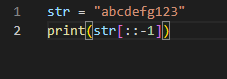
**Звіт про виконання практичних завдань до лекції з курсу Технології програмування на мові Python**

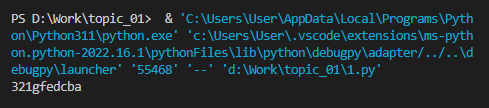
Під час виконання практичного завдання до Теми №1 я надав наступні варіанти рішень до задач:

**1. Перетворення рядка**

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



**Результат виконання:**



**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

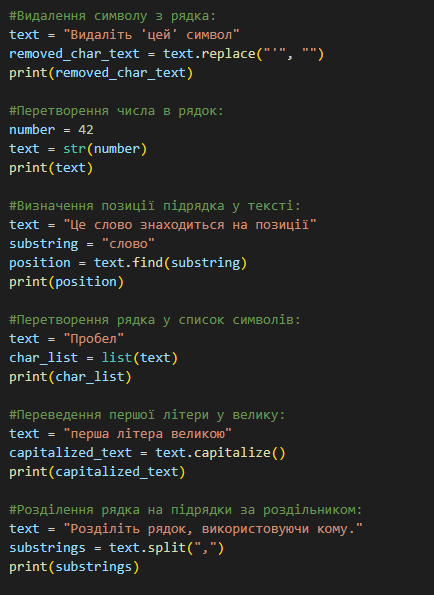
**isdigit -** Перевірка, чи складається рядок лише з цифр

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

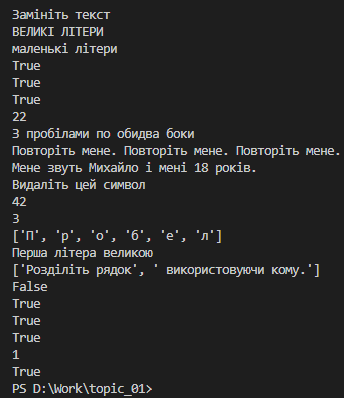
**count -** Визначення кількості входжень підрядка в рядок







**Результат роботи:**



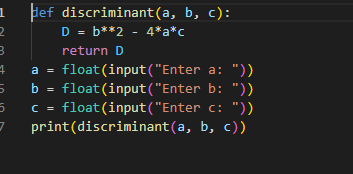
**3. Написання функції пошуку Дискримінанту**

def discriminant(a, b, c):

D = b\*\*2 - 4\*a\*c

return D

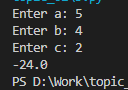
a = float(input("Enter a: "))

b = float(input("Entr b: "))

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

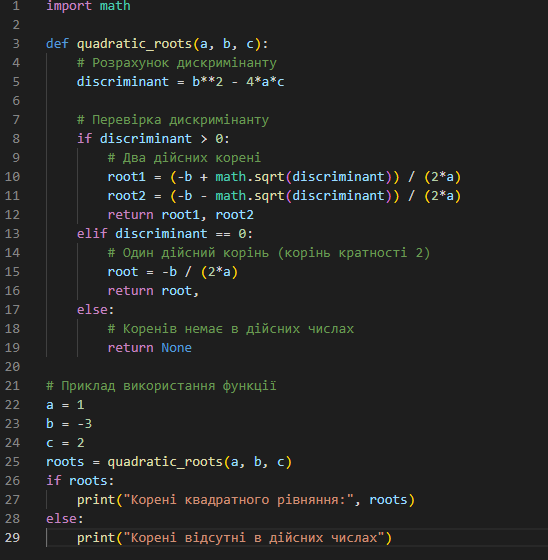
print(discriminant(a, b, c))

