Міністерство освіти і науки України

Національний університет «Львівська політехніка»



Звіт

до лабораторної роботи №8

з дисципліни: “Кросплатформні засоби програмування”

на тему: «Файли та виключення у Python»

Варіант №12

Виконала:

ст.гр. КІ-307

Коваль К. О.

Прийняв:

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:

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

2. Особливості роботи блоку except?

3. Яка функція використовується для відкривання файлів у Python?

4. Особливості використання функції open?

5. В яких режимах можна відкрити файл?

6. Як здійснити читання і запис файлу?

7. Особливості функцій у мові Python?

8. Для чого призначенйи оператор with?

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

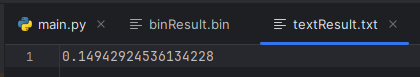
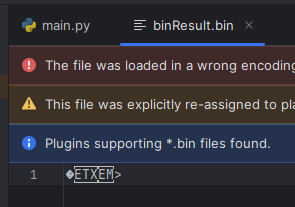
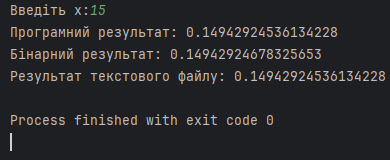
10. Як поєднуються обробка виключних ситуацій і оператор with?

**Варіант 12: y=sin(x)/tg(4x).**

**Виконання:**

import os  
import struct  
import sys  
import math  
  
# Оголошення функції запису результату в txt файл  
def wResultTxt(fName, result):  
 with open(fName, 'w') as f:  
 f.write(str(result))  
  
# Оголошення функції читання результату з txt файлу  
def rResultTxt(fName):  
 result = 0.0  
 try:  
 if os.path.exists(fName):  
 with open(fName, 'r') as f:  
 result = f.read()  
 else:  
 raise FileNotFoundError(f"File {fName} not found.")  
 except FileNotFoundError as e:  
 print(e)  
 return result  
  
# Оголошення функції запису результату в bin файл  
def wResultBin(fName, result):  
 with open(fName, 'wb') as f:  
 f.write(struct.pack('f', result))  
  
# Оголошення функції читання результату з bin файлу  
def rResultBin(fName):  
 result = 0.0  
 try:  
 if os.path.exists(fName):  
 with open(fName, 'rb') as f:  
 result = struct.unpack('f', f.read())[0]  
 else:  
 raise FileNotFoundError(f"File {fName} not found.")  
 except FileNotFoundError as e:  
 print(e)  
 return result  
  
# Функція обчислення заданого виразу згідно варіанту 26: y=1/cos(4x)  
def calculate(x):  
 try:  
 res = math.sin(math.radians(x)) /math.tan(math.radians(4 \* x))  
 except ValueError as e:  
 print(e)  
 except ZeroDivisionError as e:  
 print(e)  
  
 return res  
# Основна програма  
if \_\_name\_\_ == "\_\_main\_\_":  
 x = float(input("Введіть x:"))  
 result = calculate(x)  
 print(f"Програмний результат: {result}")  
 try:  
 wResultTxt("textResult.txt", result)  
 wResultBin("binResult.bin", result)  
 print("Бінарний результат: {0}".format(rResultBin("binResult.bin")))  
 print("Результат текстового файлу: {0}".format(rResultTxt("textResult.txt")))  
 except FileNotFoundError as e:  
 print (e)  
 sys.exit(1)

**Результат:**

****

**Відповіді на контрольні запитання:**

1. \*\*Обробка виключень в Python:\*\*

- Використовується конструкція `try`, `except` для обробки виключних ситуацій.

```python

try:

# код, який може викликати виключення

except ExceptionType as e:

# код обробки виключення

```

2. \*\*Особливості роботи блоку `except` в Python:\*\*

- Може бути кілька блоків `except` для різних типів виключень.

- Блок `except` виконується тільки в разі виникнення відповідного типу виключення.

3. \*\*Функція для відкривання файлів у Python:\*\*

- Використовується функція `open()`.

4. \*\*Особливості використання функції `open` в Python:\*\*

- Приймає два аргументи: ім'я файлу і режим відкриття.

- Може викликати виключення, тому важливо розміщувати його в блок `try`.

5. \*\*Режими відкриття файлу в Python:\*\*

- `'r'`: читання

- `'w'`: запис (створення нового файлу або перезапис існуючого)

- `'a'`: додавання (дописування в кінець файлу)

- `'b'`: бінарний режим

- `'x'`: ексклюзивне створення (файл не повинен існувати)

6. \*\*Читання та запис файлу в Python:\*\*

- Читання: `file.read()`, `file.readline()`, `file.readlines()`

- Запис: `file.write(data)`

7. \*\*Особливості функцій у Python:\*\*

- Функції визначаються за допомогою ключового слова `def`.

- Аргументи передаються по значенню (але об'єкти передаються по посиланню).

```python

def my\_function(argument1, argument2):

# тіло функції

```

8. \*\*Призначення оператору `with` в Python:\*\*

- Використовується для автоматичного встановлення та звільнення ресурсів (наприклад, файлових дескрипторів) в блоку коду.

- Не потрібно вручну викликати методи `close()` для звільнення ресурсів.

9. \*\*Вимоги до об'єктів, що передаються під контроль оператору `with`:\*\*

- Об'єкт повинен мати методи `\_\_enter\_\_` та `\_\_exit\_\_`.

10. \*\*Поєднання обробки виключень і оператору `with`:\*\*

- Оператор `with` часто використовується для автоматичного закриття файлів навіть у випадку виникнення виключення в середині блоку коду.

**Висновок:** на даній лабораторній роботі я оволоділа навиками використання засобів мови Python для роботи з файлами.