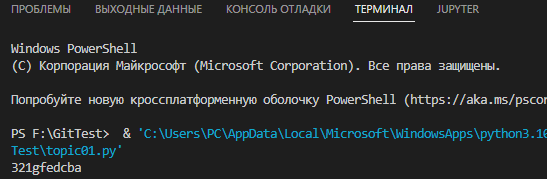
**Тема 1**

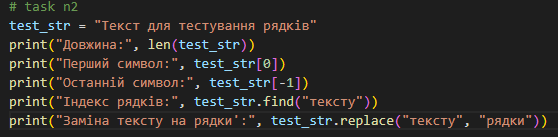
**Звіт про виконання практичних завдань по Технологіях програмування на мові Python**

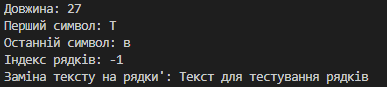
1. **Перетворення рядка**

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

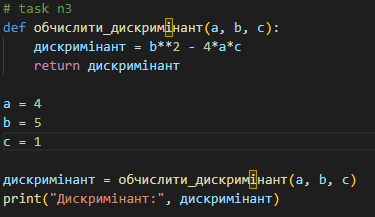
Screenshot_3

**2. Тестування базових функцій для рядків**

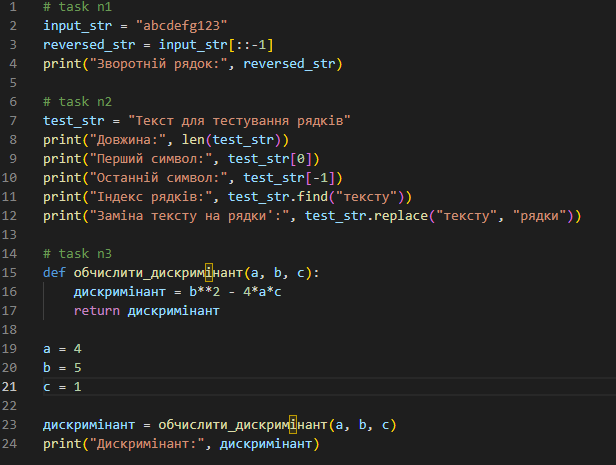


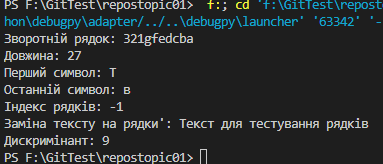


**3. Написання функції пошуку Дискримінанту**



C:\Users\PC\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_7.png

 **Загальний вигляд коду та результат**



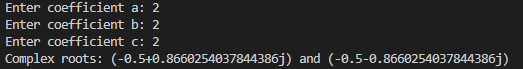
**Висновок: Під час виконання цього практичного завдання з лекції я засвоїв використання вбудованих функцій для обробки рядків у мові програмування Python.**

**Тема 2**

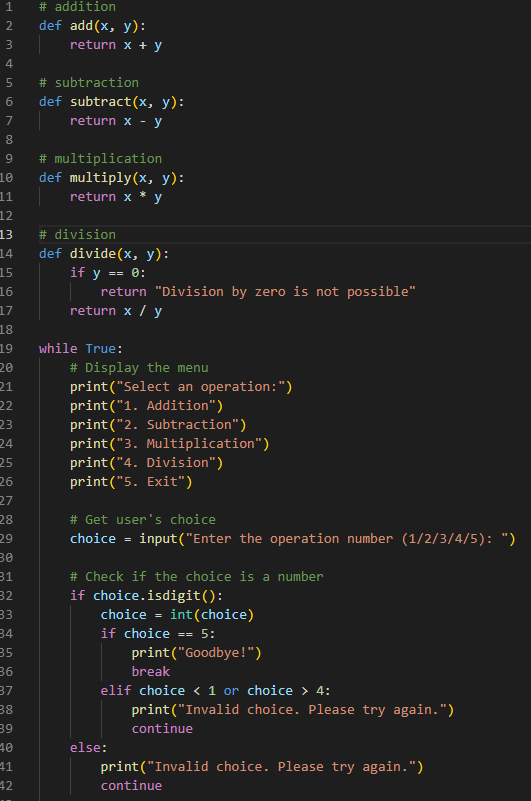
**Звіт про виконання практичних завдань по Технологіях програмування на мові Python**

