

Міністерство освіти і науки України  
Національний університет «Львівська політехніка»

Кафедра «ЕОМ»



## **Звіт**

до лабораторної роботи № 8

з дисципліни: «Кросплатформні засоби програмування»

**На тему:** «Файли та виключення у python»

**Виконав:**

студент групи КІ-307

Возний А. О.

**Перевірив:**

доцент кафедри ЕОМ

Іванов Ю. С.

Львів – 2023

**Мета роботи:** 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 завдання:**  $y = \cos(x)/\sin(x)$

### Лістинг програми:

#### Головний модуль

```
from Equations import Equations

x = float(input("Enter x: "))

# Створюємо об'єкт класу Equations
equation_solver = Equations(x)

# Обчислюємо вираз
equation_solver.calculate_expression()

if equation_solver.result is not None:
    # Записуємо результат у текстовий файл
    equation_solver.write_text_file('result.txt',
    str(equation_solver.result))

    # Записуємо результат у двійковий файл
    equation_solver.write_binary_file('result.bin', equation_solver.result)

    # Зчитуємо дані з текстового файлу
    text_data = equation_solver.read_text_file('result.txt')
    print(f"Зчитані дані з текстового файлу: {text_data}")

    # Зчитуємо дані з двійкового файлу
    binary_data = equation_solver.read_binary_file('result.bin')
    print(f"Зчитані дані з двійкового файлу: {binary_data}")
```

## Клас Equations

```
import math
import pickle

class Equations:
    def __init__(self, x):
        self.x = x
        self.result = None

    def calculate_expression(self):
        """
        Обчислює вираз  $y = \cos(x) / \sin(x)$ 

        :return: Результат виразу
        """
        try:
            self.result = math.cos(self.x) / math.sin(self.x)
        except ZeroDivisionError:
            print("Помилка:  $\sin(x)$  не може бути рівним нулю.")
        except Exception as e:
            print(f"Помилка: {e}")

    def read_text_file(self, filename):
        """
        Зчитування даних з текстового файлу.
        """
        try:
            with open(filename, 'r') as file:
                data = file.read()
            return data
        except Exception as e:
            print(f"Помилка при зчитуванні з файлу {filename}: {e}")
            return None

    def write_text_file(self, filename, data):
        """
        Запис даних у текстовий файл.
        """
        try:
            with open(filename, 'w') as file:
                file.write(data)
            print("Дані у текстовий файл успішно записані!")
            return True
        except Exception as e:
            print(f"Помилка при записі у файл {filename}: {e}")
            return False

    def read_binary_file(self, filename):
        """
        Зчитування даних з двійкового файлу.
        """
        try:
            with open(filename, 'rb') as file:
                data = pickle.load(file)
            return data
        except Exception as e:
            print(f"Помилка при зчитуванні з файлу {filename}: {e}")
            return None

    def write_binary_file(self, filename, data):
        """
        Запис даних у двійковий файл.
        """
```

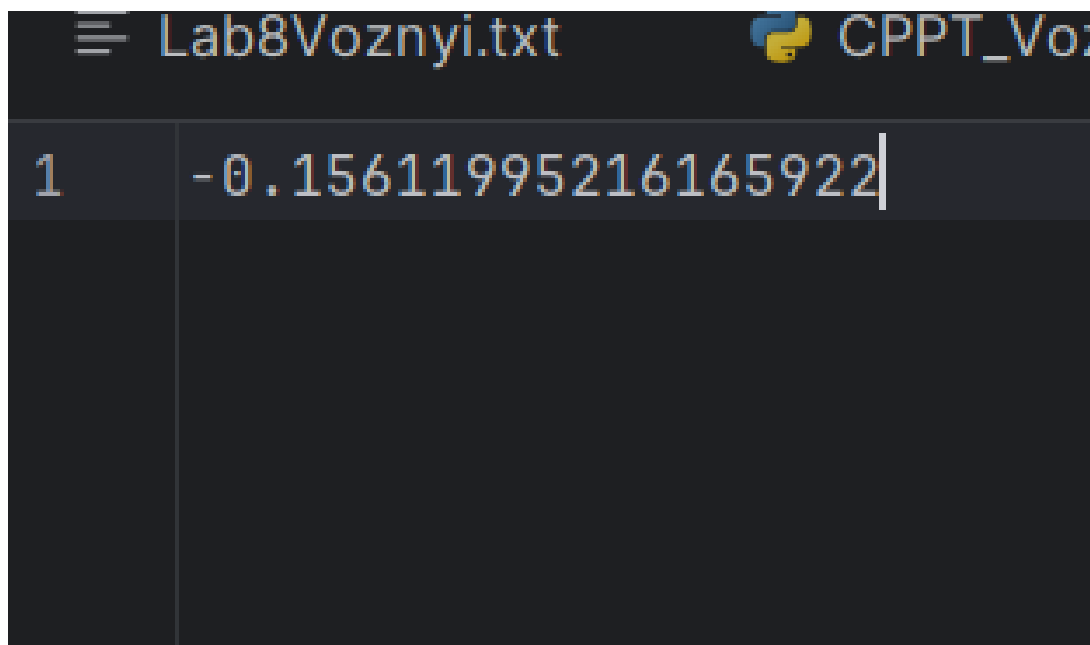
```
try:
    with open(filename, 'wb') as file:
        pickle.dump(data, file)
    print("Дані у двійковий файл успішно записані!")
    return True
except Exception as e:
    print(f"Помилка при записі у файл {filename}: {e}")
    return False
```

### Результат виконання програми:

```
Enter x: 30
Дані у текстовий файл успішно записані!
Дані у двійковий файл успішно записані!
Зчитані дані з текстового файлу: -0.15611995216165922
Зчитані дані з двійкового файлу: -0.15611995216165922

Process finished with exit code 0
```

Рис. 1. Результат виконання програми



The image shows a text editor window with the title 'Lab8Voznyi.txt'. The editor contains a single line of text: '-0.15611995216165922'. The text is displayed in a monospaced font with a light blue background.

1	-0.15611995216165922
---	----------------------

Рис. 2. Результат виконання програми в текстовому файлі

## ***Відповіді на контрольні питання:***

### **1. Обробка виключень:**

У мові Python обробка виключень використовує конструкцію `try...except`, яка дозволяє обробляти виняткові ситуації під час виконання програми.

### **2. Особливості роботи блоку `except`:**

Блок `except` використовується для обробки виключень. Він виконується, якщо сталася виняткова ситуація і співпадає з типом винятку.

### **3. Функція для відкривання файлів у Python:**

Для відкриття файлів у Python використовується функція `open()`.

### **4. Особливості використання функції `open`:**

Функція `open()` використовується для відкриття файлів у різних режимах, таких як читання, запис, додавання і бінарний режим.

### **5. Режими відкриття файлу:**

Режими включають 'r' (читання), 'w' (запис), 'a' (додавання), 'b' (бінарний режим) та інші.

### **6. Читання і запис файлу:**

Для читання файлу використовуйте методи `read()` або ітерацію по файловому об'єкту. Для запису - використовуйте метод `write()`.

### **7. Особливості функцій у Python:**

Функції в Python - це фрагменти коду, які виконують певну дію та можуть бути викликані з інших частин програми.

### **8. Призначення оператора `with`:**

Оператор `with` використовується для створення контексту, який автоматично відкриває та закриває ресурси, такі як файли.

### **9. Вимоги до об'єктів, що передаються під контроль оператору `with`:**

Об'єкти, які передаються під контроль оператору `with`, повинні мати методи `__enter__` та `__exit__`.

### **10. Поєднання обробки виключень і оператора `with`:**

Обробка виключень може бути впроваджена у методах `__enter__` та `__exit__` об'єкта, що передається під контроль `with`, для відловлювання і обробки помилок.

**Висновок:** під час виконання цієї лабораторної роботи я оволодів навиками використання засобів мови Python для роботи з файлами.