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

Зміст

[**Звіт до Теми №1** 2](#_Toc149818575)

[**Функції та змінні** 2](#_Toc149818576)

[***Завдання 1*** 2](#_Toc149818577)

[***Завдання 2*** 2](#_Toc149818578)

[***Завдання 3*** 3](#_Toc149818579)

[**Звіт до теми №2** 4](#_Toc149818580)

[**Умовний перехід** 4](#_Toc149818581)

[***Завдання 4*** 4](#_Toc149818582)

[***Завдання 5*** 6](#_Toc149818583)

[**Звіт до теми №3** 7](#_Toc149818584)

[**Цикли** 7](#_Toc149818585)

[***Завдання 6*** 7](#_Toc149818586)

[***Завдання 7*** 8](#_Toc149818587)

[***Завдання 8*** 9](#_Toc149818588)

[**Звіт до теми №4** 10](#_Toc149818589)

[**Виняткові ситуації** 10](#_Toc149818590)

[**Звіт до теми №5** 14](#_Toc149818591)

[**Бібліотеки** 14](#_Toc149818592)

[***Завдання 1*** 14](#_Toc149818593)

[***Завдання 2*** 15](#_Toc149818594)

[***Завдання 3*** 15](#_Toc149818595)

[**Зміст до теми №6** 17](#_Toc149818596)

[**Робота з файлами** 17](#_Toc149818597)

[***Завдання 1*** 17](#_Toc149818598)

[***Завдання 2*** 18](#_Toc149818599)

[**Звіт до теми №7** 19](#_Toc149818600)

[**ООП** 19](#_Toc149818601)

[***Завлання 1*** 19](#_Toc149818602)

[***Завдання 2*** 19](#_Toc149818603)

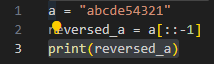
[***Завдання 3*** 21](#_Toc149818604)

**Звіт до Теми №1**

**Функції та змінні**

***Завдання 1***

“Перевернення” слова



Перший рядок створює змінну і присваює їй значення(яке потрібно перевернути). В другому рядку створюю нову змінну reversed\_а, яка використовує [::-1]. Це призводить до того, що значення змінної а перевертається позаду вперед і зберігається в змінній reversed\_а.

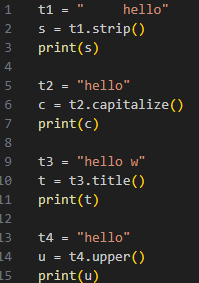
Останній рядок виводить “перевернуте” слово на екран за допомогою функцій print.

Результат



***Завдання 2***

Тестування функцій *strip, capitalize, titale, upper*



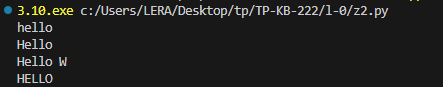
**strip()**видаляє початкові та кінцеві пробелі із строки.

**capitalize()**робить першу букву строки заголовної.

**title()**робить першу букву кожного слова в рядку заголовної, а остальні букви залишаються строчними.

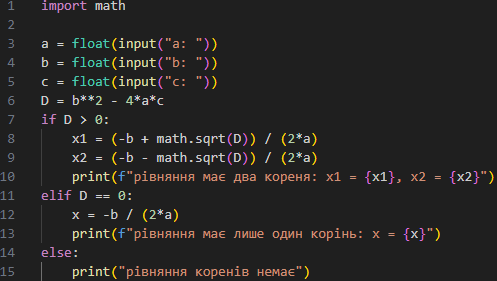
**upper()**преобразует усі букви строки в заголовні букви.

Результат



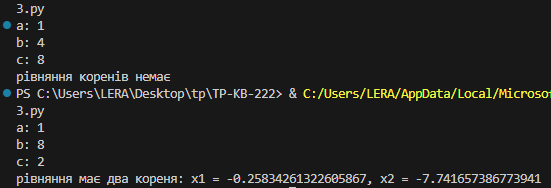
***Завдання 3***

Квадратне рівняння



Цей код спочатку запитує у користувача значення коефіцієнтів a, b і c квадратного рівня. Потім вираховує дискримінант і, в залежності від значення, виводить корні рівняння. Якщо D > 0, то у рівняння два дійсних корня. Якщо D == 0, то у рівняння один дійсний корінь. Якщо D < 0, то у рівняння немає дійсних корнів.

