Міністерство освіти і науки України

Національний університет «Львівська політехніка»



**ЗВІТ**

з лабораторної роботи №8

з дисципліни: “Кросплатформні засоби програмування”

на тему: “ Файли та виключення у Python”

Виконав:

ст. гр. КІ-305

Ніколенко О.В.

Прийняв:

Іванов Ю.С.

Львів – 2023

**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.

**Варіант №20**



**Лістинг програми:**

import math

class CalcException(Exception):

pass

class Equations:

def calculate(self, x):

rad = math.radians(x)

try:

y = math.tan(rad) \* (1 / math.tan(2 \* rad))

if x in [90, -90, 0, 180, -180] or 2 \* x in [90, -90, 180, -180]:

raise ZeroDivisionError()

except ZeroDivisionError:

if x in [90, -90, 180, -180] or 2 \* x in [90, -90, 180, -180]:

raise CalcException("Exception reason: Illegal value of X for tangent calculation")

elif x == 0:

raise CalcException("Exception reason: X = 0")

else:

raise CalcException("Unknown reason of the exception during exception calculation")

return y

def main():

print("Success")

try:

file\_name = input("Enter file name: ")

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

try:

eq = Equations()

x = int(input("Enter X: "))

result = eq.calculate(x)

fout.write(str(result))

print(result)

except CalcException as ex:

print(ex)

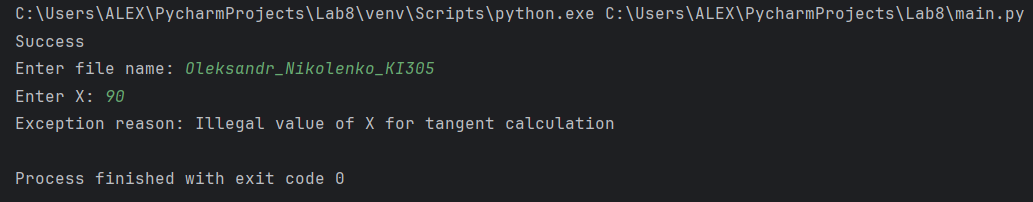
except FileNotFoundError:

print("Exception reason: Perhaps wrong file path")

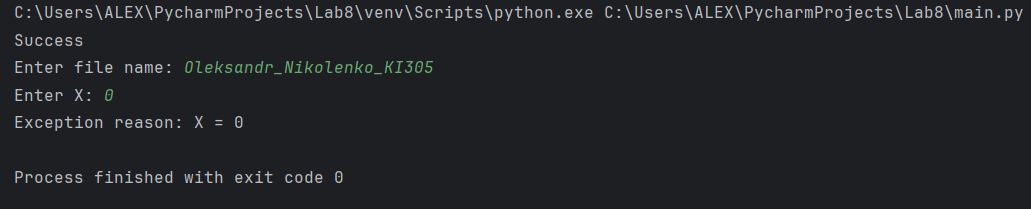
if \_\_name\_\_ == "\_\_main\_\_":

main()

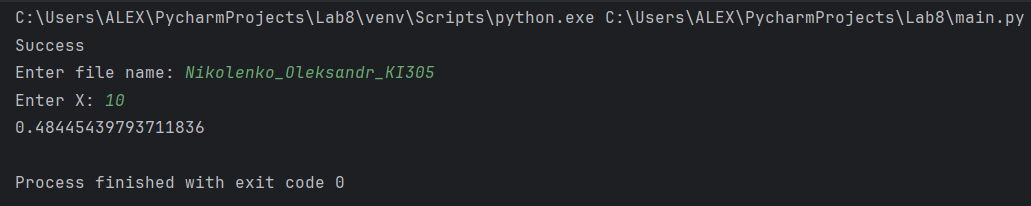
**Результат виконання програми:**

****

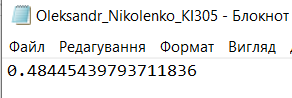
*Рис.1.Виключення для неприпустимого значення для котангеса*

****

*Рис.2.Виключення при значенні нуля*



*Рис.3. Результат виконання програми*



*Рис.4.Результат виконання програми у файлі*

**Відповіді на контрольні запитання:**

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

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

1. **Особливості роботи блоку except?**

* Блок except розміщується після блоку try і може містити один або декілька обробників винятків
* Кожен except може мати специфікований тип винятку, який він обробляє.
* Python дозволяє також використовувати загальні обробники винятків без вказівки конкретного типу.
* Обробники визначаються в порядку зверху вниз.
* В кожному except блоку ви можете вказати ім'я змінної, яка буде посилатися на об'єкт винятку.
* Багаторазове використання except

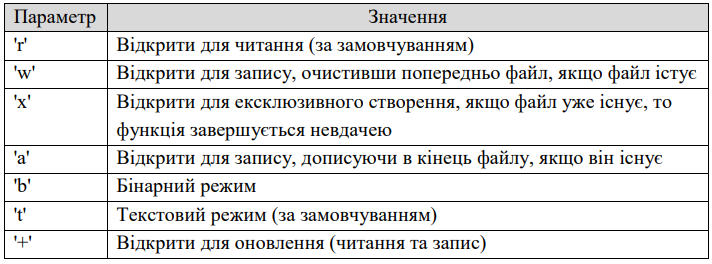
1. **Яка функція використовується для відкривання файлів у Python?**

Ключовою функцією для роботи з файлами є функція open(file, mode='r', buffering=-1, encoding=None, errors=None, newline=None, closefd=True, opener=None).

1. **Особливості використання функції open?**

* Режими відкривання файлу
* Закриття файлу
* Робота з файловими об’єктами
* Обробка помилок
* Кодування
* Пошук файлу

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

****

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?**

* Синтаксис визначення функцій
* Аргументи функції
* Повернення значення
* Область видимості
* Рекурсія
* Анонімні функції
* Замикання
* Документація функцій
* Декоратори
* Іменовані аргументи

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

Оператор with використовується для автоматизації процесів закриття ресурсу і коректної обробки виключних ситуацій (аналог оператора try-з-ресурсами у Java). Наприклад, автоматичне закриття файлу, чи з’єднання після завершення роботи з ним, а також, при виникненні виключень.

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

* Присутність методів ‘\_\_enter\_\_’ та ‘\_\_exit\_\_’
* Можливість здійснювати налаштування перед входом
* Можливість здійснювати очищення після виходу
* Обробка винятків

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

* Обробка винятку всередині блоку with
* Використання блоку finally
* Каскадне відкриття і обробка винятків

**Висновок:** я оволодів навиками використання засобів мови Python для роботи з файлами.