```
points = [10, 210, 100, 210, 100, 310]
```

```
canvas.create_polygon(points)
```

За замовчуванням контури всіх об'єктів і заливка багатокутника мають чорний колір.

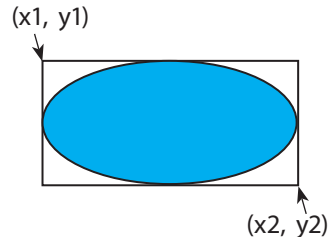


Рис. 10.2

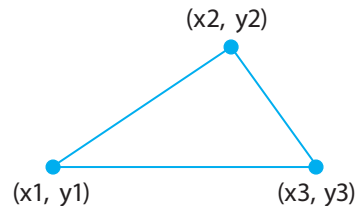


Рис. 10.3

Щоб створити кольорові рисунки, до переліку параметрів графічних методів потрібно додати параметри `fill` (колір заливки) і `outline` (колір лінії). Можна також збільшити товщину лінії, задавши значення параметра `width`.

! Щоб отримати багатокутник без заливки, слід обов'язково задати значення параметра `fill = ''`.

- 6 Намалюємо червоний круг із синім контуром завтовшки 3 пікселі:  
`canvas.create_oval(20, 30, 120, 130, fill = 'red', outline = 'blue', width = 3)`

## Виведення зображень із графічних файлів

Засобами `tkinter` можна вивести на полотно зображення з графічного файла, наприклад, фотографію.

Для цього потрібно:

- 1) графічні дані з файла завантажити до змінної класу `PhotoImage`, зазначивши, як параметр, ім'я файла (`file = '<ім'я файла>'`);
- 2) викликати метод `create_image`.

- 7 Виведемо на полотно рисунок `picture1.gif`, який міститься в кореневому каталозі диска D:  
`my_image = PhotoImage(file = 'd:\\picture1.gif')`  
`canvas.create_image(0, 0, anchor = NW, image = my_image)`

Параметр `anchor` визначає розташування рисунка на полотні. Значення `NW` вказує на верхній лівий кут полотна (рис. 10.4).

Засобами `tkinter` можна завантажувати тільки зображення формату GIF (файли з розширенням `.gif`).

! Методи `canvas.create_...` повертають числове значення — ідентифікатор створеного об'єкта.

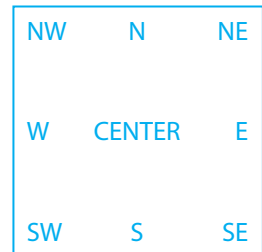


Рис. 10.4

Його зручно зберегти в змінній для подальших дій із об'єктом: `ball = canvas.create_oval(10, 10, 30, 30)` — змінна `ball` зберігатиме ідентифікатор круга.

Ми розглянули найпростіші способи роботи з об'єктами класу `Canvas` модуля `tkinter`. За допомогою графічних методів `Canvas` можна створювати рисунки як комбінацію геометричних фігур.

## Питання для самоперевірки



1. Опишіть систему координат полотна Canvas.
2. Опишіть алгоритм створення полотна Canvas.
3. Як вивести на полотно зображення з графічного файла?
4. Як побудувати круг зеленого кольору радіусом 100, центр якого збігається з центром полотна?
5. Запишіть фрагмент програми для побудови трикутника з вершинами в точках (100, 100), (150, 100), (80, 70); колір ліній — червоний, заливка відсутня.
6. Складіть програми для побудови наведених малюнків (рис. 10.5).

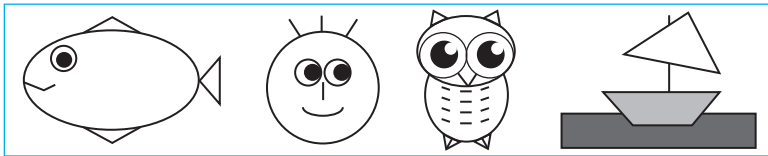


Рис. 10.5

## Вправа 10



- Скласти програму для заповнення полотна випадковими прямокутниками.

У Python IDLE виберіть команду File → New File.

- 1) Імпортуйте необхідні модулі:

```
from tkinter import*
from random import*
```

- 2) Створіть графічне вікно й додайте до нього полотно розміром 200 × 200.

- 3) Опишіть функцію користувача draw\_rectangle() для накреслення прямокутника, координатами вершин якого є випадкові числа. Функція draw\_rectangle() приймає аргументи width і height, і буде прямокутник з випадковими координатами лівого верхнього кута в діапазонах (0, width) і (0, height):

```
def draw_rectangle(width, height):
    x1 = randint(0, width)
    y1 = randint(0, height)
    x2 = randint(0, width)
    y2 = randint(0, height)
    canvas.create_rectangle(x1, y1, x2, y2)
```

- 4) З основної програми в циклі `for` 30 разів викличте функцію `draw_rectangle()`. Запустіть програму — полотно заповниться чорними прямокутниками з випадковими координатами вершин (рис. 10.6):

```
for x in range(0, 30):
    draw_rectangle(200, 200)
```

- 5) Змініть функцію `draw_rectangle`, додавши аргумент `fill_color` — колір заливки прямокутника. До переліку параметрів методу `create_rectangle` додайте параметр `fill` і присвойте йому значення `fill_color`. Після запуску полотно буде заповнене кольоровими прямокутниками (рис. 10.7):

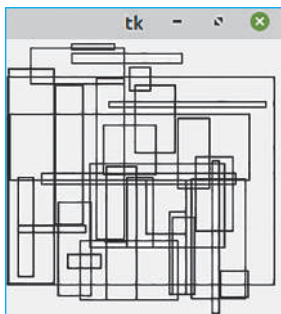


Рис. 10.6

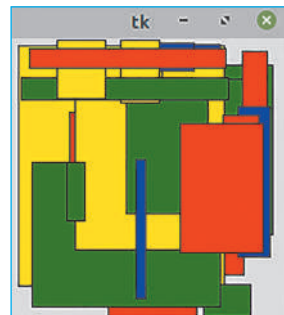


Рис. 10.7

```
def draw_rectangle(width, height, fill_color):
    x1 = randint(1, width)
    y1 = randint(1, height)
    x2 = randint(1, width)
    y2 = randint(1, height)
    canvas.create_rectangle(x1, y1, x2, y2, fill = fill_color)
c = ['red', 'green', 'blue', 'yellow'] # Вираз x%4 може набувати
for x in range(0, 30): # значень 0, 1, 2, 3, що відповідає
    draw_rectangle(200, 200, c[x%4]) # індексам елементів списку c
```

- 6) Змініть функцію `draw_rectangle` так, щоб малювалися кола з випадковими координатами.



### Комп'ютерне тестування

Виконайте тестове завдання 10 із автоматичною перевіркою результату.



## § 11. Створення анімації

У програмах можуть застосовуватися анімація, що створює ілюзію руху об'єктів, яка отримується швидким змінням зображень на екрані. Кожна із побудованих нами раніше на полотні фігур є окремим об'єктом, завдяки чому можна програмувати їх рух і взаємодію.

### Рух графічних об'єктів

Розглянемо методи, призначені для переміщення об'єкта полотном.

- Метод **canvas.move** переміщає графічний об'єкт, змінюючи значення його координат на задані величини:

```
canvas.move(item, x, y),
```

де *item* — ідентифікатор графічного об'єкта (див. § 10); *x*, *y* — значення зміщення по горизонталі та вертикалі (у пікселях).

- Метод **tk.update()** примусово оновлює зображення.
- Метод **time.sleep()** затримує виконання програми:

```
time.sleep(t),
```

де *t* — величина затримки (в секундах).

- 1 Організуємо горизонтальний рух круга по полотну. Перший створений об'єкт (круг) автоматично отримує ідентифікатор 1:

```
import time
from tkinter import*
tk = Tk()
canvas = Canvas(tk, width = 400, height = 400)
canvas.pack()
canvas.create_oval(10, 10, 30, 30, fill = 'red')
for x in range(0, 60):
    canvas.move(1, 5, 0)      # Переміщення об'єкта із ідентифікатором 1
    tk.update()
    time.sleep(0.05)
```

Блок коду в циклі `for` призначено для пересування круга по полотну. Зокрема, команда `canvas.move(1, 5, 0)` пересуне об'єкт із ідентифікатором 1 (це і є круг) на 5 пікселів праворуч і 0 пікселів по вертикалі. Для пересування ліворуч знадобилася б команда `canvas.move(1, -5, 0)`. Для руху по діагоналі слід змінювати одночасно координати *x* і *y*: `canvas.move(1, 5, 5)`.

## Програмування реакції об'єктів на події

Можна зробити так, щоб об'єкт рухався у відповідь на подію, що відбувається під час роботи програми — пересування миші, натискання певної клавіші. Для обробки події потрібно описати функцію, а потім повідомити tkinter, що ця функція має викликатися кожного разу, коли відбудеться зазначена подія.

Для встановлення зв'язку між подією та її обробником існує метод `bind_all()`:

```
canvas.bind_all(<подія>, функція),
```

де подія — текстова константа, системна назва події; функція — назва функції-обробника події.

2 Змусимо круг рухатися праворуч в разі натискання на клавішу зі стрілкою «→». Опишемо функцію — обробник події:

```
def move_oval(event):
    canvas.move(1, 5, 0)
```

Функція має єдиний аргумент `event`, через який передається інформація про подію. Об'єкт `event` має властивість `keysym`, яка набуває значення символу натиснутої клавіші. Якщо натиснуто клавішу «→», це означає, що відбулася подія '<KeyPress-Right>', і властивість `event.keysym` отримує значення 'Right' (з англ. вправо) (рис. 11.1).

```
def move_oval(event):
    ...
    event.keysym = 'Right'
    canvas.bind_all('<KeyPress-Right>', move_oval)
```

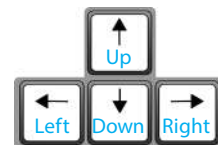


Рис. 11.1

Програмний код виглядатиме так:

```
from tkinter import*
def move_oval(event):
    canvas.move(1, 5, 0)
tk = Tk()
canvas = Canvas(tk, width = 400, height = 400)
canvas.pack()
canvas.create_oval(10, 10, 30, 30, fill = 'red')
canvas.bind_all('<KeyPress-Right>', move_oval)
```



Можна обробляти й інші події: '<Key>' — натискання якоїсь клавіші на клавіатурі; '<1>' — натискання лівої кнопки миші, '5' — натискання клавіші з цифрою 5 тощо.

- 3 Буде цікавіше змінювати напрямок руху круга, застосовуючи всі чотири клавіші зі стрілками. В тілі функції `move_oval()` запрограмуємо зміну координат об'єкта залежно від того, яке значення передане для аргумента `event`:

```
def move_oval(event):
    if event.keysym == 'Up': canvas.move(1, 0, -3)
    elif event.keysym == 'Down': canvas.move(1, 0, 3)
    elif event.keysym == 'Left': canvas.move(1, -3, 0)
    elif event.keysym == 'Right': canvas.move(1, 3, 0)
```

У функції `move_oval()` перевіряється, чи містить змінна `keysym` значення 'Up' (з англ. вгору). Якщо так, викликається метод `canvas.move(1, 0, -3)`, і круг зміщується вгору. Якщо ні, перевіряється, чи містить змінна `keysym` значення 'Down' (з англ. вниз), і, якщо так, робиться крок униз. Так само опрацьовується значення 'Left' (англ. вліво) і 'Right'. Таким чином, при натисканні на клавішу об'єкт пересувається в заданому напрямку.

Далі повідомимо `tkinter`, що функцію `move_oval()` треба застосовувати для обробки натискання будь-якої клавіші:

```
canvas.bind_all('<Key>', move_oval)
```

## Створення анімації з використанням графічних файлів

Використовуючи принцип анімації, можна запрограмувати рух об'єктів на полотні таким чином.

- 1) підготувати декілька графічних файлів із зображеннями фаз руху деякого об'єкта;
- 2) запрограмувати послідовне завантаження на полотно зображень із цих файлів;
- 3) між завантаженнями слід затримати виконання програми для того, щоб користувач встиг побачити чергове зображення.

- 4 Створимо анімацію Політ метелика із трьох кадрів. Розглянемо кроки створення анімації.

1. Підготуємо три файли з зображеннями фаз руху метелика (рис. 11.2). Імена файлів зберігатимуться у списку `anim`.



1.gif



2.gif



3.gif

Рис. 11.2

2. Анімацію реалізує така програма:

```
import time
from tkinter import *
def play():
    for i in range(3):
        canvas.create_image(0, 0, anchor = NW, image = frames[i])
        root.update()           # Оновлення полотна
        time.sleep(0.1)         # Пауза на 0.1 секунди
        root.after(0, play)     # Повторний виклик функції play()
root = Tk()
root.title('Політ метелика')
root.geometry('200x250')
canvas = Canvas(root, width = 200, height = 200)
canvas.place(x = 1, y = 1)
anim=['1.gif', '2.gif', '3.gif']           # Список імен файлів
frames = []                               # Читаємо дані зображень із файлів,..
for name in anim:                         # імена яких у списку anim,..
    frames.append(PhotolImage(file = name)) # у список frames
play()
root.mainloop()
```

Проаналізуємо опис функції `play()`, що призначена для змінування зображень. У циклі `for` по чергово викликом методу `create_image` на полотні створюється зображення на основі графічних даних чергового кадра зі списку `frames`.

Метод `time.sleep(0.1)` затримує зображення на 0.1 секунди. Метод `root.after` викликається для об'єкта `root` з аргументами: 0 мс; `play`, тобто зразу ж після завершення функція `play()` викликається повторно, і зміна зображень продовжується.

3. Зберігаємо файл з програмним кодом і випробовуємо програму.

Отже, засоби бібліотеки tkinter дозволяють створювати програми з анімаційними ефектами, а також додавати реакцію на дії користувача шляхом програмування обробки подій.

### Питання для самоперевірки



1. Які функції використовуються для переміщення об'єкта по полотну?
2. Як запрограмувати реакцію об'єкта на подію?
3. Поясніть призначення параметрів функції:  
`canvas.bind_all('<KeyPress-Up>', step)`
4. Опишіть кроки створення анімації шляхом послідовної зміни зображень.
5. Як зміниться Політ метелика (див. приклад 4), якщо з коду вилучити виклик методу `time.sleep(0.1)`?
6. Створіть програму, в якій по зоряному небу рухається космічний зонд Кассіні (рис. 11.3).



Рис. 11.3



### Вправа 11



- Створити програму, в якій реалізовано рух кульки по полотну. Об'єкт має змінювати положення у відповідь на натискання клавіш управління курсором, залишаючи слід на полотні (рис. 11.4).

У Python IDLE виберіть команду `File → New File`.

- 1) Підключіть модуль tkinter. Створіть вікно програми і додайте полотно розміром `400 × 400`:

```
from tkinter import*
```

```
tk = Tk()
```

```
canvas = Canvas(tk, width = 400, height = 400)
```

```
canvas.pack()
```

- 2) Намалуйте у лівому верхньому куті червоний круг. Ідентифікатор створеного круга зберігатиметься в змінній `ball`:  
`ball = canvas.create_oval(10, 10, 30, 30, fill = 'red')`

Слід кульки утворюватимуть відрізки, що сполучають центри послідовних положень круга.

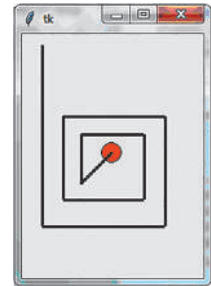


Рис. 11.4

- 3) Для обчислення координат центра круга (рис. 11.5) опишіть функцію `midpoint`:

```
def midpoint(pos):          # pos — список координат x1, y1, x2, y2
    return[(pos[0]+pos[2])/2, (pos[1]+pos[3])/2]
```

- 4) Опишіть функцію `move_oval()`. У ній будуть змінюватися координати круга залежно від того, яку клавішу натиснуто (параметр `event`), а також малюватиметься відрізок між центрами круга в початковому і кінцевому положеннях:

```
def move_oval (event):
    step = 3                # Величина зміщення
    P1=midpoint(canvas.coords(ball)) # Зберігаємо початкові координати
    if event.keysym == 'Up':
        canvas.move(ball, 0, -step)
    elif event.keysym == 'Down':
        canvas.move(ball, 0, step)
    elif event.keysym == 'Left':
        canvas.move(ball, -step, 0)
    elif event.keysym == 'Right':
        canvas.move(ball, step, 0)
    canvas.create_line(P1, midpoint(canvas.coords(ball)))
```

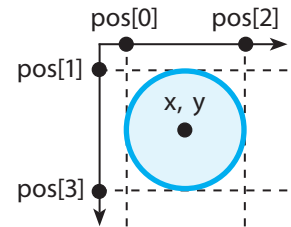


Рис. 11.5

- 5) Запишіть виклик функції `canvas.bind_all` для встановлення зв'язку між подіями «натискання на клавішу» і функцією `move_oval`, а також запуск головного циклу обробки подій: `canvas.bind_all('<Key>', move_oval)`  
`root.mainloop()`

Збережіть файл з іменем `Vprava11` і випробуйте програму.

- 6) Удоскональте програму: змініть товщину і колір сліду, який залишає круг; додайте діагональні переміщення за допомогою клавіш `A`, `S`, `Z`, `X`.



### Комп'ютерне тестування

Виконайте тестове завдання 11 із автоматичною перевіркою результату.



## § 12. Об'єктно-орієнтоване програмування

Як ви знаєте, мова Python є об'єктно-орієнтованою, тобто кожна величина є об'єктом певного класу. Добре спроектовані об'єкти — це «будівельні блоки», з яких легше будувати складні програми.

### Опис класів та створення об'єктів

У 6 класі ви дізналися, як створити свій власний клас (тип) об'єктів. Можна вважати, що клас — це своєрідна інструкція зі створення екземплярів. Створення об'єкта на основі класу називають створенням екземпляра класу.

Опис класу зручно розташовувати на початку коду програми.

**Синтаксис опису класу:**

```
class <назва класу>():
    <атрибут класу> = <значення>
    def __init__(self[, <інші параметри>]):
        [self.<атрибут екземпляра> = <значення>]
```

Атрибути класу мають однакове значення для всіх екземплярів класу, тоді як атрибути екземплярів — окреме значення для кожного екземпляра.

**Синтаксис звертання до атрибута класу:**

```
<назва класу>.<атрибут класу>
```

Метод `__init__` називається **конструктором класу**. Він автоматично виконується під час створення кожного екземпляра даного класу. Назва методу починається й закінчується двома символами підкреслення. У списку параметрів перший (його здебільшого позначають `self`) є особливим. Під час виклику в програмі методу для певного екземпляра класу, `self` отримує посилання саме на цей екземпляр. Таким чином метод «дізнається», для якого об'єкта його викликано.

**Синтаксис команди створення екземпляра класу:**

```
<змінна> = <назва класу>([<перелік значень атрибутів>])
```

Далі в програмі з'являється об'єкт, доступ до якого можна отримати за назвою змінної. Об'єкт отримує атрибути його класу.



**Атрибути класу** — це змінні, в яких зберігаються значення властивостей об'єктів.

- 1 Опишемо клас `Balloons`, який моделює повітряні кульки. Створимо об'єкт `balloon` — «червону кульку» заввишки 10 см (рис. 12.1).  
`class Balloons:`

```
def __init__(self, color, height, inflated):
    self.color = color
    self.height = height
    self.inflated = inflated
```

`balloon = Balloons('red', 10, True)`

Значення атрибутів об'єкта можна вивести на екран: `print (balloon.inflated)`

Можна змінити значення атрибута:  
`balloon.color = 'green'`

Значення атрибута можна присвоювати змінним:  
`mycolor = balloon.color`

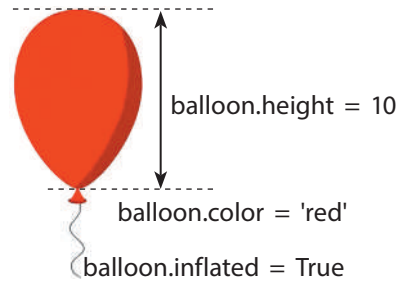


Рис. 12.1

- 2 Опишемо клас `Ball`, який представляє круги заданого радіуса, кольору і розташування на полотні. Забезпечимо підрахунок екземплярів класу. Створимо два екземпляри класу:

```
from tkinter import*
```

```
tk = Tk()
```

```
canvas = Canvas(tk, width = 500, height = 400)
```

```
canvas.pack()
```

```
class Ball:
```

```
    number = 0
```

```
    def __init__(self, canvas, x, y, color):
```

```
        self.canvas = canvas
```

```
        self.id = canvas.create_oval(x, y, x+20, y+20, fill = color)
```

```
        Ball.number = Ball.number+1
```

```
ball1 = Ball(canvas, 10, 10, 'red')
```

```
ball2 = Ball(canvas, 50, 50, 'blue')
```

- 3 Створимо список із 10 об'єктів класу `Ball` із випадковими координатами і кольорами, що вибираються зі списку `colors` (рис. 12.2).

```
list_ball = []
```

```
colors = ['red', 'orange', 'yellow', 'green', 'blue']
```

```
for i in range(10):
```

```
    x = randint(10, 400)           # Діапазон координати залежить
```

```
    y = randint(10, 350)         # від розмірів вікна
```

```
    list_ball.append(Ball(canvas, x, y, colors[i%5]))
```

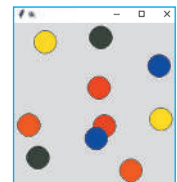


Рис. 12.2

## Методи класу

**Методи** — це дії, які можуть виконувати об'єкти даного класу. Під час виклику методу можуть змінюватися властивості об'єкта.

**Синтаксис заголовка методу класу:**

```
def <назва методу>(self[, <параметри>]:
```

Від звичайної функції метод класу відрізняється наявністю параметра `self`, опис методу класу розташовується в описі класу.

**Синтаксис виклику методу для конкретного об'єкта:**

```
<об'єкт>.<назва методу>(<параметри>)
```

4 Додамо до опису класу `Balloons` метод `boom()`, який виконується, якщо «кулька» «лопнула» (рис. 12.3).

```
class Balloons:
```

```
    def __init__(self, color, height, inflated):
```

```
        self.color = color
```

```
        self.height = height
```

```
        self.inflated = inflated
```

```
    def boom(self):
```

```
        print('Бабах! Пшшшш...')
```

```
        self.inflated = False
```

```
        self.height = 0
```

```
balloon = Balloons('red', 10, True)
```

```
balloon.boom() # Виклик методу boom() для об'єкта balloon
```

```
print (balloon.inflated) # Буде надруковано False
```



Рис. 12.3

## Принципи об'єктно-орієнтованого програмування

Поняття класу та об'єкта є основою об'єктно-орієнтованого програмування (ООП). Програма створюється як деяка сукупність об'єктів, які взаємодіють між собою. ООП базується на трьох основних принципах: інкапсуляція, успадкування та поліморфізм.

Розглянемо їх сутність. Ви вже знаєте, що списки дозволяють об'єднувати дані, а для об'єднання фрагментів коду в окремий блок застосовуються функції. Об'єкти ж дозволяють об'єднувати функції (методи класу) і дані (значення атрибутів) (рис. 12.4).

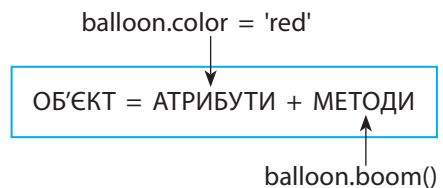


Рис. 12.4

**Інкапсуляція** — це об'єднання в межах класу певних даних і методів для роботи з ними. Наприклад, ми описали клас `Balloons`, що моделює повітряні кульки і містить дані про кульку (колір, висота, «надутість»), і методи (змінити значення атрибутів в разі проколювання кульки, збільшити розміри кульки тощо). При цьому клас стає новим типом даних у програмі, і ми можемо створювати об'єкти — змінні цього нового типу.

**Успадкування** — це можливість створення нового класу (нащадка) на базі наявного (базового).

Розглянемо такі поняття, як «людина» і «учень». Будь-яка людина має ім'я, прізвище, вік, інші загальні характеристики. Учень також має ім'я, прізвище, вік, але, крім цього, він має додаткові характеристики: навчальний заклад, клас.

З точки зору наслідування клас `Учень` є нащадком класу `Людина`. Таким чином, клас-нащадок має ті самі атрибути й методи, що й базовий клас. Але при цьому його можна розширити, додавши нові методи і атрибути. Це дозволяє не починати створення класу-нащадка «з нуля» — описувати атрибути й методи, успадковані від базового класу.

5 На основі базового класу `Animal` (тварина) створимо два класи-нащадки: `Dog` (собака) і `Cat` (кіт), які мають власні методи.

```
class Animal():
    def __init__(self, name):
        self.name = name
    def breathe(self):
        print(self.name, 'дихає')
    def eat_food(self, food):
        print(self.name, 'їсть', food)
class Dog(Animal):
    def guard(self):
        print(self.name, 'охороняє')
class Cat(Animal):
    def catchmouse(self):
        print(self.name, 'ловить мишу')
dog1 = Dog('Сірко')
cat1 = Cat('Мурчик')
cat1.breathe()
cat1.eat_food('рибку')
dog1.eat_food('кістку')
```

```
Мурчик дихає
Мурчик їсть рибку
Сірко їсть кістку
Сірко охороняє
Мурчик ловить мишу
>>> |
```

Об'єкти класів-нащадків можуть викликати метод базового класу



```
dog1.guard()           # Відправляємо собаку охороняти
cat1.catchmouse()    # Відправляємо kota на полювання
```

**Поліморфізм** — це різна поведінка методу в різних класах. Методи з однаковою назвою, описані в різних класах, можуть виконувати різні дії.