Результат

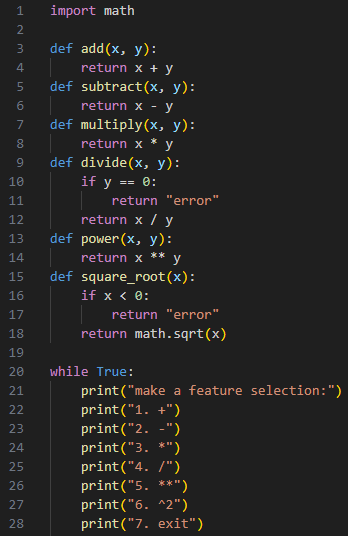


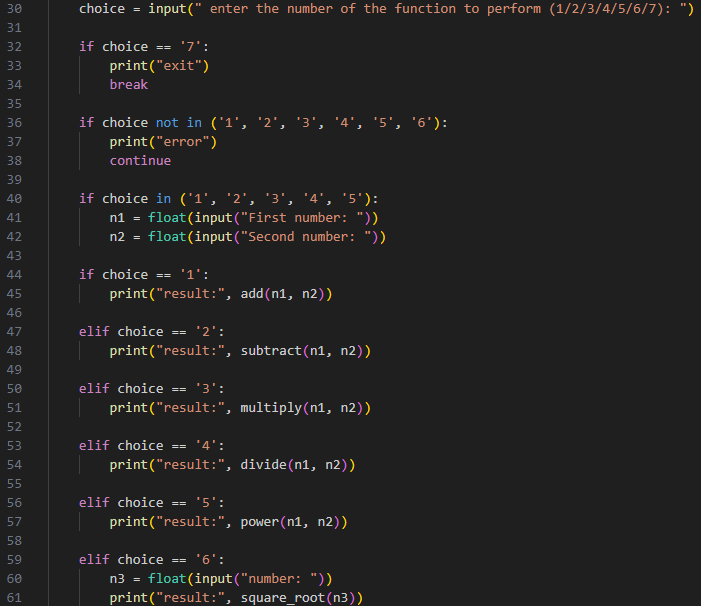
**Звіт до теми №2**

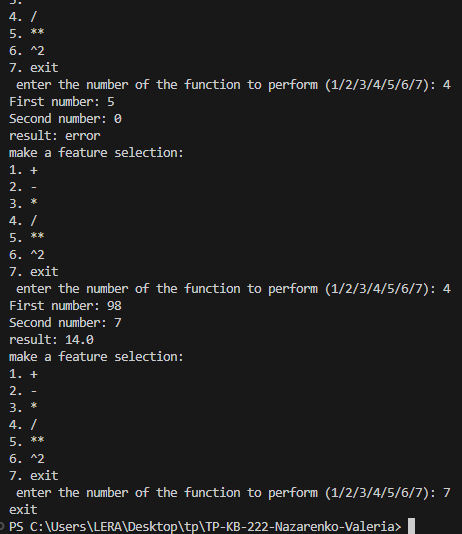
**Умовний перехід**

***Завдання 4***

Написати калькулятор

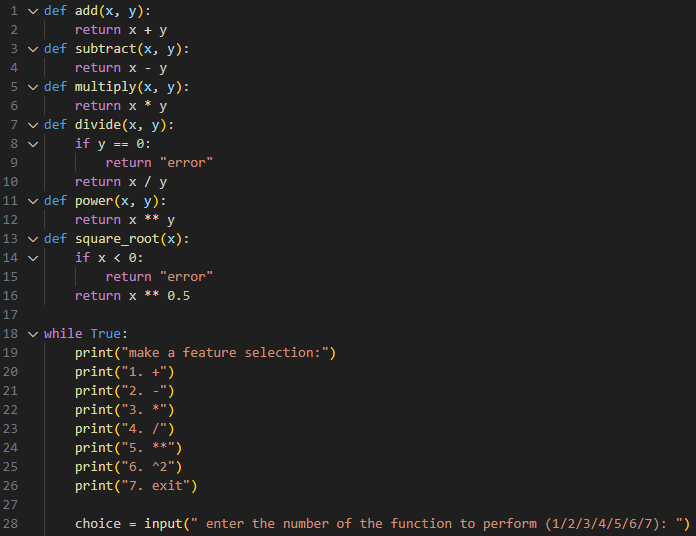


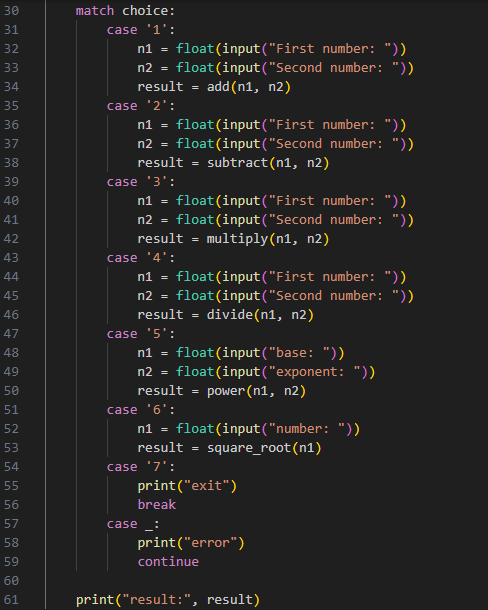


Спочатку імпортую модуль math. Потім визначаю кілька функцій для різних арифметичних операцій, таких як додавання (add), віднімання (subtract), множення (multiply), поділ (divide), зведення в ступінь (power) та вилучення квадратного кореня (square\_root). В основному циклі while True, надала список доступних операцій прошу вибрати потрібний. Залежно від вибору користувача програма виконує відповідну операцію. Якщо