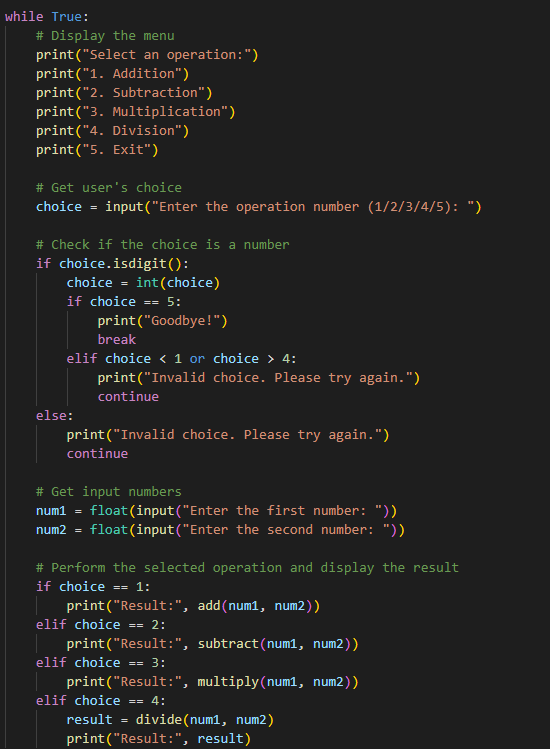
1. **Знайти розв’язки квадратного рівняння за введеними даними.**

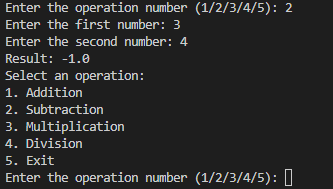
Виконати пошук коренів квадратного рівняння враховуючи значення дискримінанту:

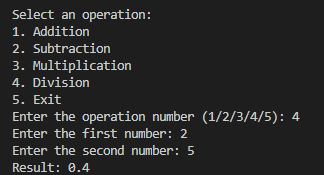


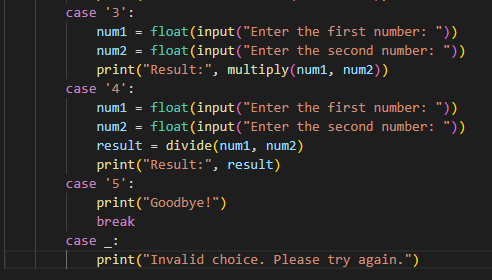
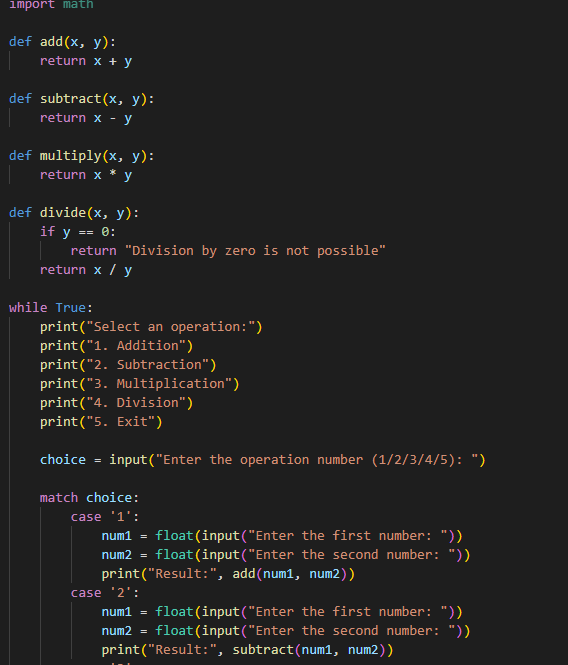
**2. Прoграма калькулятор на основі if elif else**



****



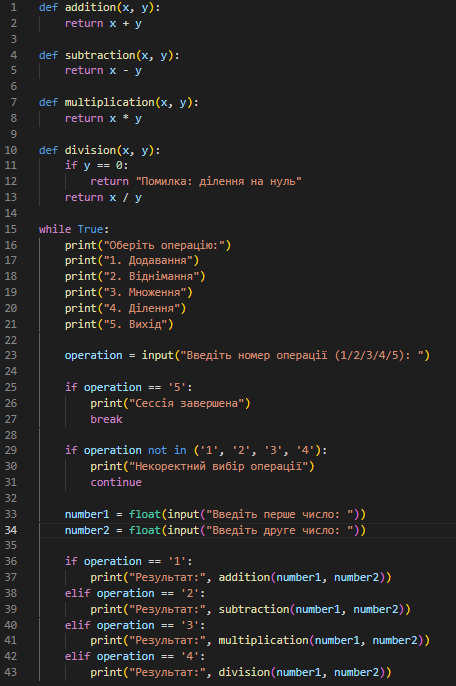
**3. Та сама програма калькулятор на основі оператора match**

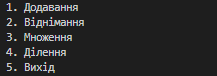


Висновок: У результаті завдання з лекції я отримав досвід у використанні вбудованих функцій для опрацювання рядків з мовою програмування Python.

