Національний університет «Львівська Політехніка»

Інститут комп’ютерних технологій, автоматики та метрології

Кафедра електронних обчислювальних машин



Звіт

Про виконання лабораторної роботи №8

З дисципліни «Кросплатформлені засоби програмування»

**Виконав:**

студент групи КІ-305

Ключко Д.С.

**Перевірив:**

Доцент кафедри ЕОМ

Іванов Ю. С.

Львів – 2023

**Тема:** Файли та виключення у Python.

**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.

**Завдання згідно варіанту:**

**Варіант: 12**

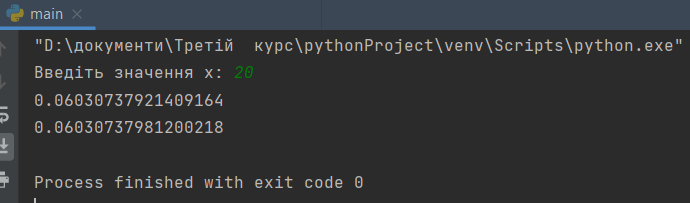


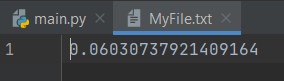
**Виконання:**

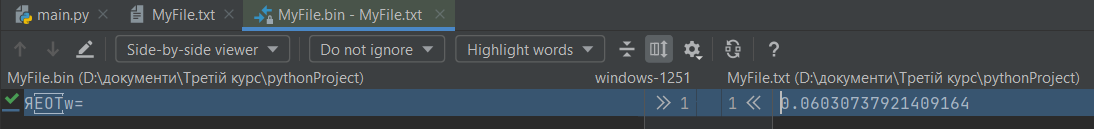
**Код програми:**

mport struct  
import sys  
import math  
import os  
# функція запису результату в бінарний файл  
def write\_bin(result):  
 with open('MyFile.bin', 'wb') as f:  
 f.write(struct.pack('f', result))  
 f.close()  
# функція читання результату з бінарного файлу  
def read\_bin():  
 result = 0.0  
 try:  
 if os.path.exists('MyFile.bin'):  
 with open('MyFile.bin', 'rb') as f:  
 result = struct.unpack('f', f.read())[0]  
 f.close()  
 else:  
 raise FileNotFoundError(f"File {'MyFile.bin'} not found.")  
 except FileNotFoundError as e:  
 print(e)  
 return result  
# функція запису результату в текстовий файл  
def write\_txt(result):  
 with open('MyFile.txt', 'w') as f:  
 f.write(str(result))  
 f.close()  
# функція читання результату з текстового файлу  
def read\_txt():  
 result = 0.0  
 try:  
 if os.path.exists('MyFile.txt'):  
 with open('MyFile.txt', 'r') as f:  
 result = f.read()  
 f.close()  
 else:  
 raise FileNotFoundError(f"File {'MyFile.txt'} not found.")  
 except FileNotFoundError as e:  
 print(e)  
 return result  
# функція обчислення заданого виразу  
def my\_calc (x):  
 rad = x \* math.pi / 180  
 y = math.sin(rad) / math.tan(4\*rad)  
 return y  
# основна програма  
if \_\_name\_\_ == "\_\_main\_\_":  
 x = int(input("Введіть значення x:"))  
 result = my\_calc(x)  
 try:  
 writetxt = write\_txt(result)  
 readtxt =read\_txt()  
 print(readtxt)  
 writebin = write\_bin(result)  
 readbin =read\_bin()  
 print(readbin)  
 except FileNotFoundError as e:  
 print(e)  
 sys.exit(1)