користувач вибирає вихід (7), програма завершує виконання. Якщо користувач вводить неправильний вибір (неіснуючу операцію), програма виводить "error" і продовжує виконання циклу. Після введення двох чисел чи одного, програма виконує вибрану операцію та виводить результат.

***Завдання 5***

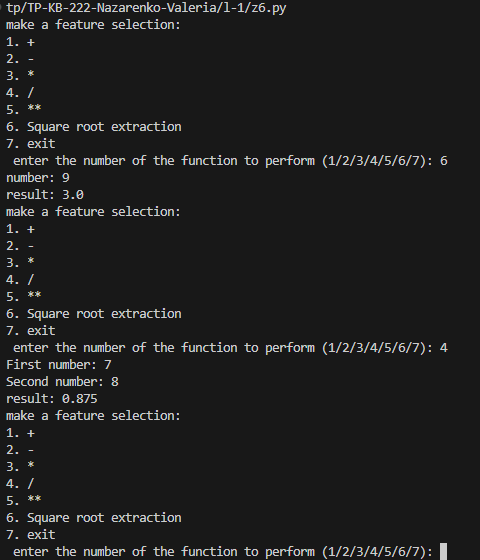
Переписати калькулятор використовуючи match





Тут використовується оператор match - новий оператор, введений у Python 3.10, який надає більш зручний та ясний спосіб зіставлення значень з різними кейсами та виконання відповідних дій. Оператор match надає більш ясне і компактне рішення для зіставлення значень з різними кейсами замість використання ланцюжка if-elif-else.

