Міністерство освіти і науки України

Національний університет „Львівська політехніка”

Кафедра “Електронних обчислювальних машин”



**з дисципліни**

**Кросплатформні засоби програмування**

**Звіт з Лабораторної роботи №7**

**ДОСЛІДЖЕННЯ БАЗОВИХ КОНСТРУКЦІЙ МОВИ PYTHON**

**Виконав:**

КІ-305

Гунька А.А.

**Перевірив:**

Іванов Ю.С

**2023**

**Мета роботи:** ознайомитися з базовими конструкціями мови PYTHONта оволодіти навиками написання й автоматичного документування простих консольних програм мовою PYTHON.

**Теоретичний вступ**

Python програми – це набір \*.py файлів. Кожен файл – це окремий модуль. Модулі

підключаються за допомогою ключового слова import після якого вказується назва

файлу без розширення py. Щоб доступитися з одного модуля до іншого вони мають бути в

одному каталозі. Якщо модуль лежить в іншому каталозі, то щоб до нього доступитися

цей каталог має бути оформлений як пакет. Для цього достатньо у нього помістити

порожній файл \_\_init\_\_.py. При підключені пакету виконується вміст файлу \_\_init\_\_.py,

який може містити визначення імен і інші необхідні для роботи з пакетом конфігураційні

дії.

**Коментарі**

Python має лише рядкові коментарі. Коментарем у Python є текст після символа ‘#’:

# Comment

**Форматування коду**

Код у Python виділяється у блоки за допомогою 4-ох пробілів відносно

попереднього блоку або одного символа табуляції. Пробіли і табуляцію не можна

змішувати.

**Запуск на виконання програми мовою Python**

Для запуску на виконання програми мовою Python слід виконати в командному

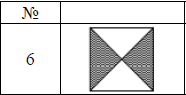
рядку: python.exe <file name>.py. Запустивши інтерпретатор Python.exe, можна вводити з

командного рядка програму по-рядково і зразу отримувати результат виконання.

Типи даних, оголошеня змінних та операції над ними

Python підтримує наступні типи даних.

**Варіант 6**



**Завдання:**

1. Написати та налагодити програму на мові Python згідно варіанту. Програма має

задовольняти наступним вимогам:

• програма має розміщуватися в окремому модулі;

• програма має генерувати зубчатий список, який міститиме лише заштриховані

області квадратної матриці згідно варіанту;

• розмір квадратної матриці і символ-заповнювач масиву вводяться з клавіатури;

• при не введені або введенні кількох символів-заповнювачів відбувається коректне

переривання роботи програми;

• сформований масив вивести на екран;

• програма має містити коментарі.

2. Завантажити код на GitHub згідно методичних вказівок по роботі з GitHub.

3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її

виконання та фрагменту згенерованої документації та завантажити його у ВНС.

4. Дати відповідь на контрольні запитання.

**Код**

def main():

matrix\_size = int(input("Введіть розмір квадратної матриці: "))

# Перевірка, чи введено коректний розмір матриці

if matrix\_size <= 0:

print("Некоректний розмір матриці. Програма завершує роботу.")

return

fill\_character = input("Введіть символ-заповнювач: ")[0]

matrix = []

pos = 0

for i in range(matrix\_size):

row = []

if i >= matrix\_size / 2:

if i != matrix\_size / 2:

pos = pos - 1

else:

pos = pos + 1

for j in range(matrix\_size):

if j < pos or j > matrix\_size - pos - 1:

row.append(fill\_character)

matrix.append(row)

pos = 0

for i in range(matrix\_size):

if i >= matrix\_size / 2:

if i != matrix\_size / 2:

pos = pos - 1

else:

pos = pos + 1

for j in range(matrix\_size):

if j < pos or j > matrix\_size - pos - 1:

print(f"{fill\_character} ", end="")

else:

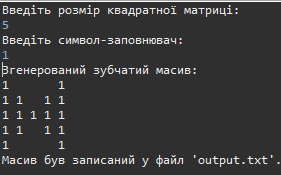
print(" ", end="")

print("")

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Результат:**



**Короткий висновок**

Під час виконання даної лабораторної роботи я ознайомився з базовими конструкціями мови PYTHONта оволодіти навиками написання й автоматичного документування простих консольних програм мовою PYTHON.