**Результат роботи програми у консолі:**







**КОНТРОЛЬНІ ПИТАННЯ**

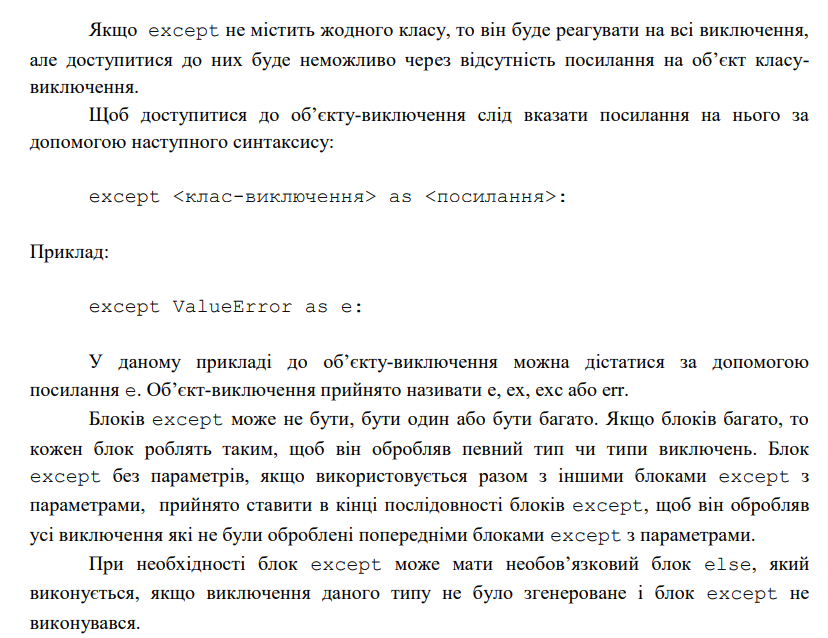
1. За допомогою якої конструкції у мові Python обробляються виключні ситуації?

**Відповідь:**

Мова Python має вбудований механізм обробки виключних ситуацій. Обробка виключних ситуацій забезпечується блоками try-except-finally

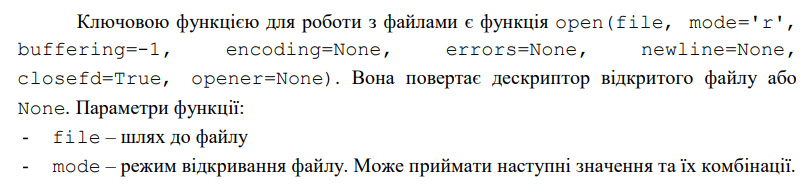
1. Особливості роботи блоку except?

**Відповідь:**



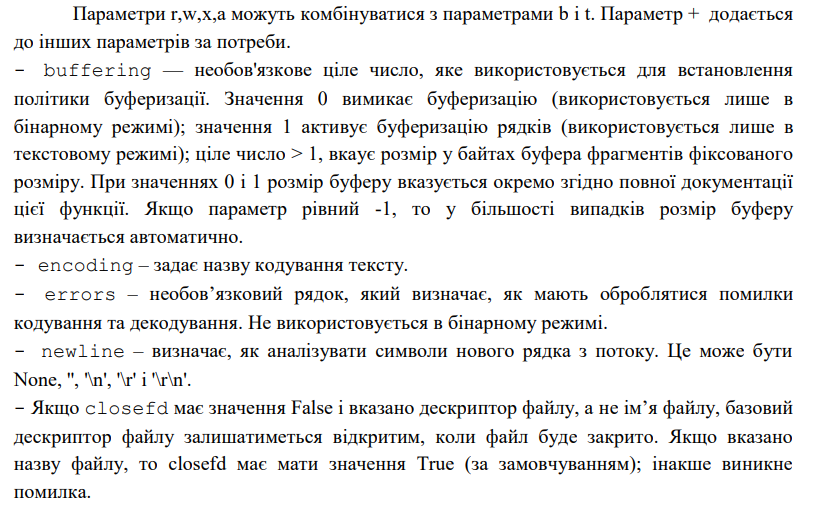
1. Яка функція використовується для відкривання файлів у Python?

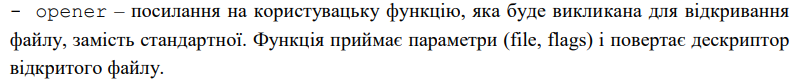
**Відповідь:**

****

1. Особливості використання функції open?