Результат

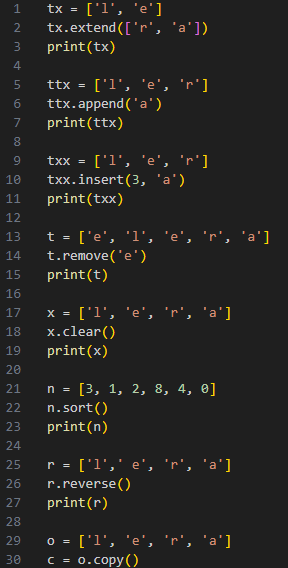


**Звіт до теми №3**

**Цикли**

***Завдання 6***

Тестування та ознайомлення з extend(), append(), insert(id, val), remove(val), clear(), sort(), reverse(), copy()



*extend()* дозволяє додати до списку всі елементи з іншого ітерабельного об'єкта, такого як інший список або кортеж.

*append()* додає вказаний елемент до кінця списку.

*insert()* вставляє вказаний елемент на певну позицію у списку за допомогою індексу.

*remove()* видаляє перший елемент із списку, який має задане значення.

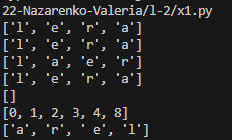
*clear()* видаляє всі елементи зі списку, роблячи його пустим.

*sort()* сортує елементи списку за зростанням (по замовчуванню) або відповідно до іншого ключа, якщо вказаний.

*reverse()* змінює порядок елементів у списку на протилежний.

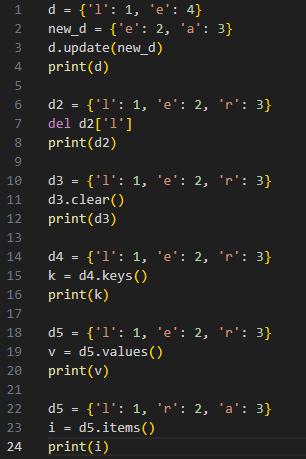
*copy()* створює копію списку. Ця операція не змінює оригінальний список.

Результат



***Завдання 7***

Тестування та ознайомлення з update(), del, clear, keys(), values(), items()



*update()* дозволяє додавати або оновлювати елементи в словнику, використовуючи інші словники або послідовності пар "ключ-значення".

*del* використовується для видалення елементів зі словника за ключем.

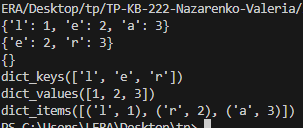
*clear()* видаляє всі елементи із словника, роблячи його пустим.

*keys()* повертає список ключів у словнику.

*values()* повертає список значень у словнику.

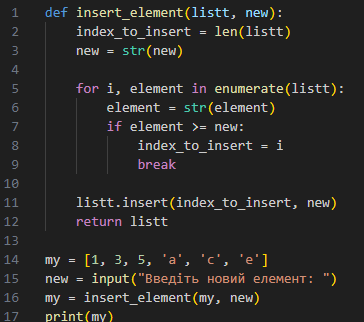
*items()* повертає список кортежів, кожен з яких містить пару "ключ-значення" зі словника.

Результат



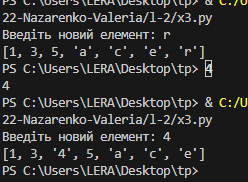
***Завдання 8***

Маючи відсортований список, написати функцію пошуку позиції для вставки нового елементу в список



