**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. **Обернути рядок в зворотному порядку.**

Необхідно написати функцію, яка приймає рядок і повертає його у зворотному порядку.

Хід виконання завдання:

Для реалізації цього завдання використовується механізм зрізу рядків у Python. Зріз із кроком -1 дозволяє обійти рядок у зворотному напрямку.

Створюється функція reverse\_string(s), яка приймає рядок і повертає його у зворотному порядку за допомогою синтаксису s[::-1].

Текст програми:

|  |
| --- |
| def reverse\_string(s):  return s[::-1]  # Тест  string = "Hello, world!"  print(reverse\_string(string)) # "!dlrow ,olleH" |

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

Потрібно протестувати роботу стандартних функцій Python для роботи з рядками: strip(), capitalize(), title(), upper(), lower().

Хід виконання завдання:

Для кожної функції створюється об’єкт stripped, capitalized, titled, uppercased, lowercased, що зберігає результат застосування відповідної функції до переданого рядка.

Усі результати зберігаються у словник, який потім виводиться за допомогою циклу for.

Текст програми:

|  |
| --- |
| def test\_string\_functions(s):  stripped = s.strip()  capitalized = s.capitalize()  titled = s.title()  uppercased = s.upper()  lowercased = s.lower()  return {  "strip": stripped,  "capitalize": capitalized,  "title": titled,  "upper": uppercased,  "lower": lowercased  }  string = " hello world "  results = test\_string\_functions(string)  for func, result in results.items():  print(f"{func}: {result}") |

1. **Написати функцію пошуку дискримінанту квадратного рівняння.**

Потрібно написати функцію, яка обчислює дискримінант для квадратного рівняння за формулою: 𝐷=𝑏2−4𝑎𝑐

Хід виконання завдання:

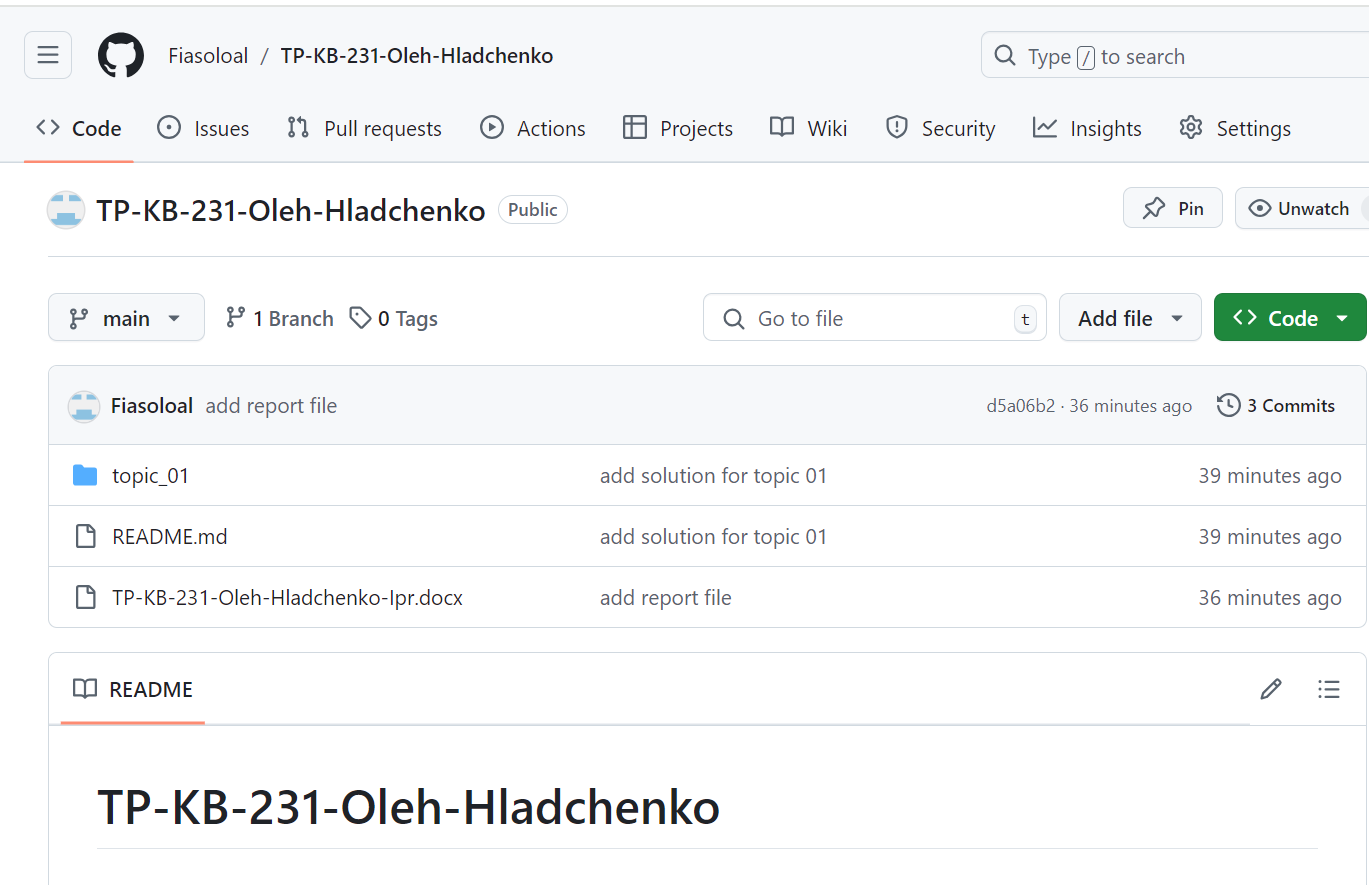
Вхідні параметри: коефіцієнти a, b, c квадратного рівняння. Створюється функція find\_discriminant(a, b, c), яка приймає три параметри і обчислює дискримінант за допомогою формули: b\*\*2 - 4\*a\*c.

Текст програми:

|  |
| --- |
| def find\_discriminant(a, b, c):  discriminant = b\*\*2 - 4\*a\*c  return discriminant  a, b, c = 1, -3, 2  d = find\_discriminant(a, b, c)  print(f"Дискримінант: {d}") |

Посилання на github:

https://github.com/Fiasoloal/TP-KB-231-Oleh-Hladchenko/tree/main

Знімок екрану з посилання на github: