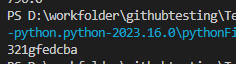
**Звіт про виконання практичних завдань до лекцій з курсу Технології програмування на мові Python**

Звіт 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

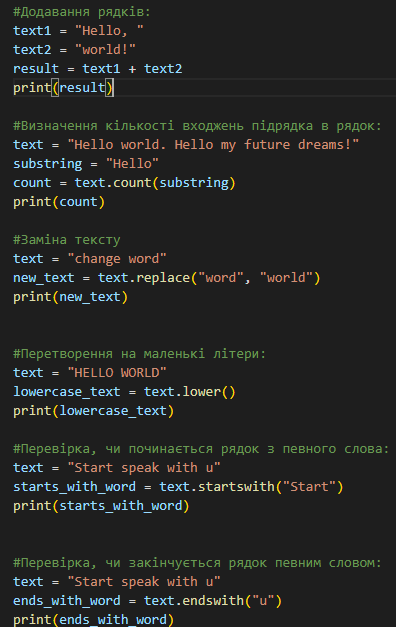
**capitalize -** Переведення першої літери у велику

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

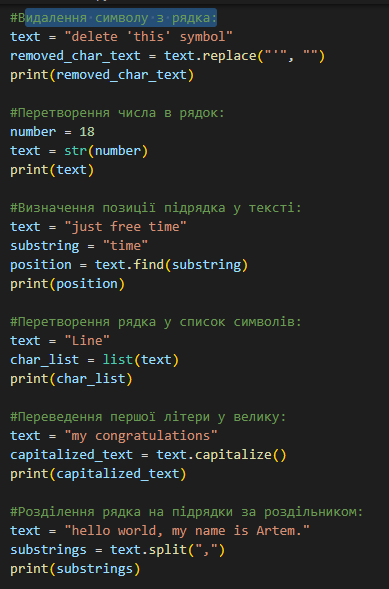
**isdigit -** Перевірка, чи складається рядок лише з цифр

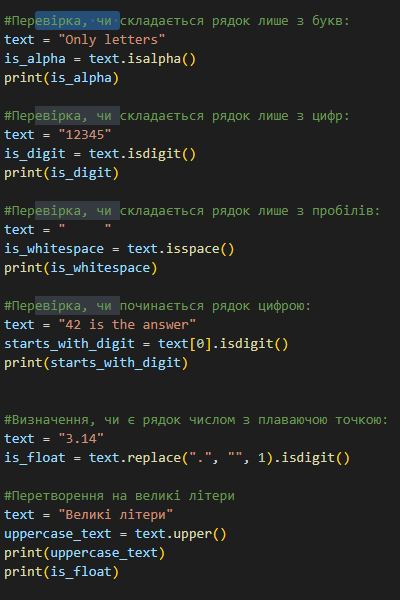
**isspace -** Перевірка, чи складається рядок лише з пробілів

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

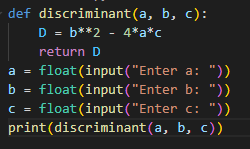
****

****

****

****

**3. Написання функції пошуку Дискримінанту**

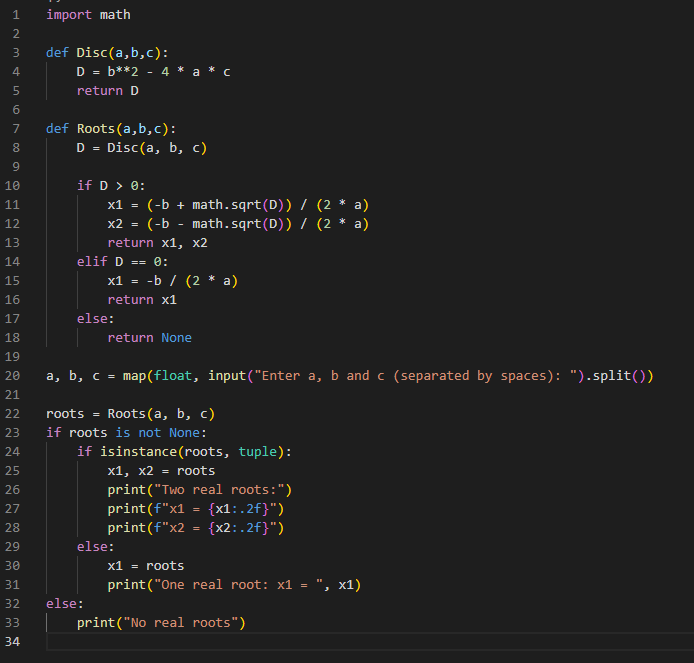
****

**Висновок:** під час виконання цього практичного завдання я навчився використовувати функції для рядків та створив міні програму для вирахування дискримінанту на мові python.

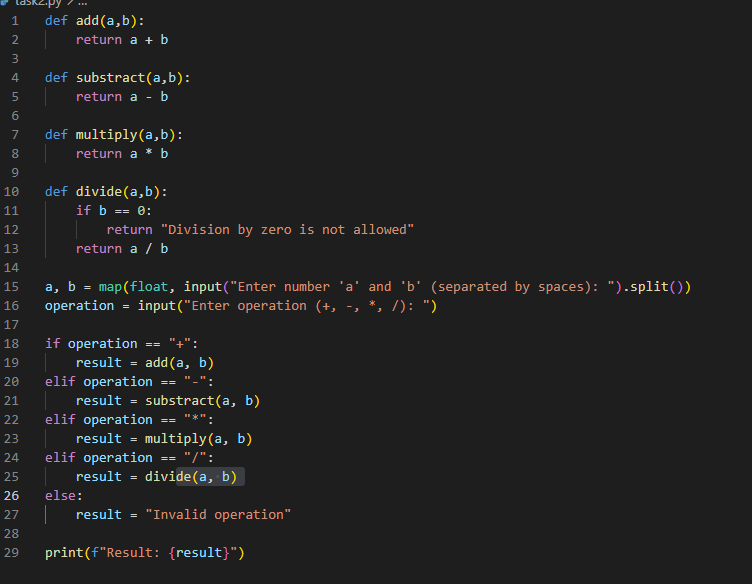
**Звіт про виконання практичних завдань до лекцій з курсу Технології програмування на мові Python**

Звіт до теми №2

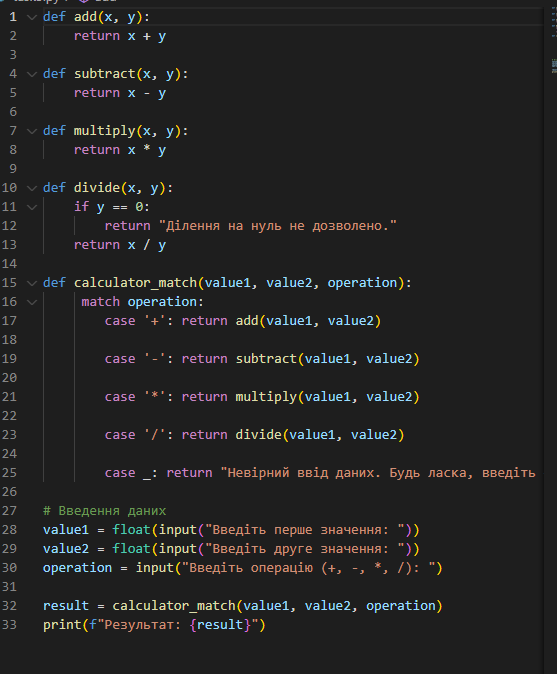
**Завдання 1:** Написати функцію пошуку коренів квадратного рівняння використовуючи функцію розрахунку дискримінанту з попередньої теми та умовні переходи.