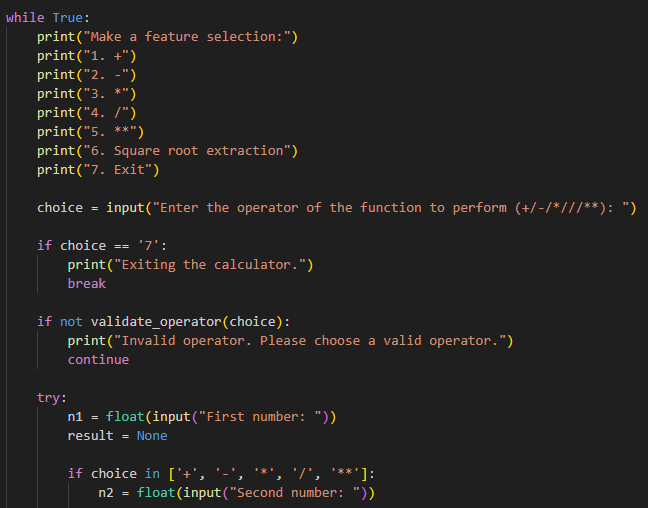
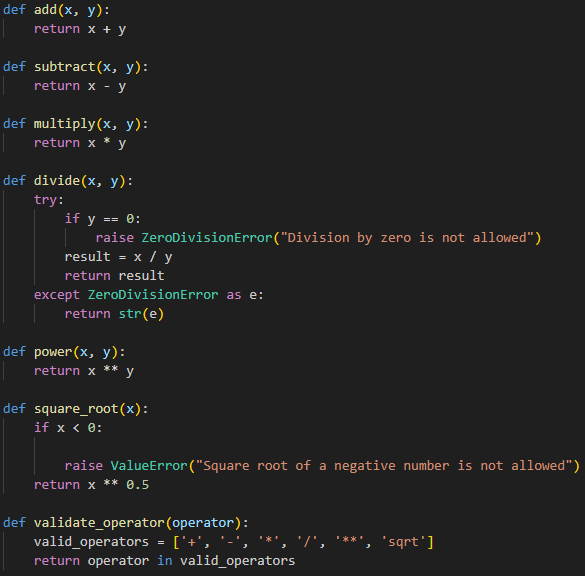
Функція приймає список listt та новий елемент new. Визначається індекс, на який потрібно вставити новий елемент. Цей індекс спочатку встановлюється як довжина списку,означає що новий елемент додається в кінець списку за замовчуванням. Новий елемент перетворюється в строковий тип. Функція проходиться циклом for по елементам списку listt, і кожен елемент також перетворюється в строковий тип. В кожній ітерації циклу функція порівнює елементи. Якщо елемент списку element більший або рівний за значенням нового елемента new, то зберігається індекс цього елемента, і цикл припиняється за допомогою break. Після того, як був знайдений відповідний індекс для вставки, функція використовує метод insert() для додавання нового елемента на визначену позицію в списку listt. Нарешті, функція повертає оновлений список listt з вставленим новим елементом.

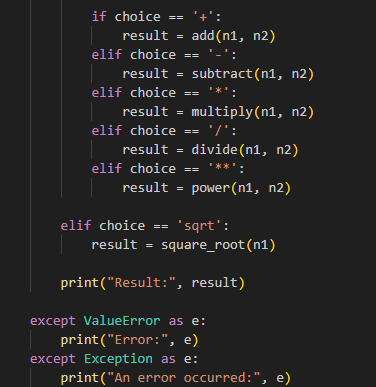
Результат



**Звіт до теми №4**

**Виняткові ситуації**





Даний код розділяє кожну операцію на окрему функцію, що полегшує розуміння та зміну окремих операцій, також включає обробку помилок для ділення на нуль і від’ємних квадратних коренів, надаючи більш інформативні повідомлення про помилки, має перевірку оператора, щоб переконатися, що приймаються лише дійсні оператори.використовуються назви функцій, які чітко описують операції, покращуючи читабельність коду. Тут можна додати більше операцій або функціональних можливостей калькулятора в майбутньому, тому що новий код забезпечує структуровану основу, яка полегшує його розширення.

Також ознайомилася з винятками ось деякі зних:

ZeroDivisionError: Це виняток, який виникає, коли ви намагаєтеся поділити на нуль.

TypeError: Виняток, який виникає, коли операція несумісна з типом даних.

ValueError: Виняток, який виникає, коли функція отримує аргумент правильного типу, але недопустимого значення.

FileNotFoundError: Цей виняток виникає, коли програма намагається відкрити файл, який не існує.

KeyError: Виняток, який виникає, коли ключ не знайдено в словнику.