6 Опишемо класи Triangle (прямокутний трикутник) і Square (квадрат). Додати до описів класів метод для обчислення площі фігури.

```
class Triangle:
    def __init__(self, width, height):
        self.width = width
        self.height = height
    def getArea(self):
        area = self.width*self.height/2.0
        return area

class Square:
    def __init__(self, size):
        self.size = size
    def getArea(self):
        area = self.size*self.size
        return area

myTriangle = Triangle(4, 5)
mySquare = Square(7)
s_triangle = myTriangle.getArea()
print (s_triangle)
s_square = mySquare.getArea()
print (s_square)
```

# Виклик методу класу Triangle  
# Буде надруковано 10.0  
# Виклик методу класу Square  
# Буде надруковано 49

Метод getArea() присутній в обох класах, але площі фігур обчислюються за різними формулами

7 Додамо метод voice (голос) до класів Dog і Cat (див. приклад 5).

```
class Dog(Animal):
    def voice(self):
        print(self.name, 'Гав-гав')

class Cat (Animal):
    def voice(self):
        print(self.name, 'Няв-няв')

cat1.voice()
dog1.voice()
```

Мурчик Няв-няв  
Сірко Гав-гав

## Питання для самоперевірки



1. Поясніть різницю між класом і його екземпляром.
2. У чому різниця між атрибутом класу і атрибутом екземпляра? Якими атрибутами ви б описали об'єкт Автомобіль, якщо моделюється: а) робота автосалону; б) автоперегони?
3. Проаналізуйте фрагмент програмного коду. Які принципи ООП використано під час складання цього коду?



```
class Figura:
    def __init__(self, x):
        self.x = x
class Kolo(Figura):
    def S(self):
        return 3.14*x*x
class Kvadrat(Figura):
    def S(self):
        return x*x
kolo1 = Kolo(5)
kvadrat1 = Kvadrat(5)
```



4. Опишіть клас Car(), що містить атрибути кількість пального в баку на даний момент, пройдений шлях. Додайте до опису класу Car() метод для зміни значення пробігу.

## Вправа 12



- Запрограмувати рух об'єктів класу Ball (М'яч) по вертикалі з відбиттям від меж вікна.

У Python IDLE виберіть команду File → New File.

- 1) Імпортуйте необхідні модулі:

```
from tkinter import*
from random import*
from time import*
```

- 2) Створіть вікно програми і додайте полотно розміром 500 × 400.
- 3) Створіть клас Ball() з атрибутами  $x$ ,  $y$  (координати правого верхнього кута квадрата, в який вписано круг) і  $color$  (колір кульки).

```
class Ball:
    def __init__(self, canvas, x, y, color):
        self.canvas = canvas
        self.id = canvas.create_oval(x, y, x+50, y+50, fill = color)
        self.dy = 2 # Величина зміни координати у
```

- 4) Додайте до опису класу Ball метод, в якому після перевірки, чи дійшов об'єкт до межі вікна, і змінення за потреби напрямку руху, виконуватиметься переміщення в нове положення.

```
def ruh(self):
    pos = canvas.coords(self.id)
    if pos[1] > 348 or pos[1] < 2: self.dy = -1*self.dy
    canvas.move(self.id, 0, self.dy)
```

- 5) Створіть список із 10 об'єктів класу Ball із випадковими координатами:

```
list_ball = []
colors = ['red', 'orange', 'yellow', 'green', 'blue']
for i in range(10):
    x = randint(10, 400)
    y = randint(10, 350)
    list_ball.append(Ball(canvas, x, y, colors[i%5]))
```

- 6) Організуйте повторення виклику методу ruh() для кожного елемента списку list\_ball.

```
def play():
    for i in range(len(list_ball)):
        list_ball[i].ruh()
    tk.update() # Оновлення полотна
    tk.after(10, play) # Затримка виконання програми
play()
tk.mainloop()
```

Збережіть програмний код у файлі з іменем Vprava12 і виконайте програму.

Поекспериментуйте зі значеннями величин зміни координати  $y$ , радіуса круга.

### Комп'ютерне тестування

Виконайте тестове завдання 12 із автоматичною перевіркою результату.





## Практична робота 3

### Моделювання руху об'єктів

**Завдання:** скласти ігрову програму Ловець бульбашок.

На ігровому полотні через певні проміжки часу з'являються бульбашки (рис. 1). Гравець ловить бульбашки, клацаючи їх. Якщо кількість бульбашок перевищує допустиму, гра припиняється.

**Обладнання:** комп'ютер з середовищем програмування Python.

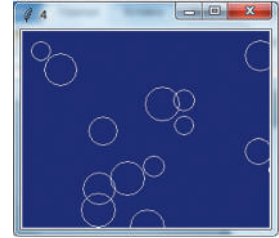


Рис. 1

#### Хід роботи

*Під час роботи з комп'ютером дотримуйтеся правил безпеки.*

У Python IDLE виберіть команду File → New File.

- 1. Завантажте модулі, що потрібні для реалізації руху об'єктів. Створіть вікно програми і додайте полотно розміром 800 × 500:

```
from tkinter import*
from random import*
tk = Tk()
canvas = Canvas(tk, width = 800, height = 500)
canvas.pack()
```

- 2. Створіть змінні для керування режимом гри:
- ```
maxB=10      # За такої кількості бульбашок на полотні гра зупиниться
genT=1000    # Проміжок часу між появою бульбашок
```
- Створіть клас Bub() — модель бульбашки (рис. 2). Екземпляр цього класу повинен:

- створюватись на полотні canvas вікна tk;
- мати випадкові координати  $x$  і  $y$  (у межах полотна);
- зразу ж після створення починати рух, зміщуючись на кожному кроці на  $dx$  і  $dy$  пікселів уздовж відповідних осей;
- відбиватись від країв полотна;
- після клацання — зникати.

Крім того, у класі мають підраховуватися кількості екземплярів-бульбашок, наявних і зловлених.

- ▶ На кроках 3–5 ми створимо клас Bubble, екземпляри якого поводитимуться на полотні canvas вікна tk саме так, як описано.

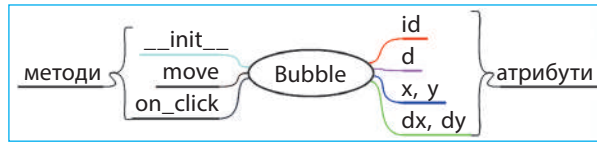


Рис. 2

- ▶ 3. Конструктор `__init__`:

```

class Bub():
    number = 0          # Лічильник бульбашок на полотні
    clicked = 0        # Лічильник клацнутих бульбашок
    def __init__(B):

        B.d = 40
        B.x = randint(2,755)
        B.y = randint(2,455)
        B.dx = randint(-5, 5)
        B.dy = randint(-5, 5)
        Bub.number = Bub.number+1

        B.id = canvas.create_oval(B.x, B.y, B.x+B.d, B.y+B.d, fill = 'blue')
        canvas.tag_bind(B.id, '<1>', B.on_click)    # Бульбашка має обробник
        B.move()    # клацання і починає рухатись після створення
  
```

- ▶ 4. Метод `move`, що забезпечує рух і відбивання кульки:

```

def move(B):
    canvas.move(B.id, B.dx, B.dy)
    tk.update()

    pos = canvas.coords(B.id)
    if len(pos)>0:
        if pos[0]<2 or pos[2]>798:    # Якщо кулька ще не зникла...
            B.dx = -1*B.dx          # ...то відбиваємо від країв
        if pos[1]<2 or pos[3]>498:
            B.dy = -1*B.dy
        tk.after(100, B.move)        # Продовження руху
  
```

- ▶ 5. Метод-обробник клацання бульбашки `on_click`:

```

def on_click(B,event):
    if Bub.number<maxB:
  
```

```

canvas.delete(B.id)
Bub.number = Bub.number-1
Bub.clicked = Bub.clicked+1
tk.title(Bub.clicked)

```

- ▶ **6.** Створіть функцію `play()`, яка забезпечить створення бульбашок у потрібні моменти, а також вчасне припинення гри:
 

```

def play():
    if Bub.number<maxB:
        Bub()
        tk.after(genT, play)
    else:
        tk.after_cancel(play)
        tk.title('Гру закінчено. Ваш результат: ' + str(Bub.clicked))
      
```
- ▶ **7.** В основній програмі запишіть команди виклику функції `play()`

```

play()
tk.mainloop()

```

 Збережіть програмний код у файлі з іменем `Практична3` і випробуйте програму. За потреби виправте помилки. Коли програма повністю запрацює, додайте до неї можливості, описані в наступних пунктах.
- ▶ **8.** Зараз діаметр всіх бульбашок однаковий. Зробіть так, щоб бульбашка під час створення отримувала випадкове значення діаметра в межах від 20 до 40 пікселів.
- ▶ **9.** По завершенні гри бульбашки, що залишились, продовжують рухатись, але не реагують на клацання. Зробіть так, щоб клацання в цей період спричиняло змінення кольору на жовтий ('Yellow').
- ▶ **10.** Кожна бульбашка під час клацання до загального рахунку додає 1 бал. Зробіть так, щоб швидкі бульбашки малого розміру додавали по 2 бали.
- ▶ **11.** Бульбашки з'являються через рівні проміжки часу, визначені змінною `genT` (початкове значення 1000 мс). Зробіть так, щоб цей інтервал поступово зменшувався.
- ▶ **12.** Запропонуйте власну ідею покращення програми і реалізуйте її.

**Зробіть висновок:** чому для моделювання руху графічних об'єктів доречно використовувати методи ООП.

# РОЗДІЛ 2

## СЛУЖБИ ІНТЕРНЕТУ



§ 13. Поштові служби інтернету

§ 14. Робота з електронною поштою

Практична робота 4. Створення електронної поштової скриньки

§ 15. Особливості електронного листування

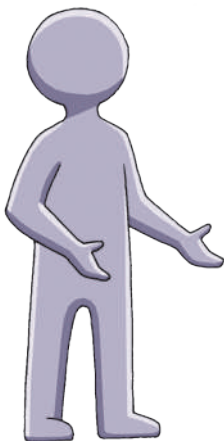
§ 16. Налаштування поштової скриньки

§ 17. Використання інтернет-ресурсів

Практична робота 5. Використання хмарних технологій

для колективного опрацювання документів та користування геосервісами

## ПОВТОРЮЄМО



Ви вже знайомі з поняттям *комп'ютерної мережі*, розрізняєте *локальну мережу*, що об'єднує комп'ютери, розташовані на невеликій відстані один від одного, і *глобальну мережу*, яка об'єднує мільйони комп'ютерів і локальних мереж. Найбільшою глобальною мережею є *інтернет*.

Вам відомо, як виконується пошук інформації за допомогою *пошукових систем*.

Ви знаєте, що сторінки сайтів (*вебсторінки*) є гіпертекстовими документами, тобто вони містять *гіперпосилання* — об'єкти (слова, малюнки тощо), що дають змогу перейти до іншого місця в документі або іншого документа.

Ви пам'ятаєте про можливі загрози під час роботи в інтернеті, дотримуетесь *основних правил безпечного користування* та критично оцінюєте інформацію.

1. Що таке локальна мережа; глобальна мережа?
2. Які можливості надає нам інтернет?
3. Якою пошуковою системою ви користуєтесь?
4. Що таке гіпертекстовий документ; гіперпосилання?
5. Опишіть відомі вам інтернет-загрози.
6. Яких правил безпечного користування інтернетом ви дотримуетесь?



У цьому розділі ви розширите свої знання про службу електронної пошти. Ви створите власну поштову скриньку та оціните переваги й недоліки електронного листування. Дізнаєтеся, що таке хмарні технології, й ознайомитеся з роботою таких сервісів, як онлайнві перекладачі, хмарні диски, колективне опрацювання документів тощо.



## § 13. Поштові служби інтернету

Із давніх часів люди використовували різноманітні засоби й способи для обміну повідомленнями: малюнки, сигнальні вогні, дими, згодом — потяги, літаки, телеграф, пошту та ін. Із появою комп'ютера та комп'ютерних мереж з'явилися нові можливості для спілкування.

Нині існує багато служб (сервісів) інтернету, і їхній перелік постійно поповнюється. Із деякими службами ви вже знайомі.

### Історія виникнення електронної пошти

Служба електронної пошти є однією з найдавніших комунікаційних служб інтернету. Вона прийшла на зміну «паперовій» пошті, й нині традиційне листування майже витіснилося електронним.



**Електронна пошта** (від англ. *electronic* — електронний, *mail* — пошта) — це служба обміну повідомленнями через комп'ютерну мережу з використанням поштової системи адресації.

» Історія електронної пошти (скорочено E-mail, читається «імейл») бере початок у 1971 р., коли американський інженер і програміст Рей Томлінсон (рис. 13.1) переслав повідомлення з одного комп'ютера на інший у межах кімнати. Пізніше він розробив систему електронних адрес для обміну поштовими повідомленнями в мережі ARPANet.



Рис. 13.1

### Принципи функціонування електронної пошти

Загалом принцип роботи електронної пошти нагадує модель роботи звичайної пошти. Навіть терміни, якими вона користується, ті самі: пошта, лист, скринька, адреса, відправник, адресат тощо.

Як звичайний лист складається з поштового конверта, що містить службову інформацію, та вкладеного в нього листа, так і електронний лист — із заголовків та тіла листа (рис. 13.2). До електронного листа можна прикріпляти (вкладати) файли з текстом, фотографіями тощо.

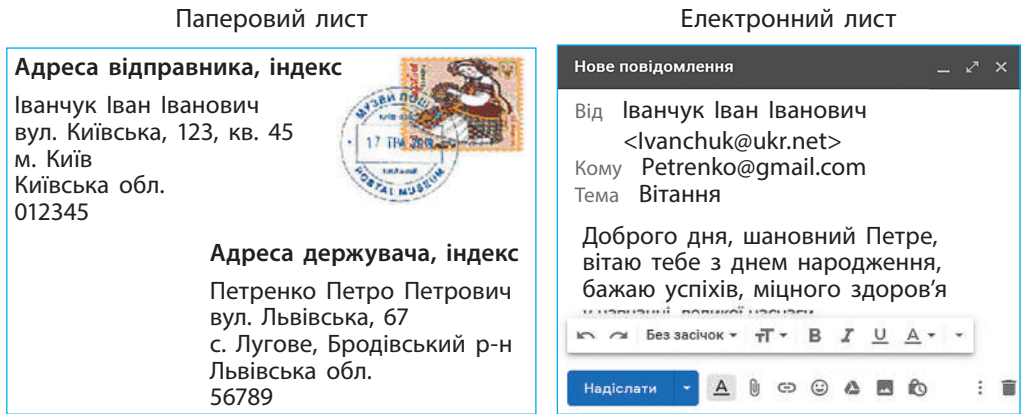


Рис. 13.2

Між комп'ютерами відправника і адресата (клієнтськими комп'ютерами) існує проміжна ланка — поштовий сервер (рис. 13.3).

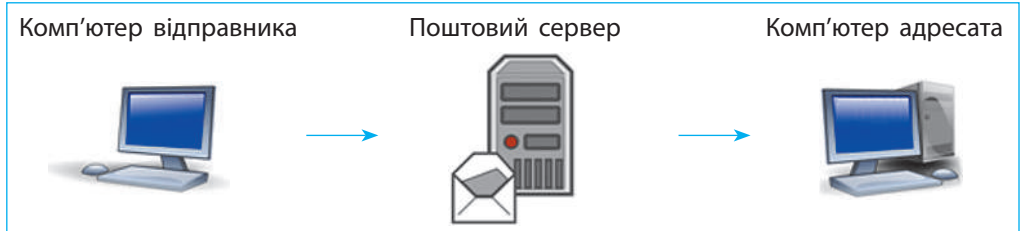


Рис. 13.3



**Поштовий сервер** — це комп'ютер, який забезпечує роботу електронної пошти завдяки встановленому на ньому спеціальному програмному забезпеченню.

Саме на поштовий сервер потрапляє і зберігається у скриньці адресата надісланий відправником лист. Користувач або користувачка може в онлайн-режимі керувати своєю поштовою скринькою на віддаленому комп'ютері-сервері засобами звичайних браузерів (рис. 13.4).



Рис. 13.4

## Структура електронної адреси

Щоб електронний лист потрапив до потрібної скриньки, відправник має зазначити адресу поштової скриньки одержувача. Кожна електронна поштова скринька має свою унікальну адресу.

Електронна адреса складається з двох частин: назви поштової скриньки та доменного імені поштового сервера, на якому розташована ця скринька, розділених символом @ (рис. 13.5). Цим символом позначають прийменник «на» (англ. *et*), тож `bober@ukr.net` слід розуміти як «bober на сервері ukr.net».



Рис. 13.5

Існують певні обмеження у формуванні адреси поштової скриньки.

- Допускається використання літер латинського алфавіту, арабських цифр і деяких символів (підкреслювання «\_», крапка «.»).
- Допускається тільки один символ @.
- Не допускається використання пропусків, розділових знаків, символів кирилиці.

Разом з тим, в електронній адресі можна знехтувати регістром літер, тобто великі й малі літери в ній рівнозначні.

1 Розгляньте різні варіанти запису однієї й тієї самої адреси.

[bober@ukr.net](mailto:bober@ukr.net)

[Bober@ukr.net](mailto:Bober@ukr.net)

[BOBER@ukr.net](mailto:BOBER@ukr.net)

[BOBER@UKR.net](mailto:BOBER@UKR.net)

[BOBER@UKR.NET](mailto:BOBER@UKR.NET)

» Томлінсон придумав використовувати в адресі символ @ (читається «ет»). У різних країнах його називають по-різному: «равлик», «собачка» тощо (рис. 13.6).



Рис. 13.6

Добираючи назву поштової скриньки (логін користувача), не варто робити її надто довгою або складною, адже її важко запам'ятати чи повідомити комусь, скажімо, телефоном.

## Правила створення паролів

До створення пароля варто поставитися серйозно. Надійність пароля зростає, якщо дотримуватися кількох рекомендацій.

- Не використовуйте як пароль свої (або рідних чи друзів) ім'я, прізвище, ініціали, дату народження, номери телефонів тощо.

- Створіть пароль завдовжки понад 8 символів.
- Комбінуйте малі й великі літери, цифри, розділові й інші знаки.

Варто створити надійну основу пароля для різних випадків реєстрування. Це підвищить вірогідність того, що користувач не заплутається, запам'ятовуючи власні паролі.

**2** Прості основи паролів, які легко вгадати чи зламати: Ivan, Lina, ABCD. Надійні основи паролів: Ivan!007, LiNa#2020.

До основи пароля в певному місці доцільно додавати ознаки точного сайту, тоді пароль до кожного сайту стане унікальним.

**3** Пароль з основою LiNa#2020 для створення скриньки на ukr.net може виглядати так: UkLiNa#2020, для реєстрації на Facebook так: FaLiNa#2020, для Twitter так: TwLiNa#2020.

## Створення поштової скриньки

Електронну поштову скриньку можна безкоштовно створити на багатьох поштових серверах, або серверах електронної пошти (рис. 13.7).



Рис. 13.7

Для створення скриньки на поштовому сервері потрібно:

- 1) відкрити сторінку реєстрації, увести затребувані сервером дані;
- 2) підтвердити реєстрацію одним із запропонованих способів.

Часто факт реєстрації потрібно підтвердити прийняттям телефонного дзвінка або СМС-повідомлення з цифровим кодом підтвердження. Номер телефону або адреса додаткової скриньки можуть знадобитись, наприклад, для відновлення пароля тощо. Деякі поштові сервери, наприклад bigmir.net, для цього пропонують дати відповідь на контрольне запитання, створене під час реєстрування скриньки.

Створюючи поштову скриньку, користувач створює свій обліковий запис (акаунт), який дає змогу використовувати додаткові інтернет-сервіси.

- ! Зареєстровану поштову скриньку (логін користувача) змінити неможливо. Особисті дані, що вводилися під час створення скриньки, можна змінити в налаштуваннях.

Користувачам варто пам'ятати, що, оскільки поштові скриньки розміщуються на віддаленому поштовому сервері, то керувати скринькою можна з будь-якого комп'ютера. Після завершення сеансу на чужому комп'ютері слід вийти зі свого облікового запису для запобігання втручанню сторонніх осіб у електронну пошту.

### Питання для самоперевірки



1. Назвіть відомі вам служби інтернету?
2. Які функції виконує поштовий сервер?
3. Яка структура поштової адреси?
4. Як створити надійний пароль?
5. Для чого може знадобитись телефон або адреса додаткової скриньки після реєстрації на поштовому сервері?
6. Що з наведеного можна змінити після створення поштової скриньки: назву скриньки, прізвище та ім'я користувача, пароль?

### Вправа 13



►► Створити поштову скриньку на поштовому сервері ukr.net або іншому.



- 1) Придумайте назву та пароль для поштової скриньки. Запустіть браузер. Відкрийте сторінку реєстрації поштового сервера mail.ukr.net. Виберіть команду Створити скриньку.
- 2) Уведіть реєстраційні дані разом із номером свого мобільного телефону. Установіть прапорець на згоду опрацювання персональних даних. Не встановлюйте прапорець імпорту листів з інших скриньок.
- 3) Клацніть кнопку Отримати код. Дочекайтеся СМС-повідомлення.
- 4) Уведіть отриманий код у віконці, що з'явилося, і натисніть кнопку Зареєструвати скриньку.
- 5) Запишіть у зошит або щоденник адресу своєї поштової скриньки. Запам'ятайте або запишіть на окремому аркуші пароль.
- 6) Клацніть значок меню поштової скриньки і вийдіть зі свого облікового запису командою Вихід. Завершіть роботу.

### Комп'ютерне тестування



Виконайте тестове завдання 13 із автоматичною перевіркою результату.



## § 14. Робота з електронною поштою

Сьогодні існує думка, що електронна пошта втрачає свої позиції. Проте за даними авторитетної американської компанії з телекомунікаційних досліджень Radicati, кількість її користувачів зростає приблизно на 3 % щороку.

### Переваги поштової скриньки

Ознайомимося з фактами, що свідчать про переваги використання поштової скриньки порівняно з іншими засобами комунікації:

- електронною поштою можна безперешкодно відправити повідомлення з будь-якої електронної адреси на будь-яку іншу адресу;
- сучасні інтернет-сервіси залежать від електронної пошти: реєстрація на більшості інтернет-ресурсів передбачає наявність скриньки для підтвердження реєстрації, відновлення пароля тощо;
- понад 50 років існування поштових сервісів є гарантією того, що електронна пошта є надійним засобом комунікації.

### Вхід до поштової скриньки

Для початку роботи з електронною поштою користувач або користувачка має увійти до своєї поштової скриньки.

Щоб увійти до поштової скриньки, потрібно:

- 1) в адресному рядку браузера записати адресу поштового сервера, на якому створено скриньку, наприклад, mail.ukr.net, gmail.com, outlook.com, mail.bigmir.net тощо — з'явиться вікно реєстрації та/чи входу до відповідної скриньки (облікового запису) (рис. 14.1);

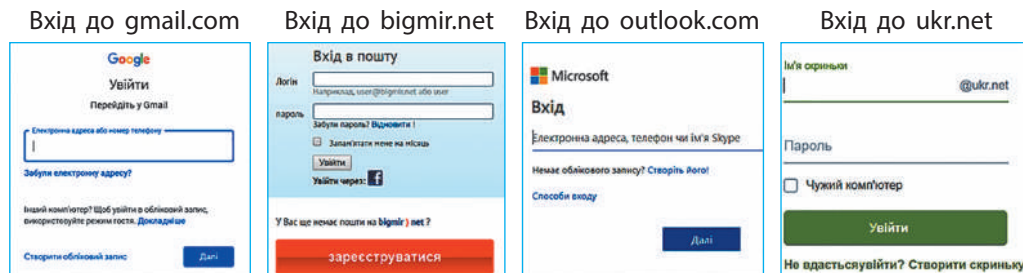


Рис. 14.1

- 2) у вікні зазначити назву поштової скриньки (логін користувача) або повну адресу скриньки чи телефон залежно від поштового сервера;
- 3) у цьому ж вікні або після клацання кнопки Далі в наступному вікні увести пароль і натиснути кнопку Увійти — з'явиться вікно поштової скриньки.

Під час роботи на чужому комп'ютері задля безпеки слід поставити відповідний прапорець і/або в спливному вікні браузера скасувати запам'ятовування пароля. Якщо ж користувач або користувачка забули пароль, їм потрібно ввести свою поштову адресу (або логін) і натиснути кнопку Забув пароль або Відновити пароль тощо, і керуватися запропонованими інструкціями.

### Вихід із поштової скриньки

Після завершення роботи на чужому комп'ютері або в чужому обліковому записі операційної системи необхідно обов'язково здійснювати вихід з електронної пошти.

Щоб **вийти з поштової скриньки** (ще говорять «із пошти», «з облікового запису», «з акаунта»), слід знайти команду чи кнопку Вихід. Вона розташовується окремо або відкривається значком меню у вигляді ≡, ⋮ або 👤 та зазвичай міститься в правому верхньому кутку вікна скриньки (рис. 14.2).

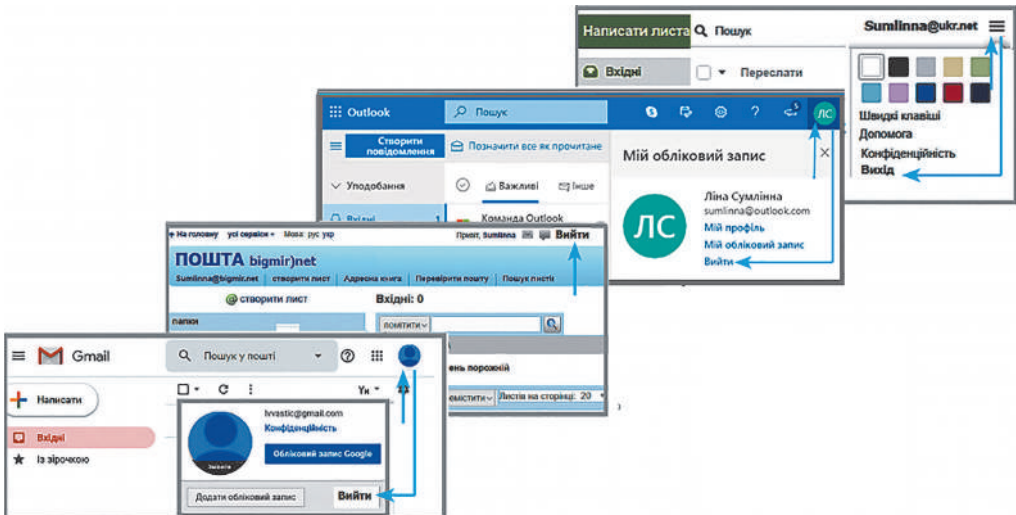


Рис. 14.2



## Створення та надсилення електронного листа

Сьогодні обмін документами здебільшого здійснюється по комп'ютерній мережі. Тож без електронного листування майже неможливо обійтися ні в діловій сфері, ні в повсякденному житті.

