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 - 7

Виконала:

ст. гр. КІ-305

Дзера А. Р.

Прийняв:

доц. каф. 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. 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.

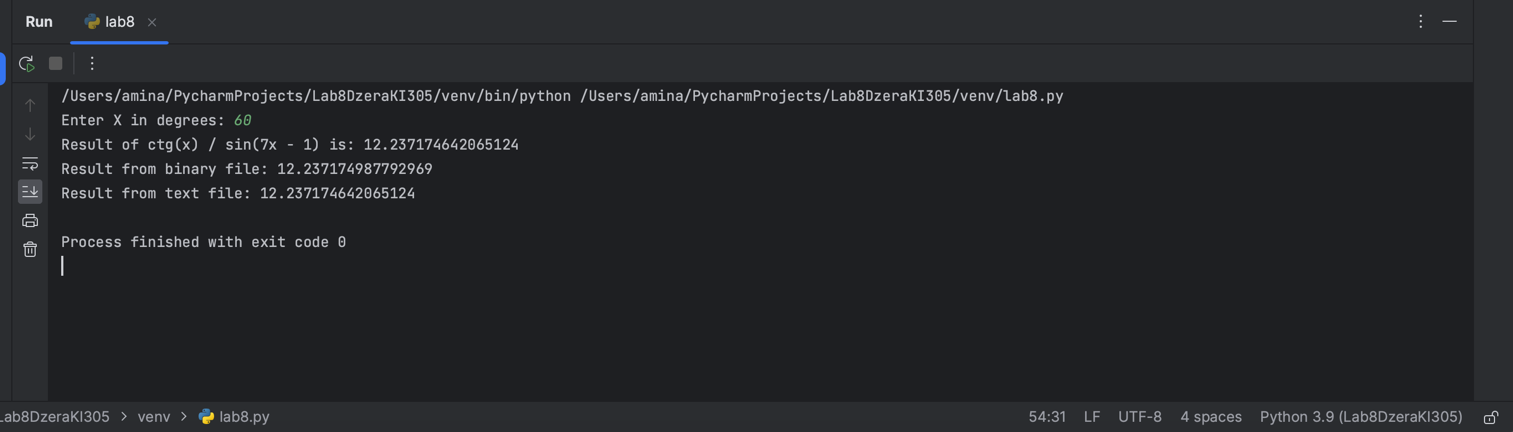
4. Дати відповідь на контрольні запитання.

Варіант завдання –

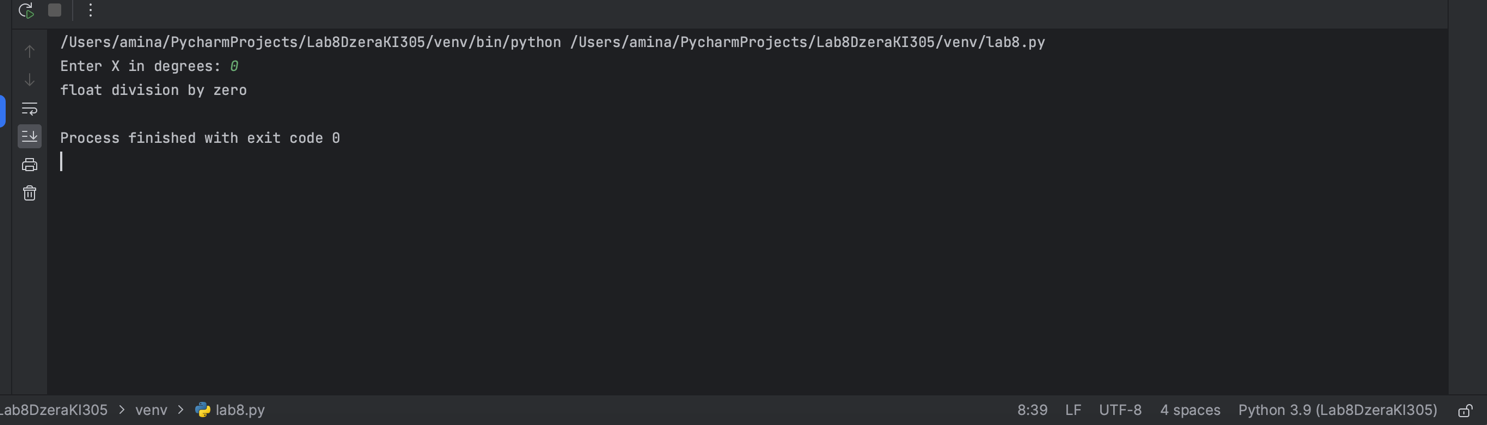
**Код програми**

import os  
import struct  
import sys  
import math  
  
# функція для запису результату у текстовий файл  
def write\_res\_txt(file\_name, result):  
 with open(file\_name, 'w') as file:  
 file.write(str(result))  
  