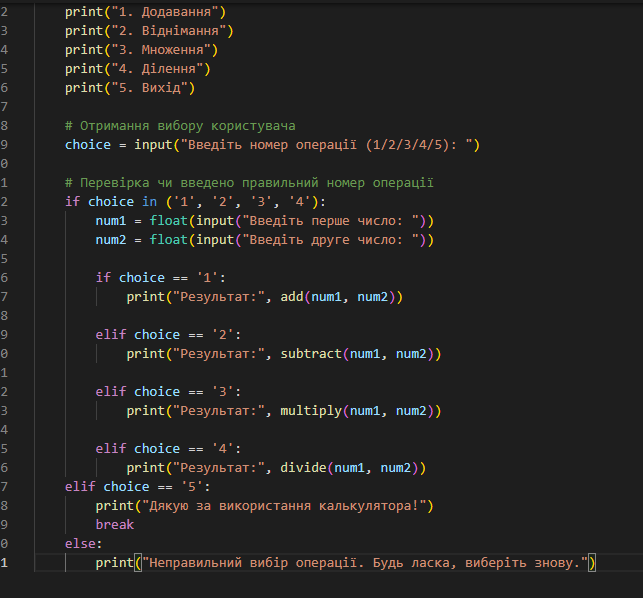
**Приклад роботи:**

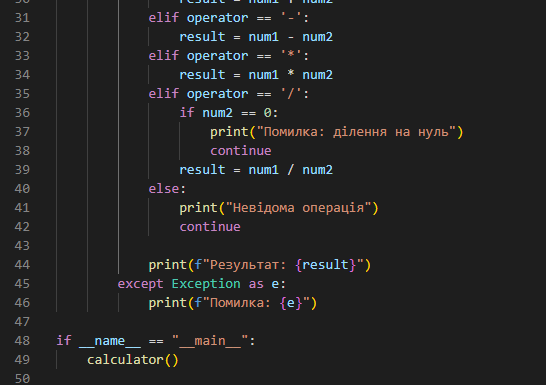


**Висновок:** Отже, завершивши лабораторну роботу з теми 1, я здобув досвід використання вбудованих функцій для обробки рядків у мові програмування Python.  
  
 **Звіт про виконання практичних завдань до лекцій з курсу Технології програмування на мові Python**

Звіт до теми №2

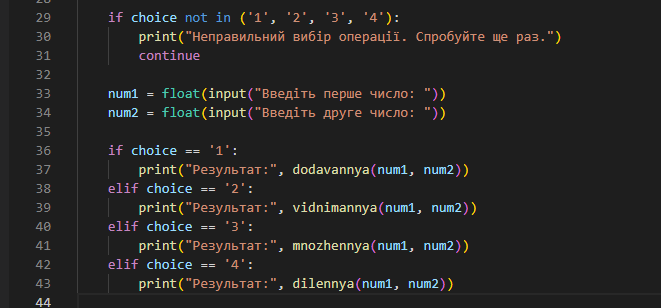
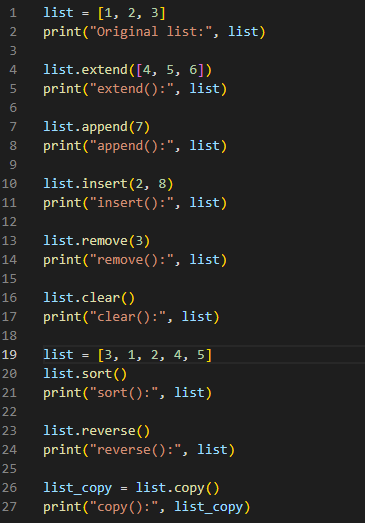
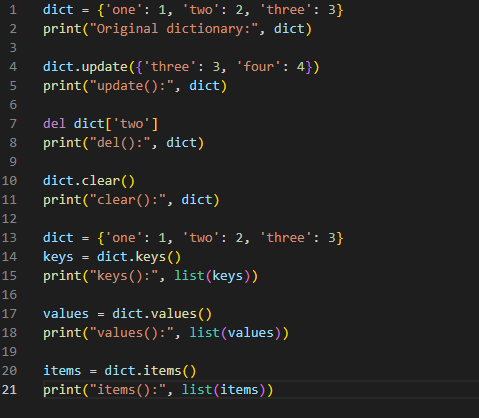
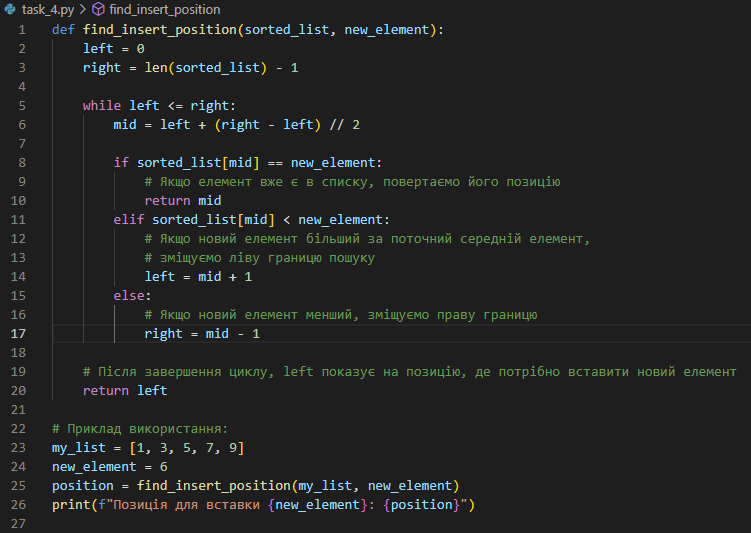
**Завдання 1:** Написати функцію пошуку коренів квадратного рівняння використовуючи функцію розрахунку дискримінанту з попередньої теми та умовні переходи.  
  
