**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**Кафедра програмних систем і технологій**

**Т.В. Ковалюк**

**Методичні вказівки до лабораторних занять з дисципліни**

**«Основи програмування»**

для студентів спеціальності 121 «Інженерія програмного забезпечення»

Київ 2019

# Лабораторна робота 1 Функції ведення, виведення та математичні операції в Python

**Мета роботи:** Ознайомитись з середовищем розробки Python. Вивчити основні типи даних, команди введення і виведення даних.

## Рейтинг лабораторної роботи №1

|  |  |  |  |
| --- | --- | --- | --- |
| № п.п | Вид діяльності студента | Рейтинговий бал | Deadline |
| 1 | Написання коду | 0,5 | Вересень |
| 2 | Захист роботи | 0,5 |
| 3 | Звіт з роботи | 0,5 |
| 4 | Якість роботи | 1 |
| Разом за роботу | | 2,5 |  |

## Теоретичний матеріал

Python- це мова об'єктно-орієнтована, мова, що інтерпретується, мова надвисокого рівня. Програмування на Python дозволяє швидко і якісно отримувати необхідні програмні модулі.

В комплекті разом з інтерпретатором Python йде IDLE (інтегроване середовище розробки). За своєю суттю воно подібне до інтерпретатора, запущеного в інтерактивному режимі з розширеним набором можливостей (підсвічування синтаксису, перегляд об'єктів, налагодження і т.п.).

Для запуску IDLE в Windows необхідно перейти в папку Python в меню "Пуск" і знайти там ярлик з ім'ям "IDLE (Python 3.X XX-bit)".

Для запуску редактора програми (коду) слід виконати команду File-> New File або поєднання клавіш Ctrl + N.

Будь-яка Python-програма складається з послідовності допустимих символів, записаних в певному порядку і за певними правилами.

Програма включає в себе:

* коментарі;
* команди;
* знаки пунктуації;
* ідентифікатори;
* ключові слова.

Коментарі в Python позначаються символом #, що передує тексту коментаря, до кінця рядка (тобто в Python всі коментарі є однорядковими), при цьому не допускається використання перед символом # лапок:

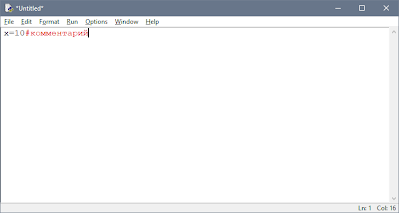
[](https://sites.google.com/site/moiboarkin/laboratornye-raboty/5-kurs/laboratornye-raboty-po-python/l-r-1/1.png?attredirects=0)

Рис.1. Вікно файлу коду із записом коментаря

## Знаки пунктуації

В алфавіт Python входить достатня кількість знаків пунктуації, які використовуються для різних цілей. Наприклад, знаки "+" або "\*" можуть використовуватися для додавання і множення, а знак коми "," - для поділу параметрів функцій.

## Ідентифікатори

Ідентифікатори в Python це імена використовуються для позначення змінної, функції, класу, модуля або іншого об'єкта.

## Ключові слова

Деякі слова мають в Python спеціальне призначення і являють собою керуючі конструкції мови.

### Ключові слова в Python

False – хибність.

True - істина.

None - "порожній" об'єкт.

and - логічне І.

with / as - менеджер контексту.

assert умова - збуджує виключення, якщо умова помилкова.

break - вихід з циклу.

class - призначений для користувача тип, що складається з методів і атрибутів.

continue - перехід на наступну ітерацію циклу.

def - визначення функції.

del - видалення об'єкта.

elif - в іншому випадку, якщо.

else - см. for / else або if / else.

except - перехопити виняток.

finally – у купі з інструкцією try, виконує інструкції незалежно від того, чи було виключення чи ні.

for - цикл for.

from - імпорт декількох функцій з модуля.

global - дозволяє зробити значення змінної, присвоєне їй всередині функції, доступним і за межами цієї функції.

if - якщо.

import - імпорт модуля.

in - перевірка на входження.

is - чи посилаються 2 об'єкти на одне і те ж місце в пам'яті.

lambda - визначення анонімної функції.

nonlocal - дозволяє зробити значення змінної, присвоєне їй всередині функції, доступним в осяжний інструкції.

not - логічне НЕ.

or - логічне АБО.

pass – конструкція, що нічого не робить.

raise - викинути виключення.

return - повернути результат.

try - виконати інструкції перехоплення виключення.

while - цикл while.

yield - визначення функції-генератора.

## Типи даних

1. None (невизначене значення змінної)

2. Логічні змінні (Boolean Type)

3. Числа (Numeric Type)

* int - ціле число
* float - число з плаваючою точкою
* complex - комплексне число

4. Списки (Sequence Type)

* list - список
* tuple - кортеж
* range - діапазон

5. Рядки (Text Sequence Type)

* str

## Введення і виведення даних

Введення даних здійснюється за допомогою команди input (список введення):

a = input ()  
print (a)

У дужках функції можна вказати повідомлення - коментар до даних, що вводяться:

a = input ( "Введіть кількість елементів:")

Команда input () за замовчуванням сприймає вхідні дані як рядок символів. Тому, щоб ввести цілочисельне значення, слід вказати тип даних int ():

a = int (input ())

Для введення дійсних чисел застосовується команда

a = float (input ())

Виведення даних здійснюється за допомогою команди print (список виведення):

a = 1  
b = 2  
print (a)  
print (a + b)  
print ( 'сума =', a + b)

Існує можливість запису команд в один рядок, розділяючи їх через крапку з комою. Однак не слід часто використовувати такий спосіб, це знижує читабельність:

a = 1; b = 2; print (a)  
print (a + b)  
print ( 'сума =', a + b)

У команді print може здаватися сепаратор - роздільник між елементами виведення:

x = 2  
y = 5  
print (x, "+", y, "=", x + y, sep = "")

Результат відобразиться з пробілами між елементами: 2 + 5 = 7

## Форматоване виведення

Для форматованого виведення використовується метод format() функції print():

Рядковий метод format() повертає отформатированную версію рядка, замінюючи ідентифікатори у фігурних дужках {}. Ідентифікатори можуть бути позиційними, числовими індексами, ключами словників, іменами змінних.

Синтаксис команди format:

поле заміни: = "{" [ім'я поля] [ "!" перетворення] [ ":" специфікація] "}"

ім'я поля: = arg\_name ( "." ім'я атрибута | "[" індекс "]") \*

перетворення: = "r" (внутрішнє уявлення) | "S" (людське уявлення)

специфікація: = см. нижче

Аргументів у format () може бути більше, ніж ідентифікаторів в рядку. В такому випадку зайві ідентифікатори ігноруються.

Ідентифікатори можуть бути або індексами аргументів, або ключами:

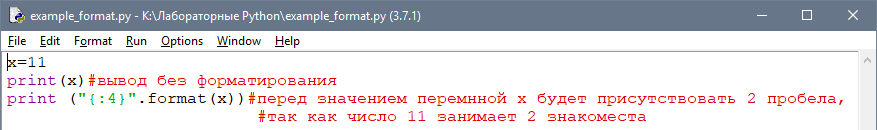


Рис. 2. Код застосування формату виведення

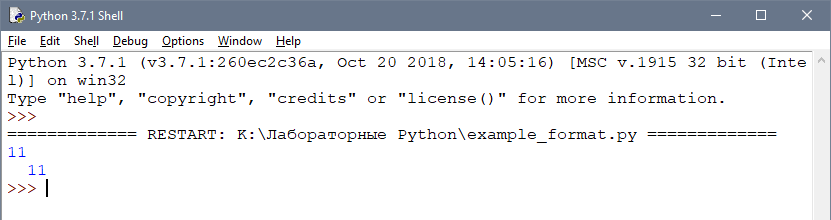
[](https://sites.google.com/site/moiboarkin/laboratornye-raboty/5-kurs/laboratornye-raboty-po-python/l-r-1/ex_format2.png?attredirects=0)

Рис.3. Результат форматування виведення

В результаті виведеться число 11, а перед ним два пробіли, так як вказано чотири знакомісця для виведення. Або з кількома аргументами:

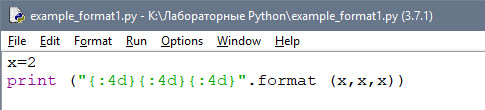
[](https://sites.google.com/site/moiboarkin/laboratornye-raboty/5-kurs/laboratornye-raboty-po-python/l-r-1/math_prost.png?attredirects=0)

Рис.4. Код форматування виведення цілих чисел

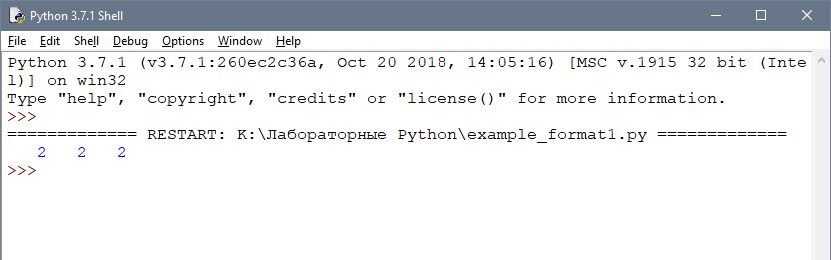
[](https://sites.google.com/site/moiboarkin/laboratornye-raboty/5-kurs/laboratornye-raboty-po-python/l-r-1/ex_format.png?attredirects=0)

Рис.5. Результат форматування

В результаті кожне з значень виводиться з розрахунку 4 знакомісця.

## Специфікація формата

|  |  |
| --- | --- |
| Специфікація | [[fill]align][sign][#][0][width][,][.precision][type] |
| Заповнювач | символ кроме '{' или '}' |
| Вирівнювання | "<" | ">" | "=" | "^" |
| Знак числа | "+" | "-" | " " |
| Ширина | integer |
| Точність | integer |
| Тип | "b" | "c" | "d" | "e" | "E" | "f" | "F" | "g" | "G" | "n" | "o" | "s" | "x" | "X" | "%" |

|  |  |
| --- | --- |
| Тип | Значення |
| 'd', 'i', 'u' | Десяткове число |
| 'o' | Число в вісімковій системі числення. |
| 'x' | Число в шістнадцятковій системі числення (букви в нижньому регістрі). |
| 'X' | Число в шістнадцятковій системі числення (літери у верхньому регістрі). |
| 'e' | Число з плаваючою точкою з експонентою (експонента в нижньому регістрі). |
| 'E' | Число з плаваючою точкою з експонентою (експонента в верхньому регістрі). |
| 'f', 'F' | Число з плаваючою крапкою (звичайний формат). |
| 'g' | Число з плаваючою крапкою. з експонентою (експонента в нижньому регістрі), якщо вона менше, ніж -4 або точності, інакше звичайний формат. |
| 'G' | Число з плаваючою крапкою. з експонентою (експонента в верхньому регістрі), якщо вона менше, ніж -4 або точності, інакше звичайний формат. |
| 'c' | Символ (рядок з одного символу або число, що є кодом символу) |
| 's' | Рядок. |
| '%' | Число множиться на 100, відображається число з плаваючою точкою, а за ним знак%. |

Для форматування дійсних чисел з плаваючою точкою використовується наступна команда:

print('{0:.2f}'.format(вещественное число))

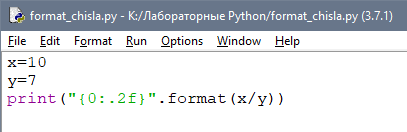
[](https://sites.google.com/site/moiboarkin/laboratornye-raboty/5-kurs/laboratornye-raboty-po-python/l-r-1/format_real_prog.png?attredirects=0)

Рис. 6. Застосування формату для дійсного числа

В результаті буде виведено число з двома знаками після коми.

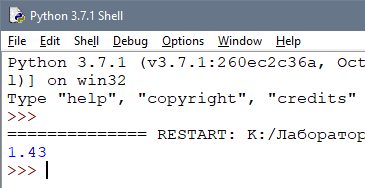
[](https://sites.google.com/site/moiboarkin/laboratornye-raboty/5-kurs/laboratornye-raboty-po-python/l-r-1/format_real_res.png?attredirects=0)

Рис. 7. Результат форматування виведення дійсного числа

## Математичні функції та операції в Python

### Цілі числа (int)

Числа в Python 3 підтримують набір звичайних математичних операцій:

|  |  |
| --- | --- |
| x + y | Додавання |
| x - y | Віднімання |
| x \* y | Множення |
| x / y | Ділення |
| x // y | Отримати цілу частину від ділення |
| x % y | Остача від ділення |
| -x | Зміна знака числа |
| abs(x) | Модуль числа |
| x \*\* y | Піднесення у степінь |
| pow(x, y) | x в степені y |

.

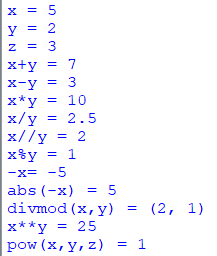


Рис. 8. Приклад застосування простих операцій над цілими числами

### Дійсні числа (float)

Дійсні числа підтримують ті самі операції, що і цілі. Однак (через представлення чисел в комп'ютері) дійсні числа неточні, і це може привести до помилок.

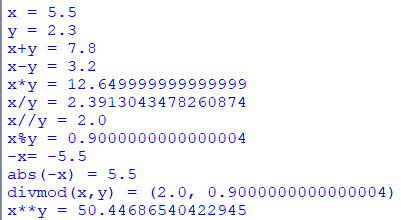


Рис. 9. Приклад застосування простих операцій над дійсними числами

### Бібліотека (модуль) math

У стандартну поставку Python входить бібліотека math, в якій міститься велика кількість математичних функцій. Для роботи з даним модулем його попередньо потрібно імпортувати командою import math.

|  |  |
| --- | --- |
| Назва функції | Опис |
| **Функції в бібліотеці math** | |
| ceil(x) | Повертає округлене x як найближче ціле значення типу int, яке дорівнює або перевищує x (округлення "вгору"). |
| fabs(x) | Повертає абсолютне значення (модуль) числа x. В Python є вбудована функція abs, але вона повертає модуль числа з тим же типом, що число, тут же завжди float abs (fabs). |
| factorial(x) | Повертає факторіал цілого числа x, якщо x не ціле виникає виключення ValueError. |
| floor(x) | На противагу ceil (x) повертає округлене x як найближче ціле значення типу int, менше або рівне x (округлення "вниз"). |
| fsum(iterable) | Повертає float суму від числових елементів об'єкта iterable. |
| isinf(x) | Перевіряє, чи є float об'єкт x плюс або мінус нескінченністю, результат відповідно True або False. |
| isnan(x) | Перевіряє, чи є float об'єкт x об'єктом NaN (not a number). |
| modf(x) | Повертає дробову і цілу частину float числа. Обидва результати зберігають знак вихідного числа x і представлені типом float. |
| trunc(x) | Повертає цілу частину числа x у вигляді int об'єкта. |
| **Степеневі та логарифмічні функції** | |
| exp(x) | Повертає e\*\*x. |
| log(x[, base]) | При передачі функції одного аргументу x, повертає натуральний логарифм x (логарифм за основою e = 2.7182 ...). При передачі двох аргументів, другий береться як основа логарифма. |
| log10(x) | Повертає десятковий логарифм x. |
| pow(x, y) | Повертає x в ступені y. На відміну від операції \*\* перетворює обидва аргументи в тип float. |
| sqrt(x) | Квадратний корінь (square root) з x. |
|  | Основа натуральних логарифмів e = 2.71828. |
| **Тригонометричні функції** | |
| acos(x) | Повертає арккосинус x, в радіанах. |
| asin(x) | Повертає арксинус x, в радіанах. |
| atan(x) | Повертає арктангенс x, в радіанах. |
| cos(x) | Повертає косинус x, де x виражений в радіанах. |
| hypot(x, y) | Повертає sqrt (x \*\* 2 + y \*\* 2). Зручно для обчислення гіпотенузи і довжини вектора. |
| sin(x) | Повертає синус x, де x виражений в радіанах. |
| tan(x) | Повертає тангенс x, де x виражений в радіанах. |
| pi | Константа π = 3.1415 ... |
| **Перетворення кутів** | |
| degrees(x) | Конвертує значення кута x з радіан в градуси. |
| radians(x) | Конвертує значення кута x з градусів в радіани. |

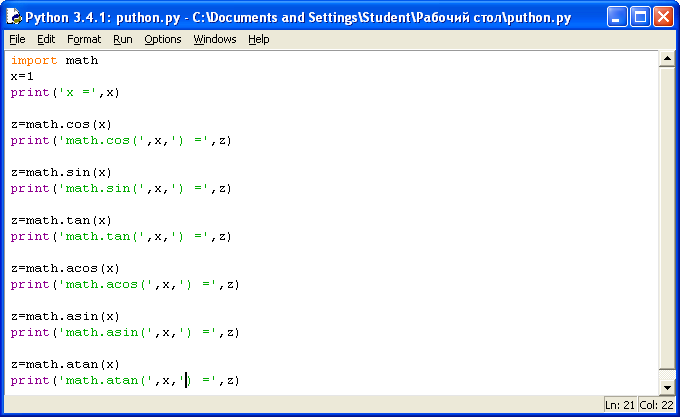


Рис. 10. Приклад програми з використанням тригонометричних функцій модуля math

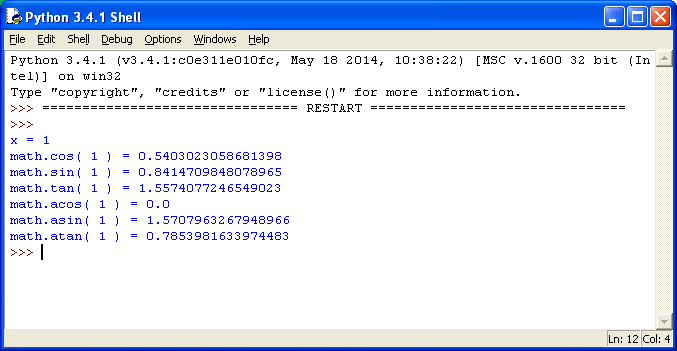


Рис.11. Результат виконання програми із застосуванням тригонометричних функцій модуля math

### Приклад завдання 1

Напишіть програму, яка запитує у користувача дані згідно з варіантом 0:

Вариант 0

- возраст ("Сколько Вам лет?")

( "Ваши фамилия, имя, отчество?")

- место жительства ("Где вы живете?")

После этого выводила бы три строки:

"Ваше имя"

"Ваш возраст"

"Вы живете в"

Рішення

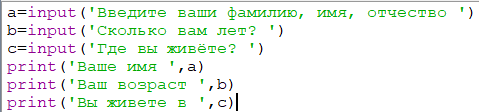
[](https://sites.google.com/site/moiboarkin/laboratornye-raboty/5-kurs/laboratornye-raboty-po-python/l-r-1/string_prog_var0.png?attredirects=0)

Рис. 12. Приклад фрагменту коду завдання 1

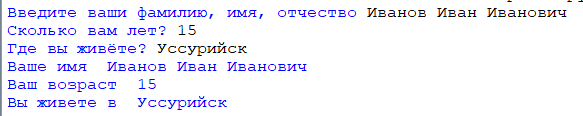
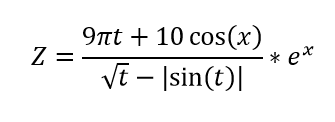
[](https://sites.google.com/site/moiboarkin/laboratornye-raboty/5-kurs/laboratornye-raboty-po-python/l-r-1/string_res_var0.png?attredirects=0)

Рис.13. Результат виконання фрагменту завдання 1

### Приклад завдання 2

У тому самому файлі програми напишіть код, який обчислює заданий арифметичний вираз при заданих змінних.



Введення змінних здійснюється з клавіатури. Вивести результат з 2-ма знаками після коми.

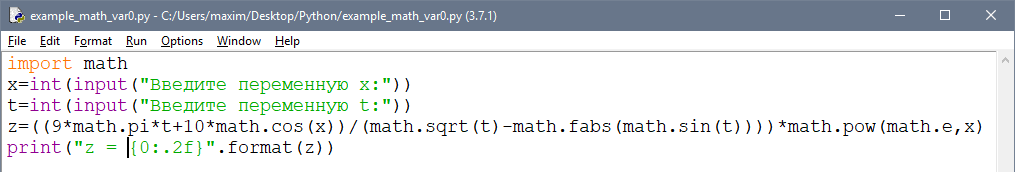


Рис. 14.Код програми прикладу розрахунку виразу

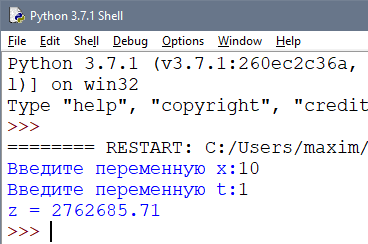


Рис. 15. Результат розрахунку значення виразу

## Завдання для самостійної роботи (за варіантами)

Завдання складається з двох частин. Перша частина ([Завдання 1](#_Завдання_1._Напишіть)) демонструє застосування функцій введення та виведення з форматуванням даних, що виводяться. Друга частина ([Завдання 2](#_Завдання_2:_Обчислити)) вимагає розрахунку виразу із застосуванням математичних функцій з модуля math.

### Завдання 1. Напишіть програму, яка запитувала б у користувача:

#### **Варіант 1**

Ім'я, Прізвище, Вік, де проживає

- прізвище, ім'я ( "Ваші прізвище, ім'я?")

- вік ( "Скільки Вам років?")

- місце проживання ( "Де ви живете?")

Після цього виводила б три рядки:

"Ваші прізвище, ім'я є:"

"Ваш вік"

"Ви живете в"

#### Варіант 2

Ім'я, Дата народження, Освіта

- ім'я ( "Ваше, ім'я?")

- дата народження ( "Ваша дата народження?")

- освіту ( "Де Ви вчитеся?")

Після цього виводила б три рядки:

"Ваше ім'я"

"Дата народження"

"Ви вчитеся в"

#### Варіант 3

Прізвище, де проживає

- Прізвище ( "Ваше прізвище?")

- місце проживання ( "Де Ви живете?")

Після цього виводила б два рядки:

"Ваше прізвище"

"Ви живете в"

#### Варіант 4

Прізвище, Місце народження, улюблена музика

- Прізвище, ( "Ваше прізвище?")

- місце народження ( "Де Ви народилися?")

- музика ( "Яка музика подобається?")

Після цього виводила б три рядки:

"Ваші ім'я, прізвище"

"Ви народилися в"

"Ваша улюблена музика"

#### Варіант 5

Ім'я, Прізвище, ПІБ мами, ПІБ батька

- ПІБ (наприклад, "Ваші прізвище, ім'я, по батькові?")

- вік ( "Скільки Вам років?")

- місце проживання ( "Де Ви живете?")

Після цього виводила б три рядки:

"Ваші ім'я, прізвище, по батькові"

"Ваш вік"

"Ви живете в"

#### Варіант 6

Ім'я, Улюблений предмет в школі, Номер класу

- ім'я ( "Ваше ім'я?")

- улюблений предмет ( "Який Ваш улюблений предмет в школі?")

- номер класу ( "В якому класі Ви навчаєтесь?")

Після цього виводила б три рядки:

"Ваше ім'я"

"Ваш улюблений предмет в школі"

"Ви вчитеся в класі номер"

#### Варіант 7

Ім'я, Прізвище, По батькові, місце народження

- ПІБ (наприклад, "Ваші прізвище, ім'я, по батькові?")

- місце народження ( "Де ви народилися?")

Після цього виводила б три рядки:

"Ваші ім'я, прізвище, по батькові"

В окремому рядку

"Місце народження:"

#### Варіант 8

Ім'я, Прізвище, По батькові, Хобі

- ПІБ (наприклад, "Ваші прізвище, ім'я, по батькові?")

- хобі ( "Чим Ви захоплюєтеся?")

Після цього виводила б два рядки:

"Ваші ім'я, прізвище, по батькові"

"Ваше хобі"

#### Варіант 9

Ім'я, Прізвище, улюблений спорт

- Прізвище, ім'я ( "Ваші прізвище, ім'я?")

- освіту ( "В якій школі Ви вчитеся?")

- ПІБ Вашого керівника по інформатики ( "ПІБ Вашого керівника по інформатики?")

Після цього виводила б три рядки:

"Ваші ім'я, прізвище"

"Ви вчитеся в школі номер:"

"ПІБ Вашого керівника з інформатики"

#### Варіант 10

Ім'я, Прізвище, Улюблений предмет в школі (в інституті), ПІБ класного керівника (куратора)

- Прізвище, ім'я ( "Ваші прізвище, ім'я?")

- улюблений предмет в школі ( "Який Ваш улюблений предмет в школі?")

- ПІБ класного керівника ( "ПІБ Вашого класного керівника?")

Після цього виводила б три рядки:

"Ваші ім'я, прізвище"

"Ваш улюблений предмет в школі"

"ПІБ Вашого класного керівника"

#### Варіант 11

 Ім'я, Прізвище, Вік, Дата народження

- Прізвище, ім'я ( "Ваші прізвище, ім'я?")

- вік ( "Скільки Вам років?")

- дата народження ( "Коли Ви народилися?")

Після цього виводила б три рядки:

"Ваші ім'я, прізвище"

"Ваш вік"

"Дата Вашого народження"

#### Варіант 12

Ім'я, Прізвище, де проживає, Родовища

- Прізвище, ім'я ( "Ваші прізвище, ім'я?")

- місце народження ( "Де Ви народилися?")

