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

Звіт до Теми №1

Функції та змінні

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

1) Необхідно рядок, що має вигляд "abcdefg123" перетворити наступним чином "321gfedcba", вважаючи сталою довжину рядку в 10 символів.

**Хід виконання завдання:**

Текст програми:

|  |
| --- |
| s = "abcdefg123"  result = s[::-1]  print(result) |

Посилання на github:

<https://github.com/vtkachova/TP-KB-242-Tkachova-Veronika/blob/main/topic_01/task1.py>

Знімок екрану з посилання на github:

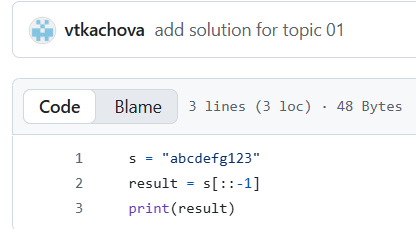


Рис. 1 – код 1 завдання на github.

Виконання програми:



Рис. 2 – результат виконання 1 завдання.

2) Виконати тестування функцій, що працюють з рядками: strip(), capitalize(), title(), upper(), lower().

**Хід виконання завдання:**

Текст програми:

# strip() – прибирає пробіли на початку та в кінці

text1 = "   hello world   "

print("strip():", text1.strip())

# capitalize() – робить першу літеру великою, решта стають маленькими

text2 = "hello world"

print("capitalize():", text2.capitalize())

# title() – робить першу літеру кожного слова великою

text3 = "hello world"

print("title():", text3.title())

# upper() – всі букви стають великими

text4 = "hello world"

print("upper():", text4.upper())

# lower() – всі букви стають маленькими

text5 = "Hello World"

print("lower():", text5.lower())

Посилання на github:

<https://github.com/vtkachova/TP-KB-242-Tkachova-Veronika/blob/main/topic_01/task2.py>

Знімок екрану з посилання на github:

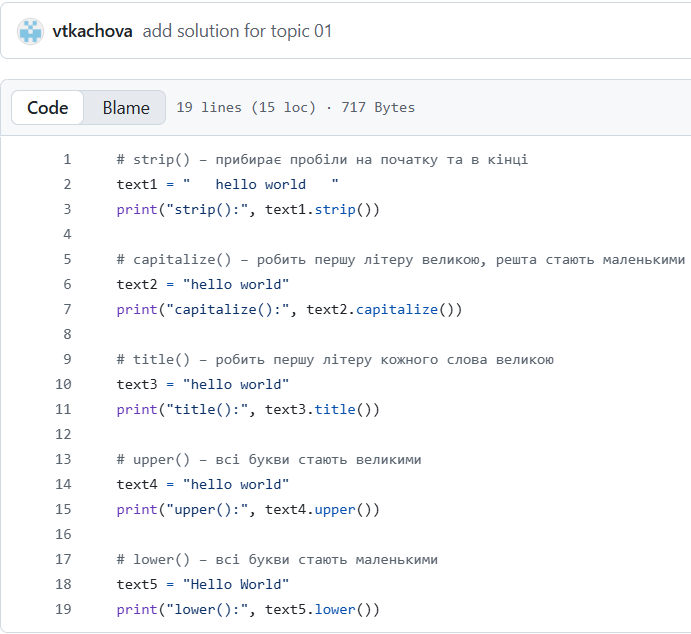


Рис. 3 – код 2 завдання на github.

Виконання програми:

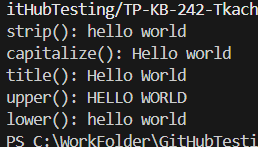


Рис. 4 – результат виконання 2 завдання.

3) Написати функцію пошуку дискримінанту квадратного рівняння.

**Хід виконання завдання:**

Текст програми:

def discr(a, b, c):

    return b\*b - 4\*a\*c

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

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

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

D = discr(a, b, c)

print("Дискримінант =", D)

Посилання на github:

<https://github.com/vtkachova/TP-KB-242-Tkachova-Veronika/blob/main/topic_01/task3.py>

Знімок екрану з посилання на github:

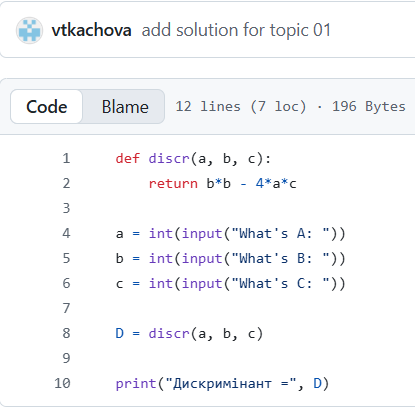


Рис. 5 – код 3 завдання на github.

