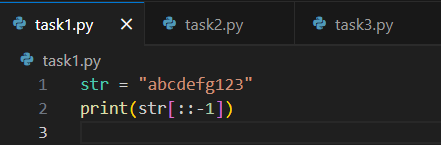
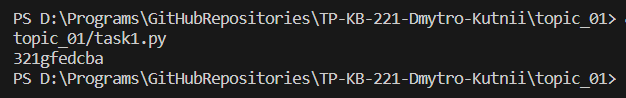
**Звіт про виконання практичних завдань по Технологіях програмування на мові Python**

**Практичне завдання до теми 1**

1. **Перетворення рядка**

Необхідно взяти рядок, що має вигляд «abcdefg123» та перетворити на «321gfedcba».

****

****

**2. Тестування базових функцій для рядків**

**+ -** Додавання рядків

**split -** Розділення рядка на слова Розділення рядка на підрядки за роздільником

**replace -**Заміна тексту Видалення символу з рядка

**upper -** Перетворення на великі літери

**lower -** Перетворення на маленькі літери

**startwith -** Перевірка, чи починається рядок з певного слова

**endwith -** Перевірка, чи закінчується рядок певним словом

**substring in -** Пошук підрядка у тексті

**len -** Визначення довжини рядка

**strip -** Вилучення пробілів з початку і кінця рядка

**\* -** Повторення рядка

**f -** Форматування рядка

**str -** Перетворення числа в рядок

**find -** Визначення позиції підрядка у тексті

**list -** Перетворення рядка у список символів

**capitalize -** Переведення першої літери у велику

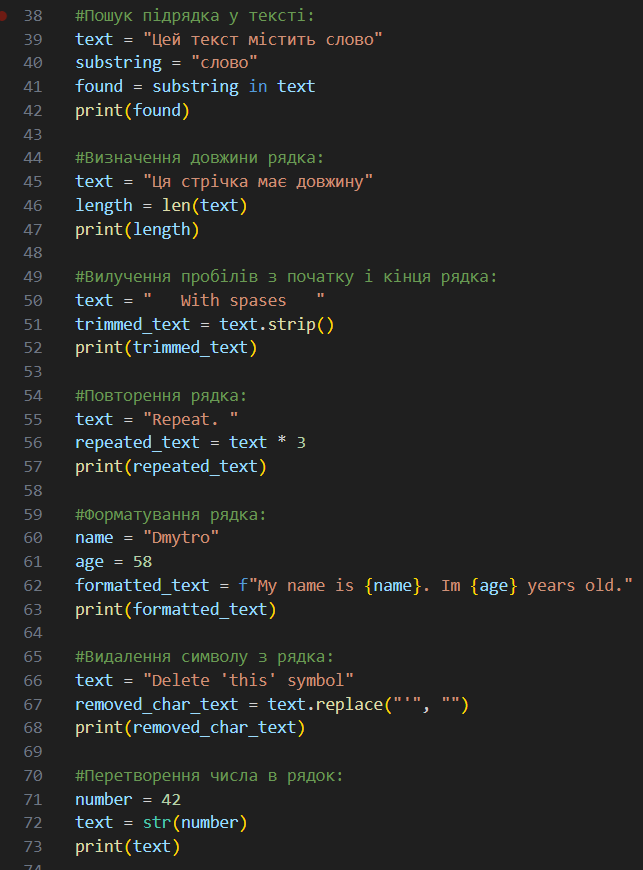
**isalpha -** Перевірка, чи складається рядок лише з букв

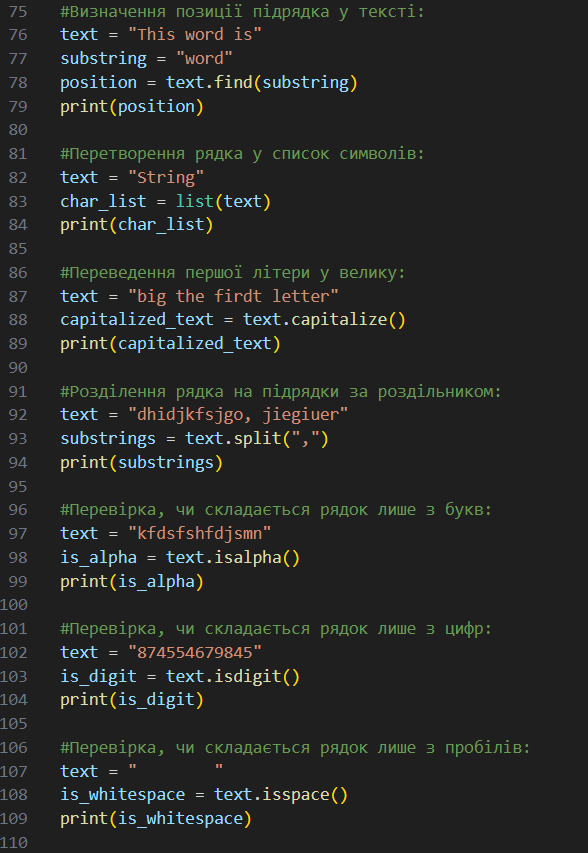
**isdigit -** Перевірка, чи складається рядок лише з цифр

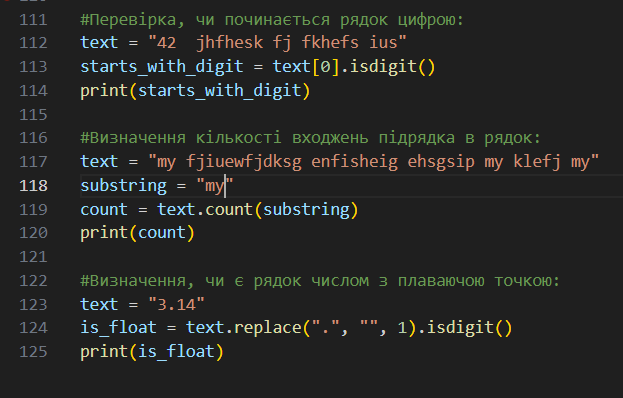
**isspace -** Перевірка, чи складається рядок лише з пробілів

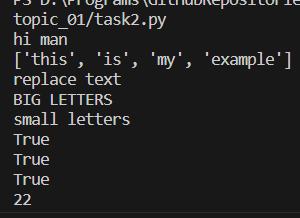
**count -** Визначення кількості входжень підрядка в рядок

