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

Звіт до Теми №1

Функції та змінні

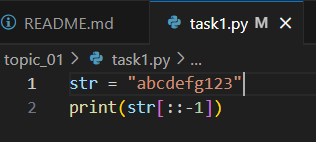
Під час виконання практичного завдання до Теми №1 було надано варіанти рішення до наступних задач:

**Перетворення рядка**

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

Хід виконання завдання:

Використовую змінну де записую наш текст, далі використовую функцію ::-1 в команді print яка виведе текст на екран в зворотному порядку





Текст програми:

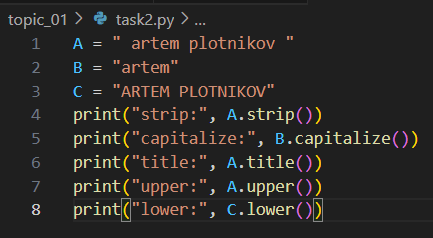
|  |
| --- |
| str = "abcdefg123"  print(str[::-1]) |

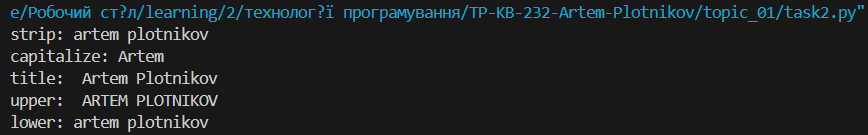
**Тестування функцій**

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

Хід виконання завдання:

Я написав кілька змінних зі своїм ім'ям, там де вони доречно будуть працювати з певними функціями, використав функції та показав їх у терміналі





Текст програми:

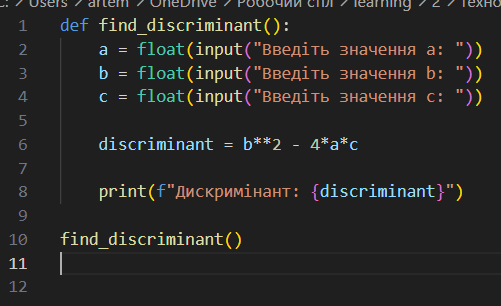
|  |
| --- |
| A = " artem plotnikov "  print(A.strip())  B = "artem"  print(B.capitalize())  print(A.title())  print(A.upper())  C = "ARTEM PLOTNIKOV"  print(C.lower()) |

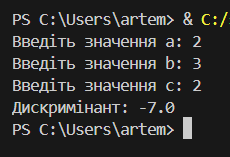
**Функція пошуку дискримінанта**

Необхідно написати функцію пошуку дискримінанту квадратного рівняння

Хід виконання завдання:

Спершу створюю змінні для вводу даних коефіцієнтів a, b, c, потім пишу функцію безпосередньо формули дискримінанту, вивожу значення в термінал та звертаюся в кінці до функції

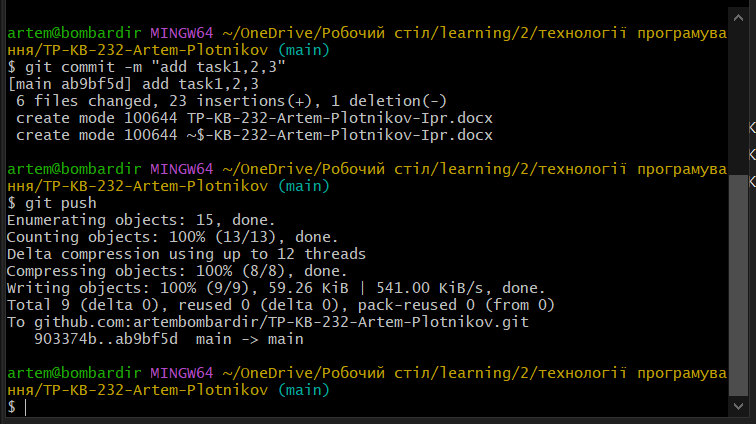




Текст програми:

|  |
| --- |
| def find\_discriminant():  a = float(input("Введіть значення a: "))  b = float(input("Введіть значення b: "))  c = float(input("Введіть значення c: "))    discriminant = b\*\*2 - 4\*a\*c    print(f"Дискримінант: {discriminant}")  find\_discriminant() |