Щоб створити електронного листа, потрібно:

- 1) натиснути кнопку Написати листа, або Написати;
- 2) набрати текст повідомлення в тілі листа;
- 3) вказати тему листа (поле Тема) й адресу отримувача (поле Кому);
- 4) за потреби додати адреси інших отримувачів у полі Кому, або Копія;
- 5) натиснути кнопку Надіслати.

Адресу отримувача листа бажано вказувати в останню чергу.

### ▶ Пересилання файлів

В електронному листі разом із повідомленням можна надсилати фотографії, музику тощо. Щоб **надіслати файл** із комп'ютера відправника до скриньки адресата, його слід прикріпити до листа й надіслати.

Щоб у скриньці **ukr.net** **прикріпити і переслати файл**, потрібно:

- 1) підготувати лист, клацнути кнопку Файл у полі Прикріпити;
- 2) у вікні провідника, що відкриється, знайти й виділити потрібний файл або кілька файлів, утримуючи клавішу Ctrl;
- 3) натиснути кнопку Відкрити — файл приєднується до листа, тобто завантажиться на поштовий сервер, після чого ім'я файла з'явиться під кнопкою Файл у вікні скриньки (рис. 14.3);
- 4) після завершення підготовки листа клацнути кнопку Надіслати — лист разом із файлом (файлами) буде надіслано всім адресатам.

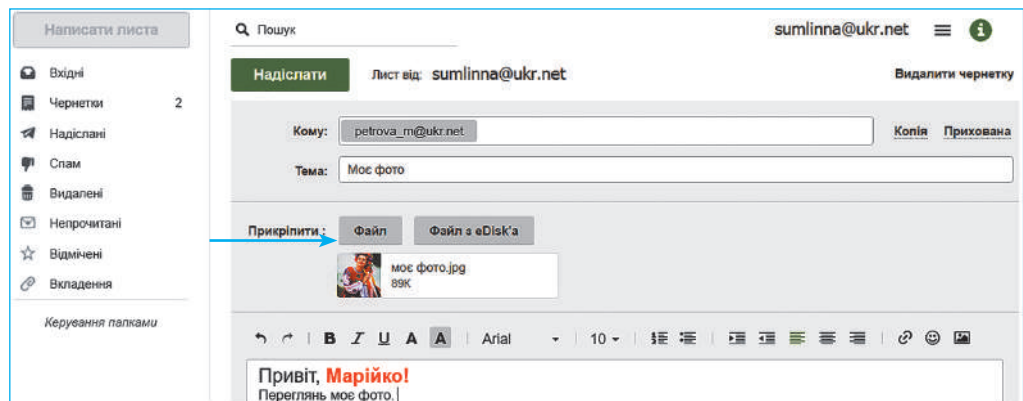


Рис. 14.3



Після цього у вікні скриньки з'явиться напис Ваш лист надіслано (рис. 14.4). Лист буде збережено в папці Відправлені скриньки відправника, а після одержання — у скриньці адресата в папці Вхідні.

### ► Редагування та форматування повідомлення

Текст електронного листа можна редагувати та форматовувати у вікні вбудованого редактора повідомлень (рис. 14.5). Такі дії аналогічні опрацюванню документів у текстовому процесорі, із яким ви знайомі.

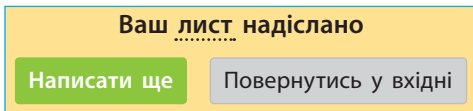


Рис. 14.4

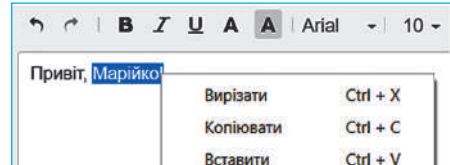


Рис. 14.5

## Перегляд та переміщення листів до папок

У папці Вхідні можна переглянути нові листи.

Якщо лист має вкладений файл, який необхідно зберегти на комп'ютері, слід клацнути кнопку Завантажити, а в діалоговому вікні збереження зазначити потрібну папку.

У поштовій скриньці є стандартні папки: Вхідні — для отриманих листів; Відправлені — для надісланих листів; Видалені — для видалених листів; Спам — для небажаних листів; Чернетки — для ненадісланих.

Окрім зазначених папок, користувачі можуть створювати й інші папки (наприклад, Важливо, Різне тощо) на свій розсуд. Це допомагає впорядковувати листи й краще в них орієнтуватися.

У випадках, коли виникає потреба **відобразити** або **приховати папки**, слід скористатися командою Керування папками (рис. 14.6).

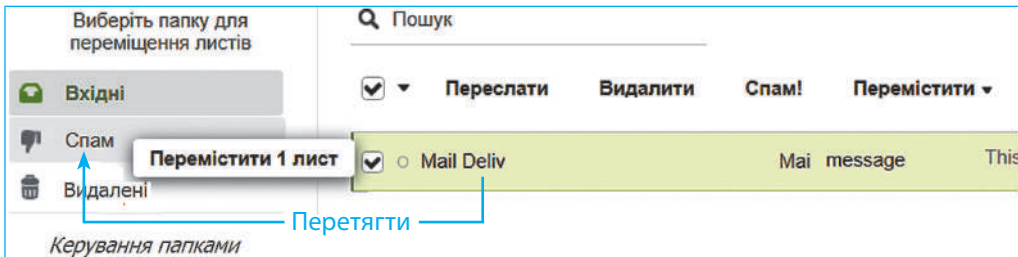


Рис. 14.6

Електронний лист можна перенести до потрібної папки перетягуванням теми на назву папки мишею або за допомогою команди Перемістити, що над списком листів. Перед цим лист потрібно відкрити або позначити у списку прапорцем.

### Питання для самоперевірки



1. Назвіть переваги електронної пошти порівняно зі звичайною поштою або з іншими сучасними засобами комунікації.
2. Як увійти до власної поштової скриньки та як вийти з неї?
3. Опишіть порядок створення та надсилання електронного листа.
4. Навіщо в електронному листі потрібне поле Тема? Знайдіть відповідь в інтернеті.
5. Які стандартні папки існують у поштової скриньці?
6. Як створити свою папку в поштової скриньці та перемістити до неї лист?

### Вправа 14



▶ Оформити та надіслати листа зі вкладеним файлом за допомогою електронної пошти.



- 1) Запустіть браузер. Знайдіть в інтернеті зображення символу @. Збережіть на комп'ютері файл із знайденим зображенням.
- 2) Зайдіть до своєї поштової скриньки. У вікні скриньки натисніть кнопку Написати листа. Напишіть у темі листа своє прізвище і клас. Уведіть, наприклад, такий текст:  
Привіт! Я вмю користуватися електронною поштою.
- 3) Збільшіть розмір символів тексту. Змініть колір слова Привіт!, наприклад, на червоний.
- 4) Прикріпіть до листа збережений графічний файл (див. п. 1).
- 5) У рядку Кому введіть дві електронні адреси — вчителя та однокласника або однокласниці. Відправте листа (кнопка Надіслати). Упевніться в тому, що листи надійшли до адресатів.
- 6) Якщо листи адресатам не доставлені, перевірте адреси і повторіть спробу. Вийдіть зі скриньки. Завершіть роботу.



### Комп'ютерне тестування

Виконайте тестове завдання 14 із автоматичною перевіркою результату.





## Практична робота 4

### Створення електронної поштової скриньки

**Завдання:** створити електронну скриньку на поштовому сервері gmail.com; оформити й надіслати листа зі вкладеним графічним файлом кільком адресатам.

**Обладнання:** комп'ютер, під'єднаний до мережі інтернет.

#### Хід роботи

*Під час роботи з комп'ютером дотримуйтеся правил безпеки.*

- ▶ 1. Придумайте назву, пароль для поштової скриньки, відкрийте браузер. Зайдіть на сторінку створення облікового запису google.com.
- ▶ 2. У вікні, що з'явилося, виберіть команду Створити обліковий запис.
- ▶ 3. За вказівками введіть потрібні дані та створіть обліковий запис.
- ▶ 4. Увійдіть до облікового запису, розгорніть меню додатків Google значком . Відкрийте електронну поштову скриньку .
- ▶ 5. У вікні скриньки натисніть кнопку Написати. Уведіть, наприклад, такий текст:  
Доброго дня. Це мій обліковий запис Google.
- ▶ 6. Відформатуйте слово Google за зразком: .  
Для форматування натисніть кнопку , що розгорне додаткову панель інструментів.
- ▶ 7. Упишіть у рядок Тема свої прізвище, ім'я, клас.
- ▶ 8. Кнопкою приєднайте до листа графічний файл, збережений на вашому комп'ютері.
- ▶ 9. У рядку Одержувачі (Кому) упишіть адресу вчителя та одного з однокласників або однокласниці.
- ▶ 10. Натисніть кнопку Надіслати. Перевірте, чи надійшли листи.
- ▶ 11. Якщо листи адресатам не надійшли, перевірте адреси отримувачів і повторіть спробу.
- ▶ 12. Вийдіть із поштової скриньки. Завершіть роботу.

**Зробіть висновок** про результати роботи.

## § 15. Особливості електронного листування

Безкоштовну електронну скриньку можна створити на багатьох поштових серверах.

Розгляньмо особливості роботи з електронною поштою на прикладі одного з найпопулярніших у світі поштових серверів — Gmail (від Google Mail) американської корпорації Google (рис. 15.1).



Рис. 15.1

### Вхід до облікового запису Google

Зареєстрованому користувачу Google перш за все необхідно **увійти у свій обліковий запис**. Для цього потрібно:

- 1) увести в адресному рядку браузера назву gmail.com і відкрити сторінку входу;
  - якщо з'явиться список із кількох облікових записів, вибрати свій;
  - якщо потрібного запису в списку немає, натиснути Вибрати інший обліковий запис;
- 2) за потреби ввести свою gmail-адресу або номер телефону, а потім увести пароль.

Увійти до облікового запису можна також зі сторінки google.com.

### Створення та використання адресної книги

Електронна пошта широко застосовується в діловому листуванні. Наприклад, деяким фахівцям за день доводиться надсилати десятки, а то й сотні листів різним адресатам. Для зручного пошуку й упорядкування контактів: адрес, прізвищ, імен, телефонів та іншої інформації, використовують адресну книгу.





**Контакт** — це сукупність даних про адресата в комунікаційних сервісах інтернету.

**Адресна книга** — це упорядкована сукупність контактів.

Адресна книга призначена для зберігання, редагування та використання контактів власником електронної поштової скриньки. Час від часу контакти доводиться переглядати та редагувати.

Щоб переглянути адресну книгу та відредагувати контакти, слід:

- 1) увійти до свого облікового запису;
- 2) в правому верхньому кутку сторінки натиснути значок меню Додатки Google ;
- 3) відкрити додаток Контакти .

## Створення та використання списків розсилки

Чи відомі вам ситуації, коли електронного листа треба розіслати кільком адресатам одночасно? Адреси двох або трьох осіб можна ввести вручну. Ви вже так робили, виконуючи справи. А якщо листа потрібно надіслати багатьом адресатам? Тут варто скористатися сервісом Списки розсилки.

У Gmail цей сервіс називають Мітки.

### ► Створення списку розсилки



**Список розсилки** — це сервіс електронної пошти, який створює можливість об'єднати певну кількість людей у єдину групу для спрощення процесу розсилання листів.



Для створення списку розсилки (мітки) слід застосувати команду Створити мітку. Потім за цими мітками можна вибрати потрібних адресатів для одночасного надсилання їм листів.

Для більш зручного користування можна створити мітки Вчителі, Друзі, Родичі, Однокласники та ін.

### ► Уведення контактів до списку розсилки

Щойно створений список розсилки порожній і не містить інформації, окрім назви списку. Додати нові контакти до списку можна, скориставшись кнопкою Створити контакт (рис. 15.2).

Щоб увести наявні контакти до списку розсилки (позначити контакт міткою), потрібно:

- 1) відкрити додаток Контакти .
- 2) вибрати Контакти або Інші контакти;
- 3) установити прапорці (галочки) на вибраних контактах;
- 4) клацнути кнопку Керування мітками  (рис. 15.3);
- 5) вибрати потрібну назву в списку міток, який з'явиться;
- 6) клацнути команду Застосувати (Застос.).

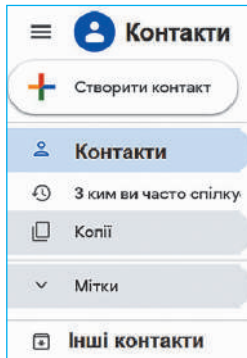


Рис. 15.2

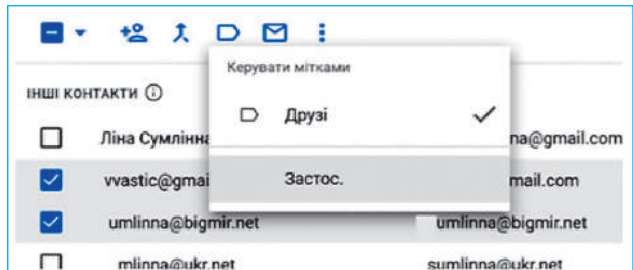





Рис. 15.3

Після цього позначені міткою контакти зі списку Інші контакти перемістяться до списку Контакти.

### ► Надсилання листів за списками розсилки (мітками)

Надіслати листа за списком розсилки (мітками) можна як з адресної книги, так і з поштової скриньки. Розгляньмо обидва способи.

Для надсилання листа з адресної книги потрібно:

- 1) відкрити адресну книгу  →  і список контактів (або вибрану мітку);
- 2) виділити потрібні контакти й клацнути значок Надіслати лист ;
- 3) написати й надіслати листа.

Для надсилання листа з поштової скриньки потрібно зайти до поштової скриньки, під час написання листа в рядку Кому увести назву мітки (списку розсилки), наприклад Друзі, яка є в адресній книзі, — з'являться всі адреси, позначені міткою.

Залишиться написати листа й надіслати його адресатам.

## Особливості написання електронного листа

Електронна пошта — це засіб зв'язку між людьми. Основними вимогами до написання як електронного, так і паперового листа є ввічливість і лаконічність.

Але існують і деякі особливості, притаманні написанню саме електронних листів. Це, окрім уже відомої вам можливості додавання (прикріплення) файлів до листа, ще й певні засоби позначення емоцій у листі, вставлення зображень до тексту листа тощо.


### ► Позначення емоцій у листах

Символьний смайлик (від англ. *smile* — посмішка) — це емоції, подані за допомогою символів, якими користуються в ході електронного листування. Смайл являє собою схематичне зображення людського обличчя, яке повернуто набік і в якому «ніс» зазвичай не зображають (рис. 15.4).

Смайлики можуть складатися з різноманітних символів. Окрім суму та радості, вони можуть висловлювати інші емоції та настрої: подив :0, підморгування ;), регіт :D та ін.

| Обличчя                                                                           | Смайли |
|-----------------------------------------------------------------------------------|--------|
|  | :-) :) |
|  | :-) :( |

Рис. 15.4

До тексту електронного листа можна також додавати художньо оформлені смайлики, інші невеличкі малюнки. Виклик меню смайликів під час написання листа здійснюється інструментом Вставити смайли .

- ! На відміну від приватного, в діловому листуванні передусім цінуються точність і інформативність. В офіційних документах смайлики не використовують.

### ► Позначення наголосів і знаків оклику

Часто в запозичених словах, деяких прізвищах або, наприклад, в омографах — словах з однаковим написанням, але різним звучанням і значенням — доцільно ставити наголос для уникнення подвійного тлумачення.

Велика літера в середині слова означає наголос на звук чи склад.

Приклади омографів: замОк (прилад) і зАмок (фортеця); кОлос (суцвіття) і колОс (велетень); пОділ (ділення) і подІл (низ).

Фрагмент тексту, введений великими літерами (рис. 15.5), в електронному листуванні зазвичай сприймається як вираження негативних почуттів і прояв роздратованості чи агресії. Писати листи великими літерами так само не етично, як кричати на співрозмовника.

|                          |                           |
|--------------------------|---------------------------|
| НЕ РОБИ ДУРНИЦЬ.         | Не роби дурниць.          |
| ↑                        | ↑                         |
| Написано                 | Написано                  |
| «голосно й роздратовано» | «спокійно і врівноважено» |

Рис. 15.5

### ► Додавання зображень до тексту

Ви вже вмієте форматувати текст електронного листа. Для надання тексту привабливого вигляду в тіло листа можна вставити малюнок і оформити як, наприклад, вітальну листівку (рис. 15.6).


Для додавання графічного зображення до тексту листа можна скористатися сполученням клавіш: скопіювати: Ctrl + C, вставити: Ctrl + V, або інструментом Вставити фотографію .



Рис. 15.6

На відміну від вкладеного файла такі зображення не потрібно додатково відкривати, отримувач одразу бачить текст із малюнком, фотографією та ін. Крім того, вставлені в текст зображення можуть бути анімованими (формат GIF), що «оживить» лист і надасть привабливості.

## Етичне і безпечне користування електронною поштою

Ми вже обговорювали важливість дотримання морально-етичних та безпекових норм спілкування під час роботи в інтернеті. Це стосується всіх його сервісів, у тому числі електронної пошти.

### ► Правила етикету електронного листування

Часто правила спілкування людей засобами інтернету називають **нетикетом** (від англ. *net* — мережа та *etiquette* — етикет).

Наведемо деякі правила електронного листування.

- Заповнюйте всі поля, призначені для службової інформації.
- Ураховуйте призначення листа, чітко окреслюйте тему.
- Пишіть повідомлення стисло, дотримуйтеся правил правопису.
- Будьте ввічливими, стриманими й толерантними.

### ► Правила безпечного листування

Робота з електронною поштою може становити загрозу, якщо не дотримуватися певних правил безпеки. Розгляньмо ці правила.

- Не повідомляйте приватну інформацію незнайомим людям.
- Не розголошуйте паролі облікових записів, поштової скриньки, коди замків, сигналізацій та ін.
- Не відповідайте на листи із заманливими пропозиціями чи погрозами, повідомляйте про них батьків і вчителів.
- Не відкривайте вкладені файли та не переходьте за посиланнями в листах від незнайомих людей.



### ► Запобігання спаму та фішингу

Зі спамом знайомий майже кожен користувач мережі.




**Спамом** зазвичай називають небажані листи переважно рекламного характеру (рис. 15.7).

Спам може містити посилання на шкідливі сайти, вкладення з комп'ютерними вірусами, заманливі пропозиції з метою переказу коштів на певні рахунки.

Спамери пильнують за адресами потенційних клієнтів за допомогою спеціальних програм-роботів.

Найнадійніший спосіб боротьби зі спамом — не дати змоги роботам розпізнати електронну адресу.

Для захисту від спаму існує кілька дієвих правил.

- Не оприлюднюйте свою електронну адресу та номери телефонів на сайтах або кодуйте адресу, або подавайте у форматі картинки, яку розпізнає тільки людина.
- Не відповідайте на спам (не переходьте за посиланнями, не відкривайте картинок), адже це тільки збільшить його кількість.
- Позначайте підозрілі листи як спам (у скриньці Gmail це значок ) , і тоді вони вміщуватимуться в папку Спам, не потрапляючи до папки Вхідні.

Одним зі способів шахрайства за допомогою спаму є фішинг.



**Фішинг** (від англ. *phishing*, за співзвучності з *fishing* риболовля) — шахрайські методи отримання особистих даних (банківських реквізитів батьків тощо) або коштів через інтернет (рис. 15.8).

До методів фішингу належить створення підробних сайтів, розсилка листів, які маскуються під офіційні повідомлення банків тощо, а також повідомлення про призи, виграші, спадок.

Щоб унеможливити свою пошту від фішингу, не слід реагувати на підозрілі повідомлення, відкривати листи від невідомих відправників тощо.

Багато поштових серверів дозволяють користувачам налаштувати власні фільтри та правила опрацювання поштових надходжень (про це далі).



Рис. 15.7



Рис. 15.8

## Питання для самоперевірки



1. Яке призначення має Адресна книга?
2. Як переглянути (відкрити) Адресну книгу (Контакти) в обліковому записі Google?
3. Що таке Список розсилки (Мітки) і як його створити?
4. Чому не варто писати текст листа великими літерами?
5. Назвіть кілька правил безпечного користування електронною скринькою.
6. Що таке спам; фішинг? Як їм запобігти?

## Вправа 15



- ▶ Створити список розсилки та надіслати електронні листи кільком користувачам.



- 1) Запустіть браузер. Відкрийте свою електронну поштову скриньку Gmail.
- 2) У новій вкладці відкрийте Адресну книгу. Додайте до списку Контакти три контакти: вчителя та двох однокласників або однокласниць.
- 3) Створіть мітку Інформатика. Додайте до списку (мітки) Інформатика три щойно створені контакти.
- 4) Відкрийте список контактів із міткою Інформатика. Створіть скріншот екрана, натиснувши клавішу PrtScr (Print Screen).
- 5) Перейдіть (відкрийте) до поштової скриньки. Напишіть текст листа з поясненням, навіщо потрібні списки розсилки (мітки). Вставте до тексту листа скріншот екрана (із буфера обміну — Ctrl + V).
- 6) Надішліть кільком однокласникам або однокласницям і вчителів листа, скориставшись міткою Інформатика (списком розсилки). Завершіть роботу.



## Комп'ютерне тестування



Виконайте тестове завдання 15 із автоматичною перевіркою результату.



## § 16. Налаштування поштової скриньки

Розгляньмо особливості налаштування профілю користувача на прикладі облікового запису Google та поштової скриньки gmail, які можуть стати вам у нагоді.



**Профілем користувача** називають сукупність персональних даних і налаштувань робочого середовища.

### Редагування профілю користувача

Після створення поштової скриньки (облікового запису) у користувача може виникнути потреба відредагувати профіль: змінити своє прізвище та/або ім'я (наприклад, якщо сталася помилка або написали ім'я з маленької літери тощо), дату народження, пароль тощо або додати персональне зображення.

#### ► Редагування персональних даних

Перед початком редагування особистих даних користувачу потрібно увійти до свого облікового запису, а потім вже відредагувати дані. Для цього потрібно виконати такі дії (рис. 16.1):

- 1) у правому верхньому кутку вікна клацнути значок із літерою (зазвичай це перша літера в імені користувача) або зображенням;
- 2) клацнути кнопку **Обліковий запис Google**;
- 3) у меню зліва вибрати команду **Особисті дані**.

Після цього в запропонованій формі можна редагувати особисті дані.

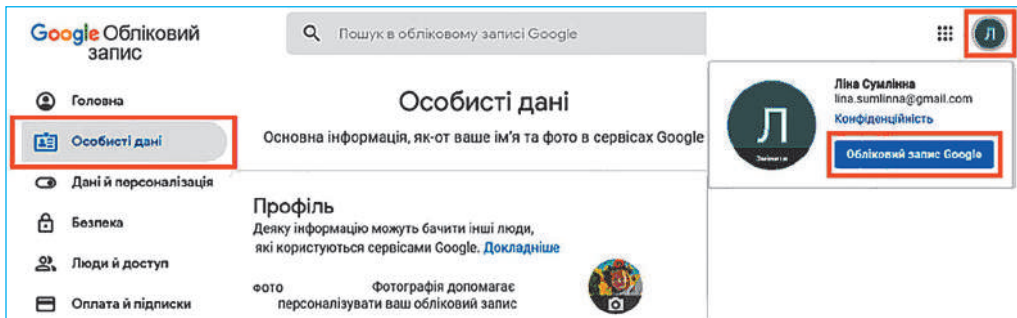


Рис. 16.1

## ▶ Додавання персонального зображення

Якщо ви хочете додати до свого профілю власну фотографію або інше зображення чи змінити його, то для цього потрібно:

- 1) зберегти на комп'ютері потрібну фотографію (малюнок);
- 2) увійти до свого облікового запису Google, у правому верхньому кутку вікна клацнути значок із літерою чи зображенням;
- 3) у віконці, що з'явилось, клацнути кнопку Змінити;
- 4) керуючись підказками, завантажити фотографію;
- 5) вибрати за допомогою кадрувальної рамки потрібний фрагмент зображення (рис. 16.2);
- 6) клацнути кнопку Зробити фотографію профілю.

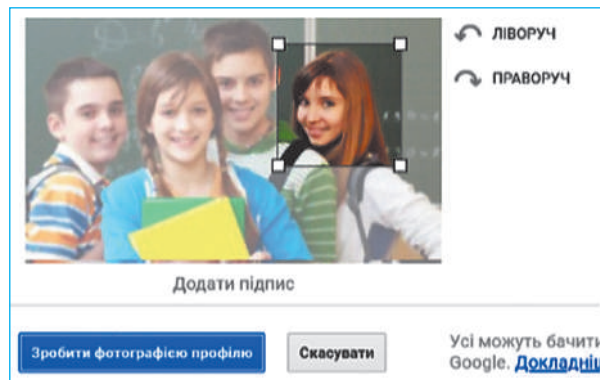





Рис. 16.2

Не варто хвилюватися, якщо одразу після додавання зображення воно не з'явилось на значку вашого облікового запису **Л**. Згодом додане зображення з'явиться: .


## Налаштування поштової скриньки

Для зручності користування поштовим сервісом і автоматизації деяких процесів під час написання листів у поштових скриньках передбачено низку налаштувань.

Щоб налаштувати поштову скриньку, користувач має спочатку її відкрити:  → .