****

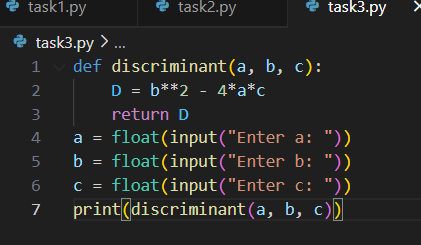
****

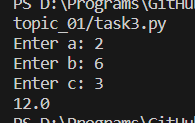
****

****

****

**3. Написання функції пошуку Дискримінанту**

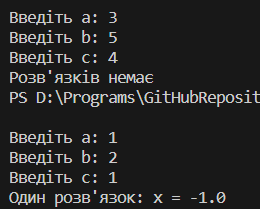
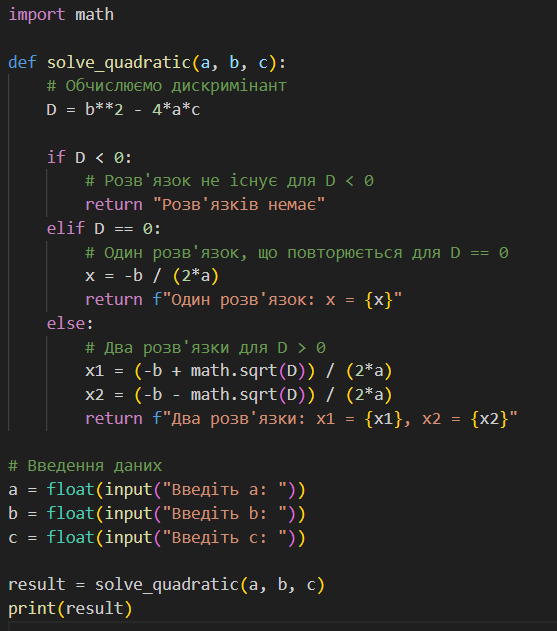
****

****

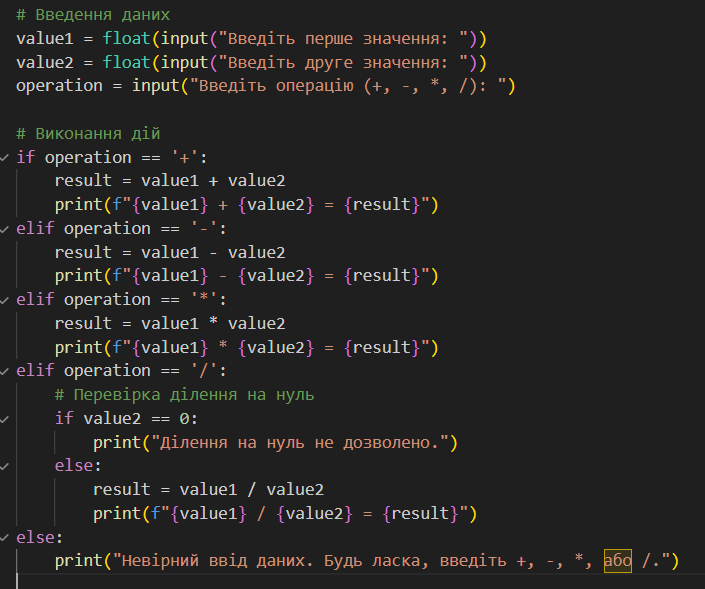
**Висновок:** під час виконання цього практичного завдання з лекції я навчився використовувати вбудовані функції для рядків на мові програмування python.

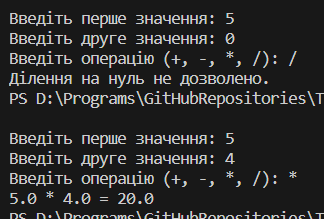
**Практичне завдання до теми 2**

1. **Знайти розв’язки квадратного рівняння за введеними даними.**

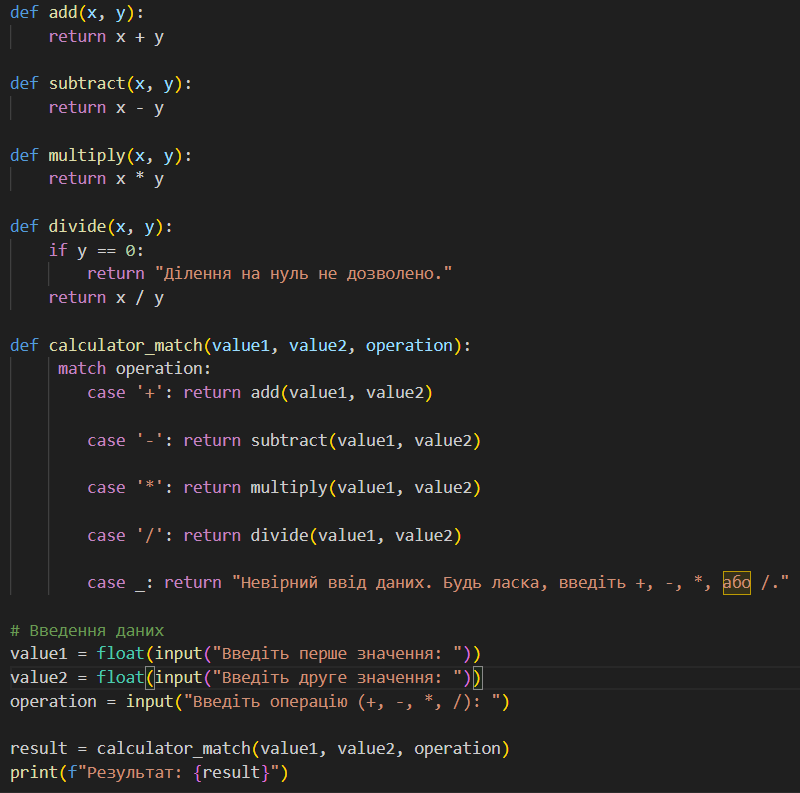
****Потрібно знайти розв’язки квадратного рівняння в залежності від значень дискримінанту.

