Міністерство освіти і науки України Національний університет «Львівська політехніка» Кафедра «Електронних обчислювальних машин»



Звіт

з лабораторної роботи № 8

з дисципліни: «Кросплатформенні засоби програмування» на тему: «Дослідження базових конструкцій мови Python»

**Виконав:** студент групи КІ-304 Мох М. П.

**Прийняв:**

Олексів М.В.

Львів – 2023

**Мета роботи:** оволодіти навиками використання засобів мови Python для роботи з файлами.

# Варіант 12

y = sin(x) / tg(4x)

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.

# Код програми:

import math  
import os.path  
import struct  
import sys  
  
# Writes the result to a text file  
def writeResTxt(fName, res):  
 with open(fName, 'w') as f:  
 f.write(str(res))  
  
# Reads the result from a text file  
def readResTxt(fName):  
 res = 0.0  
 try:  
 if os.path.exists(fName):  
 with open(fName, 'r') as f:  
 res = f.read()  
 else:  
 raise FileNotFoundError(f"\nError: File {fName} not found")  
 except FileNotFoundError as e:  
 print(e)  
 return res  
  
# Writes the result to a binary file  
def writeResBin(fName, result):  
 with open(fName, 'wb') as f:  
 f.write(struct.pack('f', result))  
  
# Reads the result from a binary file  
def readResBin(fName):  
 res = 0.0  
 try:  
 if os.path.exists(fName):  
 with open(fName, 'rb') as f:  
 res = struct.unpack('f', f.read())[0]  
 else:  
 raise FileNotFoundError(f"\nError: File {fName} not found")  
 except FileNotFoundError as e:  
 print(e)  
 return res  
  
# Get correct float from user  
def getInput():  
 try:  
 x = float(input("Enter x: "))  
 except ValueError as e:  
 print(f"\nError: Wrong input... \n{e}")  
 sys.exit("Exiting due to error.")  
 return x;  
  
  
# Calculates the result using the formula: y = sin(x) / tg(4x)  
def calculate(x):  
 try:  
 res = math.sin(x) / math.tan(4 \* x)  
 except ZeroDivisionError as e:  
 print("\nError: You can't divide by 0")  
 res = 0;  
 return res  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 # Get x and res  
 x = getInput()  
 res = calculate(x)  
  
 # Writing the result to a text file and reading it again  
 writeResTxt("result.txt", res)  
 resFromTxtFile = readResTxt("result.txt")  
 print("Result from txt file: ", resFromTxtFile)  
  
 # Writing the result to a binary file and reading it again  
 writeResBin("result.bin", res)  
 resFromBinFile = readResBin("result.bin")  
 print("Result from bin file: ", resFromBinFile)

# Результат виконання 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.

* 1. Яка функція використовується для відкривання файлів у Python? - open()
  2. Особливості використання функції open?

Функція "open()" дозволяє відкривати файли з різними параметрами, такими як режим відкриття, кодування і т.д.

* 1. В яких режимах можна відкрити файл?

У режимах читання ("r"), запису ("w"), додавання ("a") і бінарного режиму ("b"), а також їх комбінаціях.

* 1. Як здійснити читання і запис файлу?

Для читання файлу використовують функцію "read()", а для запису - "write()".

* 1. 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.

Оператор "with" використовується для автоматичного відкриття і закриття файлів (або інших ресурсів) та забезпечення правильного їхнього використання.

* 1. 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.

* 1. 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, успішно оволодів необхідними навичками. Основні засоби мови Python для роботи з файлами були детально вивчені, включаючи відкриття, читання, запис та закриття файлів.