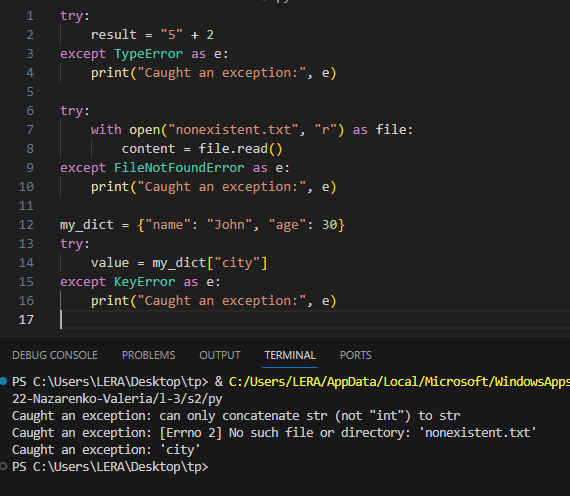
IndexError: Виняток, який виникає, коли індекс виходить за межі діапазону.

NameError: Це виняток, який виникає, коли ви використовуєте змінну або ім'я, яке не було оголошено.

AssertionError: Виняток, який виникає, коли вираз assert повертає False.

EOFError: Виняток, який виникає, коли досягнуто кінця файлу і програма намагається зчитати більше даних.

KeyboardInterrupt: Це виняток, який виникає, коли користувач вводить комбінацію клавіш Ctrl+C для переривання виконання програми.



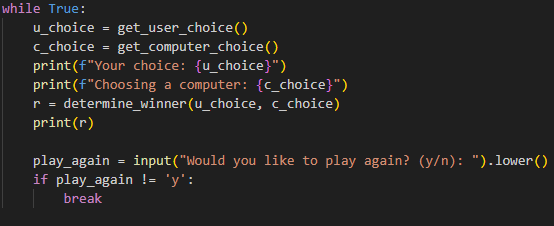
**Звіт до теми №5**

**Бібліотеки**

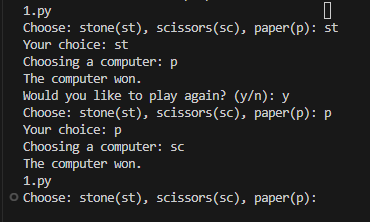
***Завдання 1***

Написати гру Камінь, Ножиці, Папір



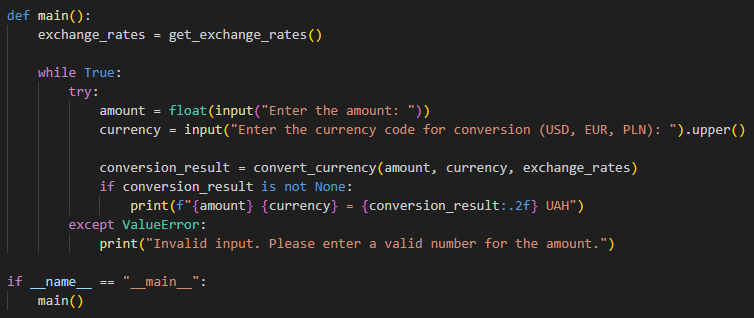
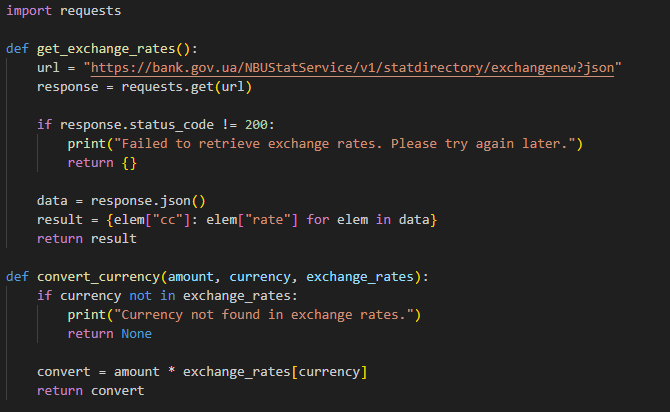