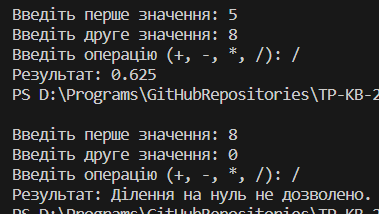
**2. Написання калькулятора через if elif.**

****

****

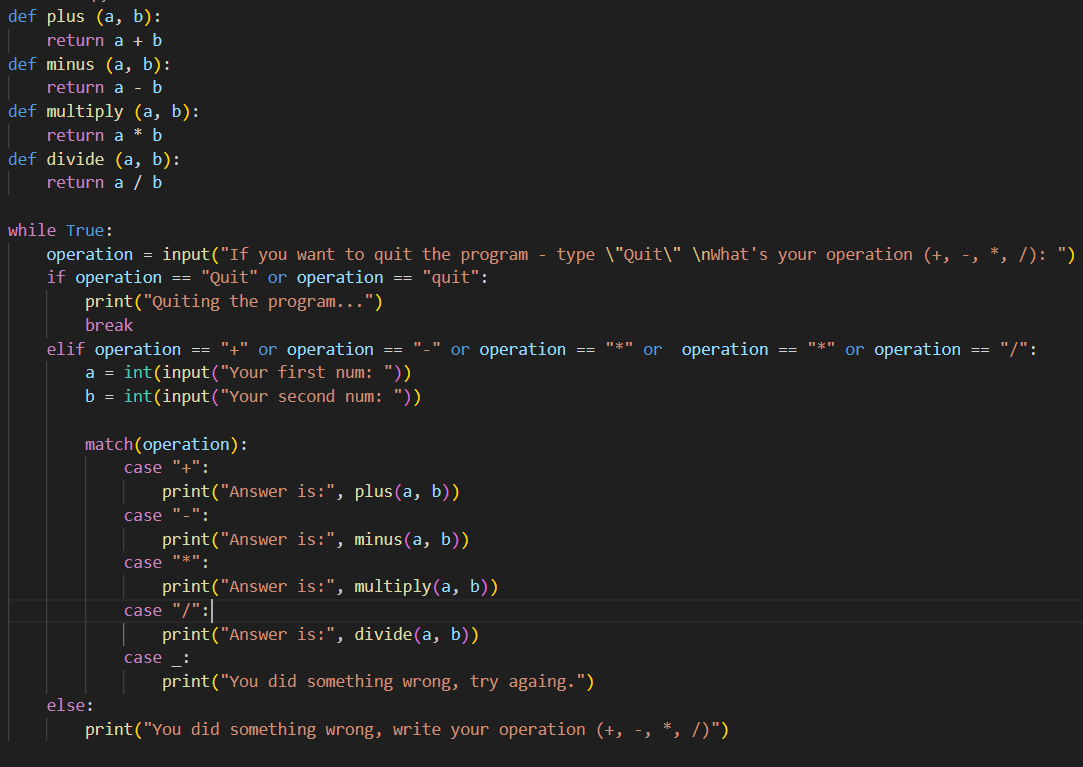
**3. Написання калькулятора через match.**

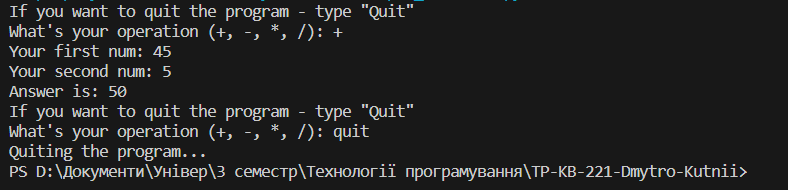
****

****

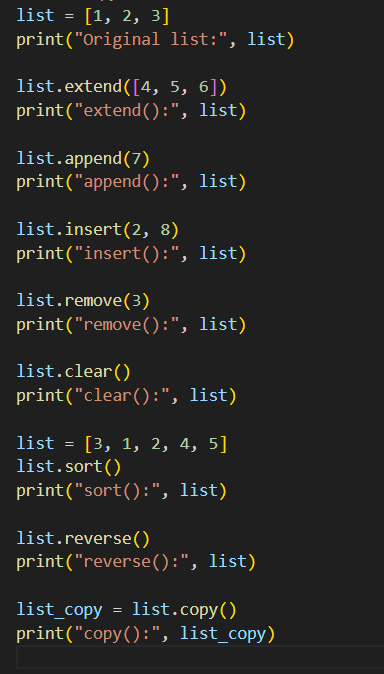
**Висновок:** під час виконання цього практичного завдання з лекції я навчився використовувати вбудовані функції для рядків на мові програмування python.

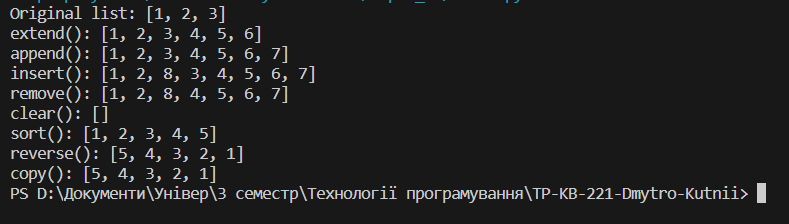
**Практичне завдання до теми 3**

1. Написати програму калькулятор з постійними запитами на введення нових даних та операцій.

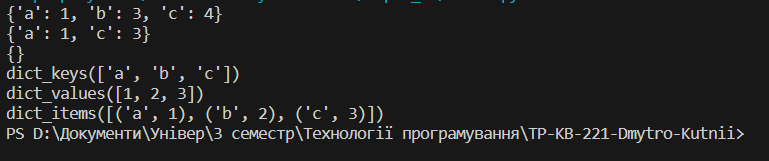


1. Написати програму тестування функцій списків.

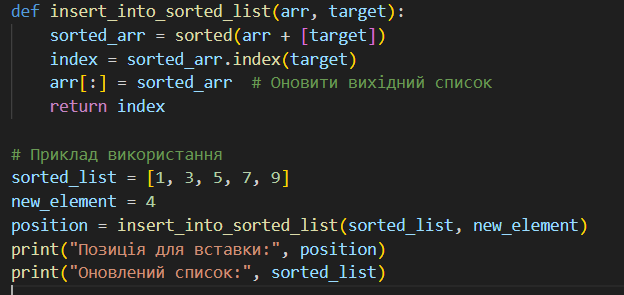




1. Написати програму тестування функцій словників.