Виконання програми:

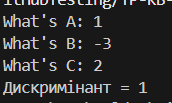


Рис. 6 – результат виконання 3 завдання.

Звіт до Теми №2

Умовний перехід

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

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

**Хід виконання завдання:**

Текст програми:

def discr(a, b, c):

    return b\*b - 4\*a\*c

# Вводимо коефіцієнти

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

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

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

# Рахуємо дискримінант

D = discr(a, b, c)

print("Дискримінант =", D)

# Шукаємо корені

if D > 0:

    x1 = (-b + D\*\*0.5) / (2\*a)

    x2 = (-b - D\*\*0.5) / (2\*a)

    print("Два корені: x1 =", x1, ", x2 =", x2)

elif D == 0:

    x = -b / (2\*a)

    print("Один корінь: x =", x)

else:

    print("Коренів немає (D < 0)")

Посилання на github:

<https://github.com/vtkachova/TP-KB-242-Tkachova-Veronika/blob/main/topic_02/task1.py>

Знімок екрану з посилання на github:

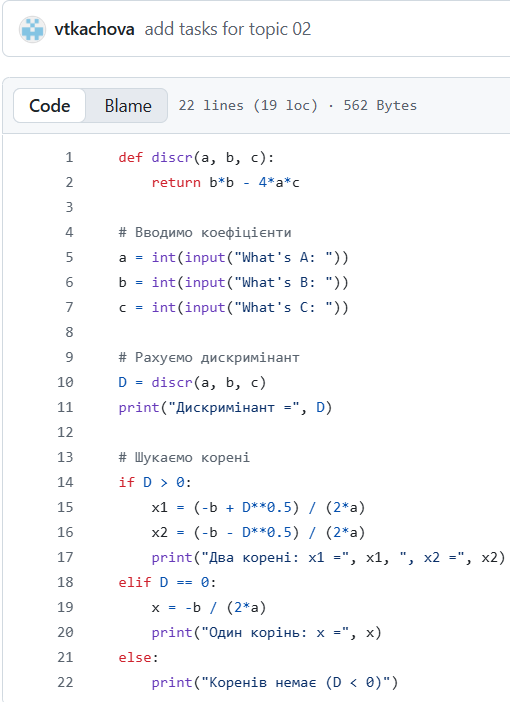


Рис. 7 – код 1 завдання topic02 на github.

Виконання програми:

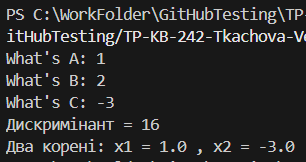


Рис. 8 – результат виконання 1 завдання у topic02.

2) Написати програму калькулятор використовуючи **if else** конструкцію. Кожна операція має бути виконана в окремій функції.

**Хід виконання завдання:**

Текст програми:

a = float(input("Введіть перше число: "))

b = float(input("Введіть друге число: "))

op = input("Введіть дію (+, -, \*, /): ")

if op == "+":

    print("Результат:", a + b)

elif op == "-":

    print("Результат:", a - b)

elif op == "\*":

    print("Результат:", a \* b)

elif op == "/":

    if b != 0:

        print("Результат:", a / b)

    else:

        print("Помилка: ділення на нуль!")

else:

    print("Невірний оператор!")

Посилання на github:

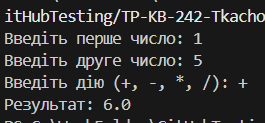
<https://github.com/vtkachova/TP-KB-242-Tkachova-Veronika/blob/main/topic_02/task2.py>

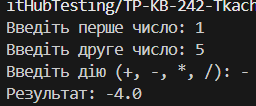
Знімок екрану з посилання на github:

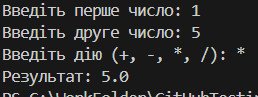


Рис. 9 – код 2 завдання topic02 на github.

Виконання завдання:







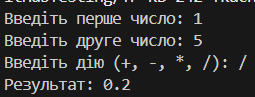


Рис. 10 – 13 – результат виконання 2 завдання у topic02.