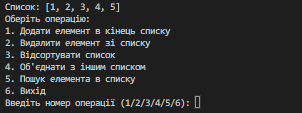
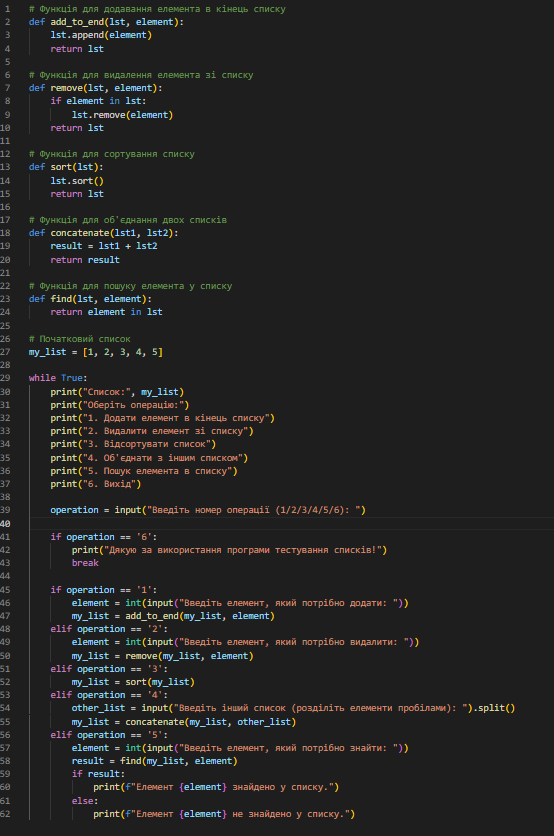
**Тема 3**

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

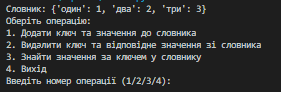
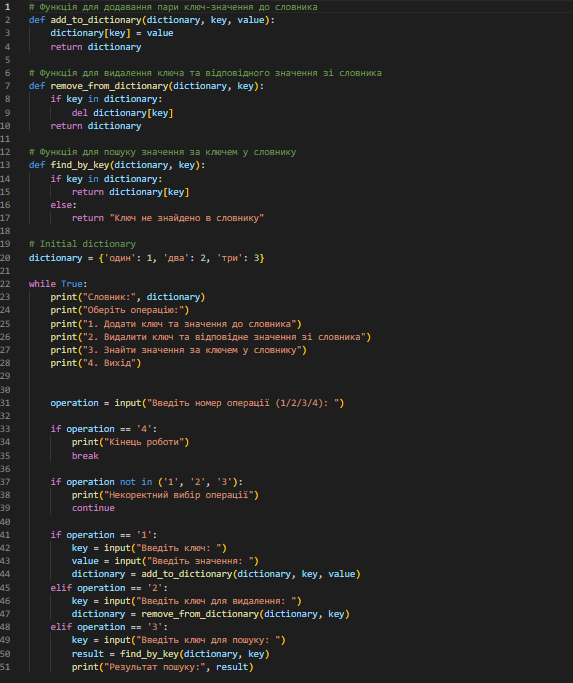




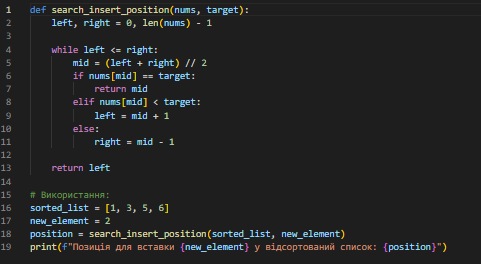
2. Написати програму тестування функцій списків:

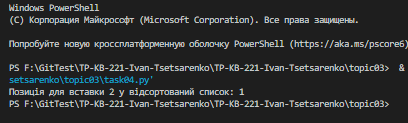


3. Написати програму тестування функцій словників:



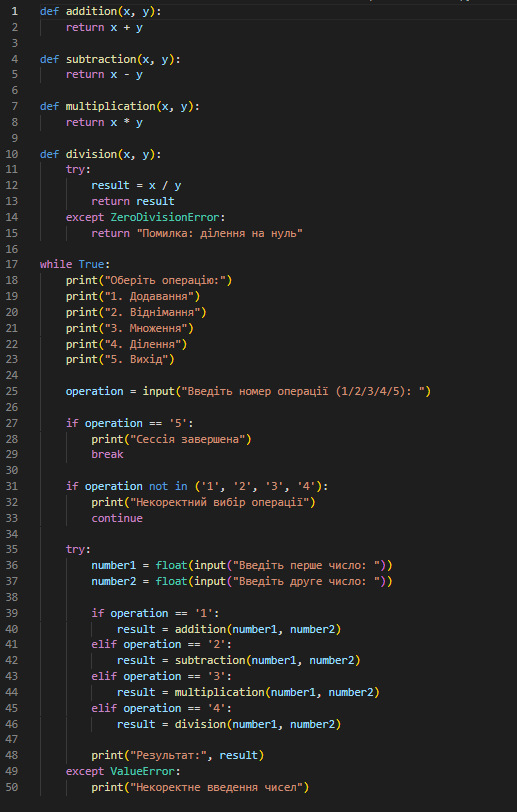
4. Написати функцію пошуку позиції для вставки нового елементу у відсортований список:





**Висновок:** Під час виконання роботи я практикувався зі створенням функцій, використанням умовних операторів, використанням циклів, використанням структур данних та бінарним пошуком. Я отримав досвід використання Python.

**Тема 4**

1. Розширити програму калькулятор функцією запитів від користувача, що обробляє виняткові ситуації.



2. Розширити функцію ділення обробкою виняткової ситуації ділення но нуль



3. Ознайомитись із винятковими ситуаціями

**SyntaxError:** Ця помилка виникає, коли інтерпретатор Python знаходить синтаксичну помилку у вашому коді.

**IndentationError:** Ця помилка виникає, коли відступи в вашому коді не коректно відформатовані.

**NameError:** Виникає, коли ви використовуєте ім'я, яке не було визначено.

**TypeError:** Виникає, коли ви використовуєте змінну або операцію з неправильним типом даних.

**ValueError:** Виникає, коли функція отримує аргумент правильного типу, але з недопустимим значенням.

**ZeroDivisionError:** Виникає, коли ви ділите на нуль.

**FileNotFoundError:** Виникає, коли програма намагається відкрити файл, який не існує.

**KeyError:** Виникає, коли ви намагаєтеся отримати доступ до ключа словника, який не існує.

**IndexError:** Виникає, коли ви намагаєтеся звернутися до елементу списку або кортежу за межами діапазону.

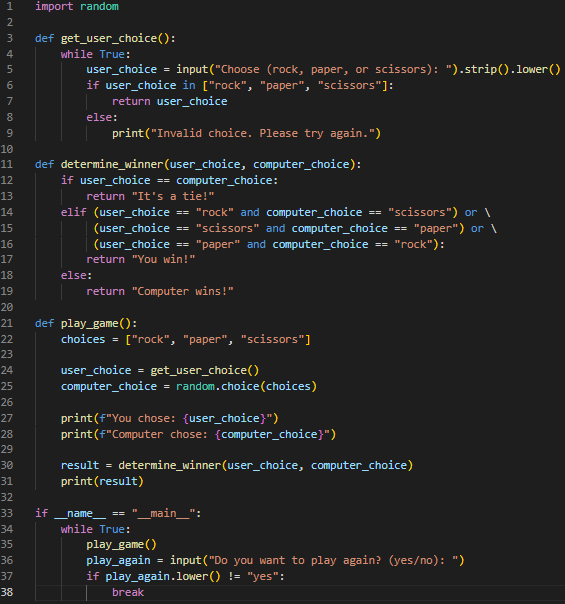
**AttributeError:** Виникає, коли ви намагаєтеся звернутися до атрибута об'єкта, який не існує.

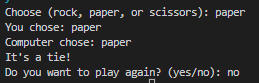
**Висновок:** Під час виконання роботи я навчився розширювати код калькулятора винятковими ситуаціями та ознайомився з усіма винятковими ситуаціями.

