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»

Виконав:

Студент групи КІ-306

Яцків Адріян

Прийняв:

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.

1. Завантажити код на GitHub згідно методичних вказівок по роботі з GitHub.
2. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації та завантажити його у ВНС.
3. Дати відповідь на контрольні запитання.

# Варіант завдання:



# Код програми:

import math

import struct

import os

class Calc:

    def \_\_init\_\_(self):

        self.result = 0.0

    def calculate(self, x):

        if x == 90 or x == -90:

            print("You entered incorrect values")

            exit(0)

        else:

            radians = math.radians(x)

            self.result = math.tan(radians)

    def get\_result(self):

        return self.result

    def write\_res\_txt(self, file\_name):

        try:

            with open(file\_name, 'w') as file:

                file.write(f"{self.result}")

        except FileNotFoundError as ex:

            print(ex)

    def read\_res\_txt(self, file\_name):

        try:

            with open(file\_name, 'r') as file:

                self.result = float(file.read())

        except FileNotFoundError as ex:

            print(ex)

    def write\_res\_bin(self, file\_name):

        try:

            with open(file\_name, 'wb') as file:

                file.write(struct.pack('>d', self.result))

        except FileNotFoundError as ex:

            print(ex)

    def read\_res\_bin(self, file\_name):

        try:

            with open(file\_name, 'rb') as file:

                self.result = struct.unpack('>d', file.read(8))[0]

        except FileNotFoundError as ex:

            print(ex)

    def writeNum(data, file\_name):

            try:

                with open(file\_name, 'w') as file:

                    file.write(f"{data.result}")

            except FileNotFoundError as ex:

                print(ex)

def main():

    obj = Calc()

    target\_directory = "C:/Users/Adriyan/Desktop/PythonLabs/Lab8"

    os.chdir(target\_directory)

    data = int(input("Enter data: "))

    with open("num.txt", 'w') as file:

         file.write(f"{data}")

    obj.calculate(data)

    print(f"Result is: {obj.get\_result()}")

    obj.write\_res\_txt("textRes.txt")

    obj.write\_res\_bin("BinRes.bin")

    obj.read\_res\_bin("BinRes.bin")

    print(f"Result from bin: {obj.get\_result()}")

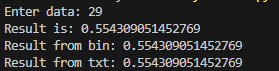
    obj.read\_res\_txt("textRes.txt")

    print(f"Result from txt: {obj.get\_result()}")

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

    main()

# Результати роботи програми:

****

**Відповіді на контрольні запитання**

* 1. За допомогою якої конструкції у мові Python обробляються виключні ситуації? - "try-except".
  2. Особливості роботи блоку except?
     + Блок "except" використовується для обробки виключних ситуацій, які можуть виникнути у блоку "try". Він містить код, який виконується у випадку виникнення виключної ситуації.
  3. Яка функція використовується для відкривання файлів у Python? - open()
  4. Особливості використання функції open?
     + Функція "open()" дозволяє відкривати файли з різними параметрами, такими як режим відкриття, кодування і т.д.
  5. В яких режимах можна відкрити файл?
     + у режимах читання ("r"), запису ("w"), додавання ("a") і бінарного режиму ("b"), а також їх комбінаціях.
  6. Як здійснити читання і запис файлу?
     + Для читання файлу використовують функцію "read()", а для запису - "write()".
  7. Особливості функцій у мові Python?
     + Функції у Python можуть мати параметри, повертати значення, і багато інших особливостей, що дозволяють створювати різноманітні функції для розв'язання завдань.
  8. Для чого призначенйи оператор with?
     + Оператор "with" використовується для автоматичного відкриття і закриття файлів (або інших ресурсів) та забезпечення правильного їхнього використання.
  9. Які вимоги ставляться до об’єктів, що передаються під контроль оператору with?
     + Об'єкти, що передаються під контроль оператору "with", повинні мати методи "enter" і "exit", які виконуються перед входом і виходом з контексту.
  10. Як поєднуються обробка виключних ситуацій і оператор with?
      + Обробка виключних ситуацій і оператор "with" можуть комбінуватися, дозволяючи обробляти виключення у контексті "with" і гарантуючи закриття ресурсів навіть у випадку виникнення виключної ситуації.

# Висновок

У ході у ході виконання даної лабораторної роботи, я успішно оволоділа необхідними навичками. Основні засоби мови Python для роботи з файлами були детально вивчені, включаючи відкриття, читання, запис та закриття файлів.