- місце проживання ( "Де Ви живете?")

Після цього виводила б три рядки:

"Ваші ім'я, прізвище"

"Ви народилися в"

"Ви живете в"

#### Варіант 13

Ім'я, Прізвище, Вік, Номер телефону

- Прізвище, ім'я ( "Ваші прізвище, ім'я?")

- вік ( "Скільки тобі років?")

- номер телефону ( "Номер Вашого телефону?")

Після цього виводила б три рядки:

"Ваші ім'я, прізвище"

"Ваш вік"

"Ваш номер телефону"

#### Варіант 14

Ім'я, Прізвище, Країна, Край, Місто

- Прізвище, ім'я ( "Ваші прізвище, ім'я?")

- країна ( "В якій країні Ви живете?")

- місто ( "У якому місті Ви живете?")

Після цього виводила б три рядки:

"Ваші ім'я, прізвище"

"Ви живете в країні"

"Ви живете в краї"

"Ви живете в місті"

#### Варіант 15

Ім'я, Прізвище, ПІБ Вашого класного керівника

- Прізвище, ім'я ( "Ваші прізвище, ім'я?")

- ПІБ Вашого класного керівника ( "ПІБ Вашого класного керівника?")

Після цього виводила б три рядки:

"Ваші ім'я, прізвище"

"ПІБ Вашого керівника з інформатики"

"ПІБ Вашого класного керівника"

### Завдання 2: Обчислити значення заданого виразу відповідно до варіантів при заданих користувачем з клавіатури значеннях параметрів A, B, C, D.

Студент може ускладнити формулу для розрахунку і отримати додаткові бали за якість роботи

|  |  |
| --- | --- |
| 1 X=(2A - B2)/(D(SinC - 4.5)) | 15. X=((1 - A)CeA)/((1-B)CosD) |
| 2 X=(2.3 + AB2)/(0.4 + CeD) | 16. X=(ACosA(1 + B3))/(C(1 - D2)) |
| 3. X=(-2.25(A + 2BC))/(B - D1/2) | 17. X=A/(1 + B/(C + Ln(D + A))) |
| 4. X=2ASin(B/2) + DC1/2 | 18. X=(CosA + SinB)/(LnC + LnD)1/2 |
| 5. X=(A - |B/C|)/(2.75Sin2D) | 19. X=(ACos2B + 1)1/2LnC/D |
| 6 X=(|A - B|2C2)/(1 - tgD2) | 20. X=Arctg(4A/(B2C) - 1)1/2 |
| 7. X=e((SinA + SinB)/(C + D)) | 21. X=A2 + |SinB/(C - D)1/2| |
| 8. X=(1 - A)/(1 + A)+|(B - 2D)/C2| | 22. X=(|A| + |B|)2/(D - C1/2) |
| 9. X=(A2-B2)(C-D)/(e(A2 - B2))1/2 | 23. X=(A1/2 + B2)eC+D/(|A| - |B|) |
| 10. X=(1-B)/(1+B)+ |(C-2A)/C2| | 24. X=1/2Ln((1+SinA)/(1- SinB)2) |
| 11. X=Sin3ACos2B/(eC + 1 + SinD) | 25. X=Sin3CCos2A/(eA + 1 + SinB) |
| 12. X=|1-B||1-A|/((1+SinC)(1+CosC))1/2 | 26. X=(1/(A2-B2)1/2)+(3C3/(C2-D2)1/2) |
| 13. X=(ASinB+BCosA)/(1-SinC\*|B+D|) | 27. X=(eACos(B/2+D/C))/(DC+SinC) |
| 14. X=A2/|1-B|+B2/|1-D|+C2/LnA | 28. X=(A-4B)2/(1+(1+SinC)(1+CosC)) |

## Список літератури

1. <https://github.com/tkovalyuk/Basics-of-programming>
2. **МакГрат М**. Программирование на Python для начинающих. –Москва: Эксмо. – 192 с.
3. **Лутц М.** Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
4. **Лутц М.** Программирование на Python, том I, 2. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
5. **Васильев А. Н.** Python на примерах. Практический курс ·по программированию. - СПб.:Наука и Техника, 2016. - 432 с.
6. **Доусон М.** Программируем на Python. - СПб.: Питер, 2014. - 416 с.
7. **Рейтц К., Шлюссер Т.** Автостопом по Python. — СПб.: Питер, 2017. — 336 с.

# Лабораторна робота 2 Процеси розгалуження в Python

**Мета роботи:** Ознайомитись з операторами розгалуження обчислювальних процесів в Python. Навчитися працювати з числами, використовуючи дану структуру. Побудувати блок-схему алгоритму, розробити програму, реалізувати її та оцінити результат її роботи.

## Рейтинг лабораторної роботи №2

|  |  |  |  |
| --- | --- | --- | --- |
| № п.п | Вид діяльності студента | Рейтинговий бал | Deadline |
| 1 | Написання коду | 0,5 | Вересень |
| 2 | Захист роботи | 0,5 |
| 3 | Звіт з роботи | 0,5 |
| 4 | Якість роботи | 1 |
| Разом за роботу | | 2,5 |  |

## Теоретичний матеріал

### Умовний оператор розгалуження if, if-else, if-elif-else

Оператор розгалуження **if** дозволяє виконати певний набір інструкцій залежно від деякої умови. Можливі наступні варіанти використання.

#### 1. Конструкція if

if логічний вираз**:**

    команда\_1

    команда\_2

    ...

    команда\_n

Після оператора **if** записується логічне вираз, результатом обчислення якої є «істина» або «хибність». Якщо цей вислів істинний, то виконуються інструкції, які визначаються даним оператором. Вираз є істинним, якщо його результатом є **число не рівне нулю, непорожній об'єкт, або логічне True**.

Після виразу потрібно поставити двокрапку ":".

ВАЖЛИВО: блок коду, який необхідно виконати, в разі вислову, відділяється чотирма пробілами зліва!

#### 2. Конструкція if - else

Бувають випадки, коли при істинній умові потрібно виконати один набір інструкцій, при помилковій - інший. Для цього використовується конструкція **if - else.**

if логічний вираз:

     команда\_1

     команда\_2

     ...

     команда\_n

else:

     команда\_1

     команда\_2

     ...

     команда\_n

#### 3. Конструкція if - elif - else

Для реалізації вибору з декількох альтернатив використовують конструкцію **if - elif - else.**

if логічне вираз\_1:

    команда\_1

    команда\_2

    ...

    команда\_n

elif логічне вираженіе\_2:

    команда\_1

    команда\_2

    ...

    команда\_n

elif логічне вираженіе\_3:

    команда\_1

    команда\_2

    ...

    команда\_n

else:

    команда\_1

    команда\_2

    ...

    команда\_n

*Приклад*

Програма запитує число у користувача і порівнює його з нулем a <0. Якщо воно менше нуля, то виводиться повідомлення про це. Якщо перше логічне вираження не істинно, то програма переходить до другого - a == 0. Якщо воно істинне, то програма виведе повідомлення, що число дорівнює нулю, в іншому випадку, якщо обидва вищевказаних логічних вирази виявилися хибними, то програма виведе повідомлення, що введене число більше нуля.

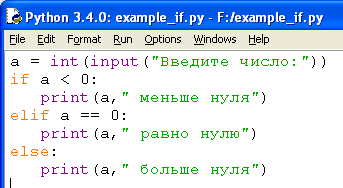


Рис.1. Приклад програми на Python

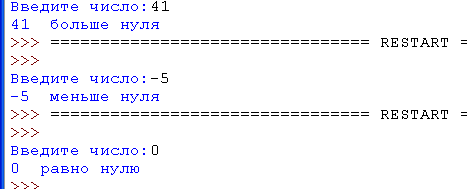


Рис.2 Результат виконання програми з використанням умовного оператора if-elif-else

## Математичні функції та операції в Python

### Цілі числа (int) та дійсні (float)

Числа в Python 3 підтримують набір звичайних математичних операцій:

|  |  |
| --- | --- |
| x + y | Додавання |
| x - y | Віднімання |
| x \* y | Множення |
| x / y | Ділення |
| x // y | Отримати цілу частину від ділення |
| x % y | Остача від ділення |
| -x | Зміна знака числа |
| abs(x) | Модуль числа |
| x \*\* y | Піднесення у степінь |
| pow(x, y) | x в степені y |

### Бібліотека (модуль) math

У стандартну поставку Python входить бібліотека math, в якій міститься велика кількість математичних функцій. Для роботи з даним модулем його попередньо потрібно імпортувати командою import math.

|  |  |
| --- | --- |
| Назва функції | Опис |
| **Функції в бібліотеці math** | |
| ceil(x) | Повертає округлене x як найближче ціле значення типу int, яке дорівнює або перевищує x (округлення "вгору"). |
| fabs(x) | Повертає абсолютне значення (модуль) числа x. В Python є вбудована функція abs, але вона повертає модуль числа з тим же типом, що число, тут же завжди float abs (fabs). |
| factorial(x) | Повертає факторіал цілого числа x, якщо x не ціле виникає виключення ValueError. |
| floor(x) | На противагу ceil (x) повертає округлене x як найближче ціле значення типу int, менше або рівне x (округлення "вниз"). |
| fsum(iterable) | Повертає float суму від числових елементів об'єкта iterable. |
| isinf(x) | Перевіряє, чи є float об'єкт x плюс або мінус нескінченністю, результат відповідно True або False. |
| isnan(x) | Перевіряє, чи є float об'єкт x об'єктом NaN (not a number). |
| modf(x) | Повертає дробову і цілу частину float числа. Обидва результати зберігають знак вихідного числа x і представлені типом float. |
| trunc(x) | Повертає цілу частину числа x у вигляді int об'єкта. |
| **Степеневі та логарифмічні функції** | |
| exp(x) | Повертає e\*\*x. |
| log(x[, base]) | При передачі функції одного аргументу x, повертає натуральний логарифм x (логарифм за основою e = 2.7182 ...). При передачі двох аргументів, другий береться як основа логарифма. |
| log10(x) | Повертає десятковий логарифм x. |
| pow(x, y) | Повертає x в ступені y. На відміну від операції \*\* перетворює обидва аргументи в тип float. |
| sqrt(x) | Квадратний корінь (square root) з x. |
|  | Основа натуральних логарифмів e = 2.71828. |
| **Тригонометричні функції** | |
| acos(x) | Повертає арккосинус x, в радіанах. |
| asin(x) | Повертає арксинус x, в радіанах. |
| atan(x) | Повертає арктангенс x, в радіанах. |
| cos(x) | Повертає косинус x, де x виражений в радіанах. |
| hypot(x, y) | Повертає sqrt (x \*\* 2 + y \*\* 2). Зручно для обчислення гіпотенузи і довжини вектора. |
| sin(x) | Повертає синус x, де x виражений в радіанах. |
| tan(x) | Повертає тангенс x, де x виражений в радіанах. |
| pi | Константа π = 3.1415 ... |
| **Перетворення кутів** | |
| degrees(x) | Конвертує значення кута x з радіан в градуси. |
| radians(x) | Конвертує значення кута x з градусів в радіани. |

### Приклад завдання

Знайти розв’язки квадратного рівняння I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_02\pictures\pr2_1.gif, коефіцієнти якого є дійсними числами, що їх вводить користувач. Залежно від значень коефіцієнтів *a, b, c* та дискримінанта I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_02\pictures\pr2_2.gif можливі такі результати: всі дійсні числа є коренями (a = 0, b = 0, c=0), коренів немає (a = 0, b = 0, cI:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_02\pictures\pr2_3.gif0), є один корінь (a = 0, bI:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_02\pictures\pr2_3.gif0), є два різних дійсних корені (aI:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_02\pictures\pr2_3.gif0, d>0), два дійсних корені збігаються (aI:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_02\pictures\pr2_3.gif0, d=0) або існує два комплексно-спряжених корені (aI:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_02\pictures\pr2_3.gif0, d < 0).

Блок-схема алгоритму:



Рис. 3. Блок-схема алгоритму обчислення евадрптних коренів

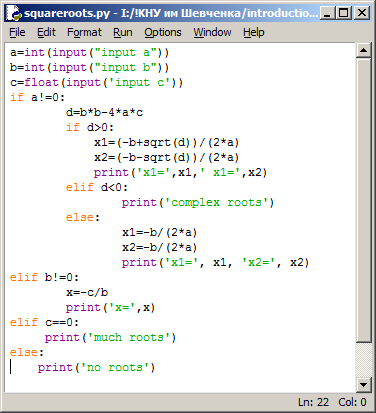


Рис. 4. Код програми розрахунку квадратних коренів

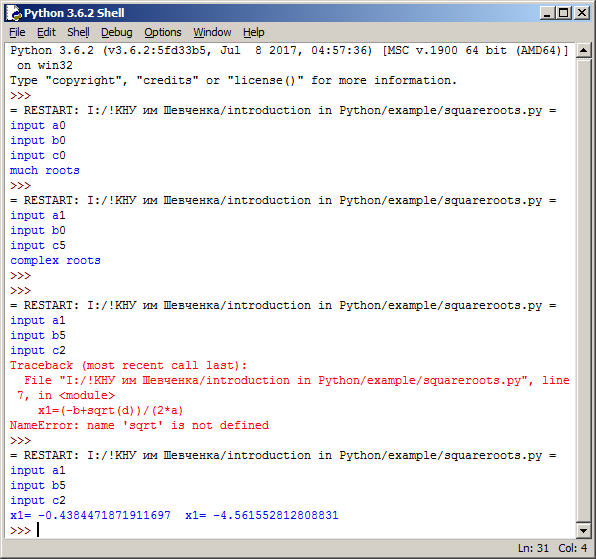


Рис.5. Результати роботи програми

## Завдання для самостійної роботи (за варіантами)

#### **Варіант 1**

Визначити тип трикутника за значеннями уведених його сторін (рівнобедрений, рівнобокій, прямокутний, різнобокій, не існує).

#### **Варіант 2**

Дано три цілих числа. Вибрати з них ті, які належать інтервалу [1,3].

#### Варіант 3

Складіть програму підрахунку площі рівнобедреного трикутника. Якщо площа трикутника парна, розділити її на 2, в іншому випадку вивести повідомлення «Не можу ділити на 2!»

#### Варіант 4

Скласти програму, яка по заданому числу (1-12) виводить назву відповідного йому місяця англійською мовою.

#### Варіант 5

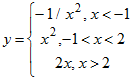
Дано три числа. Знайти кількість додатних чисел серед них;

#### Варіант 6

Якщо дійсні числа x і y мають однаковий знак, знайти їх середнє геометричне, в іншому випадку знайти їх середнє арифметичне.

#### Варіант 7

Обчислити значення функції y, якщо

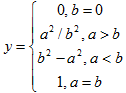


#### Варіант 8

Визначити, чи існує прямокутний трикутник зі сторонами x, y, z, значення яких введено з клавіатури. Якщо так, обчислити його площу.

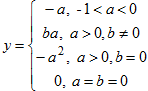
#### Варіант 9

Обчислити значення функції y, якщо



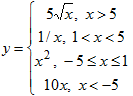
#### Варіант 10

Обчислити значення функції y, якщо



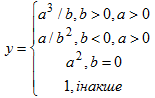
#### Варіант 11

Обчислити значення функції y, якщо



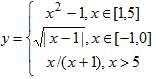
#### Варіант 12

Обчислити значення функції y, якщо



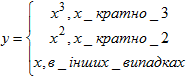
#### Варіант 13

Обчислити значення функції y, якщо



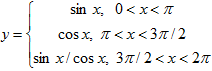
#### Варіант 14

Обчислити значення функції y, якщо



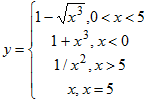
#### Варіант 15

Обчислити значення функції y, якщо



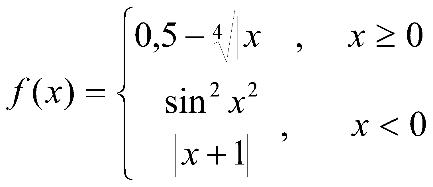
#### Варіант 16

Обчислити значення функції y, якщо



#### Варіант 17

Обчислити значення функції f(x), якщо

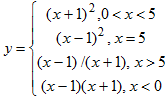


#### Варіант 18

Запросити у користувача два числа. Якщо перше більше другого, то обчислити їх різницю і вивести дані на друк. Якщо друге число більше першого, то обчислити їх суму і вивести на друк. Якщо обидва числа рівні, то вивести це значення на друк.

#### Варіант 19

Обчислити значення функції y, якщо



#### Варіант 20

Визначити умови взаємного розташування двох прямих, що задаються коефіцієнтами рівнянь вигляду *ax* + *by* + *c* = 0 (паралельні, перпендикулярні, перетинаються, збігаються).

## Список літератури

1. <https://github.com/tkovalyuk/Basics-of-programming>
2. **МакГрат М**. Программирование на Python для начинающих. –Москва: Эксмо. – 192 с.
3. **Лутц М.** Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
4. **Лутц М.** Программирование на Python, том I, 2. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
5. **Васильев А. Н.** Python на примерах. Практический курс ·по программированию. - СПб.:Наука и Техника, 2016. - 432 с.
6. **Доусон М.** Программируем на Python. - СПб.: Питер, 2014. - 416 с.
7. **Рейтц К., Шлюссер Т.** Автостопом по Python. — СПб.: Питер, 2017. — 336 с.

# Лабораторна робота 3 Циклічні процеси в Python

**Мета роботи:** Ознайомитись з операторами циклу в Python. Побудувати блок-схему алгоритму, розробити програму з використанням циклів, реалізувати її та оцінити результат її роботи.

## Рейтинг лабораторної роботи №3

|  |  |  |  |
| --- | --- | --- | --- |
| № п.п | Вид діяльності студента | Рейтинговий бал | Deadline |
| 1 | Написання коду | 1 | Вересень |
| 2 | Захист роботи | 1 |
| 3 | Звіт з роботи | 1 |
| 4 | Якість роботи | 1 |
| Разом за роботу | | 4 |  |

## Теоретичний матеріал

В Python існують два типи циклічних виразів:

* 1. цикл while
  2. цикл for

### Цикл while в Python

Інструкція while в Python повторює вказаний блок коду до тих пір, поки вказаний в циклі логічний вираз буде залишатися істинним.

**while логічний вираз:**

**команда 1**

**команда 2**

**...**

**команда n**

Після ключового слова **while** вказується умовний вираз, і поки цей вислів повертає значення **True**, буде виконуватися блок інструкцій, який йде далі. Всі інструкції, які відносяться до циклу **while**, розташовуються на наступних рядках і повинні мати відступ від початку рядка (4 пробіли).

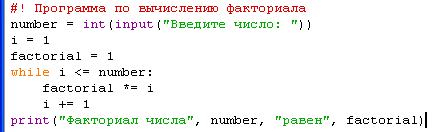


Рис.1. Приклад циклу while для обчислення факторіалу числа

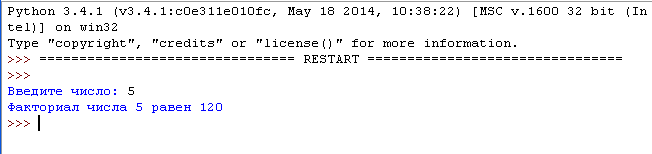


Рис.2. Результат обчислення факторіалу числа

### Цикл for

Оператор **for** виконує вказаний набір інструкцій задану кількість разів, яка визначається кількістю елементів в наборі (наприклад, в рядках або списках).

Синтаксис циклу for:

**for int in range ():**

**команда 1**

**команда 2**

**...**

**команда n**

Змінній **int** присвоюється значення першого елемента функції **range (),** після чого виконуються команди.

Потім змінній **int** присвоюється наступне по порядку значення і так далі до тих пір, поки не будуть перебрані всі елементи функції **range ().**

**Приклад.** Знайти суму n елементів наступного ряду чисел: 1 -0.5 0.25 -0.125 ... n. Кількість елементів (n) вводиться з клавіатури. Вивести на екран кожен член ряду і його суму. Вирішити задачу використовуючи циклічну конструкцію **for**.

**Рішення:**

1. В даному випадку ряд чисел складається з елементів, де кожен наступний менше попереднього в два рази по модулю і має зворотний знак. Значить, щоб отримати наступний елемент, треба попередній розділити на -2.
2. На початку будь-якій змінній треба присвоїти значення першого елемента ряду (в даному випадку це 1).
3. Далі в циклі додавати її значення до змінної, в якій накопичується сума, після чого присвоїти їй значення наступного елемента ряду, розділивши поточне значення на -2.
4. Цикл повинен виконуватися n раз.

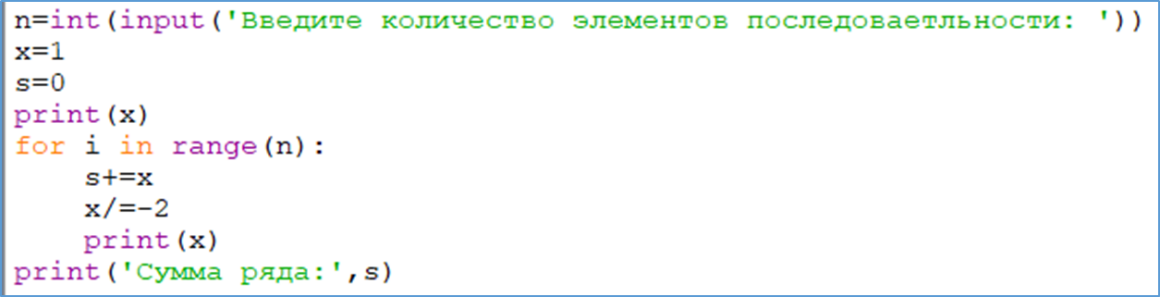


Рис.3. Приклад циклу for для обчислення суми послідовності чисел

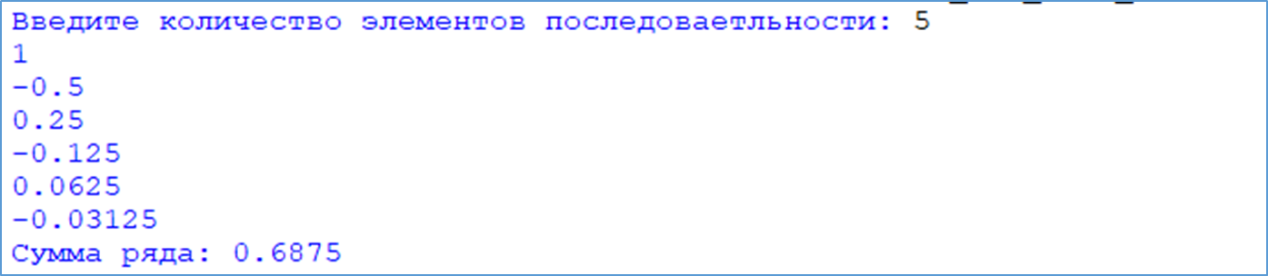


Рис.4. Результат обчислення суми послідовності чисел

### 3. Оператори break і continue

При роботі з циклами використовуються оператори **break і continue.**

Оператор **break** призначений для дострокового переривання роботи циклу **while** або **for.**

Оператор **continue** запускає цикл заново, при цьому код, розташований після цього оператора, не виконується.

Блок **else** дозволяє перевірити чи виконався цикл повністю. Якщо ключове слово **break** не було викликане, то буде виконаний блок **else**.

Приклад. Перевірити, чи задане число є простим.

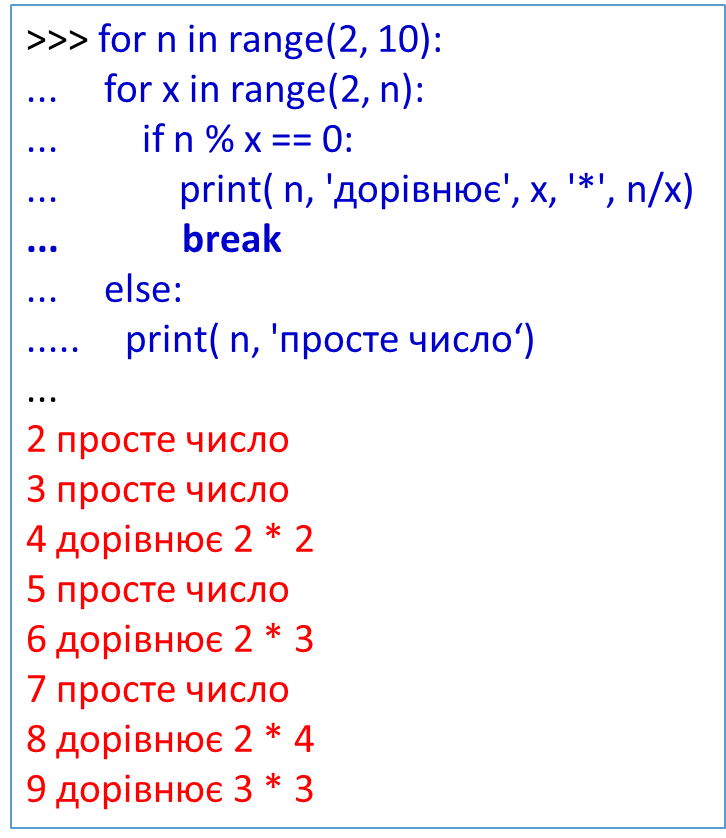


Рис.5. Код програми перевірки числа на простоту та результати його роботи.

## Завдання для самостійної роботи (за варіантами)

Обчислити суму ряду, елементи якого розраховуються з використанням операцій піднесення у степінь, розрахунку факторіалу числа, операцій множення та ділення.

