

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

МЕТОДИЧНІ ВКАЗІВКИ
ДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ
З ДИСЦИПЛІНИ «ПРОГРАМУВАННЯ»
(для студентів спеціальності "Комп'ютерна інженерія")

Ухвалено
на засіданні кафедри ОТ
протокол № 13 від 10.05.2023 р.

Погоджено
на засіданні методичної
комісії ФІОТ
протокол №11 від 30.06.2023 р.

Методичні вказівки до виконання лабораторних робіт з дисципліни «Програмування» (для студентів спеціальності "Комп'ютерна інженерія"). /Укл.: Новотарський М. А.: НТУУ «КПІ» – 2023 .

Методичні вказівки до виконання лабораторних робіт з дисципліни «Програмування» містять завдання до лабораторних робіт. До кожної роботи наведено методичні рекомендації з теорії, яка використовується при виконанні лабораторних робіт, приведені деякі алгоритми вирішення завдань, вимоги до виконання роботи, індивідуальні завдання та контрольні питання.

Укладач: проф. Новотарський М.А.

Рецензент: проф. Стіренко С.Г.

Лабораторна робота №1

Тема: «Загальні поняття про технічні характеристики та програмне забезпечення сучасних комп'ютерів».

Мета: вивчення основних складових частин персональних та професійних комп'ютерів, рівнів програмного забезпечення сучасних комп'ютерів. Інсталяція середовища розробки PyCharm & Anaconda.

Загальне завдання:

1. Вивчити матеріал лекцій 1 та 2.
2. Інсталювати Anaconda – дистрибутив для мови програмування Python.
3. Інсталювати IDE PyCharm community або PyCharm professional.

Теоретичні основи:

Знання теоретичних основ по темі даної лабораторної роботи необхідне для успішної здачі тесту №1. Необхідними також є базові навички володінням комп'ютером під керуванням операційної системи Windows та принципами пошуку інформації в Internet.

Anaconda – дистрибутив для мови програмування Python з відкритим кодом для обробки даних великого обсягу, побудови аналітичних прогнозів і наукових обчислень. Anaconda використовують для того, щоб спростити управління і використання пакетів. Скачати Anaconda3 2020.07 можна за посиланням <https://anaconda.com>

Anaconda працює під операційними системами Windows, Mac OS X і Linux.

PyCharm – інтегроване середовище розробки для мови програмування Python. Надає засоби для аналізу коду, графічний редактор коду, інструмент для запуску юніт-тестів і підтримує веб-розробку на Django.

PyCharm розроблена чеською компанією JetBrains на основі IntelliJ IDEA.

PyCharm працює під операційними системами Windows, Mac OS X і Linux.

Під час навчання будемо користуватися спеціальною версією PyCharm & Anaconda, яка містить вбудований плагін для зв'язку з Anaconda. Існує два дистрибутиви PyCharm for Anaconda.

1. PyCharm Community for Anaconda є безкоштовною.
2. PyCharm Professional for Anaconda платна, але для викладачів та студентів можна одержати безкоштовну ліцензію.

Скачати PyCharm&anaconda можна за посиланням:

<https://jetbrains.com>

Вимоги до програмного забезпечення:

1. Для успішної задачі лабораторної роботи на Вашому ноутбукі повинен бути встановлений дистрибутив **Anaconda** у випадку, якщо ви працюєте під ОС Windows 10 або вище.
2. Повинно бути встановлено інтегроване середовище розробки **PyCharm**

Порядок виконання лабораторної роботи:

1. При встановленні програмного забезпечення у корені диску створити папку проекту та назвати її своїм прізвищем, записаним латиницею.
2. Скориставшись інструкціями з лекції 2, створити файл `myfirst.py`.
3. Записати у вікні редактора коду PyCharm свою першу програму, яка виводить текст: `Hello, World!`
4. Знати назву основних вікон IDE PyCharm
5. Вміти відкрити вікно «Python Console»
6. Вміти запустити програму в інтерпретаторі та в компіляторі Python
7. Користаючись засобами операційної системи, зайти в папку проекту і запустити програму **Jupyter notebook**.
8. Створити свій перший зошит у папці проекту. Порядок створення зошиту та його вміст розглянуто у лекції 2.
9. Створити та запустити на виконання комірку, яка виводить текст «Hello, World!»
10. Знати відповіді на контрольні питання.

Порядок оформлення звіту

За результатами лабораторної роботи необхідно підготувати звіт. Звіт можна готувати у паперовому вигляді, або у формі файлу з розширенням `pdf`. Звіт обов'язково повинен складатися з таких частин.

1. Титульний лист

Титульний лист повинен мати вигляд, що показаний нижче:

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Програмування

Лабораторна робота №1

«Загальні поняття про технічні характеристики
та програмне забезпечення сучасних комп'ютерів»

Виконав:
студент групи ІО-ХХ
П.І.Б. студента
Номер у списку групи: ХХ
Перевірив:
П.І.Б. викладача

Київ – 2023

2. Вміст звіту

1. Тема лабораторної роботи

Тему переписуємо з цих методичних вказівок.

2. Мета лабораторної роботи та загальне завдання

Мету роботи також скопіювати з даних методичних вказівок.

3. Короткі теоретичні відомості по темі лабораторної роботи.

Коротенький конспект матеріалу (1–2 сторінки), який необхідно знати для успішного виконання лабораторної роботи.

4. Відобразити результати виконання лабораторної роботи у вигляді роздруківки коду (код повинен бути надрукований чорними літерами на білому фоні).

5. Навести скріншоти, які підтверджують виконання роботи.

У скріншотах мають бути відображені одержані персонально вами результати

6. Висновки.

У висновках необхідно провести аналіз проблем, які виникли при виконанні лабораторної роботи і описати шляхи їх вирішення.

Контрольні питання (надавати відповіді у звіті не потрібно)

1. Назвіть обладнання, яке входить до складу персонального комп'ютера.

2. Які рівні програмного забезпечення Вам відомі.

3. Наведіть приклади програм системного рівня.

4. Мета застосування IDE.

5. Поясніть призначення дистрибутиву Anaconda.

6. Як запустити Jupyter notebook із заданої папки.

7. Як створити папку та зошит у середовищі Jupyter.

8. Як перейменувати зошит у середовищі Jupyter.

9. Як запустити виконання комірки в Jupyter.

10. Як запускати програми в IDE PyCharm.

Лабораторна робота №2

Тема: «Типи даних, змінні та оператори мови програмування Python».

Мета: вивчити типи даних, які використовуються в мові програмування Python. Змінні та правила їх іменування, операції над змінними. Оператори та їх застосування.

Завдання:

1. Вивчити матеріал лекцій 3, 4, 5 та 6.
2. Виконати індивідуальне завдання лабораторної роботи, вибране відповідно до варіанту.

Теоретичні основи:

Основні типи даних

`bool` – логічний тип даних. –
`NoneType` – об'єкт зі значенням `None`.
`int` – цілі числа.
`float` – дійсні числа.
`complex` – комплексні числа.
`str` – **Unicode**-рядки.
`bytes` – незмінювана послідовність байтів.
`bytearray` – змінювана послідовність байтів.
`list` – списки.
`tuple` – кортежі.
`range` – діапазони.
`dict` – словники.
`set` – множини.

Змінні

~~12x~~ – не може починатися з цифри
~~привіт~~ – не використовуємо кирилицю
~~break~~ – не повинна співпадати з ключовими словами
~~%, %b~~ – не починаємо зі службових знаків
`x=23` – присвоєння значення
`x = int(input("x = "))` – ввід значення
`x, y, z = 1, 2, 3` – позиційне присвоювання
`del x` – видалення змінної

Математичні оператори

`+` – додавання.
`-` – віднімання.

* – множення.
/ – ділення.
// – Ділення з округленням униз.
% – остача від ділення.
** – піднесення до степеня.
Унарний мінус (-) і унарний плюс (+) .

Двійкові математичні оператори

~ – двійкова інверсія.
& – двійкове І.
| – двійкове АБО.
^ – двійкове виключення.
<< – зсув вліво.
>> – зсув вправо.

Оператори для послідовностей

+ - конкатенація.
* - повторення.
in - перевірка на входження.
not in - перевірка на невходження.

Оператори присвоювання

= - присвоює змінній значення.
+= - збільшує значення.
-= - зменшує значення.
*= - множить значення.
/= - ділить значення.
//= - ділення з округленням вниз й присвоювання.
%= - ділення по модулю й присвоювання.
** = - піднесення до степеня і присвоювання.

Оператори порівняння

== - дорівнює.
!= - не дорівнює.
< - менше.
> - більше.
<= - менше або дорівнює.
>= - більше або дорівнює.
is - перевіряє, чи посилаються дві змінні на той самий об'єкт
or - логічне АБО.
not – логічне заперечення.
and – логічне І.

Оператори розгалуження й цикли

Оператор розгалуження if ... else

```
if <Логічний вираз>:  
    <Блок, виконуваний, якщо умова дійсна>  
    [elif <Логічний вираз>:  
        <Блок, виконуваний, якщо умова дійсна>  
    ]  
[else:  
    <Блок, виконуваний, якщо всі умови неправильні>  
]
```

Оператор циклу for

```
for <Поточний елемент> in <Послідовність>:  
<Інструкції усередині циклу>  
[else:  
<Блок, виконуваний, якщо не використовувався оператор break>  
]
```

Оператор циклу while

```
<Початкове значення>  
while <Условие>:  
    <Інструкції>  
    <Збільшення>  
[else:  
    <Блок, виконуваний, якщо не використовувався оператор  
break>  
]
```

Функції range() і enumerate()

```
range ([<Початок>,,] <Кінець> [, <Крок>])  
enumerate ( <Об'єкт> [, start=0])
```

Оператор continue

continue дозволяє перейти до наступної ітерації циклу до завершення виконання всіх інструкцій всередині циклу

Оператор break

break дозволяє перервати виконання циклу достроково

Числа

int – цілі числа.

float – дійсні числа;

complex – комплексні числа.

Двійкові числа починаються з комбінації символів 0b

Вісімкові числа починаються з нуля й наступної за ним латинської букви o (регістр не має значення) і містять цифри від 0 до 7:

Шістнадцяткові числа починаються з комбінації символів 0x (або 0X) і можуть містити цифри від 0 до 9 і букви від А до F

Операції з фіксованою точністю

```
>>> from decimal import Decimal
>>>Decimal("0.3")-Decimal("0.1")-Decimal("0.1")-
Decimal("0.1")
Decimal ('0.0')
```

Операції з дробами

```
from fractions import Fraction
>>> Fraction(4, 5)
Fraction(4, 5)
```

Функції для роботи з числами

```
bin(<Число>)
oct(<Число>)
hex(<Число>)
float([<число або рядок>])
round(<Число>[,<Кількість знаків після точки>])
Abs(<Число>) Повертає абсолютне значення
pow(<Число>,<Степінь>[,<Дільник>])
max(<Список чисел через кому>)
min ( <список чисел через кому>)
sum (<Послідовність>[,<Початкове значення>])
divmod(x, y) Повертає кортеж із двох значень (x // y, x % y)
```

Модуль *math*. Математичні функції

Pi – повертає число π
e – повертає значення константи e:
sin(), cos(), tan() – стандартні тригонометричні функції
asin(), acos(), atan()– обернені тригонометричні функції
degrees() – перетворює радіани в градуси.
radians()– перетворює градуси в радіани.
exp() – експонента.
log(<число> [, <База>]) – логарифм по заданій базі.
log10() – десятковий логарифм.
log2() – логарифм по базі 2.
sqrt() – квадратний корінь.
ceil() – значення, округлене до найближчого більшого цілого
floor() – значення, округлене до найближчого меншого цілого
pow(<Число>,<Степінь>) – підносить <Число> до <Степеня>.
fabs () – абсолютне значення.

`fmod()` – залишок від ділення.

`factorial()` – факторіал числа.

`fsum(<Список чисел>)` – повертає точну суму чисел із заданого списку.

Модуль *random*. Генерація випадкових чисел

```
import random
```

```
from random import *
```

`random()` – повертає псевдовипадкове число від 0.0 до 1.0.

`seed([<параметр>][,version=2])` – налаштовує генератор випадкових чисел на нову послідовність.

`uniform(<початок>, <кінець>)` – повертає псевдовипадкове дійсне число в діапазоні від <Початок> до <Кінець>.

`randint(<початок>, <кінець>)` – повертає псевдовипадкове ціле число в діапазоні від <Початок> до <Кінець>.

`randrange([<початок>,]<кінець>[, <Крок>])` – повертає випадковий елемент із числової послідовності. Параметри аналогічні параметрам функції `range()`.

`choice(<послідовність>)` – повертає випадковий елемент із заданої послідовності (рядка, списку, кортежу).

`shuffle(<список>[, <Число від 0.0 до 1.0>])` – перемішує елементи списку випадковим чином.

`sample(<Послідовність>, <Кількість елементів>)` – повертає список із зазначеної кількості елементів, які будуть обрані випадковим чином із заданої послідовності.

Індивідуальні завдання

Завдання 1

1. Відповідно до **номера у списку** вибрати номер варіанта виразу.
2. Написати програму обчислення виразу.
3. Забезпечити ввід значень для всіх змінних, що знаходяться в правій частині виразу, з клавіатури комп'ютера.
4. Програма повинна працювати з дійсними числами.
5. Забезпечити вивід результатів обчислень.
6. Програма обчислення виразу повинна коректно працювати при всіх можливих значеннях змінних.

№	Вираз	№	Вираз
1	$R = \frac{e^{2x} + \sin y}{\ln(3.8x + y)}$	16	$F = \frac{4x^3 + \ln y}{e^{z+y} + 7.2 \sin y}$
2	$R = \ln x + \frac{3.5x + 1}{\cos 2y}$	17	$F = \frac{y^2 - 0.8y + \sqrt{y}}{23.1x^2 + \cos x}$
3	$R = \frac{\ln(x - y) + y^4}{e^y + 2.355x^2}$	18	$F = \frac{\sqrt{\sin^2 y + 6.835}}{\ln(x + y) + 3y^2}$
4	$R = \frac{9.33x^3 + \sqrt{x}}{\ln(y + 3.5) + \sqrt{y}}$	19	$F = \frac{\ln(0.7y + 2x)}{\sqrt{3y^2 + 0.5y + 4}}$
5	$R = \frac{7.8x^2 + 3.52y}{\ln(x + 2z) + e^z}$	20	$F = \frac{2x^2 + 3y + 7.2}{\ln z + e^{2y}}$
6	$R = \frac{0.81 \cos x}{\ln y + 2x^3}$	21	$F = \frac{\sqrt{x + 2.6y \sin z}}{a - b^3}$
7	$R = \frac{x^2 + 2.8x + 0.355}{\cos 2y + 3.6}$	22	$F = \frac{4.351y^3 + 2x \ln x}{\sqrt{\cos 2y + 4.351}}$
8	$R = \frac{2.37 \sin(x + 1)}{\sqrt{4y^2 - 0.1y + 5}}$	23	$F = \frac{\sin^2 y + 9.3x}{e^y + \ln x}$
9	$R = \frac{(2x + y)^3}{\ln(y + 0.75)}$	24	$F = \frac{\ln(2x + 4.3)}{e^{x+y} + \sqrt{y}}$
10	$R = \frac{2x + y \cos x}{\sqrt{y + 4.831}}$	25	$F = \cos^2 x + \frac{3y^2 + 4}{\sqrt{x + y}}$
11	$R = x^2 + \frac{0.5y + 4.8}{\sin x}$	26	$F = \frac{\sin 2x}{\ln(2y + x)}$
12	$R = \frac{\sin(2x + 1)^2 + 0.3}{\ln(x + y)}$	27	$F = \frac{\sin(x + 0.4)^2}{y^2 + 7.325x}$
13	$R = \frac{\sin(2x + y) + y^2}{e^y + x}$	28	$F = \frac{0.004x + e^{2y}}{e^{y/2}}$
14	$R = \frac{e^{y+2.5} + 7.1x^3}{\ln \sqrt{y + 0.04x}}$	29	$F = \frac{0.355x^2 - 4.355}{e^{x+y} + \sqrt{2.7y}}$
15	$R = \frac{2 \sin(0.354y + 1)}{\ln(y + 2x)}$	30	$F = \frac{3y^2 + \sqrt{y + 1}}{\ln(x + y) + e^x}$

Завдання 2

1. Відповідно до **номера у списку** вибрати індивідуальне завдання.
2. Написати програму на мові Python.
3. Забезпечити ввід даних з клавіатури комп'ютера
4. Забезпечити вивід результатів обчислень.
5. У звіті до лабораторної роботи описати алгоритм, за яким побудована програма.

№	Завдання
1	<p>Ввести з клавіатури два довільні числа a та b. Визначити, що більше, сума квадратів чи квадрат суми цих чисел. Розглянути ситуацію, коли</p> <p>Вивести:</p> <ol style="list-style-type: none">1. Значення квадрата суми2. Значення суми квадратів3. Повідомлення про те, яка величина є більшою <p>Розглянути ситуацію, коли сума квадратів дорівнює квадрату суми</p>
2	<p>Розрахувати надбавку до зарплати за стаж. Якщо стаж від 2 до 5 років, то надбавка становить 2%, якщо стаж від 5 до 10 років – 5%.</p> <ol style="list-style-type: none">1. Ввести з клавіатури зарплату і стаж.2. Вивести надбавку і суму до виплати. <p>1. Зарплата може бути представлена не цілим числом гривень (гривні з копійками)</p> <p>2. Стаж може – складається з заданої кількості років, місяців та днів.</p>
3	<p>Ввести з клавіатури координати точок $A(x_0, y_0)$ і $B(x_1, y_1)$.</p> <ol style="list-style-type: none">1. Координати можуть бути представлені дійсними числами.2. Визначити, яка з точок, A чи B, більше віддалена від початку координат $O(0,0)$.3. Вивести значення відстаней точок A та B від точки O4. Вивести повідомлення про те, яка з точок є більш віддаленою від точки O.
4	<p>Ввести з клавіатури значення трьох сторін трикутника a, b і c</p> <ol style="list-style-type: none">1. Довжини сторін трикутника представлені дійсними числами2. Визначити, чи є даний трикутник прямокутним.3. Якщо трикутник є прямокутним, то вказати, які сторони з введених є катетами, а яка сторона представляє гіпотенузу4. Якщо трикутник не є прямокутним, то вказати, чи є він рівнобедреним або рівностороннім5. Відповідь вивести у вигляді повідомлення, вказавши назву змінної та сторони трикутника, яку вона представляє
5	<p>Ввести з клавіатури чотири числа.</p> <ol style="list-style-type: none">1. Кожне з додатних чисел піднести до квадрату та обчислити їх суму.

№	Завдання
	2. Всі від'ємні числа додати, а результат додавання піднести до квадрату. 3. Вивести результати обчислень та повідомлення про те, яка з двох одержаних величин є більшою
6	Ввести з клавіатури координати точки $A(x, y)$. 1. Координати можуть бути представлені дійсними числами. 2. Визначити, в якому квадранті лежить дана точка. 3. Відповідь вивести у вигляді повідомлення. 4. Передбачити ситуацію, коли точка може знаходитися на одній з осей, про що також потрібно вивести спеціальне повідомлення
7	Ввести з клавіатури координати точки $A(x, y)$ 1. Координати можуть бути представлені дійсними числами. 2. Визначити чи лежить дана точка всередині кола з радіусом R . 3. Центром кола є початок координат. 4. Відповідь вивести у вигляді повідомлення. 5. Передбачити ситуацію, коли точка A лежить на колі радіуса R
8	Ввести з клавіатури значення для довжин сторін двох трикутників a_1, b_1, c_1 і a_2, b_2, c_2 . 1. Довжини сторін трикутника представлені дійсними числами 2. Визначити, площа якого трикутника є більшою. 3. Відповідь вивести у вигляді повідомлення. 4. Передбачити ситуацію, коли площі трикутників дорівнюють одна одній
9	Ввести довжину сторони квадрата a та радіус круга R 1. Визначити площу квадрата (зі стороною a) 2. Визначити площу круга (з радіусом R). 3. Значення a і R ввести з клавіатури у вигляді дійсних чисел. 4. Вивести значення площ квадрата та круга. 5. Визначити, площа якої з фігур є більшою. 6. Відповідь вивести у вигляді повідомлення. 7. Визначити довжину сторони квадрата, рівновеликого до круга з радіусом R .
10	Ввести з клавіатури чотири дійсні числа 1. Визначити, які числа є додатними, а які від'ємними. 2. Додатні числа піднести до третього степеню. 3. Від'ємні піднести до другого степеню. 4. Визначити, який з одержаних результатів є максимальним. 5. Вивести значення всіх результатів обчислень. 6. Вивести окремо максимальний результат.
11	Ввести натуральне число N . 1. Визначити, чи є воно парним 2. Визначити, чи ділиться воно без остачі на 3. 3. Визначити чи ділиться воно без остачі на 5

№	Завдання
	4. Вивести повідомлення по кожній з даних перевірок 5. Вивести числа, які потрібно додати до числа N , щоб отримати подільність на 3 та на 5.
12	Ввести з клавіатури координати точки $A(x,y)$. 1. Координати можуть бути представлені дійсними числами. 2. Визначити, чи лежить дана точка в першому квадранті. 3. Відповідь вивести у вигляді повідомлення. 4. Передбачити ситуацію, коли точка розташована на одній з осей або в центрі координат. Вивести відповідне повідомлення.
13	Ввести суму банківського вкладу та термін договору. 1. Банківський вклад може бути виражений не цілим числом гривень. 2. Якщо договір укладено на півроку то на депозит нараховують 6% річних 3. Якщо договір укладено на рік то на депозит нараховують 8% річних. 4. Розрахувати суму щомісячних виплат відсотків по депозиту, 5. Вивести суму вкладу, саму відсотків, та загальну суму при вчасному поверненні депозиту
14	Ввести з клавіатури два дійсних числа. 1. Обчислити модуль різниці квадратів цих чисел. 2. Обчислити квадрат різниці цих чисел. 3. Визначити, яке число є більшим і за яких умов. 4. Вивести всі проміжні дані 5. Вивести результат порівняння у вигляді повідомлення.
15	Ввести з клавіатури координати точок $A(x_0,y_0)$ і $B(x_1,y_1)$. Визначити, яка з точок A чи B найменше віддалена від початку координат $O(0,0)$. Відповідь вивести у вигляді повідомлення.
16	Ввести з клавіатури координати точки $A(x,y)$. 1. Координати можуть бути представлені дійсними числами. 2. Ввести параметри тора, утвореного колами з радіусами r і R з центром в точці $O(0,0)$. 3. Визначити, чи лежить дана точка на торі 4. Відповідь вивести у вигляді повідомлення.
17	Ввести з клавіатури координати точки $A(x,y)$. 1. Координати можуть бути представлені дійсними числами. 2. Визначити, чи лежить дана точка в четвертому квадранті. 3. Відповідь вивести у вигляді повідомлення. 4. Передбачити ситуацію, коли точка розташована на одній з осей або в центрі координат. Вивести відповідне повідомлення.
18	Ввести з клавіатури значення трьох сторін трикутника a , b і c

№	Завдання
	<ol style="list-style-type: none"> 1. Довжини сторін a, b і c можуть бути представлені дійсними числами. 2. Визначити, чи може існувати трикутник з заданими величинами сторін 3. Визначити, чи є даний трикутник рівнобедреним. 4. Визначити, чи є даний трикутник рівностороннім. 5. Усі відповіді вивести у вигляді відповідних повідомлень.
19	<p>Ввести з клавіатури три цілих числа (a, b, c).</p> <ol style="list-style-type: none"> 1. Визначити, чи є вони трійкою Піфагора ($c^2 = a^2 + b^2$ або $a^2 = b^2 + c^2$ або $b^2 = a^2 + c^2$). 2. Відповідь вивести у вигляді повідомлення. 3. Якщо числа a та b не утворюють трійку Піфагора з числом c, то визначити таке число c', яке утворюватиме таку трійку.
20	<p>Визначити, яка об'ємна швидкість перенесення рідини більше, 1л/с чи 10^{-3} м³/хв.</p> <ol style="list-style-type: none"> 1. Ввести об'ємні швидкості перетікання рідини 2. Визначити скільки рідини буде перенесено за годину з одного резервуару до іншого з швидкістю перенесення 1л/с 2. Визначити скільки рідини буде перенесено за годину з одного резервуару до іншого з швидкістю перенесення 10^{-3} м³/хв. 3. Вивести відповідне повідомлення про те, яка швидкість перетікання є більшою 4. Вивести .всі результати обчислень.
21	<p>Ввести з клавіатури площу кола S_1 та площу квадрата S_2</p> <ol style="list-style-type: none"> 1. Визначити, чи поміститься коло в квадраті. 2. Визначити, чи вміститься квадрат в колі. 3. Вивести відповідне повідомлення. 4. Передбачити повідомлення, коли коло є вписаним в квадрат. 5. Передбачити повідомлення, коли коло квадрат є вписаним в коло.
22	<ol style="list-style-type: none"> 1. Ввести з клавіатури дані про розмір сторони куба та масу тіла кубічної форми. 2. Ввести з клавіатури дані про величину радіуса та масу тіла у формі кулі. 3. Вивести повідомлення матеріал якого з тіл має більшу щільність? 4. Вивести результати обчислень. 5. Всі величини є дійсними числами
23	<p>Ввести з клавіатури дві швидкості</p> <ol style="list-style-type: none"> 1. Одну швидкість виразити в кілометрах за годину, 2. Іншу швидкість представити в метрах за секунду. 3. Визначити, яка з них більша та вивести повідомлення. 4. Швидкості виразити дійсними числами.