**Тема 5**

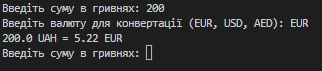
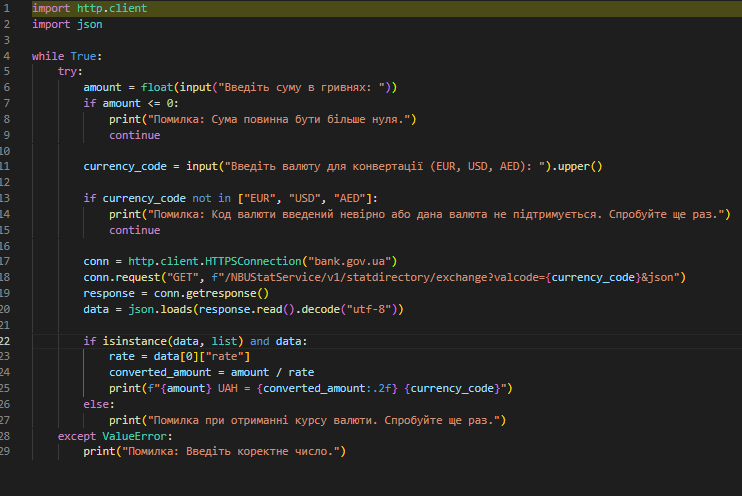
**Бібліотеки**

1. Гра Камінь Ножиці Папер





2. Конвертор валют

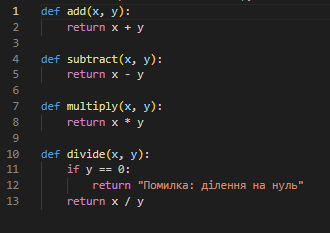


3. Модулі для програми калькулятор

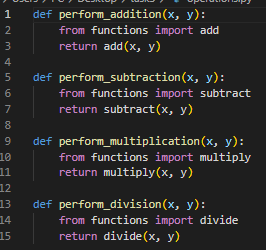
1. calc.py

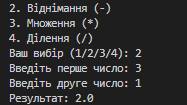


2. functions.py



3. operations.py



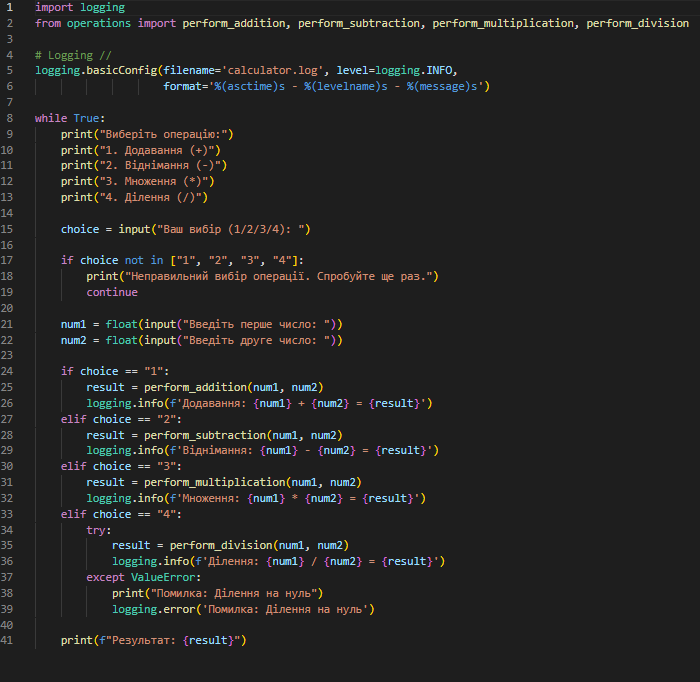


**Висновок:** Я навчився розділяти програму-калькулятор на частини та створив міні-гру камень, ножиці та папер а також написав програму для конвертації валют.

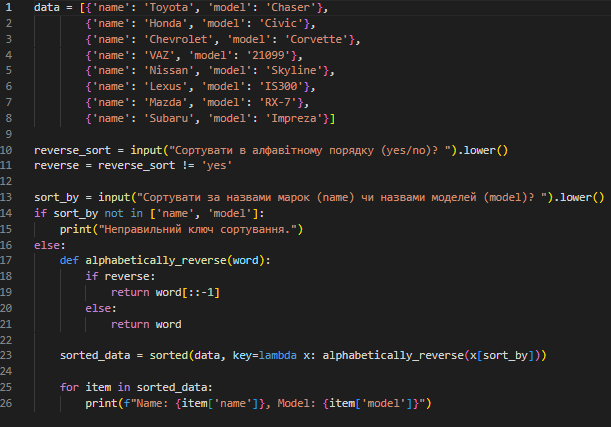
**Тема 6**

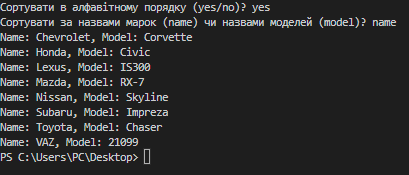
**Робота з файлами**

1. Логування всіх дій в застосунку Калькулятор



2. Використання lambda функцій для функції сортування





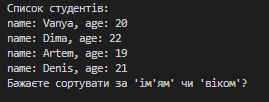
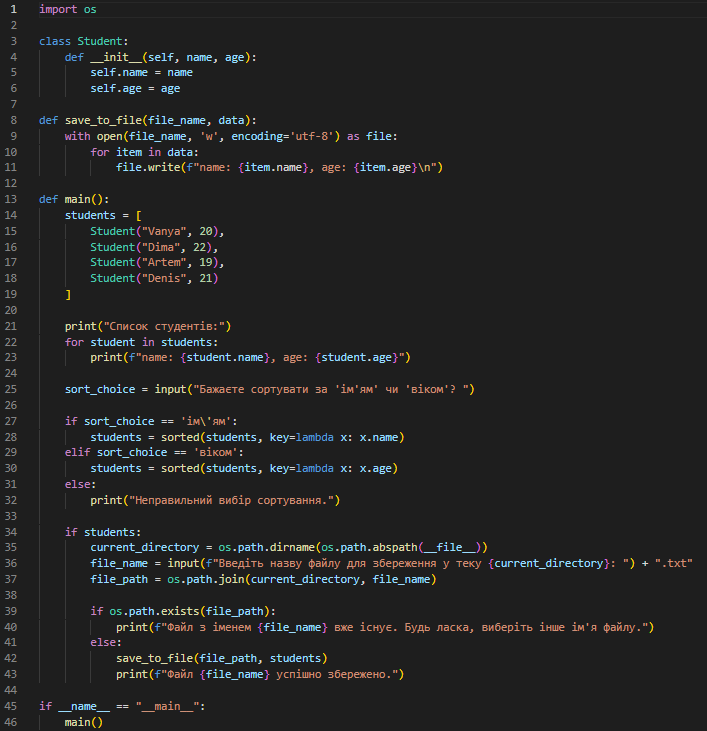
Висновок: Я попрацював у середовищі Python та навчився використовувати lambda для функцій сортування та створювати логування дій.

**Тема 7**

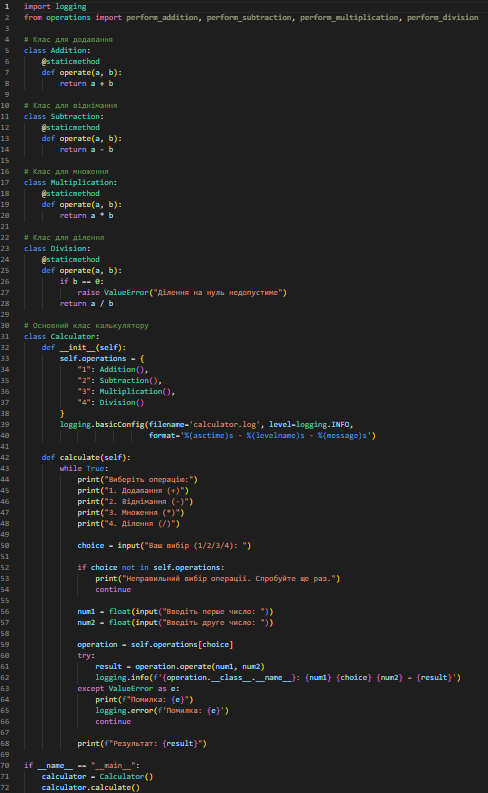
**ООП**

1. Ознайомитисьз існуючими методами коасу та надати приклади використання.

2. Створити клас Student з атрибутами name та age. Створити список об'єктів класу Student і вивести їх на екран у відсортованому порядку за ім'ям, використовуючи функцію sorted з lambda-функцією для визначення ключа сортування.



3. Використовуючи принципи ООП переписати калькулятор.



Висновки:

* Класи дозволяють структурувати дані та функції в об'єкти зі спільними характеристиками.
* У Python можна використовувати функцію sorted для сортування списків об'єктів за заданим ключем.
* lambda-функція дозволяє створювати короткі функції для визначення ключа сортування.
* Ми розглянули принципи ООП та модульний підхід при створенні класу та використанні його для обробки даних.
* Важливо розуміти, як працює функція sorted та як використовувати lambda для налаштування сортування.