**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:

Створив функцію що розвертає рядок.

Текст програми:

|  |
| --- |
| string = "abcdefg123"  def reverse(var):  var = var[::-1]  return var  reversedString = reverse(string)  print(reversedString) |

**Тестування методів для рядків**

Необхідно протестувати деякі методи для рдяків.

Хід виконання завдання:

Протестував такі методи:

Strip() – видаляє пробіли і табуляції спереду та позаду рядку.

Capitalize() – робить 1 букву рядку вищого регістру, інші нижчого.

Title() – всі 1 букви кожного слова в рядку робить вищого регістру.

Upper() – всі букви до верхнього регістру.

Lower() – всі букви до нижчого регістру.

Find() – шукає позицію символа в рядку.

Split() – розбиває рядок по заданому символу та записує частини в масив.

Count() – рахує кількість входжень символу до рядку.

Swapcase() – міняє місцями регістри літер.

Текст програми:

|  |
| --- |
| #strip() method  stringWithSpaces = " I am Denis "  print (stringWithSpaces.strip())  #capitalize() method  notCapitalizedString = "i am a student"  print (notCapitalizedString.capitalize())  #title() method  string = "Hello i am denis"  print (string.title())  #upper() and lower() methods  weirdString = "hElLo i aM dEnIs aNd i aM a sTuDeNt"  print (weirdString.upper())  print (weirdString.lower())  #some other  #find() method  print(string.find("i"))  #split() method  print(weirdString.upper().split(" "))  #count() method  print(string.count("l"))  #swapcase() method  print (weirdString.swapcase()) |

**Знаходження дискримінанта**

Необхідно створити функцію що знаходить дискримінант.

Хід виконання завдання:

Формула дискримінанта b\*\*2 – 4\*a\*c. Запрошуємо ввід 3-х коефіцієнтів від користувача та передаємо їх функції на обчислення.

Текст програми:

|  |
| --- |
| def dSearching(a, b, c):  res = b\*\*2-4\*a\*c  return res  a = float(input("Enter a: "))  b = float(input("Enter b: "))  c = float(input("Enter c: "))  D = dSearching(a, b, c)  print(D) |

Посилання на github:

<https://github.com/Aatheeriss/TP-KB-221-Denis-Sazonenko.git>

Звіт до Теми №2

Умовні переходи

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

**Пошук коренів квадратного рівняння**

Необхідно дописати умови перевірки дискримінанта та знайти корені рівняння.

Хід виконання завдання:

Створив умови перевірки чи дискримінант більший, дорівнює, або інше (тобто менший) за нуль. Створив функцію для знаходження коренів.

Код програми:

|  |
| --- |
| #calculating D  def dSearching(a, b, c):      res = b\*\*2 - 4\*a\*c      return res  #calculating D  def dSearching(a, b, c):      res = b\*\*2 - 4\*a\*c      return res  #variables input  a = float(input("Enter a: "))  b = float(input("Enter b: "))  c = float(input("Enter c: "))  #D output  D = dSearching(a, b, c)  print("\nD =", D)  #topic\_02  #roots calculating function  def calculatingRoots(b, D, a):      x1 = (-b + D\*\*0.5) / (2\*a)      x2 = (-b - D\*\*0.5) / (2\*a)      if x1 == x2:          return x1      else:          return x1, x2    #D conditions check and roots output  if D > 0:      print("\nEquation has 2 roots: ")      print("x =", calculatingRoots(b, D, a))  elif D == 0:      x = -b / (2\*a)      print("\nEquation has 1 root: ")      print("x =", calculatingRoots(b, D, a))  else:      print("\nEquation has no roots.") |

**Калькулятор з конструкціями if else**

Потрібно створити калькулятор з умовними переходами.

Хід виконання завдання:

Створив калькулятор з 4 діями. Ділення на 0 – виключення.

Код програми:

|  |
| --- |
| a = float(input("Enter the first number: "))  b = float(input("Enter the second number: "))  action = input("Enter action (+, -, \*, /): ").strip()  if action == "+":      print(a + b)  elif action == "-":      print(a - b)  elif action == "\*":      print(a \* b)  elif action == "/":      try:          result = a / b          print(result)      except ZeroDivisionError:          print("Division by zero is not allowed.")  else:      print("Invalid action. Please enter +, -, \*, or /.") |

**Калькулятор з конструкцією match case**

Потрібно створити калькулятор з конструкцією match case.

Хід виконання завдання:

Створив калькулятор з 4 діями. Ділення на 0 – виключення.

Код програми:

|  |
| --- |
| a = float(input("Enter the first number: "))  b = float(input("Enter the second number: "))  action = input("Enter action (+, -, \*, /): ").strip()  match action:      case "+":          print(a + b)      case "-":          print(a - b)      case "\*":          print(a \* b)      case "/":          try:              result = a / b              print(result)          except ZeroDivisionError:              print("Division by zero is not allowed.")      case \_:          print("Invalid action. Please enter +, -, \*, or /.") |