### ► Налаштування мови інтерфейсу поштової скриньки

Щоб налаштувати мову інтерфейсу поштової скриньки, потрібно:

- 1) розгорнути значком  меню налаштувань (рис. 16.3);
- 2) вибрати команду Налаштування і відкрити вкладку Загальні;
- 3) вибрати зі списку мов інтерфейсу Gmail потрібну мову;
- 4) перейти в нижню частину вікна налаштувань і клацнути кнопку Зберегти зміни.

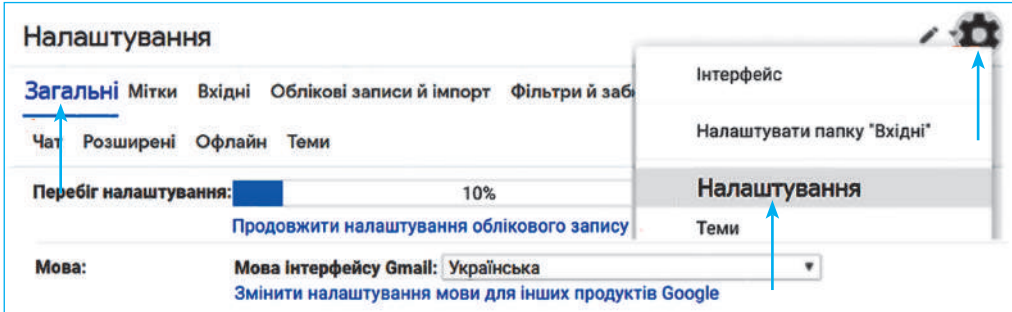


Рис. 16.3

### ► Налаштування автоматичного підпису до листа

У випадку ділового листування закінчувати електронного листа варто підписом, у якому зазначаються ім'я, прізвище, організація (компанія), телефон тощо. Щоб до кожного нового листа ці дані не доводилося вводити вручну, можна налаштувати автоматичний підпис.

Для налаштування автоматичного підпису потрібно:

- 1) відкрити вікно налаштувань на вкладці Загальні (див. рис. 16.3);
- 2) прокрутити колесом миші список налаштувань і вибрати форму Підпис (рис. 16.4);
- 3) установити перемикач у режим підпису і ввести текст підпису;
- 4) клацнути кнопку Зберегти зміни.

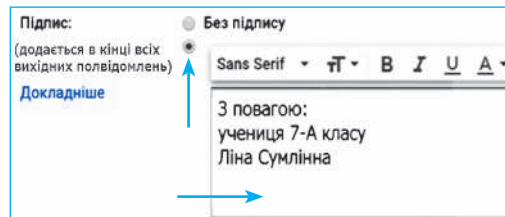


Рис. 16.4

Відтепер у кінці кожного листа автоматично ставитиметься налаштований підпис.

▶ **Перенаправлення повідомлень**

Часто користувачі інтернету мають по кілька поштових скриньок. Як зробити так, щоб пошта автоматично перенаправлялася з одної поштової скриньки на іншу? Щоб задіяти службу перенаправлення повідомлень і перенаправляти на gmail.com листи з інших ваших поштових адрес, наприклад, з mail.ukr.net, потрібно виконати такі дії:

- 1) зайти до поштової скриньки, з якої ви бажаєте перенаправляти листи до Gmail, і відкрити вікно налаштування пересилання листів: ☰ → Поштові програми → Пересилання листів (рис. 16.5);
- 2) увести адресу скриньки, до якої планується пересилати листи, та клацнути кнопку Додати (див. рис. 16.5) — на пошту Gmail буде надіслано лист із формою підтвердження;

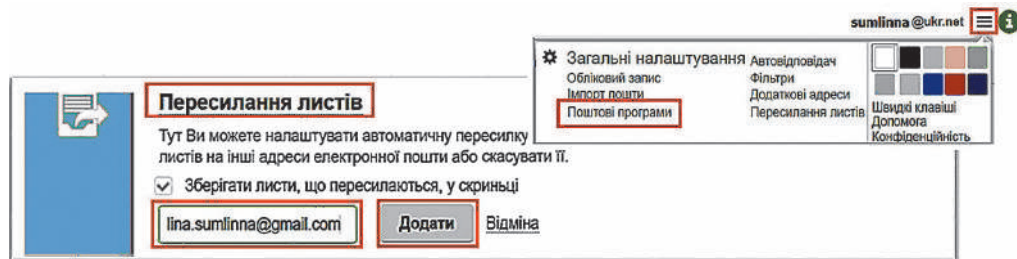


Рис. 16.5

- 3) зайти до поштової скриньки Gmail (на яку ви хочете перенаправляти листи), відкрити отриманий від mail.ukr.net лист і клацнути вказане в листі посилання.

На вашу скриньку mail.ukr.net надійде лист із повідомленням про налаштування пересилання листів (рис. 16.6).

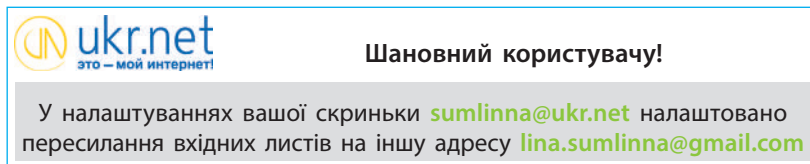


Рис. 16.6

Служба електронної пошти дає також змогу імпортувати листи. Електронну скриньку можна налаштувати так, що її поштова програма сама збиратиме листи з інших скриньок і навіть перенесе всі наявні в інших скриньках листи до основної скриньки.

## ► Групування листів

Якщо у вашій поштовій скриньці накопичилося багато листів, то для комфортної роботи з кореспонденцією ці листи варто згрупувати, наприклад, за темами, змістом, адресами, призначенням тощо.

Окрім стандартних папок: Вхідні, Відправлені, Видалені (Кошик), Спам, Чернетки, — у поштовій скриньці можна створити додаткові папки. Так само, як для впорядкування файлів на електронних носіях, їх розташовують у папках за різними ознаками, так і в поштовій скриньці електронні листи можна розподілити в окремі папки.

У скриньці Gmail папки називаються **мітками**.

Для **створення у поштовій скриньці нової папки (мітки)** потрібно:

- 1) розгорнути головне меню (значок ☰ ліворуч угорі) (рис. 16.7);
- 2) клацнути кнопку Більше (рис. 16.7, а);
- 3) вибрати команду Створити нову мітку (рис. 16.7, б);
- 4) увести у вікні, що з'явилось, назву папки й клацнути кнопку Створити (рис. 16.8).

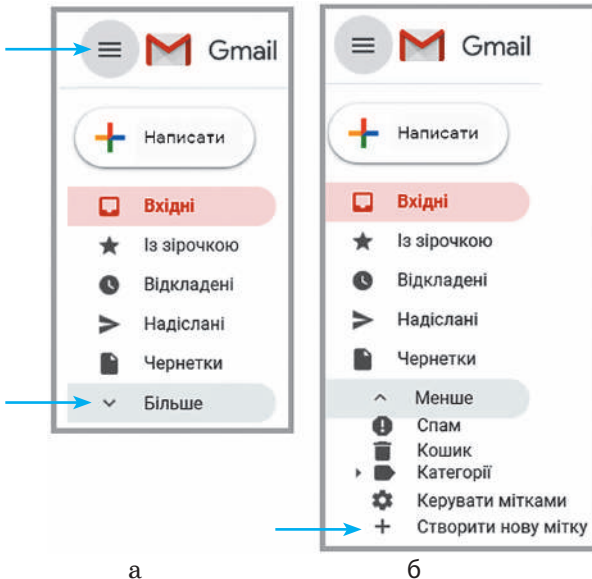


Рис. 16.7

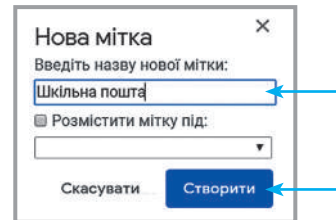




Рис. 16.8

Прапорець в полі  Розмістити мітку під означає, що ви бажаєте розмістити свою папку в тій, яку виберете із запропонованого списку.



Будь-який лист можна перемістити з однієї папки до іншої за допомогою миші або інструменту  Перемістити у (рис. 16.9). Для цього потрібно розгорнути будь-яку папку (наприклад, Вхідні), вибрати потрібний лист і перетягнути його до іншої папки, що в головному меню, або скористатися інструментом .

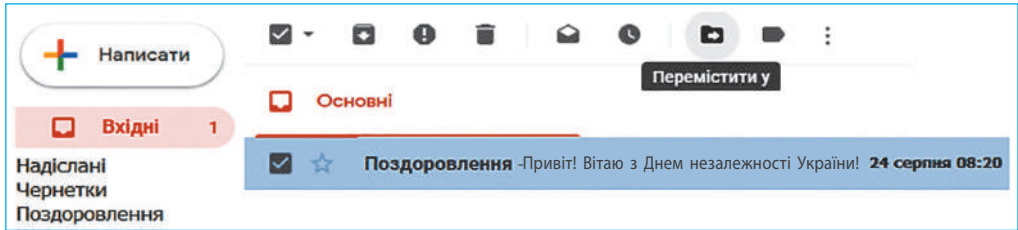


Рис. 16.9

Налаштування фільтрів дає змогу розподіляти електронні листи під час їхнього надходження за певними правилами та умовами в зазначені папки, пересилати на інші адреси тощо.

Для створення фільтра необхідно:


- 1) увійти до електронної скриньки й клацнути:  → команда Налаштування → вкладка Фільтри і заблоковані адреси;
- 2) на вкладці вибрати команду Створити новий фільтр;
- 3) у вікні, що з'явилося, зазначити параметри листів, які треба відфільтрувати, наприклад вибрати листи відправника з адресою darina@i.ua (рис. 16.10), клацнути кнопку Створити фільтр;
- 4) у новій формі прапорцем позначити необхідні дії й клацнути кнопку Створити фільтр.

Рис. 16.10



## Питання для самоперевірки



1. Що називають профілем користувача?
2. Як швидко додати персональне зображення до профілю користувача на Gmail?
3. Як додати до листів автоматичний підпис? Складіть текст підпису для ділового листування.
4. Як налаштувати перенаправлення листів зі скриньки mail.ukr.net до gmail.com?
5. Як створити у скриньці папку; як перемістити лист з однієї папки до іншої?
6. Яка функція фільтра у поштовій скриньці?

## Вправа 16



►► Навчитися редагувати профіль користувача та налаштовувати параметри поштової скриньки Google.

- 1) Запустіть браузер, увійдіть до свого облікового запису Google. Установіть персональне зображення в обліковому записі Google.
- 2) Відредагуйте профіль, напишіть українською мовою з великої літери своє прізвище та ім'я, виправте помилки, якщо вони є. Установіть українську мову інтерфейсу поштової скриньки.
- 3) Налаштуйте автоматичний підпис листа для ділового листування з учителем. Наприклад, так:

З повагою,  
учениця 7-А класу  
Ліна Сумлінна



- 4) Відкрийте нову вкладку браузера. Увійдіть до своєї поштової скриньки на mail.ukr.net. Налаштуйте перенаправлення листів з mail.ukr.net на gmail.com (не видаляйте листи від поштових служб).
- 5) Створіть у скриньці gmail.com папку Різне.
- 6) Перемістіть до папки Різне листи від поштових служб. Завершіть роботу.

## Комп'ютерне тестування

Виконайте тестове завдання 16 із автоматичною перевіркою результату.



## § 17. Використання інтернет-ресурсів

Уявімо, що ви розмістили або знайшли на каналі YouTube цікаве навчальне відео і хочете поділитися ним з друзями. Для цього відео-файл можна завантажити на комп'ютер, прикріпити до листа й надіслати. Це довго та неефективно.

Чи існують інші способи, як поділитись інтернет-ресурсом?

### Гіперпосилання в листах

Інші способи є. Можна надіслати друзям не сам файл, а гіперпосилання (його ще називають посиланням) на інтернет-ресурс.




**Гіперпосилання** — це пов'язаний з адресою ресурсу об'єкт сторінки (текст, рисунок тощо), клацанням якого здійснюється перехід до цього ресурсу.

Щоб надіслати в листі посилання на ресурс, достатньо скопіювати адресу ресурсу в адресному рядку браузера, а потім вставити її до тексту листа. Тепер, клацнувши зазначену адресу, отримувач перейде до відповідного ресурсу в інтернеті.

1 Гіперпосилання в листі може виглядати так:

<https://www.slovnky.ua/index.php?sword=%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B0%D1%86%D1%96%D1%8F>

Щоб не захаращувати лист рядками адрес ресурсів, можна зробити гіперпосиланням фрагмент тексту листа. Для цього під час написання листа достатньо виконати такі дії:

- 1) відкрити нову вкладку браузера та знайти потрібний ресурс (фотографію, відео, аудіо тощо);
- 2) скопіювати з адресного рядка браузера адресу знайденого ресурсу (сполучення клавіш Ctrl + C);
- 3) перейти на вкладку зі сторінкою поштового сервера, де розпочато написання листа;
- 4) виділити в листі фрагмент тексту й клацнути значок  Вставити посилання (рис. 17.1);

5) у вікні, що з'явилося, ввести скопійовану адресу (сполучення клавіш Ctrl + V) (рис. 17.2).

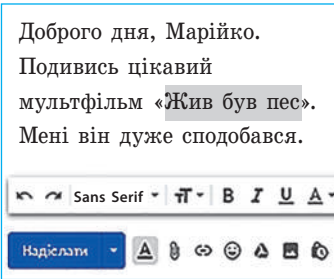


Рис. 17.1

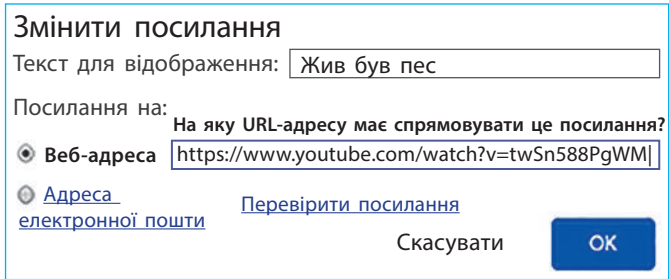




Рис. 17.2

Текст із гіперпосиланням буде виділено кольором і підкресленням, а наведений на нього вказівник миші змінюватиме вигляд (із такого  на такий ), що свідчатиме про пов'язане з текстом гіперпосилання (рис. 17.3).

мультфільм «[Жив був пес](#)»

Рис. 17.3

За гіперпосиланням отримувач листа перейде на відповідний ресурс.

## Хмарні сервіси

Із розвитком інтернету все більшого поширення набуває використання віддалених ресурсів, так званих хмарних сервісів.



**Хмарні сервіси** — це сервіси, пов'язані з наданням користувачам постійного доступу до віддалених інтернет-ресурсів (серверів, сховищ тощо).

Хмарні сервіси мають багато переваг (рис 17.4):

- користування послугами поштових серверів;
  - зберігання даних у віддалених сховищах, таких як Dropbox, SkyDrive, Google Drive та ін.;
  - використання вебпрограм: Office Web Apps (Office 365), сервіси Google-Apps та ін.;
  - користування елементами дистанційної освіти, організованої навчальними закладами (спілкування, тестування тощо);
  - телекомунікація засобами Skype, Viber, Zoom та ін.;
  - використання ресурсів віддалених серверів для обчислень тощо.
- Найактивніші постачальники хмарних сервісів — Google і Microsoft.



Рис. 17.4

▶ **Хмарний диск**



Одним із хмарних інтернет-сервісів є так званий хмарний диск.



**Хмарний диск** — це сховище даних, розташоване на віддаленому сервері й доступне через інтернет.

Для використання хмарного Google Диска достатньо створити на сайті google.com свій обліковий запис.

Власнику скриньки Gmail для збереження кореспонденції та інших файлів безкоштовно надається *хмарний диск обсягом 15 Гб*. За певних умов або за додаткову плату можна отримати й більші обсяги.

Для **відкриття хмарного диску** слід зайти до облікового запису, та кнопкою  відкрити перелік додатків і вибрати  Диск (рис. 17.5).

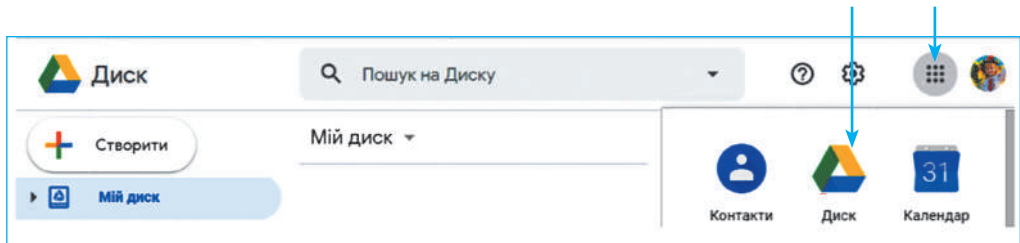


Рис. 17.5

Ознайомимося з особливостями завантаження і створення документів на хмарному диску. На хмарний диск можна завантажити файли й папки з комп'ютера (рис. 17.6) або створити їх безпосередньо на хмарному диску.

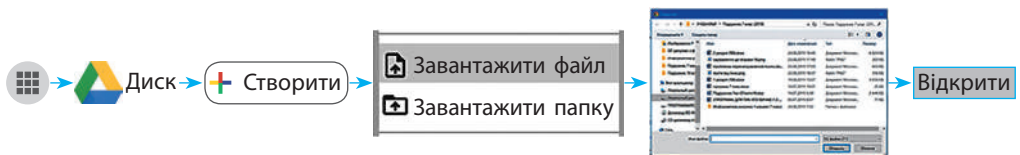


Рис. 17.6

Створити можна текстовий документ, електронну таблицю, презентацію, малюнок, карту, форму опитування тощо.

Крім браузера, жодного додаткового програмного забезпечення користувачеві не потрібно.



## Колективна робота з документами

Документи, збережені на хмарному диску, можуть бути опрацьовані будь-якими користувачами, які мають до них доступ.

### ► Типи та рівні доступу до файлів

Доступ до файлів надає власник хмарного диску.

Існують *два типи доступу* до файла:

- 1) всім, хто має посилання на файл: увімкнено: , вимкнено: ;
- 2) лише тим користувачам, кому надано доступ за їх іменами або електронними адресами.

Існують *три рівні доступу* до файла:

- 1) перегляд; 2) коментування; 3) редагування.

Кожен наступний рівень надає користувачеві всі можливості попередніх рівнів доступу. Так, найвищий рівень «редагування» включає можливість перегляду та коментування даних у файлах.

### ► Відкриття доступу

Для надання доступу до даних потрібно в контекстному меню файла вибрати команду Надати доступ (рис. 17.7). Або в режимі онлайн-перегляду файла клацнути кнопку Спільний доступ (рис 17.8).

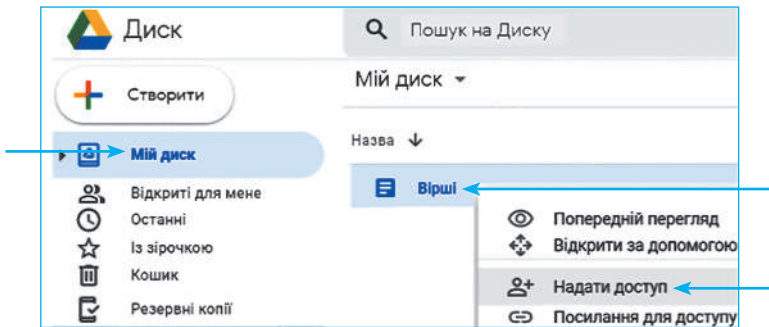


Рис. 17.7

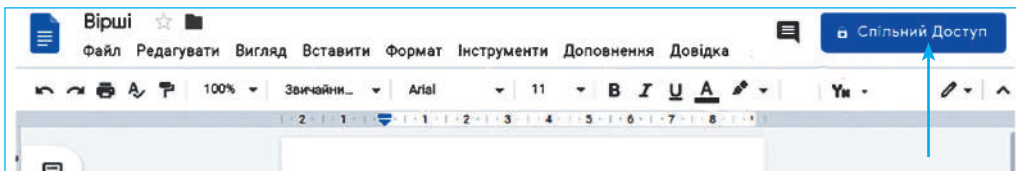


Рис. 17.8

Після цього у формі, що відкриється, потрібно встановити необхідний тип і рівень доступу та клацнути кнопку Готово (рис. 17.9).

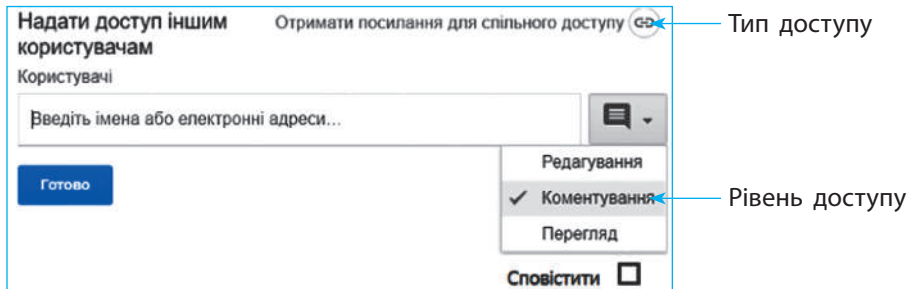


Рис. 17.9

Якщо в адресному рядку форми доступу зазначити електронні адреси, то адресати або отримають автоматичний лист-повідомлення про те, що їм надано доступ до файла (за умови встановлення прапорця Сповістити), або дізнаються про це, увійшовши до свого Google-диску.

## Онлайн-перекладачі

Ще одним хмарним сервісом є онлайн-перекладачі.

Розглянемо, напевне, найпопулярніший у світі Google Перекладач (рис. 17.10). Користуватися ним можуть не лише ті, хто має обліковий запис Google, а й будь-хто з користувачів інтернету.



Рис. 17.10

Google Перекладач може перекладати з понад 100 різних мов. Він також надає можливість голосового введення тексту, прослуховування правильної вимови вихідного або перекладеного тексту.

Для здійснення перекладу Google Перекладачем потрібно:

- 1) вибрати мову вихідного тексту (або автоматичне визначення мови); вибрати мову, на яку слід перекласти;
- 2) зазначити, що перекладається: уведений текст чи файл, який містить текст;
- 3) увести вихідний текст чи завантажити файл.

Після цього Google Перекладач запропонує варіанти перекладу, розташовані за частотою їхнього вживання (рис. 17.11).

Клацнувши перекладений текст, ви отримаєте можливість запропонувати свій варіант перекладу, який буде враховано в подальшому.



Кінцеві пристрої (речі) завдяки датчикам і сенсорам приймають і/або передають інформацію, а завдяки виконавчим механізмам виконують потрібні дії. Мережа передавання даних може бути бездротовою та/або дротовою. Для обміну даними IoT існує кілька стандартів. Одні з них працюють на основі мобільних мереж, інші використовують власні незалежні мережі або звичайні електромережі. Узгоджене використання цих стандартів є запорукою успішного розвитку глобальної мережі IoT. Система управління мережею і пристроями базується на використанні комп'ютерів, контролерів і відповідного програмного забезпечення.

За різними даними, у світі до інтернету під'єднано понад 16 млрд пристроїв. Передбачається, що до 2022 року їх буде вже 29 млрд, 18 із яких належатимуть до інтернету речей.

### Питання для самоперевірки



1. Що таке хмарні сервіси? Назвіть відомі вам хмарні сервіси.
2. Що таке хмарний диск?
3. Як створити документ на Google Диск? Як відкрити Google Диск?
4. Які типи та рівні доступу до файлів вам відомі?
5. Які функції можуть надавати онлайн-перекладачі?
6. Що таке інтернет речей? Наведіть приклади його використання.

### Вправа 17



- Створити на хмарному диску й колективно опрацювати документ за допомогою онлайн-перекладача.
- 1) Запустіть браузер і зайдіть до свого облікового запису Google. Відкрийте Google Диск. Створіть на хмарному диску Google-документ. Дайте документу назву: Хмарні технології.
  - 2) Знайдіть в інтернеті визначення поняття «хмарні технології». Скопіюйте текст означення до створеного Google-документа.
  - 3) Відкрийте нову вкладку Google Перекладач, перекладіть знайдене означення на одну з мов, яку ви вивчаєте. За можливістю прослухайте переклад. Скопіюйте його до створеного Google-документа.
  - 4) Відкрийте доступ для коментування документа вчителю, однокласнику чи однокласниці з автоматичним сповіщенням.





- 5) Зайдіть до своєї поштової скриньки Gmail. Відкрийте лист-сповіщення про доступ і перегляньте документ, доступ до якого вам надано однокласником або однокласницею.
- 6) Зробіть коментар до перекладеного в документі означення, наприклад, чи задовольняє вас якість перекладу. Завершіть роботу.



### Комп'ютерне тестування

Виконайте тестове завдання 17 із автоматичною перевіркою результату.



## Практична робота 5

### Використання хмарних технологій для колективного опрацювання документів та користування геосервісами

**Завдання:** створити Google-презентацію із використанням хмарних сервісів, наприклад геосервісів, карт, хмарного диску.