Для розрахунку першої суми з параметром від 1 до 5 використати цикл **for**. Для розрахунку внутрішньої суми з параметром від 0 до нескінченності використати цикл **while**.

Для визначення нескінченності потрібно задати точність розрахунку поточного элементу ряду. Підсумовування припиняється, якщо досягнута задана користувачем точність розрахунку. Перевірити знаменник на рівність нулю для запобігання ділення на нуль.

Результат подати у вигляді таблиці із чотирма колонками, що визначають параметри зовнішньої та внутрішньої сум, значення члена ряду та накопичувану суму (рис. 6)

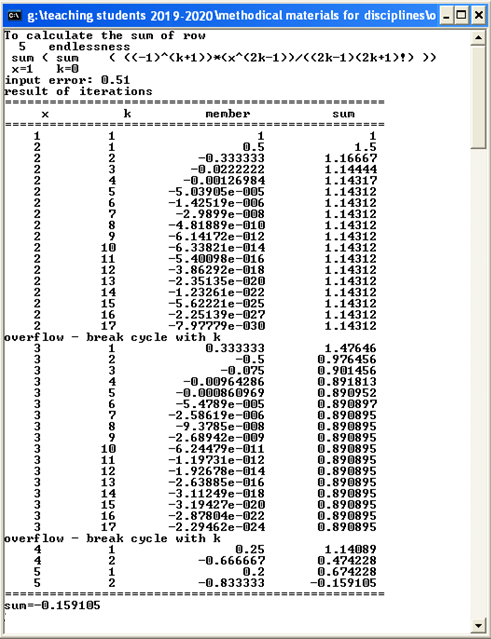
****

Рис. 6. Лістинг результату роботи програми обчислення суми ряду

### Варіанти завдань

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var1.gif | 12. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var12.gif |
| 2. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var2.gif | 13. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var13.gif |
| 3. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var3.gif | 14. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var14.gif |
| 4. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var4.gif | 15. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var15.gif |
| 5. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var5.gif | 16. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var16.gif |
| 6. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var6.gif | 17. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var17.gif |
| 7. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var7.gif | 18. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var18.gif |
| 8. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var8.gif | 19. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var19.gif |
| 9. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var9.gif | 20. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var20.gif |
| 10. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var10.gif | 21. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var21.gif |
| 11. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var11.gif | 22. | I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_03\pictures\var22.gif |

## Список літератури

1. <https://github.com/tkovalyuk/Basics-of-programming>
2. **МакГрат М**. Программирование на Python для начинающих. –Москва: Эксмо. – 192 с.
3. **Лутц М.** Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
4. **Лутц М.** Программирование на Python, том I, 2. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
5. **Васильев А. Н.** Python на примерах. Практический курс ·по программированию. - СПб.:Наука и Техника, 2016. - 432 с.
6. **Доусон М.** Программируем на Python. - СПб.: Питер, 2014. - 416 с.
7. **Рейтц К., Шлюссер Т.** Автостопом по Python. — СПб.: Питер, 2017. — 336 с.

# Лабораторна робота 4 Функції та рекурентні співвідношення в Python

**Мета роботи:** Ознайомитись з технологією використання функцій в Python та застосуванням рекурентних співвідношень. Побудувати блок-схему алгоритму, розробити програму з використанням рекурентних співвідношень для розкладання математичних функцій в ряди Тейлора або Маклорена.

## Рейтинг лабораторної роботи №4

|  |  |  |  |
| --- | --- | --- | --- |
| № п.п | Вид діяльності студента | Рейтинговий бал | Deadline |
| 1 | Написання коду | 1 | Жовтень |
| 2 | Захист роботи | 1 |
| 3 | Звіт з роботи | 1 |
| 4 | Якість роботи | 1 |
| Разом за роботу | | 4 |  |

## Теоретичний матеріал

Функції в програмуванні можна представити як іменований блок коду, звернення до якого в процесі виконання програми може бути багаторазовим.

Різновиди функцій в Python:

1. Вбудовані функції
2. Функції з імпортованих модулів
3. Функції користувача/

### Вбудовані функції Python

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Built-in Functions** <https://pythoner.name/documentation/library/functions> | | | | |
| abs() | [delattr()](https://docs.python.org/3/library/functions.html) | hash() | memoryview() | set() |
| all() | dict() | help() | min() | [setattr()](https://docs.python.org/3/library/functions.html) |
| any() | dir() | hex() | next() | slice() |
| ascii() | divmod() | id() | object() | sorted() |
| bin() | enumerate() | input() | oct() | [staticmethod()](https://docs.python.org/3/library/functions.html) |
| bool() | eval() | int() | open() | [str()](https://docs.python.org/3/library/functions.html) |
| breakpoint() | exec() | isinstance() | ord() | sum() |
| bytearray() | filter() | issubclass() | pow() | super() |
| bytes() | float() | iter() | print() | tuple() |
| callable() | format() | len() | property() | type() |
| chr() | frozenset() | list() | range() | [vars()](https://docs.python.org/3/library/functions.html) |
| classmethod() | getattr() | locals() | repr() | zip() |
| compile() | globals() | map() | reversed() | \_\_import\_\_() |
| complex() | hasattr() | max() | round() |  |

### Функції користувача можна поділити на:

1. Функції без параметрів, що не повертають значення
2. Функції з параметрами, що не повертають значення
3. Функції з параметрами, що повертають значення.

### Оголошення та виклик функції

|  |  |
| --- | --- |
| Формат оголошення | Оператор виклику функції |
| def <ім’я\_функції>() :  блок коду | ім’я функції() |
| def <ім’я\_функції>(параметри) :  блок коду | ім’я функції(аргументи) |
| def <ім’я\_функції>(параметри) :  блок коду з оператором return | Змінна = ім’я функції(аргументи) |

Визначення функції має бути розташоване перед викликом функції.

***Параметрами*** називають змінні, за допомогою яких здійснюється передавання даних у функцію, що викликається, з програмного блока, який здійснює виклик.

Значення параметрів функції, що вказуються під час її виклику, називають ***аргументами* *функції*.**

### Приклад програми визначення чисел Фібоначчі.

Числа Фібоначчі - це ряд чисел, в якому кожне наступне число дорівнює сумі двох попередніх: 1, 1, 2, 3, 5, 8, 13, ...

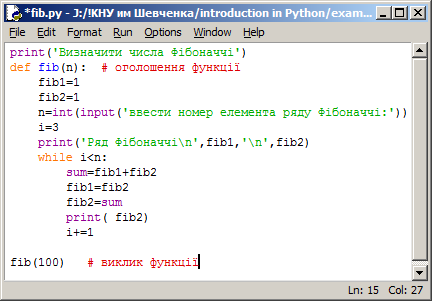


Рис.1. Код програми визначення послідовності Фібоначчі - – варіант1.

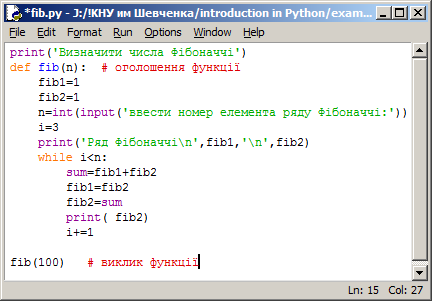


Рис.2. Результат роботи програми визначення послідовності Фібоначчі.

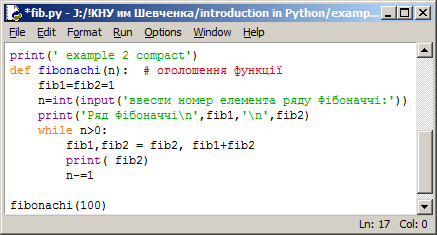


Рис.3. Код програми визначення послідовності Фібоначчі – варіант2.

### Функції, що повертають значення

Функція може повернути значення у точку її виклику. Під час виклику функції, що повертає значення в точку виклику, її ім’я може бути інтерпретовано як ім’я деякої змінної величини. Функцію, що повертає значення, можна викликати у виразах. Для повернення значення з функції в точку виклику використовують оператор повернення **return**.

Оператор повернення значення з функції:

**return (вираз)**

В Python функції здатні повертати кілька значень одночасно:

Наприклад, **return x1, x2**

Всередині функції може міститися довільна кількість операторів **return**. Однак спрацює лише один з них.

Якщо функція не викликає return явно, буде отримано результат **None**. **None**– це спеціальне значення в Python, яке визначає порожній об’єкт, якщо функція нічого не повертає.

### Глобальні змінні

Змінні, які оголошені всередині функції, називають ***локальними***. Ці змінні стають недоступными після виходу з функції.

Змінні, які оголошені поза функцією, але доступні всередині функції, називають **глобальними**. Щоб змінити значення глобальної змінної всередині функції, треба явно вказати що ця змінна є глобальною. Робиться це за допомогою інструкції **global**.

### Параметри за замовчуванням

Значення за замовчуванням використовуються в тому випадку, якщо викликаючи функцію не було вказано відповідний аргумент.

**Механізм параметрів за замовчуванням** діє так: якщо змінну проініціалізовано до виклику функції, то у функцію передається саме це значення, в іншому випадку у функцію передається значення за замовчуванням.

def func(a, b, c=2):

return a+b\*c

print(func(1, 2))

### Передавання у функцію змінної кількості аргументів

Для передавання у функцію змінного числа аргументів в Python можна скористатися символом \* перед списком аргументів змінної довжини.

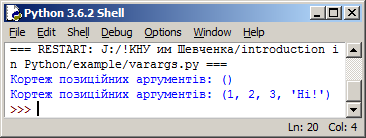
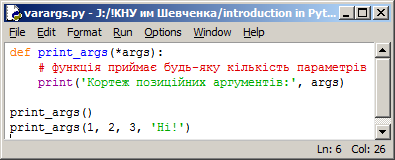


Рис. 4. Приклад функції зі змінною кількістю параметрів та результати роботи програми

### Приклад фрагменту коду лабораторного завдання

**Завдання**. Використовуючи розкладання функції sin(x) в степеневий ряд Маклорена (або Тейлора), написати програму обчислення значення функції sin(x) при заданих аргументах. Для розкладання в степеневий ряд створити функцію мовою Python, що повертає значення і має параметри.

**Теорія**. Розкладання функції sin(x) у степеневий ряд Маклорена має такий вираз:



#### Алгоритм обчислення функції sin(x), розкладанням її в ряд Маклорена, у вербальній формі

1. Ввести значення аргументу функції.
2. Ввести точність розрахунку.
3. Задати початкове значення суми ряду, рівним значенню аргументу.
4. Задати початкове значення елемента ряду, рівним значенню аргументу.
5. Задати початкове значення лічильника циклу рівним 2.
6. Поки модуль поточного елемента ряду більше точності розрахунку повторювати такі дії:

6.1. Обчислити поточний член ряду за формулою:

6.2. Додати поточний елемент до суми ряду.

6.3. Збільшити лічильник циклу на 2. Перейти на п. 6.1.

7. По закінченні циклу повернути значення суми ряду.

#### Блок-схеми програми



Рис. 5. Блок-схема розрахунку значення функції sin(x)

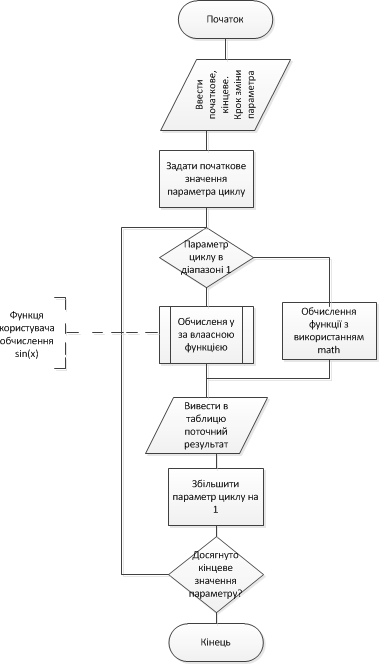


Рис. 6. Блок-схема головної програми

#### Код програми

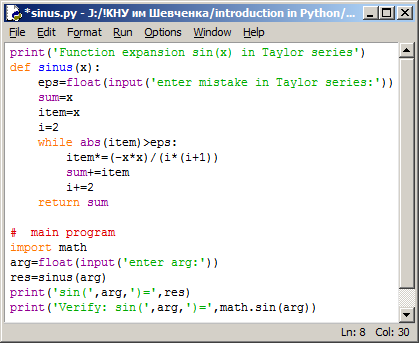
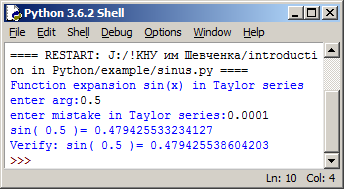
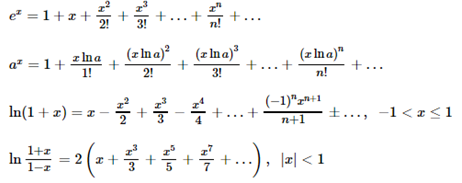
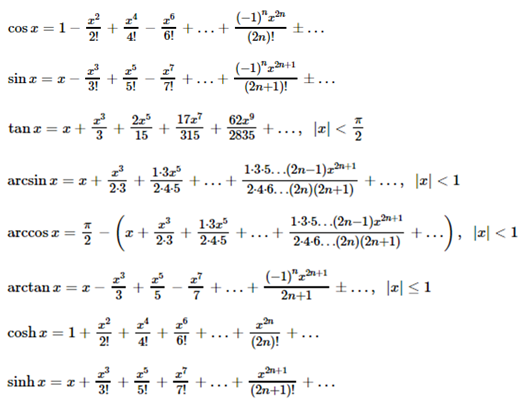
 

Рис. 7. Приклад коду та результати роботи

### Розкладання деяких функцій в ряд Маклорена







## Завдання для самостійної роботи (за варіантами)

Обчислити значення функції, розвинувши її у ряд Маклорена (абоТейлора). Параметр функції має змінюватися від заданого з клавіатури початкового значення, до заданого користувачем кінцевого значення із певним кроком. Розвинення функції в ряд здійснювати із заданої з клавіатури точністю (наприклад, 10-5). Для розвинення функції у ряд Маклорена (абоТейлора) створити власну функцію, яка розраховує суму ряду за рекурентним співвідношенням. У випадку, коли для вибраних значень аргументів функція не визначена, вивести відповідне повідомлення (‘function not defined’).

Виведення результатів подати в табличному вигляді відповідно до рис. 8.

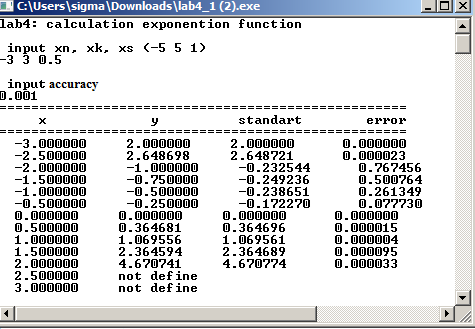


Рис. 8. Зразок результату роботи програми лабораторної роботи №4

### Варіанти завдань

|  |  |
| --- | --- |
| Обчислити значення функції у, розвинувши функцію cos(x) у ряд Тейлора. Аргумент х змінюється від -2 до 2 з кроком 0.5. Визначити похибку. J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var1.gif | Обчислити значення функції у, розвинувши функцію у ряд Тейлора. Аргумент х змінюється від -2 до 2 з кроком 0.5. Визначити похибку. J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var2.gif |
| Обчислити значення функції у, розвинувши функцію е-ху ряд Тейлора. Аргумент х змінюється від -2 до 2 з кроком 0.5. Визначити похибку.J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var3.gif | Обчислити значення функції у, розвинувши функцію sin(x) у ряд Тейлора. Аргумент х змінюється від -2 до 2 з кроком 0.5. ВизначитиJ:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var4.gif |
| Обчислити значення функції у, розвинувши функцію ln(x) у ряд Тейлора. Аргумент х змінюється від -1 до 3 з кроком 0.5. Визначити похибкуJ:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var5.gif | Обчислити значення функції у, розвинувши функцію arctg(x) у ряд Тейлора. Аргумент х змінюється від 0 до 3 з кроком 0.5. Визначити похибку J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var6.gif |
| Обчислити значення функції у, розвинувши функцію sh(x) у ряд Тейлора. Аргумент х змінюється від 0 до 3 з кроком 0.5. Визначити похибкуJ:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var7.gif | Обчислити значення функції у, розвинувши функцію ex у ряд Тейлора. Аргумент х змінюється від -2 до 2 з кроком 0.5. Визначити похибку. J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var8.gif |
| 1. Обчислити значення функції у, розвинувши функцію ln(x) у ряд Тейлора. Аргумент х змінюється від -2 до 2 з кроком 0.5. Визначити **похибку** | Обчислити значення функції у, розвинувши функцію sin(x) у ряд Тейлора. Аргумент х змінюється від -2 до 2 з кроком 0.5. Визначити похибку J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var10.gif |
| Обчислити значення функції у, розвинувши функцію cos(x) у ряд Тейлора. Аргумент х змінюється від -2 до 2 з кроком 0.5. Визначити похибку J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var11.gif | Обчислити значення функції у, розвинувши функцію e-x у ряд Тейлора. Аргумент х змінюється від -2 до 2 з кроком 0.5. Визначити похибку. J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var12.gif |
| Обчислити значення функції у, розвинувши функцію у ряд Тейлора. Аргумент х змінюється від -2 до 2 з кроком 0.5. Визначити похибку. J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var13.gif | Обчислити значення функції у, розвинувши функцію tg(x) у ряд Тейлора подавши її через sin(x) та cos(x). Аргумент х змінюється від -3 до 3 з кроком 0.5. Визначити похибку J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var14.gif |
| Обчислити значення функції у, розвинувши функцію cos(x) у ряд Тейлора. Аргумент х змінюється від -2 до 3 з кроком 0.5. Визначити похибку J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var15.gif | Обчислити значення функції у, розвинувши функцію arctg(x) у ряд Тейлора. Аргумент х змінюється від -3 до 3 з кроком 0.5. Визначити похибку |
| Обчислити значення функції у, розвинувши функцію tg(x) у ряд Тейлора подавши її через sin(x) та cos(x).. Аргумент х змінюється від -3 до 3 з кроком 0.5. Визначити похибкуJ:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var17.gif | Обчислити значення функції у, розвинувши функцію sh(x) у ряд Тейлора. Аргумент х змінюється від -2 до 2 з кроком 0.5. Визначити похибку y=J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var18.gif |
| Обчислити значення функції у, розвинувши функцію ln(1+x) у ряд Тейлора. Аргумент х змінюється від -3 до 3 з кроком 0.5. Визначити похибку | Обчислити значення функції у, розвинувши функцію ex у ряд Тейлора. Аргумент х змінюється від -2 до 2 з кроком 0.5. Визначити похибку. |
| Обчислити значення функції у, розвинувши функцію e-x у ряд Тейлора. Аргумент х змінюється від -2 до 2 з кроком 0.5. Визначити похибку.J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var21.gif | Обчислити значення функції у, розвинувши функцію sh(x) у ряд Тейлора. Аргумент х змінюється від -3 до 3 з кроком 0.5. Визначити похибку. J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var20.gif |
| Обчислити значення функції у, розвинувши функцію ln(x) у ряд Тейлора. Аргумент х змінюється від -2 до 4 з кроком 0.5. Визначити похибку. J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var23.gif | Обчислити значення виразу,J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var24.gifвикориставши ряди для визначення констант π та е: π/4 = 1 - 1/3 + 1/5 - 1/7 + 1/9 -… ;e = 1 + 1/1! + 1/2! + 1/3! +…. Визначити похибку обчислень. |

## Список літератури

1. <https://github.com/tkovalyuk/Basics-of-programming>
2. **МакГрат М**. Программирование на Python для начинающих. –Москва: Эксмо. – 192 с.
3. **Лутц М.** Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
4. **Лутц М.** Программирование на Python, том I, 2. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
5. **Васильев А. Н.** Python на примерах. Практический курс ·по программированию. - СПб.:Наука и Техника, 2016. - 432 с.
6. **Доусон М.** Программируем на Python. - СПб.: Питер, 2014. - 416 с.
7. **Рейтц К., Шлюссер Т.** Автостопом по Python. — СПб.: Питер, 2017. — 336 с.

# Лабораторна робота 5 Рекурсивні функції в Python

**Мета роботи:** Ознайомитись з технологією використання рекурсивних функцій в Python. Побудувати блок-схему алгоритму, розробити програму з використанням рекурентних співвідношень для реалізації рекурсивних процесів.

## Рейтинг лабораторної роботи №5

|  |  |  |  |
| --- | --- | --- | --- |
| № п.п | Вид діяльності студента | Рейтинговий бал | Deadline |
| 1 | Написання коду | 1 | Жовтень |
| 2 | Захист роботи | 0,5 |
| 3 | Звіт з роботи | 0,5 |
| 4 | Якість роботи | 1 |
| Разом за роботу | | 3 |  |

## Теоретичний матеріал

Означення називається рекурсивним, якщо воно задає елементи множини за допомогою інших елементів цієї самої множини. Об'єкти, що задані рекурсивним означенням, також називаються рекурсивними. Під терміном "рекурсивно задана функція" розуміється така функція, значення якої для даноrо apгумента обчислюється за допомогою значень для попередніх apгументів. При кожній конкретній реалізації процес обчислення такої функції задається *рекурентною послідовністю*.

Рекурсія — це такий спосіб організації обчислювального процесу, за якого функція звертається сама до себе. Такі звернення називаються *рекурсивними викликами,* а функція, що містить рекурсивні виклики, — *рекурсивною*.

Перед тим, як записати рекурсивну функцію будь-якою мовою програмування, необхідно записати рекурентне співвідношення, що визначає метод обчислення функцій. Рекурентне співвідношення повинне містити як мінімум дві умови:

* умову продовження рекурсії (крок рекурсії);
* умову закінчення рекурсії.

Розглянемо функцію , де аргумент  є цілим додатним числом, таким, що  або . Потрібно обчислити значення членів послідовності  Рекурсивне завдання послідовності  як послідовності значень функції  включає такі два етапи.

1. Функція  задається безпосередньо у вигляді числових значень для деякої конечної множини початкових значень аргументу .
2. Задається метод або формула, які дозволяють, знаючи усі значення функції при , обчислити її значення при , тобто знайти .

В результаті отримуємо peкурентні співвідношення, що описують рекурсивно задану функцію , рівносильну послідовності  .

Отриманий запис цих двох етапів



називається рекурентним співвідношенням.

Рекурсію реалізовують за допомогою виклику функції самої себе. При цьому в тілі функції спочатку слід перевіряти умову продовження рекурсії. Якщо вона істинна, то виходити з функції. Інакше здійснювати рекурсивний крок.

В рекурсивних підпрограмах можна виділити два процеси: *рекурсивне занурення* підпрограми у себе, що відбувається доти, доки параметр не сягне граничного значення, та *рекурсивне повернення* з підпрограми, що відбувається доти, доки параметр не сягне початкового значення. Рекурсія реалізує нескінченний цикл, отже застосовувати її слід тільки у випадках, коли без неї неможливо обійтися.

Розглянемо рекурсивну функцію, що обчислює *факторіал числа*. Значення факторіалу визначаються рекурентним співвідношенням:



Програмний код, що реалізує рекурсивні обчислення, такий:

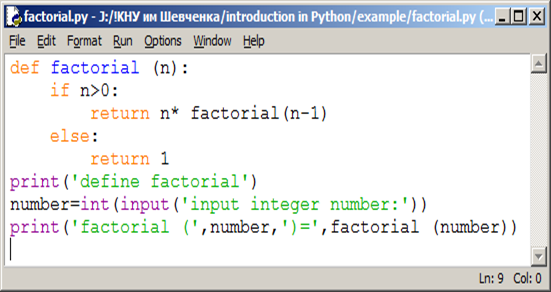


Рис.1. Код обчислення факторіалу числа

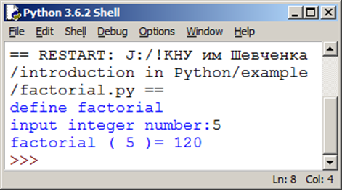


Рис.2. Результат обчислення факторіалу числа

## Завдання для самостійної роботи (за варіантами)

Для виконання завдань використовувати рекурсивні виклики функцій. Для оцінки ефективності рекурсії визначити її глибину.

### Варіанти завдань

1. Знайти суму цифр натурального числа *n* та значення глибини рекурсії, використовуючи рекурентне означення функції *f(n)*:



*Підказка*. Умова продовження рекурсії: сума цифр числа дорівнює останній цифрі плюс сума цифр числа без останньої цифри (числа, що ділиться без остачі на 10). Умова закінчення рекурсії: якщо число дорівнює 0, то сума його цифр дорівнює 0.

1. Знайти кількість одиниць в двійковому представленні числа *n* та значення глибини рекурсії*,* використовуючи рекурентне означення функції *f(n)* (& - операція побітового логічного множення):

****

1. Знайти значення біноміального коефіцієнта та глибини рекурсії при заданих *n, k* за формулою:

 = ,

використовуючи рекурентне співвідношення:



1. Знайти значення функції Аккермана *A(m, n),* використовуючи рекурентне співвідношення:



Визначити глибину рекурсії.

