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

Звіт до теми №2

**Умовний перехід**

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

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

Хід виконання завдання

Функція «find\_roots» використовує умовні переходи для підрахунку кількості та значень коренів квадратного рівняння, а також використовує функцію «findD» для розрахунку дискримінанта. Якщо дискримінант більше нуля, то виконується пошук коренів, якщо дисримінант дорівнює нулю, то тоді виконується пошук одного кореня. У інших випадках програма виведе напис: «Немає дійсних коренів».

Код програми

import math

def findD(a, b, c):

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

return discriminant

def find\_roots(a, b, c):

discriminant = findD (a, b, c)

if discriminant > 0:

# Два різних корені

root1 = (-b + math.sqrt(discriminant)) / (2\*a)

root2 = (-b - math.sqrt(discriminant)) / (2\*a)

return (root1, root2)

elif discriminant == 0:

# Один подвійний корінь

root1 = -b / (2\*a)

return (root1,)

else:

# Немає дійсних коренів

return None

a = int(input("Введіть значення a: "))

b = int(input("Введіть значення b: "))

c = int(input("Введіть значення c: "))

roots = find\_roots(a, b, c)

if roots:

if len(roots) == 2:

print(f"Два корені: {roots[0]} та {roots[1]}")

else:

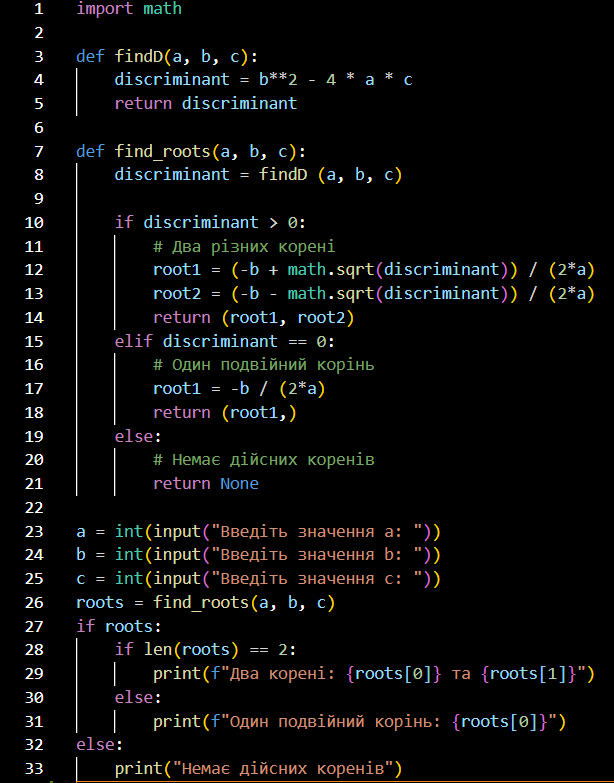
print(f"Один подвійний корінь: {roots[0]}")

else:

print("Немає дійсних коренів")

Посилання на github: <https://github.com/dmitrijpryhodko/TP-KB-221-Dmytrii-Pryhodko/blob/main/topic_02/task1.py>

Скріншот коду програми



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

Хід виконання завдання

Спочатку створив змінні, які будуть приймати значення першого та другого числа. У нашому випадку це змінна «a» і «b». Потім створив змінну, яка буде приймати дію, яку потрібно виконати над числами. Далі створив 4 функції, для кожної з дій. Це додавання, віднімання, ділення та множення. І за допомого конструкцій if та elif задав коли необхідно буде виконувати які дії.

Код програми

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

b = int(input("Enter b: "))

operation = input("Operation: ")

def sum(a,b):

return a + b

def min(a,b):

return a - b

def div(a,b):

return a / b

def mult(a,b):

return a \* b

if operation == "+":

result = sum(a,b)

elif operation == "-":

result = min(a,b)

elif operation == "/":

result = div(a,b)

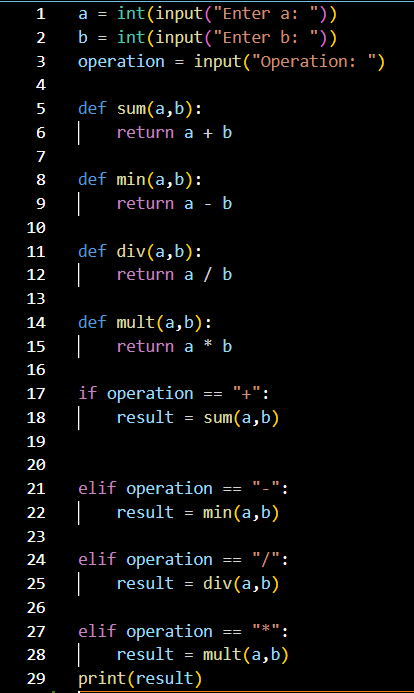
elif operation == "\*":

result = mult(a,b)

print(result)

Посилання на github: <https://github.com/dmitrijpryhodko/TP-KB-221-Dmytrii-Pryhodko/blob/main/topic_02/task2.py>

Скріншот коду програми



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

Хід виконання завдання

Спочатку створив змінні, які будуть приймати числа та дії над ними. Потім використовуючи конструкцію **match** написав які дії потрібно виконувати при виборі тієї чи іншої математичної операції.

Код програми

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

b = int(input("Enter b: "))

operation = input("Operation: ")

match operation:

case "+":

print (a + b)

case "-":

print (a - b)

case "/":

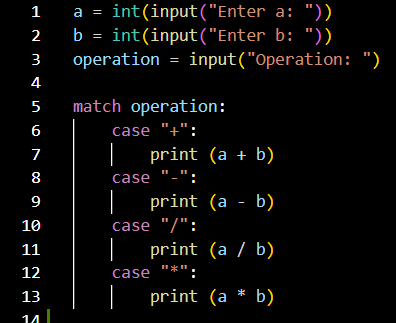
print (a / b)

case "\*":

print (a \* b)

Посилання на github: <https://github.com/dmitrijpryhodko/TP-KB-221-Dmytrii-Pryhodko/blob/main/topic_02/task3.py>

Скріншот коду програми



Висновок: На цьому практичному завданні я створив програму для пошуку квадратних коренів та калькулятор, застосовуючи 2 різних способи. Вивчив конструкцію **match.**