№	Завдання
	5. Передбачити ситуацію, коли швидкості співпадають
24	<p>Ввести значення площі рівностороннього трикутника S_{tr} та круга S_{cir}</p> <ol style="list-style-type: none"> Дані площі можуть бути представлені дійсними числами. Відомо, що існують співвідношення $S_{tr} = a^2 \frac{\sqrt{3}}{4}$ і $r = a \frac{\sqrt{3}}{6}$, де a – довжина сторони трикутника; r – радіус круга Визначити, чи вміститься круг в трикутнику; Визначити, чи вміститься трикутник в крузі. Вивести повідомлення та числові результати.
25	<p>Введіть маси та радіуси двох довільних планет Сонячної системи. Наприклад: Венера ($M=4.86 \times 10^{27}$ г, $R=6175$ км) Сатурн ($M=5.68 \times 10^{29}$ г, $R=57750$ км)</p> <ol style="list-style-type: none"> Визначити, яка з планет має більше прискорення сили тяжіння і у скільки разів. Формула для визначення прискорення сили тяжіння має такий вигляд: $g = \frac{G \cdot M}{R^2}$, $G = 6.7 \times 10^{-8} \frac{\text{см}^2}{\text{г} \cdot \text{сек}^2}$ – універсальна гравітаційна стала. Визначити, чи є густина Марса більшою за густину Землі. Вивести відповідне повідомлення.
26	<p>Введіть діаметр та масу трьох планет Сонячної системи.</p> <ol style="list-style-type: none"> Визначте, яка з планет має найбільшу густину Виведіть відповідне повідомлення. Виведіть результати проміжних обчислень
27	<p>Ввести з клавіатури поточний час у Києві</p> <ol style="list-style-type: none"> Визначити час доби (am – з 0 до 12:00, pm – з 12:00 до 24:00) у Парижі, Лондоні та Нью-Йорку. Визначити різницю у часі між цими містами Вивести результати обчислень.
28	<p>Ввести з клавіатури географічну довготу і широту місця на Землі.</p> <ol style="list-style-type: none"> Значення координат задати в градусах, мінутах та секундах. Визначити, в якій півкулі знаходиться вибране місце (в східній або західній, південній або північній). Визначити відстань між вибраним місцем та Північним полюсом Визначити найкоротшу відстань між вибраним місцем та Гринвіцьким меридіаном Вивести результати обчислень
29	<p>Ввести відстані до двох найяскравіших зірок північної півкулі: Сіріус (Сузір'я Великого Пса) - 8.14×10^{12} км Арктур (сузір'я Волопаса) – 103 парсека (1 пс = 3.259 світлових роки).</p>

№	Завдання
	<ol style="list-style-type: none"> 1. Визначити, яка зірка більше віддалена від нас. 2. Вивести відповідне повідомлення. 3. Визначити різницю відстаней в кілометрах парсеках та світлових роках 4. Вивести результати обчислень
30	<p>Ввести натуральне число.</p> <ol style="list-style-type: none"> 1. Визначити, чи є воно парним або непарним 2. Визначити, чи ділиться воно на 7 без остачі. 3. Визначити, чи ділиться воно на 11 без остачі. 4. Вивести результати обчислень..

Завдання 3

Відповідно до **номера у списку** групи вибрати індивідуальне завдання. Написати програму на мові Python . Забезпечити ввід даних з клавіатури комп'ютера та друк результатів обчислень. У звіті до лабораторної роботи описати алгоритм, за яким побудована програма.

№	Завдання
1	<ol style="list-style-type: none"> 1.Ввести з клавіатури два цілих числа A і B ($A < B$). 2.Знайти всі цілі числа, що розташовані між даними числами (включаючи самі ці числа), 3. Вивести числа в порядку їх зростання. 4.Вивести кількість N цих чисел.
2	<ol style="list-style-type: none"> 1.Ввести з клавіатури ціле число N (> 0). 2. Знайти добуток $1 \cdot 2 \cdot \dots \cdot N$. 3. Для того, щоб уникнути переповнення, обчислювати цей добуток за допомогою дійсної змінної 4. Вивести результат, як дійсне число.
3	<ol style="list-style-type: none"> 1.Ввести з клавіатури дійсне число A і ціле число N (> 0). 2. Знайти A в степені N шляхом множення числа A самого на себе $A^N = A \cdot A \cdot \dots \cdot A$ (числа A перемножуються N разів). 3. Вивести проміжні результати множення та кінцевий результат.
4	<ol style="list-style-type: none"> 1.Ввести з клавіатури дійсне число A і ціле число N (> 0). 2.Знайти всі степені числа A, які не більше від числа N. 3.Вивести значення степенів та відповідні цілі показники степеня числа A.
5	<ol style="list-style-type: none"> 1.Ввести з клавіатури дійсне число A і ціле число N (> 0). 2.Обчисли вираз: $B = 1 + A + A^2 + A^3 + \dots + A^N$ 3. Обчислити вираз:

№	Завдання
	$C = 1 - A + A^2 - A^3 + A^4 - A^5 + \dots + (-1)^N A^N$.
6	1. Ввести з клавіатури ціле число N (> 0). 2. Сформувати послідовність чисел від 1 до N . 3. Знайти найменше ціле число M серед послідовності чисел, при якому виконується нерівність $3M > N$. 4. Вивести на друк число M та число $3M$.
7	1. Ввести з клавіатури ціле число N (> 0). 2. Сформувати послідовність чисел від 1 до N . 3. Знайти найбільше ціле число M серед послідовності чисел, при якому виконується нерівність $3M < N$. 4. Вивести на друк число M та число $3M$.
8	1. Ввести з клавіатури дійсне число A (> 1). 2. Вивести найменше із цілих чисел N , для яких сума $S = 1 + \frac{1}{2} + \dots + \frac{1}{N}$ буде більше A , (тобто, $S > A$) 3. Вивести значення суми S .
9	1. Ввести з клавіатури два цілих числа A і B ($A < B$). 2. Знайти всі цілі числа, які розташовані між даними числами (не включаючи самі ці числа), в порядку їх зменшення, 3. Знайти кількість N цих чисел.. 4. Вивести знайдені числа та їх кількість.
10	1. Ввести з клавіатури ціле число N . 2. Знайти добуток чисел $2 \cdot \frac{1}{2} \cdot \frac{1}{3} \cdot \dots \cdot \frac{1}{N}$ за умови, що $N > 0$. 3. Знайти добуток чисел $2 \cdot \frac{1}{2} \cdot \left(-\frac{1}{3}\right) \cdot \dots \cdot (-1)^{ N } \frac{1}{ N }$ за умови, що $N < 0$ 4. Вивести результати обчислень
11	1. Ввести з клавіатури дійсне число X і ціле число N (> 0). 2. Знайти значення $B = 1 + X + \frac{X^2}{2} + \dots + \frac{X^N}{N}$ 3. Вивести результат обчислення
12	1. Ввести з клавіатури дійсне число X і ціле число N (> 0). 2. Знайти значення

№	Завдання
	$B = X - \frac{X^3}{3} + \frac{X^5}{5} + \dots + (-1)^N \frac{X^{(2N+1)}}{2N+1}.$ <p>3. Вивести результат обчислення.</p>
13	<p>1.Ввести з клавіатури дійсне число X і ціле число N (> 0).</p> <p>2.Знайти значення:</p> $B = \frac{X^2}{2} - \frac{X^4}{4} + \dots + (-1)^{(N+1)} \frac{X^{2N}}{2N}.$ <p>3. Вивести результат обчислення.</p>
14	<p>1.Ввести з клавіатури дійсне число X (X <1) і ціле число N (> 0).</p> <p>2.Знайти значення</p> $B = X - \frac{X^2}{2} + \frac{X^3}{3} + \dots + (-1)^{(N+1)} \frac{X^N}{N}$ <p>3. Вивести результат обчислення.</p>
15	<p>1.Ввести з клавіатури дійсне число X (X <1) і ціле число N (> 0).</p> <p>2.Знайти значення:</p> $B = X + \frac{X^3}{3} - \frac{X^5}{5} + \dots + (-1)^{(N+1)} \frac{X^{(2N+1)}}{2N+1}$ <p>3. Вивести результат обчислення.</p>
16	<p>1.Ввести з клавіатури ціле число N (> 2)</p> <p>2. Ввести дві дійсні точки на числовій осі: A, B (A <B).</p> <p>3.Відрізок [A, B] розбити на рівні відрізки довжини H з кінцями в N точках з координатами A, A + H, A + 2H, A + 3H, ..., B.</p> <p>4.Вивести значення H і набір координат з N точок, який утворює розбиття відрізка [A, B].</p>
17	<p>1.Ввести з клавіатури ціле число N (> 2) і дві дійсні точки на числовій осі: A, B (A <B).</p> <p>2.Функцію $F(x)$ задати формулою $F(x) = 1 - \sin(x)$</p> <p>3. Вивести значення функції $F(x)$ в N рівновіддалених точках, що утворюють розбиття відрізка [A, B], яке представлене координатами: $F(A), F(A + H), F(A + 2H), \dots, F(B)$.</p>
18	<p>1.Ввести з клавіатури число $D > 0$.</p> <p>2.Визначити послідовність чисел $\{a_i\}_{i=1}^{\infty}$ таким чином: $a_1 = 2, a_i = 2 + \frac{1}{a_{i-1}}$</p>

№	Завдання
	<p>,де $i = 2, 3, \dots$.</p> <p>3. Знайти перший з індексів i, для яких виконується умова $a_i - a_{i-1} < D$.</p> <p>4. Вивести цей індекс, а також числа a_{i-1} і a_i.</p>
19	<p>1. Ввести з клавіатури число $D > 0$.</p> <p>2. Визначити послідовність чисел $\{a_i\}_{i=1}^{\infty}$ таким чином: $a_1 = 1, a_2 = 2, a_i = \frac{(a_{i-2} + a_{i-1})}{2}$, де $i = 3, 4, 5, \dots$</p> <p>3. Знайти перший з індексів i, для яких виконується умова $a_i - a_{i-1} < D$.</p> <p>4. Вивести цей індекс, а також числа a_{i-1} і a_i.</p>
20	<p>1. Ввести з клавіатури натуральне число $N > 100$.</p> <p>2. Сформувати послідовність чисел від 10 до N.</p> <p>3. Вивести на непарні числа</p> <p>4. Вивести кратні п'яти числа.</p>
21	<p>1. Ввести з клавіатури натуральне число $N > 100$.</p> <p>2. Сформувати послідовність від 10 до N.</p> <p>3. Вивести на друк числа, яким відповідають натуральні числа, як квадратні корені та самі ці квадратні корені (наприклад: 36 та 6).</p> <p>4. Вивести на друк числа, яким відповідають натуральні числа, як кубічні корені та самі ці кубічні корені (наприклад: 27 та 3).</p>
22	<p>1. Послідовно ввести значення x.</p> <p>2. Якщо $0 < x < 3.14$, то обчислити $r = \sin(x)$, якщо $-3.14 \leq x < 0$, то $r = x^3$, якщо $x < -2 \cdot 3.14$ то $r = x^2$, якщо або $x > 2 \cdot 3.14$ та у всіх інших випадках $r = 0$.</p> <p>3. Вивести результати обчислень.</p>
23	<p>1. Ввести з клавіатури число n.</p> <p>2. Для заданого n в одному циклі обчислити $n!$ і 2^n.</p> <p>3. Вивести результати обчислень</p>
24	<p>1. Ввести з клавіатури натуральне число $N > 100$</p> <p>2. Сформувати послідовність натуральних чисел від 10 до N.</p> <p>3. Знайти числа які містять найбільшу цифру у десятковій системі числення.</p> <p>4. Вивести ці числа та їх кількість</p>

№	Завдання
25	1. Ввести з клавіатури натуральне число $N > 100$. 2. Сформувати послідовність натуральних чисел від 10 до N . 3. Для натуральних чисел послідовності знайти числа з найменшою сумою цифр, 4. Вивести ці числа 5. В кожному числі з вибраних чисел знайти найменшу цифру. 3. Вивести результати пошуку
26	1. Ввести з клавіатури натуральне число $N > 100$. 2. Сформувати послідовність натуральних чисел від 10 до N . 3. Визначити симетричні числа, тобто такі числа, які однаково читаються зліва направо і справа наліво (паліндром). Наприклад, числа 123321, 202, 9889, 5555. 4. Вивести результати пошуку
27	1. Ввести з клавіатури натуральне число $N > 100$. 2. Сформувати послідовність натуральних чисел $i = 11, 12, \dots, N$ 3. Обчислити послідовність $s_i = s_{i-1} + i^2$ за умови, що $s_{10} = 10$, 4. Вивести на друк елементи послідовності $\{s_i\}$
28	1. Ввести з клавіатури натуральне число $N > 100$. 2. Сформувати послідовність натуральних чисел $i = 21, 23, \dots, N$ 3. Обчислити послідовність $s_i = i^2 - s_{i-1}$ за умови, що $s_{20} = 20$, 4. Вивести на друк елементи послідовності $\{s_i\}$.
29	1. Ввести з клавіатури дійсне число A і натуральні числа від 1 до N , де $N > 0$. 2. Знайти різниці квадратів A з кожним з натуральних чисел від 1 до N . 3. Вивести значення квадратів та різниці
30	1. Ввести з клавіатури натуральне число $N > 100$. 2. Сформувати послідовність натуральних чисел від 10 до N 3. Знайти та вивести на друк суму цифр кожного числа.

Зміст звіту:

1. Титульний лист повинен мати такий вигляд:

Національний технічний університет України
«Київський політехнічний інститут ім. І. Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Програмування
Лабораторна робота №2
«Типи даних, змінні та оператори мови програмування Python».

Виконав:
студент групи ІО-XX
П.І.Б. студента
Номер у списку групи: XX
Перевірив:
(П.І.Б. викладача)

Київ 2023р.

2. Мета лабораторної роботи та загальне завдання
3. Короткі теоретичні відомості, які відображають типи даних та оператори, **що були використані при написанні лабораторної роботи.**
4. Скріншот вашого варіанту з методички
5. Опис алгоритм розв'язування (у довільній формі).
6. Роздруківка того фрагменту тексту програми, який написаний індивідуально.
7. Скріншот результатів виконання програми з контрольним прикладом
8. Аналіз результатів та висновки.

Контрольні питання

1. Які числові типи даних вам відомі?
2. У якому кодуванні зберігаються рядки тексту при застосуванні типу str?
3. Які типи даних задають послідовності?
4. Як відрізняється запис двійкових операторів І та АБО від логічних операторів І та АБО?
5. Запишіть найпростіший і повний оператор розгалуження?
6. Які види операторів циклу вам відомі?

Лабораторна робота №3

Тема: «Робота з даними типу str, bytes та bytearray».

Мета: вивчити способи створення рядків та даних типу bytes і bytearray, операції над ними. Форматування рядків. Функції та методи роботи з рядками. Налаштування локалі.

Завдання:

1. Вивчити матеріал лекцій 7, 8, 9 та 10.
2. Виконати індивідуальне завдання лабораторної роботи, вибране відповідно до варіанту.

Теоретичні основи:

Створити рядок можна такими способами:

1. За допомогою функції

```
str ( [ <Об'єкт> [, <Кодування> [, <Обробка помилок>] ] ] )  
>>> str(), str([1, 2])
```

2. Указавши рядок між апострофами або подвійними лапками

```
>>> print ('рядок1\nрядок2')
```

3. Указавши рядок між потроєними апострофами або потроєними лапками.

```
>>> print(''''Рядок1  
Рядок2''')
```

Перелік спеціальних символів, припустимих всередині рядка, перед яким немає модифікатора r:

\n - перевід рядка;
\r - повернення каретки;
\t - знак табуляції;
\v - вертикальна табуляція;
\a - **дзвінок**;
\b - **вибій**;
\f - перевід формату;
\0 - нульовий символ (не є кінцем рядка);
\" - лапки;
' - апостроф;
\N - вісімкове значення N. Наприклад, \74 відповідає символу <;
\xn - шістнадцяткове значення N. Наприклад, \x6a відповідає символу j;
\ - зворотний (обернений) слеш;
\uxxxx - 16-бітний символ Unicode. Наприклад, \u043a відповідає російській букві к;

\Uxxxxxxxx-32-бітний символ Unicode

Створити об'єкт типу `bytes` можна у декілька способів.

Спосіб 1. За допомогою функції

```
bytes([<Рядок>, <Кодування>[, <Обробка помилок>]])
```

Спосіб 2. За допомогою методу рядків

```
encode([encoding="utf-8"][, errors="strict"])
```

Спосіб 3.

Указавши букву `b` (регістр не має значення) перед рядком в апострофах, лапках, потрійних апострофах або потрійних лапках.

Спосіб 4.

За допомогою функції `bytes(<Послідовність>)`, яка перетворює послідовність цілих чисел від 0 до 255 в об'єкт типу `bytes`.

Спосіб 5

За допомогою функції `bytes(<Число>)`, яка задає кількість елементів у послідовності. Кожний елемент буде містити нульовий символ

Спосіб 6

За допомогою методу `bytes.fromhex (<Рядок>)`.

Перетворити об'єкт типу `bytes` у рядок дозволяє метод `decode()`.

```
decode([encoding="utf-8"][, errors="strict"])
```

Для перетворення можна також скористатися функцією `str()`:

```
>>> b = bytes("рядок", "cp1251")
>>> str(b, "cp1251")
'рядок'
```

Створити об'єкт типу `bytearray` можна такими способами:

Спосіб 1. За допомогою функції

```
bytearray([<Рядок>, <Кодування>[, <Обробка помилок>]])
```

Спосіб 2

За допомогою функції `bytearray (<Послідовність>)`, яка перетворює послідовність цілих чисел від 0 до 255 в об'єкт типу `bytearray`.

Спосіб 3

За допомогою функції `bytearray(<Число>)`, яка задає кількість елементів у послідовності.

Спосіб 4

За допомогою методу `bytearray.fromhex (<Рядок>)`.

Операції з рядками

1. Доступ до символу за індексом в квадратних дужках:

```
>>> s = "Python"
>>> s[0]
'P'
```

2. Доступ за від'ємним індексом відраховується з кінця:

```
>>> s = "Python"
>>> s[-1]
'n'
```

3. Змінити рядок по індексу неможливо

4. Операція добування зрізу для зміни рядка

```
[<Початок>:<Кінець>:<Крок>]
```

```
>>> s[2:5] # повертаються символи з індексами 2, 3 и 4
'tho'
```

Форматування рядків

<Рядок спеціального формату> % <Значення>

Всередині параметра <Рядок спеціального формату> можуть бути зазначені специфікатори, що мають наступний синтаксис:

% [(<Ключ>)] [<Прапор>] [<Ширина>] [. <Точність>] <Тип перетворення>

Для форматування рядків можна також використовувати наступні методи:

`expandtabs ([<Ширина поля>])` – заміняє символ табуляції пробілами таким чином, щоб загальна ширина фрагмента разом з текстом, розташованим перед символом табуляції, дорівнювала зазначеній величині.

`center (<Ширина> [, <Символ>])` – виконує вирівнювання рядка по центру всередині поля зазначеної ширини.

`ljust (<Ширина> [, <Символ>])` – виконує вирівнювання рядка по лівому краю всередині поля зазначеної ширини.

`rjust (<Ширина> [, <Символ>])` – виконує вирівнювання рядка по правому краю усередині поля зазначеної ширини.

`zfill (<Ширина>)` – виконує вирівнювання фрагмента по правому краю усередині поля зазначеної ширини.

Метод format()

<Рядок> = <Рядок спеціального формату> . `format` (*args, **kwargs)

У параметрі <Рядок спеціального формату> усередині символів фігурних дужок: { i } вказуються специфікатори, що мають наступний синтаксис:

```
{ [ <Поле> ] [ ! <Функція> ] [ : <Формат> ] }
>>> print("Символи {{i}} - {0}".format("спеціальні"))
Символи {i} - спеціальні
```

У параметрі <Формат> вказується значення, що має наступний синтаксис:

```
[ [ <Заповнювач> ] <Вирівнювання> ] [ <Знак> ] [ # ] [ 0 ]
[ <Ширина> ] [ , ] [ . <Точність> ] [ <Перетворення> ]
```

За замовчуванням значення усередині поля вирівнюється по правому краю. Управляти вирівнюванням дозволяє параметр `<Вирівнювання>`. Можна вказати наступні значення: `<`, `>`, `,` `=`

- `<` – по лівому краю;
- `>` – по правому краю;
- `^` – по центру поля.
- `=` – знак числа вирівнюється по лівому краю, а число по правому краю
- `+` - задає обов'язковий вивід знака як для від'ємних, так і для додатних чисел.

Функції

`str([<Об'єкт>])` – перетворює будь-який об'єкт у рядок.
`repr(<Об'єкт>)` – повертає строкове представлення об'єкта.
`ascii(<Об'єкт>)` – повертає строкове представлення об'єкта.
`len(<Рядок>)` – повертає кількість символів у рядку.

Методи

`strip([<Символи>])` – видаляє зазначені в параметрі символи на початку й наприкінці рядка.
`lstrip([<Символи>])` – видаляє «пропускові» або зазначені символи на початку рядка.
`rstrip([<Символи>])` – видаляє «пропускові» або зазначені символи наприкінці рядка.
`split([<Роздільник>[, <Ліміт>]])` – розділяє рядок на підрядки по зазначеному роздільнику й додає ці підрядки в список, який повертається як результат.
`rsplit([<Роздільник>[, <Ліміт>]])` – аналогічний методу `split()`, але пошук символу-роздільника проводиться не зліва направо, а, навпаки, справа наліво.
`splitlines([True])` – розділяє рядок на підрядки по символу переведу рядка (`\n`) і додає їх у список.
`partition(<Роздільник>)` – знаходить перше входження символу-роздільника в рядок і повертає кортеж із трьох елементів: перший елемент буде містити фрагмент, розташований перед роздільником, другий елемент – сам роздільник, а третій елемент – фрагмент, розташований після роздільника. Пошук проводиться зліва направо.
`rpartition(<Роздільник>)` – метод аналогічний методу `partition()`, але пошук символу-роздільника проводиться не зліва направо, а, навпаки, справа наліво.
`join()` – перетворить послідовність у рядок. Елементи додаються через вказаний роздільник. Формат методу:
`<Рядок>=<Роздільник>.join(<Послідовність>)`

Налаштування локалі

Для установки локалі (сукупності локальних налаштувань системи) служить функція `setlocale()` з модуля `locale`. Перш ніж використовувати функцію, необхідно підключити модуль за допомогою виразу:

```
import locale
```

```
setlocale(<категорія>[, <Локаль>]);
```

Параметр `<категорія>` може набувати наступних значень:

```
locale.LC_ALL – установлює локаль для всіх режимів;  
locale.LC_COLLATE – для порівняння рядків;  
locale.LC_CTYPE – для переведення символів у нижній або верхній  
регістр;  
locale.LC_MONETARY – для відображення грошових одиниць;  
locale.LC_NUMERIC – для форматування чисел;  
locale.LC_TIME – для форматування виводу дати й часу.
```

Одержати поточне значення локалі дозволяє функція

```
getlocale([<Категорія>]).
```

Одержати налаштування локалі дозволяє функція `localeconv()`.

Зміна регістру символів

`upper()` – замінює всі символи рядка відповідними прописними буквами.

```
>>> print("рядок".upper())
```

РЯДОК

`lower()` – замінює всі символи рядка відповідними малими літерами

`swapcase()` – замінює всі малі літери відповідними прописними буквами, а всі прописні літери – малими:

`capitalize()` – робить першу букву рядка прописною

`title()` – робить першу букву кожного слова прописною

`casefold()` – те ж саме, що й `lower()`, але додатково перетворить усі символи з діакритичними знаками й лігатури в букви стандартної латиниці.

Функції для роботи з символами

`chr(<Код символу>)` – повертає символ по зазначеному коду.

`ord(<Символ>)` – повертає код зазначеного символу.

Пошук і заміна в рядку

`find()` – шукає підрядок в рядку.

```
<Рядок>.find(<Підрядок>[, <Початок>[, <Кінець>]])
```

Якщо початкова позиція не зазначена, то пошук буде здійснюватися з початку рядка. Якщо параметри `<Початок>` і `<Кінець>` зазначені, то проводиться операція добування зрізу:

`<Рядок> [<Початок> : <Кінець>]`

і пошук підрядка буде виконуватися в цьому фрагменті.

`index()` – метод аналогічний методу `find()`, але якщо підрядок в рядок не входить, то виконується виключення `ValueError`.

`<Рядок>.index(<Підстрока> [, <Початок> [, <Кінець>]])`

`rfind()` – шукає підрядок в рядку.

`<Рядок>.rfind(<Підстрока> [, <Початок> [, <Кінець>]])`

`rindex()` – метод аналогічний методу `rfind()`, але якщо підрядок в рядок не входить, то виконується виключення `ValueError`.

`<Рядок>.rindex(<Підстрока> [, <Початок> [, <Кінець>]])`

`count()` – повертає число входжень підрядка в рядок.

`<Рядок>.count(<Підрядок> [, <Початок> [, <Кінець>]])`

Перевірка типу вмісту рядка

`isalnum()` – повертає `True`, якщо рядок містить тільки букви й (або) цифри, у протилежному випадку – `False`.

`isalpha()` – повертає `True`, якщо рядок містить тільки букви, а якщо ні, то – `False`.

`isdigit()` – повертає `True`, якщо рядок містить тільки цифри, а якщо ні, то – `False`.

`isdecimal()` – повертає `True`, якщо рядок містить тільки десяткові символи, а якщо ні, то – `False`.

`isnumeric()` – повертає `True`, якщо рядок містить тільки числові символи, а якщо ні, то – `False`.

`isupper()` – повертає `True`, якщо рядок містить букви тільки верхнього регістру, а якщо ні, то – `False`.

`islower()` – повертає `True`, якщо рядок містить букви тільки нижнього регістру, а якщо ні, то – `False`.

`istitle()` – повертає `True`, якщо всі слова в рядку починаються із великої літери, а якщо ні, то – `False`.

`isprintable()` – повертає `True`, якщо рядок містить символи, що друкуються, а якщо ні, то – `False`.

`isspace()` – повертає `True`, якщо рядок містить тільки «пропускові» символи, у протилежному випадку – `False`:

`isidentifier()` – повертає `True`, якщо рядок є припустимим з погляду Python ім'ям змінної, функції або класу, а якщо ні, то – `False`:

Методи bytearray

`append (<Число>)` – додає один елемент у кінець об'єкта

`extend(<Послідовність>)` – додає елементи послідовності в кінець об'єкта

+ і += використовувати оператори для додавання декількох елементів

Присвоювання значення зрізу:

```
>>> b = bytearray( "string", "ascii")
```

```
>>> b[len(b):] = b"123"
```

```
# Додаємо елементи в послідовність
```

```
>>>b
```

```
bytearray(b'string123')
```

`insert(<Індекс>, <Число>)` – додає один елемент у зазначену позицію.

`pop ([<Індекс>])` – видаляє елемент, розташований по зазначеному індексу, і повертає його.

Вилучити елемент списку дозволяє також оператор `del`.

`remove (<Число>)` – видаляє перший елемент, що містить зазначене значення.

`reverse ()` – змінює порядок проходження елементів на протилежний.

`decode()` – перетворює об'єкт типу `bytearray` в рядок

```
decode ( [encoding="utf-8"] [, errors="strict"])
```

Індивідуальні завдання

Завдання 1

1.Відповідно до номера в списку групи вибрати індивідуальне завдання.

2.Написати програму на мові Python.

3.Забезпечити ввід даних з клавіатури комп'ютера та друк результатів обчислень.

4.У звіті до лабораторної роботи описати алгоритм, за яким побудована програма.

При виводі даних обов'язково використати форматування!!!!

№вар	Завдання
1	У заданому рядку замінити пробіли, які йдуть підряд, на один пробіл.
2	В заданому рядку порахувати кількість слів. Роздільником слів вважається один або кілька пробілів.
3	В заданому рядку замінити кожен кириличну літеру символом «*».
4	В заданому рядку видалити всі латинські літери.
5	Ввести рядок. Дописати в кінець рядка його довжину.
6	У заданому рядку дописати після кожного символу «*» символ «_».
7	У заданому рядку вставити перед кожним символом «!» Символ «?»
8	Задати два однакові за довжиною рядки. Побудувати новий рядок, в якому на парних місцях розташовані елементи першого рядка, а на непарних – елементи другого рядка.
9	Ввести рядок і замінити кожен знайдений пробіл двома пробілами.
10	Ввести рядок. Побудувати новий рядок, в якому всі символи записано в зворотному порядку.
11	Побудувати рядок, що складається з малих букв латинського алфавіту (розташованих за алфавітом).
12	Побудувати рядок, що складається з великих кириличних літер (розташованих за алфавітом).
13	Ввести рядок і поміняти місцями символи, які стоять поруч.
14	Ввести рядок та окремий символ. Видалити введений символ, якщо він знаходиться в рядку.
15	Ввести рядок. Визначити останній символ в рядку. Видалити всі однакові символи, якщо вони знайдуться в рядку.
16	Ввести рядок. Розташувати всі його символи по зростанню їх кодів.
17	Ввести рядок. Замінити кожен символ «№» рядком «номер по порядку».
18	Ввести рядок. Перед кожною великою латинською літерою поставити «. » (крапка, пробіл).
19	Розрядити пробілами введений рядок.
20	Ввести рядок. Замінити поєднання символів «-+» символом «0».
21	Ввести рядок. Замінити кожен символ рядка наступним за кодом символом. Букву «я» замінити пробілом.
22	Ввести рядок. Поміняти кожен символ рядка попереднім за таблицею кодування символом.
23	Ввести рядок. Після кожного символу вставити число, що відповідає коду цього символу.
24	Ввести рядок. Отримати з нього два рядки. Перший має

№вар	Завдання
	складатися з символів введеного рядка, які мають парні і непарні індекси, а другий – з символів з непарними індексами
25	Ввести два рядки однакової довжини. Побудувати новий рядок, який складається з символів як одного, так і іншого рядка, що чергуються між собою.
26	Ввести рядок. У рядку після кожного слова дописати його довжину.
27	Ввести рядок. Отримати передостаннє слово цього рядка. Роздільником слів вважаються один або кілька пробілів.
28	Ввести рядок. Замінити парну кількість пробілів символом «П», а непарну – символом «Н».
29	Ввести рядок. Після кожної кириличної літери поставити її код, а після кожної латинської – символ «_».
30	Ввести рядок. Після кожного слова типу «sin», «cos» або «log», поставити дужки «()».

Завдання 2

1. Відповідно до номера в списку групи вибрати індивідуальне завдання.

2. Написати програму на мові Python з використанням типів даних `bytes` та `bytearray`.

3. Забезпечити ввід даних з клавіатури комп'ютера та друк результатів обчислень.

4. У звіті до лабораторної роботи описати алгоритм, за яким побудована програма.

При виводі даних обов'язково використати форматування!!!

Всі операції в завданні 2 обов'язково виконувати тільки зі змінними типу `bytes` або `bytearray`!!!

№вар	Завдання
1	<p>1. Ввести послідовність символів, в якому зустрічаються структури <code><s></code>, <code><#s></code>.</p> <p>2. Замінити кожне входження <code><s></code> на <code><#s></code>, а кожне входження <code><#s></code> на <code><#></code>.</p> <p>3. Зауваження: в програмі слід врахувати, що <code>s</code> може бути як малим, так і великим.</p>
2	<p>1. Ввести послідовність символів, що містить число в двійковій системі числення.</p> <p>2. Перевірити правильність введення цього числа (в його записі</p>

№вар	Завдання
	повинні бути тільки символи 0 і 1). Якщо число введено невірною, повторити ввід. 3. При правильному введенні перетворити число в десяткову систему числення.
3	1. Ввести послідовність символів, що містить текст. 2. Визначити довжину максимальної серії символів, відмінних від букв, яка входить в послідовність. 3. Вивести результати пошуку.
4	1. Ввести послідовність символів. 2. Замінити крапками всі двокрапки (:), що зустрічаються серед першої половини символів, 3. Замінити крапками всі знаки оклику, які зустрічаються серед символів, що стоять у другій половині початкової послідовності. 4. Вивести результати.
5	1. Ввести послідовність символів. 2. Вказати ті слова, які містять хоча б одну букву k.
6	1. Ввести послідовність символів, що містить текст. 2. У рядку між словами вставити замість пробілу кому і пробіл.
7	1. Ввести послідовність символів, що містить текст, який закінчується крапкою. 2. Вивести на екран слова, що містять три букви.
8	1. Ввести послідовність символів. 2. Перетворити її, видаливши кожен символ *, і повторивши кожен символ, відмінний від *.
9	1. Ввести послідовність символів, що містить текст. 2. Підрахувати кількість букв k в останньому слові послідовності.
10	1. Ввести послідовність символів. 2. Підрахувати, скільки різних символів зустрічається у ньому. 3. Вивести їх на екран.
11	1. Ввести послідовність символів. 2. Підрахувати найдовшу послідовність букв a, які йдуть підряд.
12	1. Ввести послідовність символів, серед яких є дужки, які відкриваються та закриваються. 2. Вивести на екран послідовності символів, що розташовані всередині цих дужок.
13	1. Ввести послідовність символів, що містить текст. 2. Визначити процентне відношення малих і великих літер до загальної кількості символів в ньому.
14	1. Ввести послідовність символів, серед яких є одна дужка, яка відкривається і одна дужка, що закривається. 2. Вивести на екран всі символи, розташовані всередині цих

№вар	Завдання
	дужок.
15	1.Ввести послідовність символів, що містить букви латинського алфавіту і цифри. 2.Вивести на екран довжину найбільшої послідовності цифр, що йдуть підряд.
16	1.Ввести послідовність символів з 11 елементів. 2.Прибрати зайві пробіли (Більше одного поспіль).
17	1.Введіть 5 послідовностей символів довжиною 8 елементів. 2.Розташувати послідовності в алфавітному порядку (як в словнику).
18	1.Ввести послідовність символів. 2.Визначити символ який найчастіше зустрічається і кількість його повторень.
19	1.Ввести послідовність символів. 2.Розташувати слова в алфавітному порядку за першою літерою слова.
20	1.Вивести слова в зворотному порядку, не використовуючи додаткову пам'ять.
21	1.Ввести послідовність символів, що містить пробіли. 2.Знайдіть найдовше слово, виведіть на екран це слово і його довжину.
22	1.Ввести послідовність символів, що містить дві однакові літери. 2.Визначте ці літери.
23	1.Введіть список символів з 7 елементів. 2.Визначте, чи є він симетричним (Симетричним вважається список, який однаково читається зліва направо і справа наліво).
24	1.Ввести дві послідовності символів. 2.Визначте однакові символи, які містять обидва списки і їх кількість.
25	1.Введіть послідовність символів з 17 елементів. 2.Визначте символ з найбільшим числом повторень.
26	1.Введіть послідовність символів з 11 елементів. 2.Обчислити добуток цілих чисел, які входять в нього (без урахування їх знаків).
27	1.Введіть послідовність символів. 2.Виведіть всі слова, що містить послідовність, як окремі списки символів (без пробілів)
28	1.Введіть послідовність символів та ціле число. 2.Виведіть слова з кількістю символів, яка відповідає введеному числу.
29	1.Введіть послідовність символів.

№вар	Завдання
	2.Виведіть всі слова, які починаються з даної літери.
30	1.Введіть послідовність символів і окреме слово. 2.Визначте, скільки разів задане слово зустрічається в даній послідовності.

Зміст звіту:

1. Титульний лист повинен мати такий вигляд:

Національний технічний університет України
«Київський політехнічний інститут»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Програмування
Лабораторна робота №3
«Робота з даними типу str, bytes та bytearray ».

Виконав:
студент групи ІО-XX
П.І.Б. студента
Номер у списку групи: XX
Перевірив:
(П.І.Б. викладача).

Київ 2023р

.

2. Мета лабораторної роботи та загальне завдання.
3. Короткі теоретичні відомості, які відображають операції, функції та методи обробки символьних даних, які були **використані** при написанні лабораторної роботи.
4. Скріншот вашого варіанту
5. Роздруківка того фрагменту тексту програми, який написаний індивідуально. Чорними символами на білому фоні.
6. Скріншоти результатів виконання програми з контрольним прикладом.
7. Аналіз результатів та висновки.

Контрольні питання (в звіті не відображаються)

1. Способи створення рядків.
2. Спеціальні символи, що застосовуються при обробці рядків.
3. Операції над рядками.
4. Математичні функції модуля Math
5. Оператори розгалуження та циклу.
6. Форматування рядків за допомогою оператора %
7. Метод форматування format()
8. Функції та методи для перетворень символьних типів даних

Лабораторна робота №4

Тема: «Списки, кортежі, множини і діапазони в Python».

Мета: вивчити способи створення списків, кортежів, множин та задавання діапазонів. Операції над списками, кортежами та діапазонами. Функції для перетворень списків, кортежів та множин.

Завдання:

1. Вивчити матеріал лекцій 11, 12, 13 та 14.
2. Виконати індивідуальне завдання лабораторної роботи, вибране відповідно до варіанту.

Теоретичні основи:

1. **Списки** є змінюваними типами даних.
2. **Кортежі** є незмінюваними типами даних.
3. **Множини** можуть бути як змінюваними, так і незмінюваними.
4. **Діапазони** є наборами чисел, сформованими на основі заданих початкового, кінцевого значень і величини кроку між числами.

Створити список можна такими способами:

1. За допомогою функції `list (<Послідовність>)`.
2. Перелічивши **всі елементи** списку всередині квадратних дужок.
3. Застосувавши метод `append()` для заповнення списку поелементно.
4. За допомогою генератора списків.

Створити копію списку можна такими способами:

1. **Перший спосіб** – застосуванням операції добування зрізу.
2. **Другий спосіб** – використанням функції `list()`,
3. **Третій спосіб** – застосуванням методу `copy()`.
4. Щоб одержати глибоку копію списку, слід скористатися функцією `deepcopy()` з модуля `copy`.

Операції над списками

1. Присвоювання значень елементам списку по індексу: $x[0] = 1$

(Ця операція не застосовна до рядка!)

2. Присвоювання з кінця за від'ємним індексом: $y[-1] = 2$

3. Присвоювання за допомогою зрізу:

Операція добування зрізу: `[<Початок>:<Кінець>:<Крок>]`

```
>>> arr = [2, 6, 10, 14, 18, 22]
>>> arr[1:3] = [8, 7]
>>> arr
[2, 8, 7, 14, 18, 22]
```

4. Оператори `+` та `+=`
5. Оператори `in` та `not in`

Багатовимірні списки

Будь-який елемент списку може містити список.

Створити вкладений список можна, наприклад, так:

Двовимірний список:

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

Тривимірний список:

```
>>> t = [[[1,2],[3,4]], [[5,6],[7,8]]]
>>> t
[[[1, 2], [3, 4]], [[5, 6], [7, 8]]]
>>> t[0][0][0]
1
>>> t[1][1][0]
7
```

Чотиривимірний список:

```
>>> f = [[[[1,2],[3,4]], [[5,6],[7,8]]],
          [[9,10],[11,12]], [[13,14],[15,16]]]
>>> f[0][0][0][0]
1
>>> f[0][1][1][0]
7
>>> f[1][1][1][1]
16
```

Перебір елементів списку

1. За допомогою циклу for: `for i in arr: print(i, end=" ")`
2. За допомогою циклу while.
3. За допомогою генераторів списків

```
>>> arr = [i for i in range(1,10)]
>>> arr
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Функція `map()`

Дозволяє застосувати функцію до кожного елемента послідовності

Повертає об'єкт, що підтримує ітерації. Щоб одержати список, необхідно результат передати у функцію `list()`.

```
map(<Функція>, <Послідовність1>[, ..., <Послідовність N>])
```

Функція `zip()`

Вбудована функція `zip()` на кожній ітерації повертає кортеж, що містить елементи послідовностей, які розташовані на однаковому зсуві.

`zip(<Послідовність1>[, ... , <ПослідовністьN>])`

Функція `filter()`

Дозволяє виконати перевірку елементів послідовності.

`filter(<Функція>, <Послідовність>)`

Функція повертає об'єкт, що підтримує ітерації. Щоб одержати список, необхідно результат передати у функцію `list()`.

Функція `reduce()`

Функція `reduce()` з модуля `functools` застосовує зазначену функцію до пар елементів і накопичує результат.

`reduce(<Функція>, <Послідовність>[, <Початкове значення>])`

Додавання й видалення елементів списку

`append(<Об'єкт>)` – додає один об'єкт у кінець списку.

`extend(<Послідовність>)` – додає елементи послідовності в кінець списку.

Оператор `+=` – додає елементи за допомогою операції конкатенації.

`insert(<Індекс>, <Об'єкт>)` – додає один об'єкт у зазначену позицію.

`pop([<Індекс>])` – видаляє елемент, розташований по зазначеному індексу, і повертає його.

`remove(<Значення>)` – видаляє перший елемент, який містить зазначене значення.

`clear()` – видаляє всі елементи списку, очищаючи його.

Методи

`index()` – повертає індекс елемента всередині списку.

`index(<Значення>[, <Початок>[, <Кінець>]])`

`count(<Значення>)` – повертає загальну кількість елементів

Функції

`max()` і `min()` – визначають максимальне й мінімальне значення списку відповідно.

`any (<Послідовність>)` – повертає значення `True`, якщо в послідовності існує хоч один елемент, який у логічному контексті повертає значення `True`.

`all (<Послідовність>)` – повертає значення `True`, якщо всі елементи послідовності в логічному контексті повертають значення `True` або послідовність не містить елементів.

Перевертання й перемішування списку

`reverse()` змінює порядок проходження елементів списку на протилежний.

`reversed(<Послідовність>)` – змінює порядок проходження та створює новий список.

`shuffle(<Список> [, <Число від 0.0 до 1.0>])` – з модуля `random` «перемішує» список випадковим чином.

Вибір елементів випадковим чином

Одержати елементи зі списку випадковим чином дозволяють наступні функції з модуля `random`:

`choice (<Послідовність>)` – повертає випадковий елемент із будь-якої послідовності.

`sample(<Послідовність>, <Кількість елементів>)` – повертає список із зазначеної кількості елементів.

Сортування списку

Відсортувати список дозволяє метод `sort()`:

```
sort([key=None] [, reverse=False])
```

Функція `sorted(<Послідовність>[, key=None] [, reverse=False])` – формує новий відсортований список, а початковий залишає без змін.

Заповнення списку числами:

```
>>> list(range(11))  
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Функція `sample(<Послідовність>, <Кількість елементів>)` з модуля `random`. Формує список з випадковими числами (або випадковими елементами з іншого списку)

Перетворення списку на рядок

Метод `join()`:

`<Рядок> = <Роздільник>.join(<Послідовність>)`

Кортежі

Створити кортеж можна такими способами:

1. за допомогою функції `tuple` (`[<Послідовність>]`)
2. Указавши всі елементи через кому всередині круглих дужок (або без дужок)

Кортежі підтримують функції:

`len()`, `min()`, `max()`, методи `index()` і `count()`

Множини

Оголосити множину можна за допомогою функції `set()`.

Одержати кількість елементів множини дозволяє функція `len()`.

Оператори `|` і `union()` – поєднують дві множини.

Оператори `a |= b` і `a.update(b)` – додають елементи множини `b` у множину `a`.

Оператори `-` і `difference()` – обчислюють різницю множин.

Оператори `a -= b` і `a.difference_update(b)` – видаляють елементи з множини `a`, які існують і в множині `a`, і в множині `b`.

Оператори `&` і `intersection()` – виконують перетин множин. Дозволяють одержати елементи, які існують в обох множинах.

Оператор `^` і `symmetric_difference()` – повертають усі елементи обох множин, крім елементів, які присутні в обох цих множинах.

Оператори `a ^= b` і `a.symmetric_difference_update(b)` – у множині `a` будуть усі елементи обох множин, крім тих, що містяться в обох цих множинах.

Оператор `in` – перевірка наявності елемента в множині.

Оператор `not in` – перевірка відсутності елемента в множині.

Оператор `==` – перевірка на рівність.

Оператор `a <= b` і `a.issubset(b)` – перевіряють, чи входять усі елементи множини `a` в множину `b`.

Оператор `a < b` – перевіряє, чи входять усі елементи множини `a` в множину `b`, причому множина `a` не повинна дорівнювати множині `b`.

Оператори `a >= b` і `a.issuperset(b)` – перевіряють, чи входять усі елементи множини `b` у множину `a`.

Оператор `a > b` – перевіряє, чи входять усі елементи множини `b` у множину `a`, причому множина `a` не повинна дорівнювати множині `b`.

Оператор `a.isdisjoint(b)` – перевіряє, чи є множини `a` й `b` повністю різними, тобто такими, що не містять жодного однакового елемента.

Методи

Метод `copy()` – створює копію множини.

Метод `add(<Елемент>)` – додає `<Елемент>` у множину.

Метод `remove(<Елемент>)` – видаляє `<Елемент>` із множини.

Метод `discard(<Елемент>)` – видаляє `<Елемент>` із множини, якщо він присутній.

Метод `pop()` – видаляє довільний елемент із множини й повертає його.

Метод `clear()` – видаляє всі елементи із множини.

Тип множин `frozenset`. На відміну від типу `set`, множину типу `frozenset` не можна змінити.

Оголосити множину можна за допомогою функції `frozenset()`.

Функція `frozenset()` дозволяє перетворити елементи послідовності в множину.

Множини `frozenset` підтримують оператори, які не змінюють саму множину, а також наступні методи:

```
copy(), difference(), intersection(), issubset(),  
issuperset(), symmetric_difference() та union().
```

Діапазони

Для створення діапазону застосовується функція `range()`:

```
range([<Початок>], <Кінець> [, <Крок>])
```

Діапазони підтримують:

- доступ до елемента по індексу,
- одержання зрізу (у результаті повертається також діапазон),
- перевірку на входження
- перевірку на невходження,
- функції `len()`, `min()`, `max()`,
- методи `index()` і `count()`.

Оператор `==` – повертає `True`, якщо діапазони рівні, і `False` в протилежному випадку.

Оператор `!=` – повертає `True`, якщо діапазони не рівні, і `False` в протилежному випадку.

Модуль `itertools`

Перш ніж використовувати функції, необхідно підключити модуль за допомогою інструкції: `import itertools`

1. Функція `count([start=0][, step=1])`.

Створює нескінченно наростаючу послідовність значень.

2. Функція `cycle(<Послідовність>)`

На кожній ітерації повертається черговий елемент послідовності.

3. Функція `repeat(<Об'єкт>[, <Кількість повторів>])`

Повертає об'єкт зазначену кількість разів.

Основні поняття про матрицю

Нехай є дві скінченні множини:

$$M = \{1, 2, \dots, m\} \text{ і } N = \{1, 2, \dots, n\}$$

де m і n — натуральні числа.

Розглянемо таблицю, яка складається з m рядків та n стовпців.

Кожен елемент цієї таблиці характеризується двома індексами: i та j , які визначають положення даного елемента в таблиці. Будемо позначати через a_{ij} елемент, який знаходиться на перетині i -го рядка та j -го стовпця. Далі будемо називати елемент a_{ij} елементом матриці, а саму таблицю – матрицею A .

Отже, матриця $m \times n$ має такий вигляд:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{pmatrix}$$

В мові програмування Python матриця A може бути представлена двовимірним списком або кортежем. Оскільки [нумерація елементів у списках починається з нуля](#), то індекси елементів матриці A матимуть такий вигляд:

$$A_{\text{Python}} = \begin{pmatrix} a_{00} & a_{01} & \dots & a_{0n-1} \\ a_{10} & a_{11} & \dots & a_{1n-1} \\ \dots & \dots & \dots & \dots \\ a_{m-1,0} & a_{m-1,1} & \dots & a_{m-1,n-1} \end{pmatrix}$$

Отже, для задавання матриці $A(m, n)$ розмірності $m \times n$ потрібно створити список з діапазоном індексів $(0 \dots (m-1))$, кожен елемент якого є списком, що містить елементи з діапазоном індексів $(0 \dots (n-1))$

Приклад, що буде корисний при виконанні лабораторної роботи:

Спосіб "виокремлення" цифр з числа, якщо відома кількість розрядів (цифр) числа.

1.Залишок від ділення на 10 початкового числа дає останню цифру числа.

2.Якщо знайти залишок від ділення на 100 початкового числа, то ми отримаємо останні дві цифри числа. Далі можна просто розділити отримане двозначне число без остачі на 10, і у нас виявиться друга цифра числа.

3.Якщо початкове тризначне число розділити без остачі на 100, то отримаємо першу цифру числа.

```
n = input("Введіть тризначне число: ")
n = int(n)

d1 = n % 10
d2 = n % 100 // 10
d3 = n // 100

print("Сума цифр числа:", d1 + d2 + d3)
```

Результат роботи скрипта:

```
Введіть тризначне число: 742
Сума цифр числа: 13
```

Індивідуальні завдання

Завдання 1

Відповідно до номера в списку вибрати індивідуальне завдання. Написати програму мовою Python. Забезпечити ввід даних з клавіатури комп'ютера та друк результатів обчислень. У звіті до лабораторної роботи описати алгоритм, за яким побудована програма. **При виводі даних обов'язково використати форматування.**

№	Завдання
1	Випадковим чином створити два цілочисельних списки. Сформувати новий список, на парних місцях якого будуть елементи з непарними індексами з першого списку, а на непарних – з парними індексами з другого. Сформувати з нового списку вдвічі коротший список з випадковими елементами. Перетворити цей список на рядок.
2	Випадковим чином створити список, що складається з елементів цілого типу. Отримати новий список, який складається з цифр, що знаходяться в молодших розрядах елементів початкового списку. Сформувати з нього список з випадковими елементами, вдвічі довший за початковий.