  
**Завдання 2:** Написати програму калькулятор використовуючи **if else** конструкцію.

   
  
  
  
  
  
**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**

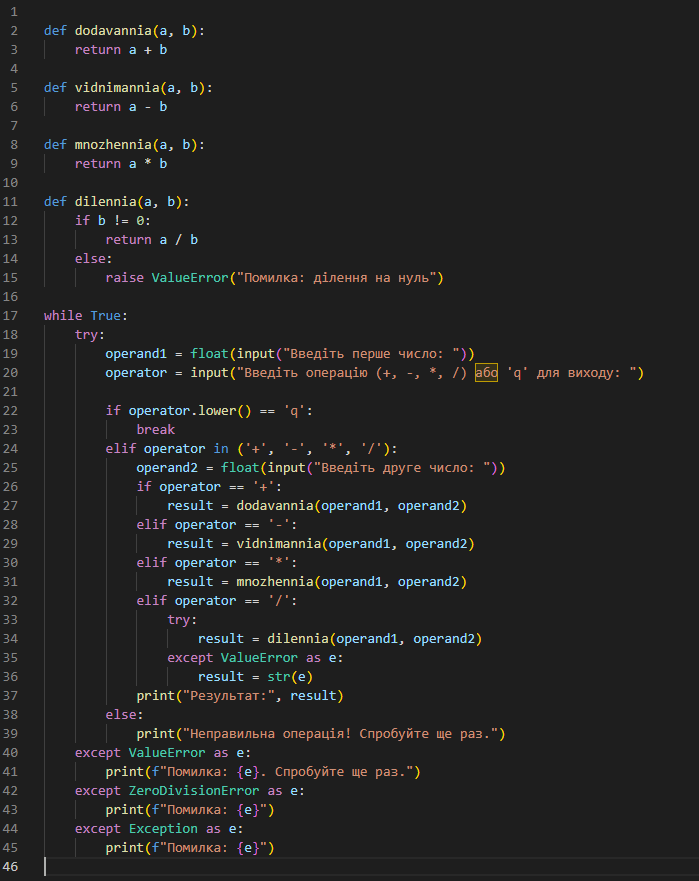
Звіт до теми №3  
Завдання №1«Написати програму калькулятор з постійними запитами на введення нових даних та операцій.»

  
  
  
  
Завдання №2 «Написати програму тестування функцій списків»  
  
  
Завдання №3 «Написати програму тестування функцій словників»  
  
  
  
Завдання №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.