# функція для читання результату з текстового файлу  
def read\_res\_txt(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"File {file\_name} not found.")  
 except FileNotFoundError as e:  
 print(e)  
 return result  
  
# функція для запису результату у бінарний файл  
def write\_res\_bin(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 read\_res\_bin(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"File {file\_name} not found.")  
 except FileNotFoundError as e:  
 print(e)  
 return result  
  
# функція для обчислення виразу ctg(x) / sin(7x - 1)  
def calculate\_expression(x):  
 try:  
 rad = math.radians(x)  
 sin\_value = math.sin(7 \* rad - 1)  
 if sin\_value == 0:  
 raise ZeroDivisionError("Exception reason: sin(7x-1) = 0")  
 result = (1.0 / math.tan(rad) / sin\_value)  
 return result  
 except ZeroDivisionError as e:  
 raise CalcException(str(e))  
  
# клас для власного виключення CalcException  
class CalcException(Exception):  
 pass  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 try:  
 # зчитуємо введене користувачем значення  
 data = float(input("Enter X in degrees: "))  
 # обчислюємо вираз  
 result = calculate\_expression(data)  
 # виводимо результат  
 print(f"Result of ctg(x) / sin(7x - 1) is: {result}")  
  
 # записуємо результат у текстовий та бінарний файли  
 write\_res\_txt("textRes.txt", result)  
 write\_res\_bin("binRes.bin", result)  
  
 # читаємо результат з бінарного файлу та текстового файлу і виводимо їх  
 print("Result from binary file: {0}".format(read\_res\_bin("binRes.bin")))  
 print("Result from text file: {0}".format(read\_res\_txt("textRes.txt")))  
  
 except CalcException as e:  
 print(e)  
 except FileNotFoundError as e:  
 print(e)  
 sys.exit(1)