№	Завдання
3	Випадковим чином створити цілочисельний список, що складається з двозначних цілих чисел. Обчислити суму чисел цього списку. Суму чисел списку вставити в середину початкового списку, якщо його довжина – парне число, і в кінець, якщо його довжина непарна.
4	Випадковим чином створити два списки дійсних чисел. Сформувати третій список з упорядкованих за зростанням значень елементів обох списків. Визначити, чи є серед елементів елемент зі значенням 0.
5	Випадковим чином створити два списки X і Y, що складаються з елементів цілого типу. Сформувати список S, що складається з однакових елементів початкових списків. Визначити довжину отриманого списку, вставити значення в кінець списку.
6	Розрахувати значення елементів списку Y за формулою $y_i = i^2 - 2i + 19.3 \cos i$. Сформувати новий список, розмістивши в ньому спочатку елементи, значення яких менші за середнє арифметичне, а потім – ті значення, що дорівнюють або більші за середнє арифметичне елементів початкового списку. Вивести обидва списки.
7	Випадковим чином створити список дійсних чисел. Визначити різницю між сумою елементів з парними індексами і сумою елементів, індекси яких кратні трьом. Вставити цю різницю в кінець списку.
8	Випадковим чином створити цілочисельний список. Визначте індекси найбільшого з непарних за значенням додатних елементів та найбільшого з парних за значенням додатних елементів. Знайдіть індекс елемента, найближчого за значенням до різниці двох попередніх елементів. Видаліть цей елемент, а також сформууйте новий список з тих елементів, що у сумі з виведеним перевищують число 10.
9	Випадковим чином створити список $X = [x_1, x_2, \dots]$, який складається з елементів цілого типу. Розрахувати елементи списку $Y = [y_1, y_2, \dots]$ за формулою $y_i = \cos x_i^2 + 2.971 \lg^2 i^2$. Сформувати третій список з упорядкованих за спаданням значень з обох списків. Вибрати із сформованого списку випадковим чином два елементи і визначити, чи входять вони у список X.
10	Випадковим чином створити список $X = [x_1, x_2, \dots]$, що складається з елементів цілого типу. Розрахувати елементи списку $Y = [y_1, y_2, \dots]$ за формулою: $y_i = \begin{cases} (x_i)^3 - 7.5 & \text{якщо } \cos(x_i) > 0, \\ x_i^2 - 5e^{\sin(x_i)} & \text{якщо } \cos(x_i) \leq 0. \end{cases}$ Впорядкувати список Y за зростанням, список X за спаданням і сформувати новий список R, елементами якого є парні за індексами елементи списків X і Y. Вибрати зі списку R випадковим чином один елемент і визначити, чи входить він у список Y.
11	Випадковим чином створити список, що складається з двозначних

№	Завдання
	чисел цілого типу. Отримати новий список, що складається з сум цифр відповідних елементів початкового списку. Створити список, елементами якого є кортежі, що складаються з елементів, які знаходяться на однаковому зсуві з елементами початкових списків.
12	Випадковим чином створити три списки, що складаються 1) з довільних додатних елементів дійсного типу, 2) з довільних елементів цілого типу, 3) з елементів логічного типу. Створити список, що утворюється додаванням елементів першого та другого списків, які знаходяться на однаковому зсуві, якщо елемент третього списку, який знаходиться на тому ж зсуві, дорівнює <code>True</code> , та відніманням у протилежному випадку.
13	Випадковим чином створити цілочисельний список. Обчислити суму цифр елементів. Якщо сума цифр елемента виявиться непарною, то замінити нею дане число. Сформувати з отриманого списку список з випадковими елементами, вдвічі коротший за початковий.
14	Випадковим чином створити три списки, що складаються з довільних елементів дійсного типу. Замінити місцями першу половину першого списку на другу симетрично. Створити список, що утворюється добутком суми елементів першого та другого списків, які знаходяться на однаковому зсуві, на елемент третього списку, який знаходиться на тому ж зсуві.
15	Випадковим чином створити список дійсних чисел. Визначити елемент списку (значення і індекс), який найбільш віддалений від заданого дійсного числа S . Перемішати список випадковим чином, визначити елемент списку (значення і індекс), який найбільш віддалений від цього ж заданого дійсного числа S .
16	Випадковим чином створити цілочисельний список. Визначити суму і кількість елементів, величина яких є меншою за середнє арифметичне елементів списку. Дописати значення суми і кількість елементів як елементи в кінець початкового списку та сформувати третій список з тих елементів, які дорівнюють середньому арифметичному елементів одержаного списку та початкового списку.
17	Визначити кількість локальних мінімумів у випадковим чином створеному списку чисел. (Локальний мінімум в списку чисел – це послідовність трьох чисел, що стоять поруч та в яких середнє число менше за тих, що стоять зліва і справа від нього). Перемішати список випадковим чином, визначити кількість локальних мінімумів в отриманому списку чисел. Якщо кількість локальних мінімумів у першому списку більша за кількість локальних мінімумів у другому списку, то сформувати список з елементами, що містять кортежі з елементами першого та другого списку, які знаходяться на однаковому зсуві. У протилежному випадку сформувати третій список з сумами елементів першого та другого списку.

№	Завдання
18	<p>Випадковим чином створити три цілочисельних списки. Створити список, що утворюється піднесенням до степеня різниці елементів першого та другого списків, які знаходяться на однаковому зсуві. Значення степеня дорівнює елементу третього списку, який знаходиться на тому ж зсуві.</p> <p>(Наприклад: перший список: 8, 9. Другий список: 10, 5. Третій список: 2, 3. Створений список: перший елемент: $(8-10)^{**2}$, другий елемент: $(9-5)^{**3}$)</p>
19	<p>Створити випадковим чином список з додатних, від'ємних і нульових чисел. Визначити суму елементів. Якщо одержана сума буде додатною, то вивести послідовність значень елементів, які розташовані між першим від'ємним і першим нульовим елементами. Якщо одержана сума буде від'ємною, то сформуванати третій список, кожен елемент якого буде дорівнювати середньому арифметичному від елементів створеного списку.</p>
20	<p>У створеному випадковим чином списку чисел визначити послідовності чисел, які монотонно зменшуються (кожне наступне число менше за попереднє). Створити списки, елементами якого є кортежі, кожен з яких містить індекс та значення відповідного елемента. Якщо сума елементів послідовності виявиться парною, то для кожної з них створити третій список, що містить як елементи суму елементів одержаних кортежів. В протилежному випадку суму замінити різницею.</p>
21	<p>У створеному випадковим чином списку цілих чисел видалити ті елементи, які зустрічаються більше, ніж двічі. Створити списки, відсортовані за зростанням та за спаданням. З даних списків сформуванати третій список, кожен елемент якого дорівнює середньому геометричному двох попередніх елементів, які знаходяться на однаковому зсуві.</p>
22	<p>Сформуванати випадковим чином список, що складається з елементів цілого типу та сформуванати новий список, розташувавши спочатку всі від'ємні елементи, нульові, а далі – додатні, зберігаючи порядок їх слідування в початковому списку. Сформуванати кортеж, що містить як елементи кількість додатних чисел, кількість нулів та кількість від'ємних чисел. Якщо сума першого та другого елементу цього кортежу буде парною, то сформуванати третій список з подвоєними елементами початкового списку. У протилежному випадку третій список повинен містити елементи, які є остачами від ділення першого списку на 2.</p>
23	<p>Сформуванати випадковим чином список X, що складається з</p>

№	Завдання
	<p>елементів цілого типу. Обчислити елементи списку Y за формулою $y_i = x_i^2 + 0.3$ і знайти P за формулою</p> $P = \frac{x_1 y_1 \cdot x_3 y_3 \cdot \dots \cdot x_9 y_9}{x_0 y_0 \cdot x_2 y_2 \cdot \dots \cdot x_8 y_8}.$ <p>Визначити залишок від ділення. Записати його після кожного непарного числа у списку X.</p>
24	Випадковим чином сформувати список, що складається з двозначних чисел цілого типу. Отримати новий список, який складається з різниць цифр елементів початкового списку. Випадковим чином вибрати елемент з новоутвореного списку та видалити перший елемент списку, який містить це значення.
25	Випадковим чином сформувати список, що складається з елементів цілого типу. Впорядкувати список так, щоб всі від'ємні числа були розташовані з початку за зростанням, а всі додатні – в кінці за спаданням. Після ділянки з від'ємними числами вставити як елемент списку їх кількість.
26	Випадковим чином сформувати два списки дійсних чисел. Замінити нулями ті елементи першого списку, які є у другому. Створити список, елементами якого є кортежі, що складаються з елементів цих двох списків, які знаходяться на однаковому зсуві.
27	Випадковим чином заданий цілочисельний список. Визначити кількість ділянок списку, на яких елементи монотонно зростають (кожне наступне число більше за попереднє). Після кожної з ділянок вставити як елемент списку кількість елементів цієї ділянки та знайти суму елементів одержаного списку.
28	Випадковим чином створити два цілочисельні списки. Створити список, елементами якого є залишки від ділення елементів з парними індексами одного списку на елементи з непарними індексами другого. Впорядкувати його за зростанням та визначити суму елементів, додавши її в кінці списку.
29	Випадковим чином створити цілочисельний список. Визначити процентний вміст елементів, що перевищують середнє значення суми всіх елементів списку. Вставити це значення в кінець списку.
30	Випадковим чином створити два списки дійсних чисел. Визначити максимальні елементи в кожному списку і поміняти їх місцями.

Завдання 2

Відповідно до номера в списку вибрати індивідуальне завдання. Написати програму мовою Python. Забезпечити ввід даних з клавіатури комп'ютера та друк результатів обчислень. У звіті до лабораторної роботи описати алгоритм, за яким побудована програма. **При виводі даних обов'язково використати форматування.**

№вар	Завдання
1	Згенерувати матрицю $A(m,n)$ розмірністю $m \times n$ з випадкових елементів, що є цілими числами від 0 до 10. Видалити рядки, які містять елементи, що повторюються.
2	Згенерувати матрицю $A(m,n)$ розмірністю $m \times n$ з випадкових елементів, що є від'ємними та додатними цілими числами. Побудувати список, елементами якого є сума модулів від'ємних чисел у кожному рядку матриці $A(m,n)$
3	Згенерувати матрицю $A(m,n)$ розмірністю $m \times n$ з випадкових елементів, що є цілими числами. Побудувати список, елементами якого є середнє арифметичне рядків матриці $A(m,n)$
4	Згенерувати матрицю $A(m,n)$ розмірністю $m \times n$ з випадкових елементів, що є цілими числами. Знайти мінімальний елемент на кожній з діагоналей матриці.
5	Згенерувати матрицю $A(m,n)$ розмірністю $m \times n$ з випадкових елементів, що є цілими числами. Відсортувати рядки з непарним індексом за зростанням, а з парним індексом – за спаданням.
6	Згенерувати матрицю $A(m,n)$ розмірністю $m \times n$ з випадкових елементів, що є цілими числами. Знайти мінімальні елементи в кожному стовпці та рядку матриці.
7	Згенерувати матрицю $A(m,n)$ розмірністю $m \times n$ з випадкових елементів, що є цілими числами. Поміняти місцями мінімальний та максимальний елементи у кожному стовпці матриці
8	Згенерувати матрицю $A(m,n)$ розмірністю $m \times n$ з випадкових елементів, що є дійсними числами. Знайти спочатку мінімальні елементи в кожному рядку матриці, а потім обчислити їх добуток.
9	Згенерувати матрицю $A(m,n)$ розмірністю $m \times n$ з випадкових елементів, що є дійсними числами. Визначити середнє арифметичне кожного стовпця, визначити максимум і мінімум кожного рядка.
10	Згенерувати матрицю $A(m,n)$ розмірністю $m \times n$ з випадкових елементів, що є цілими числами. Визначити кількість непарних елементів кожного стовпця.
11	Згенерувати матрицю $A(m,n)$ розмірністю $m \times n$ з випадкових елементів, що є цілими числами. Знайти кількість парних і непарних чисел у матриці.
12	Згенерувати матрицю $A(m,n)$ розмірністю $m \times n$ з випадкових елементів, що є цілими числами. Підібрати спільний дільник для

№вар	Завдання
	всіх елементів матриці таким чином, щоб одержати принаймні дві остачі від ділення, які дорівнюють 2.
13	Згенерувати матрицю $A(m,n)$ розмірністю $m \times n$ з випадкових елементів, що є цілими числами. Знайти рядок матриці, у якому середнє арифметичне є найбільшим.
14	Згенерувати матрицю $A(m,n)$ розмірністю $m \times n$ з випадкових елементів, що є цілими числами. Сформувати нову матрицю, кожен елемент якої є середнім арифметичним даного та попереднього рядка. Для першого рядка одержати середнє арифметичне з останнім.
15	Згенерувати квадратну матрицю $A(m,m)$ розмірністю $m \times m$ з випадкових елементів, що є цілими числами. Сформувати нову матрицю, рядками якої будуть упорядковані за зростанням стовпці початкової матриці.
16	Згенерувати квадратну матрицю $A(m,m)$ розмірністю $m \times m$ з випадкових елементів, що є цілими числами. Сформувати нову матрицю, стовпцями якої будуть упорядковані за спаданням рядки початкової матриці.
17	Згенерувати квадратну матрицю $A(m,m)$ розмірністю $m \times m$ з випадкових елементів, що є цілими числами. Поміняти місцями елементи, які стоять на діагоналях на відповідних місцях.
18	Згенерувати квадратну матрицю $A(m,m)$ розмірністю $m \times m$ з випадкових елементів, що є додатними та від'ємними цілими числами. Поміняти знаки чисел у кожному рядку та сформувати нову матрицю, у якій кожний стовець відповідає рядку початкової матриці після зміни знаків.
19	Згенерувати квадратну матрицю $A(m,m)$ розмірністю $m \times m$ з випадкових елементів, що є цілими числами. Знайти рядок з максимальним елементом та стовець з мінімальним елементом. Поміняти місцями знайдений рядок зі знайденим стовпцем.
20	Згенерувати матрицю $A(m,m)$ розмірністю з випадкових елементів, що є цілими числами. Замінити максимальний елемент у кожному рядку на середнє арифметичне стовпця, який містить даний максимальний елемент.
21	Згенерувати матрицю $A(m,n)$ розмірністю $m \times n$ з випадкових елементів, що є цілими числами. Замінити мінімальний елемент у кожному стовпці на середнє арифметичне рядка, який містить даний мінімальний елемент.

№вар	Завдання
22	Згенерувати матрицю $A(m,n)$ розмірністю $m \times n$ з випадкових елементів, що є цілими числами. Знайти мінімальне значення серед сум елементів рядків матриці і замінити ним максимальний елемент у кожному рядку.
23	Згенерувати матрицю $A(m,n)$ розмірністю $m \times n$ з випадкових елементів, що є цілими числами. Порахувати кількість елементів у кожному рядку матриці та кількість елементів у всій матриці, які перевищують середнє арифметичне матриці.
24	Згенерувати матрицю $A(m,n)$ розмірністю $m \times n$ та матрицю $B(n,m)$ розмірністю $n \times m$ з випадкових елементів, що є цілими числами. Сформувати матрицю $C(m,n)$, кожен елемент якої є остачею від ділення розміщених на відповідних позиціях елементів рядка матриці A на елементи стовпця матриці B
25	Згенерувати матриці $A(m,n)$ та $B(m,n)$ розмірністю $m \times n$. Визначити середнє арифметичне для кожного стовпця матриці A та поділити на нього елементи відповідного рядка матриці B
26	Згенерувати матрицю $A(m,n)$ розмірністю $m \times n$ з випадкових елементів, що є цілими двохранрядними числами. Сформувати нову матрицю, кожен елемент якої містить число, яке є остачею від ділення елемента матриці A на суму його цифр.
27	Згенерувати матрицю $A(m,m)$ розмірністю $m \times m$ з випадкових елементів, що є цілими двохранрядними числами. Знайти максимальний елемент на головній діагоналі та поділити на цей елемент всі елементи матриці, крім нього самого.
28	Згенерувати матрицю $A(m,n)$ розмірністю $m \times n$ з випадкових елементів, що є цілими двохранрядними числами. Знайти суму непарних елементів у кожному рядку і поділити її на кожен з парних елементів у цьому ж рядку.
29	Згенерувати матрицю $A(m,n)$ розмірністю $m \times n$ з випадкових елементів, що є цілими числами. Знайти суму елементів матриці і створити нову матрицю, кожен елемент якої дорівнює відсотку, який складає відповідний елемент матриці A по відношенню до суми її елементів.
30	Згенерувати матриці $A(m,m)$ та $B(m,m)$ розмірністю $m \times m$. Сформувати матрицю C , кожен елемент якої дорівнює середньому арифметичному між відповідними елементами стовпця матриці A та рядка матриці B .

Зміст звіту:

1. Титульний лист повинен мати такий вигляд:

Національний технічний університет України
«Київський політехнічний інститут»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Програмування
Лабораторна робота №4
«Списки, кортежі, множини і діапазони в Python»

Виконав:
студент групи ІО-ХХ
П.І.Б. студента
Залікова книжка № _____
Перевірів (П.І.Б. викладача)

Київ 2023р.

2. Мета лабораторної роботи та загальне завдання
3. Короткі теоретичні відомості, які відображають операції, функції та методи обробки даних, які **були використані** при написанні лабораторної роботи.
4. Скріншот вашого варіанту
5. Роздруківка того фрагменту тексту програми, який написаний індивідуально чорними символами на білому фоні.
6. Скріншот результатів виконання програми з контрольним прикладом.
7. Аналіз результатів та висновки.

Контрольні питання (не відображаються у звіті)

1. Створення списків.
2. Операції над списками
3. Багатовимірні списки
4. Перебір елементів списку
5. Додавання й видалення елементів списку
6. Перевертання й перемішування списку
7. Вибір елементів випадковим чином
8. КORTEжі
9. Операції над множинами
10. Діапазони

Лабораторна робота №5

Тема: «Словники та функції користувача».

Мета: вивчити способи створення словників та функцій користувача. Операції над словниками. Методи для роботи зі словниками. Генератори словників. Функції користувача, особливості їх створення та використання.

Завдання:

3. Вивчити матеріал лекцій 15, 16 та 17.
4. Виконати індивідуальне завдання лабораторної роботи, вибране відповідно до варіанту.

Теоретичні основи:

Створити словник можна такими способами:

Спосіб 1. За допомогою функції `dict()`.

Спосіб 2. Указавши всі елементи словника усередині фігурних дужок.

Спосіб 3. Заповнивши словник поелементно.

Спосіб 4. За допомогою методу

`dict.fromkeys(<Послідовність>[, <Значення>])`

Операції над словниками

Перевірити існування ключа можна за допомогою оператора `in`.

Перевірити, чи відсутній який-небудь ключ у словнику, дозволить оператор `not in`.

Метод `get(<Ключ>[, <Значення за замовчуванням>])`

дозволяє уникнути порушення виключення `KeyError` за відсутності в словнику зазначеного ключа.

Метод `setdefault(<Ключ>[, <Значення за замовчуванням>])`

Якщо ключ присутній у словнику, то метод повертає значення, відповідне до цього ключа. Якщо ключ відсутній, то в словнику створюється новий елемент зі значенням, зазначеним у другому параметрі. Якщо другий параметр не зазначений, значенням нового елемента буде `None`.

Одержати кількість ключів у словнику дозволяє функція `len()`:

Вилучити елемент зі словника можна за допомогою оператора `del`:

Перебрати всі елементи словника можна за допомогою циклу `for`, хоча словники й не є послідовностями.

Щоб вивести елементи з сортуванням по ключах, слід одержати список ключів, а потім скористатися методом `sort()`.

Для сортування ключів замість методу `sort()` можна скористатися функцією `sorted()`

Методи для роботи зі словниками

Метод `keys()` – повертає об'єкт `dict_keys`, що містить усі ключі словника.

Метод `values()` – повертає об'єкт `dict_values`, що містить усі значення словника. Цей об'єкт підтримує ітерації.

Метод `items()` – повертає об'єкт `dict_items`, що містить усі ключі й значення у вигляді кортежів. Цей об'єкт підтримує ітерації.

Метод `<Ключ> in <Словник>` – перевіряє існування зазначеного ключа в словнику. Якщо ключ знайдений, то повертається значення `True`, а якщо ні, то – `False`.

Метод `<Ключ> not in <Словник>` – перевіряє відсутність зазначеного ключа в словнику. Якщо такого ключа немає, то повертається значення `True`, а якщо ні, то – `False`.

Метод `get (<Ключ> [, <Значення за замовчуванням>])` – якщо ключ присутній у словнику, то метод повертає значення, відповідне до цього ключа. Якщо ключ відсутній, то повертається `None` або значення, зазначене в другому параметрі.

Метод `setdefault (<Ключ> [, <Значення за замовчуванням>])` – якщо ключ присутній у словнику, то метод повертає значення, відповідне до цього ключа.

Метод `pop(<Ключ>[, <Значення за замовчуванням>])` – видаляє елемент із зазначеним ключем і повертає його значення.

Метод `popitem()` – видаляє довільний елемент і повертає кортеж із ключа й значення.

Метод `clear()` – видаляє всі елементи словника. Метод нічого не повертає як значення.

Метод `update ()` – додає елементи в словник. Метод змінює поточний словник і нічого не повертає.

Формати методу:

```
update(<Ключ1>=<Значення1>[, ... , <Ключn>=<Значенняn>])
update(<Словник>)
update(<Список кортежів з двома елементами>)
update(<Список списків з двома елементами>)
```

Метод `copy()` – створює поверхневу копію словника.

Щоб створити глибоку копію словника, слід скористатися функцією `deepcopy()` з модуля `copy`.

Синтаксис **генераторів словників** схожий на синтаксис генераторів списків, але має дві відмінності:

1. Вираз міститься у фігурних дужках, а не у квадратних;
2. Всередині виразу перед циклом `for` вказуються два значення через двокрапку, а не одне. Значення, розташоване ліворуч від двокрапки, стає ключем, а значення, розташоване праворуч від двокрапки, – значенням елемента.

Функція створюється (або, як говорять програмісти, визначається) за допомогою ключового слова `def` за наступною схемою:

```
def <Ім'я функції> ([<Параметри>]):  
    [""" Рядок документування """]  
    <Тіло функції>  
    [return <Результат>]
```

Розташування визначень функцій

Усі інструкції в програмі виконують послідовно зверху вниз. Це означає, що, перш ніж використовувати в програмі ідентифікатор, його необхідно попередньо визначити, присвоївши йому значення. Тому **визначення функції повинно бути розташоване перед викликом функції**.

Щоб зробити деякі параметри необов'язковими, слід у визначенні функції **присвоїти цьому параметру початкове значення**.

Якщо значення параметрів утримуються в словнику, то розпакувати словник можна, указавши перед ним дві зірочки: `* *)`.

Якщо перед параметром у визначенні функції вказати символ `(*)`, то функції можна буде передати довільну кількість параметрів. Усі передані параметри зберігаються в кортежі.

Анонімну функцію описують за допомогою ключового слова `lambda` за наступною схемою:

```
lambda    [<Параметр1>[, ..., <Параметрn>]] :<значення,    що повертається, >
```

Функцією-генератором називають функцію, яка може повертати одне значення з декількох значень на кожній ітерації. Призупинити виконання функції й перетворити функцію в генератор дозволяє ключове слово `yield` (вироблення продукції).

Існує можливість викликати одну функцію-генератор з іншої. Для цього застосовується розширений синтаксис ключового слова `yield`:

`yield from` <Викликувана функція-генератор>

Декоратори дозволяють змінити поведінку звичайних функцій – наприклад, виконати які-небудь дії перед виконанням функції.

Рекурсія – це можливість функції викликати саму себе. Рекурсію зручно використовувати для перебору об'єкта, що має заздалегідь невідому структуру, або для виконання невизначеної кількості операцій.

Глобальні змінні – це змінні, оголошені в програмі поза функцією

Отже, *локальні змінні* – це змінні, що оголошуються усередині функцій.

Одну функцію можна вкласти в іншу функцію, причому рівень вкладеності не обмежений. При цьому вкладена функція одержує свою власну локальну область видимості, має доступ до змінних, оголошених всередині функції, у яку вона вкладена (функції-предка).

Тепер у заголовку функції допускається вказувати призначення кожного параметра, дані якого типу він може приймати, а також тип значення, що повертається функцією. Анотації мають наступний формат:

```
def <Ім'я функції>(  
[<Параметр1>[: <Вираз>] [= <Значення за замовчуванням>]  
[,...<Параметрn>[: <Вираз>] [= <Значення за  
замовчуванням>]]]  
) -> < значення, що повертається>:  
<Тіло функції>
```

Індивідуальні завдання

Завдання

Відповідно до номера в списку вибрати індивідуальне завдання. Написати програму мовою Python. Забезпечити ввід даних з клавіатури комп'ютера та друк результатів обчислень. У звіті до лабораторної роботи описати алгоритм, за яким побудована програма. **При виводі даних обов'язково використати форматування.**

	Завдання
1	Використовуючи словник, задайте меню, вказуючи як ключ назву страви, а як значення – її ціну. Виведіть меню, відсортоване за ціною страв та за назвами страв за алфавітом.
2	Задайте словник результатів тестування, вказуючи як ключ прізвище учасника, а як значення – кількість балів. Кількість балів згенерувати випадковим чином. Максимальна кількість балів – 100. Визначити трьох переможців та передбачити відповідь на запитання про наявність людини серед учасників.
3	Задайте словник, вказуючи як ключ місяць року, а як значення – кількість сонячних днів, які задані випадковим чином. Виведіть місяць, який має середню кількість сонячних днів, місяць, що має найбільшу кількість сонячних днів, та місяць з найменшою кількістю сонячних днів.
4	Задайте словник, вказуючи як ключ назву підручника, а як значення – кількість сторінок, які задані випадковим чином. Знайти найдорожчий

	Завдання
	підручник, якщо ціна друку однієї сторінки однакова у всіх підручниках і складає 50 коп.
5	Задайте словник, вказуючи як ключ назву міста, а як значення – кількість мешканців. Виведіть міста, упорядковані за спаданням кількості мешканців та у алфавітному порядку за назвою.
6	Задайте словник континентів нашої планети, задаючи як значення площу кожного континенту. Виведіть континент, площа якого є найближчою до середнього арифметичного всіх площ континентів.
7	Задайте словник країн, вказавши як значення чисельність населення. Передбачити ввід назви країни та чисельності її населення, якщо країни немає в словнику. Виведіть країни за спаданням чисельності населення.
8	Введіть словник планет Сонячної системи. Як значення задайте кортеж, що містить відстань до Сонця, об'єм та орбітальну швидкість. Виведіть три списки планет, які упорядковані за відстанню від сонця, об'ємом та орбітальною швидкістю.
9	Введіть словник факультетів КПІ. Як значення задайте кортеж з прізвища декана та номера телефону. Виведіть список факультетів, які упорядковані в алфавітному порядку за прізвищами деканів та факультет, телефонний номер якого найближчий до середнього арифметичного всіх номерів.
10	Введіть словник країн. Як значення задайте кортеж, що містить площу та чисельність населення. Виведіть список країн, які упорядковані за густиною населення.
11	Введіть словник букв кириличного алфавіту. Для кожної букви задайте кортеж слів, які починаються з цієї букви. Оформити вивід даних у форматі книги-словника, використовуючи функцію <code>format()</code> .
12	Створити словник, у якому в якості ключів використати пункти каталогу товарів Інтернет-магазину « rozetka.com.ua ». Значеннями для даного словника будуть словники з категоріями товару з даного пункту і значеннями, які представлені у вигляді кортежів з назви товару, його коду та ціни. Програма повинна виводити категорії товарів після введення пункту каталогу, виводити товари після запиту категорії, виводити ціну та назву після введення коду.
13	Написати програму вводу англо-українського словника з можливістю вводу кількох варіантів перекладу. Програма також повинна виводити український переклад введеного англійського слова.
14	Написати програму, яка буде використовувати як ключ слова української мови. Значення словника – кортеж синонімів. Виводити на друк синоніми до заданого набору слів. Якщо словник програми не містить синоніму до введеного слова, то програма повинна сповістити про це користувача та запросити ввід синоніму до нового слова.