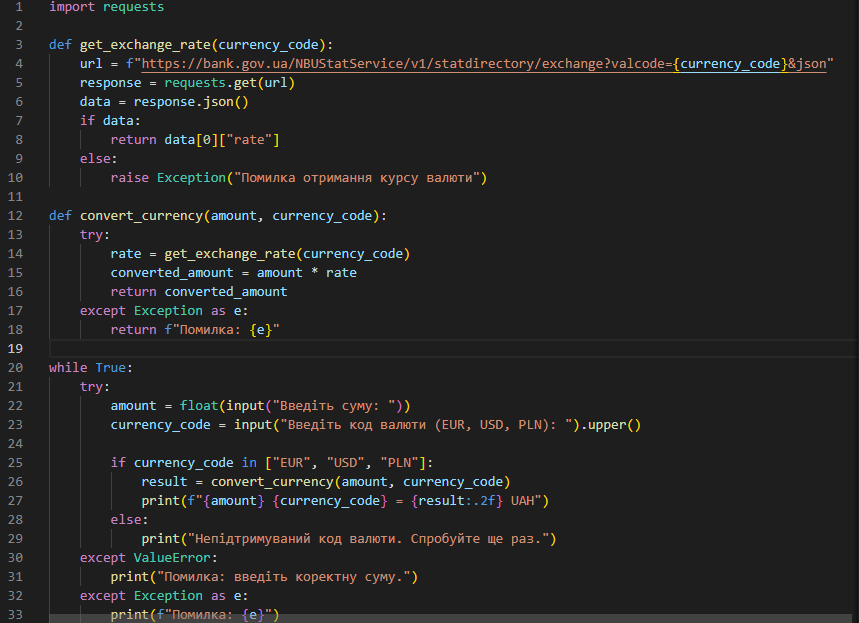
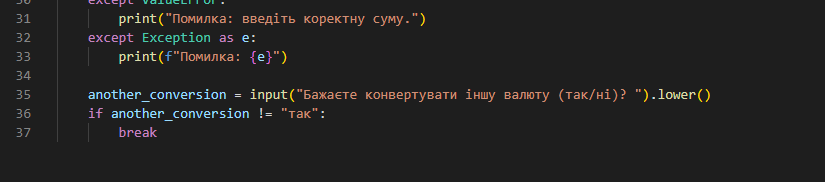
**IndexError:** Виникає, коли ви намагаєтеся звернутися до елементу списку або кортежу за межами діапазону.

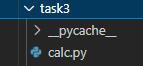
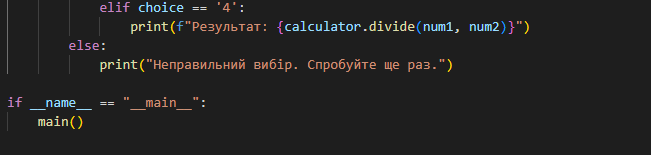
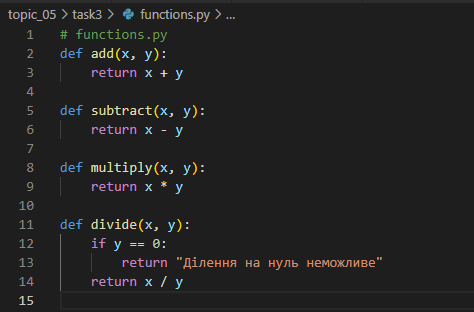
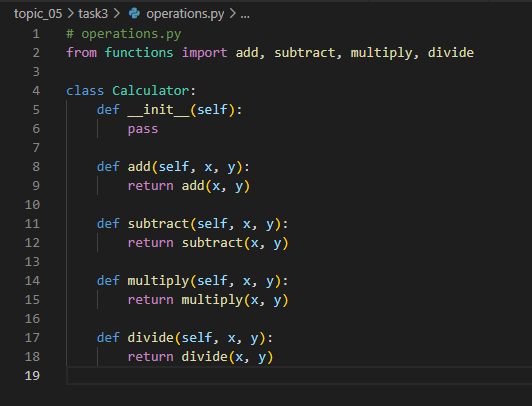
**AttributeError:** Виникає, коли ви намагаєтеся звернутися до атрибута об'єкта, який не існує.

