Міністерствоосвіти і науки

Національний університет „Львівська політехніка”

**Кафедра ЕОМ**

****

**Звіт про виконання лабораторної роботи №8**

З дисципліни: «Кросплатформні засоби програмування»

На тему: «Файли та виключення у Python»

**Виконала:** ст. групи КІ-305

Циніцька Олена

**Прийняв:** доцент каф. ЕОМ

Іванов Ю. С.

**Львів – 2023**

Завдання

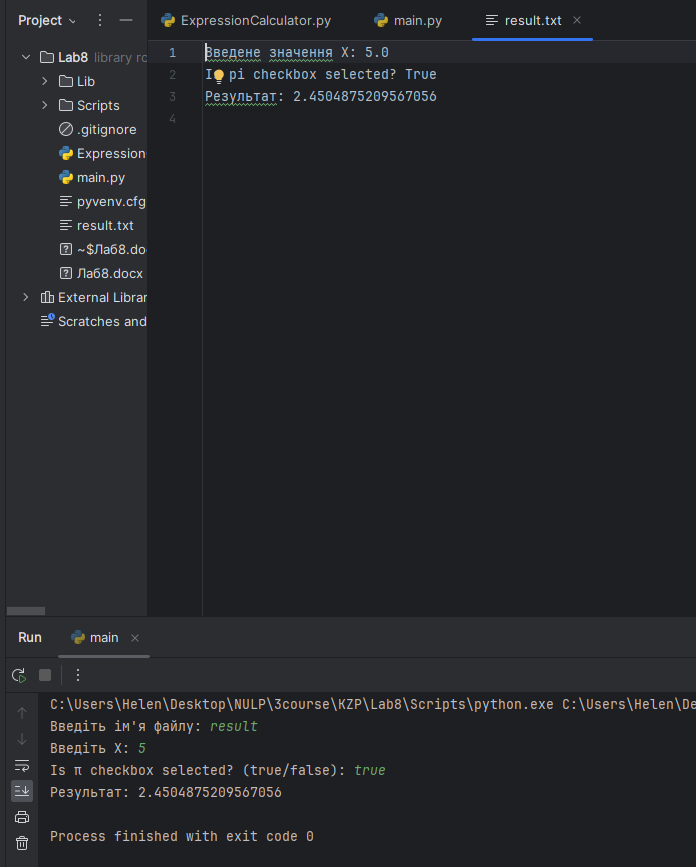
1. Написати та налагодити програму на мові Python згідно варіанту, а саме

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.

import math  
class ExpressionCalculator:  
 @staticmethod  
 def calculate\_expression(x, pi\_check\_box):  
 if pi\_check\_box and (abs(4 \* x) == 0.5 or (4 \* x - 0.5) % 1 == 0):  
 result = 0  
 else:  
 result = 1 / math.cos(4 \* x)  
  
 if math.isinf(result) or math.isnan(result):  
 ExpressionCalculator.write\_result\_to\_file(fileName, x, result)  
 raise ArithmeticError("Division by zero: cos(4x) equals zero.")  
  
 return result  
  
 @staticmethod  
 def write\_result\_to\_file(file\_name, x, result):  
 with open(file\_name + ".txt", "w") as file:  
 file.write(f"Введене значення X: {x}\n")  
 file.write(f"Значення y при x = {x} дорівнює {result}\n")

import math  
from ExpressionCalculator import ExpressionCalculator  
  
def main():  
 file\_name = input("Введіть ім'я файлу: ")  
 x = float(input("Введіть X: "))  
  
 # Get input for pi\_check\_box  
 pi\_check\_box\_input = input("Is π checkbox selected? (true/false): ")  
 pi\_check\_box = pi\_check\_box\_input.lower() == "true"  
  
 fileName = file\_name # Set fileName to file\_name here  
  
 try:  
 with open(file\_name + ".txt", "w", encoding="utf-8") as file\_out:  
 file\_out.write(f"Введене значення X: {x}\n")  
 file\_out.write(f"Is pi checkbox selected? {pi\_check\_box}\n") # Changed "π" to "pi"  
  
 result = ExpressionCalculator.calculate\_expression(x, pi\_check\_box)  
 file\_out.write(f"Результат: {result}\n")  
 print(f"Результат: {result}")  
  
 except FileNotFoundError as ex:  
 print("Error writing to file:", ex)  
 except ArithmeticError as ex:  
 print("Division by zero: cos(4x) equals zero.")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

****

**Відповіді на контрольні питання**

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

В мові Python виключні ситуації обробляються за допомогою конструкції "try-except."

1. **Особливості роботи блоку except?**

* У блоку "except" вказується код, який виконується, якщо виникає виключна ситуація.
* Можна вказати один або декілька типів виключних ситуацій, які будуть оброблятися цим блоком.
* Блок "except" виконується лише, якщо відповідна виключна ситуація виникла в блоку "try."

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

Для відкривання файлів у Python використовується функція "open."

1. **Особливості використання функції open?**

* Функція "open" приймає два аргументи: ім'я файлу і режим відкриття.
* Режими включають "r" (читання), "w" (запис), "a" (додавання), "rb" (читання в двійковому режимі), "wb" (запис в двійковому режимі) та інші.
* Функція "open" повертає об'єкт файлу, який використовується для подальшої роботи з файлом.

1. **В яких режимах можна відкрити файл?**

Файл можна відкрити в різних режимах, таких як "r" (читання), "w" (запис), "a" (додавання), "rb" (читання в двійковому режимі), "wb" (запис в двійковому режимі) і багато інших.

1. **Як здійснити читання і запис файлу?**

Для читання файлу використовується функція "read," а для запису - функція "write."

1. **Особливості функцій у мові Python?**

Особливості функцій у мові Python включають можливість передавати аргументи, повертати значення, та обробляти виключні ситуації за допомогою блоку "try-except."

1. **Для чого призначенйи оператор with?**

Оператор "with" призначений для створення контексту, в якому можна автоматично відкривати і закривати ресурси, такі як файли. Він гарантує, що ресурс буде коректно закритий після виходу з контексту.

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

До об'єктів, які передаються під контроль оператору "with," ставляться вимоги щодо наявності методів "enter" і "exit," які визначають, як відкривати і закривати ресурси.

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

Обробка виключних ситуацій може бути поєднана з оператором "with," якщо виключна ситуація виникає в контексті "with," то метод "exit" буде викликаний автоматично для закриття ресурсів.

**Висновок**

Завдання лабораторної роботи виконано, а саме у мові Python були реалізовані функції читання і запису файлів у різних форматах, використовуючи конструкції "try-except" для обробки виключних ситуацій. Код був організований в окремому модулі та містить коментарі для кращого розуміння