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

Чуб Софія, КБ-231

Звіт до Теми №1

Функції та змінні

Під час виконання практичного завдання до Теми №1 було надано варіанти рішення до наступних задач:

1. Перетворення рядка.

Необхідно рядок, що має вигляд "room318" перетворити наступним чином "813moor", вважаючи сталою довжину рядку в 7 символів.

Хід виконання завдання:

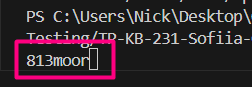
Для виконання завдання треба використати функцію наш рядок записати у змінну number і вивести змінну з операцією зрізу (slice) [::-1] за допомогою функції input. Операція зрізу має формат [start:stop:step], де: start – індекс, з якого починається зріз (включно); stop — індекс, до якого триває зріз (не включено); step – крок, з яким вибираються елементи (якщо крок від'ємний, елементи беруться у зворотному порядку).

Текст програми:

number = "room318"

input(number[::-1])

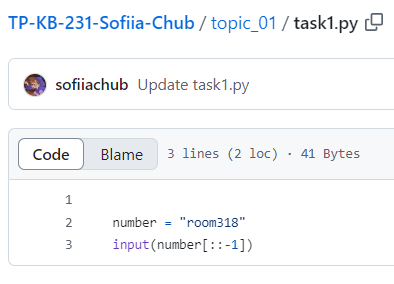
Запущений код:



Посилання на github:

<https://github.com/sofiiachub/TP-KB-231-Sofiia-Chub/blob/main/topic_01/task1.py>

Знімок екрану з посилання на github:



1. Базові функції.

Необхідно протестувати функції, що працюють з рядками: strip(), capitalize(), title(), upper(), lower().

Хід виконання завдання:

У змінну text записали текст “ Good day today” який буде виводитись функцією print та змінюватись при використанні таких функції:

* Функція strip() видаляє всі пробіли з початку та кінця рядка (не всередині тексту). Результатом виконання стане: “Good day today”;
* Функція capitalize() робить першу літеру рядка великою, а всі інші – малими. Результатом виконання стане: “ good day today”;
* Функція title() робить кожне слово в рядку таким, що починається з великої літери. Результатом виконання стане: “ Good Day Today”;
* Функція upper() перетворює всі символи в рядку на великі. Результатом виконання стане: “ GOOD DAY TODAY”;
* Функція lower() перетворює всі символи в рядку на малі літери. Результатом виконання стане: “ good day today”.

Текст програми:

text = " Good day today"

print(text)

print(text.strip())

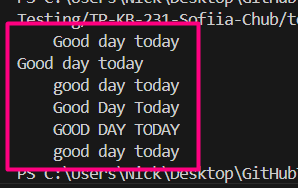
print(text.capitalize())

print(text.title())

print(text.upper())

print(text.lower())

Запушений код:



Посилання на github:

<https://github.com/sofiiachub/TP-KB-231-Sofiia-Chub/blob/main/topic_01/task2.py>

Знімок екрану з посилання на github:



1. Функція пошуку дискримінанту.

Необхідно написати функцію пошуку дискримінанту квадратного рівняння. Програма приймає 3 значення, обраховує та виводить результат.

Хід виконання завдання:

Програма починає роботу з виклику функції askParameters(), що запитує користувача 3 значення (a, b, c). Кожна змінна виводить повідомлення, а користувач вказує значення для кофіцієнту. Змінна визначає тип данних int. Після цього значення змінних a, b, c повертаються за допомогою return. Функція calcDiscriminant(a, b, c) обчислює дискримінант для квадратного рівняння за формулою та повертає результат у result. Після цього функція print виводить повідомлення "Discriminant = (число) ", замість (числа) підставляється result.

Текст програми:

def askParameters():

a = int(input("What's a: "))

b = int(input("What's b: "))

c = int(input("What's c: "))

return a, b, c

def calcDiscriminant(a, b, c):

result = (b \*\* 2) - 4 \* a \*c

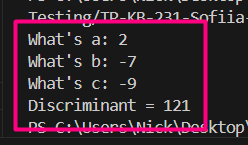
return result

param1, param2, param3 = askParameters()

result = calcDiscriminant(param1, param2, param3)

print("Discriminant =", result)

Запущений код:



Посилання на github:

<https://github.com/sofiiachub/TP-KB-231-Sofiia-Chub/blob/main/topic_01/task3.py>

Знімок екрану з посилання на github:



**Висновки:** Відпрацювала навички та вивчила нові функції на мові Python.

Звіт до Теми №2

Умовний перехід

Під час виконання практичного завдання до Теми №2 було надано варіанти рішення до наступних задач:

1. Функція пошуку коренів квадратного рівняння.

Необхідно написати функцію пошуку коренів квадратного рівняння використовуючи функцію розрахунку дискримінанту з попередньої теми та умовні переходи.

Хід виконання завдання:

Основу коду було взято з 3 завдання 1 теми.

Принцип роботи: Викликається функція askParameters(), яка просить користувача ввести три параметри: a, b та c. Функція повертає значення.

Для обчислення дискримінанту викликається функція calcDiscriminant(a, b, c), яка приймає коефіцієнти a, b та c. Вона обчислює D за формулою і повертає його значення.

Для обчислення коренів рівняння викликається функція calcRoots(a, b, c, D), яка приймає коефіцієнти та значення дискримінанту. В залежності від значення дискримінанту, функція визначає кількість коренів

Після обчислення коренів, програма виводить результат "Roots : ...".

Припустимо, що користувач вводить такі значення a=1, b=-3, c=2. Обчисливши дискримінант ми отримуємо , тобто , отже коренями будуть та .

Текст програми:

import math

def askParameters():

a = int(input("What's a: "))

b = int(input("What's b: "))

c = int(input("What's c: "))

return a, b, c

def calcDiscriminant(a, b, c):

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

return D

def calcRoots(a, b, c, D):

if D > 0:

x1 = (-b + math.sqrt(D)) / (2 \* a)

x2 = (-b - math.sqrt(D)) / (2 \* a)

return x1, x2

elif D == 0:

x = -b / (2 \* a)

return x

else:

root = "There are no"

return root

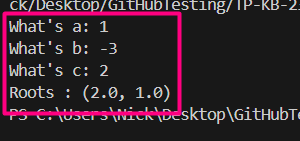
param1, param2, param3 = askParameters()

discriminant = calcDiscriminant(param1, param2, param3)

roots = calcRoots(param1, param2, param3, discriminant)

print("Roots :", roots)

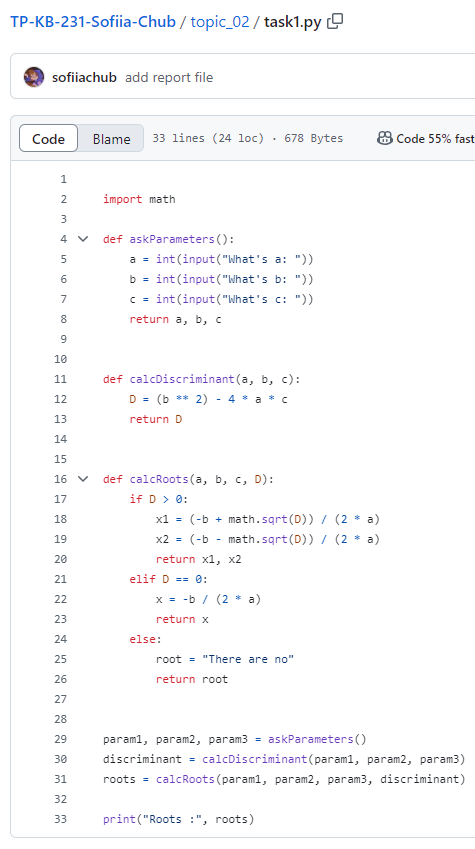
Запущений код:



Посилання на github:

<https://github.com/sofiiachub/TP-KB-231-Sofiia-Chub/blob/main/topic_02/task1.py>

Знімок екрану з посилання на github:



1. Програма калькулятор з конструкцією if else.

Необхідно написати програму калькулятор використовуючи if else конструкцію. Кожна операція має бути виконана в окремій функції.

Хід виконання завдання:

Введення чисел та операції:

Принцип роботи: Користувач запрошується ввести два числа: num1 та num2. Користувач також вибирає математичну операцію (+, -, \*, /).

Викоикається функція calculator(num1, num2, operation), яка приймає два числа та обрану операцію. У функції calculator в залежності від значення змінної operation, викликається відповідна математична функція:

* Якщо операція +, викликається функція add(num1, num2).
* Якщо операція -, викликається функція subtract(num1, num2).
* Якщо операція \*, викликається функція multiply(num1, num2).
* Якщо операція /, викликається функція divide(num1, num2).
* Якщо введена невідома операція, функція повертає "Error.".

У функції divide, якщо num2 дорівнює 0, повертається повідомлення "Error: Division by zero.". Інакше виконується ділення num1 / num2.

Обчислений результат повертається з функції calculator і виводиться на екран у форматі "Result: ...".

Припустимо, що користувач вводить такі значення a=10, b=2 та operation=/. Результатом буде число 5.

Текст програми:

def add(num1, num2):

return num1 + num2

def subtract(num1, num2):

return num1 - num2

def multiply(num1, num2):

return num1 \* num2

def divide(num1, num2):

if num2 != 0:

return num1 / num2

else:

return "Error: Division by zero."

def calculator(num1, num2, operation):

if operation == "+":

result = add(num1, num2)

elif operation == "-":

result = subtract(num1, num2)

elif operation == "\*":

result = multiply(num1, num2)

elif operation == "/":

result = divide(num1, num2)

else:

result = "Error."

return result

num1 = int(input("What's a: "))

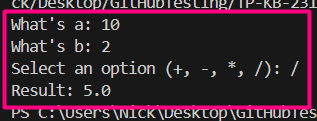
num2 = int(input("What's b: "))

operation = input("Select an option (+, -, \*, /): ")

result = calculator(num1, num2, operation)

print("Result:", result)

