

Міністерство освіти і науки України  
Національний університет «Львівська політехніка»

Кафедра ЕОМ



**Звіт**

до лабораторної роботи № 8

З дисципліни: «Кросплатформенні засоби програмування»

На тему: «**ФАЙЛИ ТА ВИКЛЮЧЕННЯ У PYTHON**»

Варіант 16

**Виконав:** ст.гр. КІ-301

Онисько М.М.

**Прийняв:** доц.

Майдан М.В.

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:**

**Виконання роботи:**

**Код програми:**

```
import math

class CalcException(ArithmeticError):
    def __init__(self, cause=None):
        super().__init__(cause)

class Equations:
    def calculate(self, x):
        y = 0
        rad = 2*x - 4
        try:
            y = 7*x / math.tan(rad)
            if math.isnan(y) or math.isinf(y) or x == 90 or x == -90:
                raise ArithmeticError()
        except ArithmeticError:
            if rad == math.pi/2.0 or rad == -math.pi/2.0:
                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():
    try:
        print("Enter file name: ")
```

```

fName = input()
with open(fName, 'w') as fout:
    try:
        eq = Equations()
        print("Enter X: ")
        x = int(input())
        fout.write(str(eq.calculate(x)))
    finally:
        fout.flush()
except FileNotFoundError:
    print("Exception reason: Perhaps wrong file path")
except CalcException as ex:
    print(ex.args[0])

if __name__ == "__main__":
    main()

```

The screenshot shows a Python IDE (VS Code) with the following components:

- EXPLORER:** Shows a file tree with 'Lab7Onysko.py', 'Lab8Onysko.py', and 'Lab9Onysko.py'.
- EDITOR:** Displays the code from the previous block, with line numbers 1 through 24.
- TERMINAL:** Shows the execution output:
 

```

PS C:\Users\onisk\Desktop\K301 4\PythonLab> & C:\Users\onisk\AppData\Local\Programs\Python\Python311\python.exe "c:/Users/onisk/Desktop/K301 4/PythonLab/Lab8Onysko.py"
Enter file name:
sss
Enter X:
10
PS C:\Users\onisk\Desktop\K301 4\PythonLab>

```
- STATUS BAR:** Shows the current file is 'Lab8Onysko.py', line 10, column 22, using UTF-8 encoding, CRLF line endings, and Python 3.11.3 64-bit.

Рис.1 Код та вивід програми

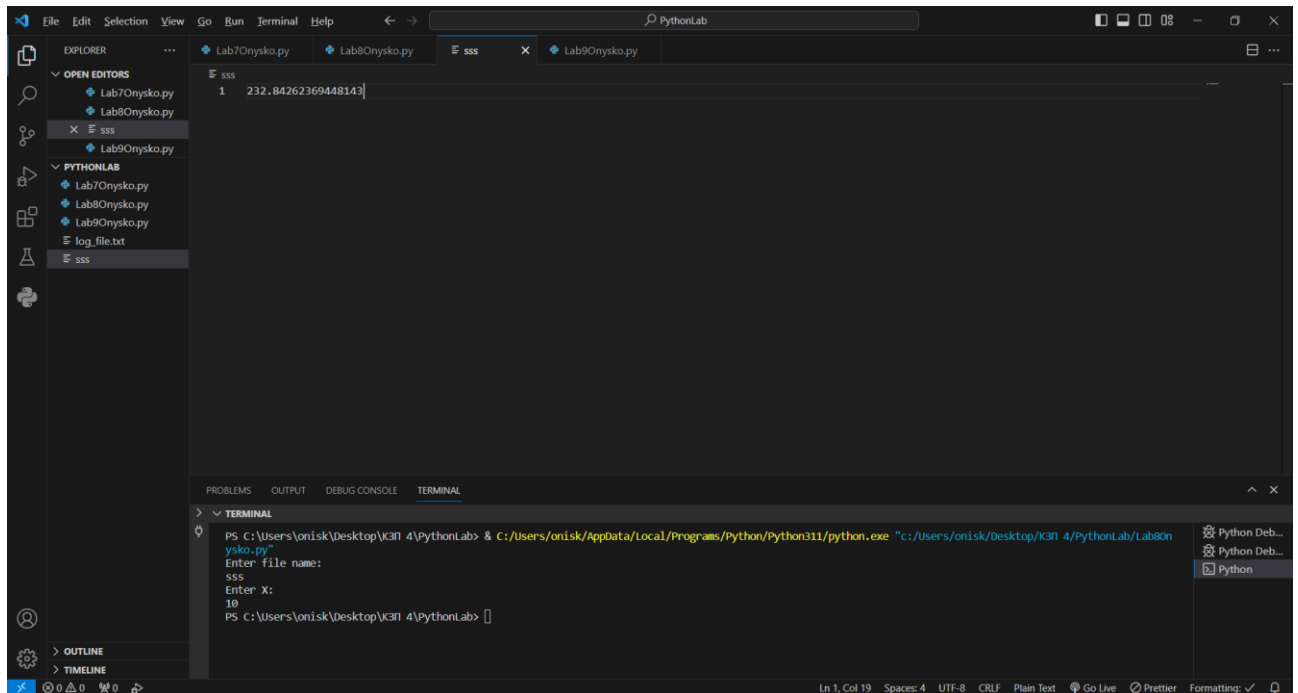


Рис.2 Записана інформація у файл

### Висновок:

Під час виконання завдання були освоєні навички роботи з файлами в мові Python, включаючи відкриття, читання, запис та закриття файлів. Також було здобуто розуміння використання текстового та двійкового режимів, а також обробки виразів для обчислення результатів.

### Контрольні питання:

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

Обробка виключень в Python виконується за допомогою конструкції try-except.

2. Особливості роботи блоку except?

Блок except використовується для обробки виключень, які виникають у блоку try. Коли виникає виключення, існуючі блоки except перевіряються на відповідність типу винятку, і виконується той, який підходить до типу винятку.

3. Яка функція використовується для відкривання файлів у Python?

Для відкриття файлів у Python використовується функція open().

4. Особливості використання функції open?

Функція `open()` використовується для створення об'єкту файлу. Її особливості включають параметри, такі як шлях до файлу, режим відкриття, кодування та інші параметри.

5. В яких режимах можна відкрити файл?

Файл можна відкрити у різних режимах, таких як читання ('r'), запис ('w'), додавання ('a'), читання та запис ('r+'), тощо.

6. Як здійснити читання і запис файлу?

Читання файлу здійснюється за допомогою методу `read()`, а запис - за допомогою методу `write()`.

7. Особливості функцій у мові Python?

Функції в Python можуть мати параметри, можуть повертати значення або бути безповоротними (процедурами).

8. Для чого призначений оператор `with`?

Оператор `with` призначений для автоматичного виклику методів `__enter__` та `__exit__` об'єкта вказаного контекст-менеджера, що вказаний під час входу в блок `with`.

9. Які вимоги ставляться до об'єктів, що передаються під контроль оператора `with`?

Об'єкти, що передаються під контроль оператора `with`, повинні мати методи `__enter__` та `__exit__`, щоб визначити дії, які слід виконати при вході в блок `with` і виході з нього відповідно.

10. Як поєднуються обробка виключних ситуацій і оператор `with`?

Обробка виключень і оператор `with` можуть поєднуватися, наприклад, для автоматичного закриття файлу у випадку виникнення помилки. Можна використовувати блок `try-except` для обробки виключень під час виконання коду в середині блоку `with`.