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

**Хід виконання завдання:**

Текст програми:

a = float(input("Введіть перше число: "))

b = float(input("Введіть друге число: "))

op = input("Введіть дію (+, -, \*, /): ")

match op:

    case "+":

        print("Результат:", a + b)

    case "-":

        print("Результат:", a - b)

    case "\*":

        print("Результат:", a \* b)

    case "/":

        if b != 0:

            print("Результат:", a / b)

        else:

            print("Помилка: ділення на нуль!")

    case \_:

        print("Невірний оператор!")

Посилання на github:

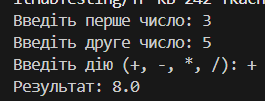
<https://github.com/vtkachova/TP-KB-242-Tkachova-Veronika/blob/main/topic_02/task3.py>

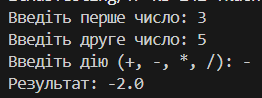
Знімок екрану з посилання на github:

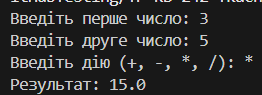


Рис. 14 – код 3 завдання topic02 на github.

Виконання завдання:







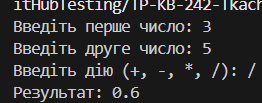


Рис. 15 – 18 – результат виконання 3 завдання.

Звіт до Теми №3

Цикли

Під час виконання практичного завдання до Теми №3 було надано варіанти рішення до наступних задач:

1. Написати програму калькулятор з постійними запитами на введення нових даних та операцій. За основу взяти програму калькулятор з попередньої теми. Реалізувати механізм завершення програми після отримання відповідної команди.

**Хід виконання завдання:**

Текст програми:

print(">>> Калькулятор <<<")

while True:

    op = input("Введіть дію (+, -, \*, /) або 'вихід' для завершення: ")

    if op == "вихід":

        print("Програма завершена.")

        break

    a = float(input("Введіть перше число: "))

    b = float(input("Введіть друге число: "))

    if op == "вихід":

        print("Програма завершена.")

        break

    if op == "+":

        print("Результат:", a + b)

    elif op == "-":

        print("Результат:", a - b)

    elif op == "\*":

        print("Результат:", a \* b)

    elif op == "/":

        if b != 0:

            print("Результат:", a / b)

        else:

            print("Помилка: ділення на нуль!")

    else:

        print("Невірний оператор!")

    print()

Посилання на github:

<https://github.com/vtkachova/TP-KB-242-Tkachova-Veronika/blob/main/topic_03/task1.py>

Знімок екрану з посилання на github:



Рис. 19 – код 1 завдання topic03 на github.

Виконання завдання:

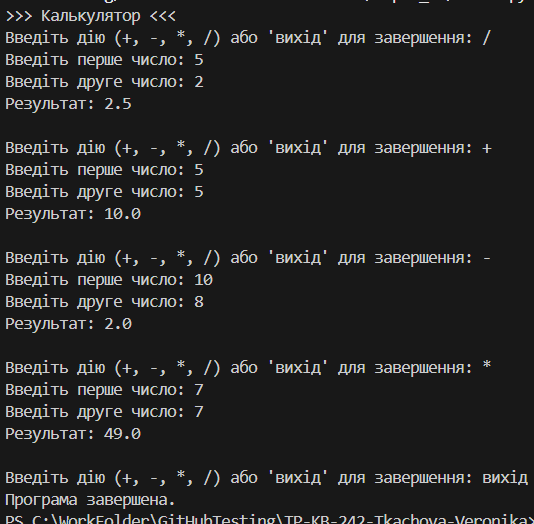


Рис. 20 – результат виконання 1 завдання.

2) Написати програму тестування функцій списків таких як: extend(), append(), insert(id, val), remove(val), clear(), sort(), reverse(), copy().

**Хід виконання завдання:**

Текст програми:

print(">>> Тестування функцій списків <<<")

# Створюємо початковий список

numbers = [3, 1, 4]

print("Початковий список:", numbers)

# append() — додає один елемент в кінець списку

numbers.append(5)

print("Після append(5):", numbers)

# extend() — додає кілька елементів

numbers.extend([6, 7])

print("Після extend([6, 7]):", numbers)

# insert() — вставляє елемент у вказане місце

numbers.insert(1, 10)

print("Після insert(1, 10):", numbers)

# remove() — видаляє зазначений елемент зі списку

numbers.remove(4)

print("Після remove(4):", numbers)

# copy() — створює копію списку

numbers\_copy = numbers.copy()

print("Копія списку:", numbers\_copy)