Результат

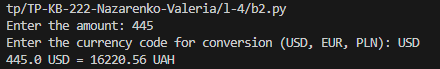


***Завдання 2***

Конвертор валют

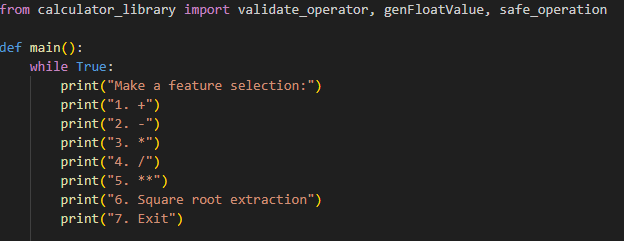


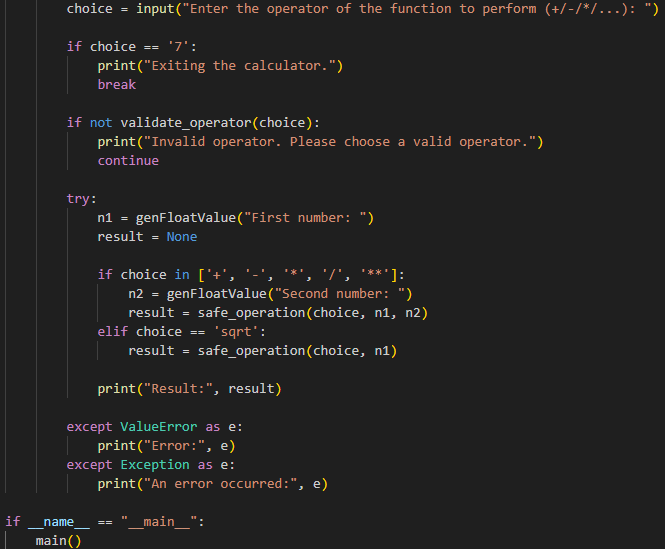
Результат



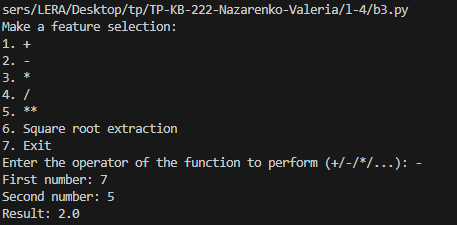
***Завдання 3***

Модулі для програми калькулятор





Результат: все працює)

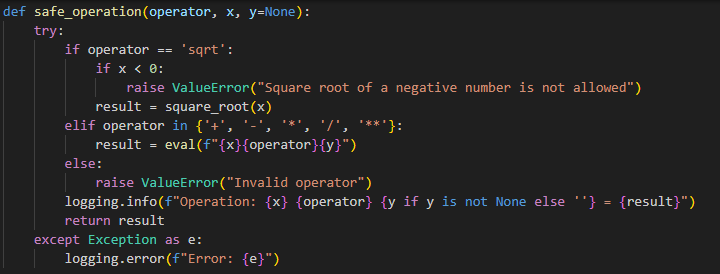


**Зміст до теми №6**

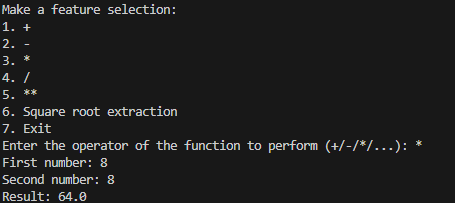
**Робота з файлами**

***Завдання 1***

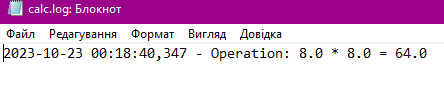
Логування всіх дій в застосунку Калькулятор



Якщо виникають помилки під час обчислень, функція логує помилку за допомогою logging.error() та піднімає ту ж помилку, щоб вона могла бути оброблена в основному коді. Якщо операція виконується успішно, результат також логується за допомогою logging.info() і повертається з функції.