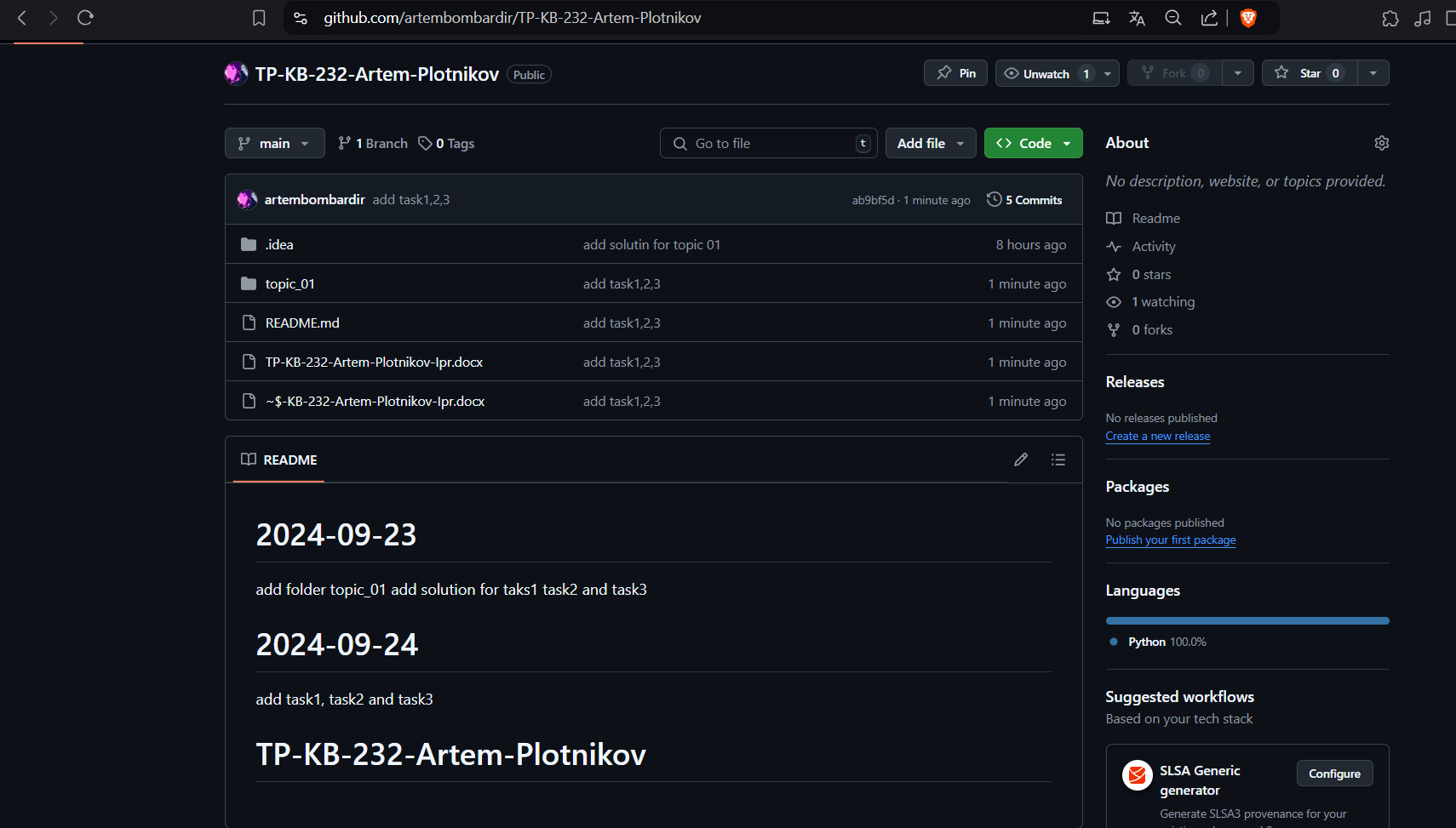
Зберігаю, роблю змінні в файлі read me та відсилаю на гітхаб



Посилання на github:

https://github.com/artembombardir/TP-KB-232-Artem-Plotnikov

Знімок екрану з посилання на github:



Звіт до Теми №2

Умовний перехід

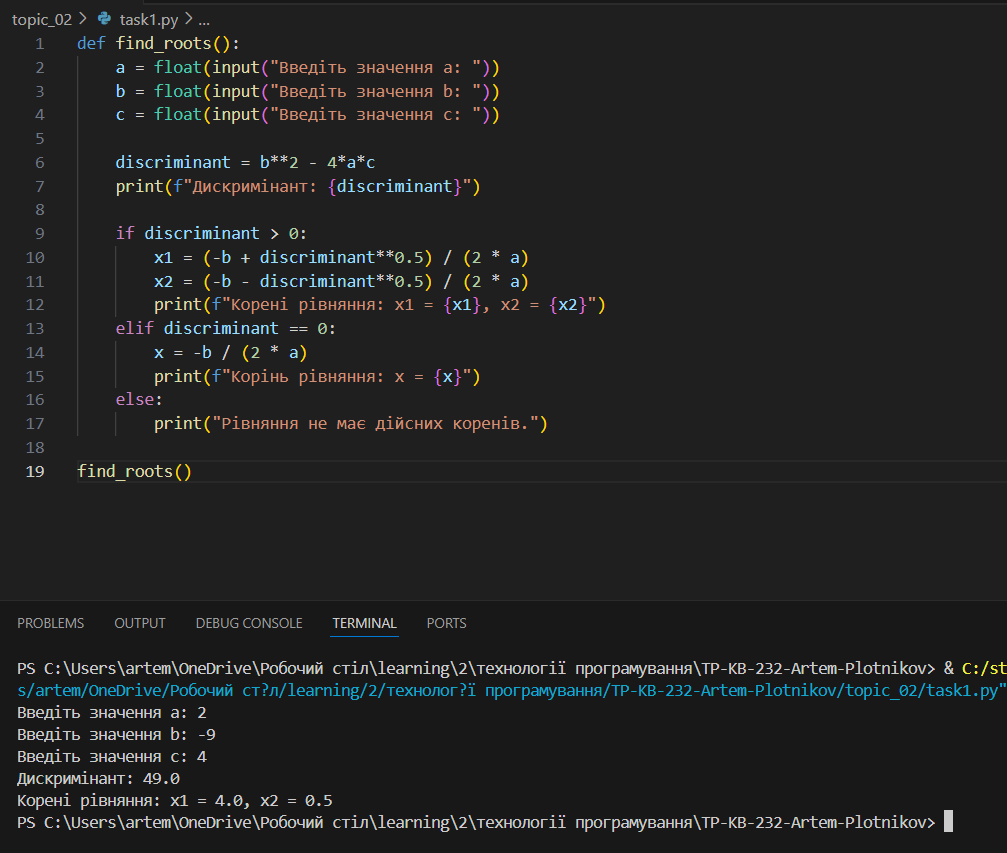
Під час виконання практичного завдання до Теми №2 було надано варіанти рішення до наступних задач:

**Пошук коренів**

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

Хід виконання завдання:

Використовую наступні конструкції: if D > 0, print два корені рівняння, elif D == 0 print один корінь, else print дійсних коренів не існує



Текст програми:

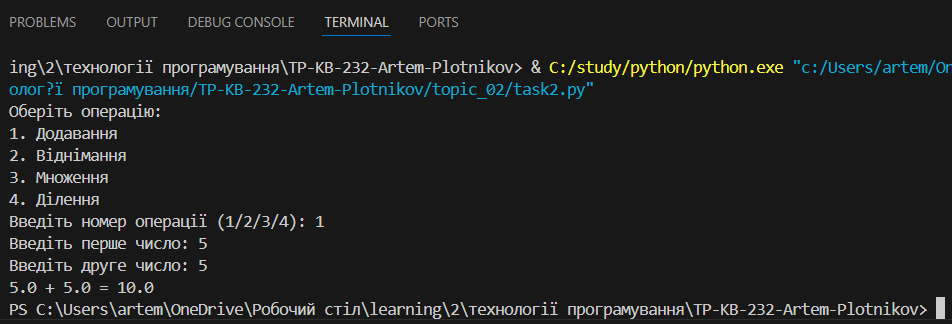
|  |
| --- |
| def find\_roots():  a = float(input("Введіть значення a: "))  b = float(input("Введіть значення b: "))  c = float(input("Введіть значення c: "))    discriminant = b\*\*2 - 4\*a\*c  print(f"Дискримінант: {discriminant}")    if discriminant > 0:  x1 = (-b + discriminant\*\*0.5) / (2 \* a)  x2 = (-b - discriminant\*\*0.5) / (2 \* a)  print(f"Корені рівняння: x1 = {x1}, x2 = {x2}")  elif discriminant == 0:  x = -b / (2 \* a)  print(f"Корінь рівняння: x = {x}")  else:  print("Рівняння не має дійсних коренів.")  find\_roots() |