# sort() — сортує список

numbers.sort()

print("Після sort():", numbers)

# reverse() — перевертає список

numbers.reverse()

print("Після reverse():", numbers)

# clear() — очищає список

numbers.clear()

print("Після clear():", numbers)

Посилання на github:

<https://github.com/vtkachova/TP-KB-242-Tkachova-Veronika/blob/main/topic_03/task2.py>

Знімок екрану з посилання на github:

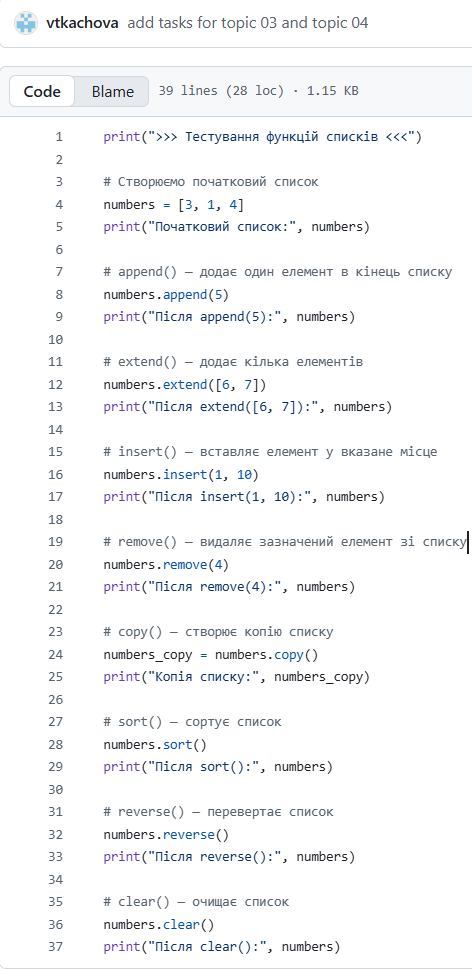


Рис. 21 – код 2 завдання topic03 на github.

Виконання завдання:

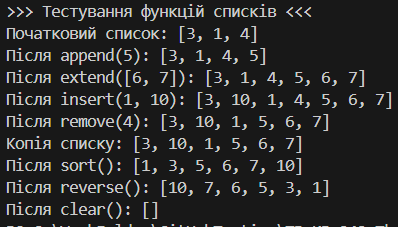


Рис. 22 – результат виконання 2 завдання.

3) Написати програму тестування функцій словників таких як: update(), del(), clear(), keys(), values(), items().

**Хід виконання завдання:**

Текст програми:

print(">>> Тестування функцій словників <<<")

student = {"name": "Veronika", "age": 18}

print("Початковий словник:", student)

# update() — оновлює словник

student.update({"city": "Chernihiv"})

print("Після update():", student)

# keys() — повертає всі ключі

print("Ключі словника:", student.keys())

# values() — повертає всі значення

print("Значення словника:", student.values())

# items() — повертає пари ключ:значення

print("Пари ключ:значення:", student.items())

# del — видаляє зазначений елемент за ключем

del student["city"]

print("Після del:", student)

# clear() — очищає словник

student.clear()

print("Після clear():", student)

Посилання на github:

<https://github.com/vtkachova/TP-KB-242-Tkachova-Veronika/blob/main/topic_03/task3.py>

Знімок екрану з посилання на github:

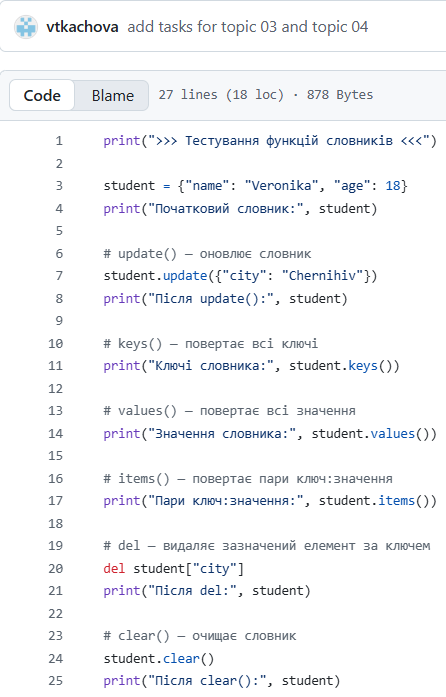


Рис. 23 – код 3 завдання topic03 на github.