**Завдання 2:** Написати програму калькулятор використовуючи **if else** конструкцію.



**Завдання 3:** Написати програму калькулятор використовуючи **match** конструкцію

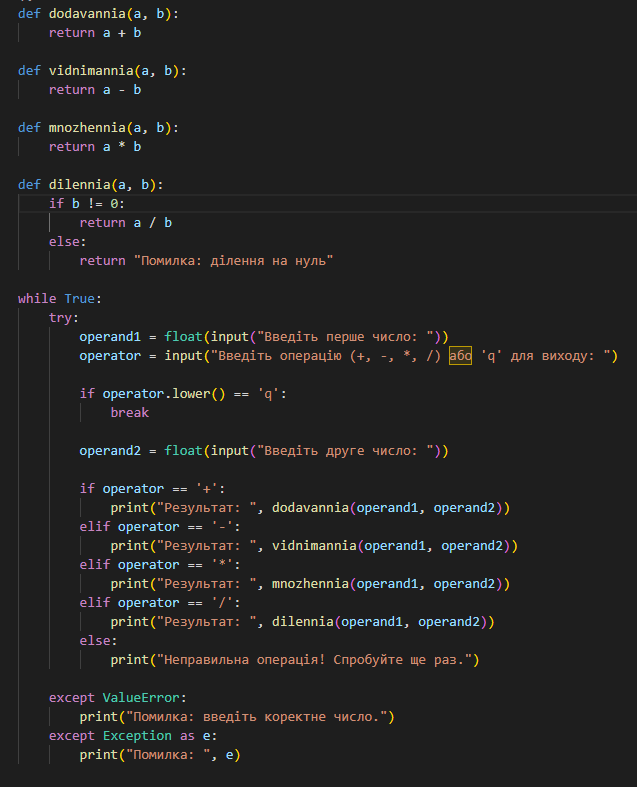


Висновок: на практичному завданні я розібрав та створив програму для пошуку квадратних коренів та калькулятор , та вивчив конструкцію **match.**

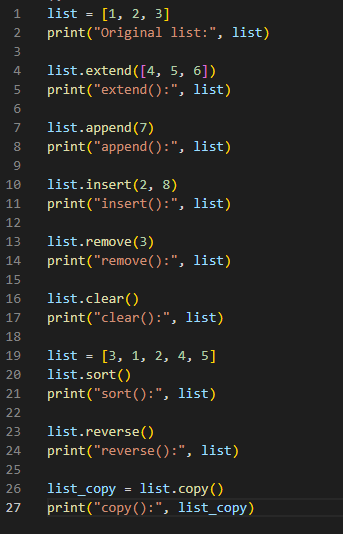
**Звіт про виконання практичних завдань до лекцій з курсу Технології програмування на мові Python**

Звіт до теми №3

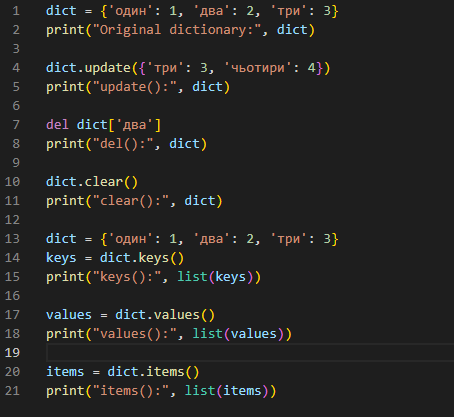
1. Написати програму калькулятор з постійними запитами на введення нових даних та операцій:



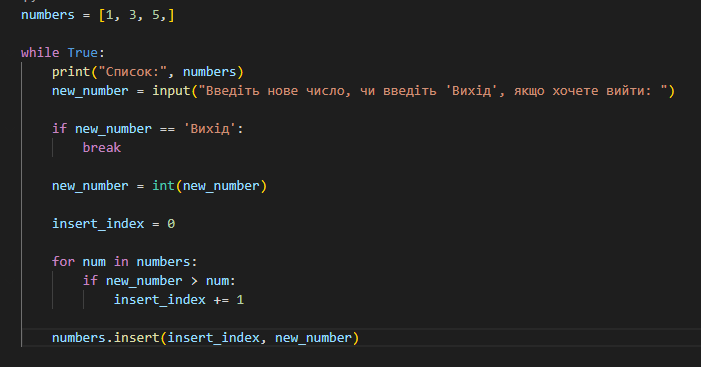
2. Написати програму тестування функцій списків:



3. Написати програму тестування функцій словників:



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

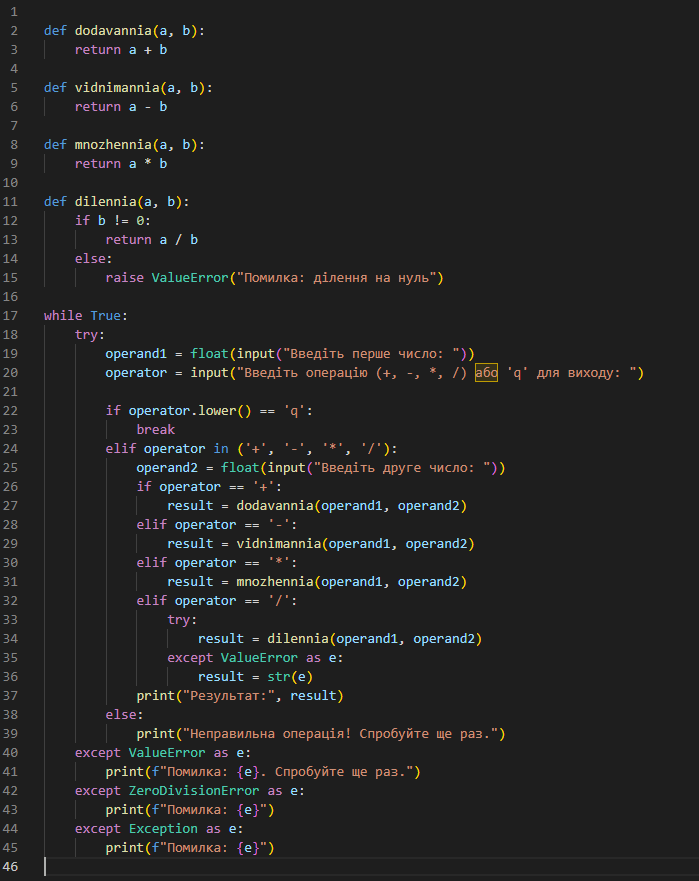


**Висновок:** Під час виконання 3 лабораторної роботи я навчився створювати : калькулятор з постійними запитами на введення нових даних та операцій, програму дял тестування функцій списків , програму длятестування функцій словників та написав функцію для пошуку позиції для вставки нового елементу у відсортований список.

**Звіт про виконання практичних завдань до лекцій з курсу Технології програмування на мові Python**

Звіт до теми №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.

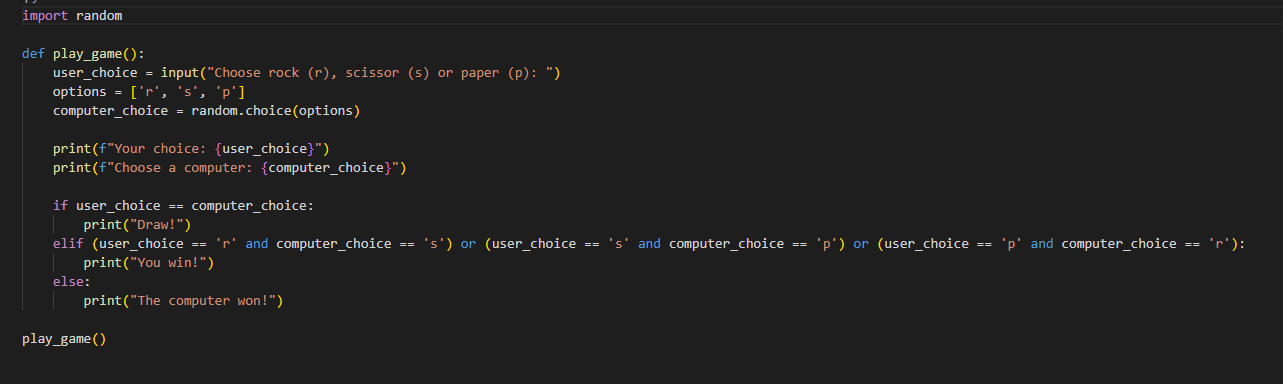
**AttributeError:** Виникає, коли ви намагаєтеся звернутися до атрибута об'єкта, який не існує.