**Програма калькулятор**

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

Хід виконання завдання:

Додав окремі функції для додавання, віднімання, множення, ділення(використовую if - else, спираючись на те що ділення на 0 неможливе), після чого використав if else для вибору користувачем його завдання для калькулятора



Текст програми:

|  |
| --- |
| 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):  if y != 0:  return x / y  else:  return "Помилка: Ділення на нуль неможливе!"  def calculator():  print("Оберіть операцію:")  print("1. Додавання")  print("2. Віднімання")  print("3. Множення")  print("4. Ділення")    choice = input("Введіть номер операції (1/2/3/4): ")    num1 = float(input("Введіть перше число: "))  num2 = float(input("Введіть друге число: "))    if choice == '1':  print(f"{num1} + {num2} = {add(num1, num2)}")  elif choice == '2':  print(f"{num1} - {num2} = {subtract(num1, num2)}")  elif choice == '3':  print(f"{num1} \* {num2} = {multiply(num1, num2)}")  elif choice == '4':  print(f"{num1} / {num2} = {divide(num1, num2)}")  else:  print("Невірний вибір операції. Спробуйте ще раз.")  calculator() |

**Програма калькулятор 2**

Необхідно написати програму калькулятор використовуючи **match** конструкцію. Кожна операція має бути виконана в окремій функції.

Хід виконання завдання:

Все теж саме що і в минулому завданні, тільки замість if else використовую match case

Текст програми:

|  |
| --- |
| match choice:  case '1':  print(f"{num1} + {num2} = {add(num1, num2)}")  case '2':  print(f"{num1} - {num2} = {subtract(num1, num2)}")  case '3':  print(f"{num1} \* {num2} = {multiply(num1, num2)}")  case '4':  print(f"{num1} / {num2} = {divide(num1, num2)}")  case \_:  print("Невірний вибір операції. Спробуйте ще раз.") |

Звіт до Теми №3

Цикли

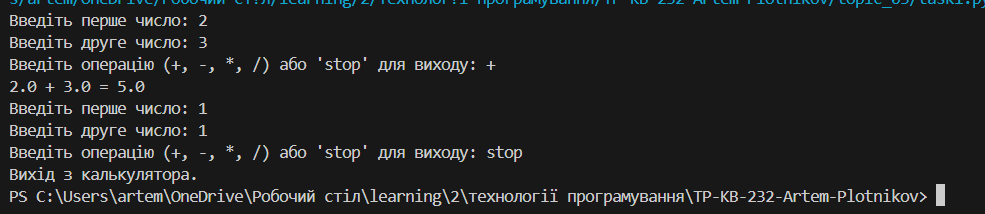
Під час виконання практичного завдання до Теми №1 було надано варіанти рішення до наступних задач:

**Калькулятор з постійними запитами на введення нових даних**

Необхідно 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:

Використовую цикл while до калькулятора, допоки користувач не введе слово stop у виборі операції



Текст програми:

|  |
| --- |
| 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):  if y != 0:  return x / y  else:  return "Помилка: ділення на нуль!"  def calculator():  while calculator:  num1 = float(input("Введіть перше число: "))  num2 = float(input("Введіть друге число: "))  operation = input("Введіть операцію (+, -, \*, /) або 'stop' для виходу: ")  if operation == 'stop':  print("Вихід з калькулятора.")  break    match operation:  case '+':  print(f"{num1} + {num2} = {add(num1, num2)}")  case '-':  print(f"{num1} - {num2} = {subtract(num1, num2)}")  case '\*':  print(f"{num1} \* {num2} = {multiply(num1, num2)}")  case '/':  print(f"{num1} / {num2} = {divide(num1, num2)}")  case \_:  print("Невірна операція.")  calculator() |