

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра «Електронних обчислювальних машин»



Звіт
з лабораторної роботи № 8
з дисципліни: «Кросплатформенні засоби програмування»
на тему: «ФАЙЛИ ТА ВИКЛЮЧЕННЯ У PYTHON»

Виконав:

студент групи КІ-306

Хмільовський С. Р.

Прийняв:

доцент кафедри ЕОМ

Іванов Ю. С.

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

Завдання (варіант № 22)

1. Написати та налагодити програму на мові Python згідно варіанту ($y = \tan(4x)/x$). 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.

Вихідний код програми:

Файл *main.py*

```
import os
import struct
import sys
import math

#функція запису у ТЕКСТОВИЙ файл
def write_res_txt(f_name, result):
    with open(f_name, 'w') as f:
        f.write(str(result))
        print("Результат був записаний у текстовий файл!")

#функція читання з ТЕКСТОВОГО файлу
def read_res_txt(f_name):
    result = 0.0
    try:
        if os.path.exists(f_name):
            with open(f_name, 'r') as f:
                result = float(f.read()) #Перетворення зчитаного тексту у число
        else:
            raise FileNotFoundError(f"File {f_name} not found.")
    except FileNotFoundError as e:
        print(e)
    return result

#функція запису у БІНАРНИЙ файл
def write_res_bin(f_name, result):
```

```

with open(f_name, 'wb') as f:
    f.write(struct.pack('f', result))
    print("Результат був записаний у бінарний файл!")

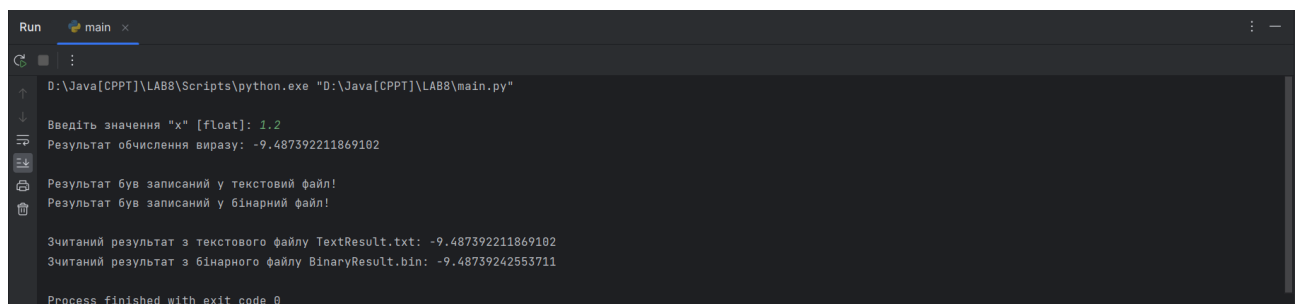
#функція читання з БІНАРНОГО файлу
def read_res_bin(f_name):
    result = 0.0
    try:
        if os.path.exists(f_name):
            with open(f_name, 'rb') as f:
                result = struct.unpack('f', f.read())[0] #Перетворення зчитаних
данних у float
        else:
            raise FileNotFoundError(f"File {f_name} not found.")
    except FileNotFoundError as e:
        print(e)
    return result

#функція, яка виконує усі необхідні обчислення
def calculate_expression(x):
    try:
        result = math.tan(4 * x) / x
    except ZeroDivisionError:
        print("Ділення на нуль неможливе!")
        return None
    return result

#main функція програми:
if __name__ == "__main__":
    try:
        data = float(input("\nВведіть значення \"x\" [float]: "))
        result = calculate_expression(data)
        if result is not None:
            print(f"Результат обчислення виразу: {result}")
            print()
            write_res_txt("TextResult.txt", result)
            write_res_bin("BinaryResult.bin", result)
            print()
            print("Зчитаний результат з текстового файлу TextResult.txt:
{0}".format(read_res_txt("TextResult.txt")))
            print("Зчитаний результат з бінарного файлу BinaryResult.bin:
{0}".format(read_res_bin("BinaryResult.bin")))
        except ValueError as e:
            print(f"Error: {e}")
        except FileNotFoundError as e:
            print(e)
    except:
        sys.exit(1)

```

Результат виконання програми:



```

Run main x
D:\Java[CPPT]\LAB8\Scripts\python.exe "D:\Java[CPPT]\LAB8\main.py"
Введіть значення "x" [float]: 1.2
Результат обчислення виразу: -9.487392211869102
Результат був записаний у текстовий файл!
Результат був записаний у бінарний файл!
Зчитаний результат з текстового файлу TextResult.txt: -9.487392211869102
Зчитаний результат з бінарного файлу BinaryResult.bin: -9.487392211869102
Process finished with exit code 0

```

Вміст файлу TextResult.txt:

Відповіді на контрольні запитання:

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

Обробка виключень в мові Python виконується за допомогою конструкції try-
except.

2. Особливості роботи блоку except?

Блок except використовується для обробки виключень, тобто код, який виконується, якщо виникає виключна ситуація в блоку try.

3. Яка функція використовується для відкривання файлів у Python?

Для відкриття файлів у Python використовується функція open().

4. Особливості використання функції open?

Функція open() має різні параметри, такі як шлях до файлу, режим відкриття (наприклад, 'r' для читання, 'w' для запису), інші параметри, такі як encoding.

5. В яких режимах можна відкрити файл?

Файл можна відкрити в різних режимах, таких як читання ('r'), запис ('w'), додавання ('a'), тощо.

6. Як здійснити читання і запис файлу?

Для читання використовується метод read(), для запису - метод write().

7. Особливості функцій у мові Python?

Особливості функцій включають в себе параметри, локальні та глобальні змінні, анотації типів, значення за замовчуванням та інші.

8. Для чого призначений оператор `with`?

Оператор `with` використовується для забезпечення коректного відкриття та закриття ресурсів, таких як файли.

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

Об'єкти, що передаються під контроль оператору `with` (наприклад, файли), повинні мати методи `enter` та `exit` для правильного відкриття та закриття.

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

Обробка виключень може бути впроваджена у метод `exit` об'єкта, який передається під контроль оператору `with`, для відповідного управління виключеннями при відкритті та закритті ресурсів.

Висновок:

На даній лабораторній роботі я оволодів навичками використання засобів мови Python для роботи з файлами. Також розробив програму яка обчислює вираз, заданий варіантом та записує результат у файли, бінарний та текстовий.