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

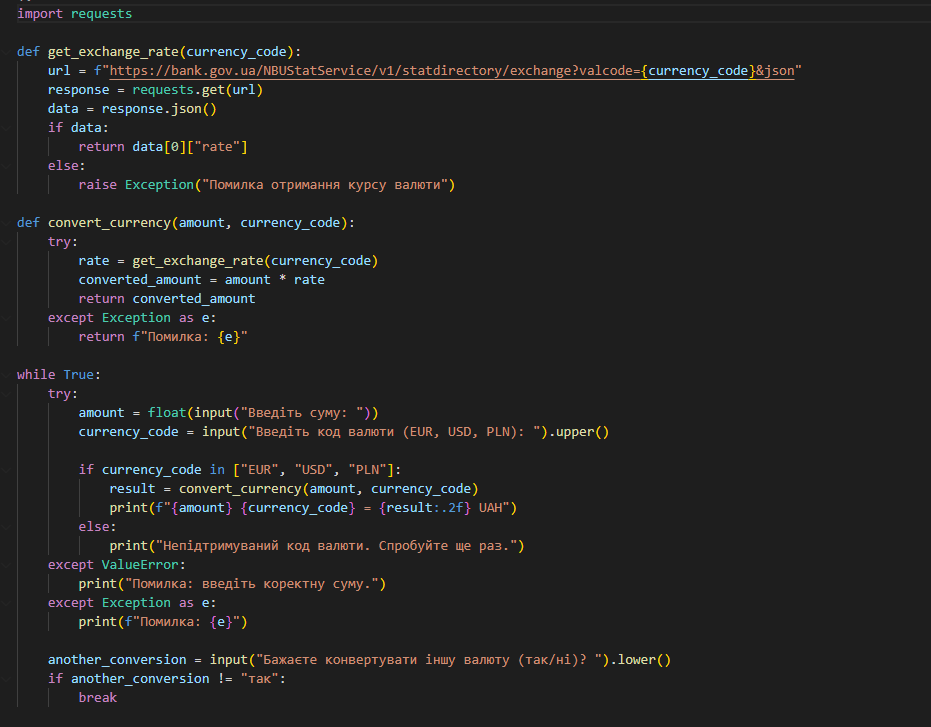
**Звіт про виконання практичних завдань до лекцій з курсу Технології програмування на мові Python**

Звіт до теми №5

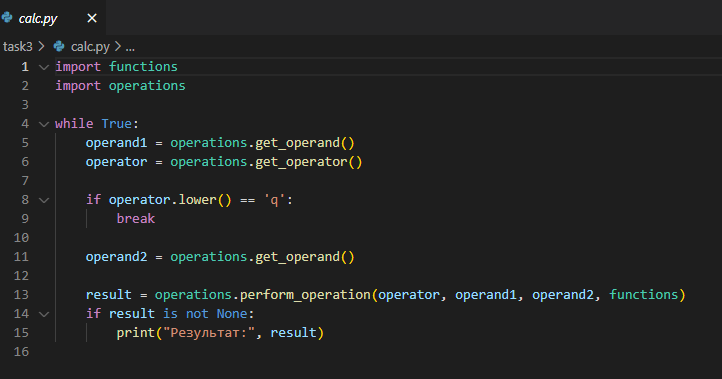
1. Гра з комп’ютером: камінь, ножиці, папір. Програма виконує запит від користувача на введення одного із значень ["rock", "scissor", "paper"]. Наступним кроком, використовуючи модуль random, програма у випадковому порядку вибирає одне із значень ["stone", "scissor", "paper"]. В залежності від умови, що камінь перемагає ножиці, ножиці перемагають папір, а папір перемагає камінь визначити переможця.

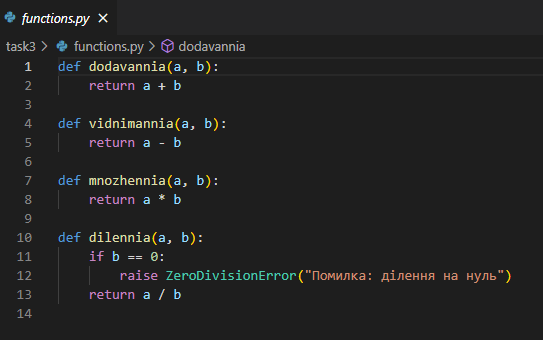


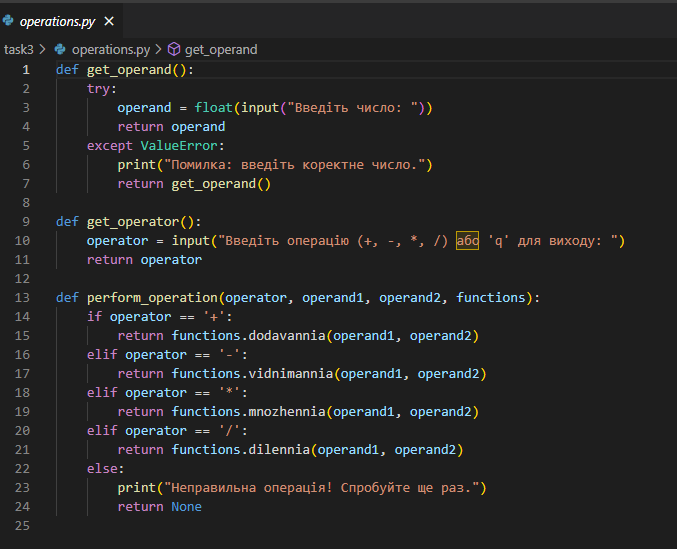
1. Програма конвертування іноземної валюти в українську гривню. Для отримання актуальних курсів валют необхідно використовувати API НБУ та модуль, що надає можливість виконувати запити до сторонніх сервісів requests. Достатня умова роботи – можливість конвертації для трьох іноземних валют EUR, USD, PLN. Користувачу надається можливість введення кількості та типу валюти, результат роботи програми – конвертоване значення в українських гривнях.



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.





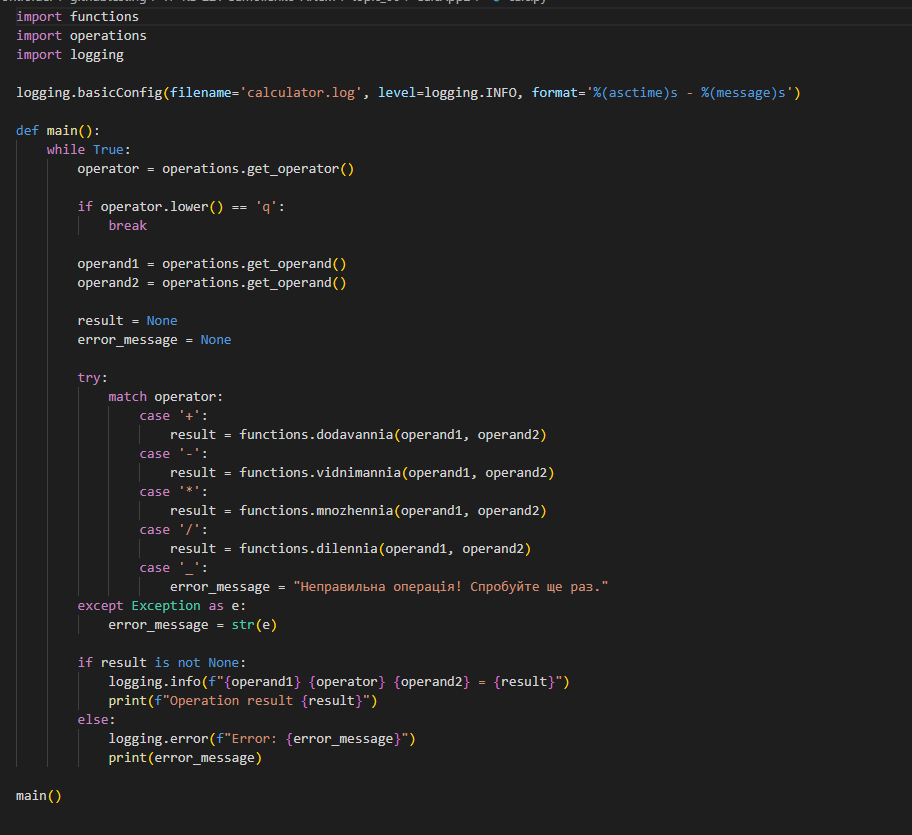


**Висновок:** Під час виконання роботи я навчився використовувати бібліотеки такі як random, requests та також начився використовувати модулі для програми калькулятор .

