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-306

Фодор Андрій

**Прийняв:**

доцент кафедри ЕОМ

Іванов Ю. С.

Львів – 2023

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

**Завдання (варіант № 21)**

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 21. y=sin(3x-5)/ctg(2x)**

**Вихідний код програми**

import math

import struct

class FileWorkAndCalc:

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

self.result = 0.0

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

# Writing result to a text file

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

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

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

try:

# Reading result from a text file

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:

# Writing result to a binary file

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

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

except (FileNotFoundError, IOError) as ex:

print(ex)

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

try:

# Reading result from a binary file

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

packed\_data = file.read(8) # Assuming a double is 8 bytes

self.result = struct.unpack('d', packed\_data)[0]

except (FileNotFoundError, IOError) as ex:

print(ex)

def calculate(self, x):

try:

rad = x \* math.pi / 180.0

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

y = math.sin(3 \* rad - 5) / y

if math.isnan(y) or math.isinf(y) or x == 0 or x == 90:

raise ArithmeticError

except ArithmeticError:

if rad == math.pi / 2.0 or rad == 0.0:

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

elif math.tan(2 \* rad) == 0 or 1 / math.tan(2 \* rad) == 0:

raise CalcException("Exception reason: Division by 0")

else:

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

self.result = y

def get\_result(self):

return self.result

class CalcException(ArithmeticError):

def \_\_init\_\_(self, cause=None):

super().\_\_init\_\_(cause)

def main():

obj = FileWorkAndCalc()

print("Enter x: ", end='')

try:

data = input()

obj.calculate(float(data))

print("Result is:", obj.get\_result())

obj.write\_res\_txt("test1.txt")

obj.write\_res\_bin("test2.bin")

obj.read\_res\_bin("test2.bin")

print("Result from binary file is:", obj.get\_result())

obj.read\_res\_txt("test1.txt")

print("Result from test file is:", obj.get\_result())

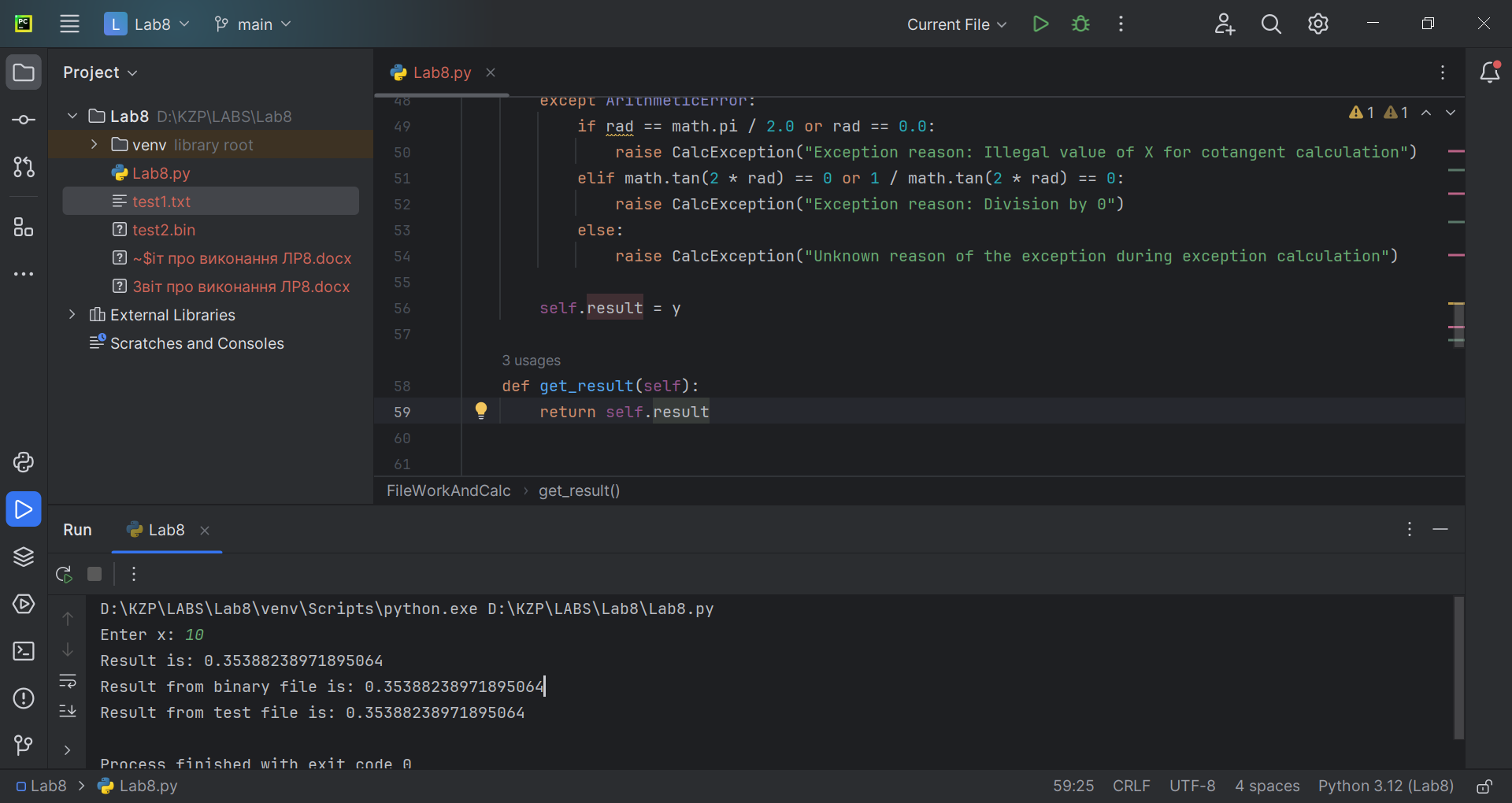
except (ValueError, CalcException) as ex:

print(ex)

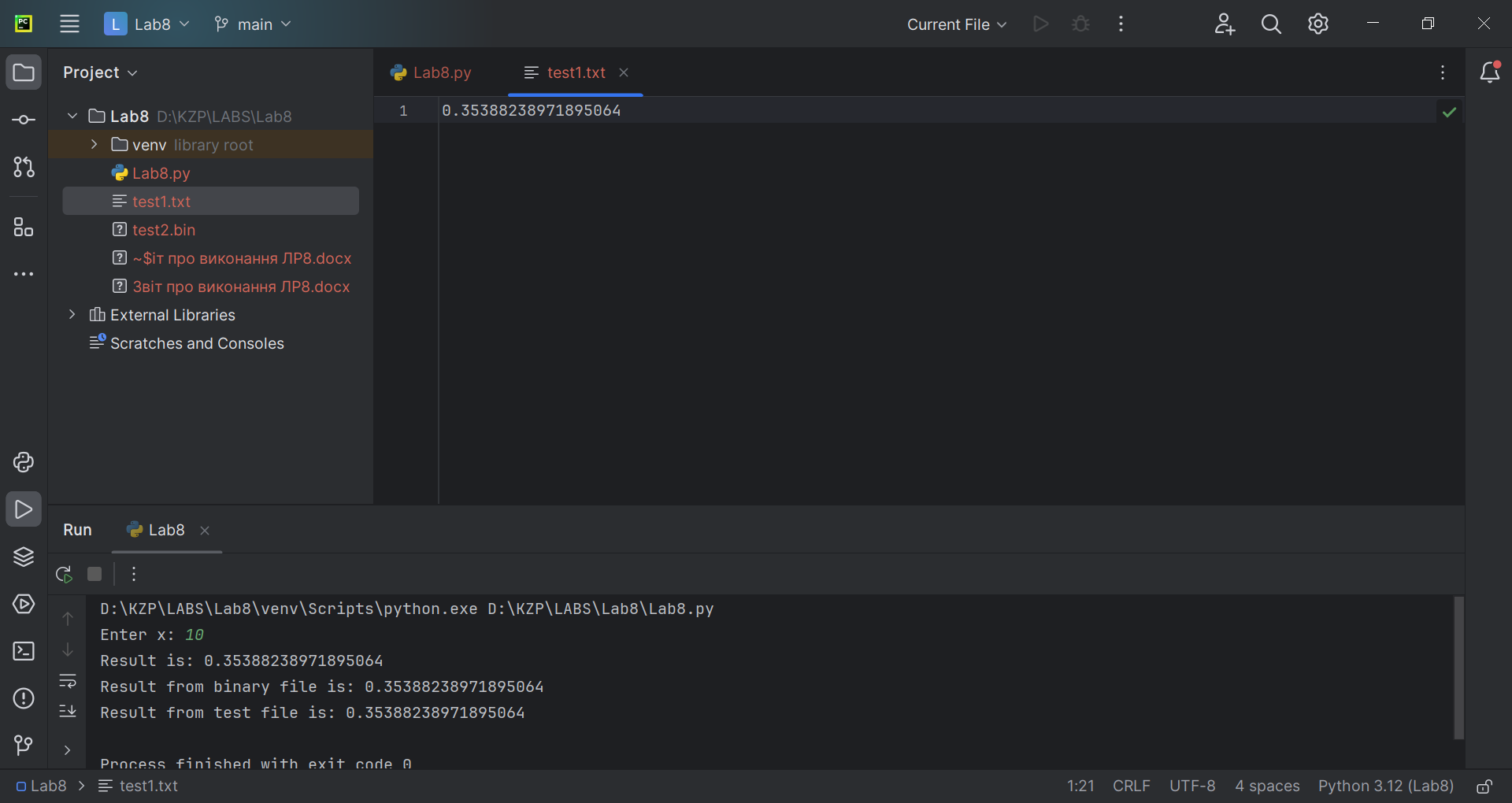
if \_\_name\_\_ == "\_\_main\_\_":