	Завдання
15	Введіть словник міст. Як значення задайте кортеж, що містить їх географічні координати. Програма повинна виводити список трьох міст, довгота або широта яких найближча до введених.
16	Введіть словник столиць. Як значення задайте чисельність їх населення. Програма повинна виводити список міст, чисельність населення яких відповідає введеному діапазону.
17	Введіть список марок автомобілів. Як значення задайте кортеж, який містить ціну, рік випуску та пробіг. Програма повинна виводити список автомобілів заданої марки та року випуску з заданими діапазонами ціни та пробігу.
18	Створіть словник, який описує бібліотеку. В якості ключів використайте жанри книг, наприклад: пригоди, фантастика, історичний роман і т.д. Як значення задайте кортеж, який містить прізвище автора, його ім'я, назву книги та видавництво. Програма повинна за запитом виводити книги, що написав відповідний автор, книги за жанром, за діапазоном сторінок, за видавництвом.
19	Створити словник предметів, які ви вивчаєте. В якості значень задати кортежі, які містять як елементи прізвище викладача, кількість годин на тиждень, та наявність екзамену, диференційованого заліку або простого заліку. Програма повинна виводити після введення предмету прізвище викладача та наявність екзамену або заліку. Також за запитом необхідно виводити список предметів, які мають однакову кількість годин на тиждень.
20	Введіть слова українською мовою, як ключі словника. Як значення для кожного ключа задайте кортеж синонімів українською мовою та англійською мовою. Напишіть програму інтерактивного словника, яка видає англійські переклади або українські синоніми у відповідь на введене слово.
21	Створіть словник, ключами якого є розділи Вікіпедії: Економіка, Історія, Кінематографія, Література, Міста, Міфологія, Мова, Персоналії, Політика, Традиції. Відповідними значеннями є словники з ключами, що відображають підкатегорії, та значеннями у вигляді кортежів, які містять прізвища відомих людей у даній сфері діяльності. Програма повинна виводити підкатегорії при введенні назви розділу та прізвища людей при введенні підкатегорії.
22	Виведіть на екран список міністерств України та їх скорочені назви. Створіть словник, ключами якого є скорочені назви міністерств. В якості значення використайте кортеж, який складається з контактного телефону, електронної адреси та прізвища міністра. Програма повинна виводити назву міністерства після введення прізвища міністра, телефон і email після введення назви міністерства.
23	Створити словник з ключами – українськими літерами та значеннями – словниками. Кожен такий словник містить іншомовні слова як ключі та

	Завдання
	їх синоніми як значення. Програма повинна виводити всі іншомовні слова, що починаються з введеної букви, та виводити всі синоніми при введенні іншомовного слова.
24	Виведіть на екран список управлінь КМДА та їх скорочених назв. Створіть словник, у якому як ключ використовується скорочена назва управління КМДА. В якості значення задати прізвище керівника та контактний телефон. Програма повинна виводити на друк прізвище керівника за скороченою назвою управління та повну назву управління за прізвищем керівника.
25	Створіть словник, ключами якого є міста України, у яких існують філіали магазину "WINTIME". Як значення запишіть кортеж, який містить місцеву адресу та телефон. Програма повинна виводити на друк всі міста з філіалами магазину, які починаються з введеної букви, та виводити адресу і телефон філіалу магазину, який розміщений у місті, що вводиться у запиті.
26	Створити словник з ключами – назвами галактик. В якості значень записати інформацію про назви галактик в різних каталогах та координати розташування. Програма повинна виводити всі інші синоніми назв галактик при вводі одного з можливих синонімів. Вивести координати галактики при вводі основної назви.
27	Створити словник з ключами – назвами континентів нашої планети. Значеннями цього словника будуть словники з ключами-назвами країн, які розміщені на даному континенті, і значеннями-кортежами. Елементами таких кортежів повинні бути назви столиць відповідних країн, їх площа та чисельність населення. Програма повинна виводити список країн при введенні назви континенту і виводити континент, чисельність населення і площу при введенні назви країни.
28	Створити список з ключами-назвами районів Києва. Як значення задавати кортежі, що містять таку інформацію про район: площа, населення, місцезнаходження та історичні місцевості, які він включає. Програма повинна виводити на друк інформацію про район при вводі його назви. Також необхідно за запитом виводити райони, які знаходяться на лівому березі, або на правому березі. При вводі історичної місцевості вивести назву району, який її включає.
29	Створити список з ключами-країнами Євросоюзу. В якості значень розмістити інформацію про площу країни, кількість населення та місцезнаходження. Програма повинна виводити всі країни, що знаходяться в одній зоні місцезнаходження, наприклад: Східна Європа, Північна Європа і т. ін. Також програма повинна виводити дані про країну при введенні її назви.

Зміст звіту:

1. Титульний лист повинен мати такий вигляд:

Національний технічний університет України
«Київський політехнічний інститут ім. І. Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Програмування
Лабораторна робота №5
«Словники та функції користувача»

Виконав:
студент групи ІО-XX
П.І.Б. студента
Номер у списку групи: XX
Перевірив:
(П.І.Б. викладача).

Київ 2023р.

2. Мета лабораторної роботи та загальне завдання
3. Короткі теоретичні відомості, які відображають інформацію про словники, що були використані при написанні лабораторної роботи.
4. Скріншот вашого варіанту
5. Алгоритм його розв'язування (у довільній формі).
6. Роздруківка того фрагменту тексту програми, який написаний індивідуально чорними символами на білому фоні.
7. Роздруківка результатів виконання програми з контрольним прикладом.
8. Аналіз результатів та висновки.

Лабораторна робота №6

Тема: «Основи об'єктно-орієнтованого програмування. Модулі та пакети».

Мета: вивчити способи створення та підключення модулів та пакетів. Основи ООП. Методи і атрибути класів та робота з ними. Побудова програми у стилі ООП.

Завдання:

1. Вивчити матеріал лекцій 18, 19, 20 та 21.
2. Виконати індивідуальне завдання лабораторної роботи, вибране відповідно до варіанту.

Теоретичні основи:

Модулі й пакети

Модулем у мові Python називають будь-який файл з програмним кодом.

Кожний модуль може імпортувати інший модуль, одержуючи, таким чином, доступ до атрибутів (змінних, функцій і класів), оголошених усередині імпортованого модуля.

Слід відмітити, що імпортований модуль може містити програму не тільки мовою Python – так, можна імпортувати скомпільований модуль, написаний мовою C.

Усі програми, які ми запускали раніше, були розташовані в модулі з назвою "`__main__`". Одержати ім'я модуля дозволяє визначений атрибут `__name__`.

Інструкція `import`

Імпортувати модуль дозволяє інструкція `import`.

Інструкція `import` має наступний формат:

```
import  <Назва модуля1> [<as Псевдонім1>] [, ...,  
          <Назва модуляN> [<as ПсевдонімN>]]
```

Після імпортування модуля його назва стає ідентифікатором, через який можна одержати доступ до атрибутів, визначених усередині модуля.

Доступ до атрибутів модуля здійснюється за допомогою точкової нотації. Наприклад, звернутися до константи `pi`, розташованої усередині модуля `math`, можна так: `math.pi`

Функція `getattr()` дозволяє одержати значення атрибута модуля по його назві, заданій у вигляді рядка. За допомогою цієї функції можна сформулювати назву атрибута динамічно під час виконання програми.

Формат функції:

```
getattr (<Об'єкт модуля>, <Атрибут> [, <Значення за  
замовчуванням>])
```


Якщо зазначений атрибут не знайдений, виконується виключення `AttributeError`. Щоб уникнути виводу повідомлення про помилку, можна в третьому параметрі вказати значення, яке буде повертатися, якщо атрибут не існує.

Перевірити існування атрибута дозволяє функція

`hasattr` (`<Об'єкт>`, `<Назва атрибута>`).

Якщо атрибут існує, функція повертає значення `True`.

Якщо назва модуля занадто довга, і її незручно вказувати щоразу для доступу до атрибутів модуля, то можна створити псевдонім. Псевдонім задають після ключового слова **as**. Доступ до атрибутів модуля може здійснюватися за допомогою ідентифікатора-псевдоніма.

Кожний модуль Python **може бути скомпільований** і перетворений в особливе внутрішнє представлення (байт-код), – для прискорення виконання коду.

Файли з відкомпільованим кодом зберігаються в папці `__pycache__`, що автоматично створюється в папці, де перебуває сам файл з початковим, невідкомпільованим кодом модуля, і мають імена виду:

`<ім'я файлу з початковим кодом>.cpython-<перші дві цифри номера версії Python>.pyc`.

Для одержання відкомпільованого файлу необхідно виконати код, показаний у прикладі:

Приклад.

```
>>> import py_compile
>>> py_compile.compile('tests.py')
'__pycache__\tests.cpython-35.pyc'
```

Таким чином, після запуску цього коду відкомпільований `tests.py` буде збережений у файлі

`tests.cpython-35.pyc`.

Для імпортування модуля достатньо мати тільки файл з відкомпільованим кодом, файл із початковим кодом у цьому випадку не потрібний.

Імпортування модуля виконується тільки при першому виклику інструкції `import` (або `from`). При кожному виклику інструкції `import` перевіряється наявність об'єкта модуля в словнику `modules` з модуля `sys`. Якщо посилання на модуль перебуває в цьому словнику, то модуль повторно імпортуватися не буде.

Для імпортування тільки деяких визначених ідентифікаторів з модуля можна скористатися інструкцією `from`. Формат інструкції:

```
from<Назв. модуля> import<Ідентифікатор1> [as<Псевдонім1>]  
[, ...,<ІдентифікаторN> [as <Псевдонім N>]]
```

```
from <Назв. модуля> import (<Ідентифікатор1> [as  
<Псевдонім1>]), [...,<ІдентифікаторN> [as <Псевдонім N>]]
```

```
from <Назва модуля> import *
```

Шляхи пошуку модулів

Якщо модулі розміщені в одній папці з файлом основної програми, немає необхідності налаштовувати шляхи пошуку модулів, оскільки папка з файлом, що виконується, автоматично додається в початок списку шляхів.

Одержати повний список шляхів пошуку дозволяє наступний код:

Список `sys.path` містить шляхи пошуку, одержувані з наступних джерел:

- **шлях до поточного каталогу** з кодом основної програми;
- **значення змінної оточення PYTHONPATH**.
- **шляхи пошуку стандартних модулів**;
- **вміст файлів з розширенням `pth`**,

Кожний шлях (абсолютний або відносний) повинен бути розташований на окремому рядку.

Повторне завантаження модулів

Як ви вже знаєте, модуль завантажується тільки один раз при першій операції імпорту. Усі наступні операції імпортування цього модуля будуть повертати вже завантажений об'єкт модуля, навіть якщо сам модуль був змінений. Щоб повторно завантажити модуль, слід скористатися функцією `reload()` з модуля `imp`.

Формат функції:

```
from imp import reload  
reload(<Об'єкт модуля>)
```

Пакети

Пакетом називають каталог з модулями, у якому розташований файл ініціалізації `__init__.py`.

Файл ініціалізації може бути порожнім або містити код, який буде виконаний при першому доступі до пакета.

У будь-якому разі він обов'язково повинен бути присутнім всередині каталогу з модулями.

Для встановлення пакетів на Python використовуємо систему управління пакетами **pip**.

pip – використовується для установки і управління програмними пакетами, написаними на Python.

Для версії Python 3.5 **pip** постачається разом з інтерпретатором Python.

Якщо з якоїсь причини **pip** відсутній, то він може бути завантажений з <https://bootstrap.pypa.io/>

Назва файлу: [get-pip.py](#)

Інструкція по **pip** розміщена:

<https://pip.pypa.io/en/latest/installing/>

Приклад. Як завантажити пакет PyGame під Windows

1. Вибираємо «Пуск-Выполнить»
2. В рядку «Открыть» набираємо «cmd». ОК.
3. У вікні «Адміністратор» набираємо: `pip install PyGame`

При цьому має бути підключення до мережі Internet

Об'єктно-орієнтоване програмування

Об'єктно-орієнтоване програмування (ООП) – це спосіб організації програми, що дозволяє використовувати той самий код багаторазово. На відміну від функцій і модулів, ООП дозволяє не тільки розділити програму на фрагменти, але й описати предмети реального світу у вигляді зручних сутностей – об'єктів, а також організувати зв'язки між цими об'єктами.

Основною «цеглинкою» ООП є клас. *Клас* – це складний тип даних, що включає набір змінних і функцій для керування значеннями, що зберігаються в цих змінних.

Змінні називають *атрибутами*, а функції – *методами*. Клас є фабрикою об'єктів, тобто дозволяє створити необмежену кількість екземплярів, заснованих на цьому класі.

ООП ґрунтується на трьох основних концепціях розробки: **інкапсуляція, спадкування й поліморфізм.**

Визначення класу й створення екземпляра класу

Клас описують за допомогою ключового слова `class` за наступною схемою:

```
class <Назва класу>[(<Клас1>[, ... , <Класn>]):  
[""" Рядок документування """]  
<Опис атрибутів і методів>
```

Інструкція створює новий об'єкт і присвоює посилання на нього ідентифікатору, зазначеному після ключового слова `class`.

Це означає, що назва класу повинна повністю відповідати правилам іменування змінних. Після назви класу в круглих дужках можна вказати один або кілька базових класів через кому. Якщо ж клас не успадковує базові класи, то круглі дужки можна не вказувати.

Всі вирази всередині інструкції `class` виконуються при створенні класу, а не його екземпляра.

Створення атрибута класу аналогічно створенню звичайної змінної.

Метод всередині класу створюється так само, як і звичайна функція, за допомогою інструкції `def`.

Методам класу в першому параметрі, який обов'язково слід вказати явно, автоматично передають посилання на екземпляр класу. **Загальноприйнято** цей параметр називати ім'ям `self` (не обов'язково).

Доступ до атрибутів і методів класу всередині обумовленого методу проводиться через змінну `self` за допомогою точкової нотації – до атрибута `x` з методу класу можна звернутися так: `self.x`.

Щоб використовувати атрибути й методи класу, необхідно створити екземпляр класу згідно з наступним синтаксисом:

```
<Екземпляр класу> = <Назва класу>((<Параметри>))
```

При доступі до методів класу використовують такий формат:

```
<Екземпляр класу>.<Ім'я методу>((<Параметри>))
```

При виклику методу не потрібно передавати посилання на екземпляр класу як параметр, як у визначенні методу всередині класу. Посилання на екземпляр класу інтерпретатор передає автоматично.

Доступ до атрибутів класу здійснюється аналогічно:

```
<Екземпляр класу>.<Ім'я атрибута>
```

Методи `__init__()` і `__del__()`

При створенні екземпляра класу інтерпретатор автоматично викликає метод ініціалізації `__init__()`. В інших мовах програмування такий метод прийнято називати конструктором класу.

Формат методу:

```
def __init__(self[, <Значення1>[, ... , <Значенняn>]]):  
<Інструкції>
```

За допомогою методу `__init__` () можна присвоїти початкові значення атрибутам класу. При створенні екземпляра класу параметри цього методу вказують після імені класу в круглих дужках:

```
<Екземпляр класу> = <Ім'я класу>([<Значення1>[, ... ,  
<Значенняn>]])
```

Якщо **конструктор** викликають при створенні об'єкта, то перед знищенням об'єкта автоматично викликається метод, називаний **деструктором**. У мові Python деструктор реалізується у вигляді визначеного методу `__del__` (). Метод не буде викликаний, якщо на екземпляр класу існує хоча б одне посилання.

Інкапсуляція

Розглянемо концепцію розробки, що одержала назву «**інкапсуляція**». Ідея інкапсуляції полягає в тому, щоб сховати логіку функціонування від зовнішнього доступу, а користувачеві даного фрагмента коду надати тільки інтерфейс для його використання.

1. У цьому випадку ви можете звертатися до цього фрагмента з різних областей програми, що заощаджує розмір коду.

2. Якщо необхідно модифікувати інкапсульований фрагмент, то це не вплине на працездатність всієї програми.

В Python інкапсуляція виконується за допомогою методів класу.

Перезавантаження операторів

В Python є можливість перехоплювати за допомогою спеціальних вбудованих методів виконання стандартних операцій і застосовувати нові варіанти цих операцій до екземпляра класу.

Спадкування

Спадкування є найголовнішим поняттям ООП.

Припустимо, у нас є клас (наприклад, `Class1`). За допомогою спадкування ми можемо створити новий клас (наприклад, `Class2`), у якому буде реалізований доступ до всіх атрибутів і методів класу `Class1`.

УВАГА! Конструктор базового класу автоматично не викликається, якщо він перевизначений у підкласі.

Щоб викликати однойменний метод з базового класу, можна також скористатися функцією `super()`.

Формат функції:

```
super([<Клас>, <Показчик self>])
```

За допомогою функції `super()` інструкцію

```
Class1.__init__(self) #Викликаємо конструктор базового класу
```

можна записати так:

```
super().__init__() #Викликаємо конструктор базового класу  
або так:
```

```
super(Class2, self).__init__() # Викликаємо конструктор базового класу
```

При використанні функції `super()` **не потрібно явно** передавати вказівник `self` у викликуваний метод. Крім того, у першому параметрі функції `super()` вказують похідний клас (підклас), а не базовий. Пошук ідентифікатора буде проводитися у всіх базових класах. Результатом пошуку стане перший знайдений ідентифікатор у ланцюжку спадкування.

Множинне спадкування

У визначенні класу в круглих дужках можна вказати **кілька** базових класів через кому.

Поліморфізм

Парадигма об'єктно-орієнтованого програмування крім спадкування включає ще одну важливу особливість — поліморфізм. Слово «поліморфізм» можна перевести як «багато форм».

В ОО (об'єктно-орієнтованому) програмуванні цим терміном позначають можливість використання того самого імені операції або методу до об'єктів різних класів, при цьому дії, виконувані з об'єктами, можуть суттєво різнитися. Тому можна сказати, що в одного слова багато форм.

Перевизначення методів

Використання поліморфізму при спадкуванні класів дозволяє перевизначати методи суперкласів їх підкласами. Наприклад, може виникнути ситуація, коли всі підкласи реалізують визначений метод із суперкласу, і лише один підклас повинен мати його іншу реалізацію. У такому випадку метод перевизначають в підкласі.

Домішки та їх використання

Множинне спадкування, підтримуване Python, дозволяє реалізувати спосіб розширення функціональності класів за допомогою так званих *домішок*.

(*mixins*). Домішка – це клас, що включає атрибути й методи, які необхідно додати до інших класів. Оголошуються вони так само, як і звичайні класи.

Для доступу до атрибутів і методів можна використовувати наступні функції:

Функція `getattr()` – повертає значення атрибута по його назві, заданій у вигляді рядка.

За допомогою цієї функції можна сформувати ім'я атрибута динамічно під час виконання програми. Формат функції:

```
getattr(<Об'єкт>, <Атрибут>[, <Значення за замовчуванням>])
```

Якщо зазначений атрибут не знайдений, виконується виключення `AttributeError`. Щоб уникнути виводу повідомлення про помилку, можна в третьому параметрі вказати значення, яке буде повертатися, якщо атрибут не існує.

Функція `setattr()` – задає значення атрибута. Назва атрибута вказується у вигляді рядка.

Формат функції:

```
setattr(<Об'єкт>, <Атрибут>, <Значення>)
```

Другим параметром методу `setattr()` можна передати ім'я неіснуючого атрибута – у цьому випадку атрибут із зазначеним іменем буде створений.

Функція `delattr(<Об'єкт>, <Атрибут>)` – видаляє зазначений атрибут. Назва атрибута вказується у вигляді рядка.

Функція `hasattr(<Об'єкт>, <Атрибут>)` – перевіряє наявність зазначеного атрибута. Якщо атрибут існує, функція повертає значення `True`.

Спеціальні методи

Класи підтримують наступні спеціальні методи:

Метод `__call__()` – дозволяє обробити виклик екземпляра класу як виклик функції.

Формат методу:

```
__call__(self[, <Параметр1>[, ... , <Параметрn>]])
```

Метод `__getattr__(self, <Атрибут>)` – викликається при доступі до неіснуючого атрибута класу.

Метод `__getattr__(self, <Атрибут>)` – викликається при доступі до будь-якого атрибута класу. Необхідно враховувати, що використання точкової нотації (для доступу до атрибута класу) всередині цього методу призведе до зациклення. Щоб уникнути зациклення, слід викликати метод `__getattr__()` об'єкта `object`. Всередині методу потрібно повернути значення атрибута або виконати виключення `AttributeError`.

Метод `__setattr__(self, <Атрибут>, <Значення>)` – викликається при спробі присвоювання значення атрибуту екземпляра класу. Якщо всередині методу необхідно присвоїти значення атрибуту, то слід використовувати словник `__dict__`, інакше при точковій нотації метод `__setattr__()` буде викликаний повторно, що призведе до зациклення.

Метод `__delattr__(self, <Атрибут>)` – викликають при видаленні атрибута за допомогою інструкції `del <Екземпляр класу>.<Атрибут>;`

Метод `__len__(self)` – викликають при використанні функції `len()`, а також для перевірки об'єкта на логічне значення при відсутності методу `__bool__()`. Метод повинен повертати додатне ціле число.

Метод `__bool__(self)` – викликають при використанні функції `bool()`.

Метод `__int__(self)` – викликають при перетворенні об'єкта в ціле число за допомогою функції `int()`.

Метод `__float__(self)` – викликають при перетворенні об'єкта в дійсне число за допомогою функції `float()`.

Метод `__complex__(self)` – викликають при перетворенні об'єкта в комплексне число за допомогою функції `complex()`.

Метод `__round__(self, n)` – викликають при використанні функції `round()`.

Метод `__index__(self)` – викликають при використанні функцій `bin()`, `hex()` і `oct()`.

Методи `__repr__(self)` і `__str__(self)` – слугують для перетворення об'єкта в рядок.

Метод `__repr__()` викликають при виводі в інтерактивній оболонці, а також при використанні функції `repr()`.

Метод `__str__()` викликають при виводі за допомогою функції `print()`, а також при використанні функції `str()`. Якщо метод `__str__()` відсутній, то буде викликаний метод `__repr__()`.

Як значення методи `__repr__()` і `str()` повинні повертати рядок.

Метод `__hash__(self)` – цей метод слід перевизначити, якщо екземпляр класу заплановано використовувати як ключ словника або всередині множини.

Підсумок по ООП для Python

Найважливіші положення концепції ООП (об'єктно-орієнтованого програмування) для Python:

1. **Створення екземплярів класів.** На цьому етапі важливим моментом є заповнення атрибутів екземпляра класу під час його створення.
2. **Створення методів,** які інкапсулюють логіку поведінки в методах класу.
3. **Перевантаження операторів.** Задає індивідуальну поведінку екземпляра класу при виконанні вбудованих операцій, наприклад, при виводі на друк.
4. **Спадкування.** Дозволяє використовувати методи й атрибути класів-предків (суперкласів), а також модифікувати їх.
5. **Модифікація конструкторів.** Дозволяє змінити або додати логіку при створенні суперкласу.

Індивідуальні завдання

Відповідно до **номера у списку** вибрати індивідуальне завдання. Написати програму. Забезпечити ввід даних з клавіатури комп'ютера та друк результатів. При виводі використовувати форматування.

