**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.

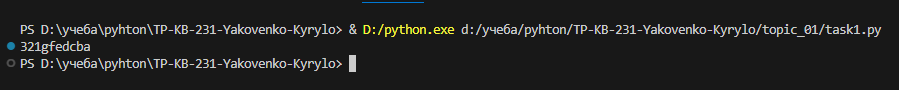
Хід виконання завдання:

1. Для початку я стоврит змінну input\_string в яку помістили текст "abcdefg123".
2. Потім з допомогою функції зрізу reversed\_string = input\_string[::-1] я перевернув рядок.
3. В кінці добавив функцію виводу print(reversed\_string), щоб побачити результат.

Текст програми:

|  |
| --- |
| # Вхідний рядок  input\_string = "abcdefg123"  # Перевертаємо рядок за допомогою зрізів  reversed\_string = input\_string[::-1]  # Виводимо результат  print(reversed\_string) |

Перевірка:



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

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

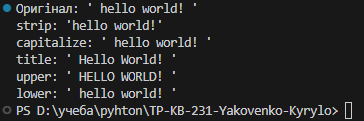
Хід виконання завдання:

1. Спочатку виводиться оригінальний рядок, що буде тестуватись. У цьому випадку рядок test\_str = " hello world! "
2. Надаю цьому рядку функцію print(f"Оригінал: '{test\_str}'"), щоб показати початковий вигляд тексту. Літера f перед рядком дозволяє безпосередньо вбудовувати значення змінних або виразів у рядок.
3. **Strip** - видаляє пробіли або інші вказані символи з початку і кінця рядка. Надаю цьому рядку функцію print(f"strip: '{test\_str.strip()}'") для виведення тексту.
4. **Capitalize** - робить першу літеру рядка великою, а всі інші — малими. Надаю цьому рядку функцію print(f"capitalize: '{test\_str.capitalize()}'") для виведення тексту.
5. **Title** - робить кожне слово в рядку з великої літери. Надаю цьому рядку функцію print(f"title: '{test\_str.title()}'") для виведення тексту.
6. **Upper** - переводить всі символи рядка в верхній регістр. Надаю цьому рядку функцію print(f"upper: '{test\_str.upper()}'") для виведення тексту.
7. **Lower** -переводить всі символи рядка в нижній регістр. Надаю цьому рядку функцію print(f"lower: '{test\_str.lower()}'") для виведення тексту.

Текст програми:

|  |
| --- |
| # Тестовий рядок  test\_str = " hello world! "  print(f"Оригінал: '{test\_str}'")  # 1. strip(видаляє пробіли або інші вказані символи з початку і кінця рядка)  print(f"strip: '{test\_str.strip()}'")  # 2. capitalize(робить першу літеру рядка великою, а всі інші — малими.)  print(f"capitalize: '{test\_str.capitalize()}'")  # 3. title(робить кожне слово в рядку з великої літери.)  print(f"title: '{test\_str.title()}'")  # 4. upper(переводить всі символи рядка в верхній регістр.)  print(f"upper: '{test\_str.upper()}'")  # 5. lower(переводить всі символи рядка в нижній регістр.)  print(f"lower: '{test\_str.lower()}'") |

Перевірка



**Написати функцію пошуку дискримінанту квадратного рівняння.**

Треба написати функцію пошуку дискримінанту квадратного рівняння.

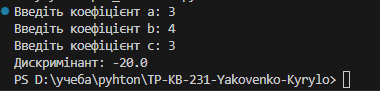
Хід виконання завдання:

1. Спочатку я створив функцію, яка буде обчислювати дискримінант на основі коефіцієнтів a, b і c. Дискримінант обчислюється за формулою D=b^2−4ac, тому ця формула буде основою для функції.
2. Для того, щоб можна було ввести коефіцієнти, використовуємо функцію input(). Важливо перетворити введені дані у числа (тип float), щоб обробляти не лише цілі, але й дробові значення.
3. Запитуємо кожен коефіцієнт окремо і зберігаємо їх у змінні.
4. Після того як користувач ввів значення коефіцієнтів, їх потрібно передати в нашу функцію discriminant(a,b,c). Результат обчислення дискримінанту зберігаємо у змінну D.
5. Використовуємо функцію print(), щоб вивести результат обчислення дискримінанту на екран. Використовуємо f-строку для зручного форматування виведення.