**Обладнання:** комп'ютер, під'єднаний до мережі інтернет.

#### Хід роботи

*Під час роботи з комп'ютером дотримуйтеся правил безпеки.*

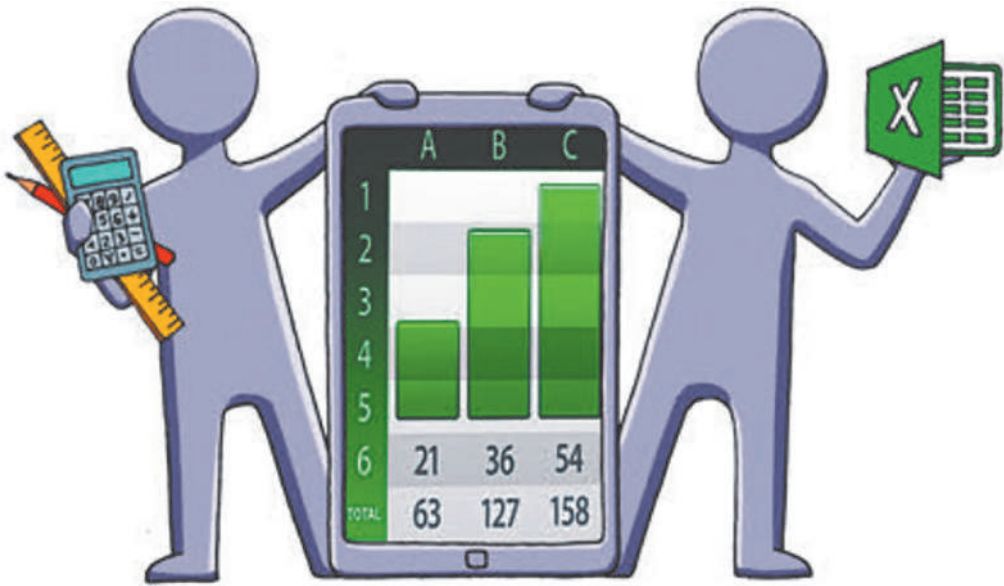
- ▶ 1. Відкрийте браузер. Зайдіть до своєї поштової скриньки Gmail.
- ▶ 2. Оформіть листа вчителю з поясненням у ньому, яке із великих міст України чи іншої європейської держави вам подобається. Не надсилайте листа.
- ▶ 3. У новій вкладці браузера відкрийте Google Диск. Створіть на хмарному диску презентацію. За назву презентації візьміть назву міста, зазначеного в листі.
- ▶ 4. На *першому* слайді помістіть тему презентації, прізвище ім'я і клас автора презентації.

- ▶ 5. У новій вкладці браузера знайдіть в інтернеті автомобільний маршрут від населеного пункту, в якому ви мешкаєте, до європейського міста, про яке йдеться в презентації.
- ▶ 6. Скопіюйте карту з маршрутом і відстанню в кілометрах і помістіть отримане зображення на *другий* слайд Google-презентації.
- ▶ 7. У новій вкладці браузера знайдіть сторінку Google Earth (Google Планета Земля).  
Завантажте з офіційного джерела інсталяційний файл Google Earth. Запустіть цей файл і дочекайтеся встановлення програми Google Earth. Запустіть Google Earth.  
Замість програми можна скористатись онлайн-сервісом Планета Земля (Google Earth), що відкривається у браузері Google Chrome.
- ▶ 8. У пошуковому рядку вікна програми Google Earth уведіть назву вибраного міста та клацніть кнопку Пошук.  
Прокрутіть колесо й перетягніть з натиснутою лівою кнопкою миші карту так, щоб «опустіться» на мінімально можливу висоту над обраним місцем до появи тривимірного зображення будинків і вулиць.  
«Помандруйте» вулицями міста.
- ▶ 9. «Зайдіть» до приміщення кафе, музею, вокзалу тощо.  
Зробіть скріншот із зображенням одного з приміщень.  
Помістіть скріншот на *третій* слайд Google-презентації.
- ▶ 10. Відкрийте доступ усім інтернет-користувачам, які мають посилання для перегляду та коментування Google-презентації, із автоматичним поштовим сповіщенням однокласника чи однокласниці.
- ▶ 11. Дочекайтеся отримання сповіщення про доступ до документа від однокласників.  
Відкрийте лист-сповіщення про доступ, перегляньте доступний документ (презентацію) і додайте до назви міста коментар, наприклад, чи подобається вам місто, про яке йдеться у презентації.
- ▶ 12. До свого листа вчителеві додайте гіперпосилання на файл Google-презентації, пов'язавши його з назвою улюбленого міста в тексті листа. Надішліть листа вчителеві.  
Завершіть роботу.

**Зробіть висновок** про результати роботи.

# РОЗДІЛ 3

## ОПРАЦЮВАННЯ ТАБЛИЧНИХ ДАНИХ



§ 18. Середовище табличного процесора

§ 19. Робота з табличними даними

§ 20. Типи даних в електронних таблицях

§ 21. Автоматичні дії над вмістом клітинок

§ 22. Формули в електронних таблицях

§ 23. Форматування електронних таблиць

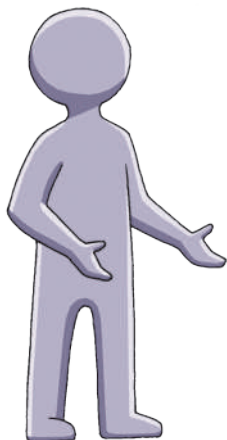
Практична робота 6. Створення й опрацювання електронної таблиці

§ 24. Моделі і моделювання

§ 25. Реалізація математичних моделей засобами електронних таблиць

Практична робота 7. Побудова математичної моделі

## ПОВТОРЮЄМО



У 5 класі, вивчаючи опрацювання текстових даних, ви навчилися працювати з *таблицею* — об'єктом, який складається з рядків і стовпців. Таблиці використовують для упорядкування й унаочнення даних.

Під час виконання вправ на комп'ютері ви набули практичних навичок у створенні таблиць, заповнені їх даними, редагуванні та форматуванні документів із таблицями й іншими типами даних — текстовими, графічними тощо.

Таблиці, з якими ви працювали, не призначені для складних розрахунків або автоматичного опрацювання даних. Проте деяка можливість опрацювання табличних даних формулами все ж існує.

Здобуті знання стануть вам у нагоді під час опанування *табличного процесора* — програми, призначеної для роботи з електронними таблицями.

1. Що являє собою таблиця в текстовому документі?
2. Яке призначення таблиць у текстовому документі?
3. Як можна вставити таблицю в текстовий документ?
4. Чи можна змінити кількість рядків (стовпців)?
5. Як змінити ширину рядка (висоту стовпця)?
6. Чи можна виконувати обчислення в таблиці у текстовому документі?



У цьому розділі ви ознайомитеся з табличним процесором, дізнаєтеся, як редагувати та формувати електронну таблицю, як опрацьовувати дані в електронних таблицях. Ви навчитесь реалізовувати математичні моделі засобами табличного процесора.



» Завдяки відомим вам хмарним технологіям із ЕТ можна працювати в онлайн-режимі, створивши на Google Диску Google-таблицю (рис. 18.3). Користування Google-таблицею є безкоштовним і не потребує додаткового програмного забезпечення. Не викликає проблем копіювання чи переміщення даних із комп'ютерних таблиць до хмарних Google-таблиць і навпаки.

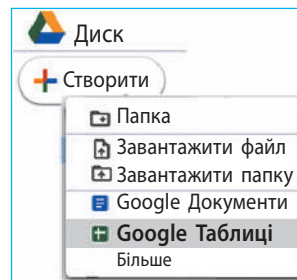
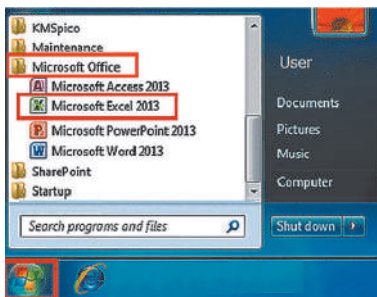


Рис. 18.3

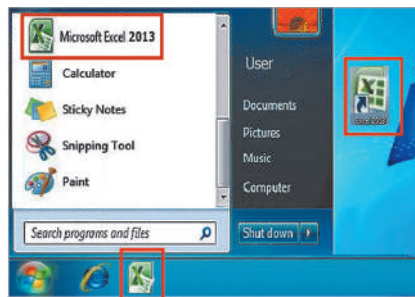
## Запуск Excel і створення документа

В ОС Windows стандартним є запуск програм за допомогою меню Пуск. Для запуску Excel в ОС Windows XP та/або Windows 7 потрібно клацнути: Пуск → Всі програми → Microsoft Office → Microsoft Excel (рис. 18.4, а).

Щоб швидко запустити програми, слід розмістити ярлики Microsoft Excel безпосередньо в меню Пуск, на панелі завдань або на робочому столі командами (рис. 18.4, б) контекстного меню програми.



а



б

Рис. 18.4

Версії Excel від 2010 до 2019 року випуску різняться незначно. Навчившись користуватись Excel 2013, ви орієнтуватиметесь і в решті версій.

Після запуску Excel на екрані з'являється перелік останніх документів і вікно онлайн-пошуку шаблонів (рис. 18.5).

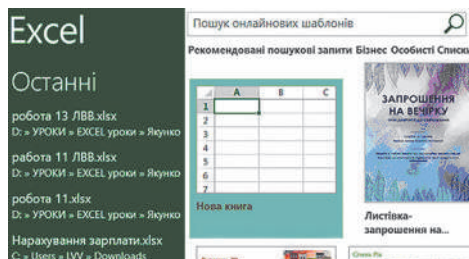


Рис. 18.5



Знайшовши потрібний шаблон, можна відкрити заповнений вибраними даними документ.

Для створення документа з порожньою таблицею після запуску Excel потрібно клацнути зображення з написом Нова книга. Після цього відкриється вікно програми (рис. 18.6).

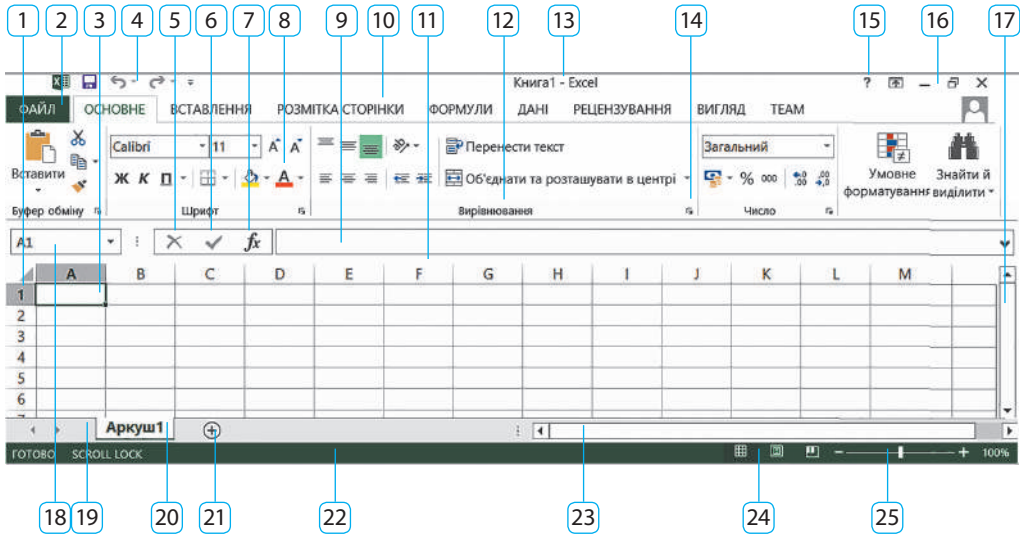


Рис. 18.6

## Інтерфейс користувача

Інтерфейс Excel подібний до інтерфейсу програм Word та PowerPoint.

Пригадаємо відомі вам загальні елементи вікна офісних програм (рис. 18.6): 2 — вкладка Файл (Офіс); 4 — панель швидкого доступу; 8 — стрічка інструментів; 10 — меню вкладок; 12 — група інструментів; 13 — рядок заголовка; 15 — кнопка довідки; 16 — кнопки керування вікном; 17, 23 — смуги вертикального та горизонтального прокручування; 22 — рядок стану; 24 — кнопки режиму перегляду; 25 — елементи масштабування.

Вікно містить також специфічні елементи, притаманні лише Excel: 1 — номери рядків, позначені натуральними числами; 11 — заголовки (номери) стовпців, позначені зазвичай однією або кількома латинськими літерами; 20 — вкладка аркуша; вкладок може бути кілька; 21 — кнопка створення нового аркуша.

Про інші об'єкти ЕТ йтиметься далі.

## Робота з об'єктами електронної таблиці

Об'єктами ЕТ є клітинка, діапазон, рядок, стовпець, аркуш, книга тощо. З ними можна виконувати низку операцій: редагування, форматування, копіювання, видалення тощо.

### ▶ Виділення клітинок і діапазонів клітинок

! Перед початком роботи з необхідним об'єктом ЕТ його потрібно виділити.

Виділення клітинки найчастіше здійснюється одинарним (для введення даних чи форматування) або подвійним (для редагування наявних даних) клацанням або введенням адреси чи імені клітинки в поле Ім'я (рис. 8.6, позначка 18, про це — далі).

Якщо потрібно опрацювати дані кількох клітинок одночасно, то працюють із діапазоном.



**Діапазон** — це сукупність клітинок, які можна опрацювати як єдине ціле.

Розрізняють зв'язні й незв'язні діапазони.

**Зв'язним** називають прямокутний діапазон, який можна виділити переміщенням вказівника з натиснутою лівою кнопкою миші (далі — ЛКМ). Щоб **виділити прямокутний діапазон**, потрібно перетягнути вказівник від однієї кутової клітинки до діагонально протилежної або ж клацнути кутову клітинку, а потім із натиснутою клавішею Shift — діагонально протилежну. Виділений діапазон позначається сірим кольором, окрім клітинки, з якої почалося виділення (рис. 18.7).

**Незв'язним** називають діапазон, який складається з кількох діапазонів і/або клітинок. Щоб **виділити незв'язний діапазон**, потрібно, утримуючи клавішу Ctrl, виділити кілька клітинок і/або зв'язних діапазонів (рис. 18.8).

Для зняття виділення діапазону клітинок достатньо клацнути на будь-якій клітинці ЕТ.

|   | A | B | C |
|---|---|---|---|
| 1 |   |   |   |
| 2 |   |   |   |
| 3 |   |   |   |
| 4 |   |   |   |

Рис. 18.7

|   | A | B | C | D |
|---|---|---|---|---|
| 1 |   |   |   |   |
| 2 |   |   |   |   |
| 3 |   |   |   |   |
| 4 |   |   |   |   |

Рис. 18.8



### ► Виділення рядків і стовпців

Для виділення одного рядка (стовпця) потрібно клацнути на його заголовку (номері).

Декілька суміжних рядків (стовпців) виділяють перетягуванням вказівника по їхніх заголовках (номерах). Під час виділення у сірому прямокутнику зазначається кількість рядків — R і стовпців — C, що входять до виділеного діапазону (рис. 18.9).

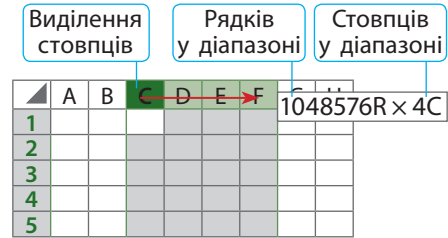






Рис. 18.9

Для виділення усієї ЕТ потрібно у лівому верхньому кутку таблиці клацнути значок  або натиснути сполучення клавіш Ctrl + A.

### Збереження таблиці

Як і інші документи, ЕТ для подальшого використання слід зберегти. Для цього потрібно виконати знайомі вам дії.

- 1) На панелі швидкого доступу клацнути значок Зберегти  або на вкладці ФАЙЛ вибрати команду Зберегти (або Зберегти як...).
- 2) Зазначити місце збереження — одну з наявних у списку папок або вибрати іншу папку, скориставшись кнопкою .
- 3) Увести назву файлу у поле Ім'я файлу і натиснути кнопку Зберегти. Для збереження наявного документа в іншу папку та/або з іншим іменем слід вибрати команду Зберегти як...

Для повернення в режим редагування з будь-якого кроку збереження потрібно натиснути кнопку  або клавішу Esc.

### Питання для самоперевірки




1. Що таке електронна таблиця? Для чого її застосовують?
2. Що таке табличний процесор?
3. Опишіть кроки стандартного запуску Excel. Чи можна запустити Excel іншим чином?
4. Як знайти потрібний шаблон ЕТ і як створити порожню ЕТ?
5. Назвіть кілька специфічних для Excel елементів вікна.
6. Опишіть порядок виділення діапазону, рядка, стовпця ЕТ.

## Вправа 18



▶ Ознайомитися з табличним процесором шляхом створення електронної таблиці та виділення її об'єктів.

- 1) Налаштуйте запуск  Microsoft Excel ярликами на робочому столі, панелі завдань, меню Пуск.
- 2) Запустіть Excel. Знайдіть у розділі Календарі онлайн шаблон Календар на 12 місяців, відкрийте знайдений шаблон. Збережіть документ із назвою Календар.
- 3) Створіть порожню ЕТ. Виділіть у таблиці кілька суміжних рядків, запишіть у зошит кількість стовпців у них. Визначте загальну кількість рядків в ЕТ та запишіть значення в зошит. Підрахуйте і запишіть у зошит загальну кількість клітинок в ЕТ.
- 4) Виділіть незв'язний діапазон, як зображено на рис. 18.10, рядки і стовпці виділіть повністю. Зробіть скриншот екрану.

|    | A | B | C | D | E | F | G | H | I | J |
|----|---|---|---|---|---|---|---|---|---|---|
| 1  |   |   |   |   |   |   |   |   |   |   |
| 2  |   |   |   |   |   |   |   |   |   |   |
| 3  |   |   |   |   |   |   |   |   |   |   |
| 4  |   |   |   |   |   |   |   |   |   |   |
| 5  |   |   |   |   |   |   |   |   |   |   |
| 6  |   |   |   |   |   |   |   |   |   |   |
| 7  |   |   |   |   |   |   |   |   |   |   |
| 8  |   |   |   |   |   |   |   |   |   |   |
| 9  |   |   |   |   |   |   |   |   |   |   |
| 10 |   |   |   |   |   |   |   |   |   |   |
| 11 |   |   |   |   |   |   |   |   |   |   |
| 12 |   |   |   |   |   |   |   |   |   |   |

Рис. 18.10



- 5) Підрахуйте і запишіть у зошит кількість клітинок виділеного діапазону (п. 4).
- 6) Збережіть документ з назвою Вправа 18. Надішліть листа вчителю зі скриншотом (п. 4), записами в зошиті (п. 3, 5) і прикріпленими файлами: Календар.xlsx (п. 2), Вправа 18.xlsx. Завершіть роботу.



## Комп'ютерне тестування

Виконайте тестове завдання 18 із автоматичною перевіркою результату.



## § 19. Робота з табличними даними

З табличними даними можна виконувати низку операцій.

### Відкриття таблиці

Існує декілька способів відкриття створеної раніше таблиці Excel:

- у відповідній папці знайти документ і відкрити його;
- запустити Excel, клацнути потрібну назву останніх документів;
- у вікні Excel виконати команду Файл → Відкрити, знайти і відкрити ET;
- у меню Пуск (якщо ярлик Excel закріплено в меню) розгорнути випадний список ярлика й вибрати файл потрібної таблиці.

У всіх наведених випадках відкриється вибрана таблиця, з якою користувач може продовжити роботу.

### Введення даних

У клітинках ET дані можна зберігати й опрацьовувати.

Якщо якусь клітинку виділити клацанням, то наявні в клітинці дані буде замінено новими, що вводитимуться. Під час введення даних стають активними два значки: Скасувати ✕ і Ввід ✓ (рис. 19.1).

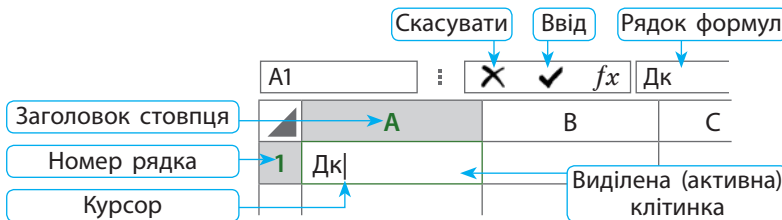


Рис. 19.1

Уводити й редагувати дані можна як у клітинці, так і в рядку формул. Якщо у вікні рядок формул відсутній, на вкладці Вигляд слід встановити позначку:  Рядок формул.

Підтвердити введення даних можна клацанням значка Ввід ✓ (рис. 19.1) або іншої клітинки, а введення даних із переходом до сусідньої клітинки праворуч (нижче) — натисканням клавіші Tab (Enter).

Щоб скасувати введення щойно набраних даних потрібно натиснути клавішу Esc або клацнути значок Скасувати **X** (рис. 19.1).

► **Особливості відображення даних**

Після підтвердження введення даних текст, який не вміщується в клітинку, буде виступати на порожні клітинки праворуч (рис. 19.2).

|          |   |           |                                     |   |   |   |  |
|----------|---|-----------|-------------------------------------|---|---|---|--|
| <b>X</b> | ✓ | <i>fx</i> | Ми вивчаємо табличний процсор Excel |   |   |   |  |
|          | C | D         | E                                   | F | G | H |  |
|          |   |           | Ми вивчаємо табличний процсор Excel |   |   |   |  |

Рис. 19.2

Якщо ж клітинка праворуч теж містить дані, то текст у клітинці, що міститься ліворуч, буде сховано за її правою межею. Введені дані при цьому не втрачаються, їх можна побачити у рядку формул.

Дробова частина числа, яка не вміщується в клітинку, буде автоматично округлятися (рис. 19.3, а). Якщо ціла частина не поміщається в клітинці, то число подається в стандартному вигляді (рис. 19.3, б), а якщо й це не вдається, виводиться «решітка» (рис. 19.3, в).

|          |   |           |            |          |   |           |            |          |   |           |            |
|----------|---|-----------|------------|----------|---|-----------|------------|----------|---|-----------|------------|
| <b>X</b> | ✓ | <i>fx</i> | 1234567,89 | <b>X</b> | ✓ | <i>fx</i> | 1234567,89 | <b>X</b> | ✓ | <i>fx</i> | 1234567,89 |
|          | C | D         | E          |          | C | D         | E          |          | C | D         | E          |
|          |   |           | 1234567,9  |          |   |           | 1E+06      |          |   |           | ####       |


а
б
в

Рис. 19.3

### Редагування даних

Часто виникає потреба в редагуванні (змінінні) табличних даних. У цілому редагування даних в електронній таблиці схоже з редагуванням даних у текстовому документі.

**!** Для редагування наявних у клітинці даних її потрібно виділити подвійним клацанням.

Для видалення символів у клітинці потрібно скористатися клавішами Backspace або Delete, для скасування останніх дій — сполученням клавіш Ctrl + Z або кнопкою Скасувати .

Доступні також операції копіювання, вирізання, вставлення даних відповідними сполученнями клавіш Ctrl + C, Ctrl + X, Ctrl + V або командами контекстного меню.

## Форматування даних

Звичайно, ЕТ призначені для опрацювання даних комп'ютером, проте працюють із ними користувачі-люди. Отже, для кращого сприйняття табличних даних варто приділяти увагу естетичному оформленню таблиці. Це налаштовується шляхом форматування даних.

### ► Формат шрифту

Пригадаємо призначення відомих вам інструментів форматування шрифту (рис. 19.4): 1 — гарнітура шрифту; 2 — розмір шрифту; 3 — стиль накреслення шрифту; 4 — встановлення меж клітинки; 5 — колір заливки; 6 — змінення розміру шрифту; 7 — колір шрифту; 8 — відкриття вікна налаштування шрифту.

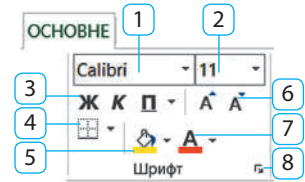


Рис. 19.4

Для форматування наявних у клітинці даних достатньо клацнути клітинку та вибрати потрібний інструмент. Для форматування окремих текстових даних — клацнути двічі та виділити фрагмент тексту. Форматування окремих фрагментів нетекстових даних неможливе.

### ► Вирівнювання даних у клітинці

У клітинці текст автоматично вирівнюється за лівим краєм, а числа — за правим. Щоб змінити **вирівнювання вмісту**, потрібно спочатку виділити клітинку (одинарним клацанням!) чи діапазон, а потім скористатись відповідними інструментами вкладки Основне групи Вирівнювання. Залежно від вибраних інструментів горизонтального і вертикального вирівнювання маємо 9 результатів розташування (рис. 19.5).

Інструменти горизонтального вирівнювання

Інструменти вертикального вирівнювання





Рис. 19.5

### ► Орієнтація запису даних

Іноді записи у клітинках таблиці доводиться повертати. Пригадайте сторінку журналу успішності, на якій дати записано знизу вгору.

Щоб повернути запис на довільний кут, потрібно:

- 1) у списку  вибрати команду Формат вирівнювання клітинки, або розгорнути вікно налаштувань значком , або в контекстному меню клітинки вибрати команду Формат клітинок;
- 2) у вікні Формат клітинок відкрити вкладку Вирівнювання (рис. 19.6);
- 3) у блоці Орієнтація вказати кут повороту або перетягнути маркер Напис на потрібний кут і натиснути кнопку ОК.

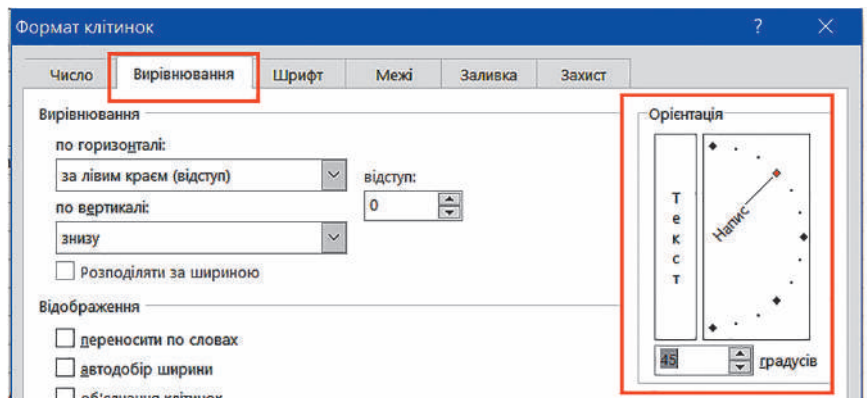




Рис. 19.6

### ► Формат за зразком

Щоб застосувати формат однієї клітинки до інших, не змінюючи їхнього вмісту, потрібно виділити цю клітинку, на вкладці Основне клацнути кнопку Формат за зразком  та протягнути вказівник-пензель по клітинках, для яких потрібно застосувати вибраний формат.

Для багаторазового використання форматування за зразком слід двічі клацнути кнопку , а щоб вийти з цього режиму — клавішу Esc.

### Зміна розмірів стовпців і рядків

Щоб змінити ширину стовпця (висоту рядка), достатньо перетягнути його праву (нижню) межу між заголовками (номера-ми). При цьому виводяться поточні розміри: у пунктах і пікселях — для рядків, у символах і пікселях — для стовпців при звичайному режимі перегляду (рис. 19.7).

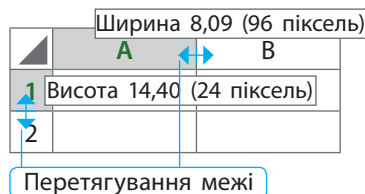


Рис. 19.7

### Питання для самоперевірки



1. Опишіть способи відкриття збереженої ЕТ.
2. Чим відрізняється уведення даних після одинарного і подвійного клацання клітинки?
3. Як скасувати й підтвердити процес уведення даних у клітинку?
4. Як побачити дані, які не поміщаються у клітинці?
5. Як вирівняти і як повернути запис у клітинці?
6. Як змінити ширину стовпця або висоту рядка ЕТ?

### Вправа 19



► Ознайомитися з методами уведення, редагування та форматування даних в електронній таблиці.

- 1) Запустіть Excel. Створіть новий документ з порожньою таблицею. Заповніть таблицю даними за зразком (рис. 19.8). Зверніть увагу на дані у рядку формул.

|    |       |          |      |                           |   |   |  |
|----|-------|----------|------|---------------------------|---|---|--|
| B1 |       | : X ✓ fx |      | Прізвище ім'я по батькові |   |   |  |
|    | A     | B        | C    | D                         | E | F |  |
| 1  | Номер | Прізвище | Дата |                           |   |   |  |

Рис. 19.8

- 2) Зменште приблизно втричі ширину стовпців A і C за зразком (рис. 19.9). З'ясуйте, чому текст у стовпці A видно не повністю, а у стовпці C — повністю.

|   |    |          |      |   |
|---|----|----------|------|---|
|   | A  | B        | C    | D |
| 1 | Но | Прізвище | Дата |   |

Рис. 19.9



- 3) Збільште приблизно в'ятеро висоту першого рядка та вчетверо ширину стовпця B. Запитайте в однокласників і однокласниць ПІБ і заповніть чотири клітинки за зразком (рис. 19.10).

|   |               |                                 |      |
|---|---------------|---------------------------------|------|
|   | A             | B                               | C    |
|   | Но<br>ме<br>р | Прізвище ім'я по батькові       | Дата |
| 1 |               |                                 |      |
| 2 |               | <i>Петренко Петро Петрович</i>  |      |
| 3 |               | <i>Іванов Іван Іванович</i>     |      |
| 4 |               | <i>Сумлінна Ліна Богданівна</i> |      |
| 5 |               | <i>Бойко Олена Вікторівна</i>   |      |

Рис. 19.10



- 4) Поверніть уміст клітинок верхнього рядка за зразком.
- 5) Вирівняйте вміст по горизонталі і вертикалі за зразком.
- 6) Установіть для ПІБ учнів і учениць колір — червоний, тип накреслення — жирний курсив. Збережіть документ із назвою Вправа 19. Надішліть листа вчителеві із поясненням до пункту 2 і вкладеним файлом Вправа 19.xlsx. Завершіть роботу.

### Комп'ютерне тестування



Виконайте тестове завдання 19 із автоматичною перевіркою результату.



## § 20. Типи даних в електронних таблицях

Важливою особливістю ЕТ є типізація даних у клітинках.

### Доцільність типізації даних

Призначенням ЕТ є опрацювання різноманітних даних.

Для зберігання й опрацювання, скажімо, текстових (рядкових) даних потрібні обсяг пам'яті і набір операцій, відмінні від зберігання й опрацювання даних числового типу.

- 1) Додавання текстів відрізняється від додавання чисел:

| Текст              | Число           |
|--------------------|-----------------|
| 123 & 456 = 123456 | 123 + 456 = 579 |

Порівняння текстів відрізняється від порівняння чисел:

| Дата            | Число     |
|-----------------|-----------|
| триста < сімсот | 300 < 700 |

Типізація даних в ЕТ необхідна для раціонального використання пам'яті, відображення даних у прийнятному форматі та можливості коректного опрацювання даних різного типу.



## Класифікація типів даних

За особливостями опрацювання та подання розрізняють такі дані:

| Тип даних | Опис                                                                                                                                   |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------|
| Числові   | Дані подано у вигляді числа, з яким можна проводити обчислення. Можуть містити допоміжні символи: %, \$ тощо                           |
| Текстові  | Дані подано у вигляді символічного рядка. До них не можна застосувати математичні обчислення                                           |
| Формули   | Арифметичні, логічні та/або інші вирази для опрацювання даних у клітинках. Можуть містити посилання, функції, позначення операцій тощо |

Якщо формат клітинки користувачем не було змінено, то в ній встановлено тип даних — Загальний. Після введення даних до такої клітинки їхній тип визначається табличним процесором автоматично.

### ► Текстові дані

Текст може складатися з будь-яких символів, зокрема цифр. Число в клітинці з текстовим типом вважається рядком, але до нього можуть бути застосовані числові операції.

### ► Числові дані

Числові дані можуть бути використані для обчислень. Їх вигляд на екрані залежить від вибраного формату.

Числовий тип даних може бути поданий у таких форматах:

| Формат         | Вигляд                      | Пояснення                                                                |
|----------------|-----------------------------|--------------------------------------------------------------------------|
| Числовий       | -41,53                      | Виведення чисел із вибраною кількістю десяткових знаків                  |
| Грошовий       | 431,53 ₴                    | До числа додається знак грошової одиниці                                 |
| Фінансовий     | 421,53 ₴<br>0,327 ₴         | Вирівнювання за роздільником цілої і дробової частин                     |
| Дробовий       | 3/7<br>1 31/49              | Звичайні дроби                                                           |
| Експоненційний | 5,78E+07                    | Стандартний вигляд числа $5,78 \cdot 10^7$                               |
| Дата           | 20.03.20<br>20 березня 2020 | Для відображення дати (не може мати від'ємного значення)                 |
| Час            | 13:36<br>20.03.20 13:36     | Для відображення часу або дати й часу (не може мати від'ємного значення) |

Користувач, опанувавши правила, може створювати власні формати виведення числових даних (усі формати).

Одиницею формату Дата та/або Час є доба. Формат Час розглядається програмою як частина доби, а Дата — як кількість днів від 01.01.1900 року. Якщо формат Час із відображенням дати і часу перевести у формат Числовий із десятковими знаками (або Дробовий), то побачимо дробове число, ціла частина якого позначає кількість днів від 01.01.1900 року, а дробова — частину доби, що пройшла від опівночі.

2 У числовому поданні час 12:30 є часткою доби  $\approx 0,52083$ , дата 01.03.2020 = 43891 днів, рахуючи від 01.01.1900.

Час 01.03.2020 12:30 у форматі Числовий (із п'ятьма десятковими знаками) 43891,52083.

Отже, залежно від формату число 43891,52083 може бути виведене як 01.03.2020 або 12:30, або 01.03.2020 12:30 тощо.

### ► Загальний формат

Як уже зазначалося, загальний формат у клітинках ЕТ установлено за замовчуванням. Excel визначає тип уведених даних автоматично. При цьому програма інколи помиляється.

3 Після введення тексту «Я вчуса» у клітинці побачимо: #NAME? (або #ИМЯ?). Це програма «вирішила», що введено від'ємне число, бо на початку стоїть знак «-».


Для коректного визначення даних у загальному форматі слід дотримуватися певних правил їх введення. Розглянемо деякі з них:

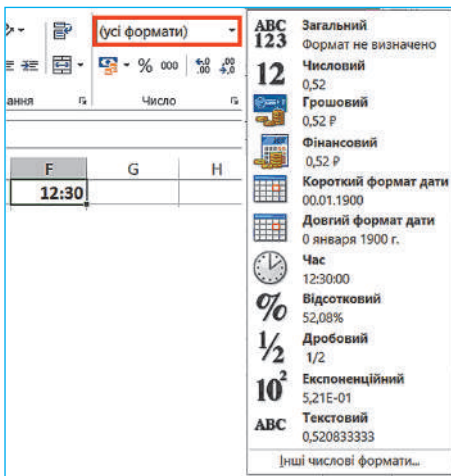
| Що вводиться    | Правило                                                                                       | Приклад    |
|-----------------|-----------------------------------------------------------------------------------------------|------------|
| Текст           | Введення розпочинати з апострофа: «'»                                                         | '-Я вчуса  |
| Відсотки        | Закінчувати введення знаком відсотка: «%»                                                     | 10%        |
| Звичайний дріб  | У форматі: ціла частина чисельник/знаменник (ціла частина вводиться завжди, навіть коли нуль) | 0 3/4      |
| Десятковий дріб | Ціла і дробова частини відокремлюються комою (крапкою — для англомовного Excel)               | 4,5        |
| Дата            | У форматі: день:місяць:рік (рік уводити не обов'язково)                                       | 15.12.2020 |
| Час             | У форматі: година:хвилина:секунда (секунди вводити не обов'язково)                            | 20:03:37   |

## Встановлення формату даних

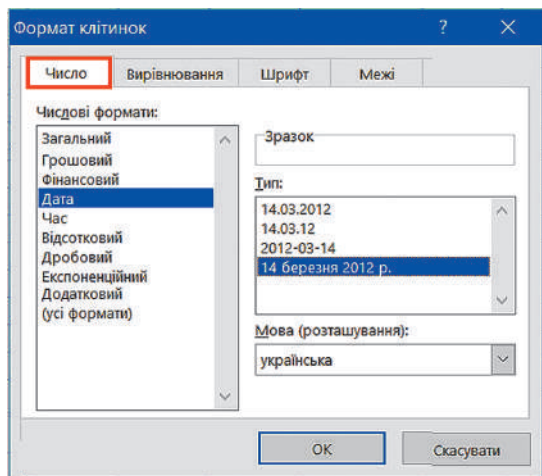
Щоб уникнути помилок даних #NAME?, користувач може заздалегідь установити потрібний формат (тип) даних. Для **встановлення формату** слід виділити потрібну клітинку чи діапазон і скористатись списком форматів (усі формати) на вкладці Основне.

Для **детальнішого налаштування формату даних** потрібно:

- 1) розгорнути значком  діалогове вікно налаштувань групи інструментів Число, що на вкладці Основне (рис. 20.1, а); можна також скористатися командою Формат клітинок контекстного меню клітинки;
- 2) вибрати потрібний формат (у нас — Дата) (рис. 20.1, б);
- 3) вибрати за потреби тип і мову (у нас тип — розлогий, мова — українська) (рис. 20.1, б);
- 4) натиснути кнопку ОК.



а



б

Рис. 20.1

Якщо розгорнути список форматів для виділеної клітинки з даними (див. рис. 20.1, а), то побачимо, як виглядатимуть ці дані у кожному з форматів.

Окрім розглянутих типів даних, аркуш електронної таблиці може містити також дані графічного типу: графіки, малюнки, діаграми, зображення, кнопки тощо. Але ці об'єкти не є даними клітинок, а розташовані «над таблицею».

## Питання для самоперевірки



1. У чому полягає необхідність типізації даних в ЕТ?
2. Назвіть типи даних ЕТ.
3. Охарактеризуйте числовий тип даних і його формати.
4. Опишіть особливості форматів числових даних для виведення дати і часу.
5. Охарактеризуйте загальний формат даних і назвіть кілька правил уведення даних різного типу.
6. Як установити потрібний формат даних клітинки?

## Вправа 20



► Ознайомитися з особливостями використання різних типів даних в електронній таблиці.

- 1) Запустіть Excel. Створіть Нову книгу.
- 2) Уведіть до однієї з клітинок текст =семи.  
Запишіть у зошит пояснення, який результат спостерігається після підтвердження введення.
- 3) Виправте дані у клітинці так, щоб відображався напис =семи.
- 4) Знайдіть в інтернеті точну дату початку повстання козаків на Запорізькій Січі проти Речі Посполитої й обрання гетьманом Богдана Хмельницького (рис. 20.2). Уведіть дату до наступної клітинки. Запишіть у зошит пояснення результату, який спостерігається у цій клітинці. Виправте дані у клітинці з датою повстання так, щоб дата відображалась коректно.
- 5) Уведіть до наступної клітинки поточну дату, встановіть для неї формат — числовий. Запишіть у зошит пояснення, який результат спостерігається у цій клітинці.
- 6) Збережіть документ із назвою Вправа 20. Надішліть листа вчителю із поясненнями, записаними в зошиті (п. 2, 4, 5) та вкладеним файлом Вправа 20.xlsx.  
Завершіть роботу.



Рис. 20.2



## Комп'ютерне тестування



Виконайте тестове завдання 20 із автоматичною перевіркою результату.



## § 21. Автоматичні дії над вмістом клітинок

Табличний процесор може виводити підказки під час уведення даних, змінювати дані, створювати нові, обчислені за формулами, тощо.

### Автозаповнення

Автозаповнення допомагає заповнювати однотипними даними суміжні клітинки списками та/або за заданими правилами.

#### ► Автозаповнення текстовими списками

Для застосування функції автозаповнення використовують маркер автозаповнення (рис. 21.1). Після перетягування маркера ЛКМ вздовж стовпця (рядка) суміжні клітинки діапазону заповнюються списком автозаповнення, якщо такий є (рис. 21.2), або скопійованим текстом з активної клітинки, якщо підходящого списку немає (рис. 21.3).

Маркер автозаповнення

|   | A | B |
|---|---|---|
| 1 |   |   |
| 2 |   |   |
| 3 |   |   |

Рис. 21.1

|   | A  | B |
|---|----|---|
| 1 | пн |   |
| 2 | вт |   |
| 3 | ср |   |
| 4 | чт |   |
| 5 | пт |   |
| 6 |    |   |

Рис. 21.2

|   | A        | B |
|---|----------|---|
| 1 | Меркурій |   |
| 2 | Меркурій |   |
| 3 | Меркурій |   |
| 4 | Меркурій |   |
| 5 | Меркурій |   |
| 6 |          |   |

Рис. 21.3

Автозаповнення текстовими списками може відбуватися через одне, два значення і т. д., тобто відбуватися з певним кроком.

1 На рис. 21.4 подано покрокове автозаповнення скороченими назвами днів тижня через день починаючи з вівторка.

Крок 1

|   | A  |
|---|----|
| 1 | вт |
| 2 | чт |
| 3 |    |
| 4 |    |
| 5 |    |
| 6 |    |

Увести значення

Крок 2

|   | A  |
|---|----|
| 1 | вт |
| 2 | чт |
| 3 |    |
| 4 |    |
| 5 |    |
| 6 |    |

Виділити

Крок 3

|   | A  |
|---|----|
| 1 | вт |
| 2 | чт |
| 3 | сб |
| 4 | пн |
| 5 | ср |
| 6 | пт |

Протягнути маркер

Рис. 21.4

Якщо протягнути маркер правою кнопкою миші, то з'являється контекстне меню, в якому можна вибрати параметри автозаповнення.

► **Автозаповнення обчислювальними значеннями**

Дані в клітинках, над якими можна проводити обчислення, або текстові дані з цілочисельними значеннями на початку чи в кінці рядка називатимемо *обчислювальними*.

Якщо під час автозаповнення числовими послідовностями, або обчислювальними даними утримувати натиснутою клавішу Ctrl, то замість автозаповнення відбувається копіювання даних (рис. 21.5, а).

Для автозаповнення даними однієї клітинки зростаючою числовою послідовністю з кроком 1 слід протягнути маркер, утримуючи клавішу Ctrl, а для копіювання даних — маркер без утримання Ctrl (рис. 21.5, б).



Рис. 21.5

► **Створення власного списку автозаповнення**

Для створення списку автозаповнення потрібно:

- 1) виконати Файл → Параметри → Додатково → Редагувати користувацькі списки (рис. 21.6);

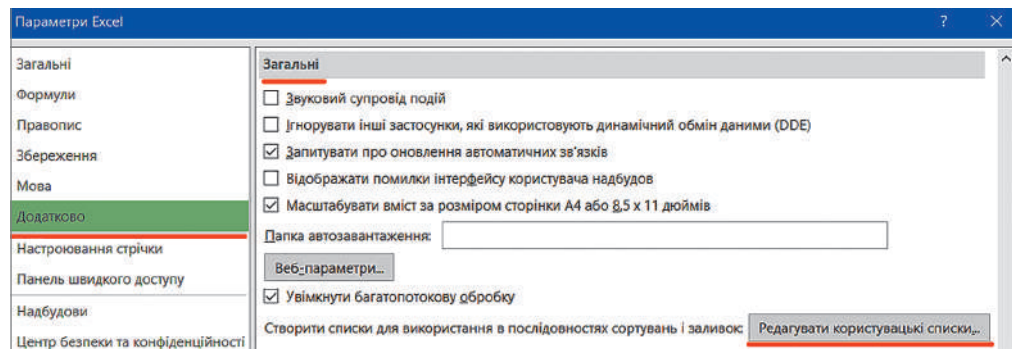


Рис. 21.6

- 2) у вікні Списки в полі Елементи списку ввести потрібні дані (у стовпець або рядок через кому);
- 3) натиснути кнопку Додати та/або ОК (рис. 21.7).

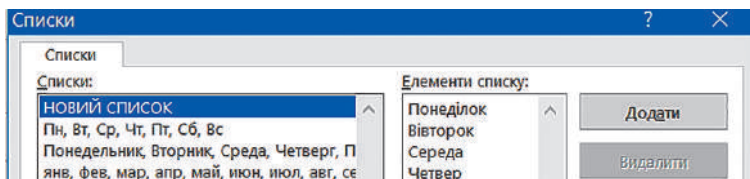


Рис. 21.7

## Автозавершення

Функція автозавершення застосовується тільки для **текстових** даних. Після набору перших символів Excel пропонує завершення слів із числа наявних у стовпці.

- 2 Після того як у клітинку буде введено літери «карто» на сірому тлі, пропонується варіант автозавершення «пля» (рис. 21.8). Після натискання клавіші Enter або Tab у клітинці з'явиться слово «картопля». Якщо ввести меншу кількість літер, варіант автозавершення не з'явиться.

|          |
|----------|
| карета   |
| картуз   |
| картопля |
| картопля |

Рис. 21.8

Слід пам'ятати, що функція автозавершення аналізує лише суміжні клітинки з даними та не враховує регістр символів, що вводяться.

Щоб відмовитися від послуг автозавершення, треба введення тексту розпочати з апострофа (').

У контекстному меню клітинки можна вибрати потрібне слово із суміжних у стовпці клітинок командою Вибрати з розкритого списку (рис. 21.9).

|          |
|----------|
| карета   |
| картуз   |
| картопля |
| картопля |
| карета   |
| картопля |
| картуз   |

Рис. 21.9

## Питання для самоперевірки



1. Що таке автозаповнення? Наведіть приклади.
2. Як створити (додати) новий список автозаповнення?
3. Як заповнити стовпець послідовністю чисел із певним кроком?
4. Як скопіювати числову послідовність без автозаповнення?
5. Що таке автозавершення?
6. Наведіть приклади автозавершення.

## Вправа 21



► Ознайомитися з автозаповненням і автозавершенням під час уведення даних до ЕТ.

- 1) Запустіть Excel. Створіть Нову книгу. Заповніть клітинки стовпця А натуральними числами від 1 до 100 за зразком (рис. 21.10).

|     | A   | B   | C         | D        | E        | Z   | AA   |
|-----|-----|-----|-----------|----------|----------|-----|------|
| 1   | 1   | 200 | півень    | січень   | Меркурій | Вен | урн  |
| 2   | 2   | 198 | південь   | лютий    |          |     | Уран |
| 3   | 3   | 196 | півкуля   | березень |          |     |      |
| 4   | 4   | 194 | півострів | квітень  |          |     |      |
| 5   | 5   | 192 | півень    | травень  |          |     |      |
| 6   | 6   | 190 | півкуля   | червень  |          |     |      |
| 7   | 7   | 188 | південь   | липень   |          |     |      |
| 8   | 8   | 186 | півострів | серпень  |          |     |      |
| 9   | 9   | 184 | півень    | вересень |          |     |      |
| 10  | 10  | 182 | південь   | жовтень  |          |     |      |
| 11  | 11  | 180 | півкуля   | листопад |          |     |      |
| 12  | 12  | 178 | південь   | грудень  |          |     |      |
| 13  | 13  | 176 |           | січень   |          |     |      |
| 98  | 98  | 8   |           | лютий    |          |     |      |
| 99  | 99  | 4   |           | березень |          |     |      |
| 100 | 100 | 2   |           | квітень  |          |     |      |

Рис. 21.10

- 2) Заповніть клітинки стовпця В парними числами від 200 до 2.
- 3) Заповніть 12 клітинок стовпця С автозавершенням за зразком.
- 4) Заповніть назвами місяців стовпець D. За потреби створіть список автозаповнення з назвами місяців і повторіть спробу.
- 5) Пригадайте або знайдіть в інтернеті назви планет Сонячної системи. Створіть список автозаповнення з назвами планет.
- 6) Заповніть за зразком рядок 1 від стовпця E до AA назвами планет. Збережіть файл із назвою Вправа 21. Надішліть листа вчителю із вкладеним файлом Вправа 21.xlsx. Завершіть роботу.



## Комп'ютерне тестування



Виконайте тестове завдання 21 із автоматичною перевіркою результату.





## § 22. Формули в електронних таблицях

Для опрацювання даних потрібно зазначити, в яких клітинках таблиці містяться ці дані.

### Адресація

Кожна клітинка (діапазон) ЕТ має свою унікальну адресу.

#### ► Адреси й імена клітинок

Адреса клітинки складається із заголовка стовпця (А, В, С...) і номера рядка (1, 2, 3...), на перетині яких вона знаходиться. Адреса активної клітинки відображається у полі Ім'я. Якщо у ньому на місці адреси клітинки ввести ім'я, яке не збігатиметься з наявним або адресою, натиснути Enter, то це ім'я присвоюється виділеній клітинці (діапазону).

#### ► Адреси діапазонів

Адреса *зв'язного* діапазону складається з адрес двох клітинок: верхньої зліва і нижньої справа, розділених двокрапкою.

Адреса *незв'язного* діапазону складається з окремих адрес клітинок і/або зв'язних діапазонів, розділених крапкою з комою.

**1** Розглянемо рис. 22.1. Адреса клітинки на перетині стовпця А і рядка 2 — А2 (*а*); адреса зв'язного діапазону із чотирьох клітинок — В2:С3 (*б*); адреса незв'язного діапазону, який об'єднує два перших приклади — А2;В2:С3 (*в*); адреса третього стовпця — С (замість діапазону С1:С1048576).

|   | A | B |
|---|---|---|
| 1 |   |   |
| 2 |   |   |
| 3 |   |   |

а

|   | A | B | C |
|---|---|---|---|
| 1 |   |   |   |
| 2 |   |   |   |
| 3 |   |   |   |

б

|   | A | B | C |
|---|---|---|---|
| 1 |   |   |   |
| 2 |   |   |   |
| 3 |   |   |   |

в

Рис. 22.1

### Формули

Особливістю ЕТ є можливість використання формул і функцій.

#### ► Поняття формули



**Формулою** в електронній таблиці називають послідовність символів, що починається зі знака рівності.

У формулах можуть використовуватися числа, адреси або імена клітинок і/або діапазонів, круглі дужки, функції та знаки порівнянь і арифметичних дій:

- + додавання;
- віднімання;
- \* множення;
- / ділення;
- ^ піднесення до степеня;
- % обчислення відсотків.

► **Правила запису формул**

Запис формули починають знаком «=», далі йде сама формула.

Для внесення у формулу адреси клітинки достатньо клацнути відповідну клітинку.

2 Для розрахунку підсумкової оцінки (середнього балу) Сумлінної Ліни знайдемо суму оцінок і поділимо на їх кількість (рис. 22.2).

|   | A             | B      | C  | D | E | F | G             |
|---|---------------|--------|----|---|---|---|---------------|
| 1 | Прізвище      | Оцінки |    |   |   |   | Підсумкова    |
| 2 | Сумлінна Ліна | 12     | 11 |   | 9 |   | =(B2+C2+E2)/3 |

Рис. 22.2

У результаті після введення формули та натискання клавіші Enter у клітинці з формулою відобразиться результат обчислень.

Для приховування дробової частини числа (округлення) можна зменшити ширину стовпця (рис. 22.3).

|   |               |    |    |  |   |  |    |
|---|---------------|----|----|--|---|--|----|
| 2 | Сумлінна Ліна | 12 | 11 |  | 9 |  | 11 |
|---|---------------|----|----|--|---|--|----|

Рис. 22.3

► **Використання функцій**

Застосована у прикладі 2 формула має суттєвий недолік: якщо учениця отримає нову оцінку, доведеться переробляти формулу. Зробити формулу однаковою для будь-якої кількості оцінок можна завдяки застосуванню функцій.



**Функція** в електронній таблиці — це іменоване позначення певних дій над даними таблиці.

Дані, які опрацьовує функція, вказують у дужках після назви функції і називають **аргументами**. У формулах для опрацювання даних є багато вбудованих функцій. Розглянемо застосування функції AVERAGE (у російськомовному інтерфейсі — назва СРЗНАЧ).

Функція AVERAGE повертає середнє арифметичне числових даних.

- 3 Для розрахунку підсумкової оцінки застосуємо функцію AVERAGE (рис. 22.4). Як аргумент задамо діапазон B2:F2. Порожні клітинки і клітинки з даними нечислового типу функцією нехтуються.

|   | A             | B      | C  | D | E | F               | G |
|---|---------------|--------|----|---|---|-----------------|---|
| 1 | Прізвище      | Оцінки |    |   |   | Підсумкова      |   |
| 2 | Сумлінна Ліна | 12     | 11 |   | 9 | =AVERAGE(B2:F2) |   |

Рис. 22.4

### ► Копіювання формул

Копіювання формул дозволяє прискорити їх уведення і модифікацію. Копіювання здійснюється за допомогою команд контекстного меню клітинки, або сполученнями клавіш Ctrl + C, Ctrl + V, або перетягуванням маркера автозаповнення.