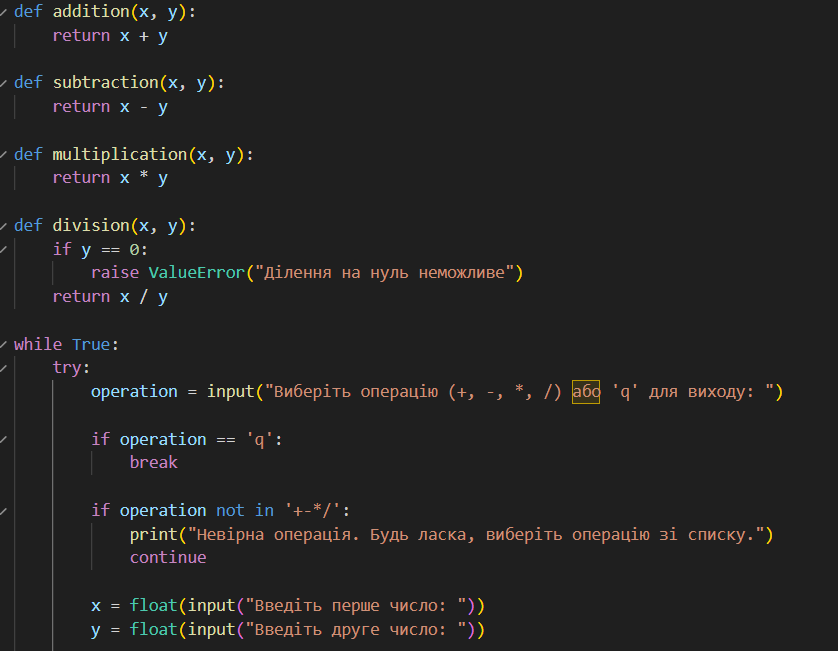
1. Написати функцію пошуку позиції для вставки нового елементу у відсортований список.

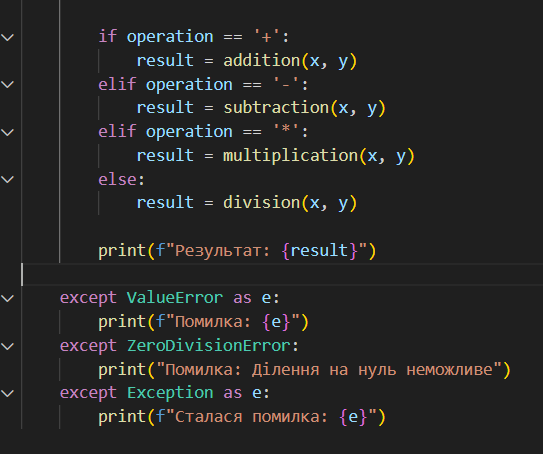


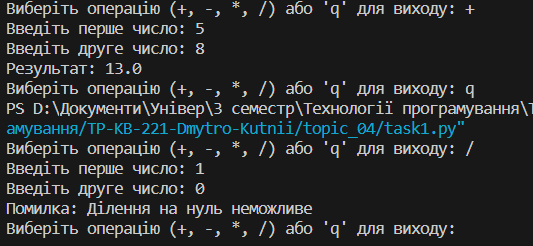
Висновок: в ході виконання даного практичного завдання я створив нескінченний калькулятор через цикл while, навчився тестувати функції списків та словників, а також працювати з відсортованим рядком та додавати в нього нові елементи.

**Практичне завдання до теми 4**

**Наявні 2 завдання:** Розширити програму калькулятор функцією запитів від користувача, що обробляє виняткові ситуації та розширити функцію ділення обробкою виняткової ситуації ділення но нуль.





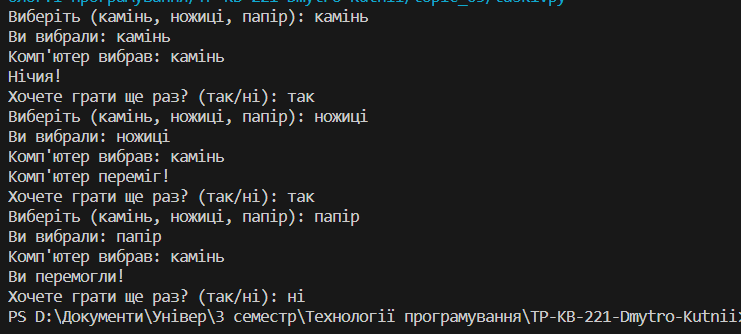


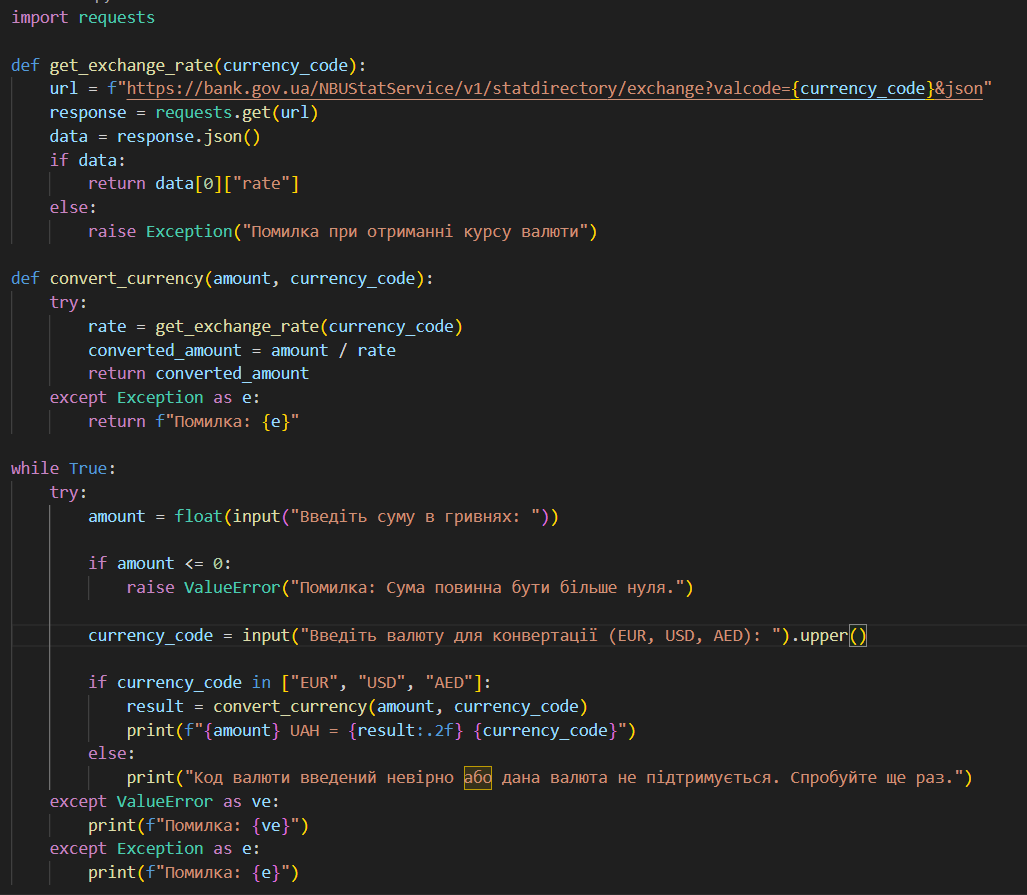
**Висновок**: виконуючи дане практичне завдання я навчився застосовувати виняткові ситуації а також ознайомився з ними.