**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**

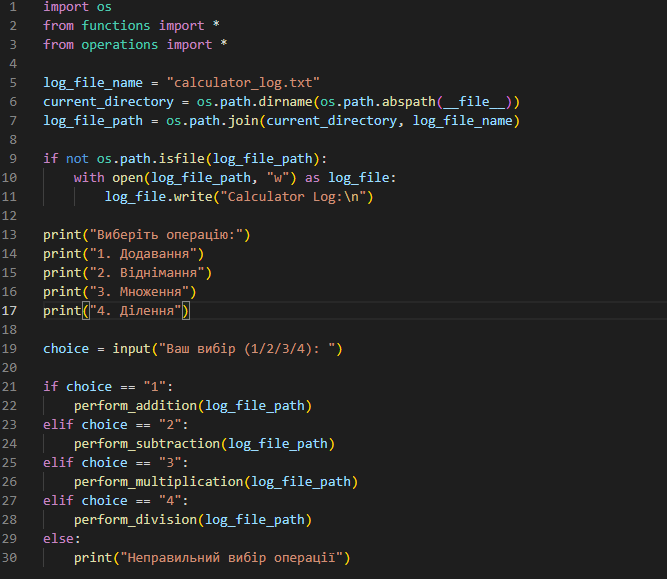
Звіт до теми №5

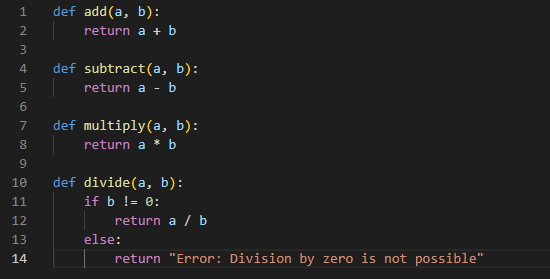
1. Гра з комп’ютером: камінь, ножиці, папір. Програма виконує запит від користувача на введення одного із значень ["камінь", "ножиці", "папер"]. Наступним кроком, використовуючи модуль random, програма у випадковому порядку вибирає одне із значень ["к", "н", "п"]. В залежності від умови, що камінь перемагає ножиці, ножиці перемагають папір, а папір перемагає камінь визначити переможця.
2.   
     
     
   2.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.  
     
     
     
   3.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.

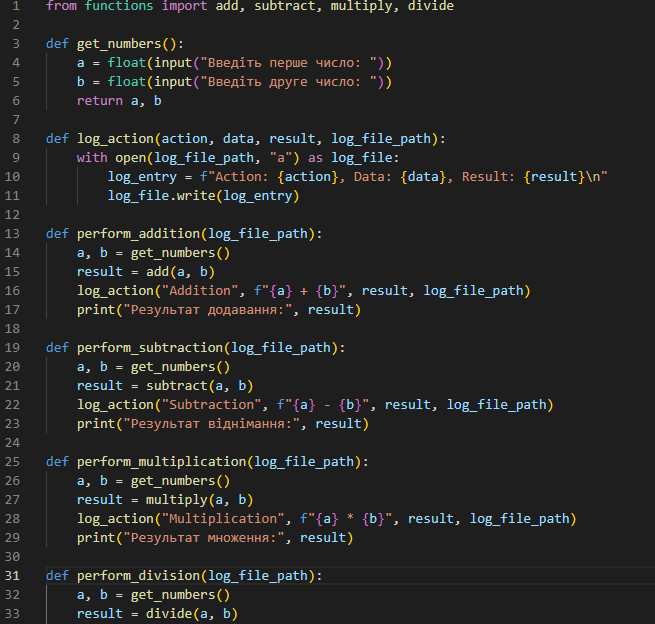
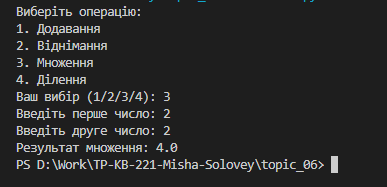
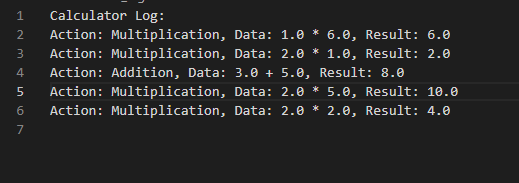
  
  
  
  
  
  
  
  
**Висновок:** Під час виконання роботи я навчився використовувати бібліотеки такі як random, requests та також начився використовувати модулі для програми калькулятор .