	Завдання
1	Створити клас, який описує роботу кафе і містить асортимент страв, їх базову ціну, знижки за днями тижня, знижки для постійних відвідувачів та метод обчислення вартості замовлення. Створити об'єкт даного класу, що дозволяє визначити вартість обіду для довільного клієнта з урахуванням знижок. Програма повинна виводити рахунок зі списком

	Завдання
	замовлених страв, їх ціною та знижкою з урахуванням особи клієнта та дня тижня.
2	<p>Створити клас, який описує проведення змагань і містить прізвища учасників, стать, кількість набраних балів, наявність допінгу, вагову категорію та містить метод визначення двох таблиць учасників (за статтю), сформованих у порядку спадання результатів. Окремо створити таблицю дискваліфікованих. Визначення результатів відбувається з урахуванням вагової категорії.</p> <p>Створити об'єкт даного класу, що дозволяє визначити розстановку учасників у таблиці змагань та таблиці дискваліфікованих за умови генерування кількості балів випадковим чином. Максимальна кількість балів – 100. Наявність допінгу у учасника також поставити випадковим чином. Програма повинна виводити таблицю всіх учасників, таблиці переможців з урахуванням статі та таблицю дискваліфікованих.</p>
3	<p>Створити клас, який описує місяць року і містить назву місяця, список днів даного місяця, довжина якого формується в залежності від номеру місяця.</p> <p>Кожен день може бути сонячним, хмарним або дощовим. Кожен день також характеризується середньодобовою температурою. Клас має містити методи визначення середньої кількості сонячних хмарних та дощових днів, методи визначення мінімальної, максимальної та середньої температури.</p> <p>Створити об'єкти даного класу, для кожного місяця року, задати кількості сонячних, хмарних, дощових днів та середньодобову температуру кожного дня випадковим чином з урахуванням пори року та номеру місяця. Програма повинна виводити за запитом середні кількості сонячних, хмарних, дощових днів та середню температуру для кожного місяця за запитом.</p>
4	<p>Створити клас, який описує книгу та містить назву книги, автора, кількість сторінок, кількість ілюстрацій та містить метод визначення ціни книги в залежності від ціни сторінки тексту, ціни ілюстрації та видавництва.</p> <p>Створити об'єкти даного класу для певного набору книг, виданих видавництвами України. Ціну сторінки та ціну ілюстрації задати випадковим чином.</p> <p>Програма повинна виводити ціну книги, за назвою. Вивести середню ціну книги для введеного автора та для окремого видавництва.</p>
5	<p>Створити клас, який описує місто та містить назву міста, кількість мешканців, кількість районів, кількість ВНЗ (вищих навчальних закладів), площу, географічне розташування (північ, південь, центр і т. ін.). Клас має також містити методи обчислення кількості мешканців на одиницю площі, середню площу району міста, кількість жителів на один ВНЗ.</p>

	Завдання
	<p>Створити об'єкти даного класу для деякої кількості міст України. Якщо необхідні параметри неможливо знайти в Інтернет, то задати їх випадковим чином.</p> <p>Програма повинна вивести на друк після вводу назви міста: кількість мешканців, площу, кількість районів, кількість мешканців на одиницю площі, середню площу району міста, кількість жителів на один ВНЗ та географічне положення міста.</p>
6	<p>Створити клас, який описує континент нашої планети та містить назву континенту, площу, географічне положення, чисельність населення та кількість країн, що на ньому розміщені. Методи даного класу повинні визначати чисельність населення на одиницю площі; приналежність континенту до північної, південної, східної, західної півкулі або кількох цих півкуль одночасно.</p> <p>Створити об'єкти даного класу для всіх континентів та ввести дані, користуючись даними з Вікіпедії.</p> <p>Програма повинна вивести на друк після вводу назви континенту: кількість мешканців, площу, кількість країн. Також програма повинна виводити список континентів, які розташовані у півкулі, за запитом, а також таблицю континентів, відсортованих за густиною населення.</p>
7	<p>Створити клас, який описує країну. Він повинен містити назву країни і назву континенту, на якому вона розміщена, чисельність населення, список міст-мільйонників, та чисельність населення в кожному з таких міст. Методи класу обчислюють відсоток міського населення в країні та густину населення.</p> <p>Створити об'єкти даного класу для кількох країн та ввести дані, користуючись даними з Вікіпедії.</p> <p>Програма повинна обчислювати та виводити за запитом відсоток міського населення та густину населення в країні. Програма повинна також виводити за запитом найбільше та найменше місто серед тих, що були введені в програму, друкуючи його назву та чисельність населення.</p>
8	<p>Створити клас, який описує планету Сонячної системи. Цей клас повинен містити назву планети, її порядковий номер, рахуючи від Сонця, відстань від Сонця, об'єм та масу, орбітальну швидкість. Методи класу мають визначати густину планети, відношення маси Землі до маси даної планети, відношення орбітальної швидкості Землі до о.ш. планети.</p> <p>Створити об'єкти даного класу для планет Сонячної системи та ввести дані, користуючись даними з Вікіпедії.</p> <p>Програма повинна обчислювати та виводити за запитом густину планети, відношення маси та о.ш. Землі до маси та о.ш. планети, введеної за запитом. Програма повинна також виводити за запитом список планет у порядку зростання їх густини з зазначенням номеру планети та її орбітальної швидкості.</p>

	Завдання
9	<p>Створити клас, який описує факультет КПІ. Клас повинен містити назву, скорочену назву, прізвище декана та номер телефону деканату, список спеціальностей, список кафедр. Методи класу повинні визначати, чи належить кафедра до даного факультету, чи належить спеціальність до даного факультету. Створити об'єкти даного класу, користуючись інформацією сайту НТУУ «КПІ».</p> <p>При введенні скороченої назви факультету програма повинна виводити його повну назву, прізвище декана, номер телефону деканату.</p> <p>При вводиті назви спеціальності програма повинна виводити список факультетів, які готують студентів за даною спеціальністю. При вводиті назви кафедри програма має виводити відповідний факультет та номер телефону деканату.</p>
10	<p>Створити клас, який описує область України. Клас повинен містити назву області, її площу, чисельність населення та список головних міст. Методи класу повинні визначати, чи належить місто до даної області та густину населення.</p> <p>Створіть об'єкти даного класу, які описують області України, та введіть потрібну інформацію, користуючись даними Вікіпедії.</p> <p>При введенні назви області програма повинна виводити площу, чисельність населення, густину населення та список головних міст. При вводиті назви міста програма повинна виводити назву області, у якій воно розміщено. За запитом програма повинна виводити список областей, відсортованих за густиною населення з зазначенням назви області та густини населення.</p>
11	<p>Створити клас, який описує магазин одягу на прикладі інтернет-магазину http://www.lila-shop.ua/. Клас містить назву магазину, список товарів, список постійних клієнтів, накопичувальну скидку для постійних клієнтів, сезонну знижку та знижку, яка враховує час приходу в магазин. Клас містить метод обчислення накопичувальної знижки для конкретного постійного клієнта, метод обчислення сезонної знижки та метод обчислення в залежності від години відвідування магазину.</p> <p>Створити об'єкт даного класу, який описує роботу деякого магазину, ввівши асортимент товару, постійних покупців та відсотки для трьох видів знижок.</p> <p>Програма повинна виводити розрахунок вартості конкретного вибраного товару для покупця, враховуючи всі види знижок. Програма також повинна виводити список постійних покупців зі значеннями накопичувальної знижки та список товарів з їх цінами.</p>
12	<p>Створити клас, який описує інтернет-магазин на прикладі http://rozetka.com.ua. Клас повинен містити багаторівневу структуру зберігання інформації про товари, яка включає каталог товарів, кожен пункт якого містить категорії. Кожна категорія містить словник товарів, ключом якого є код товару, а кортеж значень включає назву, ціну та інші дані. Клас повинен мати методи, які дають можливість сортування</p>

	Завдання
	<p>товарів за спаданням ціни та за зростанням ціни, пошуку товару за фрагментом назви, за діапазоном ціни.</p> <p>Створити об'єкт даного класу, використовуючи дані з магазину «rozetka».</p> <p>Програма повинна мати простий інтерфейс пошуку товару, який базується на використанні методів класу. Програма також має формувати корзину покупок з виводом списку вибраних товарів та загальної суми.</p>
13	<p>Створити клас, який описує англо-український словник. Клас повинен містити список можливих варіантів перекладу. Методи класу повинні забезпечити простий пошук англійських слів за введеним підрядком.</p> <p>Створити об'єкт даного класу, який дозволяє, користуючись методами та атрибутами класу, одержати український переклад.</p> <p>Програма повинна виводити український переклад введеного англійського слова з можливими варіантами. За приклад взяти словник http://e2u.org.ua/.</p>
14	<p>Створити клас, який описує тлумачний словник української мови. Клас повинен містити словник з ключів-українських слів та значень у вигляді кортежу синонімів. Методи класу мають реалізувати пошук слова за підрядком.</p> <p>Створити об'єкт даного класу, який дозволяє, користуючись методами та атрибутами класу, одержати синоніми українського слова.</p> <p>Програма повинна виводити синоніми введеного українського слова з можливими варіантами. За приклад взяти словник http://sum.in.ua/</p>
15	<p>Створити клас, який описує найбільші міста нашої планети. Клас повинен містити словник, в якості ключів якого є континенти, а в якості значень є словники, ключами яких є міста-мільйонники даного континенту, а значеннями – кортежі з площею, чисельністю населення та географічними координатами. Методи класу повинні знаходити найближче розташоване місто до введеного, а також найближче місто за площею та найближче за густиною населення.</p> <p>Створити об'єкт даного класу, який дозволяє з використанням методів класу знаходити найближче розташоване міста та подібні за площею і густиною населення міста. Для вводу інформації про міста користуватися даними з Вікіпедії.</p> <p>Програма повинна виводити за запитом найближче розташоване місто, а при повторних запитах міста, які відсортовані за відстанню від даного міста. Також програма повинна виводити найближче за площею та густиною населення місто і всі міста для введеного континенту.</p>
16	<p>Введіть клас держав нашої планети. Клас повинен містити словник з ключами-державами та кортежами-значеннями, що включають назву столиці, її площу та чисельність населення. Методи класу повинні визначати список держав, столиці яких мають чисельність населення в заданому діапазоні. Визначати список держав, які мають столиці з</p>

	Завдання
	<p>заданим діапазоном площі, визначати відношення площі столиці до площі держави, а також визначати назву держави, до якої належить столиця.</p> <p>Створити об'єкт класу, який використовує всі методи класу, для введення інформації скористатися даними Вікіпедії.</p> <p>Програма повинна виводити список держав, які містять чисельність населення з введеного діапазону у порядку спадання. Також при виводі держав, що мають столиці в заданому діапазоні площ, використати сортування за спаданням. В обох згаданих випадках виводити також назви столиць, чисельність їх населення та площі. При вводі назви столиці, програма повинна виводити державу, до якої вона належить.</p>
17	<p>Створити клас, який описує роботу салону з продажу автомобілів. Салон повинен містити словник автомобілів, структурований за виробниками та марками. Кожен автомобіль повинен характеризуватися ціною, роком випуску та пробігом. Методи класу повинні визначати наявність в салоні марок автомобілів заданого виробника, визначати автомобілі, які потрапляють в заданий діапазон цін, визначати автомобілі з пробігом, який знаходиться в заданому діапазоні.</p> <p>Створити об'єкт класу, який використовує всі методи класу, для введення інформації скористатися даними http://privat-auto.info/.</p> <p>Програма повинна виводити на друк список автомобілів та їх параметри за введеним виробником, виводити автомобілі, які знаходяться у заданому діапазоні цін або пробігу. Також програма повинна виводити на друк список автомобілів та їх параметри заданої марки.</p>
18	<p>Створити клас, який описує бібліотеку. Клас повинен містити назву бібліотеки, словник з ключами-жанрами книг (пригоди, фантастика, історичний роман і т.д.) Як значення задати кортеж, який містить прізвище автора, його ім'я, назву книги, кількість сторінок та видавництво. Методи класу повинні визначати список книг з введеного жанру, визначати жанри, у яких існують книги даного автора, визначати видавництво і автора при введенні назви книги, знаходити всі книги, у назвах яких зустрічається підрядок вводу.</p> <p>Створити об'єкт класу, який описує конкретну бібліотеку та реалізує задані методи класу. В якості джерела інформації для введення даних скористайтесь http://library.kpi.ua/</p> <p>Програма повинна за запитом виводити книги, що написав відповідний автор, книги за жанром, за діапазоном сторінок, за видавництвом. При написанні програми для пошуку даних застосувати пошук за підрядком.</p>
19	<p>Створити клас дисциплін, які ви вивчаєте. Клас має містити словник з ключами-назвами предметів. В якості значень задати кортежі, які</p>

	Завдання
	<p>містять як елементи прізвище викладача, кількість годин на тиждень та наявність екзамену, диференційного заліку або простого заліку.</p> <p>Методи класу мають визначати прізвище викладача за назвою предмету та назву предмету за прізвищем викладача, якщо викладач читає кілька предметів, то вони повинні бути перераховані при виводі.</p> <p>Створити об'єкт даного класу, користуючись своїм розкладом.</p> <p>Програма повинна виводити після введення предмету прізвище викладача і навпаки. При виводі повідомляти про наявність екзамену або заліку. Також за запитом необхідно виводити список предметів, які мають однакову кількість годин на тиждень.</p>
20	<p>Створіть клас, який описує квартиру. Він повинен містити номер квартири, кількість квадратних метрів, кількість спожитої за місяць холодної води, кількість спожитої гарячої води, величину спожитої електроенергії, суму, яка сплачена за місяць за холодну воду, сплачену суму за споживання води, суму за електроенергію. Методи класу повинні обраховувати величину переплати чи заборгованості по кожному платежу з урахуванням тарифів, які прийняті в Україні. Ці тарифи можна дізнатися: http://kyivenergo.ua/ та https://www.giok.kiev.ua/law/tariff/</p> <p>Створити об'єкти даного класу для квартир багатоквартирного будинку. Задати випадковим чином суми оплати та кількості спожитих послуг.</p> <p>Програма повинна виводити кількості спожитих послуг, суми необхідної оплати за них, реальну оплату за кожну послугу та величину переплати або заборгованості. При введенні назви послуги та визначенні значення параметра «переплата» або «заборгованість» програма повинна виводити номери квартир та відповідно їх переплати або заборгованості. Програма також повинна виводити загальну переплату або заборгованість по будинку.</p>
21	<p>Створіть клас, який описує Вікіпедію. Клас повинен містити словник з ключами-розділами Вікіпедії (Україна, Культура і мистецтво, Технології та інженерія, Прикладні та природничі науки і т. ін.). Відповідними значеннями є словники з ключами, що відображають підкатегорії, та значеннями у вигляді кортежів. При створенні структури даних користуватися інформацією за адресою https://uk.wikipedia.org/. Методи класу повинні визначати статті, які містять у назві підрядок з запиту, визначати всі підкатегорії введеної категорії.</p> <p>Створити об'єкт даного класу, який наслідує методи класу. Для заповнення даними скористатися інформацією з Вікіпедії.</p> <p>Програма повинна виводити список статей за введеним підрядком, виводити всі підкатегорії за введеною категорією.</p>
22	<p>Створіть клас, який описує структуру кабінету міністрів. Клас повинен містити повні та скорочені назви міністерств, словник, ключами якого є скорочені назви міністерств. В якості значення використайте кортеж,</p>

	Завдання
	<p>який складається з контактного телефону, електронної адреси міністерства та прізвища міністра. Методи класу повинні визначати повну назву міністерства за його скороченою назвою, визначати контактний телефон та ін. за скороченою назвою.</p> <p>Створити об'єкт класу, який описує Кабінет Міністрів України. Для заповнення даними скористатися інформацією http://www.kmu.gov.ua/ Програма повинна виводити назву міністерства після введення прізвища міністра, телефон і email після введення назви міністерства, виводити повну назву після введення скороченої назви, виводити всі міністерства, які у повній назві мають підрядок запиту.</p>
23	<p>Створіть клас, який описує українську мову. Клас повинен містити багаторівневий словник з ключами – назвами частин мови. Кожна ключ-частина мови повинен мати значення у вигляді словника другого рівня і т. д. відповідно до класифікації, яка подана у Вікіпедії: https://uk.wikipedia.org/wiki/Частини_мови. На найнижчому рівні словника повинні знаходитися кортежі зі словами. Методи класу повинні визначати всі характеристики слова як частини мови при його введенні, визначати приклади слів, які відповідають введеній характеристиці частини мови, методи вводу та модифікації слів.</p> <p>Створити об'єкт класу та ввести дані, використовуючи інформацію з Вікіпедії.</p> <p>Програма повинна виводити всі характеристики слова як частини мови при його введенні, виводити приклади слів, які відповідають введеній характеристиці частини мови, вводити та редагувати слова.</p>
24	<p>Створіть клас, який описує структуру міської державної адміністрації. Клас повинен містити назви Управлінь міської державної адміністрації та їх скорочені назви. Створіть словник, у якому як ключ використовується скорочена назва Управління. Як значення задати прізвище керівника та контактний телефон.</p> <p>Методи класу повинні визначати повну назву управління після введення скороченої назви, визначати управління при пошуку за підрядком, визначати керівника та контактні дані за запитом, додавати нові та редагувати введені управління</p> <p>Створити об'єкт класу, який описує Київську міську державну адміністрацію (КМДА) та ввести дані, використовуючи інформацію з https://kievcity.gov.ua/.</p> <p>Програма повинна виводити на друк повну назву управління після введення скороченої назви, виводити управління при пошуку за підрядком, виводити керівника та контактні дані за запитом, додавати нові та редагувати введені управління</p>
25	<p>Створити клас, який описує структуру торгової мережі супермаркетів. Клас повинен містити назву мережі, інформацію про акції та товари, що підпадають під акційні знижки, перелік товарів з поділом на тематичні</p>

	Завдання
	<p>групи та ціною і кодами. Методи класу повинні визначати знижку відповідно до акції або акцій, в яких бере участь той чи інший товар. Створити об'єкт даного класу, який описує мережу сілпо. Для вводу інформації скористатися http://silpo.ua/ua/.</p> <p>При вводі товару програма повинна виводити на друк список товарів, які містять підрядок назви товару з їх кодами. При вводі коду товару програма повинна вивести ціну товару, розрахунок поточної знижки на товар та вартість товару з урахуванням знижок.</p>
26	<p>Створити клас, який описує структуру торгової мережі, яка має філіали, що розташовані в містах України. Клас повинен містити назву мережі, словник, ключами якого є міста України, у яких існують філіали мережі. Як значення запишіть кортеж, який містить місцеву адресу, телефон та іншу інформацію за бажанням.</p> <p>Методи класу мають визначати всі міста з філіалами при введенні підрядка назви міста, визначати відомі дані про філіал при вводі міста, вводити інформацію про нові філіали та редагувати введені філіали. Створити об'єкт класу, який описує мережу магазинів «WINTIME» та ввести дані, використовуючи інформацію з http://www.winetime.ua/store/.</p> <p>Програма повинна виводити на друк всі міста з філіалами магазину, які починаються з введеної букви; виводити адресу, телефон та іншу доступну інформацію філіалу магазину, який розміщений у місті, що вводиться у запиті, вводити нові філіали та редагувати параметри введених філіалів.</p>
27	<p>Створити клас, який описує континент. Клас повинен містити назву континенту, словник з ключами – назвами країн, які розміщені на даному континенті, і значеннями-кортежами. Елементами таких кортежів повинні бути назви столиць відповідних країн, їх площа та чисельність населення. Методи класу повинні визначати приналежність країни до континенту, визначати параметри країни при вводі її назви.</p> <p>Створити об'єкти даного класу для кількох континентів, користуючись даними Вікіпедії.</p> <p>Програма повинна виводити список країн при введенні назви континенту і виводити континент, чисельність населення і площу при введенні назви країни.</p>
28	<p>Створити клас, який описує район міста. Клас повинен містити назву району, інформацію про район: площа, населення, місцезнаходження та історичні місцевості, які він включає. Методи класу мають визначати інформацію про район за його назвою, визначати, на якому березі район знаходиться, визначати перелік історичних назв місцевостей.</p> <p>Створити об'єкти даного класу, які описують райони міста Києва. Для вводу інформації скористатися https://uk.wikipedia.org/wiki/Адміністративно-територіальний устрій Києва.</p>

	Завдання
	Програма повинна виводити на друк інформацію про район при вводиті його назви. Також необхідно за запитом виводити райони, які знаходяться на лівому березі, або на правому березі. При вводиті історичної місцевості вивести назву району, який її включає.
29	<p>Створити клас, який описує країну-члена Євросоюзу. Клас має містити назву країни, інформацію про площу країни, чисельність населення, рік вступу та місцезнаходження. Методи класу повинні визначати зону місцезнаходження країни, визначати інформацію про країну за запитом.</p> <p>Створити об'єкти даного класу, які описують країни Євросоюзу.</p> <p>Для вводу інформації скористатись даними з Вікіпедії: https://uk.wikipedia.org/wiki/Європейський Союз.</p> <p>Програма повинна виводити всі країни, що знаходяться в одній зоні місцезнаходження, наприклад: Східна Європа, Північна Європа і т. ін. Також програма повинна виводити дані про країну при введенні її назви, виводити сусідні країни при введенні назви країни.</p>

Зміст звіту:

1. Титульний лист повинен мати такий вигляд:

Національний технічний університет України
«Київський політехнічний інститут ім. І. Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Програмування Лабораторна робота №6

«Основи об'єктно-орієнтованого програмування. Модулі та пакети»

Виконав:
студент групи ІО-ХХ
П.І.Б. студента
Номер у списку групи: ХХ
Перевірив:
П.І.Б. викладача.

Київ 2023р.

2. Мета лабораторної роботи та загальне завдання
3. Короткі теоретичні відомості, які відображають інформацію про модулі, пакети та елементи ООП, **що були використані при написанні лабораторної роботи.**
4. Скріншот вашого варіанту
5. Алгоритм (у довільній формі).
6. Роздруківка того фрагменту тексту програми, який написаний індивідуально чорними символами на білому фоні.
7. Скріншот результатів виконання програми з контрольним прикладом
8. Аналіз результатів та висновки.

Лабораторна робота №7

Тема: «Обробка виключень та робота з файлами».

Мета: вивчити основні способи роботи з виключеннями. Виключення користувача. Відкриття файлів, зчитування та запис у файл. Шляхи доступу до файлів. Функції, методи та атрибути для роботи з файлами.

Завдання:

1. Вивчити матеріал лекцій 24, 25, 26 та 27.
2. Виконати індивідуальне завдання лабораторної роботи, вибране відповідно до варіанту.

Теоретичні основи:

Обробка виключень

Виключення – це повідомлення інтерпретатора, які виникають у випадку виникнення помилки в програмному коді або при настанні якої-небудь події. Якщо в коді не передбачена обробка виключення, то виконання програми переривається, і виводиться повідомлення про помилку.

Існують три типи помилок у програмі:

Синтаксичні помилки – це помилки в імені оператора або функції, відсутність закриваючих або відкриваючих лапок і т. д., тобто помилки в синтаксисі мови.

Логічні помилки – це помилки в логіці програми, які можна виявити тільки за результатами її роботи.

Помилки часу виконання – це помилки, які виникають під час роботи програми. Причиною є події, не передбачені програмістом.

Інструкція *try...except...else...finally*

Для обробки виключень призначена інструкція `try`.

Формат інструкції:

```
try:
<Блок, у якому перехоплюються виключення>
[except [<Виключення>[ as <Об'єкт виключення>]]]:
<Блок, виконуваний при виникненні виключення>
[ . . .
except [<Виключення>[ as <Об'єкт виключення>]]]:
<Блок, виконуваний при виникненні виключення>]]
[else:
<Блок, виконуваний, якщо виключення не виникло>]
[finally:
<Блок, виконуваний у будь-якому випадку>]
```

Інструкція *with...as*

Мова Python підтримує *протокол менеджерів контексту*. Цей протокол гарантує виконання завершальних дій (наприклад, закриття файлу) незалежно від того, відбулося виключення усередині блоку коду чи ні.

Для роботи з протоколом призначена інструкція *with...as*. Інструкція має наступний формат:

```
with <Вираз1>[ as <Змінна>][, ...,  
    <Виразn> [as <Змінна>]]:  
    <Блок, у якому перехоплюємо виключення>
```

Спочатку обчислюється <Вираз1>, який повинен повертати об'єкт, що підтримує протокол. Цей об'єкт повинен мати два методи: `__enter__()` і `__exit__()`. Метод `__enter__()` викликається після створення об'єкта. Значення, що повертається цим методом, присвоюється змінній, зазначеній після ключового слова *as*. Якщо змінна не зазначена, значення, що повертається, ігнорується. Формат методу `__enter__()`:

```
__enter__(self)
```

Далі виконуються інструкції усередині тіла інструкції *with*. Якщо при виконанні виникло виключення, то керування передається методу `__exit__()`. Метод має наступний формат:

```
__exit__(self, <Тип виключення>, <Значення>, <Об'єкт  
traceback>)
```

Виключення користувача

Для виконання виключень користувача призначено дві інструкції: *raise* і *assert*.

Інструкція *raise* виконує задане виключення. Вона має кілька варіантів формату:

```
raise <Екземпляр класу>  
raise <Назва класу>  
raise <Екземпляр або назва класу> from <Об'єкт  
виключення>  
raise
```

У першому варіанті формату інструкції *raise* вказується екземпляр класу порушеного виключення. При створенні екземпляра можна передати дані конструктору класу.

Ці дані будуть доступні через другий параметр в інструкції *except*.

Робота з файлами

Відкриття файлу

Функція відкриття має наступний формат:

```
open(<Шлях до файлу>[, mode='r'] [, buffering=-1] [,
encoding=None) [,errors =None] [, newline=None) [,
closefd=True])
```

Відносний шлях буде автоматично перетворений в абсолютний шлях за допомогою функції `abspath()` з модуля `os.path`. Можливі наступні варіанти:

1. Якщо файл, що відкривається, перебуває в поточному робочому каталозі, то можна вказати тільки назву файлу.
2. Якщо файл, що відкривається, розташований у вкладеній папці, то перед назвою файлу приводять назви вкладених папок через слеш.
3. Якщо папка з файлом розташована вище рівнем, то перед назвою файлу вказують дві крапки й слеш ("`.. \`").
4. Якщо на початку шляху розташований слеш, то шлях відлічується від кореня диска. У цьому випадку місце розташування поточного робочого каталогу не має значення.

В абсолютному й відносному шляхах допускається вказувати як прямі, так і зворотні слеші. Усі вони будуть автоматично перетворені з урахуванням значення атрибута `sep` з модуля `os.path`. Значення цього атрибута залежить від використовуваної операційної системи.

Необов'язковий параметр `mode` у функції `open()` може приймати наступні значення:

`r` – тільки читання (значення за замовчуванням).

`r+` – читання й запис.

`w` – запис.

`w+` – читання й запис.

`a` – запис.

`a+` – читання й запис.

`x` – створення файлу для запису.

`x+` – створення файлу для читання й запису.

Після вказівки режиму може слідувати модифікатор:

`b` – файл буде відкритий у бінарному режимі. Файлові методи приймають і повертають об'єкти типу `bytes`;

t – файл буде відкритий у текстовому режимі (значення за замовчуванням у Windows).

У параметрі `errors` можна вказати рівень обробки помилок. Можливі значення:

"strict" (при помилці виконується виключення `ValueError` – значення за замовчуванням),
"replace" (невідомий символ замінюється символом питання або символом з кодом `\ufffd`),
"ignore" (невідомі символи ігноруються),
"xmlcharrefreplace" (невідомий символ замінюється послідовністю `&#xxxx;`),
"backslashreplace" (невідомий символ замінюється послідовністю `\uxxxx`).

Параметр `newline` задає режим обробки символів кінця рядків. Підтримувані ним значення такі:

- `None` (значення за замовчуванням) – виконується стандартна обробка символів кінця рядка. Наприклад, в Windows при читанні символи `\r\n` перетворюються в символ `\n`, а при записі проводиться зворотне перетворення;
- `" "` (порожній рядок) – обробка символів кінця рядка не виконується;
- `"<Спеціальний символ>"` – зазначений спеціальний символ використовується для позначення кінця рядка, і додаткова обробка не виконується. Як спеціальний символ можна вказати лише `\r\n`, `\r` і `\n`.