1. Визначити рекурсивну функцію та глибину рекурсії в процесі обчислення степеня дійсного числа *х* з цілим показником *n* ()*,* згідно з рекурентним співвідношенням:



1. Коефіцієнти розкладання бінома (*a+b*)*i*, тобто біноміальні коефіцієнти, утворюють *і*-й рядок трикутника Паскаля. Кожне число в трикутнику, крім перших трьох, є сумою чисел, розташованих над ним у попередньому рядку. Число в *i*-му рядку (*i*= 0, 1, 2, …) на *j-*му місці (*j*= 0, 1, …, *i*), задається формулою . «Верхівка» трикутника має наступний вигляд.



Надрукувати перші  рядків трикутника Паскаля, задавши значення *n* з клавіатури, використавши рекурсивне визначення його елементів.

1. Задані натуральні числа а, c, m. Визначити рекурсивну функцію та глибину рекурсії для обчислення  за формулою:

,

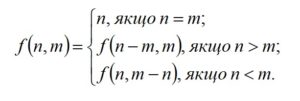
де  - залишок від ділення  на 10.

1. По заданим *b, p, m* обчислити значення виразу . Обчислення виконати, використовуючи алгоритм, що базується на двійковому розкладанні показника степені p:



Операція mod визначає остачу від ділення цілих чисел (ділення за модулем). Визначити глибину рекурсії.

1. Визначити найбільший спільний дільник двох цілих чисел за формулою Евкліда та розрахувати глибину рекурсії:

[](https://www.bestprog.net/wp-content/uploads/2019/01/05_02_02_06_07_01u.jpg)

1. Дано натуральне число *n*> 1. Перевірте, чи є воно простим. Програма повинна вивести слово YES, якщо число просте і NO, якщо число складене.

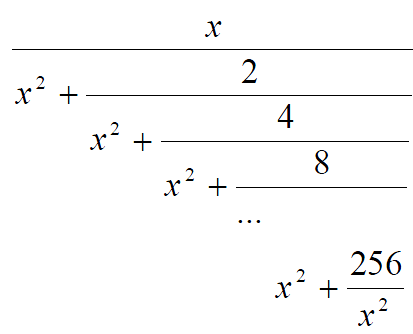
*Підказка*. потрібно зробити ще один параметр рекурсії: дільник числа, і саме за цим параметром і робити рекурсію.

1. Дано натуральне число *n*>1. Вивести всі прості дільники цього числа в порядку неспадання з урахуванням кратності. Використати рекурсію та розрахувати її глибину.
2. Дано натуральне число *N*. Обчислити суму його цифр. Використати рекурсію та розрахувати її глибину
3. Визначити рекурсивну функцію NOD(*a, b*) цілого типу, яка знаходить найбільший спільний дільник (НСД) двох натуральних чисел *a* і *b*, використовуючи алгоритм Евкліда: НСД (*a, b*) = НСД (*b, a* mod *b*), якщо b≠0, НСД (a, 0) = a. За допомогою цієї функції знайти НСД (*a, b*), НСД (*a, c*), НСД (*a, d*), якщо задані числа *a,b,c,d.* Визначити глибину рекурсії. Операція mod визначає остачу від ділення двох цілих чисел.
4. Створити рекурсивну функцію для перекладу числа з десяткової системи числення в двійкову та з двійкової системи числення в десяткову. Визначити глибину рекурсії.
5. Дані два числа *a і b*. Визначити, скільки існує послідовностей з *a* нулів і *b* одиниць, в яких ніякі два нуля не стоять поруч. Оригінал умови за URL: <https://habr.com/ru/post/275813/>. Визначити глибину рекурсії.
6. . Описати рекурсивну функцію RootK (*X, K, N*), що знаходить наближене значення кореня *K*-го ступеня з числа *X* за формулою:

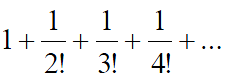


де *Yn* позначає RootK (*X, K, N*) при фіксованих *X і K*. Параметри функції: X (> 0) - дійсне число, k > 1 і n > 0 - цілі. За допомогою функції RootK знайти для даного числа X наближені значення його кореня K-го ступеня при різних значеннях *n.* Оригінал умови за URL: <http://coolcode.ru/rekursiya-gruppa-recur/>

1. Дано натуральне число *n*. Обчислити суму ряду, елементи якого відповідають виразу: 1 × 2 + 2 × 3 + 3 × 4 +…+ (*n* –1) × *n*. Використати рекурсію та обчислити її глибину.
2. Обчислити ланцюговий дріб, задавши значення *х* з клавіатури та використавши рекурсію. Обчислити глибину рекурсії.



1. Знайти суму ряду, використавши рекурсивні функції обчислення факторіалу цілого числа та суми ряду. Визначити глибину рекурсії



1. Обчислити значення виразу



Для цього використати рекурсію при *n*=5, 10 (*n* – кількість членів послідовності). Розрахувати глибину рекурсії.

1. Обчислити значення *x =√ a* , використовуючи формулу , в якості початкового наближення використовувати значення *x*0 = (1+*a*)/2.
2. Обчислити добуток двох цілих додатних чисел p = a× b⋅ за наступним рекурентним співвідношенням:

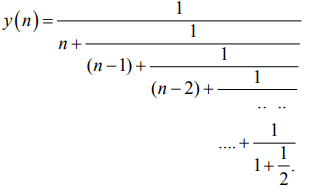


Використати рекурсію та розрахувати її глибину.

1. Обчислити значення полінома ступеня *n* за формулою

Використати рекурсію та розрахувати її глибину.

1. Обчислити ланцюговий дріб, задавши значення *n* з клавіатури та використавши рекурсію. Обчислити глибину рекурсії для заданого *n*



1. Обчислити значення виразу *m*!/(*m*+*n*)! при введених з клавіатури значеннях *m* i *n.* Використати рекурсивну функцію для обчислення факторіалу та розрахувати глибину рекурсії.

## Список літератури

1. <https://github.com/tkovalyuk/Basics-of-programming>
2. **МакГрат М**. Программирование на Python для начинающих. –Москва: Эксмо. – 192 с.
3. **Лутц М.** Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
4. **Лутц М.** Программирование на Python, том I, 2. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
5. **Васильев А. Н.** Python на примерах. Практический курс ·по программированию. - СПб.:Наука и Техника, 2016. - 432 с.
6. **Доусон М.** Программируем на Python. - СПб.: Питер, 2014. - 416 с.
7. **Рейтц К., Шлюссер Т.** Автостопом по Python. — СПб.: Питер, 2017. — 336 с.

# Лабораторна робота 6 Обробка рядків в Python

**Мета роботи:** Ознайомитись з технологією обробки рядків в Python. Побудувати блок-схему алгоритму, розробити програму з використанням функцій і методів обробки рядків

## Рейтинг лабораторної роботи №6

|  |  |  |  |
| --- | --- | --- | --- |
| № п.п | Вид діяльності студента | Рейтинговий бал | Deadline |
| 1 | Написання коду | 1 | Жовтень |
| 2 | Захист роботи | 1 |
| 3 | Звіт з роботи | 1 |
| 4 | Якість роботи | 1 |
| Разом за роботу | | 4 |  |

## Теоретичний матеріал

Рядок - це послідовність символів, укладених в одинарні або подвійні лапки.

Рядок може складатися з декілька підрядків, тоді використовуються по три одинарні або три подвійні лапки

Нумерація символів в рядку починається з 0. Номери символів в рядку називаються **індексом**.

В Python немає символьного типу, тобто типу даних, об'єктами якого є поодинокі символи. Однак Python дозволяє розглядати рядки як об'єкти, що складаються з підрядків довжиною в один і більше символів.

Рядок - це впорядкована послідовність елементів. Отже, з неї можна витягувати окремі символи і зрізи.

Створивши рядок, можна отримати будь-яку його частину як новий рядок. Ця частина називатиметься **зрізом**.

Якщо вказати від'ємне значення індексу, то номер буде відраховуватися з кінця, починаючи з номера -1.

Рядки в Python відносяться до категорії **незмінних послідовностей**, тобто всі функції і методи можуть лише створювати новий рядок.

**Зріз (slice)** - витяг з рядка одного символу або деякого фрагмента підрядка або підпослідовності.

**Взяття одного символу рядка**: S [i] – це зріз, що складається з одного символу, який має номер i.

**Зріз з двома параметрами:** S [a: b] повертає підрядок з **b - a** символів, починаючи з символу з індексом **a** до символу з індексом b, не включаючи його.

**Зріз з трьома параметрами:** Якщо задати зріз з трьома параметрами S [a: b: d], то третій параметр задає крок, тобто будуть взяті символи з індексами a, a + d, a + 2 \* d і т. д

### Функції і методи роботи з рядками

|  |  |
| --- | --- |
| **Функція або метод** | **Призначення** |
| **S = 'str'; S = "str";**  **S = '''str'''; S = """str"""** | Літерали рядків |
| **S = "s\np\ta\nbbb"** | Екрановані послідовності |
| **S = r"C:\temp\new"** | Неформатовані рядки (пригнічують екранування) |
| **S = b"byte"** | Рядок байтів |
| **S1 + S2** | Конкатенація (додавання рядків) |
| **S1 \* 3** | Повторення рядка |
| **S[i]** | Звернення за індексом |
| **S[i:j:step]** | Витяг зрізу |
| **len**(S) | Довжина рядка |
| **S.find**(str, [start],[end]) | Пошук підрядка в рядку. Повертає номер першого входження або -1 |
| **S.rfind**(str, [start],[end]) | Пошук підрядка в рядку. Повертає номер останнього входження або -1 |
| **S.index(str, [start],[end])** | Поиск подстроки в строке. Возвращает номер первого вхождения или вызывает ValueError |
| **S.rindex(str, [start],[end])** | Поиск подстроки в строке. Возвращает номер последнего вхождения или вызывает ValueError |
| **S.replace(шаблон, замена)** | Замена шаблона |
| **S.split(символ)** | Разбиение строки по разделителю |
| **S.isdigit()** | Состоит ли строка из цифр |
| **S.isalpha()** | Состоит ли строка из букв |
| **S.isalnum()** | Состоит ли строка из цифр или букв |
| **S.islower()** | Состоит ли строка из символов в нижнем регистре |
| **S.isupper()** | Состоит ли строка из символов в верхнем регистре |
| **S.isspace()** | Состоит ли строка из неотображаемых символов (пробел, символ перевода страницы ('\f'), "новая строка" ('\n'), "перевод каретки" ('\r'), "горизонтальная табуляция" ('\t') и "вертикальная табуляция" ('\v')) |
| **S.istitle()** | Начинаются ли слова в строке с заглавной буквы |
| **S.upper()** | Преобразование строки к верхнему регистру |
| **S.lower()** | Преобразование строки к нижнему регистру |
| **S.startswith(str)** | Начинается ли строка S с шаблона str |
| **S.endswith(str)** | Заканчивается ли строка S шаблоном str |
| **S.join(список)** | Сборка строки из списка с разделителем S |
| **ord(символ)** | Символ в его код ASCII |
| **chr(число)** | Код ASCII в символ |
| **S.capitalize()** | Переводит первый символ строки в верхний регистр, а все остальные в нижний |
| **S.center(width, [fill])** | Возвращает отцентрованную строку, по краям которой стоит символ fill (пробел по умолчанию) |
| **S.count(str, [start],[end])** | Возвращает количество непересекающихся вхождений подстроки в диапазоне [начало, конец] (0 и длина строки по умолчанию) |
| **S.expandtabs([tabsize])** | Возвращает копию строки, в которой все символы табуляции заменяются одним или несколькими пробелами, в зависимости от текущего столбца. Если TabSize не указан, размер табуляции полагается равным 8 пробелам |
| **S.lstrip([chars])** | Удаление пробельных символов в начале строки |
| **S.rstrip([chars])** | Удаление пробельных символов в конце строки |
| **S.strip([chars])** | Удаление пробельных символов в начале и в конце строки |
| **S.partition(шаблон)** | Возвращает кортеж, содержащий часть перед первым шаблоном, сам шаблон, и часть после шаблона. Если шаблон не найден, возвращается кортеж, содержащий саму строку, а затем две пустых строки |
| **S.rpartition(sep)** | Возвращает кортеж, содержащий часть перед последним шаблоном, сам шаблон, и часть после шаблона. Если шаблон не найден, возвращается кортеж, содержащий две пустых строки, а затем саму строку |
| **S.swapcase()** | Переводит символы нижнего регистра в верхний, а верхнего – в нижний |
| **S.title()** | Первую букву каждого слова переводит в верхний регистр, а все остальные в нижний |
| **S.zfill(width)** | Делает длину строки не меньшей width, по необходимости заполняя первые символы нулями |
| **S.ljust(width, fillchar=" ")** | Делает длину строки не меньшей width, по необходимости заполняя последние символы символом fillchar |
| **S.rjust(width, fillchar=" ")** | Делает длину строки не меньшей width, по необходимости заполняя первые символы символом fillchar |
| **S.format(\*args, \*\*kwargs)** | [Форматирование строки](https://pythonworld.ru/osnovy/formatirovanie-strok-metod-format.html) |

#### Приклад програми1 обробки рядків. HIPO діаграма

#### Умова

Ввести рядок. Витягти ім’я файлу з URL. Видалити пробіли на початку та в кінці рядка. Підрахувати кількість великих та маленьких латинських літер в рядку. Змінити цифрові символи на пробіли. Перевірити складність пароля на наявність цифрових символів, великих літер, довжини не менше 5.

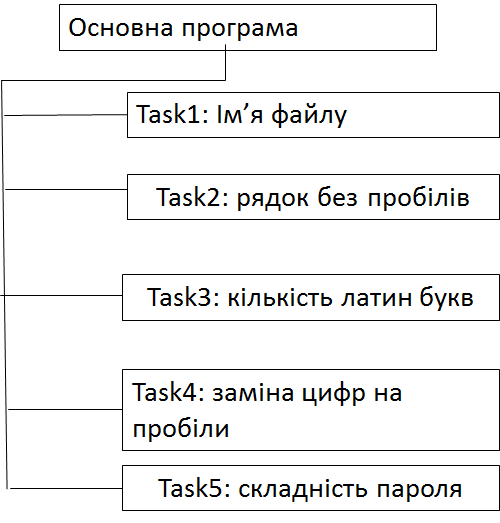


Рис.1. HIPO діаграма виклику функцій

#### Приклад коду

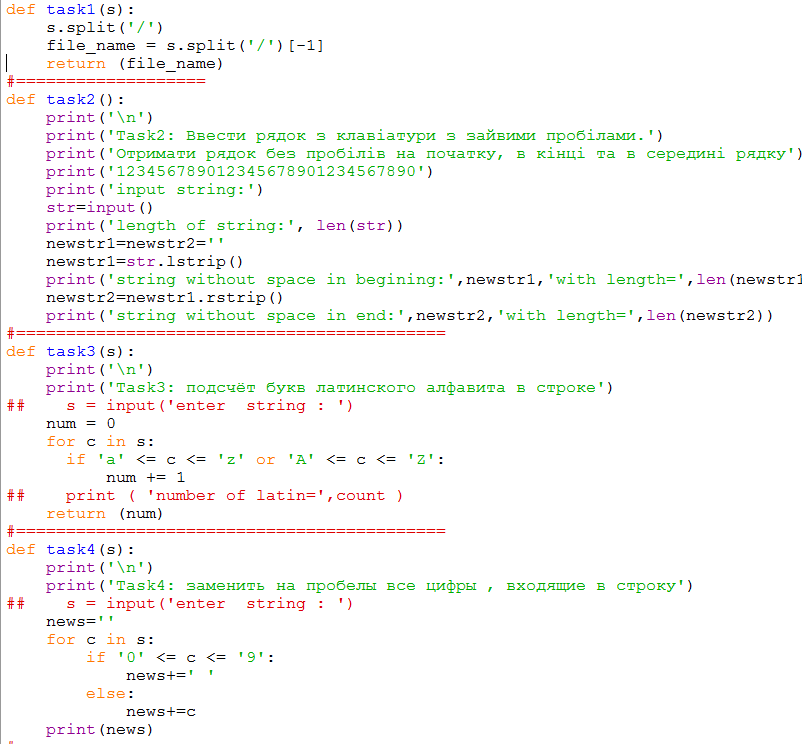


Рис.2. Частина коду програми. Функції.

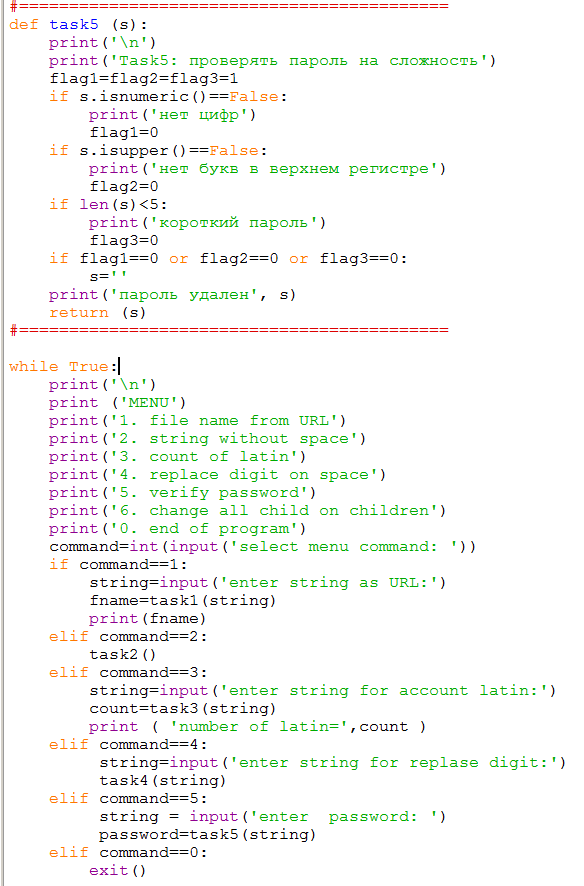


Рис.3. Частина коду програми. Основна програма.

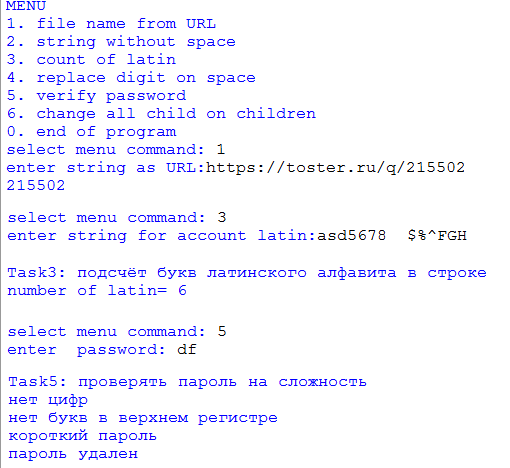


Рис.2. Фрагменти результатів роботи програми.

## Завдання для самостійної роботи (за варіантами)

### Варіанти завдань

1. У рядку символів визначити кількість слів і знайти найдовше слово. Вилучити з рядка однолітерні слова та зайві пропуски, коми, крапки. Слова відділяються один від одного довільною кількістю пропусків, ком і крапок.
2. У рядку символів визначити кількість слів, що повторюються, та вилучити дублікати. Слова відокремлюються пробілами.
3. З рядку вилучити всі слова на непарних порядкових позиціях, а слова на парних позиціях надрукувати перевернутими.
4. Ввести два рядки, вилучити з першого рядка всі слова, які зустрічаються у другому рядку.
5. У рядку символів визначити слова-паліндроми, які однаково читаються справа наліво і навпаки. Знайти найбільшу довжину цих слів. Якщо таких слів немає, то з кожного слова на непарній позиції вилучити голосні та надрукуйте їх.
6. У рядку символів визначити слова, що починаються і закінчуються на літери, значення яких уводять з клавіатури. Надрукувати їх у перевернутому вигляді (справа наліво). Визначити частоту кожної голосної літери.
7. У рядку слів визначити кількість слів і надрукувати ті, довжина яких більша за вказану користувачем. Якщо слово починається із заданого користувачем символу, то замінити його на перше слово у рядку.
8. Увести рядок символів та отримати новий рядок, слова якого відсортовані за алфавітом. Отримати інший рядок, у якому слова відсортовані за довжиною слів. Надрукувати вхідний та відсортовані рядки.
9. Розбити на склади згідно з правилами перенесення слів кожне слово на парній позиції у рядку. Слова на непарних позиціях інвертувати (записати у зворотному порядку).
10. Увести рядок, що містить арифметичний вираз (дужки, символи арифметичних операцій, операнди). Визначити коректність виразу (парність дужок, коректність операцій та операндів).
11. Ви перебуваєте в квест-кімнаті по мультфільму WALL-E! Щоб вибратися з кімнати, необхідно виконати такі кроки: а) Дізнайтеся довжину рядка s наведеного тексту і виведіть це значення в консоль. б) Виведіть весь текст в нижньому регістрі.
12. Вводиться рядок, що включає великі та малі літери. Потрібно вивести той самий рядок в одному регістрі, який залежить від того, яких букв більше. При рівній кількості перетворити усі літери в нижній регістр та порахувати кількість голосних.
13. Визначити, скільки разів в тексті зустрічається кожне слово. Слово, яке повторюється найчастіше, розкласти на склади. Вивести слова та їх кількість повторів.
14. Ввести рядок і символ. Підрахувати найдовшу послідовність поспіль букви, значення якої введено з клавіатурі. Перетворити рядок, замінивши точками все знаки оклику.
15. Ввести рядок символів, серед яких є дужки, що відкриваються і закриваються. Вивести на екран усі символи, розташовані всередині цих дужок, і підрахувати їх кількість.
16. Ввести рядок, що складається з декількох речень. Кожне речення закінчується крапкою. Порахувати кількість речень. Перетворити рядок так, щоб кожне речення починалося з великої літери та з нового рядка.
17. Роздрукуйте АSCII таблицю, подавши її в 10 колонок. Кожна пара колонок має заголовок «Код» та «Значення символу».
18. Визначте загальну кількість слів у тексті, загальну кількість різних слів (без повторів) та кількість унікальних слів, що зустрічаються тільки один раз. Знайдіть найкоротше слово в тексті.
19. Визначте, з якого слова найчастіше починаються речення у тексті, а також яким найчастіше закінчуються. Визначте частоту появи речень з різною кількістю слів.
20. .Визначте процент води у тексті (це кількість "стоп-слів" поділена на загальну кількість слів). Стоп-слова – це слова, які ігноруються при індексації сторінок пошуковими системами, не несуть смислового навантаження, замінюються маркерами і негативно впливають на якість текстів, знижуючи їх корисність. Списки стоп-слів див. в Інтернеті.
21. Визначте максимальну, мінімальну та середню довжину слів, речень та абзаців у тексті. Текст – це масив рядків.
22. Ввести речення, що закінчується крапкою. Знайти слово, яке найчастіше повторюється та вивести позиції повторів цього слова.
23. Увести декілька рядків. Об'єднати перші два, встановивши довжину об'єднаного рядка не більше за задану користувачем кількість символів. Символи, що залишилися, приєднати до третього рядка. Повторити операцію для наступної групи рядків.
24. Увести два рядки символів та замінити кожне слово на парній позиції першого рядка на слово, що стоїть на непарній позиції у другому рядку. Кількість слів у рядках може бути різною.
25. Увести рядок символів та отримати новий рядок, слова якого відсортовані за алфавітом. Отримати інший рядок, у якому слова відсортовані за довжиною слів.

## Список літератури

1. <https://github.com/tkovalyuk/Basics-of-programming>
2. **МакГрат М**. Программирование на Python для начинающих. –Москва: Эксмо. – 192 с.
3. **Лутц М.** Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
4. **Лутц М.** Программирование на Python, том I, 2. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
5. **Васильев А. Н.** Python на примерах. Практический курс ·по программированию. - СПб.:Наука и Техника, 2016. - 432 с.
6. **Доусон М.** Программируем на Python. - СПб.: Питер, 2014. - 416 с.
7. **Рейтц К., Шлюссер Т.** Автостопом по Python. — СПб.: Питер, 2017. — 336 с.

# Лабораторна робота 7 Обробка списків в Python

**Мета роботи:** Ознайомитись з технологією обробки списків в Python. Побудувати блок-схеми алгоритмів, розробити програми з використанням функцій і методів обробки списків у вигляді одновимірних масивів та матриць.

## Рейтинг лабораторної роботи №7

|  |  |  |  |
| --- | --- | --- | --- |
| № п.п | Вид діяльності студента | Рейтинговий бал | Deadline |
| 1 | Написання коду | 3 | Листопад |
| 2 | Захист роботи | 1 |
| 3 | Звіт з роботи | 1 |
| 4 | Якість роботи | 1 |
| Разом за роботу | | 6 |  |

## Теоретичний матеріал

Список у Python – це впорядкована колекція (тобто індексована послідовність) об'єктів довільних типів (в більшості мов програмування використовується термін "масив")

Список являє собою послідовність елементів, пронумерованих від 0, як символи в рядку.

Списки у Python належать до **змінюваних (mutable) типів.**

Списки можуть складатися з різних об’єктів: чисел, рядків та інших списків, в останньому випадку списки називають вкладеними.

Списки служать для того, щоб зберігати об’єкти в певному порядку, особливо якщо порядок або вміст можуть змінюватися.

Одне і те саме значення може зустрічатися в списку кілька разів.

Операції зі списками:

* + змінити список,
  + додати в нього нові елементи,
  + видалити або перезаписати існуючі елементи,
  + змінити кількість елементів у списку,
  + змінити окремі елементи.