Текст програми:

|  |
| --- |
| def discriminant(a, b, c):      # Обчислюємо дискримінант за формулою      D = b\*\*2 - 4\*a\*c      return D  # Отримуємо коефіцієнти від користувача  a = float(input("Введіть коефіцієнт a: "))  b = float(input("Введіть коефіцієнт b: "))  c = float(input("Введіть коефіцієнт c: "))  # Обчислюємо дискримінант  D = discriminant(a, b, c)  # Виводимо результат  print(f"Дискримінант: {D}") |

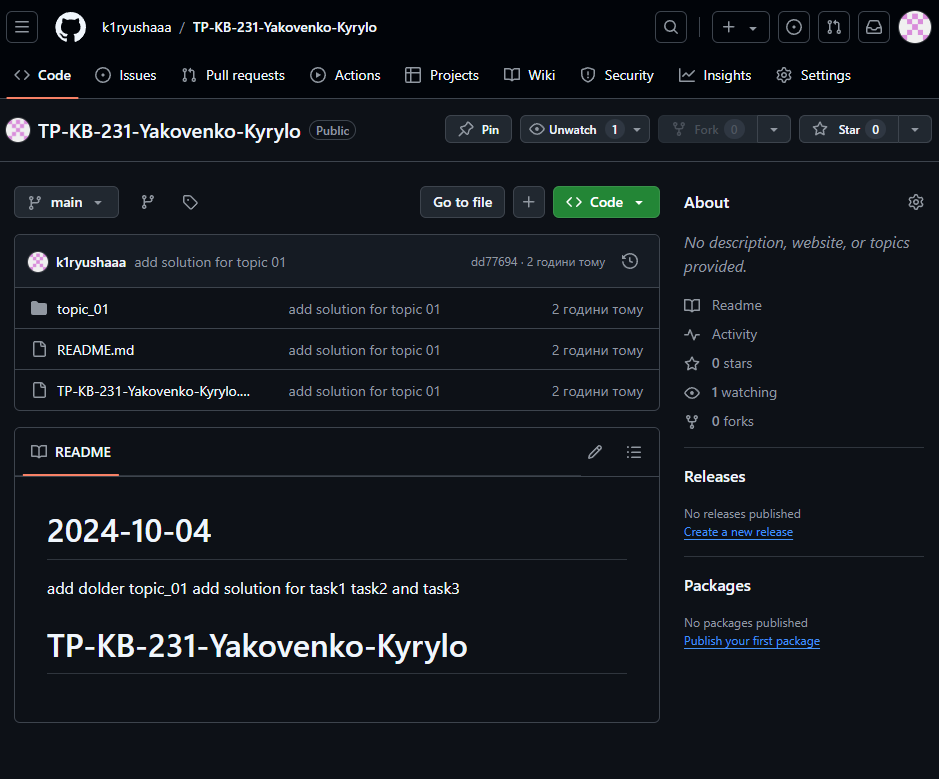
Перевірка



Посилання на github:

https://github.com/k1ryushaaa/TP-KB-231-Yakovenko-Kyrylo

Знімок екрану з посилання на github:



Звіт до Теми №2

Умовний перехід

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

**Написання функції пошуку коренів квадратного рівняння використовуючи функцію розрахунку дискримінанту з попередньої теми та умовні переходи.**

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

1. Спочатку викликав функцію «enter()», яка вивела підказку для користувача і попросила ввести коефіцієнти для рівняння вигляду ax^2 + bx + c. Функція «enter()» зчитала значення коефіцієнтів a, b і c від користувача та повернула їх.

2. Далі, з допомогою функції «discriminant(a, b, c)» я обчислив дискримінант за формулою D = b^2 - 4ac.

3. Потім викликав функцію «find\_roots(a, b, c)», яка: спочатку обчислила дискримінант D (використовуючи функцію «discriminant»).

- Якщо D > 0, обчислила два дійсні корені x\_1 і x\_2 за формулами:

x1 = (-b + D\*\*0.5) / (2 \* a)

x2 = (-b - D\*\*0.5) / (2 \* a)

і повернула їх як результат.

Якщо D = 0, обчислила один дійсний корінь x за формулою:

x = -b / (2 \* a)

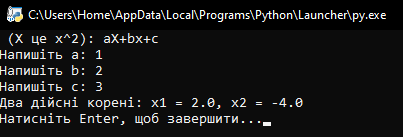
і повернула його як результат.

Якщо D < 0, повернула повідомлення про відсутність дійсних коренів.

4. На завершення вивів результат виконання функції «find\_roots», щоб побачити корені рівняння або повідомлення про їхню відсутність.

5. Додав команду «input("Натисніть Enter, щоб завершити...")», щоб користувач зміг переглянути результат перед закриттям програми.  
Текст програми:

|  |
| --- |
| def enter():  print(" (X це x^2): aX+bx+c")  a = float(input('Напишіть a: '))  b = float(input('Напишіть b: '))  c = float(input('Напишіть c: '))  return a, b, c  def discriminant(a, b, c):  return b\*\*2 - 4 \* a \* c  def find\_roots(a, b, c):  D = discriminant(a, b, c)    if D > 0:  x1 = (-b + D\*\*0.5) / (2 \* a)  x2 = (-b - D\*\*0.5) / (2 \* a)  return f"Два дійсні корені: x1 = {x1}, x2 = {x2}"  elif D == 0:  x = -b / (2 \* a)  return f"Один дійсний корінь: x = {x}"  else:  return "Дійсних коренів немає"    a, b, c = enter()  d = discriminant(a, b, c)  print(find\_roots(a, b, d))  input("Натисніть Enter, щоб завершити...") |

Перевірка:  


**Написання програми калькулятора використовуючи if else конструкцію.**

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

1. Спочатку визначив функції для кожної математичної операції:

* add(x, y) — функція для додавання, яка повертає суму чисел x та y .
* subtract(x, y) — функція для віднімання, яка повертає різницю чисел x та y .
* multiply(x, y) — функція для множення, яка повертає добуток чисел x та y .
* divide(x, y) — функція для ділення, яка перевіряє, чи y не дорівнює нулю, і якщо це так, виконує ділення x на y. Якщо y = 0, повертає повідомлення "Помилка: ділення на нуль".