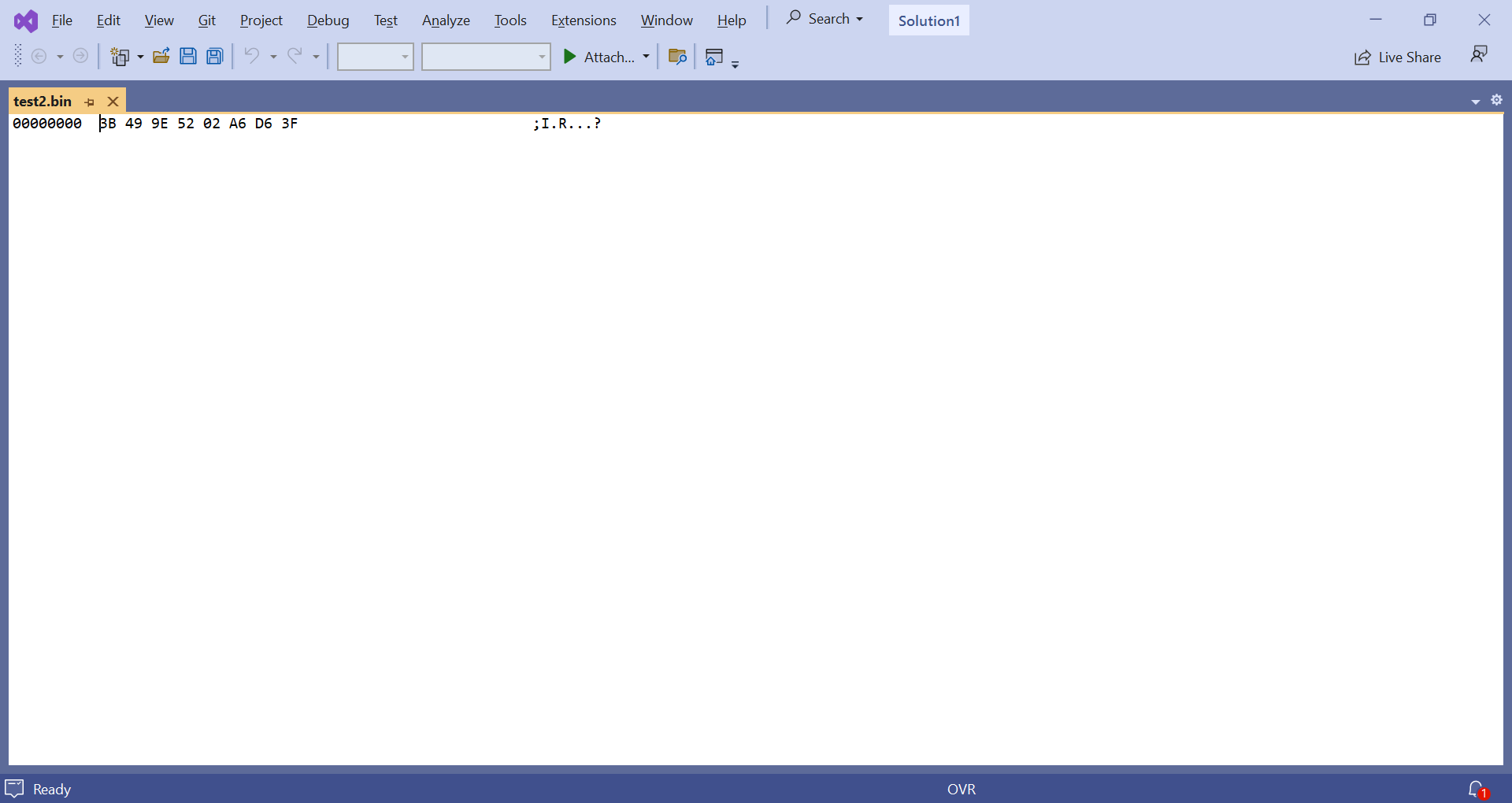
main()

**Результат виконання програми**

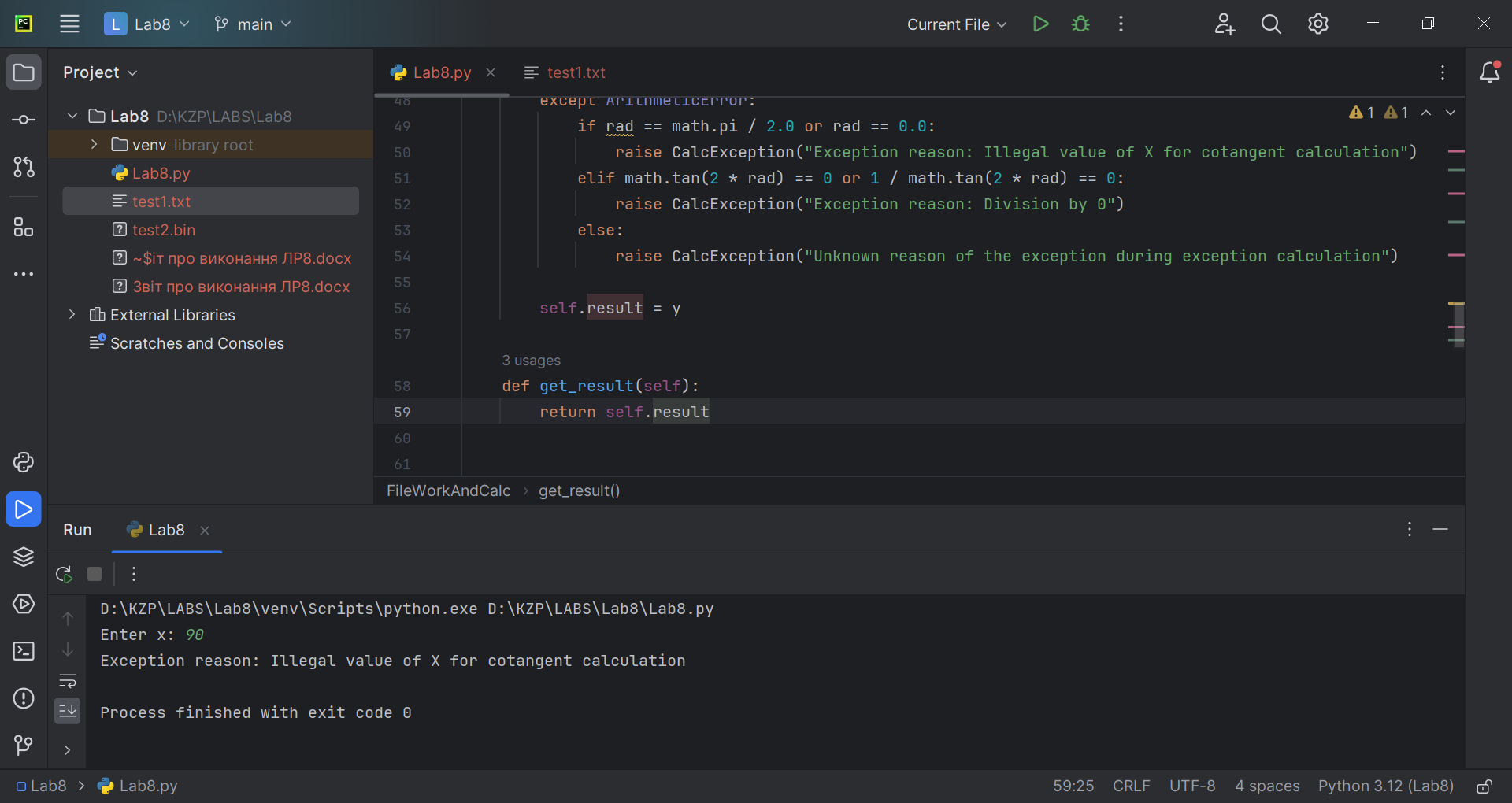


Вміст текстового та бінарного файлів відповідно:





При цьому за допомогою опрацювань виключень програма не починає обчислювати котангенс 0 або 180 градусів, оскільки такі величини не входять в ОДЗ цієї тригонометричної функції. (90\*2=180)



**Висновок:** на цій лабораторній роботі я оволодів навиками використання засобів мови Python для роботи з файлами та виключеннями. Закріпив отримані знання на прикладі виконання індивідуального завдання