### Методи роботи зі списками

|  |  |
| --- | --- |
| Метод | Призначення |
| list.append(x) | Додає елемент в кінець списку |
| list.extend(L) | Розширює список list, додаючи в кінець все елементи списку L |
| list.insert(i, x) | Вставляє значення x на i-ий індекс |
| list.remove(x) | Видаляє перший елемент у списку, який має значення x. Повертає ValueError, якщо такого елемента не існує |
| list.pop([i]) | Видаляє i-ий елемент і повертає його. Якщо індекс не вказано, видаляється останній елемент |
| list.index(x, [start [, end]]) | Повертає положення першого елемента зі значенням x (при цьому пошук ведеться від start до end) |
| list.count(x) | Повертає кількість елементів зі значенням x |
| list.sort([key=функция]) | Сортує список на основі функції |
| list.reverse() | Перевертає список |
| list.copy() | Копія списку |
| list.clear() | Очищає список |

#### Введення списків

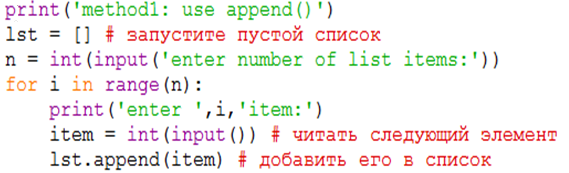


Рис.1. Введення списку з клавіатури через метод append()

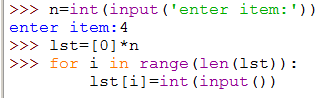


Рис.2. Введення списку з клавіатури через операцію присвоєння

#### Генерація списків

1. Для генерації застосовують функції **randint()** або **random()** модуля **random**
2. Генерація списку може здійснюватися циклос з умовою або без неї

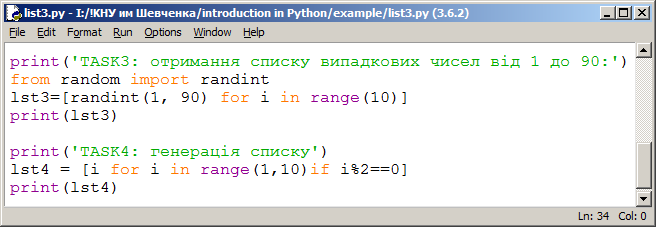


Рис.3. Приклади генерації списку

#### Операції над списками

##### Конкатенація, множення на ціле число та функція len()

Конкатенація двох списків утворює новий список, що складається з першого списку до кінця якого дописано другий список.

Множення списку на ціле число утворює новий список, до кінця якого дописано той самий список.

Функція len() повертає кількість елементів (довжину) списку, який передається як аргумент функції

##### Перевірка входження елементу у список

Операція **in** повертає значення **True**, якщо елемент **x** входить до списку **lst.**

Операція **not in** повертає значення **True**, якщо елемент **x** не входить до списку **lst**

##### Присвоєння списків

Операція **=** повертає копію списку, при цьому зміна елементів одного списку **приводить** до зміни відповідного елемента копії. Операція **=** зрізу всього списку повертає копію списку, при цьому зміна елементів одного списку **не приводить** до зміни відповідного елемента зрізу (тобто копії) списку.

#### Приклад програми1 обробки списків (одновимірних масивів)

##### Умова

1. Ввести список цілих чисел з клавіатури.
2. Вивести усі непарні числа та числа, що мають парні індекси.
3. Обчислити суму від'ємних елементів списку.
4. Знайти перший додатний елемент списку та його індекс.
5. Знайти мінімальний додатний елемент списку та його індекс.

##### Код

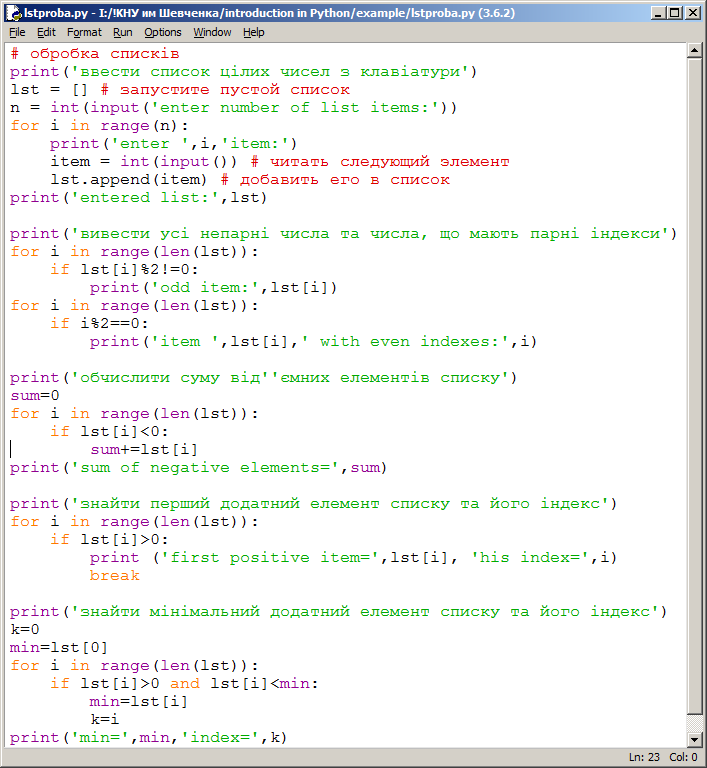


Рис.4. Приклад коду програми.

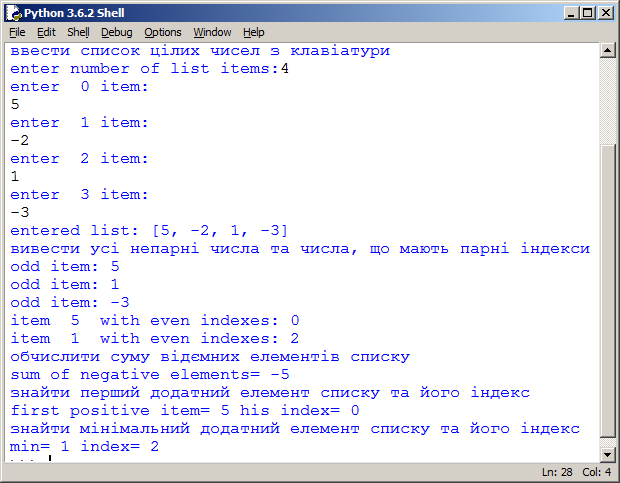


Рис.5. Результат роботи програми.

#### Приклад програми2 обробки матриць (двовимірних списків)

##### Умова

Перемножити дві матриці, які вводяться з клавіатури. У випадку несумісності матриць (кількість стовпців першої матриці не дорівнює кількості рядків другої) вивести відповідне повідомлення.

##### Приклад коду

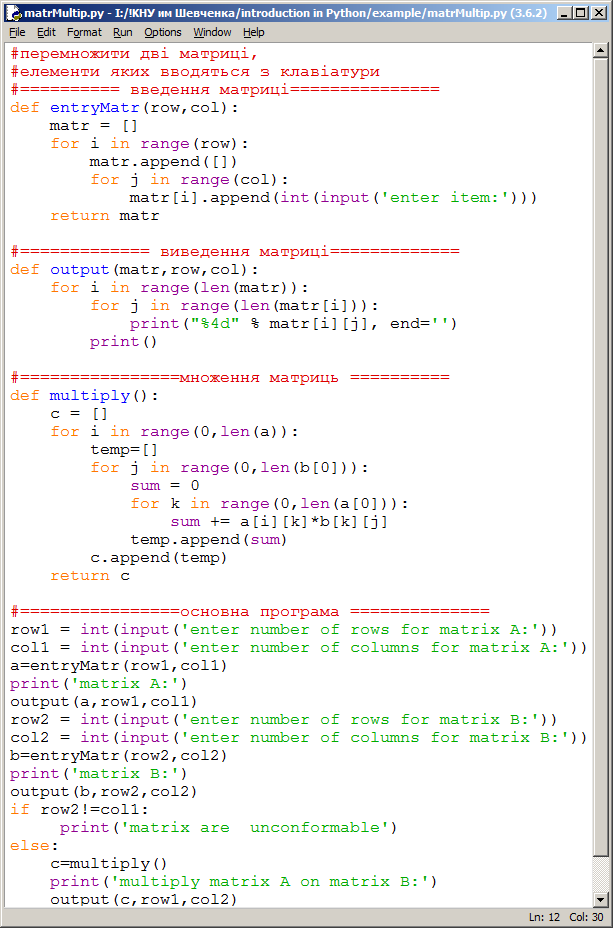


Рис. 6 Код програми множення двох матриць

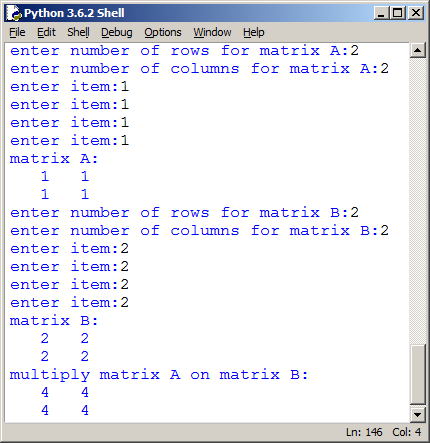


Рис. 7. Результат роботи програми множеня матриць

## Завдання для самостійної роботи за варіантами

Кожне завдання складається з двох задач: обробка одновимірних масивів (списків), обробка матриць (складених списків). Завдання має бути реалізовано у вигляді одної програми, яка має меню вибору задач. Код завдання повинний складатися з функцій, наприклад, введення вхідних даних, виведення результатів, розрахунок результату тощо.

**Під час виконання лабораторної роботи не використовувати бібліотеки Pandas та Numpy**.

### Варіанти завдань

1. **Скласти програму для обробки одновимірного масиву та матриці. Передбачити меню вибору задач.**
2. Ввести список чисел з клавіатури. Елементи списку є додатними, від’ємними та рівними нулю числами. З введеного списку сформувати новий список, взявши в нього тільки ті елементи із вхідного списку, які більше за модулем введеного з клавіатури числа M. Вивести на екран число M, вхідний та отриманий списки.
3. Побудувати квадратну матрицю, в якій елементи на головній діагоналі, що проходить з лівого верхнього кута в правий нижній мають значення 1, елементи, що знаходяться вище головної діагоналі - значення 0, елементи, що знаходяться нижче головної діагоналі - значення 2.
4. **Скласти програму для обробки одновимірного масиву та матриці. Передбачити меню вибору задач.**
5. Створити список чисел, кількість елементів якого задана користувачем. Визначити максимальне та мінімальне значення серед елементів із парними та непарними індексами. Вивести список, мінімальний, максимальний елементи та їх індекси.
6. Ввести вимірність і значення елементів квадратної матриці з клавіатури. Перевірити, чи є ця матриця симетричною відносно головної діагоналі. Вивести слово "YES", якщо матриця симетрична, і слово "NO" в іншому випадку.
7. **Скласти програму для обробки одновимірного масиву та матриці. Передбачити меню вибору задач.**
8. Створити числовий список, кількість елементів якого задана користувачем. Обчислити суму елементів між максимальним та мінімальними значеннями списку. Вивести на екран список, максимальний та мінімальний елементи масиву та їх індекси.
9. Згенерувати матрицю розміром n × m. Значення n, m вводяться з клавіатури. Побудувати транспоновану матрицю і вивести її на екран. В транспонованій матриці рядки вихідного масиву стають стовпцями транспонованого, стовпці вихідного масиву стають рядками транспонованого. Рішення оформити у вигляді функції, яка отримує на вхід даний масив і повертає новий масив.
10. **Скласти програму для обробки одновимірного масиву та матриці. Передбачити меню вибору задач.**
11. Створити два числових списки, кількість елементів яких задана користувачем. Знайти найменший серед тих елементів першого списку, які співпадають із значеннями елементів другого списку. Вивести на екран списки, найменший елемент та його індекс.
12. Ввести вимірність і квадратну матрицю з клавіатури. Поміняти місцями елементи, які стоять на головній і побічній діагоналях, при цьому в кожному стовпці потрібно поміняти місцями елементи на головній діагоналі і на побічній діагоналях .Рішення оформите у вигляді функції.
13. **Скласти програму для обробки одновимірного масиву та матриці. Передбачити меню вибору задач.**
14. Створити список цілих чисел. Побудувати новий список із елементів першого списку. В новому списку спочатку стоять числа, що діляться на 2, потім ті, що діляться на 2 та 3, потім на 3. Надрукувати вхідний та вихідний списки.
15. У лівому верхньому кутку дошки розміром n × m розташований шаховий король. За один хід він може зробити хід вправо, вниз або на одну клітину по діагоналі вправо-вниз. Порахуйте кількість маршрутів, що веде з лівого верхнього кута дошки в правий нижній. Програма отримує на вхід розміри дошки (числа n і m) і повинна вивести кількість шуканих маршрутів.
16. **Скласти програму для обробки одновимірного масиву та матриці. Передбачити меню вибору задач.**
17. Створити список. Обчислити кількість елементів у найдовшій серії. Серія — це послідовність однакових елементів, розташованих поспіль. Надрукувати список, усі серії однакових елементів та довжину найдовшої серії.
18. Згенерувати матрицю розміром n × m. Повернути її на 90 градусів за годинниковою стрілкою, записавши результат в нову матрицю розміром m × n. Вивести згенеровану та повернуту матриці. Рішення оформите у вигляді функції, яка отримує на вхід дану матрицю і повертає нову матрицю.
19. **Скласти програму для обробки одновимірного масиву та матриці. Передбачити меню вибору задач**
20. У згенерованому числовому списку знайти найбільший серед від’ємних та найменший серед додатних елементів списку. Вивести список, значення знайдених елементів та їх індекси.
21. В кінотеатрі n рядів по m місць в кожному. У двовимірному масиві зберігається інформація про продані квитки: число 1 означає, що квиток на дане місце вже проданий, число 0 означає, що місце вільне. Надійшов запит на продаж k квитків на сусідні місця в одному ряду. Визначте, чи можна виконати такий запит.
22. **Скласти програму для обробки одновимірного масиву та матриці. Передбачити меню вибору задач**
23. Згенерувати список з цілих чисел та вивести його на екран. Знайти максимальний елемент та його індекс та надрукувати їх. Вивести на екран кількість елементів, які менші за значення максимального і більше за значення максимального елемента.
24. У заданій квадратній матриці значення деяких діагональних елементів дорівнюють нулю. Переставити рядки або стовпці матриці таким чином, щоб діагональні елементи стали ненульовими. Якщо це неможливо зробити, вивести відповідне повідомлення.
25. **Скласти програму для обробки одновимірного масиву та матриці. Передбачити меню вибору задач**
26. Згенерувати два упорядкованих за зростанням числових списків. Побудувати упорядкований за зростанням третій список злиттям двох згенерованих.
27. Для заданої цілочислової матриці визначити індекси та значення її сідлових точок. Елемент матриці називається сідловою точкою, якщо він є найменшим у своєму рядку та найбільшим у своєму стовпчику.
28. **Скласти програму для обробки одновимірного масиву та матриці. Передбачити меню вибору задач**
29. Ввести з клавіатури два списки цілих чисел. Кількість елементів списків вводиться з клавіатури. Побудувати третій список, в якому кожний елемент дорівнює найбільшому спільному дільнику відповідних елементів введених списків. Використати алгоритм Евкліда. Надрукувати усі три списки.
30. Задана дійсна квадратна матриця розмірності n\*n. Знайти суми тих елементів, які розміщені відповідно нижче, вище та на головній діагоналі, в рядках, що починаються з від’ємного елементу.
31. **Скласти програму для обробки одновимірного масиву та матриці. Передбачити меню вибору задач**
32. Створити числовий список, кількість елементів якого вводиться користувачем. Значення елементів в списку можуть повторюватися. Визначити індекс першого та останнього входження заданого числа в список. Знайти елемент, значення якого повторюється найбільшу кількість разів. Вивести на екран список, шуканий елемент та його індекс.
33. Для заданої матриці та вектора довільної вимірності здійснити множення матриці та вектор та вектора на матрицю. У разі несумісності вимірності матриці та вектора вивести відповідне повідомлення.
34. **Скласти програму для обробки одновимірного масиву та матриці. Передбачити меню вибору задач**
35. Створити список, значеннями елементів якого є коефіцієнти многочлена I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_07\pictures\var11.gif. Знайти значення многочлена при введеному з клавіатури аргументу *x* та похідної від многочлена *A*(*x*) при заданому *x* . Вивести на екран список, значення многочлена та його похідної.
36. В заданій матриці довільної вимірності обчислити суму елементів парних рядків, найбільші значення у непарних рядках, середнє арифметичне серед від’ємних елементів матриці та індекси нульових елементів.
37. **Скласти програму для обробки одновимірного масиву та матриці. Передбачити меню вибору задач**
38. Згенерувати список цілих чисел, кількість елементів якого ввести з клавіатури. Вивести список на екран. Отримати інший список, що складається тільки з непарних чисел вихідного списку або повідомити, що таких чисел немає. Отриманий список вивести на екран в порядку спадання елементів.
39. Згенерувати дійсну матриця розміром n х m Переставляючи її рядки і стовпці, домогтися того, щоб найбільший елемент (або один з них) перемістився у верхній лівий кут. Ввести на екран згенеровану та отриману матриці.
40. **Скласти програму для обробки одновимірного масиву та матриці. Передбачити меню вибору задач**
41. Ввести з клавіатури список цілих чисел. Знайти в списку все прості числа і скопіювати їх в новий список. Надрукувати вхідний та вихідний списки.
42. Згенерувати матрицю додатних та від’ємних чисел, вимірність якої вводиться з клавіатури. Знайти найбільший по модулю елемент. Отримати квадратну матрицю порядку n – 1 шляхом видалення з вихідної матриці рядків і стовпців, на перетині яких розташований елемент зі знайденим значенням. Надрукувати вхідну та отриману матриці.
43. **Скласти програму для обробки одновимірного масиву та матриці. Передбачити меню вибору задач**
44. Згенерувати список чисел, кількість елементів якого ввести з клавіатури. У списку чисел перевірити, чи всі елементи є унікальними, тобто кожне число зустрічається тільки один раз. Надрукувати згенерований список та повідомлення «Усі елементи унікальні» або «Не усі елементи унікальні».
45. Латинським квадратом порядку n називається квадратна матриця розміром n x n, в якій кожний рядок і кожний стовпець містять усі числа від 1 до n. Перевірити, чи є задана матриця латинським квадратом. Надрукувати вхідну матрицю і відповідний висновок.
46. **Скласти програму для обробки одновимірного масиву та матриці. Передбачити меню вибору задач**
47. Згенерувати числовий список, кількість елементів якого ввести з клавіатури. Визначити індекси елементів списку, значення яких не менше заданого мінімуму і не більше заданого максимуму. Вивести на екран список, кількість, індекси та значення елементів, що належать заданому діапазону.
48. Заповнити матрицю випадковими цілими числами. Надрукувати матрицю. Знайти максимальний серед елементів тих стовпців, які впорядковані або за зростанням, або за спаданням. Якщо впорядковані стовпці в матриці відсутні, то вивести відповідне повідомлення.
49. **Скласти програму для обробки одновимірного масиву та матриці. Передбачити меню вибору задач**
50. Згенерувати числовий список з елементами від -100 до +100. Надрукувати список. Вивести послідовність елементів між першим від’ємним і нульовим елементом, між першим додатним і нульовим елементом. Якщо таких елементів немає, по вивести відповідне повідомлення.
51. Ввести розмірність та згенерувати матрицю. У кожному стовпці і кожному рядку матриці мають міститися строго по одному нульового елементу. Перестановкою рядків домогтися розташування всіх нулів на головній діагоналі матриці.
52. **Скласти програму для обробки одновимірного масиву та матриці. Передбачити меню вибору задач**
53. Ввести з клавіатури числовий список з трьох елементів. У кожного елемента списку є вага, яка збільшується на одиницю, коли користувач вибирає цей елемент списку. Початкове значення ваг дорівнює нулю. Користувач має n спроб вибрати кожний елемент списку. Вибір елементів здійснюється випадково за допомогою генератора псевдовипадкових чисел. При випаданні потрібного елемента, його вага збільшується, отже в наступних проходах він буде випадати частіше. Вивести на екран значення ваг елементів після n спроб вибрати їх.
54. Згенерувати квадратну матрицю, задавши її вимірність з клавіатури. Привести матрицю до верхньої трикутної шляхом застосування допустимих перетворень матриць. У верхній трикутній матриці всі елементи, що знаходяться нижче головної діагоналі, дорівнюють нулю
55. **Скласти програму для обробки одновимірного масиву та матриці. Передбачити меню вибору задач**
56. Ввести з клавіатури список чисел. Визначити, скільки в цьому списку елементів, які більше двох своїх сусідів справа та зліва. Вивести кількість таких елементів. Крайні елементи списку и не враховуються, оскільки у них недостатньо сусідів.
57. Згенерувати матрицю розміром N \* N, що містить додатні та від’ємні числа. Здійснити пошук такої підматриці, сума елементів якої є максимальною.
58. **Скласти програму для обробки одновимірного масиву та матриці. Передбачити меню вибору задач**
59. Згенерувати список чисел, кількість елементів якого ввести з клавіатури. Вивести згенерований список. Обчислити кількість пар елементів, рівних один одному. Вважається, що будь-які два елементи, які рівні один одному, утворюють одну пару. Вивести індекси елементів, що утворюють пару та кількість пар.
60. Дана дійсна квадратна матриця порядку 2n. Отримати нову матрицю, переставляючи її блоки розміру n × n за годинниковою стрілкою, починаючи з блоку в лівому верхньому кутку.
61. **Скласти програму для обробки одновимірного масиву та матриці. Передбачити меню вибору задач**
62. Згенерувати список цілих додатних та від’ємних чисел, кількість елементів якого ввести з клавіатури. Знайти суму елементів, розташованих до першого від’ємного елементу та добуток елементів, розташованих після останнього від’ємного елемента. Якщо від’ємних елементів немає, то видати відповідне повідомлення.
63. Дана дійсна квадратна матриця порядку 2n. Отримати нову матрицю, переставляючи її блоки розміру n × n хрест-навхрест
64. **Скласти програму для обробки одновимірного масиву та матриці. Передбачити меню вибору задач**
65. Ввести з клавіатури список чисел. Видалити зі списку всі числа, які повторюються більше двох разів. Надрукувати вхідний список і список після видалення елементів.
66. Задати цілі додатні числа M, N, число D і набір з M чисел. Сформувати матрицю розміру M × N, у якій перший стовпець збігається з вихідним набором чисел, а елементи кожного наступного стовпця дорівнюють сумі відповідного елемента попереднього стовпчика і числа D (в результаті кожен рядок матриці буде містити елементи арифметичної прогресії).
67. **Скласти програму для обробки одновимірного масиву та матриці. Передбачити меню вибору задач**
68. Згенерувати список цілих чисел в діапазоні від -50 до +50, кількість елементів задати з клавіатури. Із отриманого списку скопіювати в інший список елементи, значення яких знаходяться в діапазоні від -5 до 5 включно і підрахувати їх кількість. Надрукувати згенерований і побудований списки.
69. Задати матрицю цілих чисел розміру M × N. Знайти номер першого з її рядків, що містять однакову кількість додатних і від’ємних елементів (нульові елементи матриці не враховуються). Якщо таких рядків немає, то вивести відповідне повідомлення.
70. **Скласти програму для обробки одновимірного масиву та матриці. Передбачити меню вибору задач**
71. Дан список цілих чисел. Розмір списку і його елементи вводяться користувачем. Упорядкувати за зростанням елементи списку, що розташовані до максимального елемента, і за спаданням – ті елементи, що розташовані після максимального елементу. Надрукувати список до та після сортування.
72. Задати квадратну матрицю порядку M. Обнулити елементи матриці, що лежать одночасно нижче головної діагоналі (включаючи цю діагональ) і вище побічної діагоналі (також включаючи цю діагональ). Умовний оператор не використовувати. Надрукувати вхідну та отриману матрицю
73. **Скласти програму для обробки одновимірного масиву та матриці. Передбачити меню вибору задач**
74. Дан список дійсних чисел. Розмір списку і значення його елементів вводяться користувачем. Знайти суму всіх локальних мінімумів списку. Локальний мінімум – це елемент списку, який менший за значенням двох сусідніх елементів даного списку (виключаючи крайні елементи). Надрукувати вхідний список, всі його локальні мінімуми та їх індекси.
75. Задати матрицю розміру M × N. Упорядкувати її рядки так, щоб їх перші елементи утворювали зростаючу послідовність. Надрукувати вхідну та упорядковану матриці.

## Список літератури

1. <https://github.com/tkovalyuk/Basics-of-programming>
2. **МакГрат М**. Программирование на Python для начинающих. –Москва: Эксмо. – 192 с.
3. **Лутц М.** Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
4. **Лутц М.** Программирование на Python, том I, 2. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
5. **Васильев А. Н.** Python на примерах. Практический курс ·по программированию. - СПб.:Наука и Техника, 2016. - 432 с.
6. **Доусон М.** Программируем на Python. - СПб.: Питер, 2014. - 416 с.
7. **Рейтц К., Шлюссер Т.** Автостопом по Python. — СПб.: Питер, 2017. — 336 с.

# Лабораторна робота 8 Обробка словників та множин в Python

**Мета роботи:** Ознайомитись з технологією обробки словників та множин в Python. Побудувати блок-схеми алгоритмів, розробити програми з використанням операцій та методів обробки словників та множин у вигляді одновимірних масивів та матриць.

