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: «Файли та виключення у Python»

**Виконав:**

студент групи КІ-307

Трофименко Є.Е.

**Прийняв:**

доцент кафедри ЕОМ

Іванов Ю. С.

Львів – 2023

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

**Завдання (варіант № 22)**

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.



**Вихідний код програми**

import math

import struct

class CalculateTheEquation:

def \_\_init\_\_(self):

self.variable = 0

# Функція обчислення виразу tg(4x)/4

def doCalculation(self):

try:

self.variable = math.tan(4\*self.variable) / self.variable

return self.variable

except ZeroDivisionError:

print("Arithmetic exception: illegal value.")

return 0

# Функція запису результату в текстовий файл

def writeResultToTxt(self, fileName):

try:

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

f.write(str(self.variable))

except IOError:

print("Error writing to file.")

# Функція запису результату в бінарний файл

def writeResultToBin(self, fileName):

try:

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

f.write(struct.pack('>d', self.variable))

except IOError:

print("Error writing to file.")

# Функція читання результату з текстового файлу

def readResultFromTxt(self, fileName):

try:

with open(fileName, 'r') as f:

self.variable = float(f.read())

except IOError:

print("Error reading from file.")

# Функція читання результату з бінарного файлу

def readResultFromBin(self, fileName):

try:

with open(fileName, 'rb') as f:

self.variable = int.from\_bytes(f.read(), 'big') / (2\*\*64)

except IOError:

print("Error reading from file.")

calc = CalculateTheEquation()

# Виведення результату з текстового файлу

try:

calc.readResultFromBin("bin.bin")

print("Result of calculation:", calc.doCalculation())

calc.writeResultToBin("bin.bin")

except IOError as e:

print(e)

# Виведення результату з бінарного файлу

try:

calc.readResultFromTxt("txt.txt")

print("Result of calculation:", calc.doCalculation())

calc.writeResultToTxt("txt.txt")

except IOError as e:

print(e)

**Результат виконання програми**

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

**Відповіді на контрольні запитання**

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

*Обробка виключних ситуацій забезпечується блоками try-except-finally.*

2. Особливості роботи блоку except?

*Блоків except може не бути, бути один або бути багато. Якщо блоків багато, то кожен блок роблять таким, щоб він обробляв певний тип чи типи виключень. Блок except без параметрів, якщо використовується разом з іншими блоками except з параметрами, прийнято ставити в кінці послідовності блоків except, щоб він обробляв усі виключення які не були оброблені попередніми блоками except з параметрами. При необхідності блок except може мати необов’язковий блок else, який виконується, якщо виключення даного типу не було згенероване і блок except не виконувався.*

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). Вона повертає дескриптор відкритого файлу або None.*

1. Особливості використання функції open?

*Параметри r,w,x,a можуть комбінуватися з параметрами b і t. Параметр + додається до інших параметрів за потреби.*

*- buffering — необов'язкове ціле число, яке використовується для встановлення політики буферизації. Значення 0 вимикає буферизацію (використовується лише в бінарному режимі); значення 1 активує буферизацію рядків (використовується лише в текстовому режимі); ціле число > 1, вкаує розмір у байтах буфера фрагментів фіксованого розміру. При значеннях 0 і 1 розмір буферу вказується окремо згідно повної документації цієї функції. Якщо параметр рівний -1, то у більшості випадків розмір буферу визначається автоматично.*

1. В яких режимах можна відкрити файл?

*Параметр - Значення*

*'r' - Відкрити для читання (за замовчуванням)*

*'w' - Відкрити для запису, очистивши попередньо файл, якщо файл істує*

*'x' - Відкрити для ексклюзивного створення, якщо файл уже існує, то функція завершується невдачею*

*'a' - Відкрити для запису, дописуючи в кінець файлу, якщо він існує*

*'b' - Бінарний режим*

*'t' - Текстовий режим (за замовчуванням)*

*'+' - Відкрити для оновлення (читання та запис)*

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?

*Функції у мові python не відрізняються за своєю суттю від функцій С/С++. У Python функції можуть мати довільну кількість параметрів. У цьому випадку їм можна передавати неіменовані або іменовані параметри, або їх комбінацію. Іншим способом передачі довільної кількості аргументів є використання іменованих параметрів, де при виклику аргументи передаються як пари значень – назва аргументу і його значення. Функція у Python не може зовсім не мати тіла. Якщо функція не має, тіла то слід вказати ключове слово pass як тіло такої функції.*

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).*

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

*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.*

1. Як поєднуються обробка виключних ситуацій і оператор with?

*Оператор with часто використовується для кращого управління ресурсами, такими як файли, сокети чи бази даних. Однак він також може бути використаний для обробки виключень, пов'язаних з цими ресурсами. Це забезпечує зручність управління ресурсами та автоматичне вивільнення їх під час виникнення виключень.*

*Висновок*

На данній лабораторній роботі я оволодів навиками використання засобів мови Python для роботи з файлами. Створив модуль, що реалізує функції читання/запису у текстовому і двійковому форматах результатів обчислення виразів, згідно індивідуального завдання.