

Міністерство освіти і науки України
Національний університет «Львівська Політехніка»

Кафедра ЕОМ



ЗВІТ

з лабораторної роботи №8

з дисципліни: “Кросплатформенні засоби програмування”

на тему: «**Файли та виключення Python**»

Варіант - 19

Виконав:

ст. гр. КІ-305

Лихограй А. В.

Прийняв:

доцент кафедри ЕОМ

Іванов Ю. С.

Львів 2023

Мета роботи: 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. Дати відповіді на контрольні запитання:
 - за допомогою якої конструкції у мові Python обробляються виключні ситуації?
 - особливості роботи блоку except?
 - яка функція використовується для відкривання файлів у Python?
 - особливості використання функції open?
 - в яких режимах можна відкрити файл?
 - як здійснити читання і запис файлу?
 - особливості функцій у мові Python?
 - для чого призначений оператор with?
 - які вимоги ставляться до об'єктів, що передаються під контроль оператору with?
 - як поєднуються обробка виключних ситуацій і оператор with?

Вхідний вираз: $y = \frac{\operatorname{ctg}(x)}{\sin(2x) + 4\cos(x)}$.

Вихідний код програми:

```
main.py:
import os
import struct
import sys
import math

# Функція для запису результату у текстовий файл
def writeResTxt(file_name, result):
    with open(file_name, 'w') as file:
        file.write(str(result))
```

```

# Функція для читання результату з текстового файлу
def readResTxt(file_name):
    result = 0.0
    try:
        if os.path.exists(file_name):
            with open(file_name, 'r') as file:
                result = float(file.read())
        else:
            raise FileNotFoundError(f"Файл {file_name} не знайдено.")
    except FileNotFoundError as e:
        print(e)
    return result

# Функція для запису результату у бінарний файл
def writeResBin(file_name, result):
    with open(file_name, 'wb') as file:
        packed_result = struct.pack('f', result)
        file.write(packed_result)

# Функція для читання результату з бінарного файлу
def readResBin(file_name):
    result = 0.0
    try:
        if os.path.exists(file_name):
            with open(file_name, 'rb') as file:
                packed_result = file.read(4) # читаємо 4 байти (розмір
типу float)
                result = struct.unpack('f', packed_result)[0]
        else:
            raise FileNotFoundError(f"Файл {file_name} не знайдено.")
    except FileNotFoundError as e:
        print(e)
    return result

# Функція для обчислення виразу (1 / Math.tan(rad)) / ( Math.sin(2 * rad)
+ 4 * Math.cos(rad))
def calculate_expression(x):
    try:
        rad = math.radians(x)
        er_value = math.sin(2 * rad) + 4 * math.cos(rad)
        if er_value == 0:
            raise ZeroDivisionError("Знаменник не може дорівнювати нулю!")
        elif x == 90 and math.tan(rad):
            raise ZeroDivisionError("Не прийнятне значення для обчислення
ctg()!")
        result = (1.0 / math.tan(rad) / er_value)
        return result
    except ZeroDivisionError as e:
        raise CalcException(str(e))

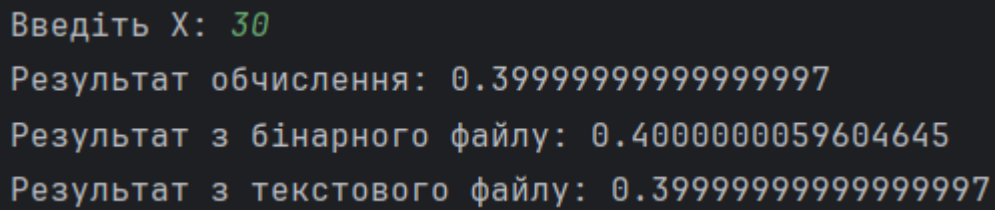
# клас для власного виключення CalcException
class CalcException(Exception):
    pass

if __name__ == "__main__":
    try:
        data = float(input("Введіть X: "))
        result = calculate_expression(data)
        print(f"Результат обчислення: {result}")
        writeResTxt("textRes.txt", result)
        writeResBin("binRes.bin", result)
        print("Результат з бінарного файлу:
{0}".format(readResBin("binRes.bin")))
        print("Результат з текстового файлу:
{0}".format(readResTxt("textRes.txt")))
    except CalcException as e:

```

```
print(e)
except FileNotFoundError as e:
    print(e)
    sys.exit(1)
```

Результат виконання програми:



Введіть X: 30
Результат обчислення: 0.39999999999999997
Результат з бінарного файлу: 0.4000000059604645
Результат з текстового файлу: 0.39999999999999997

Рис. 1. Результат роботи програми.

Відповіді на контрольні запитання:

1. Конструкція для обробки виключень у мові Python:
 - Використовується конструкція ``try...except``.
2. Особливості роботи блоку ``except``:
 - Блок ``except`` містить код, який виконується у випадку виникнення виключення. Може бути кілька блоків ``except`` для обробки різних типів виключень.
3. Функція для відкривання файлів у Python:
 - Функція ``open()`` використовується для відкриття файлів.
4. Особливості використання функції ``open``:
 - ``open()`` може приймати два аргументи: ім'я файлу та режим відкриття.
5. Режими відкриття файлу:
 - ``r``: Читання (за замовчуванням).
 - ``w``: Запис (створює новий файл або очищує вміст існуючого).
 - ``a``: Дописування (додає дані в кінець файлу).
 - ``b``: Відкриття у бінарному режимі.
6. Читання і запис файлу:

- Для читання використовується метод `read()` або ітерація по файловому об'єкту. Для запису - метод `write()`.

7. Особливості функцій у Python:

- Функції в Python можуть повертати значення за допомогою `return`.
- Аргументи функцій можуть бути позначені як обов'язкові чи необов'язкові.

8. Призначення оператора `with`:

- Оператор `with` використовується для забезпечення правильного управління ресурсами, такими як файли або мережеві з'єднання.

9. Вимоги до об'єктів для оператора `with`:

- Об'єкти, які передаються під контроль оператора `with`, повинні мати методи `__enter__` та `__exit__`.

10. Поєднання обробки виключень і оператора `with`:

- Оператор `with` може використовуватися для автоматичного виклику методу `__exit__`, навіть якщо виникає виключення у блоках `with`. Це дозволяє коректно звільнювати ресурси.

Висновок: під час виконання лабораторної роботи я ознайомився з принципами роботи з виключеннями та файлами в мові Python. Розробив програму, яка реалізовує вхідне завдання.