Міністерство освіти і науки України

Національний університет «Львівська політехніка»



**ЗВІТ**

з лабораторної роботи №8

з дисципліни: “Кросплатформні засоби програмування”

на тему: “ Файли та виключення у Python”

Виконав:

ст. гр. КІ-305

Шевчук М.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.

**Варіант №28**



**Лістинг програми:**

import os

import struct

import sys

import math

# функція запису результату в текстовий файл

def writeResTxt(fName, result):

with open(fName, 'w') as f:

f.write(str(result))

# функція читання результату з текстового файлу

def readResTxt(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

# функція запису результату в бінарний файл

def writeResBin(fName, result):

with open(fName, 'wb') as f:

f.write(struct.pack('f', result))

# функція читання результату з бінарного файлу

def readResBin(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 (з обробкою можливих помилок) 1/ctg(2x)

def calculate(x):

res = 0

try:

rad = math.radians(2 \* x)

res = math.tan(rad)

except ValueError as e:

print(e)

except ZeroDivisionError as e:

print(e)

return res

# основна програма

if \_\_name\_\_ == "\_\_main\_\_":

data = float(input("Enter data: "))

result = calculate(data)

print(f"Result is: {result}")

try:

writeResTxt("textRes.txt", result)

writeResBin("binRes.bin", result)

print("Result is: {0}".format(readResBin("binRes.bin")))

print("Result is: {0}".format(readResTxt("textRes.txt")))

except FileNotFoundError as e:

print (e)

sys.exit(1)

**Результат виконання програми:**

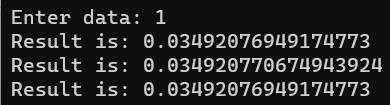
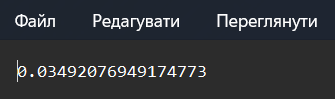


Рис.1. Результат виконання програми в консолі



*Рис.2.Результат виконання програми у файлі*

**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

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

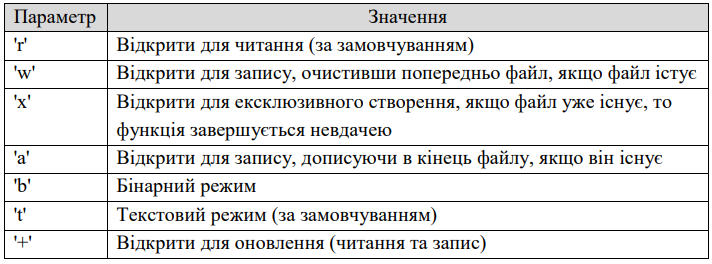
1. **Яка функція використовується для відкривання файлів у Python?**

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 призначений 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:** я оволодів навиками використання засобів мови Python для роботи з файлами.