**Звіт про виконання практичних завдань до лекцій з курсу Технології програмування на мові Python**

Звіт 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.



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**: Отже, виконавши практичне завдання до теми 5, я вдосконалив власні навички використання функції sorted() та ознайомився з lambda.

**Звіт про виконання практичних завдань до лекцій з курсу Технології програмування на мові Python**

Звіт до теми №7

**Завдання 1:** Ознайомитись з існуючими за замовченням методами класу по типу \_\_init\_\_(self) \_\_str\_\_(self)\_\_ та надати приклади використання.

У даному завданні потрібно було розібратись з декількома методами класу. Я розібрав три. Це init, str, call.

Ви згадали методи `\_\_init\_\_`, `\_\_str\_\_`, `\_\_call\_\_`, і я розповім вам, як вони працюють та як їх використовувати.

1. `\_\_init\_\_`:

- Цей метод є конструктором класу і викликається при створенні нового об'єкта цього класу.

- Він приймає обов'язковий аргумент `self`, який посилається на сам об'єкт.

- Зазвичай `\_\_init\_\_` використовується для ініціалізації атрибутів об'єкта.

- Приклад:

class Person:

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

self.name = name

self.age = age

person1 = Person("Alice", 25)

print(person1.name) # Виведе "Alice"

print(person1.age) # Виведе 25

2. `\_\_str\_\_`:

- Цей метод повертає рядок, який представляє об'єкт у зручному для читання форматі.

- Він викликається, коли ви використовуєте функцію `str()` або використовуєте об'єкт у рядковому контексті (наприклад, при виводі).

- Приклад:

class Person:

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

self.name = name

self.age = age

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

return f"Person: {self.name}, Age: {self.age}"

person1 = Person("Alice", 25)

print(str(person1)) # Виведе "Person: Alice, Age: 25"

3. `\_\_call\_\_`:

- Цей метод дозволяє об'єкту бути викликаним, як функція.

- Він викликається, коли ви викликаєте об'єкт з дужками.

- Приклад:

class Calculator:

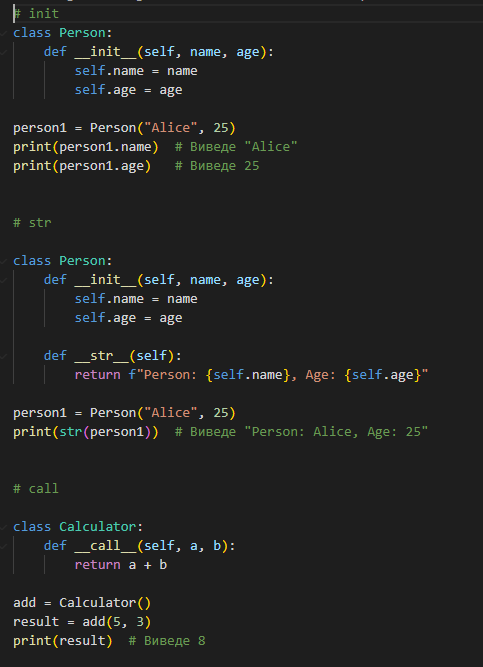
def \_\_call\_\_(self, a, b):

return a + b

add = Calculator()

result = add(5, 3)