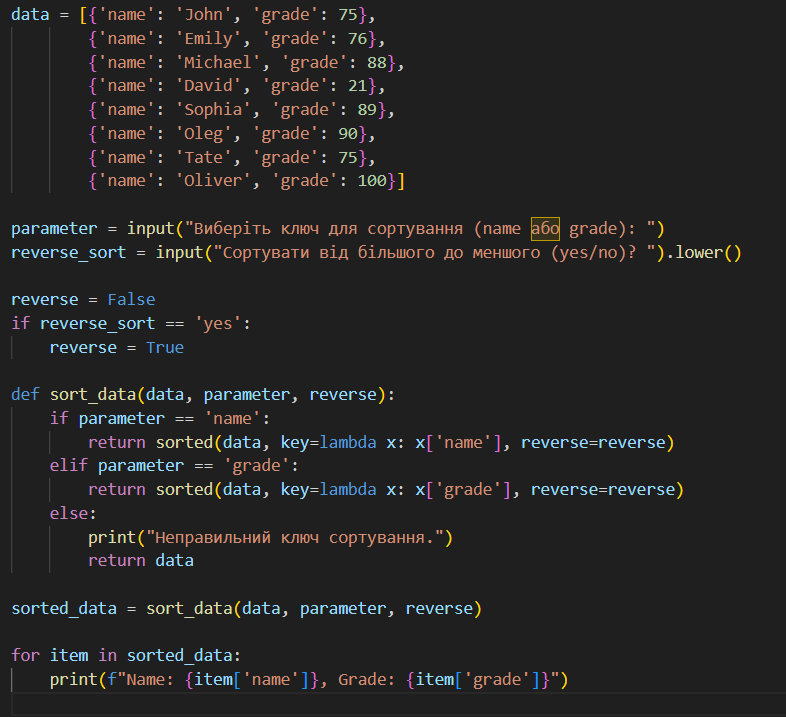
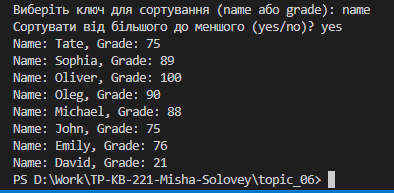
**Практичне завдання до теми 6**

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

1.   
     
     
   Результат роботи  
      
     
     
   Task2   
   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.

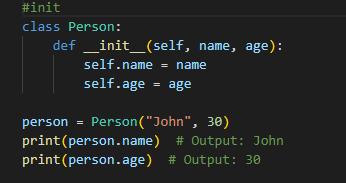
  
  
  
Висновок: В процесі виконання цього конкретного завдання я освоїв техніку реєстрації змін за допомогою лог-файлу і отримав розуміння суті лямбда-функцій, здобувши навички їх використання.  
  
**Звіт про виконання практичних завдань до лекцій з курсу Технології програмування на мові Python**

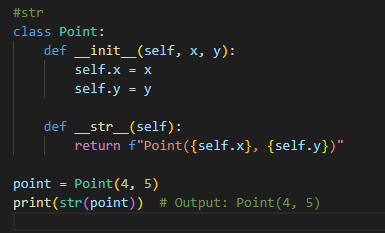
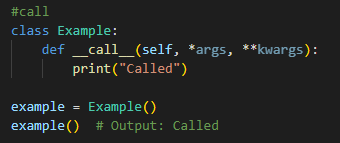
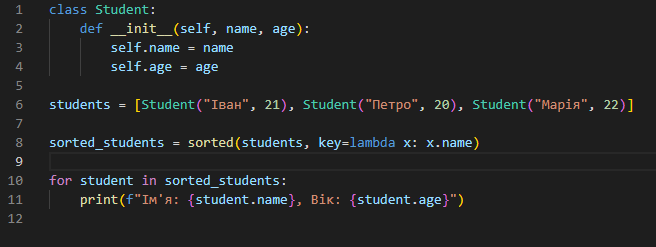
Звіт до теми №7

**Завдання 1:** Ознайомитись з існуючими за замовченням методами класу по типу init,str,call та надати приклади використання.

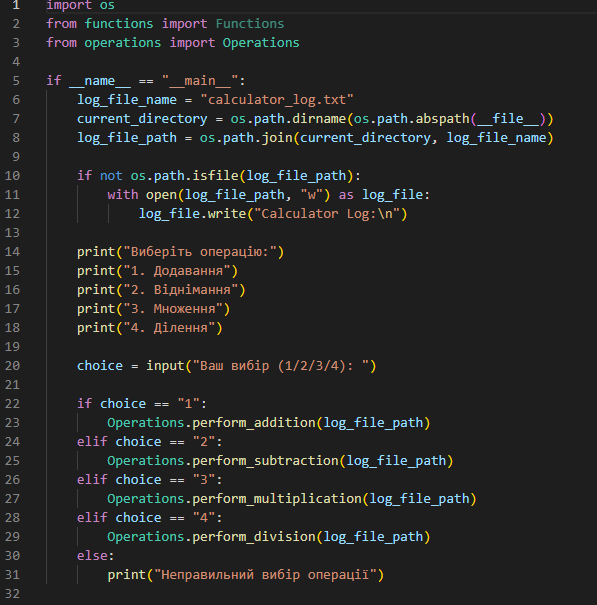
У даному завданні потрібно було розібратись з декількома методами класу. Я розібрав три. Це init, str, call.

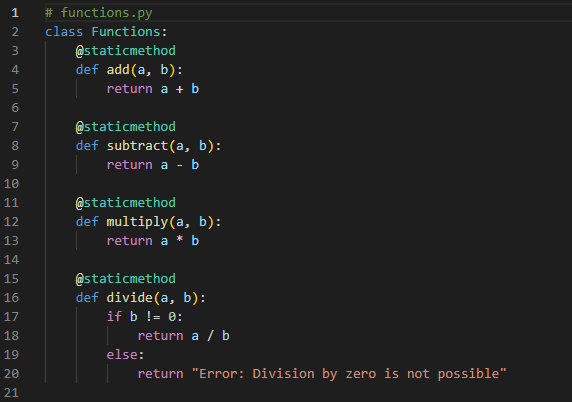
Ви згадали методи `init`, `str`, `call`, і я розповім вам, як вони працюють та як їх використовувати.

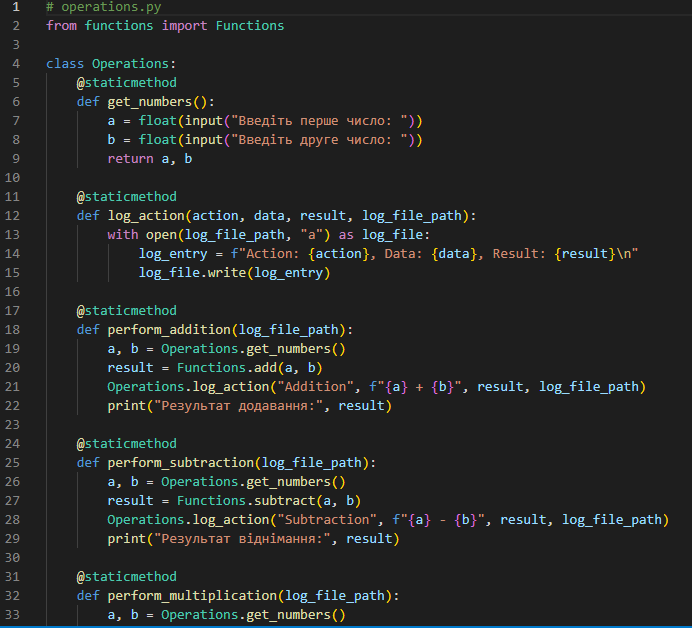
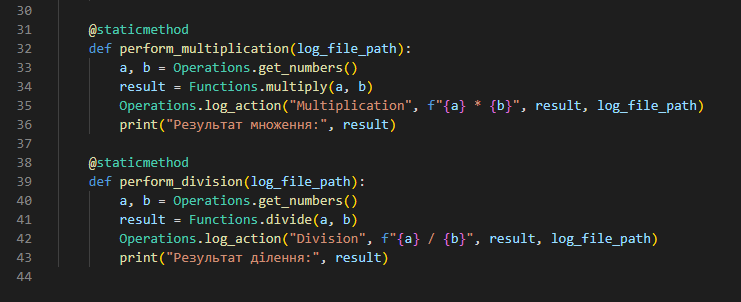
1.Class “init”  
 - Цей метод є конструктором класу і викликається при створенні нового об'єкта цього класу.  


2.Class “str”  
- Цей метод повертає рядок, який представляє об'єкт у зручному для читання форматі.  
  
  
  
  
  
  
  
  
  
  
  
3.Class “Call”  
-Завдяки цьому методу об'єкт можна викликати, якщо викликати його з дужками.  
-Він викликається у разі, коли ви викликаєте об'єкт, використовуючи дужки.  
  
  
**Завдання 2**: Створити клас "Студент", що містить два атрибути: " **name** " та " **age** ". Створити список, елементи якого є об'єктами класу "Студент". Написати цикл, який виводить на екран елементи списку у відсортованому порядку. Для сортування використати стандартну функцію "sorted". Під час використання функції "sorted" додати лямбда-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 зі стандартними методами класу, такими як "init", "str", "call", і представив приклади їх використання. Покращив функціонал калькулятора, використовуючи об'єктно-орієнтоване програмування.