## Рейтинг лабораторної роботи №7

|  |  |  |  |
| --- | --- | --- | --- |
| № п.п | Вид діяльності студента | Рейтинговий бал | Deadline |
| 1 | Написання коду | 3 | Листопад |
| 2 | Захист роботи | 1 |
| 3 | Звіт з роботи | 1 |
| 4 | Якість роботи | 1 |
| Разом за роботу | | 6 |  |

## Теоретичний матеріал

Словник у Python – це невпорядкована колекція об'єктів довільних типів з доступом по ключу.

Ключем словника може бути об'єкт незмінюваного типу (число, рядок, кортеж), за допомогою якого можна однозначно звернутися до елемента словника.

Словники називають асоціативними масивами або хеш-мапами.

Словники у Python належать до змінюваних (mutable) типів.

Значення словника зберігаються в невідсортованому порядку, більш того, ключі можуть зберігатися не в тому порядку, в якому вони додаються до колекції.

### Створення словників

1. **За допомогою літерала.**

Пари ключ та значення елементів словника записуються через двокрапку. Різні пари розділяються комою, а вся послідовність записується у фігурних дужках.

1. **За допомогою перетворення у словник іншої колекції, використовуючи ключове слово dict.**

d = dict(collection)

1. **За допомогою методу fromkeys**

d = dict.fromkeys(collection, initial\_value)

### Доступ до елементів словника

Присвоєння по існуючому ключу перезаписує значення елемента, що відповідає цьому ключу.

Присвоєння по новому ключу розширяє словник новою парою ключ-значення

### Операції над словниками

1. Перевірка приналежності ключа або значення до словника

|  |  |
| --- | --- |
| **Операція** | **Опис** |
| key **in** d | Повертає **True**, якщо ключ key входить до d |
| key **not in** d | Повертає **True**, якщо ключ key не входить до d |
| v **in** d.values() | Повертає **True**, якщо значення v входить до d |
| v **not in** d.values() | Повертає **True**, якщо значення v не входить до d |

2. Вибір елементів чи колекцій словника

|  |  |
| --- | --- |
| **Операція** | **Опис** |
| d.get(k) | Повертає значення **ключа** k або **None**, якщо ключа k немає у словнику. |
| d.get(k,v) | Повертає значення **ключа** k або v, якщо ключа k немає у словнику. |
| d.items() | Повертає колекцію всіх пар ключ-значення в словнику d. |
| d.keys() | Повертає колекцію всіх **ключів** словника d. |
| d.values() | Повертає колекцію всіх **значень** в словнику d. |

3. Аналіз словника

|  |  |
| --- | --- |
| **Операція** | **Опис** |
| len(d) | Довжина d (кількість пар ключ-значення) |
| min(d) | Найменший ключ словника d (якщо ключі можна порівнювати) |
| max(d) | Найбільший ключ словника d (якщо ключі можна порівнювати) |

4. Модифікація словника

|  |  |
| --- | --- |
| **Операція** | **Опис** |
| d.clear() | Видаляє всі елементи зі словника d |
| d.copy() | Повертає копію словника d |
| d.pop(k) | Повертає значення ключа k і видаляє зі словника елемент з ключем k (якщо ключа k немає у словнику, то видає помилку) |
| d.pop(k,v) | Повертає значення ключа k і видаляє зі словника елемент з ключем k. Якщо ключа k немає у словнику, то повертає значення v. |
| d.popitem() | Повертає і видаляє довільну пару ключ-значення зі словника d (якщо словник d порожній, то генерує помилку) |
| d.update(d1) | Додає в словник d пари ключ-значення з колекції d1, які  відсутні в словнику d, а для кожного ключа, який вже присутній в словнику d, виконує заміну відповідним значенням з d1 (колекція d1 може бути як словником, так і будь-якою колекцією, що містить пари ключ-значення). |

### Обхід словників

1. Обхід словника за ключами

for key in d:

process\_iteration

2. Обхід словника за значеннями

for val in d.values():

process\_iteration

3. Обхід словника за парами ключ-значення

for key, val in d.items():

process\_iteration

### Приклад обробки словників

***Задача 1***. Слова у рядку розділяються одним або декількома пропусками. Визначити кількість входжень кожного слова до рядка. Визначити слово, яке входить до тексту найбільшу кількість разів.

***Розв’язок***. Для розв'язання задачі побудуємо на базі рядка словник, у якому ключами будуть унікальні слова, а значеннями кількість входжень слова до рядка.

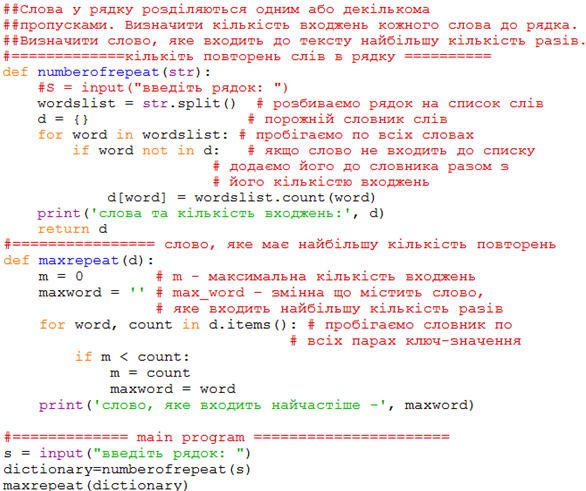
****

Рис.1 Код прикладу обробки словників.

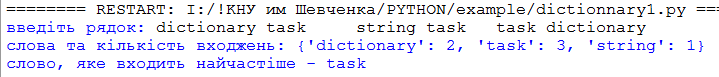


Рис. 2. Результати роботи програми обробки словників

## Приклад програми до лабораторної роботи



Рис. 3. Код функцій до програми прикладу лабораторного завдання

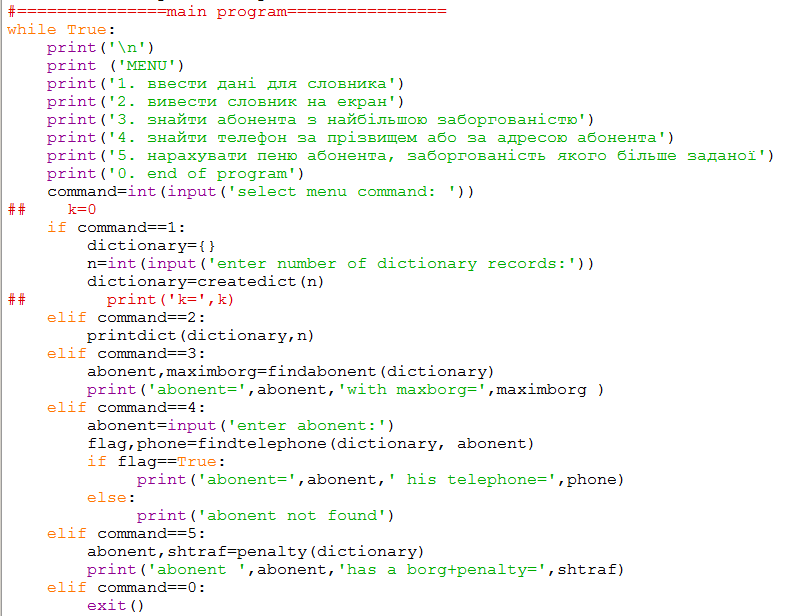


Рис. 4.Код основної програми

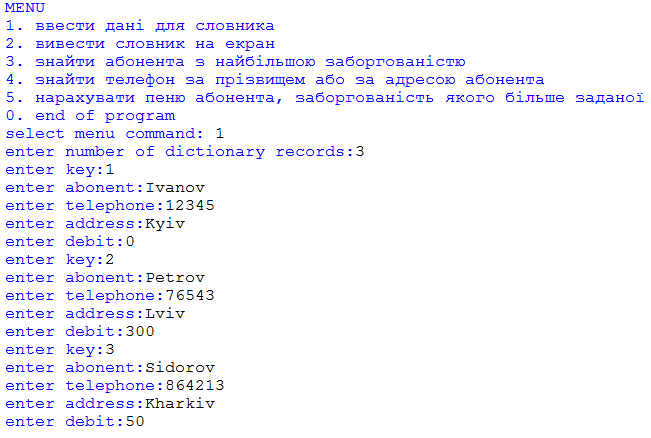


Рис.5. Результат роботи функції введення даних з клавіатури

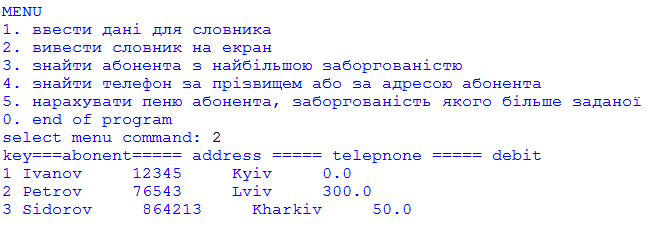


Рис.6. Результат роботи функції виведення словника

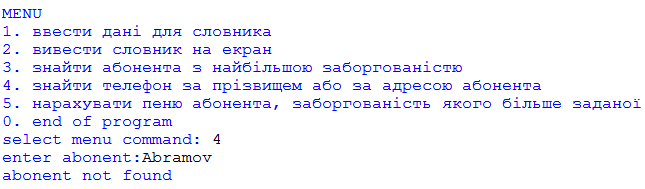


Рис.7. Результат роботи функції пошуку в словнику номера телефона за прізвищем абонента

## Завдання для самостійної роботи за варіантами

Кожне завдання складається з двох задач: перша задача спрямована на обробку словників, друга задача – обробка множин. Завдання має бути реалізовано у вигляді одної програми, яка має меню вибору задач. Код завдання повинний складатися з функцій, наприклад, введення вхідних даних, виведення результатів, розрахунок результату тощо.

**Під час виконання лабораторної роботи не використовувати бібліотеки Collections та Numpy**.

### Варіанти завдань

1. **Скласти програму для обробки словнику та множини. Передбачити меню вибору задач.**
2. Створити словник, який має декілька ключів та значень. Кожному ключу відповідає масив таких елементів: факультет, прізвище студента, середній бал. Надрукувати словник. Реалізувати запити, визначивши: 1) прізвища студентів, які мають найменший середній бал; 2) факультет, який має найбільшу кількість відмінників; 3) вилучити студента, середній бал якого менше 2.5
3. Задано множину цілих чисел від 1 до 50. Скласти програму, яка визначає, скільки з них є числами Фібоначчі і скільки чисел, в запису яких перша значуща цифра дорівнює 1 або 2.
4. **Скласти програму для обробки словнику та множини. Передбачити меню вибору задач.**
5. Створити словник, який має декілька ключів та значень. Кожному ключу відповідає масив таких елементів: назва фірми, назва вакантної посади, кількість вакантних місць, заробітна платня. Надрукувати словник. Реалізувати запити: 1) визначити фірми з найбільшою кількістю вакансій на задану посаду; 2) вивести назву вакантної посади з найбільшою зарплатою; 3) доповнити словник даними про нову фірму з вакансіями.
6. Задано текст з цифр і літер латинського алфавіту. Скласти програму, яка визначає, яких літер – голосних {*a, e, i, o, u, y*} або приголосних більше в цьому тексті.
7. **Скласти програму для обробки словнику та множини. Передбачити меню вибору задач.**
8. Створити словник, який має декілька ключів та значень. Кожному ключу відповідає масив таких елементів: марка автомобіля, прізвище власника, номерний знак, строк служби. Надрукувати словник. Реалізувати запити: 1) визначити прізвище власника за введеним з клавіатури номерним знаком; 2) вивести марки автомобілів із заданим з клавіатури строком служби; 3) вилучити зі словника дані про автомобілів заданої марки.
9. Задано множину символів А i символ ‘х’. Скласти програму, яка формує множину В з множини А за таким правилом: а) додавання елемента ‘х’, якщо він відсутній в А; б) видалення елементу ‘х’, якщо він присутній в А.
10. **Скласти програму для обробки словнику та множини. Передбачити меню вибору задач.**
11. Створити словник, який має декілька ключів та значень. Кожному ключу відповідає масив таких елементів: назва фірми, найменування товару, що виробляє фірма, вартість товару, термін придатності товару. Надрукувати словник. Реалізувати запити: 1) визначити фірми, що виробляє товар, назва якого вводиться з клавіатури; 2) визначити назву товару, вартість якого не перевищує заданого з клавіатури значення; 3) впорядкувати словник за умови зростання термінів придатності товару.
12. Задано текст з латинських літер. Скласти програму, яка визначає і виводить на екран такі множини: а) символи – перші входження в текст, зберігаючи при цьому їх вихідний взаємний порядок; б) всі літери, які входять в текст не менше двох разів; в) всі літери, які входять в текст по одному разу.
13. **Скласти програму для обробки словнику та множини. Передбачити меню вибору задач.**
14. Створити словник, який має декілька ключів та значень. Кожному ключу відповідає масив таких елементів: номер авіаційного рейсу, дата польоту, кількість вільних місць, вартість квитка. Надрукувати словник. Реалізувати запити: 1) вивести список номерів авіаційних рейсів, на які є вільні місця; 2) вивести номери авіаційних рейсів з мінімальною вартістю квитків; 3) упорядкувати словник за умови зростання вартості квитків.
15. Задано текст з латинських літер. Скласти програму, яка визначає і виводить на екран в алфавітному порядку по одному разу всі голосні літери латинського алфавіту (множина {a, e, i, o, u, y}), які входять в цей текст. Текст та елементи множини задано в одному реєстрі (нижньому або верхньому).
16. **Скласти програму для обробки словнику та множини. Передбачити меню вибору задач.**
17. Створити словник, який має декілька ключів та значень. Кожному ключу відповідає масив таких елементів: назва продукту, ціна продукту, прибуток споживача. Надрукувати словник. Реалізувати запити, визначивши: 1) назви продуктів, які може придбати споживач при заданому прибутку; 2) сумарну вартість продуктів, які внесені у словник; 3) назву продукту з максимальною вартістю.
18. Існує множину день\_тижня={пон, вівт, сер, чет, п‘ят, суб, нед}. Скласти програму, яка формує множину, що включає в себе: а) назви днів тижня визначеного місяця; б) назви робочих днів тижня визначеного місяця.
19. **Скласти програму для обробки словнику та множини. Передбачити меню вибору задач**
20. Створити словник, який має декілька ключів та значень. Кожному ключу відповідає масив таких елементів: назва книжки, прізвище автора, рік видання. Надрукувати словник. Реалізувати запити: 1) вивести авторів, які у заданому з клавіатури році видали книжку; 2) визначити назву книжок автора, прізвище якого задане з клавіатури; 3) упорядкувати словник за алфавітом прізвищ авторів
21. Ввести рядок символів. Визначити множину голосних латинських літер {*а, е, i, о, u, y*} та множину усіх цифр, що входять в цей рядок.
22. **Скласти програму для обробки словнику та множини. Передбачити меню вибору задач**
23. Створити словник, який має декілька ключів та значень. Кожному ключу відповідає масив таких елементів: прізвище викладача, назва дисципліни, кількість навчальних годин з дисципліни, кількість студентів на потоці, які вивчають дисципліну. Надрукувати словник. Реалізувати такі запити: 1) визначити викладачів, які викладають задану з клавіатури дисципліну; 2) визначити дисципліну, яку вивчає найбільша кількість студентів; 3) визначити сумарну кількість навчальних годин з усіх дисциплін, які записані у словнику.
24. Задано текст з латинських літер. Символи цього тексту формують множину. Скласти програму, яка визначає і виводить на екран: а) перші входження літер в текст, зберігаючи їх вихідний взаємний порядок; б) всі літери, які входять в текст не менше двох разів; в) всі літери, які входять в текст один раз.
25. **Скласти програму для обробки словнику та множини. Передбачити меню вибору задач**
26. Створити словник, який має декілька ключів та значень. Кожному ключу відповідає масив таких елементів: назва країни, площа країни, назва столиці. Надрукувати словник. Реалізувати такі запити: 1). Визначити країну з найбільшою площею; 2) Визначити столицю країни, назва якої вводиться з клавіатури; 3). Визначити країни, назва столиць яких починається з літери ‘А’.
27. Задано текст з цифр і латинських літер. Скласти програму, що визначає, яких літер – голосних {*а, е, i, о, і, y*} або приголосних – більше в цьому тексті.
28. **Скласти програму для обробки словнику та множини. Передбачити меню вибору задач**
29. Створити словник, який має декілька ключів та значень. Кожному ключу відповідає масив таких елементів: дата, температура повітря, кількість опадів. Надрукувати словник. Реалізувати такі запити: 1) визначити кількість опадів, що випала у вигляді дощу та у вигляді снігу; 2) визначити дату з найбільшою температурою; 3) Визначити період, коли не було опадів.
30. Маємо деякий набір продуктових товарів *n*, асортимент магазину – це множина товарів з цього набору. Компанія володіє m магазинами. Скласти програму, яка сформує такі множини: множину А продуктів, які є в усіх магазинах; В – множину продуктів, кожен з яких є хоча б в одному магазині; С – множину продуктів, яких немає в жодному магазині.
31. **Скласти програму для обробки словнику та множини. Передбачити меню вибору задач**
32. Створити словник, який має декілька ключів та значень. Кожному ключу відповідає масив таких елементів: назва спортивної команди, кількість очок, які набрала команда, призовий фонд. Надрукувати словник. Реалізувати такі запити: 1) визначити команду, яка стала чемпіоном; 2) видалити із словника команду, яка набрала найменшу кількість очок; 3) визначити суму призового фонду, який має розподілитися між призерами турніру: 50% за перше місце, 30% за друге, 20% за третє місця.
33. З діапазону цілих чисел від m до n виділити. 1) множину цілих чисел, що діляться без остачі на K, або на L (K, L- прості). 2) множину чисел, що діляться на K \* L без залостачі.
34. **Скласти програму для обробки словнику та множини. Передбачити меню вибору задач**
35. Створити словник, який має декілька ключів та значень. Кожному ключу відповідає масив таких елементів: назва авіакомпанії, прізвище пасажира, вага багажу, ліміт ваги багажу без доплати. Надрукувати словник. Реалізувати такі запити: 1) визначити пасажира, вага багажу якого перевищує ліміт; 2) визначити вартість доплати за перевищення ліміту ваги багажу, задавши вартість доплати за кожний кілограм з клавіатури; 3) визначити авіакомпанії з найменшим лімітом ваги багажу.
36. Відомі марки машин, виготовлених в країні і такі, що імпортуються в N країн за кордон. Визначити для кожної з марок машин, які з них були доставлені в усі N країн, доставлені до деяких з N країн, недоставлені в жодну з N країн.
37. **Скласти програму для обробки словнику та множини. Передбачити меню вибору задач**
38. Створити словник, який має декілька ключів та значень. Кожному ключу відповідає масив таких елементів: назва компанії, посада працівника, оклад. Надрукувати словник. Реалізувати такі запити: 1) визначити компанію з найвищим окладом її працівників; 2) визначити посаду з найменшим окладом; 3) Визначити кількість посад з окладом, значення якого ведене з клавіатури.
39. З множини латинських букв виділити множину, що складається з символів, які не входять в множину символів вашого імені. Вивести цю множину на екран.
40. **Скласти програму для обробки словнику та множини. Передбачити меню вибору задач**
41. Створити словник, який має декілька ключів та значень. Кожному ключу відповідає масив таких елементів: прізвище пасажира, кількість його речей, загальна вага його багажу. Надрукувати словник. Реалізувати такі запити: 1) визначити кількість пасажирів, які мають більше двох речей; б) визначити, чи є хоч один пасажир, багаж якого складається з однієї речі вагою менше 25 кг; в) визначити кількість пасажирів, кількість речей яких перевершує середню кількість речей всіх пасажирів.
42. Задати декілька множин цілих чисел. Знайти об’єднання усіх множин, що перетиаяються.

1. **Скласти програму для обробки словнику та множини. Передбачити меню вибору задач**
2. Створити словник, який має декілька ключів та значень. Кожному ключу відповідає масив таких елементів: прізвище учня класу, його зріст. Надрукувати словник. Реалізувати такі запити: 1) впорядкувати словник за спаданням зросту учнів; 2) додати з клавіатури до словника один запис про учня, зріст якого не співпадає із зростом жодного з учнів класу, перевищує зріст самого низького учня і менше зросту найвищого; 3) визначити прізвища усіх учнів, зріст яких менше зросту нового учня.
3. Кожна сім'я, яка живе в будинку, передплачує газету, або журнал, або і те, і інше. M сімей виписують газету, N - журнал, K - і журнал, і газету. Скласти програму, визначення кількості сімей, які живуть в будинку?
4. **Скласти програму для обробки словнику та множини. Передбачити меню вибору задач**
5. Створити словник, який має декілька ключів та значень. Кожному ключу відповідає масив таких елементів: прізвище студента, курс, три оцінки в сесію. Надрукувати словник. Реалізувати такі запити: 1) визначити, чи є студенти з однаковими прізвищами; 2) визначити середній бал кожного студента; 3) визначити кількість студентів на кожному курсі, які мають принаймні одну двійку.
6. У неділю 19 учнів класу побували в планетарії, 10 - в цирку і 6 – на стадіоні. Планетарій і цирк відвідали 5 чоловік, планетарій і стадіон - 3, цирк і стадіон - 1. Скласти програму, яка визначає, скільки учнів в класі, якщо ніхто не встиг відвідати всі три місця, а 3 учні хворіли і в вихідні були вдома.
7. **Скласти програму для обробки словнику та множини. Передбачити меню вибору задач**
8. Створити словник, який має декілька ключів та значень. Кожному ключу відповідає масив таких елементів: назва проекту, вартість фінансування проекту, кількість виконавців проекту. Надрукувати словник. Реалізувати такі запити: 1) визначити найдорожчий проект; 2) враховуючи, що 60% фінансування виділяється на зарплатню працівникам, визначити середню зарплату працівників кожного проекту; 3) знайти проект із заданим з клавіатури фінансуванням.
9. Ввести розмірність та згенерувати матрицю. У кожному стовпці і кожному рядку матриці мають міститися строго по одному нульового елементу. Перестановкою рядків домогтися розташування всіх нулів на головній діагоналі матриці.
10. **Скласти програму для обробки словнику та множини. Передбачити меню вибору задач**
11. Створити словник, який має декілька ключів та значень. Кожному ключу відповідає масив таких елементів: прізвище студента, курс, три оцінки в сесію. Надрукувати словник. Реалізувати такі запити: 1) надрукувати список студентів, які не мають оцінок нижче чотирьох; 2) надрукувати список студентів, які мають усі відмінні оцінки; 3) визначити, чи є студенти з однаковим середнім балом.
12. На уроці літератури вчитель вирішив дізнатися, хто з 40 учнів читав книги А, В і С. Результати опитування виглядають так: А - 25 уч., В - 22 уч., З - 22 уч .; одну з книг А або В - 33 уч., А або С - 32 уч., В або С - 31 уч, все три книги прочитали 10 учнів. Скласти програму, яка визначає, скільки учнів прочитали по одній книзі? Скільки учнів прочитали рівно 2 книги? Скільки учнів не прочитали жодної книги?
13. **Скласти програму для обробки словнику та множини. Передбачити меню вибору задач**
14. Створити словник, який має декілька ключів та значень. Кожному ключу відповідає масив таких елементів: місто, назва фірми, назва продукту, ціна продукту. Надрукувати словник. Реалізувати такі запити: 1) визначити місцезнаходження фірми, що виробляє заданий з клавіатури продукт; 2) визначити сумарну вартість продуктів, що виробляються у даному місті; 3) упорядкувати список продуктів за ціною для даного міста.
15. Згенерувати матрицю розміром N \* N, що містить додатні та від’ємні числа. Здійснити пошук такої підматриці, сума елементів якої є максимальною.отримує
16. **Скласти програму для обробки словнику та множини. Передбачити меню вибору задач**
17. Створити словник, який має декілька ключів та значень. Кожному ключу відповідає масив таких елементів: назва державного підприємства, його вартість, кількість працівників, їх середня зарплатня. Надрукувати словник. Реалізувати такі запити: 1) визначити, скільки коштів отримує держава, якщо підприємство буде продано за ціною, що на 50% перевищує його вартість; 2) визначити середню зарплатню працівників після приватизації підприємства, якщо володар хоче звільнити 20% працівників; 3) надрукувати назви підприємств, вартість яких менше заданої з клавіатури.
18. Задана множина персональних комп’ютерів, якими може бути забезпечений коледж. Відома множина персональних комп’ютерів, наявних в кожному коледжі. Потрібно побудувати і роздрукувати множину, що включає в себе персональні комп’ютери, якими забезпечені всі технікуми; які має хоча б один технікум
19. **Скласти програму для обробки словнику та множини. Передбачити меню вибору задач**
20. Створити словник, який має декілька ключів та значень. Кожному ключу відповідає масив таких елементів: прізвище студента, форма навчання (контракт або бюджет), середній бал, вартість контракту. Надрукувати словник. Реалізувати такі запити: 1) визначити сумарну вартість контрактів по усіх студентах; 2) визначити студентів-відмінників, середній бал яких дорівнює 5, і змінити їх форму навчання на бюджетну; 3) видалити студентів, середній бал яких менше 2.5
21. Задана множина персональних комп’ютерів, якими може бути забезпечений коледж. Відома множина персональних комп’ютерів, наявних в кожному коледжі. Потрібно побудувати і роздрукувати множину, що включає в себе персональні комп’ютери, яких немає в жодному коледжі. які є хоча б в 3-х коледжах.
22. **Скласти програму для обробки словнику та множини. Передбачити меню вибору задач**
23. Задано англо-український словник. Для кожного англійського слова у цьому словнику може бути декілька перекладів – синонімів. Написати програму, що по заданому англо-українському словнику будує україно-англійський словник.
24. З введеної множини символів, що включає букви та цифри, сформувати і надрукувати множину, що виключає всі цифри.
25. **Скласти програму для обробки словнику та множини. Передбачити меню вибору задач**
26. Створити словник, який має декілька ключів та значень. Кожному ключу відповідає масив таких елементів: назва компанії, вартість акцій компанії, дата. Надрукувати словник. Реалізувати такі запити: 1) визначити дату, на яку вартість акцій заданої компанії була максимальна; 2) середню вартість акцій кожної компанії в заданий інтервал дат; 3) спрогнозувати вартість акцій заданої компанії.
27. Сформувати множину цифр, що входять до запису заданого з клавіатури числа. Підрахувати кількість елементів отриманої множини.
28. **Скласти програму для обробки словнику та множини. Передбачити меню вибору задач**
29. Створити словник, який має декілька ключів та значень. Кожному ключу відповідає масив таких елементів: назва підприємства, плановий обсяг випуску товарів, фактичний обсяг випуску товарів. Надрукувати словник. Реалізувати такі запити: 1) визначити відсоток виконання плану кожним підприємством; 2) кількість підприємств, які недовиконали план на 10% та більше; 3) підприємство, яке має найменший фактичний обсяг випуску продукції.
30. Створити множину, що містить натуральні числа з заданого користувачем діапазону. Сформувати дві множини, перша з яких містить усі прості числа з даної множини, а друга – всі складові.
31. **Скласти програму для обробки словнику та множини. Передбачити меню вибору задач**
32. Задати список країн і для кожної країни список міст. Для кожного міста, що вводиться з клавіатури визначити країну, в якій це місто знаходиться. У разі відсутності міста в словнику країн вивести відповідне повідомлення.
33. Ввести текст з малих латинських букв, який закінчується крапкою. Побудувати множину, що складається з усіх букв, що входять в текст по одному разу і надрукувати її. Видалити з цієї множини усі голосні.