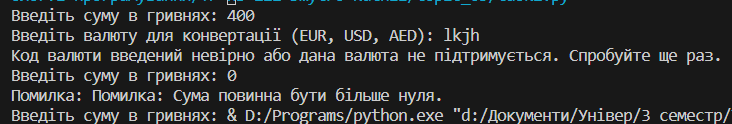
**Практичне завдання до теми 5**

1. Написати гру камінь, ножиці, папір.



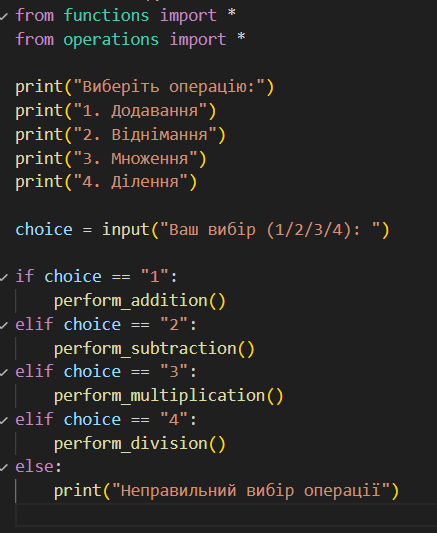


1. Зробити конвертер валют.   
   

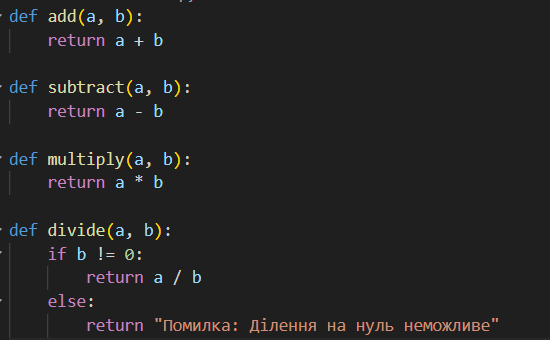


1. Використання модулів для програми калькулятор. Функції додавання, віднімання, множення та ділення перенести в файл functions.py. Функції запиту на введення даних для операцій та самих операцій перемістити в файл operations.py. Програму калькулятор реалізувати в файлі calc.py, до якого підключають файл functions.py та operations.py.

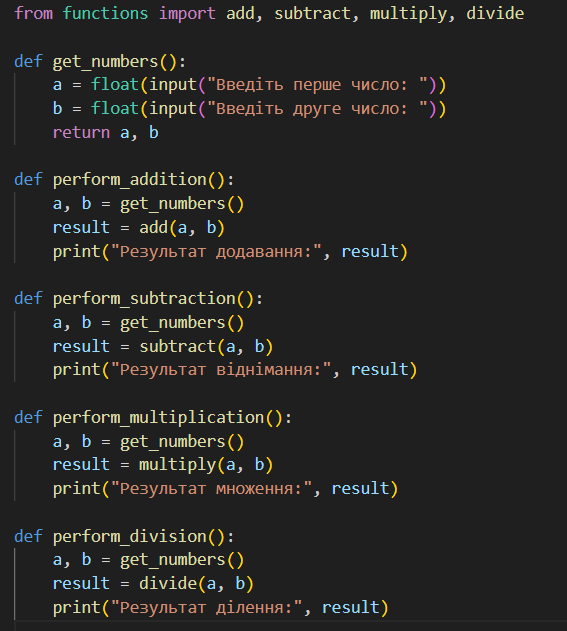
calc.py:



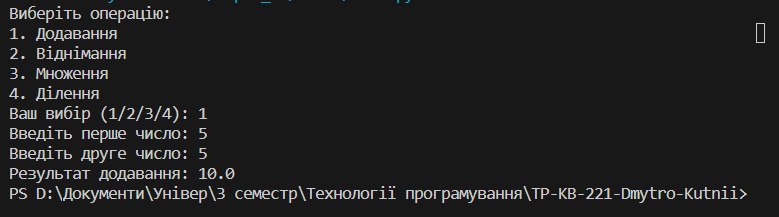
functions.py:



operations.py:



Результат:

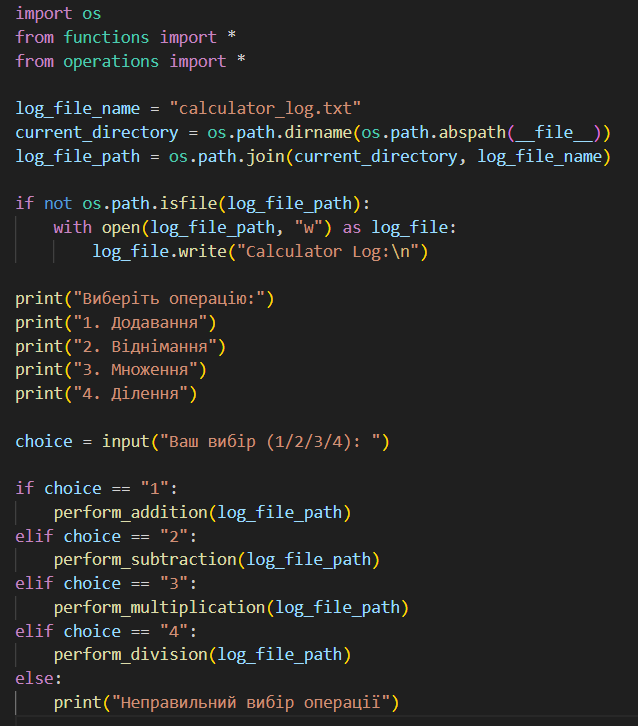


**Висновок**: за допомогою даних практичних завдань я зміг написати та зрозуміти як працює конвертер валют, гра камінь-ножиці-папір, а також розділити програму калькулятора на декілька файлів.

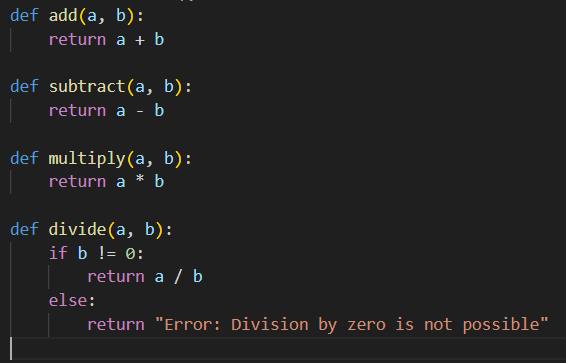
**Практичне завдання до теми 6**

1. Розробити механізм логування всіх дій, що виконує програма. Забезпечити зберігання інформації про введені данні, виконану операцію та результат виконання операції над даними.