Виконання завдання:

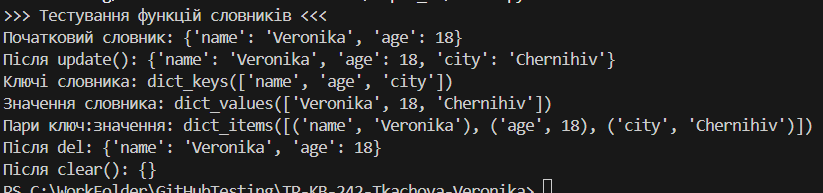


Рис. 24 – результат виконання 3 завдання.

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

**Хід виконання завдання:**

Текст програми:

print(">>> Пошук позиції для вставки <<<")

# Функція пошуку позиції для вставки

def find\_insert\_position(sorted\_list, x):

    for i in range(len(sorted\_list)):

        if x < sorted\_list[i]:

            return i  # повертаємо позицію, куди треба вставити

    return len(sorted\_list)  # якщо елемент найбільший — в кінець

numbers = [1, 3, 5, 7, 9]

print("Відсортований список:", numbers)

x = int(input("Введіть число для вставки: "))

pos = find\_insert\_position(numbers, x)

print("Позиція для вставки:", pos)

numbers.insert(pos, x)

print("Новий список:", numbers)

Посилання на github:

<https://github.com/vtkachova/TP-KB-242-Tkachova-Veronika/blob/main/topic_03/task4.py>

Знімок екрану з посилання на github:



Рис. 25 – код 4 завдання topic03 на github.

Виконання завдання:

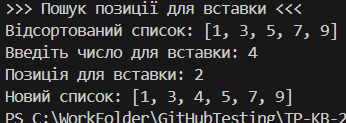


Рис. 26 – результат виконання 4 завдання.

Звіт до Теми №4

Виняткові ситуації

Під час виконання практичного завдання до Теми №4 було надано варіанти рішення до наступних задач:

**Попередні умови**: реалізована програма калькулятор, що використовує метод нескінченного введення даних для обробки. Всі дії (додавання, віднімання, множення, ділення) реалізовані як окремі функції та використовуються у відповідних місцях.

1) Розширити програму калькулятор функцією запитів даних для виконання операцій від користувача, що обробляє виняткові ситуації.

**Хід виконання завдання:**

Текст програми:

# Функції для операцій

def add(x, y):

    return x + y

def subtract(x, y):

    return x - y

def multiply(x, y):

    return x \* y

def divide(x, y):

    return x / y

while True:

    print("\nОберіть операцію: +, -, \*, / або 'exit' для виходу")

    op = input("Введіть операцію: ")

    if op.lower() == "exit":

        print("Програма завершена.")

        break

    if op not in ('+', '-', '\*', '/'):

        print("Невідома операція. Спробуйте ще раз.")

        continue

    try:

        x = float(input("Введіть перше число: "))

        y = float(input("Введіть друге число: "))

        if op == '+':

            result = add(x, y)

        elif op == '-':

            result = subtract(x, y)

        elif op == '\*':

            result = multiply(x, y)

        elif op == '/':

            result = divide(x, y)

        print(f"Результат: {result}")

    except ValueError:

        print("Помилка: введено не число. Спробуйте ще раз.")

    except Exception as e:

        print(f"Невідома помилка: {e}")

Посилання на github:

<https://github.com/vtkachova/TP-KB-242-Tkachova-Veronika/blob/main/topic_04/task1.py>

Знімок екрану з посилання на github:



Рис. 27 – код 1 завдання topic04 на github.

Виконання завдання:

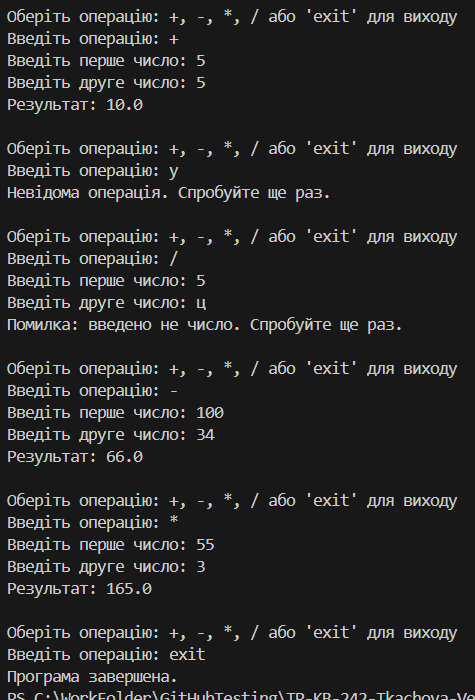


Рис. 28 – результат виконання 1 завдання.