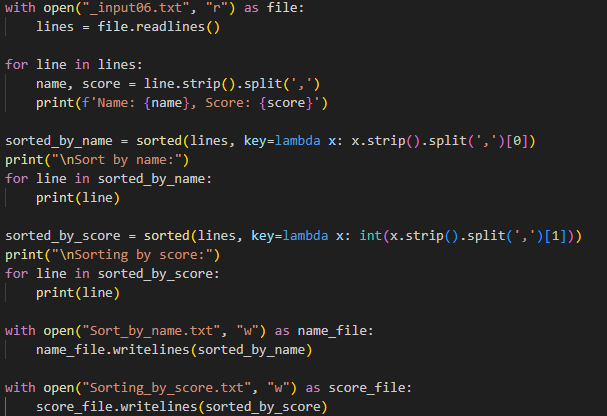


Після успішно виконаної операції, результат логувався

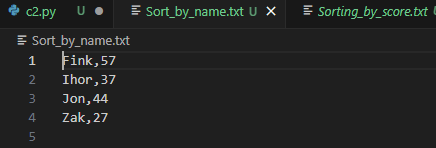


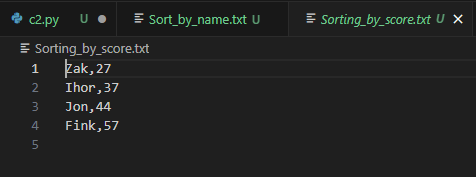
***Завдання 2***

Використання lambda функцій для функції сортування



Цей код читає дані із файла, сортує їх за ім'ям і оцінкою, а потім зберігає сортовані дані в окремі файли з відповідними назвами.





**Звіт до теми №7**

**ООП**

***Завлання 1***

Документація про Сlass

Клас (Class) у програмуванні відноситься до шаблону або структури для створення об'єктів. Класи визначають атрибути (змінні) та методи (функції), які можуть бути використані для створення конкретних об'єктів. Класи є одним із ключових понять в об'єктно-орієнтованому програмуванні (ООП).

class MyClass:

def \_\_init\_\_(self, attribute1, attribute2):

self.attribute1 = attribute1

self.attribute2 = attribute2

def some\_method(self):

my\_object = MyClass("значення1", "значення2")

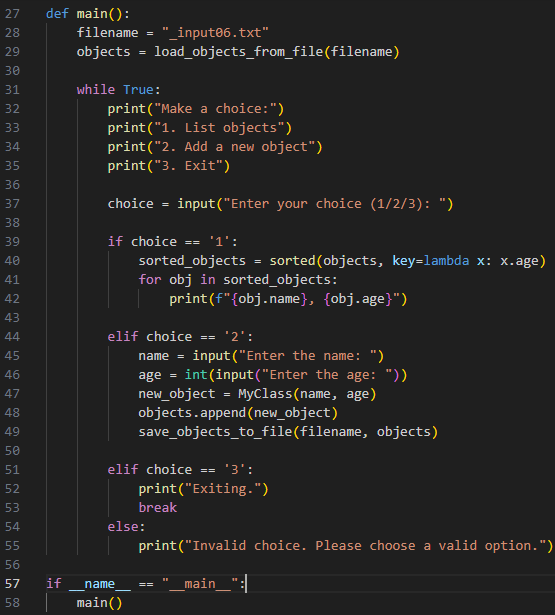
my\_object.some\_method()

***Завдання 2***

Список класів, сортування з lambda

Цей код представляє собою програму, яка дозволяє вам завантажувати, виводити та додавати об'єкти класу MyClass у файл. Основний функціонал програми включає наступні можливості:

* Завантаження об'єктів з файлу \_input06.txt при запуску програми.
* Виведення списку об'єктів, відсортованих за віком (за зростанням).
* Додавання нового об'єкта до списку та збереження його у файлі.



***Завдання 3***

Calc with OOP

У коді створено клас Calculator, який містить методи для виконання арифметичних операцій, які включають додавання, віднімання, множення, ділення, піднесення до ступеня та вираховання квадратного кореня. Ці методи допомагають виконувати відповідні операції над числами.