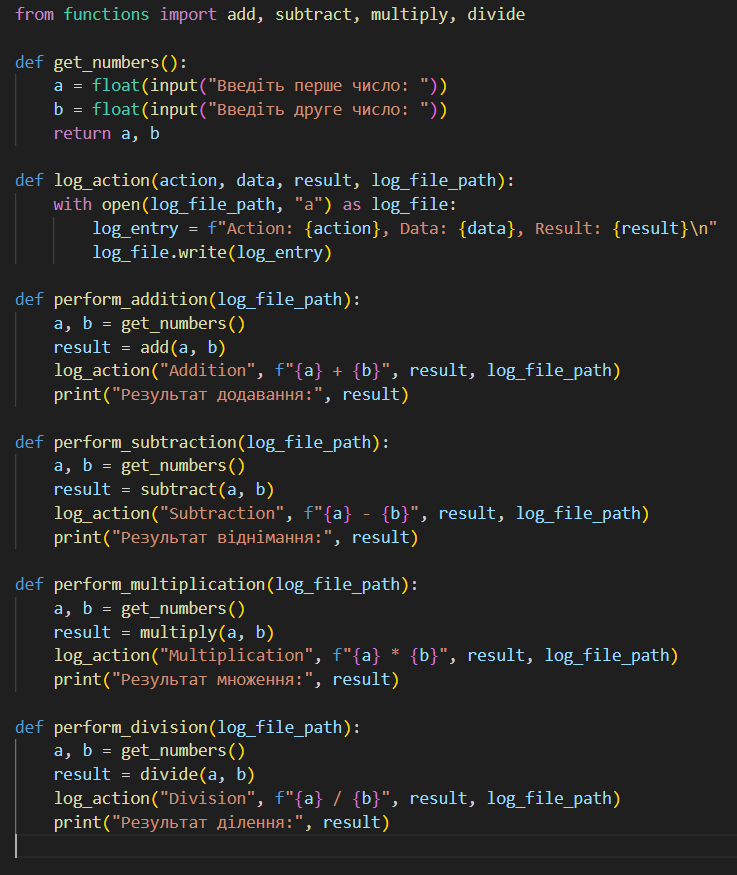
calc.py



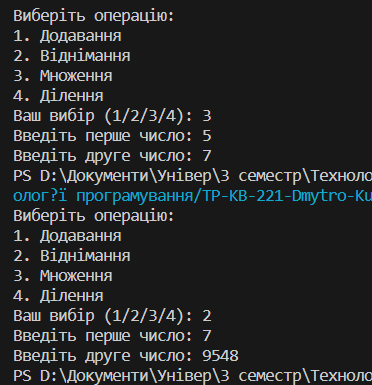
functions.py

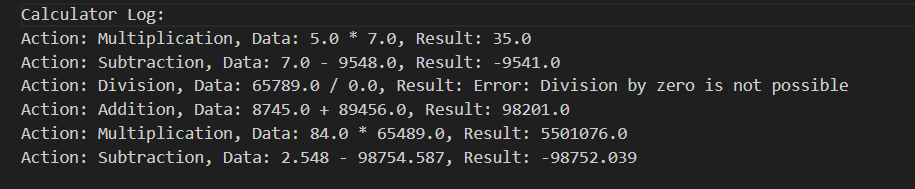


operations.py

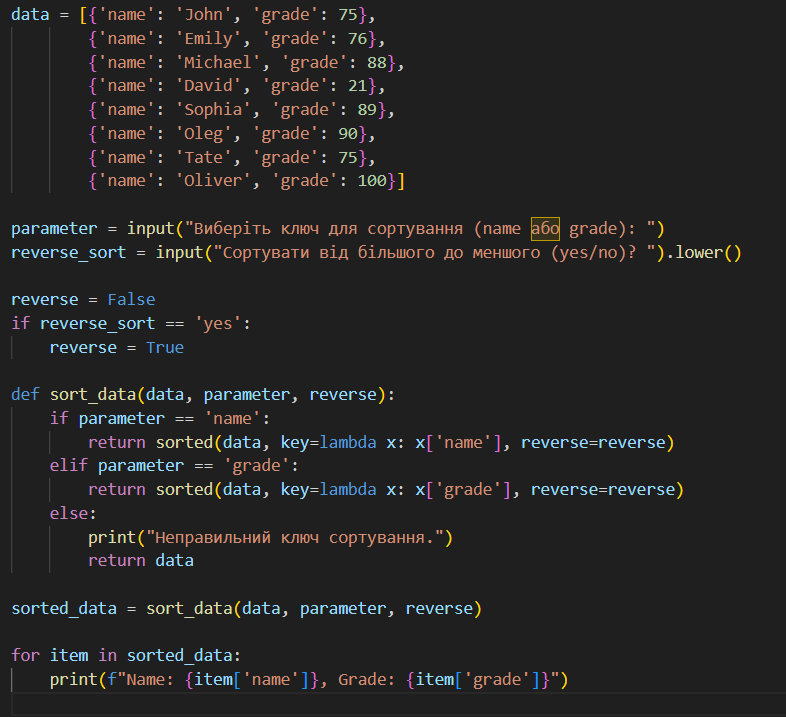


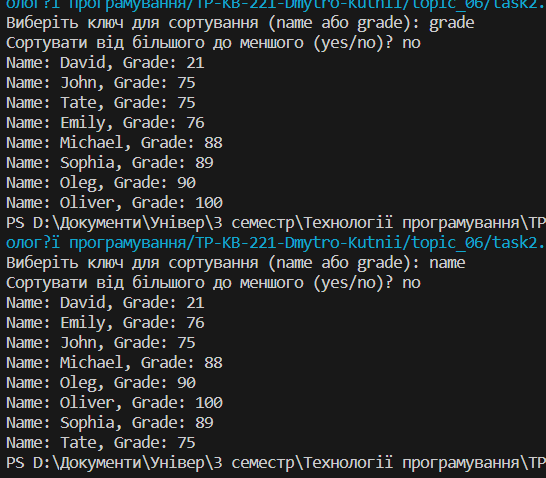
Результат:





1. Маючи не відсортований список, елементами якого є словники з двома параметрами (ім’я та оцінка) виконати сортування списку, використовуючи стандартну функцію sorted(). Другим параметром для функції sorted() має бути lambda функція, що повертає ім’я або оцінку із елемента словника.





**Висновок:** виконуючи дане практичне завдання я навчився записувати зміни за допомогою лог-файлу, а також дізнався що таке lambda функція та навчився її використовувати.

**Практичне завдання до теми 7**

1. Ознайомитись з існуючими за замовченням методами класу по типу \_\_init\_\_(self) \_\_str\_\_(self)\_\_ та надати приклади використання.
2. \_\_init\_\_(self): Цей метод викликається при створенні нового об'єкта класу і використовується для ініціалізації об'єкта. Він приймає self як перший аргумент, який посилається на сам об'єкт, що створюється.

class Person:

def \_\_init\_\_(self, name, age):

self.name = name

self.age = age

person = Person("Джон", 25)

1. \_\_str\_\_(self): Цей метод повертає рядок, який представляє об'єкт у зрозумілому для людини форматі. Він викликається, коли ми використовуєте функцію print() для об'єкта або коли ми перетворюємо об'єкт на рядок за допомогою str().

class Person:

def \_\_init\_\_(self, name, age):

self.name = name

self.age = age

def \_\_str\_\_(self):

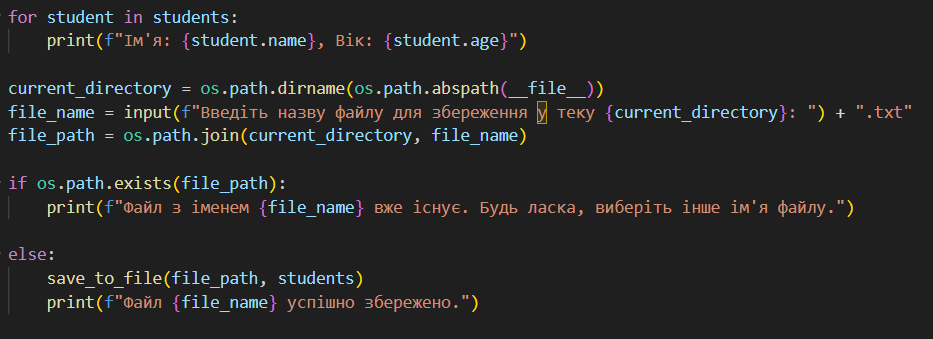
return f"Ім'я: {self.name}, Вік: {self.age}"

person = Person("Джон", 25)

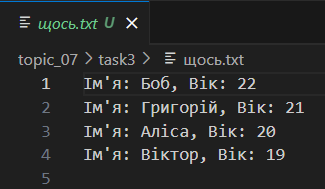
print(person)

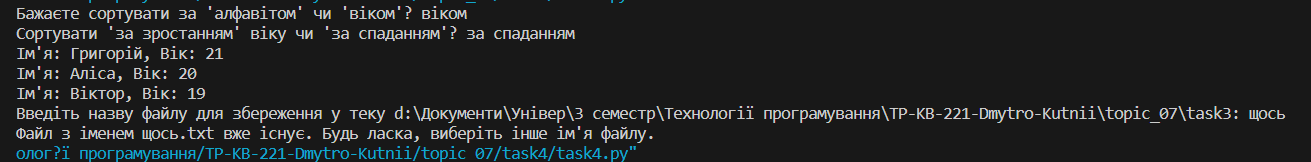
1. Розробити клас **Student** атрибутами якого э два параметра **name** та **age**. Створити список елементами якого є об'єкти класу **Student**. Написати цикл який виводить на екран елементи списку у відсортованому порядку. Для сортування використати стандартну функцію **sorted**. Функція **sorted** має використовувати **lambda** функцію для визначення ключа сортування.