Імена та адреси клітинок і діапазонів, використані у формулах, називають **посиланнями**.

Посилання, які змінюються під час переміщення та/або копіювання формул, називають **відносними**.

Під час копіювання чи переміщення формул відбувається модифікація відносних посилань у нових формулах (рис. 22.5).

Для копіювання формули до суміжних клітинок рядка (стовпця) можна виділити клітинку з формулою й перетягнути маркер автозаповнення. У формулах відносні посилання будуть модифікованими (рис. 22.6).

|   | A | B   | C   |
|---|---|-----|-----|
| 1 |   | =A1 |     |
| 2 |   |     |     |
| 3 |   |     | =B3 |

Ctrl + C      Ctrl + V

Рис. 22.5

|     |     |     |
|-----|-----|-----|
| =B2 | =C2 | =D2 |
| =B3 |     |     |
| =B4 |     |     |
| =B5 |     |     |

Перетягнути маркер

Рис. 22.6

Редагування формул відбувається аналогічно редагуванню даних іншого типу: у рядку формул або після подвійного клацання — у клітинці з формулою.

- 4 Щоб розрахувати підсумкову оцінку кільком учням, створену формулу можна скопіювати маркером автозаповнення (рис. 22.7).

|   | A               | B      | C  | D | E  | F                | G |
|---|-----------------|--------|----|---|----|------------------|---|
| 1 | Прізвище        | Оцінки |    |   |    | Підсумкова       |   |
| 2 | Сумлінна Ліна   | 12     | 11 |   | 9  | n=AVERAGE(B2:F2) |   |
| 3 | Петренко Богдан | 8      |    | 7 |    |                  |   |
| 4 | Іванов Петро    |        | 12 |   | 11 | 11               |   |
| 5 | Калиніна Дарина | 9      | 8  | 9 | 8  |                  |   |

Перетягнути маркер

Рис. 22.7

► Абсолютні та мішані посилання

Іноді формули містять посилання на значення, яке після переміщення чи копіювання формул не повинно змінюватись.

Посилання, які не змінюються під час копіювання формул, називають **абсолютними** (рис. 22.8).

Посилання, у яких під час копіювання формул не змінюється одна з величин: номер рядка або заголовки стовпця називають **мішаними**.

Абсолютні та/або мішані посилання зліва від заголовка стовпця та/або рядка містять символ \$:

|   | A | B     | C     |
|---|---|-------|-------|
| 1 |   | =A\$1 |       |
| 2 |   |       |       |
| 3 |   |       | =A\$1 |

Ctrl + C      Ctrl + V

Рис. 22.8

| Відносне посилання | Абсолютне посилання | Мішані посилання |
|--------------------|---------------------|------------------|
| A1                 | \$A\$1              | \$A1    A\$1     |

Для автоматичної зміни типу посилань необхідно встановити курсор на посилання у формулі та натиснути клавішу F4.

При копіюванні формул із *мішаними* посиланнями модифікуються лише відносні номери рядків та/або заголовки стовпців.

5 Переведемо підсумкові оцінки із 12-бальної системи в N-бальну на основі прямої пропорційності, тобто 12 — це N балів, 6 — N/2 балів і т. д. (рис. 22.9).

|   | A               | B      | C  | D | E  | F  | G          | H               | I          |
|---|-----------------|--------|----|---|----|----|------------|-----------------|------------|
|   |                 |        |    |   |    |    | Підсумкова | N-бальна оцінка | Коефіцієнт |
| 1 | Прізвище        | Оцінки |    |   |    |    |            |                 | N          |
| 2 | Сумлінна Ліна   | 12     | 11 |   | 9  | n  | 11         | =I\$2*G2/12     | 100        |
| 3 | Петренко Богдан | 8      |    | 7 |    |    | 8          |                 | 63         |
| 4 | Іванов Петро    |        | 12 |   | 11 | 11 | 11         |                 | 94         |
| 5 | Калиніна Дарина | 9      | 8  | 9 | 8  |    | 9          |                 | 71         |

Рис. 22.9

### ► Переміщення та копіювання адресних клітинок

Клітинки, на які у формулах зроблено посилання, називатимемо **адресними**.

При *копіюванні даних з адресних клітинок* до інших клітинок посилання на них у формулах не модифікуються.

При *переміщенні адресних клітинок* усі посилання на них модифікуються залежно від їхнього нового розташування (адреси) і відповідно до типу посилань (рис. 22.10).

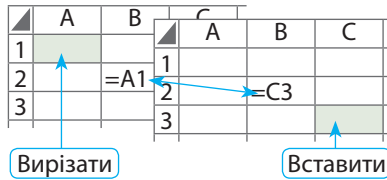


Рис. 22.10

Посилання також модифікуються під час переміщення адресної клітинки внаслідок вставлення або видалення стовпців, рядків, клітинок або діапазонів клітинок, що містяться вище або ліворуч від адресної.

Видалення адресних клітинок призводить до помилок у формулах, які містять посилання на видалені клітинки (рис. 22.11).



Рис. 22.11

### Питання для самоперевірки



1. Що являє собою адреса клітинки, діапазону, стовпця, рядка?
2. Чим формула відрізняється від інших даних ЕТ?
3. Що таке функція в ЕТ? Наведіть приклад її використання.
4. Що таке посилання у формулах ЕТ? У чому відмінність запису відносних і абсолютних посилань?
5. Яким чином можна копіювати формули до інших клітинок і що при цьому відбувається з посиланнями в них?
6. Що відбувається з формулами під час копіювання і переміщення адресних клітинок?

## Вправа 22



► Ознайомитися з обчисленнями числових даних ЕТ за допомогою формул і функцій.

- 1) Запустіть Excel. Створіть Нову книгу. Заповніть клітинки таблиці за зразком (рис. 22.12), окрім діапазонів D2:D7 і F2:F8. Тариф заповнюйте цілими числами або дробовими з одним десятковим знаком.

|   | A                            | B       | C         | D       | E       | F          |
|---|------------------------------|---------|-----------|---------|---------|------------|
| 1 | Показники лічильників        | Поточні | Попередні | Спожито | Тариф   | До сплати  |
| 2 | Електроенергія               | 7483    | 7396      | 87      | 0,9000  | 78,3000    |
| 3 | Газ                          | 2765    | 2722      | 43      | 6,5000  | 279,5000   |
| 4 | Гаряча вода                  | 1261    | 1244      | 17      | 31,0000 | 527,0000   |
| 5 | Холодна вода                 | 4589    | 4568      | 21      | 16,0000 | 336,0000   |
| 6 | Опалення                     | 3908    | 3783      | 125     | 5,7000  | 712,5000   |
| 7 | Водовідведення (каналізація) |         |           | 38      | 15,8000 | 600,4000   |
| 8 |                              |         |           |         | Всього: | 2 533,7000 |

Рис. 22.12

- 2) У діапазоні E2:F7 установіть грошовий формат гривні з трьома знаками після коми.
- 3) У клітинці D2 створіть формулу розрахунку кількості спожитої електроенергії за показами лічильників. Скопіюйте формулу для комунальних послуг (окрім водовідведення). У D7 створіть формулу розрахунку кількості відведеної води, що дорівнює загальному обсягу спожитої холодної і гарячої води.
- 4) У клітинці F2 створіть формулу підрахунку вартості спожитої електроенергії за тарифом (E2) та обсягом спожитої енергії (D2). Скопіюйте формулу для решти комунальних послуг.
- 5) У клітинці F8 створіть формулу підрахунку загальної вартості усіх комунальних послуг.
- 6) Знайдіть в інтернеті тарифи за відповідні комунальні послуги у вашому регіоні на поточний місяць. Змініть табличні дані у діапазоні E2:E6 згідно з тарифами з точністю до тисячних. Збережіть документ із назвою Вправа 22. Надішліть листа вчителеві із вкладеним файлом Вправа 22.xlsx. Завершіть роботу.



## Комп'ютерне тестування



Виконайте тестове завдання 22 із автоматичною перевіркою результату.




## § 23. Форматування електронної таблиці

Вам відомо, що дії, пов'язані з оформленням документа, називають форматуванням. Після створення ЕТ її потрібно відформатувати, щоб вона коректно відображалася під час перегляду на моніторі та/або в друкованому вигляді, була естетично оформленою, зрозумілою і легкою для сприйняття людиною.

### Встановлення параметрів сторінки

Для встановлення параметрів сторінки використовують уже знайомі вам інструменти вкладки Розмітка сторінки. Так, для налаштування розмірів полів у групі інструментів Параметри сторінки вибирають інструмент Поля, далі — значення у списку або налаштовують розміри полів вручну командою Настроювані поля.

Більш детальні налаштування параметрів сторінки можна здійснити у вікні Параметри сторінки, яке розгортається кнопкою  у правому нижньому кутку відповідної групи інструментів (рис. 23.1).

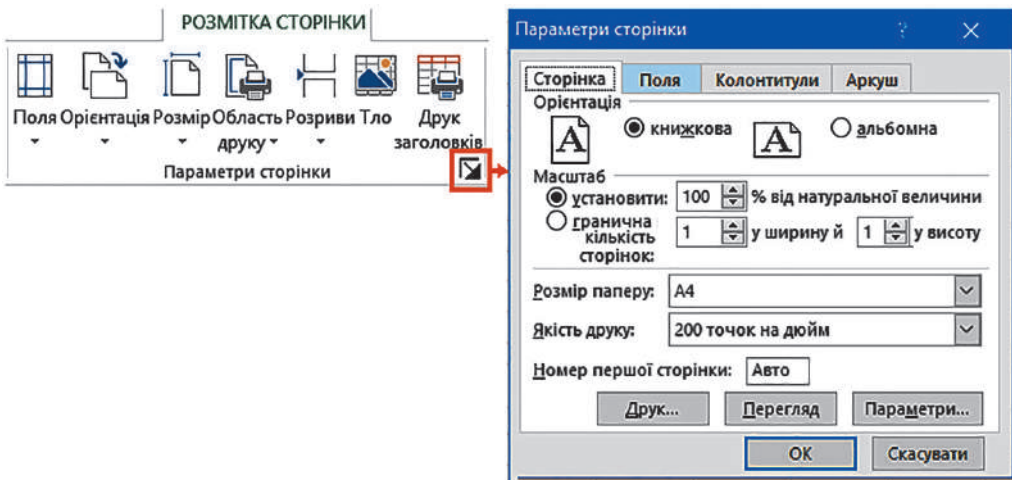


Рис. 23.1

Налаштування параметрів сторінки подібне до налаштування параметрів текстових сторінок у відомому вам текстовому процесорі Word.

## Встановлення розмірів стовпців і рядків

Ви вже знаєте, що для того, щоб змінити ширину стовпця чи висоту рядка, потрібно перетягувати межу між їхніми заголовками.

**1** Якщо потрібно змінити ширину стовпця так, щоб вона збігалася з шириною найдовшого рядка в одній із клітинок, то достатньо двічі клацнути праву межу цього стовпця у рядку заголовків.

Щоб встановити однакову ширину кількох стовпців або висоту кількох рядків, ці стовпці або рядки потрібно виділити, а потім перетягнути одну з меж або задати потрібне значення в контекстному меню **Ширина стовпця (Висота рядка)**.

Висоту (ширину) виділених рядків (стовпців) можна встановити також за допомогою відповідних команд контекстного меню.

Під час підготовки документа до друку зручно працювати в режимі **Вигляд → Макет сторінки**. У цьому режимі лінійки градуюються в сантиметрах і під час перетягування чи клацання меж рядків та стовпців їхні розміри зазначаються в сантиметрах.

Зазвичай не завжди можна встановити точні розміри рядків та стовпців у сантиметрах, оскільки розмір об'єктів ЕТ залежить від роздільної здатності монітора.

**2** Якщо за роздільної здатності монітора 21" 1920×1080 пікселів задати ширину стовпця в 0,5 см, то замість заданого значення отримаємо найбільш близьке до нього — 0,49 см (рис. 23.2).

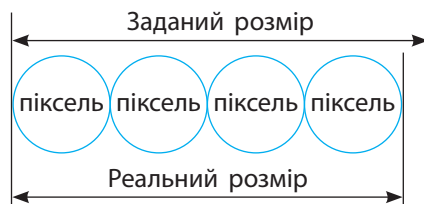



Рис. 23.2

## Об'єднання клітинок

Щоб зробити, наприклад, спільний заголовок для кількох стовпців таблиці, можна об'єднати кілька клітинок в одну. Для об'єднання клітинок потрібно їх виділити і вибрати на вкладці **Основне** в групі **Вирівнювання** кнопку  **Об'єднати й помістити у центрі**. В об'єднаній клітинці залишаться тільки дані клітинки, розташованої зверху зліва.

Для скасування об'єднання клітинок потрібно повторно клацнути ту саму кнопку. При цьому необхідно пам'ятати, що дані об'єднаних раніше клітинок не відновлюються.



### Розташування тексту в кілька рядків

У випадку коли текст не поміщається у клітинці, в Excel є можливість розташувати його в кілька рядків.

Для встановлення/відключення режиму автоматичного перенесення тексту в кілька рядків треба виконати такі дії:

- 1) виділити клітинку або діапазон;
- 2) на вкладці Основне у групі інструментів Вирівнювання клацнути інструмент Перенести текст

### Видалення та додавання рядків, стовпців, діапазонів

Видалити рядки або стовпці разом із даними в них можна так:

- 1) виділити один або кілька рядків (стовпців);
- 2) на виділеному рядку (стовпці) викликати контекстне меню;
- 3) у меню вибрати команду Видалити.

Щоб вставити порожній рядок або стовець, потрібно:

- 1) у таблиці виділити один або кілька рядків (стовпців), вище (лівише) якого необхідно вставити порожній рядок (стовпець);
- 2) на виділеному рядку (стовпці) викликати контекстне меню;
- 3) у меню вибрати команду Додати клітинки.

Видалення клітинок та/або діапазонів відбувається аналогічно видаленню рядків і стовпців із тією відмінністю, що користувач сам вказує, як має зсуватися решта клітинок таблиці на місце видалених.

Видалення даних з активних клітинок не призводить до видалення рядків, стовпців, клітинок.

### Встановлення меж клітинок

Таблицю, яку ми бачимо на робочому аркуші, розбито на клітинки умовно. Але якщо таку таблицю роздрукувати, на папері відобразяться лише дані в клітинках, не буде жодної лінії або нумерації.

Межі клітинок можна встановити інструментом , що на вкладці Основне у групі Шрифт.

Для форматування меж виділених клітинок необхідно:

- 1) на вкладці Основне кнопкою Вирівнювання відкрити вікно Формат клітинок;
- 2) у вікні Формат клітинок на вкладці Межі вибрати потрібний формат меж: тип, колір, розташування.

## Друк таблиці

Користувачам досить часто доводиться друкувати ЕТ на папері.

Для коректного відображення ЕТ у друкованому вигляді потрібно встановити параметри сторінки, оптимізувати таблицю форматуванням її об'єктів і встановити межі клітинок у таблиці.

Друк таблиці Excel здійснюється подібно до друку документа у текстовому процесорі Word.

- ! На папері друкується частина ЕТ, обмежена клітинками з наявними даними та/або зі встановленими межами.

До друку документа можна перейти за допомогою команди Файл → → Друк або сполученням клавіш Ctrl + P.

## Робота з аркушами

Документ Excel за замовчуванням називається **Книгою**. Книга містить щонайменше один аркуш із таблицею. Проте книга Excel може складатися з кількох аркушів.

Для додавання нових аркушів потрібно скористатися кнопкою ⊕ (рис. 23.3). Для переходу з одного аркуша на інший слід клацнути відповідну вкладку.



Рис. 23.3

За допомогою контекстного меню вкладки аркушів можна перейменувати, зафарбовувати, видаляти, копіювати тощо. Взаємне розташування вкладок можна змінювати, протягуючи їх мишею. Якщо вкладку перемістити з утриманням клавіші Ctrl, то створиться копія аркуша разом із наявними даними.

## Питання для самоперевірки



1. Назвіть дії з ЕТ, які можна назвати форматуванням документа.
2. Як встановити довільні поля сторінки?
3. Як встановити розмір рядка (стовпця) у сантиметрах?
4. Як об'єднати кілька клітинок і що при цьому відбувається з даними в них?
5. Як видалити або додати рядок, стовпець, клітинку?
6. Як встановити й відформатувати потрібні межі клітинок і діапазонів клітинок?

## Вправа 23



►► Здійснити форматування ЕТ, дотримуючись заданих умов.

- 1) Відкрийте документ Вправа 22.xlsx. Поверніть (зорієнтуйте) написи за зразком (рис. 23.4) і встановіть ширину всіх стовпців за найдовшим рядком даних у них.

|    | A                                                   | B                  | C         | D       | E        | F          |
|----|-----------------------------------------------------|--------------------|-----------|---------|----------|------------|
| 1  | <b>Рахунок на оплату житлово-комунальних послуг</b> |                    |           |         |          |            |
| 2  | Комунальна послуга                                  | Покази лічильників |           |         | Тариф    | До сплати  |
| 3  |                                                     | Поточні            | Попередні | Спожито |          |            |
| 4  |                                                     | Електроенергія     | 7483      | 7396    |          |            |
| 5  | Газ                                                 | 2765               | 2722      | 43      | 6,500 €  | 279,50 €   |
| 6  | Гаряча вода                                         | 1261               | 1244      | 17      | 31,000 € | 527,00 €   |
| 7  | Холодна вода                                        | 4589               | 4568      | 21      | 16,000 € | 336,00 €   |
| 8  | Опалення                                            | 3908               | 3783      | 125     | 5,700 €  | 712,50 €   |
| 9  | Водовідведення (каналізація)                        |                    |           | 38      | 15,800 € | 600,40 €   |
| 10 |                                                     |                    |           |         | Всього:  | 2 533,70 € |

Рис. 23.4



- 2) Додайте зверху два порожніх рядки, об'єднайте потрібні клітинки за зразком. Додайте та/або перемістіть необхідні написи в таблиці за зразком. За потреби змініть ширину стовпців.
- 3) Вирівняйте написи по горизонталі та вертикалі й розташуйте, де потрібно, текст у кілька рядків за зразком.
- 4) Установіть межі таблиці за зразком (зовнішні — найтовщі, внутрішні — найтонші).



- 5) Відформатуйте тексти і тло клітинок за зразком.
- 6) У діапазоні F4:F10 установіть грошовий формат гривні з двома знаками після коми. Збережіть документ із назвою Вправа 23. Надішліть листа вчителю із вкладеним файлом Вправа 23.xlsx. Завершіть роботу.

## Комп'ютерне тестування



Виконайте тестове завдання 23 із автоматичною перевіркою результату.





## Практична робота 6

### Створення й опрацювання електронної таблиці

**Завдання:** створити ЕТ зі сторінкою обліку навчальних досягнень учнів, як у шкільному журналі успішності, з автоматичним підрахунком тематичної оцінки.

**Обладнання:** комп'ютер зі встановленим табличним процесором.

#### Хід роботи

*Під час роботи з комп'ютером дотримуйтеся правил безпеки.*

- ▶ **1.** Запустіть Excel. Створіть Нову книгу, перейменуйте Аркуш1 на Практична 6.  
Установіть: поля — по 1 см; розмір аркуша — А4; колонтитули — 0 см; орієнтація — книжкова.  
Установіть режим відображення: Вигляд → Макет сторінки. Виділіть усю таблицю й установіть: шрифт — Times New Roman; розмір — 11.
- ▶ **2.** Установіть ширину стовпців А і В — по 0,7 і 6,3 см відповідно, а для стовпців від С до S включно — по 0,7 см.
- ▶ **3.** Установіть висоту рядків з 1-го по 46-й включно — 0,5 см. Змініть висоту 3-го і 4-го рядків на 1 і 1,9 см відповідно.
- ▶ **4.** Об'єднайте клітинки кожного з діапазонів С1:S2; А3:A4; В3:В4; С3:S3.
- ▶ **5.** У клітинці В1 відобразіть нижню межу мінімальної товщини. У діапазоні А3:S46 установіть усі межі, окрім бокових, мінімальної товщини (бокові межі відсутні). В об'єднаній клітинці В3 встановіть діагональну межу середньої товщини.
- ▶ **6.** У діапазоні А3:S4 установіть горизонтальні межі максимальної товщини. В діапазоні В3:В46 установіть вертикальні межі середньої товщини.
- ▶ **7.** Відповідно до зразка, наведеного на рисунку, додайте написи в клітинках. У клітинці В3 написи доцільно робити в режимі Вставлення → Текстове поле.
- ▶ **8.** Відформатуйте написи: встановіть розмір шрифту, стиль накреслення, вирівнювання, орієнтацію тощо за зразком.

Впишіть прізвища й імена кількох однокласників і однокласниць, біля кожного прізвища впишіть кілька оцінок.

| Клацніть, щоб додати верхній колонтитул |                                 |                                  |          |          |          |          |          |  |  |  |  |  |           |
|-----------------------------------------|---------------------------------|----------------------------------|----------|----------|----------|----------|----------|--|--|--|--|--|-----------|
| (назва предмета)                        |                                 | Облік навчальних досягнень учнів |          |          |          |          |          |  |  |  |  |  |           |
| № з/п                                   | Прізвище та ім'я учня (учениці) | Місяць і число                   |          |          |          |          |          |  |  |  |  |  | Тематична |
|                                         |                                 | 10.02.20                         | 17.02.20 | 24.02.20 | 02.02.20 | 09.02.20 | 16.02.20 |  |  |  |  |  |           |
| 1                                       | Петренко Богдан                 | 12                               |          | 11       | н        | 2        |          |  |  |  |  |  | 8         |
| 2                                       | Сумлінна Ліна                   | 12                               | 10       |          | 8        |          |          |  |  |  |  |  | 10        |
| 3                                       | Богданов Петро                  | 7                                |          | 9        | 5        |          |          |  |  |  |  |  | 7         |
| 4                                       | Калігіна Марійка                | 12                               | н        | 12       |          | 12       |          |  |  |  |  |  | 12        |
| 5                                       |                                 |                                  |          |          |          |          |          |  |  |  |  |  |           |
| 43                                      | 39                              |                                  |          |          |          |          |          |  |  |  |  |  |           |
| 44                                      | 40                              |                                  |          |          |          |          |          |  |  |  |  |  |           |
| 45                                      | 41                              |                                  |          |          |          |          |          |  |  |  |  |  |           |
| 46                                      | 42                              |                                  |          |          |          |          |          |  |  |  |  |  |           |

- ▶ 9. У діапазоні C4:R4 установіть тип даних Дата у форматі дд.мм.рр., у діапазоні C4:S4 — орієнтацію написів знизу вгору.
- ▶ 10. Впишіть автозаповненням дати шести уроків поспіль і слово тематична.
- ▶ 11. Створіть формули розрахунку тематичної оцінки кожному учневі або учениці, використовуючи автозаповнення. Встановіть формат числових даних для тематичних оцінок без десяткових знаків.
- ▶ 12. Збережіть файл із назвою Практична 6. Надішліть поштою вкладений файл Практична 6.xlsx на перевірку вчителів. Завершіть роботу.

Зробіть висновок про результати роботи.

## § 24. Моделі і моделювання

Всесвіт, що нас оточує та частиною якого є ми самі, складається з об'єктів (предметів, процесів і явищ). Для можливості дослідження або споглядання чи використання об'єктів Всесвіту люди вдаються до моделювання.

### Поняття і типи моделей

Вам відомо, що модель (від латин. *modulus* — міра, зразок) — це спрощене подання об'єкта, процесу або явища.

Заміну об'єкта, процесу чи явища його моделлю називають **моделюванням**.

За способом реалізації моделі можна розділити на матеріальні (або натурні) й інформаційні (або абстрактні).

1 Якщо на уроці образотворчого мистецтва ви зліпили будиночок із пластиліну, а потім намалювали його, то можна сказати, що спочатку ви створили *матеріальну* (рис. 24.1, а), а потім *інформаційну* (рис. 24.1, б) модель будинку.

Ці моделі зберігають лише деякі властивості будинку. У них немає кам'яного фундаменту, скляних вікон, дерев'яної підлоги, системи опалення та ще багатьох речей, які є у справжніх будинках.



а



б

Рис. 24.1

Створюючи модель, людина відтворює потрібні їй властивості об'єкта й ігнорує несуттєве. Тому для одного об'єкта можна побудувати багато різних моделей.

Вам відомо, що **матеріальні моделі** можуть відтворювати певні геометричні, фізичні, конструктивні чи функціональні властивості об'єктів.



**Інформаційна модель** — це опис важливих для певного дослідження властивостей об'єкта (явища, процесу).

Інформаційні моделі можна поділити на такі типи: графічні, спеціальні, алгоритмічні, словесні, структурні та математичні.

Поміркуємо над причинами, що спонукають до моделювання. Однією з причин моделювання є те, що реалізація моделей дає змогу відтворити явища, які в реальних земних умовах людині відтворити не під силу. Це, наприклад, рух материків, дія землетрусів, народження нової зірки, зміна напрямків морських підводних течій тощо.

Деякі причини, які спонукають людей замінити об'єкти чи явища їхніми моделями, подано на форзаці 2 підручника.

## Математичне моделювання

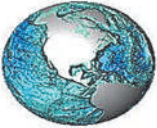
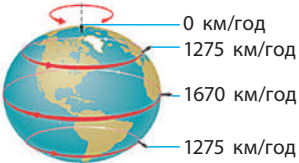
Математичне моделювання можна застосовувати, коли властивості об'єкта чи явища підлягають описанню математичними формулами.



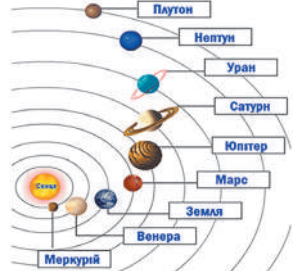
Ви вже використовували математичну модель як систему математичних співвідношень, які описують властивості досліджуваного об'єкта, процесу або явища.