**Результати роботи програми**

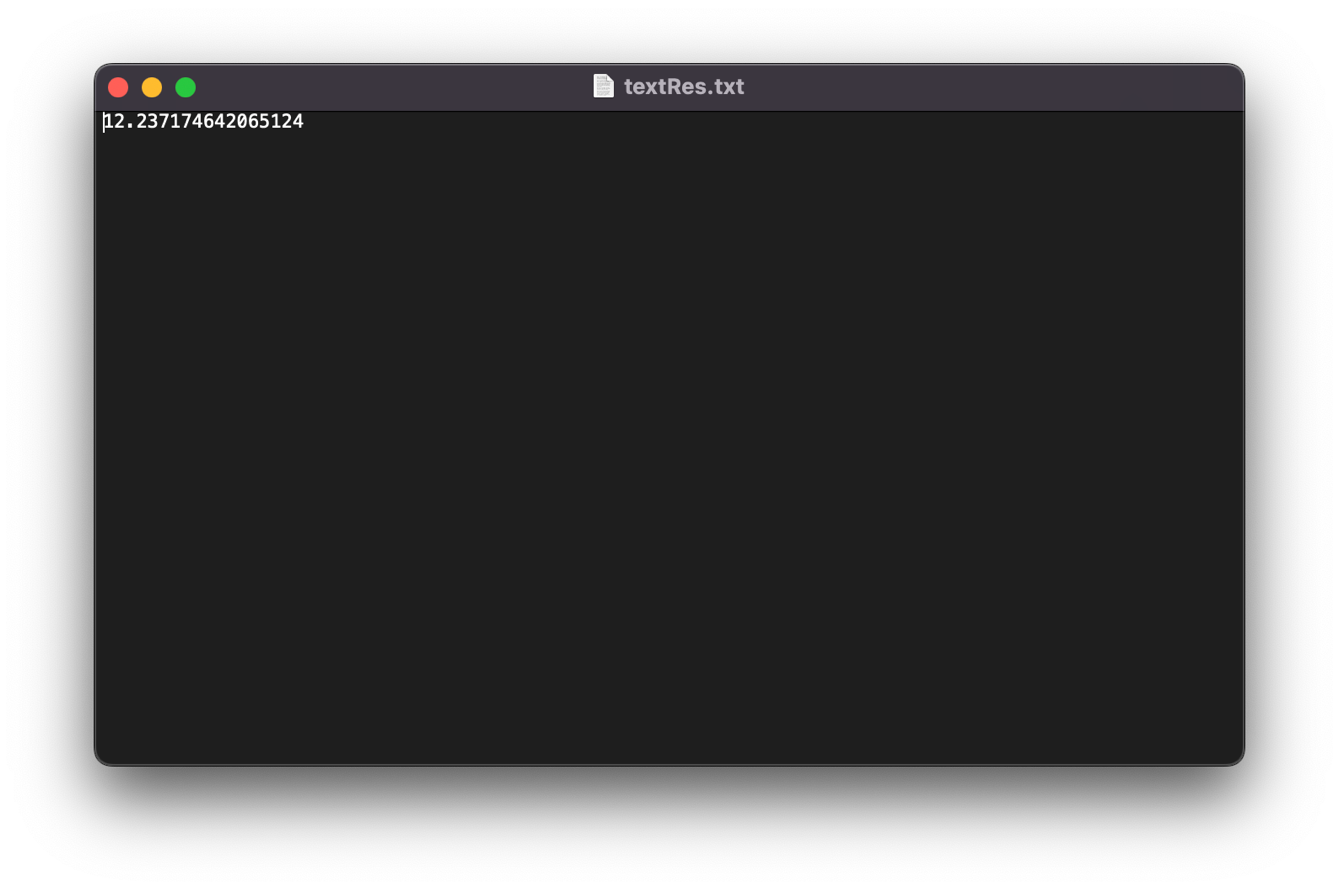
****

*Рис.1. Результат виконання програми*



*Рис.2. Виключення при значенні нуль*

**Вміст файлy textRes.txt**

****

**Контрольні питання**

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.

1. Особливості роботи блоку except?

* Блок except розміщується після блоку try і може містити один або декілька обробників винятків
* Кожен except може мати специфікований тип винятку, який він обробляє.
* Python дозволяє також використовувати загальні обробники винятків без вказівки конкретного типу.
* Обробники визначаються в порядку зверху вниз.
* В кожному except блоку ви можете вказати ім'я змінної, яка буде посилатися на об'єкт винятку.
* Багаторазове використання except

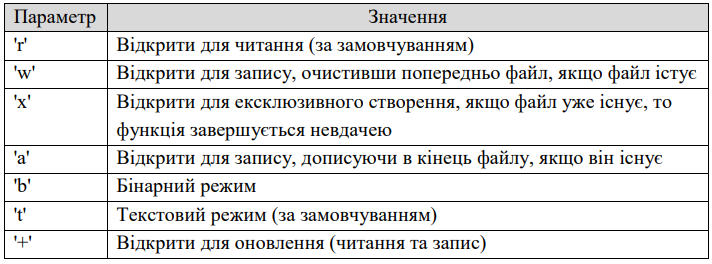
1. Яка функція використовується для відкривання файлів у Python?

Ключовою функцією для роботи з файлами є функція open(file, mode='r', buffering=-1, encoding=None, errors=None, newline=None, closefd=True, opener=None).

1. Особливості використання функції open?

* Режими відкривання файлу
* Закриття файлу
* Робота з файловими об’єктами
* Обробка помилок
* Кодування
* Пошук файлу

1. В яких режимах можна відкрити файл?

****

1. 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.

1. Особливості функцій у мові Python?

* Синтаксис визначення функцій
* Аргументи функції
* Повернення значення
* Область видимості
* Рекурсія
* Анонімні функції
* Замикання
* Документація функцій
* Декоратори
* Іменовані аргументи

1. Для чого призначенйи оператор with?

Оператор with використовується для автоматизації процесів закриття ресурсу і коректної обробки виключних ситуацій (аналог оператора try-з-ресурсами у Java). Наприклад, автоматичне закриття файлу, чи з’єднання після завершення роботи з ним, а також, при виникненні виключень.

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

* Присутність методів ‘\_\_enter\_\_’ та ‘\_\_exit\_\_’
* Можливість здійснювати налаштування перед входом
* Можливість здійснювати очищення після виходу
* Обробка винятків

1. Як поєднуються обробка виключних ситуацій і оператор with?

* Обробка винятку всередині блоку with
* Використання блоку finally
* Каскадне відкриття і обробка винятків

**Висновок**

Під час виконання даної лабораторної роботи я оволоділа навикачми використання засобів мови Python для роботи з файлами.