2) Розширити функцію ділення обробкою виняткової ситуації ділення на нуль.

**Хід виконання завдання:**

Текст програми:

# Функції для операцій

def add(x, y):

    return x + y

def subtract(x, y):

    return x - y

def multiply(x, y):

    return x \* y

def divide(x, y):

    try:

        return x / y

    except ZeroDivisionError:

        return "Помилка: ділення на нуль неможливе!"

while True:

    print("\nОберіть операцію: +, -, \*, / або 'exit' для виходу")

    op = input("Введіть операцію: ")

    if op.lower() == "exit":

        print("Програма завершена.")

        break

    if op not in ('+', '-', '\*', '/'):

        print("Невідома операція. Спробуйте ще раз.")

        continue

    try:

        x = float(input("Введіть перше число: "))

        y = float(input("Введіть друге число: "))

        if op == '+':

            result = add(x, y)

        elif op == '-':

            result = subtract(x, y)

        elif op == '\*':

            result = multiply(x, y)

        elif op == '/':

            result = divide(x, y)

        print(f"Результат: {result}")

    except ValueError:

        print("Помилка: введено не число. Спробуйте ще раз.")

Посилання на github:

<https://github.com/vtkachova/TP-KB-242-Tkachova-Veronika/blob/main/topic_04/task2.py>

Знімок екрану з посилання на github:

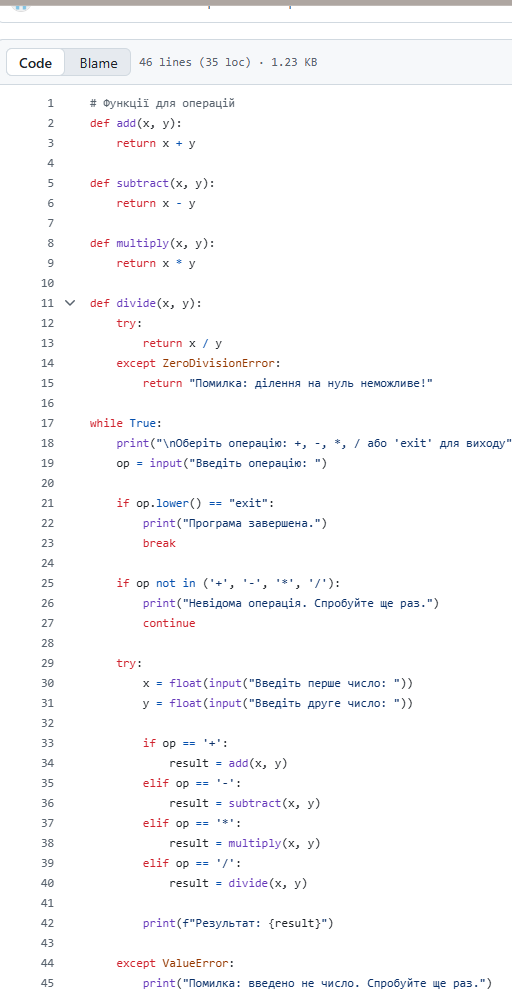


Рис. 29 – код 2 завдання topic04 на github.

Виконання завдання:

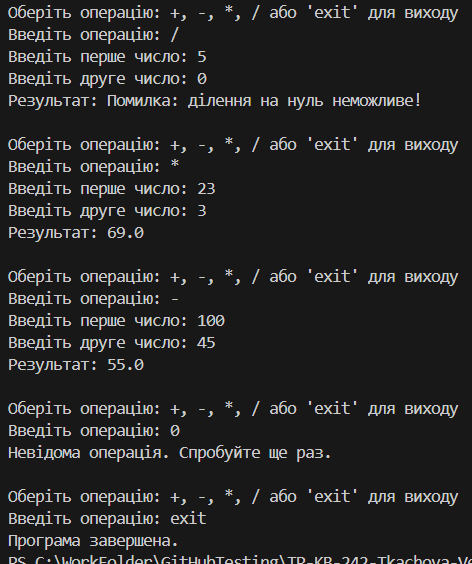


Рис. 30 – результат виконання 2 завдання.