

**Міністерство освіти і науки України**  
**Національний університет «Львівська політехніка»**

**Кафедра ЕОМ**



**Звіт**  
до лабораторної роботи № 8  
з дисципліни: «Кросплатформні засоби програмування»  
на тему: «Файли та виключення у Python»  
Варіант №13

Виконав:  
ст. гр. КІ-203  
Панченко Д. В.

Прийняв:  
доцент кафедри ЕОМ  
Іванов Ю. С.

Львів 2023

**Мета:** оволодіти навиками використання засобів мови Python для роботи з файлами.

### Завдання:

1. Написати та налагодити програму на мові Python згідно варіанту. Програма має задовольняти наступним вимогам:
  - програма має розміщуватися в окремому модулі;
  - програма має реалізувати функції читання/запису файлів у текстовому і двійковому форматах результатами обчислення виразів згідно варіанту;
  - 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.

**Варіант завдання:** 13 –  $y = \sin(x) / \text{ctg}(8x)$

### Вихідний код програми:

```
# -*- coding: cp1251 -*-

import os
import struct
import sys
import math

# Запис даних в текстовий файл
def writeResTxt(fName, result):
    with open(fName, 'w') as f:
        f.write(str(result))

# Читання даних з текстового файлу
def readResTxt(fName):
    result = 0.0
    with open(fName, 'r') as f:
        result = f.read()
    return result

# Запис даних в бінарний файл
def writeResBin(fName, result):
    with open(fName, 'wb') as f:
        f.write(struct.pack('f', result))

# Читання даних з бінарного файлу
def readResBin(fName):
    result = 0.0
    with open(fName, 'rb') as f:
        result = struct.unpack('f', f.read())[0]
    return result

# Обчислення виразу
def calculate(x):
    return math.sin(x) * math.tan(8*x)

# Точка входу в програму main
```

```

if __name__ == "__main__":
    data = float(input("Введіть число: "))
    result = calculate(data)
    print(f"\nРезультат для sin(x)/ctg(8x): {result}")
    try:
        writeResTxt("textRes.txt", result)
        writeResBin("binRes.bin", result)
        print("\nРезультат з бінарного файлу: {0}".format(readResBin("binRes.bin")))
        print("\nРезультат з текстового файлу: {0}\n".format(readResTxt("textRes.txt")))
    except FileNotFoundError as e:
        print(e)
        sys.exit(1)

```

## Результат виконання програми:

```

C:\Windows\system32\cmd.exe
Введіть число: 50

Результат для sin(x)/ctg(8x): -0.42501694010682595

Результат з бінарного файлу: -0.42501693964004517

Результат з текстового файлу: -0.42501694010682595

Press any key to continue . . .

```

## Відповіді на контрольні запитання:

1. За допомогою якої конструкції у мові Python обробляються виключні ситуації?

Мова Python має вбудований механізм обробки виключних ситуацій. Обробка виключних ситуацій забезпечується блоками try-except-finally.

2. Особливості роботи блоку except?

Блок except не є обов'язковим, за умови, що визначено блок finally. Він містить код обробки виключної ситуації. Він може приймати перелік класів-виключень при генерації об'єктів яких буде виконане тіло даного блоку або бути порожнім.

3. Яка функція використовується для відкривання файлів у Python?

Функція open. Вона повертає дескриптор відкритого файлу або None.

4. Особливості використання функції open?

Open (file, mode='r', buffering=-1, encoding=None, errors=None, newline=None, closefd=True, opener=None). Параметри функції:

- file – шлях до файлу
- mode – режим відкривання файлу
- buffering — необов'язкове ціле число, яке використовується для встановлення політики буферизації

- encoding – задає назву кодування тексту
- errors – необов’язковий рядок, який визначає, як мають оброблятися помилки кодування та декодування
- newline – визначає, як аналізувати символи нового рядка з потоку
- opener – посилання на користувацьку функцію, яка буде викликана для відкривання файлу, замість стандартної.

#### 5. В яких режимах можна відкрити файл?

Параметр	Значення
'r'	Відкрити для читання (за замовчуванням)
'w'	Відкрити для запису, очистивши попередньо файл, якщо файл існує
'x'	Відкрити для ексклюзивного створення, якщо файл уже існує, то функція завершується невдачею
'a'	Відкрити для запису, дописуючи в кінець файлу, якщо він існує
'b'	Бінарний режим
't'	Текстовий режим (за замовчуванням)
'+'	Відкрити для оновлення (читання та запис)

#### 6. Як здійснити читання і запис файлу?

Читання з файлів здійснюється за допомогою методу read об’єкту-файлу. Запис у файл здійснюється за допомогою методу write об’єкту-файлу.

#### 7. Особливості функцій у мові Python?

Функції у мові python не відрізняються за своєю суттю від функцій C/C++. Синтаксис оголошення функцій:

```
def function_name({параметри}):
    [оператори]
```

#### 8. Для чого призначений оператор with?

Оператор with використовується для автоматизації процесів закриття ресурсу і коректної обробки виключних ситуацій (аналог оператора try-з-ресурсами у Java).

#### 9. Які вимоги ставляться до об’єктів, що передаються під контроль оператору with?

Це може бути будь-який об’єкт, клас якого містить визначені методи `__enter__` та `__exit__`, які дозволяють належним чином керувати ресурсами під час входу в блок with, виходу з нього та обробки виключних ситуацій.

#### 10. Як поєднуються обробка виключних ситуацій і оператор with?

Синтаксис:

```
with EXPRESSION as VAR:
    BLOCK
```

де - BLOCK – блок коду, який має бути виконаний з використанням TARGET будучи обгорнутим при цьому конструкцією try-except-finally.

**Висновок:** Я оволодів навиками використання засобів мови Python для роботи з файлами.