Методи для роботи з файлами

- `close()` – закриває файл.
- `write (<дані>)` – записує рядок або послідовність байтів у файл.
- `writelines (<Послідовність>)` – записує послідовність у файл.
- `writable()` – повертає `True`, якщо файл підтримує запис, і `False` – якщо ні.
- `read([<Кількість>])` – зчитує дані з файлу.
- `readline([<Кількість>])` – зчитує з файлу один рядок при кожному виклику.
- `readlines()` – зчитує весь вміст файлу в список.
- `flush()` – примусово записує дані з буфера на диск;
- `fileno()` – повертає цілочисельний дескриптор файлу. Значення, що повертається, завжди буде більшим за число 2, оскільки число 0

закріплене за стандартним вводом `stdin`, 1 – за стандартним виводом `stdout`, а 2 – за стандартним виводом повідомлень про помилки `stderr`.

- `truncate ([<Кількість>])` – обрізає файл до зазначеної кількості символів (якщо заданий текстовий режим) або байтів (у випадку бінарного режиму).

- `tell()` – повертає позицію покажчика відносно початку файлу у вигляді цілого числа.

- `seek(<Зсув>[, <Позиція>]`) – установлює покажчик у позицію, що має зсув `<Зсув>` відносно позиції `<Позиція>`. У параметрі `<Позиція>` можуть бути зазначені наступні атрибути з модуля `io` або відповідні їм значення:

- `io.SEEK_SET` або 0 -- початок файлу (значення за замовчуванням);

- `io.SEEK_CUR` або 1 – поточна позиція покажчика. Додатне значення зсуву викликає переміщення до кінця файлу, від'ємне - до його початку;

- `io.SEEK_END` або 2 – кінець файлу.

- `seekable()` – повертає `True`, якщо покажчик файлу можна зсунути в іншу позицію, і `False` – якщо ні.

Крім методів, об'єкти файлів підтримують кілька [атрибутів](#):

`name` – ім'я файлу;

`mode` – режим, у якому був відкритий файл;

`closed` – повертає `True`, якщо файл був закритий, і `False` – якщо ні.

`encoding` – назва кодування, яке буде використовуватися для перетворення рядків перед записом у файл або при читанні.

Стандартний вивід `stdout` також є файловим об'єктом. Атрибут `encoding` цього об'єкта завжди містить кодування пристроїв виводу, тому рядок перетвориться в послідовність байтів у правильному кодуванні.

`buffer` – дозволяє одержати доступ до буфера. Атрибут доступний тільки в текстовому режимі. За допомогою цього об'єкта можна записати послідовність байтів у текстовий потік.

Права доступу до файлів і каталогів

Права доступу позначають буквами:

`r` – файл можна читати, а вміст каталогу можна переглядати;

`w` – файл можна модифікувати, видаляти й перейменовувати, а в каталозі можна створювати або видаляти файли. Каталог можна перейменовувати або вилучити;

`x` – файл можна виконувати, а в каталозі можна виконувати операції над файлами, у тому числі робити в ньому пошук файлів.

Таблиця. Права доступу в різних записах

Восьмерична цифра	Двійковий запис	Буквений запис	Вісімкова цифра	Двійковий запис	Буквений запис
0	000	---	4	100	r--
1	001	--x	5	101	r-x
2	010	-w-	6	110	r-w
3	011	-wx	7	111	rwX

Для визначення прав доступу до файлу або каталогу призначена функція `access()` з модуля `os`. Функція має наступний формат:

```
access(<Шлях>, <Режим>)
```

Функція повертає `True`, якщо перевірка пройшла успішно, або `False` – якщо ні. У параметрі `<Режим>` можуть бути зазначені наступні константи, що визначають тип перевірки:

`os.F_OK` – перевірка наявності шляху або файлу:

`os.R_OK` – перевірка на можливість читання файлу або каталогу;

`os.W_OK` – перевірка на можливість запису у файл або каталог;

`os.X_OK` – визначення, чи є файл або каталог виконуваним.

`chmod()` – зміна прав доступу. Функція має наступний формат:
`chmod(<Шлях>, <Права доступу>)`

Функції для маніпулювання файлами

Для **копіювання** й **переміщення** файлів призначені наступні функції з модуля `shutil`:

```
copyfile(<Копіюємий файл>, <Куди копіюємо>)
```

Функція дозволяє скопіювати вміст файлу в інший файл. Ніякі метадані (наприклад, права доступу) не копіюються.

Якщо файл існує, то він буде перезаписаний. Якщо файл не вдалося скопіювати, виконується виключення `OsError` або одне з виключень, що є підкласом цього класу.

```
copy(<Копіюємий файл>, <Куди копіюємо>)
```

Функція дозволяє скопіювати файл разом із правами доступу.

Функція `copy()` як результат повертає шлях скопійованого файлу;

```
copy2(<Копіюємий файл>, <Куди копіюємо>)
```

Функція дозволяє скопіювати файл разом з метаданими.

```
move(<Шлях до файлу>, <Куди переміщаємо>)
```

Функція переміщає файл у зазначене місце з видаленням початкового файлу. Функція `move()` як результат повертає шлях переміщеного файлу.

Для **перейменування** й **видалення** файлів призначені наступні функції з модуля `os`:

```
rename (<Старе ім'я>, <Нове ім'я>)
```

Функція перейменовує файл.

Функції для роботи з файлами

Модуль `os.path` містить додаткові функції, що дозволяють перевірити наявність файлу, одержати розмір файлу й ін. Опишемо ці функції:

```
exists (<Шлях>)
```

Функція перевіряє зазначений шлях на існування. Значенням функції буде `True`, якщо шлях існує, і `False` – якщо не існує.

```
getsize (<Шлях до файлу>)
```

повертає розмір файлу в байтах.

```
getatime (<Шлях до файлу>)
```

Функція служить для визначення часу останнього доступу до файлу. Як значення функція повертає кількість секунд, що пройшли з початку епохи.

```
getctime (<Шлях до файлу>)
```

Функція дозволяє довідатися дату створення файлу. Як значення функція повертає кількість секунд, що пройшли з початку епохи.

```
getmtime (<Шлях до файлу>)
```

Повертає час останньої зміни файлу. Як значення функція повертає кількість секунд, що пройшли з початку епохи.

Одержати розмір файлу й час створення, зміни й доступу до файлу, а також значення інших метаданих дозволяє функція `stat()` з модуля `os`. Як значення функція повертає об'єкт `stat_result`, що містить десять атрибутів: `st_mode`, `st_ino`, `st_dev`, `st_nlink`, `st_uid`, `st_gid`, `st_size`, `st_atime`, `st_mtime` і `st_ctime`.

Оновити час останнього доступу й час зміни файлу дозволяє функція `utime()` з модуля `os`.

Функція має два варіанти формату:

```
utime(<Шлях до файлу>, None)
```

```
utime(<Шлях до файлу>, (<Останній доступ>, <Зміна файлу>))
```

Як перший параметр можна вказувати не тільки шлях як рядок, але й цілочисельний дескриптор відкритого файлу, який повертає функція `open()` з модуля `os`. Якщо як другий параметр зазначене значення `None`, то час доступу й зміни файлу буде поточним. У другому варіанті формату функції `utime()` вказується кортеж з нових значень у вигляді кількості секунд, що пройшли з початку епохи.

Перетворення шляху до файлу або каталогу

Перетворити шлях до файлу або каталогу дозволяють наступні функції з модуля `os.path`:

`abspath (<Відносний шлях>)`

Функція перетворить відносний шлях в абсолютний, враховуючи місце розташування поточного робочого каталогу.

У **відносному** шляху можна вказати **як прямі**, так і **зворотні слеші**. Усі вони будуть автоматично перетворені з урахуванням значення атрибуту `sep` з модуля `os.path`. Значення цього атрибута залежить від використовуваної операційної системи.

`isabs (<Шлях>)`

Функція повертає `True`, якщо шлях є абсолютним, і `False` – якщо ні.

`basename (<Шлях>)`

Функція повертає ім'я файлу без шляху до нього.

`dirname (<Шлях>)`

Функція повертає шлях до папки, де зберігається файл.

`split (<Шлях>)`

Функція повертає кортеж із двох елементів: шляху до папки, де зберігається файл, і назви файлу.

`splitdrive (<Шлях>)`

Функція розділяє шлях на ім'я диска й іншу частину шляху. Як значення повертається кортеж із двох елементів.

`splitext (<Шлях>)`

Функція повертає кортеж із двох елементів: шлях з назвою файлу, але без розширення, і розширення файлу (фрагмент після останньої крапки).

Індивідуальні завдання

Використовуючи функції та методи мови програмування Python:

1. Написати програму створення каталогу зі шляхом та назвою: «C:\lab7\»
2. Написати програму створення підкаталогу «C:\lab7\<прізвище>»

3. Завантажити в даний підкаталог файл *.txt, де * – номер Вашого варіанту лабораторної роботи та виконати з ним дії, що описані в номері Вашого варіанту.
4. Зберегти об'єкти з даними, які створені Вами при виконанні лабораторної роботи №5, у файл, користуючись модулем `pickle`. Створений файл перемістити в попередньо створений каталог «C:\lab5». Зчитати файл, доповнити даними та записати в даний каталог з іншим ім'ям.
5. Зберегти об'єкти з даними, які створені вами при виконанні лабораторної роботи №6, у файл, користуючись модулем `shelve`. Файл перемістити в попередньо створений каталог «C:\lab6». Застосувати три відомі вам методи до модифікації файлу. Вивести на друк модифікований файл.
6. Для виконання наступного завдання вибрати файл, який відповідає номеру варіанта **з архіву, що розміщений на сайті!!!**.

Використання своїх файлів заборонено!!

№	Завдання
1	Зчитати файл «1.txt» та розбити файл на три файли однакового розміру, створивши файли «1part1.txt», «1part2.txt» і «1part3.txt» з кодуванням UTF-8. Вивести на друк розмір початкового файлу та розміри створених файлів. Створити файл, який починається з речення з заданим номером і містить задану кількість речень.
2	Розбити файл «2.txt» на 2 файли «2part1.txt», «2part2.txt», які містять однакову кількість речень, та записати їх у кодуванні «cp1251». Вивести на друк кількість символів у початковому файлі та кількість символів у створених файлах. Створити файл, який містить непарні речення.
3.	Зчитати файл «3.txt» та розбити його на 2 файли «3part1.txt», «3part2.txt», які містять непарні та парні за номером символи початкового файлу. Зберегти ці файли у кодуванні «cp1251». Зчитати дані файли та відновити з них початковий файл, зберегти цей файл у кодуванні UTF-8.
4	Перетворити файл «4.txt» у файл «41.txt», який відрізняється від попереднього тим, що кожне речення у ньому повинно починатися з нового рядка. Створити новий файл «42.txt» з речень, які починаються з заданої літери. Зберегти цей файл у кодуванні UTF-8.
5	Перетворити файл «5.txt» у файл «51.txt», який відрізняється від попереднього тим, що у ньому виконана симетрична зміна речень місцями. Створити новий файл «52.txt» з речень, які містять парну кількість символів. Зберегти цей файл у кодуванні UTF-8.

6	Перетворити файл «6.txt» у файл «61.txt», який складається з перших трьох слів кожного речення. Створити новий файл «62.txt» зі слів, які містять парну кількість символів. Зберегти цей файл у кодуванні UTF-8.
7	Зчитати файл «7.txt» та перетворити його у файл «71.txt», який складається з речень, у яких виконана симетрична зміна порядку слів, та у файл «72.txt», який містить слова, що закінчуються заданою буквою. Зберегти цей файл у кодуванні UTF-8.
8	Зчитати файл «8.txt» та перетворити його у файл «81.txt», який складається з відсортованих за довжиною речень, та у файл «82.txt», який містить слова, що починаються з заданої букви. Зберегти цей файл у кодуванні UTF-8.
9	Зчитати файл «9.txt» та перетворити його у файл «91.txt», який складається з речень, що мають парну кількість слів, та у файл «92.txt», який містить слова, що закінчуються на «ать» або «ять». Зберегти цей файл у кодуванні UTF-8.
10	Зчитати файл «10.txt» та перетворити його у файл «101.txt», який складається з речень, що мають непарну кількість букв, та у файл «102.txt», що містить слова, у яких більше ніж три голосні букви. Зберегти цей файл у кодуванні UTF-8.
11	Зчитати файл «11.txt» та перетворити його у файл «111.txt», який складається з речень, що мають більше ніж одну велику букву, лапки або апостроф, та у файл «112.txt», що містить слова, у яких більше ніж 5 приголосних букв. Зберегти цей файл у кодуванні UTF-8.
12	Зчитати файл «12.txt» та перетворити його у файл «121.txt», який складається з речень, сформованих з двох частин: у першу половину речення мають входити слова з непарного за номером речення, а у другу половину – слова з другої половини парного речення. На основі файлу «12.txt» створити також файл «122.txt», що містить слова, множина букв у яких перетинається з заданою. Зберегти цей файл у кодуванні UTF-8.
13	Зчитати файл «13.txt» та перетворити його у файл «131.txt», який складається з речень, які містять непарну кількість слів непарної довжини. На основі файлу «13.txt» створити також файл «132.txt», що містить слова, букви у яких перемішані випадковим чином. Зберегти цей файл у кодуванні UTF-8.
14	Зчитати файл «14.txt» та перетворити його у файл «141.txt», який складається з речень, слова у яких перемішані випадковим чином, крім першого та останнього слова. На основі файлу «14.txt» створити також файл «142.txt», кириличні букви у якому замінені відповідними за звучанням буквами або буквосполученнями латинського алфавіту.

15	Зчитати файл «15.txt» та перетворити його у файл «151.txt», який складається з речень, слова яких розташовані у реверсному порядку. Перше слово має завжди писатися з великої літери. На основі файлу «15.txt» створити також файл «152.txt», кожна кирилична буква у якому заміненена на букву, що йде наступною у алфавіті. М'який знак заміняємо на «а».
16	Зчитати файл «16.txt» та перетворити його у файл «161.txt», який складається з речень, кількість слів у яких є найближчою до середньої арифметичної кількості у реченнях даного тексту. На основі файлу «16.txt» створити також файл «162.txt», кожна кирилична буква у якому заміненена на букву, номер якої у алфавіті дорівнює номеру даної буква за умови, що відлік відбувається у реверсному порядку.
17	Зчитати файл «17.txt» та перетворити його у файл «171.txt», який складається з речень, що містять слово, задане шляхом вводу з клавіатури. Кожне речення має починатися з нового рядка. На основі файлу «17.txt» створити також файл «172.txt», кожна кирилична приголосна буква у якому заміненена на голосну з відповідним номером, якщо окремо лічити приголосні та голосні. Приголосні, які не мають відповідної за номером голосної, залишити без змін.
18	Зчитати файл «18.txt» та перетворити його у файл «181.txt», який складається з речень, що містять слова, які знаходяться на одному зсуві у реченнях, номери яких визначаються за модулем 10. Кожне таке речення має починатися з нового рядка і перше слово писатися з великої літери. На основі файлу «18.txt» створити також файл «182.txt», кожне слово у якому замінити числом, що дорівнює сумі його букв. Одержані послідовності чисел, що відповідають реченням, упорядкувати за спаданням.
19	Зчитати файл «19.txt» та перетворити його у файл «191.txt», який складається з речень, слова у яких упорядковані за сумою кодів символів у слові за кодуванням sr1251. Кожне таке речення має починатися з нового рядка і перше слово писатися з великої літери. На основі файлу «19.txt» створити також файл «192.txt», кожне слово у якому замінити числом, що дорівнює сумі його кодів у кодуванні utf-8. Одержані послідовності чисел, що відповідають реченням, упорядкувати за зростанням.
20	Зчитати файл «20.txt» та перетворити його у файл «201.txt», який складається з речень, перемішаних випадковим чином з наступним видаленням тих речень, сума букв у яких перевищує середню арифметичну кількість букв, пораховану по всіх реченнях даного тексту. Кожне таке речення має починатися з нового рядка і перше слово писатися з великої літери. На основі файлу «20.txt» створити також файл «202.txt», кожену голосну букву у якому замінити випадково вибраною іншою голосною буквою.

21	Зчитати файл «21.txt» та перетворити його у файл «211.txt», який складається з речень, у яких кожне парне слово замінено на випадковим чином вибране слово з наступного речення. Кожне таке речення має починатися з нового рядка і перше слово писатися з великої літери. На основі файлу «21.txt» створити також файл «212.txt», кожному приголосну букву у якому замінити випадково вибраною приголосною буквою з наступного за номером слова.
22	Зчитати файл «22.txt» та перетворити його у файл «221.txt», що складається з речень, у яких букви у словах перемішані випадковим чином. Кожне таке речення має починатися з нового рядка і перше слово писатися з великої літери. На основі файлу «22.txt» створити також файл «222.txt» з словами, у яких порядок приголосних букв замінено на реверсний.
23	Зчитати файл «23.txt» та перетворити його у файл «231.txt», що складається з речень, кожне з яких утворено шляхом виконання операції об'єднання множин букв у відповідних за номером слова кожного непарного та парного речення. Кожне таке речення має починатися з нового рядка і перше слово писатися з великої літери. На основі файлу «23.txt» створити також файл «232.txt», кожному приголосну букву у якому замінити випадково вибраною голосною, а кожному голосну – випадково вибраною приголосною.
24	Зчитати файл «24.txt» та перетворити його у файл «241.txt», що складається з речень, кожне з яких утворено шляхом виконання операції об'єднання множин букв у відповідних за номером непарних слів з непарних речень та парних слів з парних речень. Кожне таке речення має починатися з нового рядка і перше слово писатися з великої літери. На основі файлу «24.txt» створити також файл «242.txt», кожному приголосну букву у якому замінити випадково вибраною голосною з наступного слова, а кожному голосну – випадково вибраною приголосною з попереднього слова.
25	Зчитати файл «25.txt» та перетворити його у файл «251.txt», що складається з речень, які утворені шляхом виконання операції XOR множин букв парних і непарних слів у кожному реченні (1 з 2, 3 з 4 і т. д.). Кожне таке речення має починатися з нового рядка і перше слово писатися з великої літери. На основі файлу «25.txt» створити також файл «252.txt», у якому букви замінити символами, які утворені шляхом виконання побітової операції «&» над кодами сусідніх букв в кодуванні cp1251. Результат вивести у кодуванні utf-8.
26	Зчитати файл «26.txt» та перетворити його у файл «261.txt», що складається з речень, які утворені шляхом виконання операції OR над множинами букв парних слів у кожному реченні (2 з 4, 6 з 8 і т.д.). Кожне таке речення має починатися з нового рядка і перше слово писатися з великої літери. На основі файлу «26.txt» створити також файл «262.txt», у якому букви замінити символами, які

	утворені шляхом виконання побітової операції зсуву вправо на дві позиції в кодуванні cp1251. Результат вивести у кодуванні utf-8.
27	Зчитати файл «27.txt» та розбити його на 2 файли «271.txt», «272.txt», які містять непарні та парні за номером символи початкового файлу. Виконати операцію «&» над множинами символів цих файлів. Зберегти результуючий файл «273.txt» у кодуванні «cp1251». Створити файл з символів, які присутні у файлі «27.txt», але відсутні у «273.txt». Зберегти цей файл у кодуванні UTF-8.
28	Зчитати файл «28.txt» та перетворити його у файл «281.txt», який складається з речень, упорядкованих за зростанням кількості символів. Також створити файл «282.txt», який містить слова, що утворені шляхом виконання операції «&» над множинами символів відповідних за номером слів із файлів «28.txt» та «281.txt». Зберегти цей файл у кодуванні UTF-8.
29	Зчитати файл «29.txt» та розбити файл на три файли однакового розміру, створивши файли «291.txt», «292.txt» та «293.txt» з кодуванням UTF-8. Створити файл «294.txt», кожне слово у якому утворено шляхом виконання операції об'єднання множин символів відповідних за номером слів з файлів «291.txt», «292.txt» та «293.txt».

Титульний лист

1. Титульний лист повинен мати такий вигляд:

Національний технічний університет України
«Київський політехнічний інститут ім. І. Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Програмування

Лабораторна робота №7

«Обробка виключень та робота з файлами»

Виконав:
студент групи ІО-XX
П.І.Б. студента
Номер у списку групи: XX
Перевірив:
П.І.Б. викладача.

Київ 2023р.

Вміст звіту

1. Мета лабораторної роботи та загальне завдання
2. Короткі теоретичні відомості, які відображають інформацію про виключення та елементи роботи з файлами, що використані при написанні лабораторної роботи.
3. Скріншот вашого варіанту.
4. Алгоритм його розв'язування (у довільній формі).
5. Роздруківка того фрагменту тексту програми, який написаний індивідуально чорними символами на білому фоні.
6. Скріншот результатів виконання програми з контрольним прикладом
7. Аналіз результатів та висновки.

Лабораторна роботи № 8.

Тема: Розробка програми з графічним інтерфейсом на основі бібліотеки tkinter

Мета роботи: ознайомитися з організацією графічного інтерфейсу на основі бібліотеки *tkinter*.

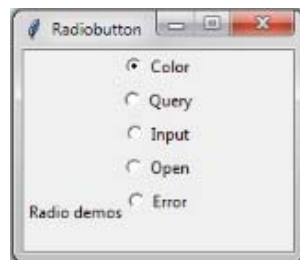
Графічний інтерфейс (GUI) та його елементи. Модуль *tkinter*

Теоретичні відомості

Приклад 1. Радіокнопка

```
from tkinter import *
ventana = Tk()
v = IntVar(); m = IntVar()
ventana.title("Radiobutton")
ventana.geometry("200x150")
etiquental = Label(ventana, text="Radio demos").place(x=1, y=110)
Color = Radiobutton(ventana, text="Color", variable=v, value=0).pack()
Query = Radiobutton(ventana, text="Query", variable=v, value=1).pack()
Input = Radiobutton(ventana, text="Input", variable=v, value=2).pack()
Open = Radiobutton(ventana, text="Open", variable=v, value=3).pack()
Error = Radiobutton(ventana, text="Error", variable=v, value=4).pack()
ventana.mainloop()
```

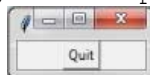
Результат:



Приклад 2. Кнопка(два модулі)

Модуль1: l8_2.py

```
from tkinter import *
from tkinter.messagebox import askokcancel
class Quitter(Frame):
    def __init__(self, parent=None):
        Frame.__init__(self, parent)
        self.pack()
        widget = Button(self, text='Quit', command=self.quit)
        widget.pack(side=RIGHT, expand=YES, fill=BOTH)
    def quit(self):
        ans = askokcancel('Verify exit', "Really quit?")
        if ans: Frame.quit(self)
if __name__ == '__main__': Quitter().mainloop()
```



Приклад 3.