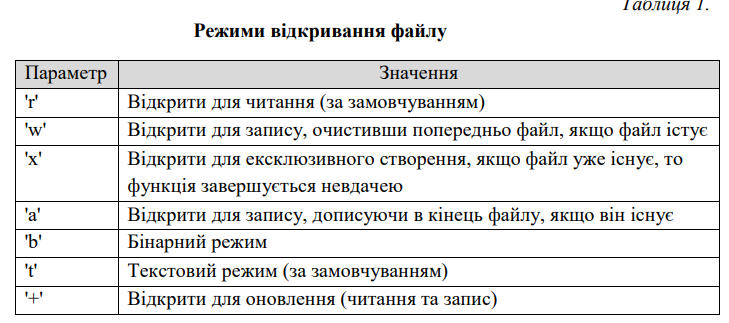
**Відповідь:**

****

****

1. В яких режимах можна відкрити файл ?

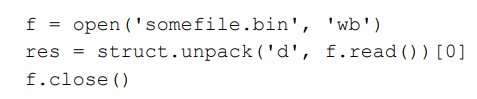
**Відповідь:**



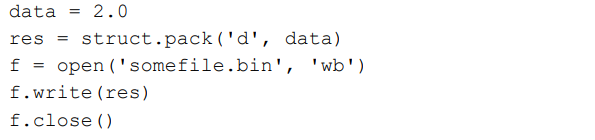
1. Як здійснити читання і запис файлу

**Відповідь:**

**Читання:**

****

**Запис:**

****

1. Особливості функцій у мові Python?

**Відповідь:**

**Синтаксис визначення**;

**Аргументи та параметри**;

**Документування функцій**;

**Вкладені функції**;

**Значення за замовчуванням**;

**Використання return**;

**Лямбда-функції**;

**Значення None**;

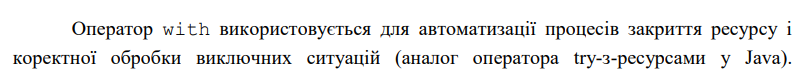
**Передача аргументів у функцію по посиланню**;

**Глобальні та локальні змінні**;

**Декоратори**;

1. Для чого призначенйи оператор with?

**Відповідь:**

****

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

**Відповідь:**

Наприклад, автоматичне закриття файлу, чи з’єднання після завершення роботи з ним, а також, при виникненні виключень. Таким ресурсом може бути будь-який об’єкт, клас якого містить визначені методи \_\_enter\_\_ та \_\_exit\_\_, які дозволяють належним чином керувати ресурсами під час входу в блок with, виходу з нього та обробки виключних ситуацій. Такий об’єкт в термінах оператора with називається менеджером контексту. Детальний опис оператора with є у PEP-343.

1. Як поєднуються обробка виключних ситуацій і оператор with? **Відповідь:**

Обробка виключень (try, except, finally) і оператор with - це два різні механізми в мові програмування Python, які можуть використовуватися разом для ефективного управління ресурсами та обробки виключень.

Обробка виключень (try, except, finally)

try: Цей блок коду визначається для виконання коду, який може викинути виключення.

except: Використовується для перехоплення та обробки виключень, що виникають у блоку try.

finally: Оператор finally використовується для виконання певного коду навіть після обробки виключення або безпосередньо перед його викиданням.

Оператор with

with: Цей оператор використовується для автоматичного керування ресурсами, які підтримують протокол менеджера контексту. Найпоширеніше використання with - це з відкриттям файлів, щоб забезпечити їх правильне закриття після завершення виконання коду.

**Висновок:** на цій лабораторній роботі я оволодів навиками використання засобів мови Python для роботи з файлами. Написав програму, що обчислює вираз згідно варіанту, а також оброблює можливі математичні помилки, а результат записує і зчитує з бінарного і текстового файлів.