print(result) # Виведе 8



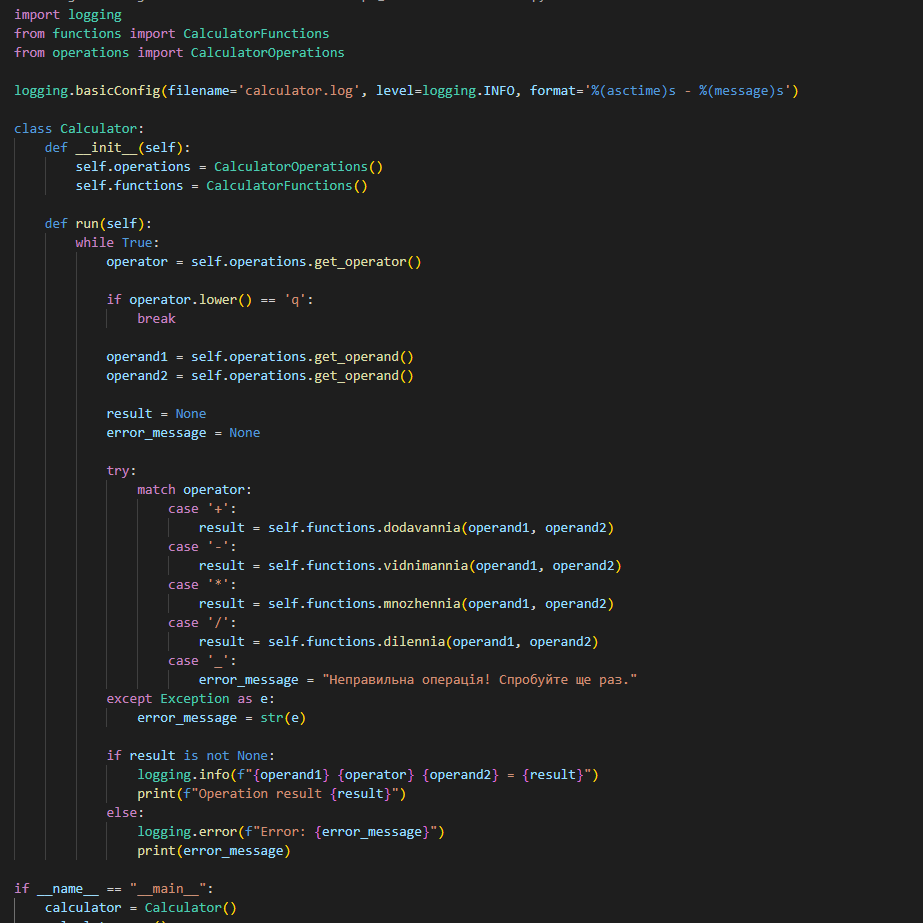
**Завдання 2:** Розробити клас **Student** атрибутами якого э два параметра **name** та **age**. Створити список елементами якого є об'єкти класу **Student**. Написати цикл який виводить на екран елементи списку у відсортованому порядку. Для сортування використати стандартну функцію **sorted**. Функція **sorted** має використовувати **lambda** функцію для визначення ключа сортування.



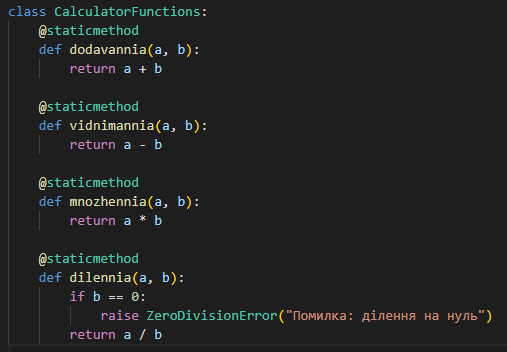
**Завдання 3:**

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

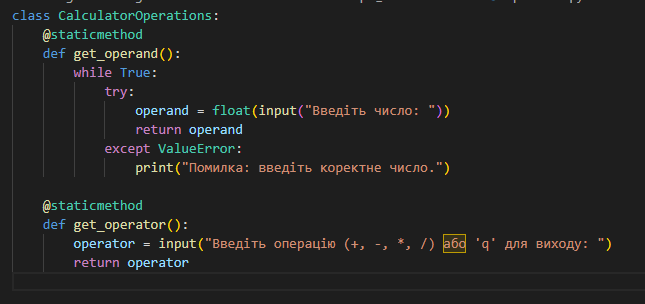
calc.py :



functions.py:



operations.py:



Висновок : Ознайомився з існуючими за замовченням методами класу по типу \_\_init\_\_(self) \_\_str\_\_(self)\_\_ та надав приклади використання. Модернізував програму калькулятор використовуючи ООП.