**2** На уроках фізики ви ознайомилися з поняттями сили тяжіння і ваги тіла. Знаєте, що, якщо тіло перебуває в стані спокою або прямолінійного рівномірного руху, то його вага збігається за напрямком із силою тяжіння і дорівнює їй за значенням:  $P = mg$ . Це математична модель ваги тіла.

### ► Спрощення в математичному моделюванні

На прикладі математичної моделі сили тяжіння і ваги розглянемо кілька чинників, які не враховуються під час моделювання (див. таблицю).

| Що не враховується                                                                                         | Пояснення                                                          | Вплив                                                                                        |
|------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
| <p>Форма Землі</p>      | Земля має некулясту форму, вона трохи сплюснута біля полюсів       | Чим ближче до екватора, тим менша сила тяжіння завдяки віддаленості від масивного ядра Землі |
| <p>Обертання Землі</p>  | Земля обертається навколо своєї осі, що зумовлює зміну дня та ночі | Завдяки обертанню Землі ближче до екватора тіла важать менше                                 |

| Що не враховується                                                                                                | Пояснення                                                      | Вплив                                                                                                                       |
|-------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| <p>Густина земної речовини</p>   | <p>Густина земної кори на різних ділянках є різною</p>         | <p>Тіла важать більше в районах, де залягають породи з більшою густиною (залізні руди тощо)</p>                             |
| <p>Вплив атмосфери Землі</p>     | <p>На всі тіла з боку земної атмосфери діє архімедова сила</p> | <p>Чим більший об'єм фізичного тіла, тим більша архімедова сила, що зменшує вагу тіла</p>                                   |
| <p>Вплив космічних об'єктів</p>  | <p>Сила всесвітнього тяжіння діє між усіма тілами Всесвіту</p> | <p>Притягання до Місяця (пригадайте морські припливи) і планет Сонячної системи впливають на вагу тіл на поверхні Землі</p> |

Оскільки на звичайне зважування тіл вплив перелічених у таблиці чинників є незначним, то ними у більшості випадків прийнято нехтувати, прийнявши значення прискорення вільного падіння за  $9,81 \text{ Н/кг}$  (рис. 24.2).

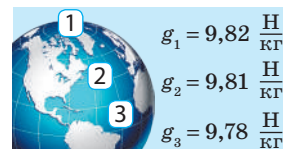


Рис. 24.2

### Питання для самоперевірки



1. Що таке модель; моделювання?
2. На які типи поділяють моделі за способом реалізації?
3. Наведіть приклади інформаційних моделей одного і того самого об'єкта чи явища.
4. Навіщо потрібне моделювання? Наведіть приклади.
5. Які математичні моделі ви використовували на уроках фізики?
6. Які саме спрощення порівняно з реальними об'єктами використовуються в одній із запропонованих вами моделей?



## Вправа 24



- Побудувати в електронній таблиці комп'ютерну модель ваги тіл на різних широтах земної поверхні (рис. 24.3).

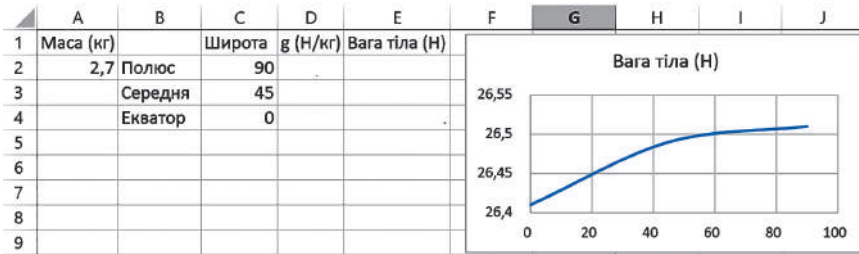


Рис. 24.3

- 1) Відкрийте Excel. Створіть Нову книгу.
- 2) Заповніть таблицю за зразком (рис. 24.3).
- 3) За даними з підручника або знайденими в інтернеті заповніть діапазон D2:D4 значеннями прискорення вільного падіння на трьох широтах.
- 4) У діапазоні E2:E4 формулою підрахуйте значення ваги тіла із заданою масою. Округліть значення ваги тіла до сотих.
- 5) Побудуйте графік залежності ваги тіла від географічної широти. Для цього виділіть значення широт і ваги разом із назвами (діапазон C1:C4;E1;E4), на вкладці Вставлення розгорніть кнопкою групу інструментів Діаграми, виберіть: Усі діаграми, вид діаграми: Точкова, Точкова діаграма з гладкими лініями, натисніть ОК.
- 6) Над діаграмою, що з'явилася, змініть назву за зразком. Зробіть висновок, чи можна за графіком правильно визначити вагу тіла з точністю до сотих ньютонів на різних широтах. Чому? Запишіть висновок у кілька об'єднаних клітинок під таблицею. Збережіть документ із назвою Вправа 24. Надішліть поштою вкладений файл Вправа 24.xlsx на перевірку вчителю. Завершіть роботу.

## Комп'ютерне тестування

Виконайте тестове завдання 24 із автоматичною перевіркою результату.



## § 25. Реалізація математичних моделей засобами електронних таблиць

У сучасному світі для побудови і дослідження математичних моделей зазвичай використовують комп'ютери.

### Комп'ютерне моделювання

Комп'ютерні моделі прості й зручні в дослідженні.



**Комп'ютерна модель** — це інформаційна модель, реалізована за допомогою програмного середовища.

### Етапи побудови комп'ютерної моделі

Ви вже використовували комп'ютерні моделі фізичних явищ і процесів, створені у середовищі програмування Python, і знаєте основні етапи комп'ютерного моделювання.

Розглянемо етапи створення математичної моделі ваги тіла на прикладі планет Сонячної системи (рис 25.1) засобами табличного процесора Excel.



Рис. 25.1

#### ► I етап. Постановка задачі та її аналіз

Постановка задачі вимагає уважного аналізу її формулювання з метою чіткого виділення початкових даних і необхідних результатів:

Що дано, що треба знайти, які існують обмеження на дані і т. п.

**1** Дослідити вагу тіл на планетах Сонячної системи.

*Що моделюється:* вага тіла.

*Мета моделювання:* дослідити вагу тіл на різних планетах.

*Що дано:* прискорення вільного падіння на всіх планетах.

*Що треба знати:* масу тіла  $m$ , вага якого визначається.

*Обмеження на дані:*  $m > 0$ .

#### ► II етап. Побудова інформаційної моделі

На цьому етапі встановлюються суттєві та несуттєві для вибраної моделі параметри й математичні співвідношення між величинами.

**2** *Суттєві параметри:* прискорення вільного падіння на планетах; маса фізичного тіла, вагу якого моделюємо.

*Несуттєві параметри й чинники (чим нехтуємо):* відсутність на деяких планетах твердої поверхні; рух планет навколо Сонця; обертальний рух планети; наявність атмосфери; неоднорідність планетної речовини; вплив інших небесних тіл.

*Математичні співвідношення:*  $P = mg$ , де  $P$  — вага тіла;  $m$  — маса тіла;  $g$  — прискорення вільного падіння.

### ► III етап. Розробка методу й алгоритму реалізації комп'ютерної моделі

Для розрахункової задачі необхідно вибрати метод і продумати алгоритм її розв'язування, що визначає послідовність арифметичних і логічних операцій.

**Метод** — це певний спосіб розв'язування задачі в межах побудованої моделі. **Алгоритм** — це послідовність вказівок виконання дій, спрямована на розв'язання задачі.

Алгоритм розв'язування задачі залежить від вибраного методу.

**3** *Метод:* використання можливостей табличного процесора Excel.

*Алгоритм:* виконати такі дії:

- 1) знайти в інтернеті значення вихідних величин;
- 2) задати значення маси тіла;
- 3) засобами Excel побудувати електронну таблицю з розрахованими значеннями ваги тіла на різних планетах Сонячної системи.

### ► IV етап. Розробка комп'ютерної моделі

Будувати і досліджувати моделі можна за допомогою різних програм. У середовищі табличного процесора зручно робити розрахунки, модель можна подати у вигляді таблиці, графіка тощо.

### ► V етап. Проведення комп'ютерного експерименту

Після створення моделі потрібно здійснити її перевірку за допомогою тестів. Метою комп'ютерного експерименту є підтвердження правильності розробленої моделі або виявлення помилок з метою їх подальшого усунення і реалізації більш якісної моделі.

## Питання для самоперевірки



1. Що таке комп'ютерна модель?
2. Назвіть кілька переваг комп'ютерного моделювання.
3. Перелічіть основні етапи комп'ютерного моделювання.

4. Назвіть суттєві параметри під час реалізації моделі ваги тіла.
5. Назвіть несуттєві параметри під час реалізації моделі ваги тіла.
6. Яка мета проведення комп'ютерного експерименту?

### Вправа 25



- Побудувати в ЕТ комп'ютерну модель ваги тіла на різних планетах Сонячної системи.
- 1) Відкрийте Excel. Створіть Нову книгу.
  - 2) Заповніть ЕТ за зразком, наведеним на рис. 25.2.

|   | A              | B             | C        | D             |
|---|----------------|---------------|----------|---------------|
| 1 | Маса тіла (кг) | Назва планети | g (Н/кг) | Вага тіла (Н) |
| 2 | 2,7            | Меркурій      |          |               |
| 3 |                | Венера        |          |               |
| 4 |                | Земля         |          |               |
| 5 |                | Марс          |          |               |
| 6 |                | Юпітер        |          |               |
| 7 |                | Сатурн        |          |               |
| 8 |                | Уран          |          |               |
| 9 |                | Нептун        |          |               |

Рис. 25.2



- 3) Знайдіть в інтернеті та впишіть у діапазон C2:C9 значення прискорення вільного падіння для кожної планети.
- 4) Розрахуйте у клітинці D2 вагу тіла на Меркурії.
- 5) Автозаповненням скопіюйте формулу для інших планет. Збережіть документ із назвою Вправа 25.



- 6) Надішліть поштою вкладений файл на перевірку вчителю. Завершіть роботу.

### Комп'ютерне тестування



Виконайте тестове завдання 25 із автоматичною перевіркою результату.





## Практична робота 7

### Побудова математичної моделі

**Завдання:** створити математичну модель біоритмів людини.

**Обладнання:** комп'ютер зі встановленим табличним процесором.

#### Хід роботи

*Під час роботи з комп'ютером дотримуйтеся правил безпеки.*

- ▶ **I етап.** Деякі дослідники вважають, що з дня народження стан людини циклічно змінюється. Потрібно визначити, у якому із станів (підйом, спад, найкращий, найгірший) перебуває людина у кожному із трьох біоритмів: фізичному, емоційному, інтелектуальному.
- ▶ **II етап.** Кожен із трьох біоритмів людини можна подати кривою лінією, так званою синусоїдою (рис. 1).

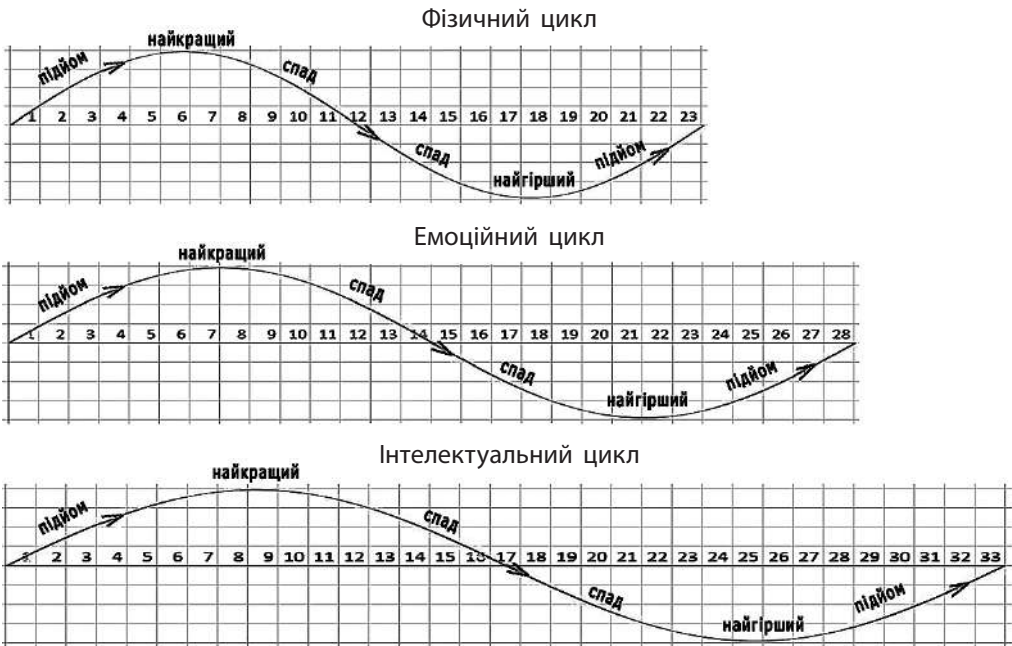


Рис. 1

Щоб з'ясувати, який день кожного з біоритмів людина проживає сьогодні (у якій точці кривих «перебуває»), потрібно знайти остачу

від ділення кількості прожитих людиною днів на періодичність повторення відповідного біоритму.

► **III етап.** Алгоритм реалізації моделі такий.

- 1. Підготуйте ЕТ із потрібними написами за зразком (рис. 2).
- 2. Задайте значення констант у клітинках А2; В2; С2.
- 3. Уведіть у клітинку В4 дату народження у форматі \*дд.мм.рррр.
- 4. Уведіть у клітинку В5 поточну дату у форматі \*дд.мм.рррр.
- 5. Створіть формулу розрахунку остачі від ділення для фізичного стану людини (рис. 2).

| D2 : X ✓ fx =MOD(\$B\$5-\$B\$4;A2) |                          |                           |                                 |               |                |                      |
|------------------------------------|--------------------------|---------------------------|---------------------------------|---------------|----------------|----------------------|
|                                    | A                        | B                         | C                               | D             | E              | F                    |
|                                    | Стала фізичного біоритму | Стала емоційного біоритму | Стала інтелектуального біоритму | Фізичний стан | Емоційний стан | Інтелектуальний стан |
| 1                                  |                          |                           |                                 |               |                |                      |
| 2                                  | 23                       | 28                        | 33                              | 17            | 25             | 31                   |
| 3                                  |                          |                           |                                 |               |                |                      |
| 4                                  | Дата народження          | 20.02.2004                |                                 |               |                |                      |
| 5                                  | Поточна дата             | 21.04.2020                |                                 |               |                |                      |

Рис. 2

- 6. Автозаповненням скопіюйте формулу для емоційного та інтелектуального станів.
  - 7. За отриманими результатами й рисунком визначте кожний стан на поточну дату та спрогнозуйте дату з найкращим інтелектуальним станом. Запишіть висновки в зошит.
- **IV етап.** Комп'ютерна модель складається з ЕТ та рисунку біоритмічних кривих.
- **V етап.** Для проведення комп'ютерного експерименту:
- 8. Знайдіть в інтернеті сервіси визначення біоритмів онлайн.
  - 9. Уведіть на сервісах дату народження, отримайте результат.
  - 10. Звірте результати, отримані в ЕТ, з онлайн-результатами.
  - 11. У разі значних розбіжностей переробіть математичну модель, виправивши помилки. Збережіть файл із назвою Практична 7.
  - 12. Надішліть лист учителю з висновками (п. 7) і вкладеним файлом Практична 7.xlsx. Завершіть роботу.

**Зробіть висновок** про результати роботи.

## КОМП'ЮТЕРНИЙ СЛОВНИК

**Браузер** — спеціальна програма для перегляду вебсторінок.

**Величина** — інформаційний об'єкт (число, символ, рядок тощо), основними характеристиками якого є назва, вид, тип і значення.

**Відмова** — подія, що виникає в разі виклику команди в неприпустимому для цієї команди стані середовища.

**Гіперпосилання** — адреса ресурсу, пов'язана з елементом сторінки, клацання якої приводить до переходу до цього ресурсу.

**Діапазон клітинок** — прямокутна частина таблиці, яка знаходиться на перетині декількох стовпців та рядків.

**Електронна пошта (E-mail)** — служба обміну електронними повідомленнями через комп'ютерну мережу з використанням поштової системи адресації.

**Електронна таблиця** — таблиця, що складається із рядків і стовпців, на перетині яких утворюються клітинки.

**Змінна** — величина, значення якої може змінюватися в ході виконання програми.

**Інкапсуляція** — об'єднання в межах класу певних даних і методів для роботи з ними.

**Інтернет користувача** — сукупність засобів операційної системи для взаємодії із застосуванням комплексу відповідних засобів.

**Інтернет речей** — сукупність під'єднаних до інтернету фізичних пристроїв — «речей», оснащених засобами приймання й передавання даних.

**Інформаційна модель** — модель, що містить інформацію про властивості і стан об'єктів, процесів, явищ.

**Клієнт (робоча станція)** — комп'ютер, який використовує ресурси мережі.

**Коментар** — текст, призначений для читання людиною; підказка, яку програміст записує в програмі.

**Комп'ютерна модель** — модель, реалізована за допомогою комп'ютера.

**Константа** — величина, значення якої не може змінюватися в ході виконання програми.

**Контакт** — сукупність даних про адресата в комунікаційних сервісах інтернету.

**Математична модель** — система математичних співвідношень між величинами, необхідними для розв'язування задачі.

**Модель** — спрощене подання об'єкта, процесу або явища.

**Повторення (цикл)** — алгоритмічна структура, в якій та сама послідовність дій може виконатися кілька разів.

**Поліморфізм** — різна поведінка методу в різних класах.

**Поштовий сервер** — виділений комп'ютер, що забезпечує роботу електронної пошти завдяки спеціальному програмному забезпеченню.

**Робоча книга** — документ Excel, що складається з одного або кількох аркушів.

**Розгалуження** — алгоритмічна структура, в якій виконання певної послідовності дій залежить від істинності умови (логічного виразу).

**Сервер** — комп'ютер, який надає ресурси.

**Синтаксис мови** — сукупність правил побудови команд мови програмування.

**Слідування** — така організація дій в алгоритмі, за якої дії виконуються послідовно одна за одною без пропусків або повторень.

**Спам** — небажана розсилка листів зазвичай рекламного характеру.

**Список** — упорядкований набір об'єктів довільних типів (чисел, рядків, списків тощо), який можна змінювати.

**Список розсилки** — сервіс електронної пошти, який створює можливість об'єднати певну кількість людей у єдину групу для спрощення процесу розсилання листів.

**Табличний процесор** — прикладна програма для опрацювання даних, поданих в електронних таблицях.

**Тіло циклу** — серія команд, які повторюються під час виконання циклу.

**Умова** — запитання, сформульоване таким чином, що допускає лише одну з двох відповідей: «так» або «ні».

**Успадкування** — можливість створення нового класу (нащадка) на базі наявного (базового).

**Фішинг** — шахрайські методи отримання особистих даних (банківських реквізитів батьків тощо) або коштів через інтернет.

**Хмарний диск** — сховище даних, розташоване на віддаленому сервері.

**Хмарні сервіси** — сервіси, пов'язані з наданням користувачам постійного доступу до віддалених інтернет-ресурсів (серверів, додатків, сховищ тощо).



## Алфавітний покажчик

### А

Автозавершення 131  
Автозаповнення 129  
Адресна клітинка 133  
Адресна книга 88  
Алгоритмічна структура 18  
Атрибут класу 65

### Б

Браузер 80

### В

Величина 11

### Г

Гіперпосилання 102

### Д

Діапазон клітинок 116  
Дублювання 30

### Е

Електронна адреса 80  
Електронна пошта 77  
Електронна таблиця 113

### З

Змінна 11

### І

Інкапсуляція 68  
Інтернет речей 107  
Інтерфейс користувача 113  
Інформаційна модель 146  
Ітерація 17

### К

Код 21  
Комп'ютерна модель 150  
Конкатенація 24  
Константа 11  
Контакт 88

### М

Математична модель 8  
Метод 43, 59  
Модель 146

### П

Повторення (цикл) 17  
Поліморфізм 69  
Посилання 135  
Поштовий сервер 79  
Профіль користувача 95

### Р

Рівень доступу 105  
Розгалуження 16

### С

Слідування 13  
Спам 93  
Список 34  
Список розсилки 89

### Т

Табличний процесор 113  
Тип даних 21  
Тіло циклу 17

### У

Успадкування 68

### Ф

Фішинг 93  
Формула 133  
Функція 134  
Функція користувача 29

### Х

Хмарний диск 104  
Хмарний сервіс 103

### Ц

Цикл 18

# Зміст

|                                                                                         |            |
|-----------------------------------------------------------------------------------------|------------|
| Передмова . . . . .                                                                     | 3          |
| <b>Розділ 1. Алгоритми та програми</b>                                                  |            |
| § 1. Етапи розв’язування задачі за допомогою комп’ютера . . . . .                       | 7          |
| § 2. Величини. Алгоритми роботи з величинами . . . . .                                  | 11         |
| § 3. Створення алгоритмів і програм із розгалуженнями та повтореннями . . . . .         | 16         |
| § 4. Рядки як послідовності символів. . . . .                                           | 21         |
| § 5. Опрацювання рядкових величин. . . . .                                              | 27         |
| <i>Практична робота 1. Створення програм для опрацювання рядкових величин . . . . .</i> | <i>32</i>  |
| § 6. Списки — змінювані послідовності. . . . .                                          | 34         |
| § 7. Опрацювання елементів списку . . . . .                                             | 39         |
| § 8. Функції та методи опрацювання списків. . . . .                                     | 43         |
| § 9. Створення інтерфейсу користувача . . . . .                                         | 47         |
| <i>Практична робота 2. Створення програм для опрацювання списків . . . . .</i>          | <i>53</i>  |
| § 10. Графічні методи модуля tkinter. . . . .                                           | 54         |
| § 11. Створення анімації . . . . .                                                      | 59         |
| § 12. Об’єктно-орієнтоване програмування. . . . .                                       | 65         |
| <i>Практична робота 3. Моделювання руху об’єктів . . . . .</i>                          | <i>72</i>  |
| <b>Розділ 2. Служби інтернету</b>                                                       |            |
| § 13. Поштові служби інтернету . . . . .                                                | 77         |
| § 14. Робота з електронною поштою . . . . .                                             | 82         |
| <i>Практична робота 4. Створення електронної поштової скриньки. . . . .</i>             | <i>87</i>  |
| § 15. Особливості електронного листування. . . . .                                      | 88         |
| § 16. Налаштування поштової скриньки . . . . .                                          | 95         |
| § 17. Використання інтернет-ресурсів. . . . .                                           | 102        |
| <i>Практична робота 5. Використання хмарних технологій . . . . .</i>                    | <i>109</i> |

### Розділ 3. Опрацювання табличних даних

|                                                                                      |            |
|--------------------------------------------------------------------------------------|------------|
| § 18. Середовище табличного процесора . . . . .                                      | 113        |
| § 19. Робота з табличними даними . . . . .                                           | 119        |
| § 20. Типи даних в електронних таблицях . . . . .                                    | 124        |
| § 21. Автоматичні дії над вмістом клітинок . . . . .                                 | 129        |
| § 22. Формули в електронних таблицях . . . . .                                       | 133        |
| § 23. Форматування електронних таблиць . . . . .                                     | 139        |
| <i>Практична робота 6. Створення й опрацювання<br/>електронної таблиці . . . . .</i> | <i>144</i> |
| § 24. Моделі і моделювання . . . . .                                                 | 146        |
| § 25. Реалізація математичних моделей засобами електронних<br>таблиць . . . . .      | 150        |
| <i>Практична робота 7. Побудова математичної моделі . . . . .</i>                    | <i>153</i> |
| Комп'ютерний словник . . . . .                                                       | 155        |
| Алфавітний покажчик . . . . .                                                        | 157        |

## Відомості про користування підручником

| № з/п | Прізвище та ім'я учня / учениці | Навчальний рік | Стан підручника |              |
|-------|---------------------------------|----------------|-----------------|--------------|
|       |                                 |                | на початку року | у кінці року |
| 1     |                                 |                |                 |              |
| 2     |                                 |                |                 |              |
| 3     |                                 |                |                 |              |
| 4     |                                 |                |                 |              |
| 5     |                                 |                |                 |              |

Навчальне видання

*БОНДАРЕНКО Олена Олександрівна  
ЛАСТОВЕЦЬКИЙ Василь Васильович  
ПИЛИПЧУК Олександр Павлович  
ШЕСТОПАЛОВ Євген Анатолійович*

### **«ІНФОРМАТИКА»**

**підручник для 7 класу закладів загальної середньої освіти**

**Рекомендовано Міністерством освіти і науки України**

Видано за рахунок державних коштів. Продаж заборонено

Редактор *Л. А. Каюда*. Технічний редактор *А. В. Пліско*.  
Художнє оформлення *В. І. Труфена*.  
Коректор *Н. В. Красна*.

Окремі зображення, що використані в оформленні підручника,  
розміщені в мережі Інтернет для вільного використання

Підписано до друку 30.06.2020. Формат 70×90/16.  
Папір офсетний. Гарнітура Шкільна. Друк офсетний.  
Ум. друк. арк. 11,70. Обл.-вид. арк. 15,2. Тираж 49830 прим. Зам. 7906-2020.

ТОВ Видавництво «Ранок»,  
вул. Кібальчича, 27, к. 135, Харків, 61071.  
Свідоцтво суб'єкта видавничої справи ДК № 5215 від 22.09.2016.  
Адреса редакції: вул. Космічна, 21а, Харків, 61145.  
E-mail: office@ranok.com.ua. Тел. (057) 719-48-65, тел./факс (057) 719-58-67.

Підручник надруковано на папері українського виробництва

Надруковано у друкарні ТОВ «ТРИАДА-ПАК»,  
пров. Сімферопольський, 6, Харків, 61052.  
Свідоцтво суб'єкта видавничої справи ДК № 5340 від 15.05.2017.  
Тел. +38 (057) 712-20-00. E-mail: sale@triada.kharkov.ua