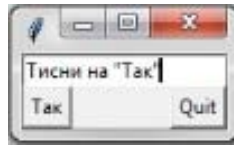
```
from tkinter import *
from l8_2 import Quitter
def fetch():
    print('Input => "%s"' % ent.get())
root = Tk()
ent = Entry(root)
ent.insert(0, 'Тисни на "Так"')
ent.pack(side=TOP, fill=X)
ent.focus()
ent.bind('<Return>', (lambda event: fetch()))
```

```

btn = Button(root, text='Так', command=fetch)
btn.pack(side=LEFT)
Quitter(root).pack(side=RIGHT)
root.mainloop()

```

Результат:



Input => "Тисни на "Так""

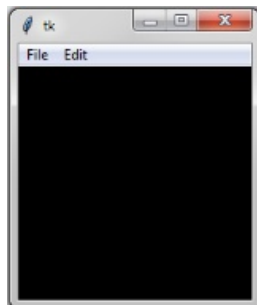
Приклад 4. Меню

```

from tkinter import *
root = Tk()
def sq1():
    fra.config(width=200,height=200)
    print('200x200')
def sq2():
    fra.config(width=400,height=400)
    print('4000x4000')
fra = Frame(root,width=300,height=100,bg="Black")
fra.pack()
m = Menu(root)
root.config(menu=m)
cm = Menu(m)
m.add_cascade(label="File",menu=cm)
sm = Menu(m)
m.add_cascade(label="Edit",menu=sm)
sm.add_command(label="200x200",command=sq1)
sm.add_command(label="4000x4000",command=sq2)
root.mainloop()

```

Результат:



Приклад 5.

```

from tkinter.filedialog import *
import fileinput
from tkinter.messagebox import *
def close_win():
    if askyesno("Exit", "Do you want to quit?"):
        root.destroy()
def about():
    showinfo("Editor", "This is text editor.\n(test version)")
def _open():
    op = askopenfilename()
    for l in fileinput.input(op):
        txt.insert(END,l)
def _save():
    sa = asksaveasfilename()
    letter = txt.get(1.0,END)
    f = open(sa,"w")
    f.write(letter)
    f.close()
root = Tk()

```

```

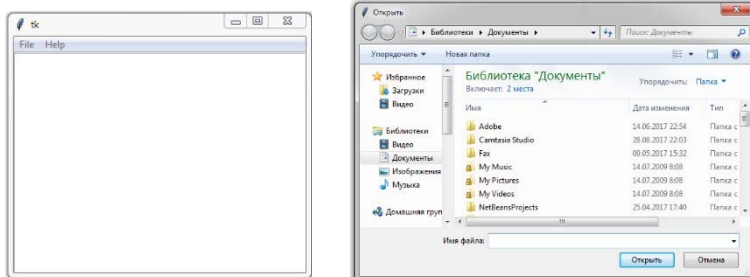
m = Menu(root)

root.config(menu=m)
fm = Menu(m)
m.add_cascade(label="File", menu=fm)
fm.add_command(label="Open...", command=_open)
fm.add_command(label="Save...", command=_save)
fm.add_command(label="Exit", command=close_win)

hm = Menu(m)
m.add_cascade(label="Help", menu=hm)
hm.add_command(label="About", command=about)
txt = Text(root, width=40, height=15, font="12")
txt.pack()
root.mainloop()

```

Результат



Приклад 6. Слуга прокрутки

```

from tkinter import *
def printer(event):
    x = var.get()
    print(x)
root = Tk()
var = IntVar()
var.set(5)
sca1 = Scale(root, orient=VERTICAL, length=200,
             from_=0, to=10, tickinterval=2, resolution=1,
             variable=var)
sca2 = Scale(root, orient=HORIZONTAL, length=200,
             from_=0, to=10, tickinterval=2, resolution=1,
             variable=var)
lab=Label(root, text='Лабораторна робота \n Graphical User Interface',
          font='Arial 18')
but=Button(root, text='Дізнатись значення змінної', width=30, height=3,
           bg='grey', fg='red', font='Arial 12')
but.bind('<Button-1>', printer)
lab.pack()
sca1.pack()
sca2.pack()
but.pack()
root.mainloop()

```

Результат

5

Приклад 7.

```

from tkinter import *
root = Tk()
myscrollbar = Scrollbar(root)
myscrollbar.pack(side=RIGHT, fill=Y)
mylistbox = Listbox(root, yscrollcommand = myscrollbar.set)
for a in range(20):

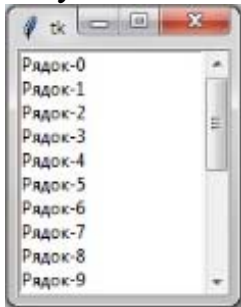
```

```

mylistbox.insert(END, 'Рядок-' + str(a))
mylistbox.pack(side=LEFT, fill=BOTH)
myscrollbar.config(command=mylistbox.yview)
root.mainloop()

```

Результат :



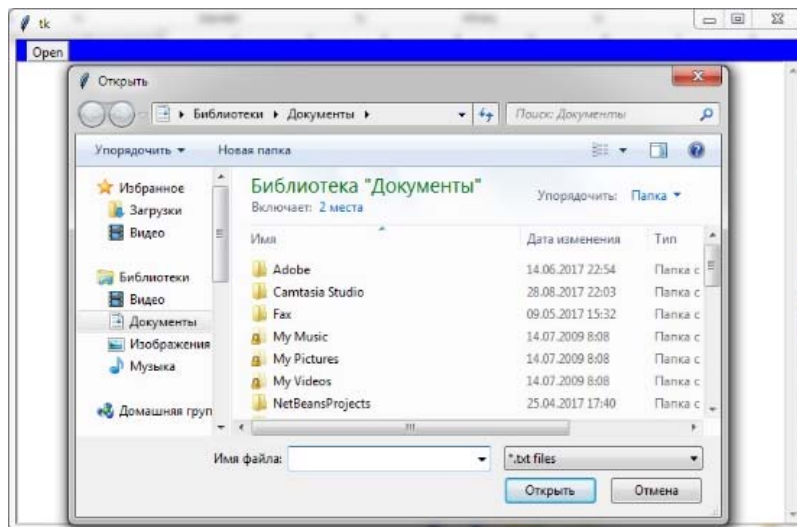
Приклад 8. Завантаження текстового файлу у редактор

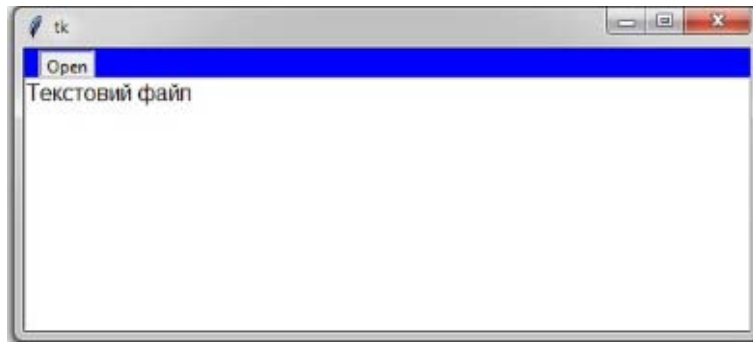
```

from tkinter import *
import tkinter.filedialog
def LoadFile(ev):
    fn=tkinter.filedialog.Open(root, filetypes=[('.*txt files', '.txt')]).show()
    if fn == '':
        return
    textbox.delete('1.0', 'end')
    textbox.insert('1.0', open(fn, 'rt').read())
root = Tk()
panelFrame = Frame(root, height = 20, bg = 'blue')
textFrame = Frame(root, height = 40, width = 50)
panelFrame.pack(side = 'top', fill = 'x')
textFrame.pack(side = 'bottom', fill = 'both', expand = 1)
textbox = Text(textFrame, font='Arial 12', wrap='word')
scrollbar = Scrollbar(textFrame)
scrollbar['command'] = textbox.yview
textbox['yscrollcommand'] = scrollbar.set
textbox.pack(side = 'left', fill = 'both', expand = 1)
scrollbar.pack(side = 'right', fill = 'y')
loadBtn = Button(panelFrame, text = 'Open')
loadBtn.bind("<Button-1>", LoadFile)
loadBtn.place(x = 10, y = 1, width = 40, height = 20)
root.mainloop()

```

Результат





Приклад 9. Геометричні фігури. При натисканні на кнопку («Трикутник», «Прямокутник» тощо) на полотні з'являється відповідна фігура, а в текстовому полі її визначення. При клацанні на фігурі, яка розташована на полотні, вона змінюється на іншу, що теж відноситься до тієї ж групи (наприклад, трикутники – прямокутні або рівносторонні). Внизу полотна з'являються пояснення, що характеризують чергову фігуру (наприклад, «прямокутний трикутник»). Можна отримати частини фігури (для трикутника це сторони), щоб продемонструвати, як вони утворюються

```
from tkinter import *
win = Tk()

i = 0
def add_triangle(event):
    coords = [(50, 130, 290, 40, 170, 250), (10, 10, 290, 30, 200, 250),
              (30, 280, 330, 60, 300, 200), (50, 200, 340, 200, 110, 60)]
    colors = ['red', 'green', 'blue', 'yellow']
    global i
    canvas.itemconfig(t, fill=colors[i], outline='white')
    canvas.coords(t, coords[i])
    i += 1
    if i == 4:
        i = 0
def triangle():
    canvas.coords(r, (0, 0, 0, 0))
    canvas.itemconfig(t, fill='yellow', outline='white')
    canvas.coords(t, (50, 200, 340, 200, 110, 60))
    text.delete(1.0, END)
    text.insert(1.0, 'Трикутник -\nце геометрична фігура, \
яка утворена трьома відрізками, які з\'єднують три точки, \
що не лежать на одній прямій.')
    text.tag_add('title', '1.0', '1.end')
    text.tag_config('title', font=('Times', 14), foreground='red')
def rectangle():
    canvas.coords(t, (0, 0, 0, 0, 0, 0))
    canvas.itemconfig(r, fill='lightblue', outline='white')
    canvas.coords(r, (80, 50, 360, 200))

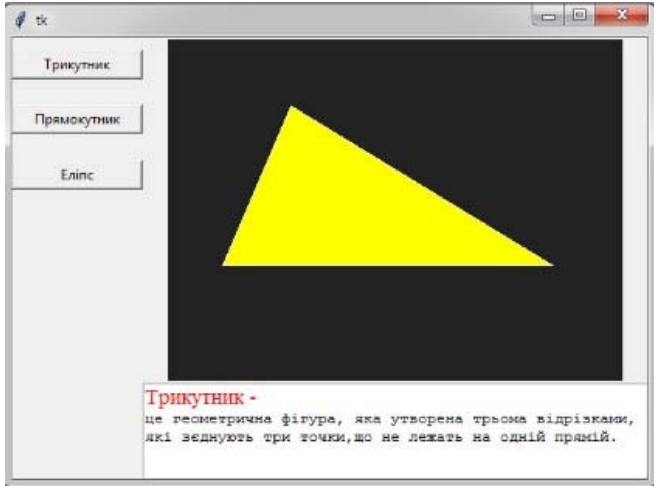
def ellipse():
    pass

b_triangle = Button(text="Трикутник", width=15, command=triangle)
b_rectangle = Button(text="Прямокутник", width=15, command=rectangle)
b_ellipse = Button(text="Еліпс", width=15, command=ellipse)
canvas = Canvas(width=400, height=300, bg='#222222')
```

```

text = Text(width=55,height=5,bg='#ffffff',wrap=WORD)
t = canvas.create_polygon(0,0,0,0,0,0)
canvas.tag_bind(t, '<Button-1>', add_triangle)
r = canvas.create_rectangle(0,0,0,0)
b_triangle.grid(row=0, column=0)
b_rectangle.grid(row=1, column=0)
b_ellipse.grid(row=2, column=0)
canvas.grid(row=0, column=1, rowspan=10)
text.grid(row=11, column=1, rowspan=3)
win.mainloop()

```



Приклад 10.

#Аварійний біп

```
from tkinter import *
```

```

class Alarm(Frame):
    def repeater(self):
        self.bell()
        self.after(self.msecs, self.repeater)
    def __init__(self):
        Frame.__init__(self)
        self.msecs = 1000
        self.pack()
        stopper = Button(self, text='Stop the beeps!', command=self.quit)
        stopper.pack()
        stopper.config(bg='navy', fg='white', bd=8)
        self.stopper = stopper
        self.repeater()

```

```

if __name__ == '__main__':
    Alarm().mainloop()

```



Приклад 11.

#Кнопки з кольорами з використанням подій

```
from tkinter import *
```

```

class curry:
    """from Scott David Daniels' recipe
    "curry -- associating parameters with a function"
    in the "Python Cookbook"
    http://aspn.activestate.com/ASPN/Python/Cookbook/

```



```

"""

def __init__(self, fun, *args, **kwargs):
    self.fun = fun
    self.pending = args[:]
    self.kwargs = kwargs.copy()

def __call__(self, *args, **kwargs):
    if kwargs and self.kwargs:
        kw = self.kwargs.copy()
        kw.update(kwargs)
    else:
        kw = kwargs or self.kwargs
    return self.fun(*(self.pending + args), **kw)

def event_lambda(f, *args, **kwds ):
    """A helper function that wraps lambda in a prettier interface.
    Thanks to Chad Netzer for the code."""
    return lambda event, f=f, args=args, kwds=kwds : f( *args, **kwds )
# ----- code for function: event_lambda (end) -----

class MyApp:
    def __init__(self, parent):
        self.myParent = parent
        self.myContainer1 = Frame(parent)
        self.myContainer1.pack()

        button_name = "OK"

        # command binding -- using curry
        self.button1 = Button(self.myContainer1,
            command = curry(self.buttonHandler, button_name, 1, "Good stuff!"))

        # event binding -- using the event_lambda helper function
        self.button1.bind("<Return>",
            event_lambda( self.buttonHandler, button_name, 1, "Good stuff!" ) )

        self.button1.configure(text=button_name, background="green")
        self.button1.pack(side=LEFT)
        self.button1.focus_force() # Put keyboard focus on button1

        button_name = "Cancel"

        # command binding -- using curry
        self.button2 = Button(self.myContainer1,
            command = curry(self.buttonHandler, button_name, 2, "Bad stuff!"))

        # event binding -- using the event_lambda helper function in two steps
        event_handler = event_lambda( self.buttonHandler, button_name, 2, "Bad
stuff!" )
        self.button2.bind("<Return>", event_handler )

        self.button2.configure(text=button_name, background="red")
        self.button2.pack(side=LEFT)

    def buttonHandler(self, argument1, argument2, argument3):
        print ("    buttonHandler routine received arguments:", \
            argument1.ljust(8), argument2, argument3)

    def buttonHandler_a(self, event, argument1, argument2, argument3):
        print ("buttonHandler_a received event", event)

```

```

        self.buttonHandler(argument1, argument2, argument3)

root = Tk()
myapp = MyApp(root)
root.mainloop()

```



Приклади задавання таблиць

Приклад 12. Найпростіша таблиця

simple 2d table

```

from tkinter import *

for i in range(6):
    for j in range(6):
        l = Label(text='%d.%d' % (i, j), relief=RIDGE)
        l.grid(row=i, column=j, sticky=NSEW)

mainloop()

```



Приклад 13. Таблиця, що складається з віджетів Label та Entry

```

from tkinter import *
colors = ['red', 'green', 'orange', 'white', 'yellow', 'blue']

r = 0
for c in colors:
    Label(text=c, relief=RIDGE, width=25).grid(row=r, column=0)
    Entry(bg=c, relief=SUNKEN, width=50).grid(row=r, column=1)
    r = r+1

mainloop()

```



Приклад 14. Таблиця з полями для вводу даних

```

from tkinter import *

rows = []
for i in range(5):
    cols = []
    for j in range(4):

```

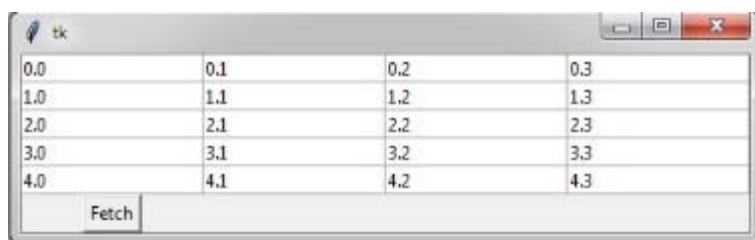
```

e = Entry(relief=RIDGE)
e.grid(row=i, column=j, sticky=NSEW)
e.insert(END, '%d.%d' % (i, j))
cols.append(e)
rows.append(cols)

def onPress():
    for row in rows:
        for col in row:
            print(col.get()),

Button(text='Fetch', command=onPress).grid()
mainloop()

```



Приклад 15. Таблиця з використанням пакувальника grid

```
from tkinter import *
```

```

class GridDemo( Frame ):
    def __init__( self ):
        Frame.__init__( self )
        self.master.title( "Демо з пакувальником grid" )

        self.master.rowconfigure( 0, weight = 1 )
        self.master.columnconfigure( 0, weight = 1 )
        self.grid( sticky = W+E+N+S )

        self.text1 = Text( self, width = 15, height = 5 )

        self.text1.grid( rowspan = 3, sticky = W+E+N+S )
        self.text1.insert( INSERT, "Текст 1" )

        self.button1 = Button( self, text = "Кнопка 1", width = 25 )
        self.button1.grid( row = 0, column = 1, columnspan = 2, sticky = W+E+N+S )

        self.button2 = Button( self, text = "Кнопка 2" )
        self.button2.grid( row = 1, column = 1, sticky = W+E+N+S )

        self.button3 = Button( self, text = "Кнопка 3" )
        self.button3.grid( row = 1, column = 2, sticky = W+E+N+S )

        self.button4 = Button( self, text = "Кнопка 4" )
        self.button4.grid( row = 2, column = 1, columnspan = 2, sticky = W+E+N+S )

        self.entry = Entry( self )
        self.entry.grid( row = 3, columnspan = 2, sticky = W+E+N+S )
        self.entry.insert( INSERT, "Редактор" )

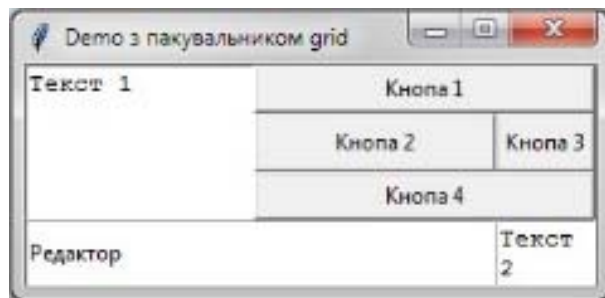
        self.text2 = Text( self, width = 2, height = 2 )
        self.text2.grid( row = 3, column = 2, sticky = W+E+N+S )
        self.text2.insert( INSERT, "Текст 2" )

        self.rowconfigure( 1, weight = 1 )
        self.columnconfigure( 1, weight = 1 )

```

```
def main():
    GridDemo().mainloop()

if __name__ == "__main__":
    main()
```



Приклад 16. Використання мишки для малювання фігур на canvas

```
from tkinter import *
trace = 0

class CanvasEventsDemo:
    def __init__(self, parent=None):
        canvas = Canvas(width=300, height=300, bg='beige')
        canvas.pack()
        canvas.bind('<ButtonPress-1>', self.onStart)
        canvas.bind('<B1-Motion>', self.onGrow)
        canvas.bind('<Double-1>', self.onClear)
        canvas.bind('<ButtonPress-3>', self.onMove)
        self.canvas = canvas
        self.drawn = None
        self.kinds = [canvas.create_oval, canvas.create_rectangle]

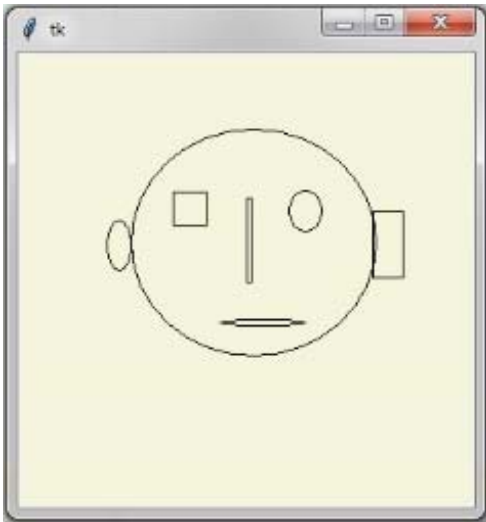
    def onStart(self, event):
        self.shape = self.kinds[0]
        self.kinds = self.kinds[1:] + self.kinds[:1]
        self.start = event
        self.drawn = None

    def onGrow(self, event):
        canvas = event.widget
        if self.drawn: canvas.delete(self.drawn)
        objectId = self.shape(self.start.x, self.start.y, event.x, event.y)
        if trace: print(objectId)
        self.drawn = objectId

    def onClear(self, event):
        event.widget.delete('all')

    def onMove(self, event):
        if self.drawn:
            if trace: print(self.drawn)
            canvas = event.widget
            diffX, diffY = (event.x - self.start.x), (event.y - self.start.y)
            canvas.move(self.drawn, diffX, diffY)
            self.start = event

if __name__ == '__main__':
    CanvasEventsDemo().mainloop()
```



Приклад 17. Таймер на віджеті Label

```
import tkinter
import time

curtime = ''
clock = tkinter.Label()
clock.pack()

def tick():
    global curtime
    newtime = time.strftime('%H:%M:%S')
    if newtime != curtime:
        curtime = newtime
        clock.config(text=curtime)
    clock.after(200, tick)

tick()
clock.mainloop()
```



1.8. Загальний порядок виконання лабораторної роботи

1. Ознайомитися з теоретичним матеріалом. Опрацювати приклади.
 2. Відповідно до свого варіанту визначити логічний вираз;
 - написати програму, яка розв'язує завдання - за допомогою стандартної бібліотеки *tkinter* запрограмувати відповідний графічний інтерфейс.
 3. Скласти звіт і захистити його по роботі.
- Захист роботи включає в себе демонстрацію працездатності програми на різних вхідних даних.

1.9. Вимоги до інтерфейсу

А) Програма повинна складатися з 3-х вікон.

Вміст вікна №1

1. Головне меню, яке повинно включати меню виклику вікна №2, вікна №3.
2. Віджети виводу П.І.Б студента, номера групи, номера у списку та віджет виводу результатів обчислення варіанту відповідно до програми, що задана у завданні – пункт загального порядку виконання лабораторної роботи).
3. Віджети для задавання кількості елементів множин A, B і C.
4. Віджети для формування випадковим чином множин A, B і C з заданою кількістю елементів.
5. Віджети, що дають можливість ручного вводу множин A, B і C.
6. Віджет для задавання діапазону цілих чисел, з яких будемо вибирати елементи множин.

Вміст вікна №2

1. Віджети для відображення елементів множин A, B і C.
2. Віджети запуску покрокового виконання виразу у відповідності з варіантом. Одним кроком вважати виконання однієї логічної операції.
3. Віджети відображення множин-операндів та множини-результату кожної логічної операції.
4. Віджет відображення множини D та віджет для виконання команди збереження даного результату у файлі.
5. Віджет, який містить зображення виразу 1 у відповідності з варіантом

Вміст вікна №3

1. Віджет для відображення елементів множин D.
2. Віджети для запуску випадкового генерування множини F з такими даними:
 - кількість елементів множини F має дорівнювати кількості елементів множини D;
 - мінімальний елемент множини F має дорівнювати мінімальному елементу множини D;
 - максимальний елемент множини F має дорівнювати максимальному елементу множини D.
3. Віджет для відображення елементів множин F.
4. Віджет відображення множини X, та віджет для виконання команди збереження даного результату у файлі.
5. Віджет, який містить зображення виразу 2 у відповідності з варіантом

Варіанти виконання лабораторної роботи

№ вар	Вираз для обчислення
1	$D = ((A \setminus B) \cup (B \cap A)) \setminus (C \cup B)$ $X = F \Delta D$
2	$D = A \cap (A \setminus (A \setminus B)) \cup C$ $X = F \setminus \bar{D}$

№ вар	Выраз для обчислення
3	$D = (A \cup (B \setminus A)) \setminus C$ $X = \bar{F} \cup D$
4	$D = (\bar{A} \cup \bar{B}) \cap (\bar{B} \cup \bar{C})$ $X = \bar{F} \Delta D$
5	$D = (A \cup (\bar{A} \cap B)) \Delta (C \cup (C \cap B))$ $X = F \cap \bar{D}$
6	$D = ((A \cap B) \cup (A \setminus B)) \Delta (C \cup B)$ $X = \bar{F} \setminus D$
7	$D = C \cap (A \cap \bar{B}) \cap (C \cup B)$ $X = F \setminus D$
8	$D = ((A \cap \bar{B}) \cup (B \cap \bar{A})) \cap (C \cup B) \cap C$ $X = \bar{F} \cup \bar{D}$
9	$D = (((A \cup \bar{A}) \cap A) \setminus B \cup B) \cap (C \cup (C \cap B))$ $X = \bar{F} \cap \bar{D}$
10	$D = (A \cap B) \cup ((A \cap \bar{C}) \cup (\bar{A} \cap B))$ $X = F \cup \bar{D}$
11	$D = A \cap (A \cup \bar{B}) \cap (C \cup (\bar{C} \cap B))$ $X = \bar{F} \cap D$
12	$D = ((A \cap \bar{B}) \cup B \setminus A) \cup C$ $X = \bar{F} \setminus \bar{D}$
13	$D = (\bar{A} \cap (A \cup \bar{B})) \setminus C$ $X = \bar{F} \Delta \bar{D}$
14	$D = (A \cap (B \cup A)) \cap (\overline{C \cup B}) \cap \bar{C}$ $X = \bar{F} \setminus D$
15	$D = (A \cup (B \cap C)) \cap (\bar{A} \cup C)$ $X = F \Delta D$
16	$D = A \Delta (B \setminus ((C \cup A) \cap (C \cup \bar{A})))$ $X = F \setminus \bar{D}$
17	$D = (A \cup B) \cap (\bar{A} \cap B) \cup (\overline{\bar{B} \cup C})$ $X = \bar{F} \cap D$

№ вар	Вираз для обчислення
18	$D = \bar{A} \cup \bar{B} \cup (\bar{A} \cap B) \cup (\bar{B} \cap C) \cup \bar{C}$ $X = \bar{F} \Delta \bar{D}$
19	$D = C \cup (\bar{A} \cap B) \cap (\bar{B} \cap \bar{A}) \cap (\bar{A} \cup B)$ $X = \bar{F} \cap \bar{D}$
20	$D = ((A \cup B) \cup C \cup (B \cup C) \cup A)$ $X = \bar{F} \Delta D$
21	$D = A \cap (\bar{B} \cup C) \cup (\bar{A} \cap C)$ $X = F \cap \bar{D}$
22	$D = B \Delta C \cup (B \cap C) \Delta (((A \setminus B) \cap B) \Delta A)$ $X = F \setminus D$
23	$D = ((A \cup B) \cup (A \cup \bar{B})) \cap \bar{B} \cap A \cap (\bar{A} \cup C)$ $X = \bar{F} \Delta D$
24	$D = \bar{A} \cup B \cup \bar{C} \cup (B \cap \bar{C}) \cup (\bar{A} \cap C) \cup (A \cap B)$ $X = F \setminus D$
25	$D = (A \cap B) \cup (A \cap \bar{B}) \cup (C \cup A) \cap (A \cup \bar{B})$ $X = \bar{F} \cap \bar{D}$
26	$D = \bar{A} \cup (\bar{A} \cup B) \cap (\bar{A} \cup C) \cup \bar{B} \cap ((B \cap C) \cup (B \cap \bar{C}))$
27	$D = (A \cap B) \cup (C \cap B) \cup (\bar{A} \cap \bar{B}) \cup (\bar{B} \cap C)$ $X = \bar{F} \setminus D$
28	$D = (\overline{A \cup B}) \cap (\overline{A \cup C}) \cap (\overline{B \cup C})$ $X = F \cup \bar{D}$
29	$D = ((A \cap \bar{B}) \cup (\bar{A} \cap B)) \cap (\bar{C} \cap (\bar{C} \cup B))$ $X = \bar{F} \cap D$
30	$D = \bar{C} \cap (A \setminus C) \cap (B \setminus C) \cap (\bar{C} \cup B)$ $X = \bar{F} \cup \bar{D}$

Титульний лист

2. Титульний лист повинен мати такий вигляд:

Національний технічний університет України
«Київський політехнічний інститут ім. І. Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Програмування

Лабораторна робота №8

«Бібліотека tkinter»

Виконав:
студент групи ІО-XX
П.І.Б. студента
Номер у списку групи: XX
Перевірив:
П.І.Б. викладача.

Київ 2023р.

Вміст звіту

8. Мета лабораторної роботи та загальне завдання
9. Короткі теоретичні відомості, які відображають інформацію про виключення та елементи роботи з файлами, що використані при написанні лабораторної роботи.
- 10.Скріншот вашого варіанту.
- 11.Алгоритм його розв'язування (у довільній формі).
- 12.Роздруківка того фрагменту тексту програми, який написаний індивідуально чорними символами на білому фоні.
- 13.Скріншот результатів виконання програми з контрольним прикладом
- 14.Аналіз результатів та висновки.