## Список літератури

1. <https://github.com/tkovalyuk/Basics-of-programming>
2. **МакГрат М**. Программирование на Python для начинающих. –Москва: Эксмо. – 192 с.
3. **Лутц М.** Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
4. **Лутц М.** Программирование на Python, том I, 2. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
5. **Васильев А. Н.** Python на примерах. Практический курс ·по программированию. - СПб.:Наука и Техника, 2016. - 432 с.
6. **Доусон М.** Программируем на Python. - СПб.: Питер, 2014. - 416 с.
7. **Рейтц К., Шлюссер Т.** Автостопом по Python. — СПб.: Питер, 2017. — 336 с.

# Лабораторна робота 9 Обробка текстових та бінарних файлів в Python

**Мета роботи:** Ознайомитись з технологією обробки текстових та бінарних файлів в Python. Побудувати блок-схеми алгоритмів, розробити програми з використанням операцій та методів обробки текстових та бінарних файлів.

## Рейтинг лабораторної роботи №9

|  |  |  |  |
| --- | --- | --- | --- |
| № п.п | Вид діяльності студента | Рейтинговий бал | Deadline |
| 1 | Написання коду | 4 | Листопад |
| 2 | Захист роботи | 1 |
| 3 | Звіт з роботи | 1 |
| 4 | Якість роботи (за наявності) | 1 |
| Разом за роботу | | 7 |  |

## Теоретичний матеріал

**Файл** (англ. file – папка) – це іменований блок даних (послідовність байтів), який зберігається на носії інформації.

Файл є найменшою одиницею збереження інформації на носії.

Файл має такі ознаки:

* + фіксоване ім'я (назва файла) – послідовність символів, що однозначно характеризує файл;
  + певне логічне зображення (що визначається типом інформації, що міститься у файлі) і відповідні йому операції читання/запису;
  + розмір файла (характеризується розміром даних, що в ньому містяться).

### Типи файлів

При роботі з файлами розрізняються двійкові (або бінарні) і текстові файли.

Двійкові файли відкриваються як послідовність байтів. При цьому відповідальність за коректну робота з даними повністю покладається на програму, що створює та використовує цей файл.

Текстові файли відкриваються як послідовність рядків (символів), що міститься у файлі. При цьому фізичний рядок у файлі відповідає рядковому літералу у програмі.

Створювати, змінювати та опрацьовувати бінарні файли можна лише за допомогою спеціальних програм.

З текстовими файлами можна працювати за допомогою будь-якого текстового редактора.

### Відкриття та закриття файлів

Стандартний спосіб створення файлового об'єкта – функція **open()**

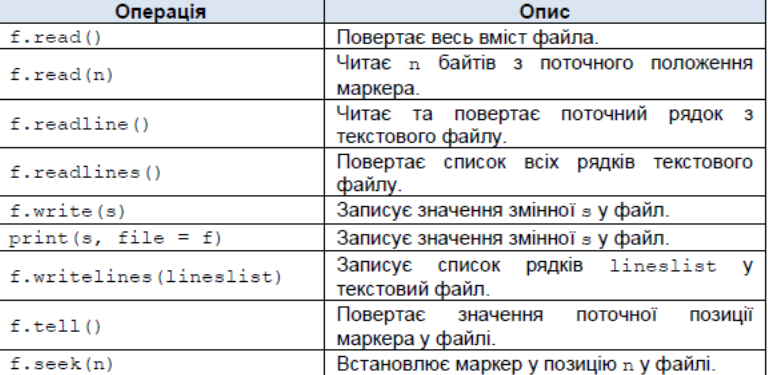
**f = open(file\_name, mode),**

де f – ім'я файлової змінної, file\_name – ім'я файла, що відкривається, mode – режим роботи з файлом (код режиму в табл.):



Робота з файлом відбувається через файлову змінну, що зберігає свої дані в оперативній пам'яті. Для того, щоб зберегти результат роботи з файлом на носій інформації, файл необхідно **закрити**. Операція закриття файлу повідомляє операційній системі, що файл розблокований і може використовуватися (для зміни) іншими програмами. Для того, щоб закрити файл використовується метод **f.close().**

### Читання та запис текстових файлів



**Приклад 1**. Створити текстовий файл, що містить послідовність чисел Фібоначчі, яка обмежена заданим числом.

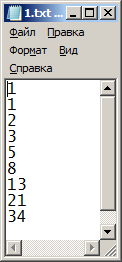
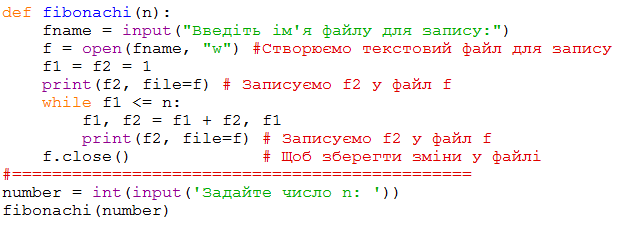
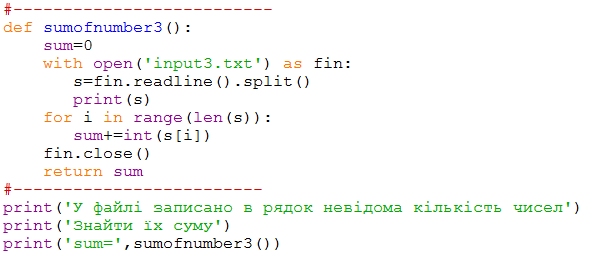


Рис.1 Код і результати роботи програми у вигляді текстового файлу

**Приклад 2.** У текстоволму файлі в рядок записана невідома кількість чисел. Знайти їх суму



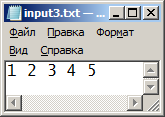
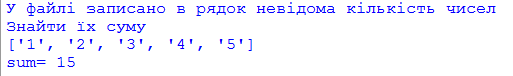


Рис. Код і результати роботи програми

### Робота з бінарними файлами

В Python існує таке поняття, як консервування даних. Воно дозволяє зберігати в файлах не просто набір символів, а більш складні структури, наприклад, списки або словники. Для здійснення всіх цих операцій, в Python передбачені два модулі:

* модуль pickle
* модуль shelve.

Відповідно, перший модуль дозволяє консервувати структури даних, а другий модуль - здійснити зберігання об'єктів в рідному для Python форматі, застосувавши полицю (shelve). За своїм інтерфейсу полиця не відрізняється від словника, і забезпечує доступ до об'єктів, що зберігаються «на полицях».

#### Модуль pickle

Цей модуль надає два методи:

* **dump (obj, file):** записує об'єкт obj в бінарний файл file
* **load (file):** зчитує дані з бінарного файлу в об'єкт

При відкритті бінарного файлу на читання або запис треба застосовувати режим "b" в доповнення до режиму запису ("w") або читання ("r").

#### Модуль shelve

Для роботи з бінарними файлами в Python може застосовуватися модуль shelve. Він зберігає об'єкти в файлі з певним ключем. Потім по цьому ключу може отримати раніше збережений об'єкт з файлу.

Процес роботи з даними через модуль shelve нагадує роботу зі словниками, які також використовують ключі для збереження і вилучення об'єктів. Ключ задається у вигляді рядка і визначається інтерфейс доступу, подібний до словників. Для серіалізації об'єкта використовується можливості модуля pickle, а щоб записати отриманий рядок по ключу в файл, застосовується модуль anydbm. Щоб відкрити файл з базою об'єктів, використовується функція open().

**open(шлях\_до\_файлу[, mode="wb"**

#### Режими доступу до бінарних файлів

|  |  |
| --- | --- |
| **Режим** | **Опис** |
| rb | Читання з бінарного файлу. Якщо файл не існує, то Python повідомить про помилку |
| wb | Запис в бінарний файл. Якщо файл існує, його вміст буде замінено. Якщо файл не існує, він буде створений |
| аb | Доповнення бінарного файлу. Якщо файл існує, нові дані будуть дописані в кінець. Якщо файл не існує, він буде створений |
| rb+ | Читання і запис в бінарний файл. Якщо файл не існує, то Python повідомить про помилку |
| wb+ | Запис і читання з бінарного файлу. Якщо файл існує, його вміст буде замінено. Якщо файл не існує, то він буде створений |
| аb+ | До запис і читання з бінарного файлу. Якщо файл існує, нові дані будуть дописані в кінець. Якщо файл не існує, він буде створений |

Для закриття підключення до файлу викликається метод **close ()**

**Приклад 3.** Створити бінарний файл із заданих списків. Завантажити з бінарного файлу дані в оперативну пам’ять.

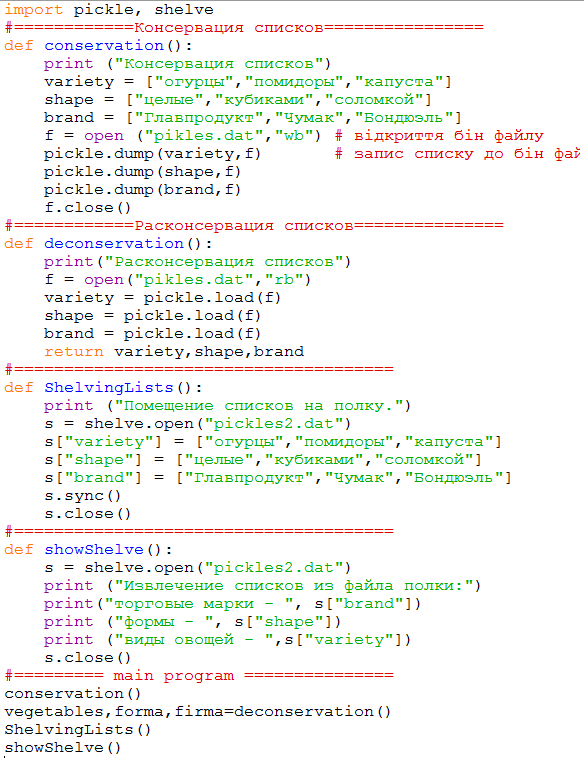


Рис. 3. Приклад використання модулів pikle та shelve

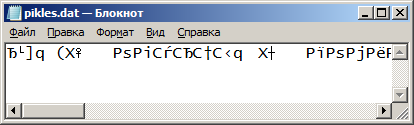
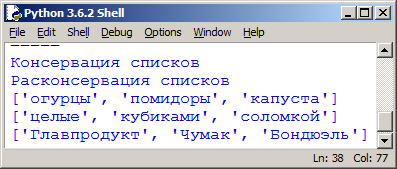


Рис. 4. Результати роботи програми

## Завдання для самостійної роботи за варіантами

Кожне завдання складається з двох задач: перша задача спрямована на обробку текстових файлів, друга задача – обробка бінарних файлів. Завдання має бути реалізовано за процедурно-орієнтованою технологією у вигляді одної або двох програм. Програми повинні мати меню вибору задач у випадку одної програми. Код кожної завдання повинний мати своє внутрішнє меню вибору команд. Код кожного завдання повинний складатися з функцій, наприклад, введення вхідних даних, виведення результатів, розрахунок результату тощо. Основна програма викликає функції кожного завдання в довільному порядку в залежності від вибору команд меню користувачем.

### Варіанти завдань

1. **Скласти програму для обробки текстового та бінарного файлів. Передбачити меню вибору задач.**
2. Створити текстовий файл, кожний рядок якого містить різні слова. Знайти найкоротше слово в кожному рядку файлу, переписати їх в новий текстовий файл, записати останнім рядком файлу кількість слів у файлі
3. У програму лабораторної роботи №8 додати нові функції обробки бінарних файлів:

* створений у лабораторній роботі №8 словник записати до бінарного файлу;
* надрукувати словник у вигляді таблиці, зчитавши дані з бінарного файлу;
* розробити функції, які реалізують операції додавання у бінарний файл нових даних, видалення даних за запитом користувача, заміни одних значень іншими на вимогу користувача;
* результати запитів, які реалізовані в лабораторній роботі №8, записати до бінарного файлу і вивести на екран;

1. **Скласти програму для обробки текстового та бінарного файлів. Передбачити меню вибору задач.**
2. Створити текстовий файл, рядок якого містить цифри i символи. В кожному рядку визначити найбільшу послідовність цифр, що йдуть поспіль. Значення довжин цифрових послідовностей записати першими символами рядків вхідного файлу.
3. У програму лабораторної роботи №8 додати нові функції обробки бінарних файлів:

* створений у лабораторній роботі №8 словник записати до бінарного файлу;
* надрукувати словник у вигляді таблиці, зчитавши дані з бінарного файлу;
* розробити функції, які реалізують операції додавання у бінарний файл нових даних, видалення даних за запитом користувача, заміни одних значень іншими на вимогу користувача;
* результати запитів, які реалізовані в лабораторній роботі №8, записати до бінарного файлу і вивести на екран.

1. **Скласти програму для обробки текстового та бінарного файлів. Передбачити меню вибору задач.**
2. Створити текстовий файл, задавши рядки з клавіатури. Визначити в кожному парному рядку слово найбільшої довжини і записати це слово в кожному непарному рядку текстового файлу
3. У програму лабораторної роботи №8 додати нові функції обробки бінарних файлів:

* створений у лабораторній роботі №8 словник записати до бінарного файлу;
* надрукувати словник у вигляді таблиці, зчитавши дані з бінарного файлу;
* розробити функції, які реалізують операції додавання у бінарний файл нових даних, видалення даних за запитом користувача, заміни одних значень іншими на вимогу користувача;
* результати запитів, які реалізовані в лабораторній роботі №8, записати до бінарного файлу і вивести на екран.

1. **Скласти програму для обробки текстового та бінарного файлів. Передбачити меню вибору задач.**
2. Увести з клавіатури рядки і записати їх у текстовий файл. У кожному непарному рядку визначити слово, що має найбільшу кількість голосних. Дописати знайдені слова в кінець кожного рядка файлу
3. У програму лабораторної роботи №8 додати нові функції обробки бінарних файлів:

* створений у лабораторній роботі №8 словник записати до бінарного файлу;
* надрукувати словник у вигляді таблиці, зчитавши дані з бінарного файлу;
* розробити функції, які реалізують операції додавання у бінарний файл нових даних, видалення даних за запитом користувача, заміни одних значень іншими на вимогу користувача;
* результати запитів, які реалізовані в лабораторній роботі №8, записати до бінарного файлу і вивести на екран

1. **Скласти програму для обробки текстового та бінарного файлів. Передбачити меню вибору задач.**
2. Увести з клавіатури декілька рядків тексту та записати їх у текстовий файл. Визначити в кожному рядку кількість слів і розділових символів. Дописати значення кількості слів на початок кожного рядка, значення кількості розділових символів в кінець рядка файлу.
3. У програму лабораторної роботи №8 додати нові функції обробки бінарних файлів:

* створений у лабораторній роботі №8 словник записати до бінарного файлу;
* надрукувати словник у вигляді таблиці, зчитавши дані з бінарного файлу;
* розробити функції, які реалізують операції додавання у бінарний файл нових даних, видалення даних за запитом користувача, заміни одних значень іншими на вимогу користувача;
* результати запитів, які реалізовані в лабораторній роботі №8, записати до бінарного файлу і вивести на екран.

1. **Скласти програму для обробки текстового та бінарного файлів. Передбачити меню вибору задач.**
2. Створити текстовий файл, перший рядок якого містить значення n та m, які визначають кількість рядків та стовпців матриці. Наступні n рядків містять по m чисел, що є елементами матриці. Визначити максимальне значення в кожному рядку матриці та дописати їх в кожний рядок файлу, починаючи з другого.
3. У програму лабораторної роботи №8 додати нові функції обробки бінарних файлів:

* створений у лабораторній роботі №8 словник записати до бінарного файлу;
* надрукувати словник у вигляді таблиці, зчитавши дані з бінарного файлу;
* розробити функції, які реалізують операції додавання у бінарний файл нових даних, видалення даних за запитом користувача, заміни одних значень іншими на вимогу користувача;
* результати запитів, які реалізовані в лабораторній роботі №8, записати до бінарного файлу і вивести на екран

1. **Скласти програму для обробки текстового та бінарного файлів. Передбачити меню вибору задач**
2. Створити текстовий файл, рядки якого містять дані про назву фірми, назву товару, ціну в доларах. Перерахувати ціни в $ у відповідності з курсом в гривні та дописати в кожний рядок файлу отримані значення.
3. У програму лабораторної роботи №8 додати нові функції обробки бінарних файлів:

* створений у лабораторній роботі №8 словник записати до бінарного файлу;
* надрукувати словник у вигляді таблиці, зчитавши дані з бінарного файлу;
* розробити функції, які реалізують операції додавання у бінарний файл нових даних, видалення даних за запитом користувача, заміни одних значень іншими на вимогу користувача;
* результати запитів, які реалізовані в лабораторній роботі №8, записати до бінарного файлу і вивести на екран

1. **Скласти програму для обробки текстового та бінарного файлів. Передбачити меню вибору задач**
2. Створити текстовий файл. Визначити кількість рядків, які починаються з заданого користувачем символу, починаються та закінчуються одним й тим самим символом, складаються з однакових символів, є пустими рядками. Отримані значення записати в окремий файл з відповідними коментаріями.
3. У програму лабораторної роботи №8 додати нові функції обробки бінарних файлів:

* створений у лабораторній роботі №8 словник записати до бінарного файлу;
* надрукувати словник у вигляді таблиці, зчитавши дані з бінарного файлу;
* розробити функції, які реалізують операції додавання у бінарний файл нових даних, видалення даних за запитом користувача, заміни одних значень іншими на вимогу користувача;
* результати запитів, які реалізовані в лабораторній роботі №8, записати до бінарного файлу і вивести на екран

1. **Скласти програму для обробки текстового та бінарного файлів. Передбачити меню вибору задач**
2. Створити текстовий файл F. Увести з клавіатури рядок символів S. Отримати всі рядки файлу F, що містять у собі підрядок S і записати їх до нового файлу G. Останнім рядком файлу G записати кількість знайдених у файлі F рядків.
3. У програму лабораторної роботи №8 додати нові функції обробки бінарних файлів:

* створений у лабораторній роботі №8 словник записати до бінарного файлу;
* надрукувати словник у вигляді таблиці, зчитавши дані з бінарного файлу;
* розробити функції, які реалізують операції додавання у бінарний файл нових даних, видалення даних за запитом користувача, заміни одних значень іншими на вимогу користувача;
* результати запитів, які реалізовані в лабораторній роботі №8, записати до бінарного файлу і вивести на екран

1. **Скласти програму для обробки текстового та бінарного файлів. Передбачити меню вибору задач**
2. Створити текстовий файл, який містить додатні, від’ємні, нульові числа та довільні символи. Визначити кількість додатних, від’ємних, нульових чисел та слів у кожному рядку файлу. Записати отримані значення з відповідними коментарями в інший текстовий файл.
3. У програму лабораторної роботи №8 додати нові функції обробки бінарних файлів:

* створений у лабораторній роботі №8 словник записати до бінарного файлу;
* надрукувати словник у вигляді таблиці, зчитавши дані з бінарного файлу;
* розробити функції, які реалізують операції додавання у бінарний файл нових даних, видалення даних за запитом користувача, заміни одних значень іншими на вимогу користувача;
* результати запитів, які реалізовані в лабораторній роботі №8, записати до бінарного файлу і вивести на екран

1. **Скласти програму для обробки текстового та бінарного файлів. Передбачити меню вибору задач**
2. Створити текстовий файл, який містить числа та символи. Визначити в кожному рядку файлу середнє арифметичне та середнє геометричне чисел рядка. Переписати в новий текстовий файл тільки числа з попереднього файлу та їх середні значення.
3. У програму лабораторної роботи №8 додати нові функції обробки бінарних файлів:

* створений у лабораторній роботі №8 словник записати до бінарного файлу;
* надрукувати словник у вигляді таблиці, зчитавши дані з бінарного файлу;
* розробити функції, які реалізують операції додавання у бінарний файл нових даних, видалення даних за запитом користувача, заміни одних значень іншими на вимогу користувача;
* результати запитів, які реалізовані в лабораторній роботі №8, записати до бінарного файлу і вивести на екран

1. **Скласти програму для обробки текстового та бінарного файлів. Передбачити меню вибору задач**
2. Створити текстовий файл, який би містить марку автомобіля, його вартість, кілометраж. Створити новий текстовий файл, в якому міститимуться рядки з першого файлу, упорядковані за алфавітом марки автомобіля. В кожний рядок нового файлу додати середній кілометраж по всіх марках у файлі.
3. У програму лабораторної роботи №8 додати нові функції обробки бінарних файлів:

* створений у лабораторній роботі №8 словник записати до бінарного файлу;
* надрукувати словник у вигляді таблиці, зчитавши дані з бінарного файлу;
* розробити функції, які реалізують операції додавання у бінарний файл нових даних, видалення даних за запитом користувача, заміни одних значень іншими на вимогу користувача;
* результати запитів, які реалізовані в лабораторній роботі №8, записати до бінарного файлу і вивести на екран

1. **Скласти програму для обробки текстового та бінарного файлів. Передбачити меню вибору задач**
2. Створити два текстових файлів. Рядки першого файлу містять назву товару та його ціну. Рядки другого файлу містять значення кількості кожного товару. Створити третій текстовий файл, кожний рядок якого має містить назву товару, його кількість та ціну. У третьому файлі рядки повинні містити назви тих товарів, що не повторюються.
3. У програму лабораторної роботи №8 додати нові функції обробки бінарних файлів:

* створений у лабораторній роботі №8 словник записати до бінарного файлу;
* надрукувати словник у вигляді таблиці, зчитавши дані з бінарного файлу;
* розробити функції, які реалізують операції додавання у бінарний файл нових даних, видалення даних за запитом користувача, заміни одних значень іншими на вимогу користувача;
* результати запитів, які реалізовані в лабораторній роботі №8, записати до бінарного файлу і вивести на екран

1. **Скласти програму для обробки текстового та бінарного файлів. Передбачити меню вибору задач**
2. Створити текстовий файл, рядки якого містять числа, що задають координати вершин трикутника. Визначити тип трикутника (прямокутний, рівнобічний тощо) за значеннями його сторін. Дописати в кожний рядок вхідного файлу слово, що визначає тип трикутника.
3. У програму лабораторної роботи №8 додати нові функції обробки бінарних файлів:

* створений у лабораторній роботі №8 словник записати до бінарного файлу;
* надрукувати словник у вигляді таблиці, зчитавши дані з бінарного файлу;
* розробити функції, які реалізують операції додавання у бінарний файл нових даних, видалення даних за запитом користувача, заміни одних значень іншими на вимогу користувача;
* результати запитів, які реалізовані в лабораторній роботі №8, записати до бінарного файлу і вивести на екран

1. **Скласти програму для обробки текстового та бінарного файлів. Передбачити меню вибору задач**
2. Створити текстовий файл, рядки якого містять по 3 числа, що задають коефіцієнти рівняння прямої в декартових координатах (Ax+By+C=0) на площині. Визначити лінії, що є паралельними, перпендикулярними та ті, що перетинаються. Створити новий текстовий файл, в рядки якого записати по 6 чисел – коефіцієнтів двох прямих та слово, що визначає їх взаємне розташування.
3. У програму лабораторної роботи №8 додати нові функції обробки бінарних файлів:

* створений у лабораторній роботі №8 словник записати до бінарного файлу;
* надрукувати словник у вигляді таблиці, зчитавши дані з бінарного файлу;
* розробити функції, які реалізують операції додавання у бінарний файл нових даних, видалення даних за запитом користувача, заміни одних значень іншими на вимогу користувача;
* результати запитів, які реалізовані в лабораторній роботі №8, записати до бінарного файлу і вивести на екран

1. **Скласти програму для обробки текстового та бінарного файлів. Передбачити меню вибору задач**
2. Створити текстовий файл. Пронумерувати рядки файлу. Визначити довжину найдовшого рядка; номер найдовшого рядка (якщо таких рядків декілька, то номери усіх із них). Записати в новий файл найдовший рядок.
3. У програму лабораторної роботи №8 додати нові функції обробки бінарних файлів:

* створений у лабораторній роботі №8 словник записати до бінарного файлу;
* надрукувати словник у вигляді таблиці, зчитавши дані з бінарного файлу;
* розробити функції, які реалізують операції додавання у бінарний файл нових даних, видалення даних за запитом користувача, заміни одних значень іншими на вимогу користувача;
* результати запитів, які реалізовані в лабораторній роботі №8, записати до бінарного файлу і вивести на екран

1. **Скласти програму для обробки текстового та бінарного файлів. Передбачити меню вибору задач**
2. У текстовому файлі, що містить код програми на Python, перевірити кількість відступів (пробілів) в тілі операторів циклів. Якщо кількість відступів в тілі операторів циклів менше 4-х, то переписати в новий файл тіло таких операторів, інакше записати повідомлення ‘everything is fine’.
3. У програму лабораторної роботи №8 додати нові функції обробки бінарних файлів:

* створений у лабораторній роботі №8 словник записати до бінарного файлу;
* надрукувати словник у вигляді таблиці, зчитавши дані з бінарного файлу;
* розробити функції, які реалізують операції додавання у бінарний файл нових даних, видалення даних за запитом користувача, заміни одних значень іншими на вимогу користувача;
* результати запитів, які реалізовані в лабораторній роботі №8, записати до бінарного файлу і вивести на екран

1. **Скласти програму для обробки текстового та бінарного файлів. Передбачити меню вибору задач**
2. Створити текстовий файл. Вилучити з файлу всі слова, довжина яких менше заданої користувачем, та пусті рядки. Записати в останній рядок текстового файлу кількість вилучених слів.
3. У програму лабораторної роботи №8 додати нові функції обробки бінарних файлів:

* створений у лабораторній роботі №8 словник записати до бінарного файлу;
* надрукувати словник у вигляді таблиці, зчитавши дані з бінарного файлу;
* розробити функції, які реалізують операції додавання у бінарний файл нових даних, видалення даних за запитом користувача, заміни одних значень іншими на вимогу користувача;
* результати запитів, які реалізовані в лабораторній роботі №8, записати до бінарного файлу і вивести на екран

1. **Скласти програму для обробки текстового та бінарного файлів. Передбачити меню вибору задач**
2. Створити текстовий файл. Отримати його копію. У вхідному файлі замінити усі входження одного слова на інше. Слова задаються користувачем з клавіатури.
3. У програму лабораторної роботи №8 додати нові функції обробки бінарних файлів:

* створений у лабораторній роботі №8 словник записати до бінарного файлу;
* надрукувати словник у вигляді таблиці, зчитавши дані з бінарного файлу;
* розробити функції, які реалізують операції додавання у бінарний файл нових даних, видалення даних за запитом користувача, заміни одних значень іншими на вимогу користувача;
* результати запитів, які реалізовані в лабораторній роботі №8, записати до бінарного файлу і вивести на екран

1. **Скласти програму для обробки текстового та бінарного файлів. Передбачити меню вибору задач**
2. Створити два тестових файлів. Визначити та вивести рядки, які співпадають в обох файлах, кількість рядків, що відрізняються. Переписати в новий файл рядки, що співпадають в обох файлах
3. У програму лабораторної роботи №8 додати нові функції обробки бінарних файлів:

* створений у лабораторній роботі №8 словник записати до бінарного файлу;
* надрукувати словник у вигляді таблиці, зчитавши дані з бінарного файлу;
* розробити функції, які реалізують операції додавання у бінарний файл нових даних, видалення даних за запитом користувача, заміни одних значень іншими на вимогу користувача;
* результати запитів, які реалізовані в лабораторній роботі №8, записати до бінарного файлу і вивести на екран

1. **Скласти програму для обробки текстового та бінарного файлів. Передбачити меню вибору задач**
2. Створити текстовий файл, що містить рядки зі знаками оклику та запитання. Замінити усі знаки оклику у файлі на знак ‘#’. Переписати в новий файл рядки, що закінчуються на знак запитання. Останній рядок нового файлу має містити число, що визначає кількість рядків зі знаком запитання.
3. У програму лабораторної роботи №8 додати нові функції обробки бінарних файлів:

* створений у лабораторній роботі №8 словник записати до бінарного файлу;
* надрукувати словник у вигляді таблиці, зчитавши дані з бінарного файлу;
* розробити функції, які реалізують операції додавання у бінарний файл нових даних, видалення даних за запитом користувача, заміни одних значень іншими на вимогу користувача;
* результати запитів, які реалізовані в лабораторній роботі №8, записати до бінарного файлу і вивести на екран

1. **Скласти програму для обробки текстового та бінарного файлів. Передбачити меню вибору задач**
2. Створити два текстових файлів, елементами яких є числа. Якщо два файли мають однакову кількість чисел, то отримати третій файл, кожен елемент якого дорівнює сумі відповідних елементів заданих файлів та більшому із відповідних елементів заданих файлів. Якщо кількість чисел у файлах різна, то в новий файл записати відповідне повідомлення.
3. У програму лабораторної роботи №8 додати нові функції обробки бінарних файлів:

* створений у лабораторній роботі №8 словник записати до бінарного файлу;
* надрукувати словник у вигляді таблиці, зчитавши дані з бінарного файлу;
* розробити функції, які реалізують операції додавання у бінарний файл нових даних, видалення даних за запитом користувача, заміни одних значень іншими на вимогу користувача;
* результати запитів, які реалізовані в лабораторній роботі №8, записати до бінарного файлу і вивести на екран

1. **Скласти програму для обробки текстового та бінарного файлів. Передбачити меню вибору задач**
2. Створити текстовий файл F. Записати до файлу G рядки файлу F в зворотному порядку, виключивши заданий користувачем символ. Якщо заданий символ у файлі F відсутній, то дописати у кожний рядок файлу G відсутній у файлі F символ.
3. У програму лабораторної роботи №8 додати нові функції обробки бінарних файлів:

* створений у лабораторній роботі №8 словник записати до бінарного файлу;
* надрукувати словник у вигляді таблиці, зчитавши дані з бінарного файлу;
* розробити функції, які реалізують операції додавання у бінарний файл нових даних, видалення даних за запитом користувача, заміни одних значень іншими на вимогу користувача;
* результати запитів, які реалізовані в лабораторній роботі №8, записати до бінарного файлу і вивести на екран

1. **Скласти програму для обробки текстового та бінарного файлів. Передбачити меню вибору задач**
2. Створити текстовий файл, записавши в нього цілі числа, значення яких генерує генератор псевдовипадкових чисел в діапазоні, заданому користувачем з клавіатури. Знайти серед цих чисел такі, що є квадратами цілих чисел і записати їх до нового текстового файлу.
3. У програму лабораторної роботи №8 додати нові функції обробки бінарних файлів:

* створений у лабораторній роботі №8 словник записати до бінарного файлу;
* надрукувати словник у вигляді таблиці, зчитавши дані з бінарного файлу;
* розробити функції, які реалізують операції додавання у бінарний файл нових даних, видалення даних за запитом користувача, заміни одних значень іншими на вимогу користувача;
* результати запитів, які реалізовані в лабораторній роботі №8, записати до бінарного файлу і вивести на екран

1. **Скласти програму для обробки текстового та бінарного файлів. Передбачити меню вибору задач**
2. Створити текстовий файл, кожний рядок якого містить слово англійською мовою та його переклад українською. Створити копію створеного попередньо файлу. В кожний рядок файлу-копії після кожного слова англійською мовою записати його транскрипція.
3. У програму лабораторної роботи №8 додати нові функції обробки бінарних файлів:

* створений у лабораторній роботі №8 словник записати до бінарного файлу;
* надрукувати словник у вигляді таблиці, зчитавши дані з бінарного файлу;
* розробити функції, які реалізують операції додавання у бінарний файл нових даних, видалення даних за запитом користувача, заміни одних значень іншими на вимогу користувача;
* результати запитів, які реалізовані в лабораторній роботі №8, записати до бінарного файлу і вивести на екран.

## Список літератури

1. <https://github.com/tkovalyuk/Basics-of-programming>
2. **МакГрат М**. Программирование на Python для начинающих. –Москва: Эксмо. – 192 с.
3. **Лутц М.** Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
4. **Лутц М.** Программирование на Python, том I, 2. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
5. **Васильев А. Н.** Python на примерах. Практический курс ·по программированию. - СПб.:Наука и Техника, 2016. - 432 с.
6. **Доусон М.** Программируем на Python. - СПб.: Питер, 2014. - 416 с.
7. **Рейтц К., Шлюссер Т.** Автостопом по Python. — СПб.: Питер, 2017. — 336 с.

# Лабораторна робота 10 Обробка виключних ситуацій в Python

**Мета роботи:** Ознайомитись з технологією обробки виключних ситуацій в Python. Побудувати блок-схеми алгоритмів, розробити програми з використанням засобів обробки виключних ситуацій.

## Рейтинг лабораторної роботи №10

|  |  |  |  |
| --- | --- | --- | --- |
| № п.п | Вид діяльності студента | Рейтинговий бал | Deadline |
| 1 | Написання коду | 2 | грудень |
| 2 | Захист роботи | 1 |
| 3 | Звіт з роботи | 1 |
| 4 | Якість роботи (за наявності) | 1 |
| Разом за роботу мовою Python | | 5 |

## Теоретичний матеріал

Якщо у програмі виникає синтаксична помилка або помилка часу виконання, то відбувається **виключна ситуація**. Будь-яка виключна ситуація генерує (породжує) **виключення.**

Виключення (eng: exception) – спеціальний тип даних, що використовується для ідентифікації та детального опису помилки, що виникла у програмі.

Вбудовані виключення у мові Python організовані у вигляді ієрархії класів:

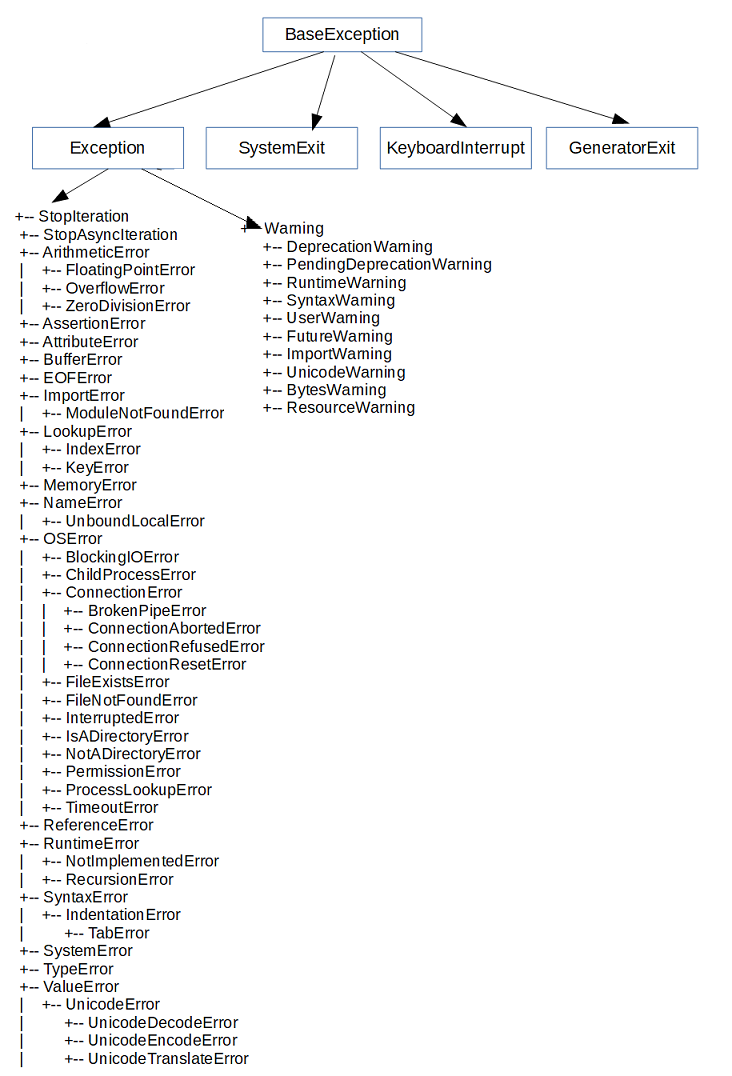


Рис. 1. Дерево класів виключень

### Обробка виключень

Для обробки виключень використовується оператор **try**.

**try**:

processing\_code #Инструкции

**except** exception\_1: #тип исключения

handling\_exception\_1 #Что делать, например, закрыть файл

**except**:

handling\_exception\_2 #Что делать при любом ещё не пойманном

#исключении в этом блоке

**else**:

state\_else #Что делать, если не было исключений

**finally**:

obligatory\_code #Что делать в любом случае после блока

**Виконання цього оператора відбувається таким чином:**

1. Інтерпретатор намагається виконати блок коду з потенційним виключенням (processing\_code).
2. Якщо у випадку виконання коду породжується виключення, що належить до класу виключень exception\_1 то у точці генерування виключення відбувається безумовний перехід на блок обробки виключення handling\_exception\_1. Аналогічне стосується інших типів виключення, що перехоплюються у програмі.
3. Якщо, породжене виключення не обробляється у жодному з блоків обробки виключень, то інтерпретатор припиняє роботу програми (за допомогою типового обробника виключень)
4. Якщо виконання блоку **try** повністю відбулося без виключень, то відбувається виконання коду state\_else блоку **else**. Якщо ж було згенеровано хоча б одне виключення цей блок не виконується.
5. Код obligatory\_code блоку **finaly** виконується завжди після виконання попередніх кроків незалежно від того, чи було породжено виключення чи ні.

Обов’язковими блоками оператора **try** є лише блок **try** і один блок **except** обробки виключень або блок **try** і блок **finally**

**Приклад.** Створити масив чисел, які вводять з клавіатури. Здійснити контроль введення через обробку виключних ситуацій. Знайти суму елементів масиву. Кількість елементів визначається в процесі вводу. Якщо користувач вводить не числове значення, то генерується виключення.

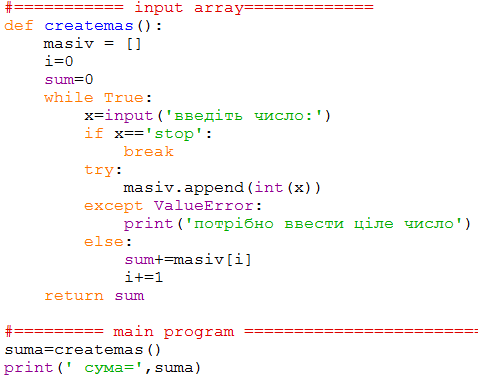


Рис. 2. Приклад коду

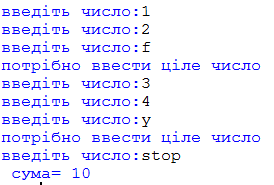


Рис. 3. Результати роботи програми

## Завдання для самостійної роботи за варіантами

Програму реалізувати мовою Python за завданнями лабораторної роботи №7 (а). Додати до коду лабораторної роботи №7 (а) обробку виключних ситуацій відповідно до завдання.

Для підвищення рейтингу написати програму лабораторної роботи №7 (а) мовою С без обробки виключних ситуацій. Кількість додаткових балів за код мовою С дорівнює 5.

### Варіанти завдань

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій**

Ввести список чисел з клавіатури. Елементи списку є додатними, від’ємними та рівними нулю числами. З введеного списку сформувати новий список, взявши в нього тільки ті елементи із вхідного списку, які більше за модулем введеного з клавіатури числа M. Вивести на екран число M, вхідний та отриманий списки. Передбачити виключні ситуації: нечислового символу, відсутність від’ємних та нульових чисел.

1. **Скласти програму для обробки одновимірного масиву. Передбачити** введення **обробку виключних ситуацій**

Створити список чисел, кількість елементів якого задана користувачем. Визначити максимальне та мінімальне значення серед елементів із парними та непарними індексами. Вивести список, мінімальний, максимальний елементи та їх індекси. Передбачити виключні ситуації: введення не додатної кількості елементів масиву та нечислових значень елементів масиву.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій**

Створити числовий список, кількість елементів якого задана користувачем. Обчислити суму елементів між максимальним та мінімальними значеннями списку. Вивести на екран список, максимальний та мінімальний елементи масиву та їх індекси. Передбачити виключні ситуації: введення нечислових значень елементів масиву, вихід індексів елементів за межі масиву.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Створити два числових списки, кількість елементів яких задана користувачем. Знайти найменший серед тих елементів першого списку, які співпадають із значеннями елементів другого списку. Вивести на екран списки, найменший елемент та його індекс. Передбачити виключні ситуації: введення нечислових значень елементів масиву, відсутність однакових елементів в двох списках.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Створити список цілих чисел. Побудувати новий список із елементів першого списку. В новому списку спочатку стоять числа, що діляться на 2, потім ті, що діляться на 2 та 3, потім на 3. Надрукувати вхідний та вихідний списки. Передбачити виключні ситуації: введення нечислових значень елементів масиву, відсутність елементів, які кратні 2 та 3.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Створити список. Обчислити кількість елементів у найдовшій серії. Серія — це послідовність однакових елементів, розташованих поспіль. Надрукувати список, усі серії однакових елементів та довжину найдовшої серії. Передбачити виключні ситуації: введення нечислових значень елементів масиву, відсутність послідовностей однакових елементів.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

У згенерованому числовому списку знайти найбільший серед від’ємних та найменший серед додатних елементів списку. Вивести список, значення знайдених елементів та їх індекси. Передбачити виключні ситуації: введення некоректної (нульової, від’ємної, нечислової) вимірності масиву, відсутність від’ємних елементів в масиві.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Згенерувати список з цілих чисел та вивести його на екран. Знайти максимальний елемент та його індекс та надрукувати їх. Вивести на екран кількість елементів, які менші за значення максимального. Передбачити виключні ситуації: введення некоректної (нульової, від’ємної, нечислової) вимірності масиву та вихід індексу за межі масиву.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Згенерувати два упорядкованих за зростанням числових списків. Побудувати упорядкований за зростанням третій список злиттям двох згенерованих. Передбачити виключні ситуації: введення некоректної (нульової, від’ємної, нечислової) вимірності масиву та вихід індексу за межі масиву.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Ввести з клавіатури два списки цілих чисел. Кількість елементів списків вводиться з клавіатури. Побудувати третій список, в якому кожний елемент дорівнює найбільшому спільному дільнику відповідних елементів введених списків. Використати алгоритм Евкліда. Надрукувати усі три списки. Передбачити виключні ситуації: введення некоректної (нульової, від’ємної, нечислової) вимірності масиву, невідповідність двох вихідних списків.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Створити числовий список, кількість елементів якого вводиться користувачем. Значення елементів в списку можуть повторюватися. Визначити індекс першого та останнього входження заданого числа в список. Знайти елемент, значення якого повторюється найбільшу кількість разів. Вивести на екран список, шуканий елемент та його індекс. Передбачити виключні ситуації: введення некоректної (нульової, від’ємної, нечислової) вимірності масиву, вихід індексів за межі масиву.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Створити список, значеннями елементів якого є коефіцієнти многочлена I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_07\pictures\var11.gif. Знайти значення многочлена при введеному з клавіатури аргументу *x* та похідної від многочлена *A*(*x*) при заданому *x*. Вивести на екран список, значення многочлена та його похідної. Передбачити виключні ситуації: введення нечислових значень коефіцієнтів многочлена та аргументу *х*

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Згенерувати список цілих чисел, кількість елементів якого ввести з клавіатури. Вивести список на екран. Отримати інший список, що складається тільки з непарних чисел вихідного списку або повідомити, що таких чисел немає. Отриманий список вивести на екран в порядку спадання елементів. Передбачити виключні ситуації: введення нечислового, нульового, від’ємного значення кількості елементів списку, відсутність непарних чисел в масиві.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Ввести з клавіатури список цілих чисел. Знайти в списку все прості числа і скопіювати їх в новий список. Надрукувати вхідний та вихідний списки. Передбачити виключні ситуації: некоректність введених даних, відсутність простих чисел.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Згенерувати список чисел, кількість елементів якого ввести з клавіатури. У списку чисел перевірити, чи всі елементи є унікальними, тобто кожне число зустрічається тільки один раз. Надрукувати згенерований список та повідомлення «Усі елементи унікальні» або «Не усі елементи унікальні», які сформувати в результаті обробки виключних ситуацій.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Згенерувати числовий список, кількість елементів якого ввести з клавіатури. Визначити індекси елементів списку, значення яких не менше заданого мінімуму і не більше заданого максимуму. Вивести на екран список, індекси та значення елементів, що належать заданому діапазону. Передбачити виключні ситуації: некоректне введення кількості елементів масиву, відсутність елементів в заданому діапазоні.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Згенерувати числовий список з елементами від -100 до +100. Надрукувати список. Вивести послідовність елементів між першим від’ємним і нульовим елементом, між першим додатним і нульовим елементом. Передбачити виключні ситуації: відсутність елементів масиву , що відповідають заданим діапазонам.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Ввести з клавіатури числовий список з трьох елементів. У кожного елемента списку є вага, яка збільшується на одиницю, коли користувач вибирає цей елемент списку. Початкове значення ваг дорівнює нулю. Користувач має *n* спроб вибрати кожний елемент списку. Вибір елементів здійснюється випадково за допомогою генератора псевдовипадкових чисел. При випаданні потрібного елемента, його вага збільшується. Вивести на екран значення ваг елементів після *n* спроб вибрати їх. Передбачити виключні ситуації: введення нечислових значень списку, генератор не генерує числа, які є в масиві.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Ввести з клавіатури список чисел. Визначити, скільки в цьому списку елементів, які більше двох своїх сусідів справа та зліва. Вивести кількість таких елементів. Крайні елементи списку не враховуються, оскільки у них недостатньо сусідів. Передбачити виключні ситуації: введення нечислових значень списку, відсутність елементів, які більше двох своїх сусідів справа та зліва

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Згенерувати список чисел, кількість елементів якого ввести з клавіатури. Вивести згенерований список. Обчислити кількість пар елементів, рівних один одному. Вважається, що будь-які два елементи, які рівні один одному, утворюють одну пару. Вивести індекси елементів, що утворюють пару та кількість пар. Передбачити виключні ситуації: введення нечислового значення кількості елементів масиву, відсутність пар елементів, рівних один одному.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Згенерувати список цілих додатних та від’ємних чисел, кількість елементів якого ввести з клавіатури. Знайти суму елементів, розташованих до першого від’ємного елементу та добуток елементів, розташованих після останнього від’ємного елемента. Передбачити виключні ситуації: введення нечислового значення кількості елементів масиву, відсутність від’ємних елементів.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Ввести з клавіатури список чисел. Видалити зі списку всі числа, які повторюються більше двох разів. Надрукувати вхідний список і список після видалення елементів. Передбачити виключні ситуації: введення нечислових значень, відсутність чисел, які повторюються більше двох разів в масиві.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Згенерувати список цілих чисел в діапазоні від -50 до +50, кількість елементів задати з клавіатури. Із отриманого списку скопіювати в інший список елементи, значення яких знаходяться в діапазоні від -5 до 5 включно і підрахувати їх кількість. Надрукувати згенерований і побудований списки. Передбачити виключні ситуації: відсутність елементів, значення яких знаходяться в діапазоні від -5 до 5.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Дан список цілих чисел. Розмір списку і його елементи вводяться користувачем. Упорядкувати за зростанням елементи списку, що розташовані до максимального елемента, і за спаданням – ті елементи, що розташовані після максимального елементу. Надрукувати список до та після сортування. Передбачити виключні ситуації: введення нечислових значень розміру та елементів списку, вихід індексів за межі масиву

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Дан список дійсних чисел. Розмір списку і значення його елементів вводяться користувачем. Знайти суму всіх локальних мінімумів списку. Локальний мінімум – це елемент списку, який менший за значенням двох сусідніх елементів даного списку (виключаючи крайні елементи). Надрукувати вхідний список, всі його локальні мінімуми та їх індекси. Передбачити виключні ситуації: введення нечислових значень розміру та елементів списку та відсутність локальних мінімумів.

## Список літератури

1. <https://github.com/tkovalyuk/Basics-of-programming>
2. **МакГрат М**. Программирование на Python для начинающих. –Москва: Эксмо. – 192 с.
3. **Лутц М.** Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
4. **Лутц М.** Программирование на Python, том I, 2. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
5. **Васильев А. Н.** Python на примерах. Практический курс ·по программированию. - СПб.:Наука и Техника, 2016. - 432 с.
6. **Доусон М.** Программируем на Python. - СПб.: Питер, 2014. - 416 с.
7. **Рейтц К., Шлюссер Т.** Автостопом по Python. — СПб.: Питер, 2017. — 336 с.