Результат:

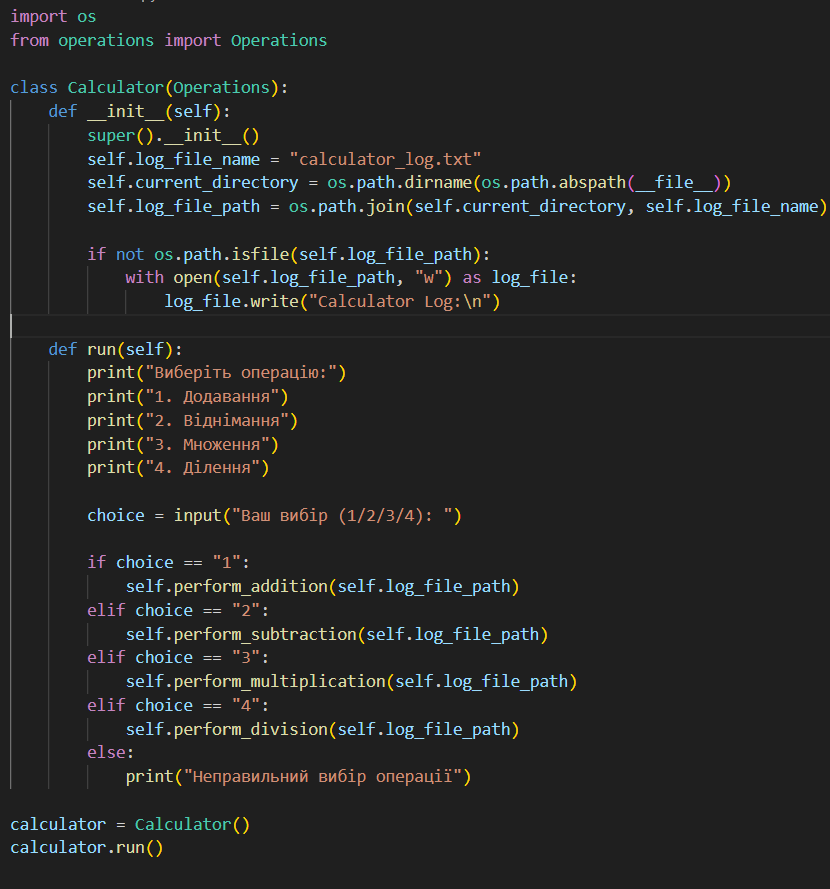




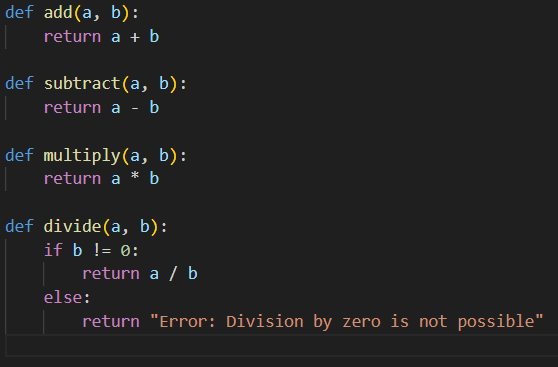
1. Використовуючи принципи ООП переписати програму Калькулятор. Завдання має бути виконано використовуючи модульний підхід.

За основу було взято код який був створений в практичному завданні до попередньої теми.

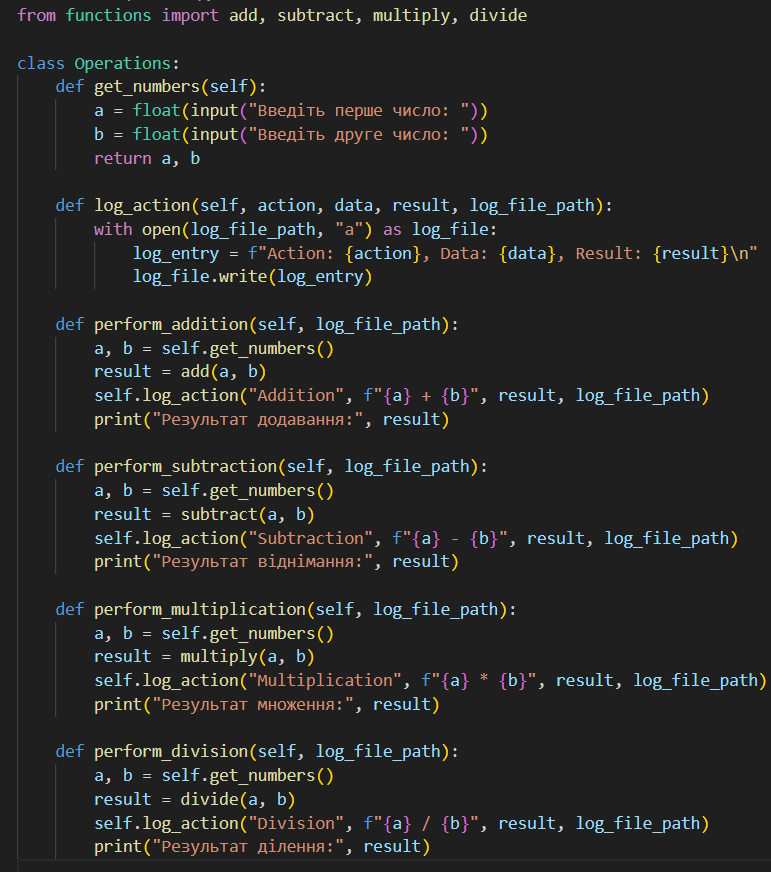
calc.py:



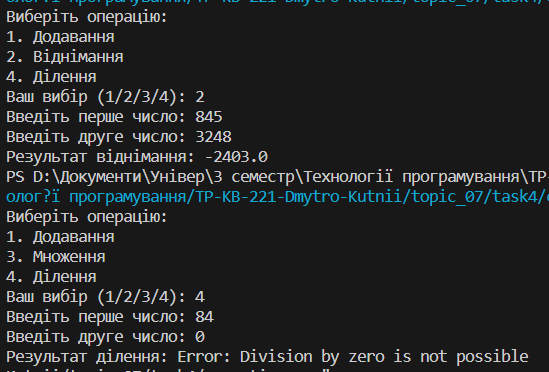
functions.py:

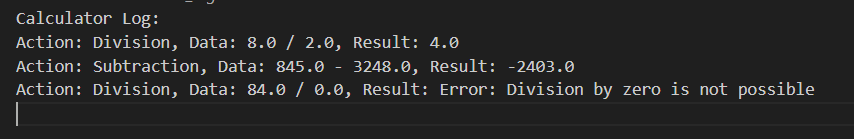


operations.py:



Результат:





**Висновок:** за допомогою даного практичного завдання я навчився використовувати принципи ООП для виконання завдань, а також ознайомився з існуючими за замовченням методами класу по типу \_\_init\_\_(self) \_\_str\_\_(self)\_\_.