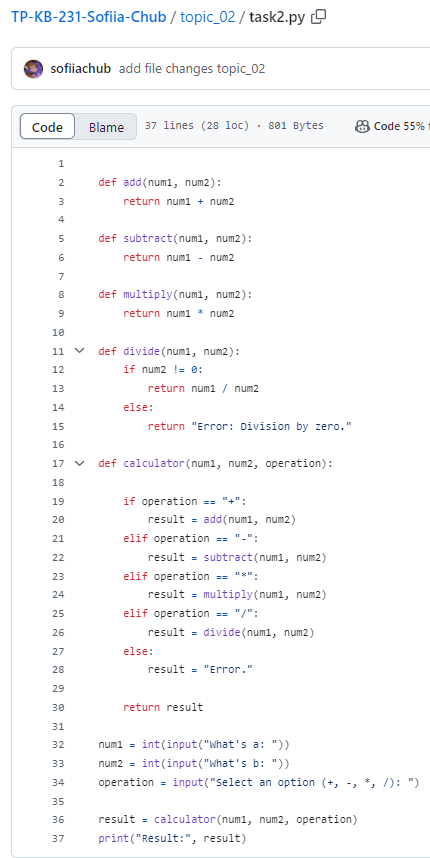
Запущений код:



Посилання на github:

<https://github.com/sofiiachub/TP-KB-231-Sofiia-Chub/blob/main/topic_02/task2.py>

Знімок екрану з посилання на github:



1. Програма калькулятор з конструкцією match.

Написати програму калькулятор використовуючи match конструкцію. Кожна операція має бути виконана в окремій функції.

Хід виконання завдання:

Принцип роботи: Користувач запрошується ввести два числа: num1 та num2. Користувач також вибирає математичну операцію (+, -, \*, /).

Викоикається функція calculator(num1, num2, operation), яка приймає два числа та обрану операцію. Викликається функція calculator(num1, num2, operation), яка приймає два числа та обрану операцію.

У функції calculator використовується конструкція match-case для визначення операції:

* Якщо операція +, викликається функція add(num1, num2).
* Якщо операція -, викликається функція subtract(num1, num2).
* Якщо операція \*, викликається функція multiply(num1, num2).
* Якщо операція /, викликається функція divide(num1, num2).
* Якщо введена невідома операція (жоден з case не спрацює), повертається "Error.".

У функції divide, якщо num2 дорівнює 0, повертається повідомлення "Error: Division by zero.". Інакше виконується ділення num1 / num2.

Обчислений результат повертається з функції calculator і виводиться на екран у форматі "Result: ...".

Припустимо, що користувач вводить такі значення a=3, b=8 та operation=\*. Результатом буде число 24.

Текст програми:

def add(num1, num2):

return num1 + num2

def subtract(num1, num2):

return num1 - num2

def multiply(num1, num2):

return num1 \* num2

def divide(num1, num2):

if num2 != 0:

return num1 / num2

else:

return "Error. Division by zero."

def calculator(num1, num2, operation):

match operation:

case "+":

return add(num1, num2)

case "-":

return subtract(num1, num2)

case "\*":

return multiply(num1, num2)

case "/":

return divide(num1, num2)

case \_:

return "Error."

num1 = int(input("What`s a: "))

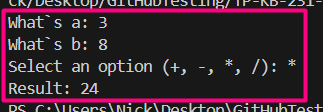
num2 = int(input("What`s b: "))

operation = input("Select an option (+, -, \*, /): ")

result = calculator(num1, num2, operation)

print("Result:", result)

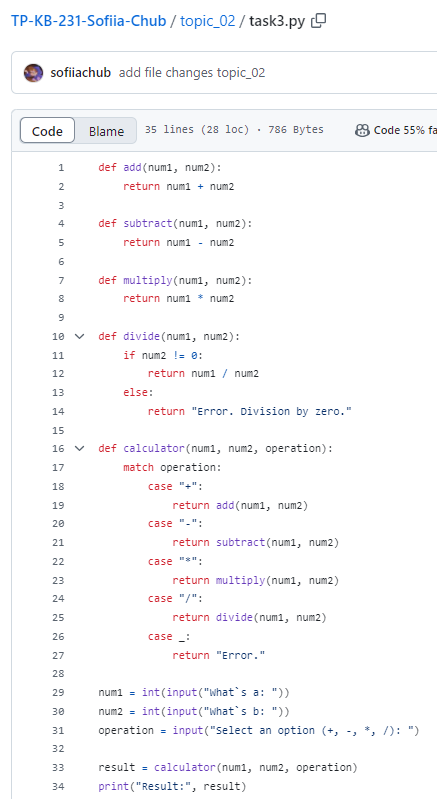
Запущений код:



Посилання на github:

<https://github.com/sofiiachub/TP-KB-231-Sofiia-Chub/blob/main/topic_02/task3.py>

Знімок екрану з посилання на github:



**Висновки:** Я навчилася застосовувати умовні конструкції. Засвоїла принцип роботи if, else та match.