2. Створив функцію calculator() , яка:

Просить користувача вибрати операцію (`+`, `-`, `\*`, `/`) і вводить її значення у змінну operation.

Запитує у користувача два числа (перше і друге) та зберігає їх у змінні num1 і num2.

3. Виконує перевірку вибраної операції:

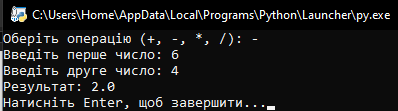
* Якщо вибрано `+`, викликає функцію add(num1, num2) і виводить результат.
* Якщо вибрано `-`, викликає функцію subtract(num1, num2) і виводить результат.
* Якщо вибрано `\*`, викликає функцію multiply(num1, num2) і виводить результат.
* Якщо вибрано `/`, викликає функцію divide(num1, num2) і виводить результат.
* Якщо вибрано некоректну операцію, виводить повідомлення "Помилка: некоректний вибір операції".

4. Викликав функцію calculator() для запуску програми.

5. Додав команду input("Натисніть Enter, щоб завершити..."), щоб користувач міг побачити результат перед закриттям програми.

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):  return x / y if y != 0 else "Помилка: ділення на нуль"  def calculator():  operation = input("Оберіть операцію (+, -, \*, /): ")  num1 = float(input("Введіть перше число: "))  num2 = float(input("Введіть друге число: "))    if operation == '+':  print("Результат:", add(num1, num2))  elif operation == '-':  print("Результат:", subtract(num1, num2))  elif operation == '\*':  print("Результат:", multiply(num1, num2))  elif operation == '/':  print("Результат:", divide(num1, num2))  else:  print("Помилка: некоректний вибір операції.")  calculator()  input("Натисніть Enter, щоб завершити...") |

Перевірка:  


**Написати програму калькулятор використовуючи match конструкцію.**

Необхідно написати програму калькулятор використовуючи match конструкцію. Кожна операція має бути виконана в окремій функції.  
  
Хід виконання завдання:

1. Спочатку визначив функції для кожної математичної операції:

* add(x, y) — функція для додавання, яка повертає суму чисел x та y .
* subtract(x, y) — функція для віднімання, яка повертає різницю чисел x та y .
* multiply(x, y) — функція для множення, яка повертає добуток чисел x та y .
* divide(x, y) — функція для ділення, яка перевіряє, чи y не дорівнює нулю, і якщо це так, виконує ділення x на y. Якщо y = 0, повертає повідомлення "Помилка: ділення на нуль".

2. Створив функцію calculator() , яка:

Просить користувача вибрати операцію (`+`, `-`, `\*`, `/`) і вводить її значення у змінну operation.

Запитує у користувача два числа (перше і друге) та зберігає їх у змінні num1 і num2.

3. Використав конструкцію match для вибору операції залежно від значення змінної operation:

Якщо операція `+`, викликає функцію add(num1, num2) і виводить результат.

Якщо операція `-`, викликає функцію subtract(num1, num2) і виводить результат.

Якщо операція `\*`, викликає функцію multiply(num1, num2) і виводить результат.

Якщо операція `/`, викликає функцію divide(num1, num2) і виводить результат.

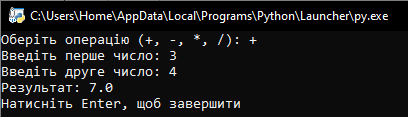
Якщо введено некоректну операцію, виконує case \_ і виводить повідомлення "Помилка: некоректний вибір операції".

4. Викликав функцію calculator() для запуску калькулятора.

5. Додав команду input("Натисніть Enter, щоб завершити"), щоб користувач міг переглянути результат перед завершенням програми.

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):  return x / y if y != 0 else "Помилка: ділення на нуль"  def calculator():  operation = input("Оберіть операцію (+, -, \*, /): ")  num1 = float(input("Введіть перше число: "))  num2 = float(input("Введіть друге число: "))    match operation:  case '+':  print("Результат:", add(num1, num2))  case '-':  print("Результат:", subtract(num1, num2))  case '\*':  print("Результат:", multiply(num1, num2))  case '/':  print("Результат:", divide(num1, num2))  case \_:  print("Помилка: некоректний вибір операції.")  calculator()  input("Натисніть Enter, щоб завершити") |

Перевірка:  
  
  
Посилання на github:

<https://github.com/k1ryushaaa/TP-KB-231-Yakovenko-Kyrylo>   
Знімок екрану з посилання на github:

