**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.

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

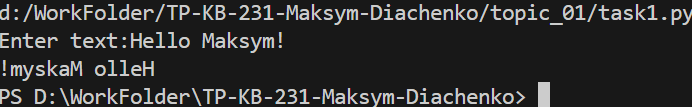
3) Написати функцію пошуку дискримінанту квадратного рівняння.

**Обернення рядка**

Необхідно рядок, наприклад "Hello Maksym!" перетворити наступним чином "!myskaM olleH".

Хід виконання завдання:

У цьому завданні я використав функцію, яка обертає вміст рядка.



Текст програми:

def reverse\_string(text):

    return text[::-1]

text = input("Enter text:")

reversed\_text = reverse\_string(text)

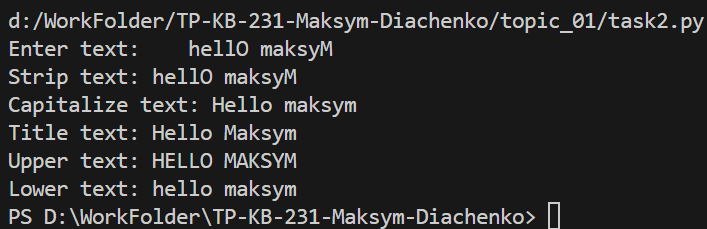
print(reversed\_text)

**Тестування функцій, що працюють з рядками**

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

Хід виконання завдання:

У цьому завданні я використав вище наведені функції.



Функція Strip прибирає пробіли по краям рядка.

Функція Capitalize робить першу букву першого слова в рядку велику, а всі інші маленькими.

Функція Tittle перетворює регістри перших букв усіх слів рядка великими.

Функція Upper перетворює регістри всіх букв усіх слів рядка великими.

Функція Lower перетворює регістри всіх букв усіх слів рядка маленькими.

Текст програми:

text = input("Enter text:")

text1=text.strip()

print("Strip text: " + text1)

text2=text1.capitalize()

print("Capitalize text: " + text2)

text3=text2.title()

print("Title text: " + text3)

text4=text3.upper()

print("Upper text: " + text4)

text5=text4.lower()

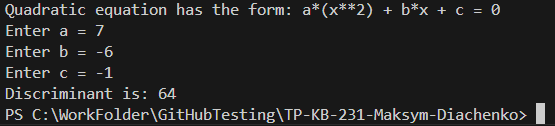
print("Lower text: " + text5)

**Написання функції пошуку дискримінанту квадратного рівняння**

Необхідно написати функцію пошуку дискримінанту квадратного рівняння.

Хід виконання завдання:

У цьому завданні я написав функцію пошуку дискримінанту квадратного рівняння.



Текст програми:

def discr(a, b, c):

    return b\*\*2 - 4\*a\*c

print("Quadratic equation has the form: a\*(x\*\*2) + b\*x + c = 0")

a = int(input("Enter a = "))

b = int(input("Enter b = "))

c = int(input("Enter c = "))

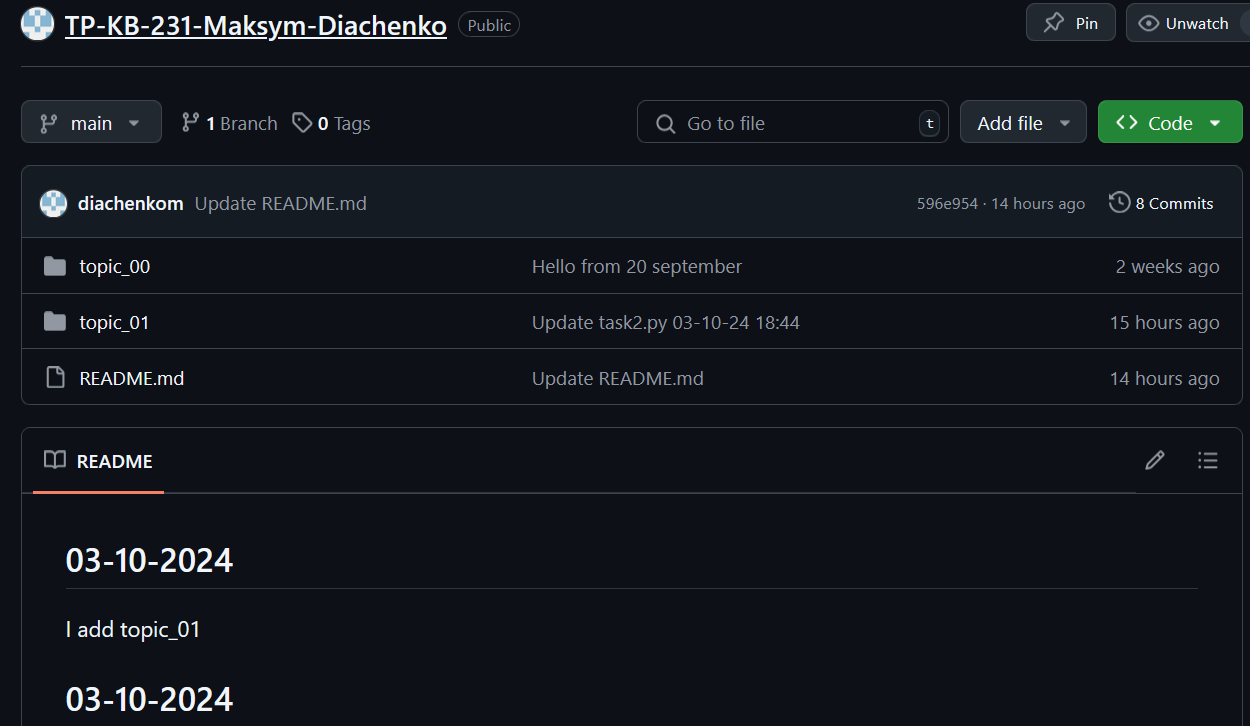
d = discr(a, b, c)

print("Discriminant is: " + str(d))

Посилання на github:

<https://github.com/diachenkom/TP-KB-231-Maksym-Diachenko>

Знімок екрану з посилання на github:



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

Звіт до Теми №2

Умовні переходи

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.

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

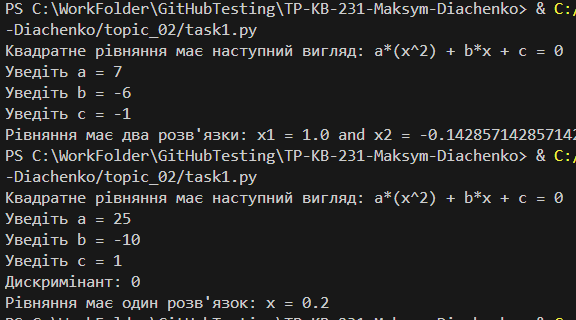
3) Написати програму калькулятор використовуючи match конструкцію. Кожна операція має бути виконана в окремій функції.

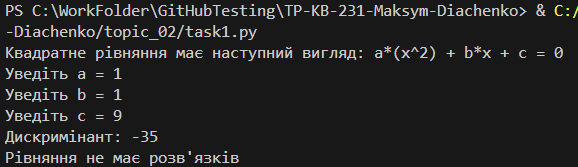
**Функція пошуку коренів квадратного рівняння**

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

Хід виконання завдання:

У цьому завданні я, використовуючи функцію розрахунку дискримінанту з попередньої теми, написав функцію пошуку коренів квадратного рівняння з використанням умовних переходів.





Текст програми:

import math

print("Квадратне рівняння має наступний вигляд: a\*(x^2) + b\*x + c = 0")

a = int(input("Уведіть a = "))

b = int(input("Уведіть b = "))

c = int(input("Уведіть c = "))

def discr(a, b, c):

    return b\*\*2 - 4\*a\*c

def equation\_solution(a, b, c):

    d = discr(a, b, c)

    print("Дискримінант: " + str(d))

    if d > 0:

        x1 = (-(b) + math.sqrt(d)) / (2\*a)

        x2 = (-(b) - math.sqrt(d)) / (2\*a)

        return f"Рівняння має два розв'язки: x1 = {x1} and x2 = {x2}"

    elif d == 0:

        x1 = (-b) / (2\*a)

        return f"Рівняння має один розв'язок: x = {x1}"

    else:

        return "Рівняння не має розв'язків"

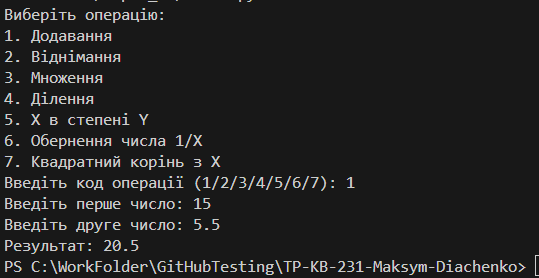
print (equation\_solution(a, b, c))

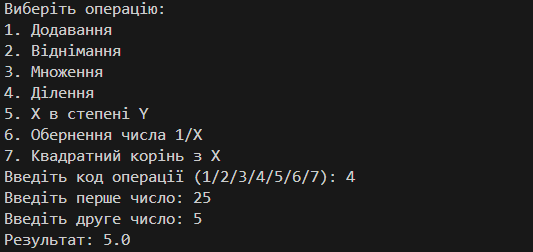
**Программа калькулятор з if else конструкцію**

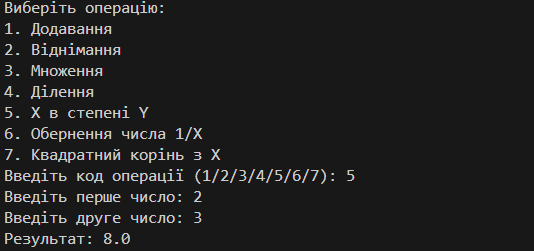
Необхідно написати програму калькулятор використовуючи if else конструкцію.

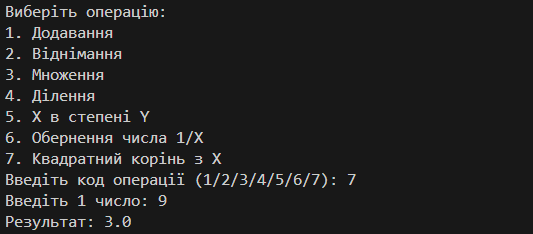
Хід виконання завдання:

У цьому завданні я написав програму калькулятор, що використовує if else конструкції, оформивши кожну операцію в окремій функції.









Текст програми:

# Додавання

def add(a, b):

    return a + b

# Віднімання

def subtract(a, b):

    return a - b

# Множення

def multiply(a, b):

    return a \* b

# Ділення

def divide(a, b):

    if b != 0:

        return a / b

    else:

        return "Помилка: Ділити на нуль не можна!"

# X в степені Y

def step(a,b):

    return a\*\*b

# Основна функція калькулятора

def calculator():

    print("Виберіть операцію:")

    print("1. Додавання")

    print("2. Віднімання")

    print("3. Множення")

    print("4. Ділення")

    print("5. X в степені Y")

    print("6. Обернення числа 1/Х")

    print("7. Квадратний корінь з Х")

    cod = input("Введіть код операції (1/2/3/4/5/6/7): ")

    if (int(cod)<1 or int(cod)>7):

        print("Неправильний ввiд Kоду операції")

        return

    elif (int(cod)>=1 and int(cod)<=5):

        num1 = float(input("Введіть перше число: "))

        num2 = float(input("Введіть друге число: "))

    elif (int(cod)>=6 and int(cod)<=7):

        num1 = float(input("Введіть 1 число: "))

    # Виконання арифметичної операції

    if cod == '1':

        print(f"Результат: {add(num1, num2)}")

    elif cod == '2':

        print(f"Результат: {subtract(num1, num2)}")

    elif cod == '3':

        print(f"Результат: {multiply(num1, num2)}")

    elif cod == '4':

        print(f"Результат: {divide(num1, num2)}")

    elif cod == '5':

        print(f"Результат: {step(num1, num2)}")

    elif cod == '6':

        print(f"Результат: {1/num1}")

    elif cod == '7':

        print(f"Результат: {num1\*\*0.5}")

    else:

        print("Неправильний код операції.")

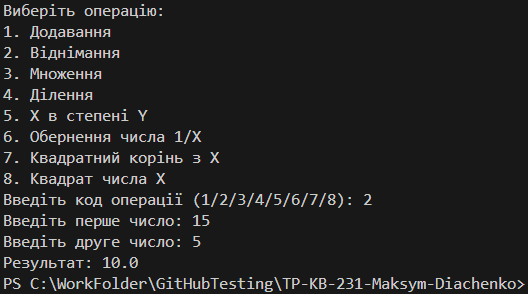
calculator()

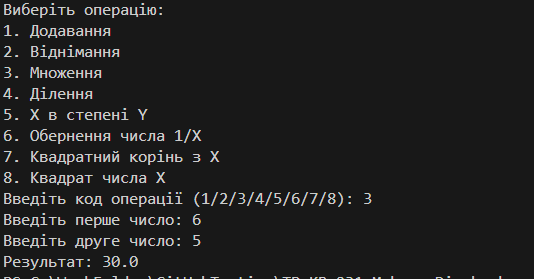
**Програма калькулятор з match конструкцією**

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

Хід виконання завдання:

У цьому завданні я написав програму калькулятор, аналогічно до попередньої. Але в цьому завданні вже з використанням match конструкції, оформивши кожну операцію в окремій функції.





Текст програми:

# Додавання

def add(a, b):

    return a + b

# Віднімання

def subtract(a, b):

    return a - b

# Множення

def multiply(a, b):

    return a \* b

# Ділення

def divide(a, b):

    if b != 0:

        return a / b

    else:

        return "Помилка: Ділити на нуль не можна!"

# X в степені Y

def step(a,b):

    return a\*\*b

def square(a):

    return a\*\*2

# Основна функція калькулятора

def calculator():

    print("Виберіть операцію:")

    print("1. Додавання")

    print("2. Віднімання")

    print("3. Множення")

    print("4. Ділення")

    print("5. X в степені Y")

    print("6. Обернення числа 1/Х")

    print("7. Квадратний корінь з Х")

    print("8. Квадрат числа Х")

    cod = input("Введіть код операції (1/2/3/4/5/6/7/8): ")

    if (int(cod)<1 or int(cod)>8):

        print("Неправильний ввiд Kоду операції")

        return

    elif (int(cod)>=1 and int(cod)<=5):

        num1 = float(input("Введіть перше число: "))

        num2 = float(input("Введіть друге число: "))

    elif (int(cod)>=6 and int(cod)<=8):

        num1 = float(input("Введіть 1 число: "))

    # Виконання арифметичної операції

    match cod:

        case '1':

            print(f"Результат: {add(num1, num2)}")

        case '2':

            print(f"Результат: {subtract(num1, num2)}")

        case '3':

            print(f"Результат: {multiply(num1, num2)}")

        case '4':

            print(f"Результат: {divide(num1, num2)}")

        case '5':

            print(f"Результат: {step(num1, num2)}")

        case '6':

            print(f"Результат: {1/num1}")

        case '7':

            print(f"Результат: {num1\*\*0.5}")

        case '8':

            print(f"Результат: {square(num1)}")

        case \_:

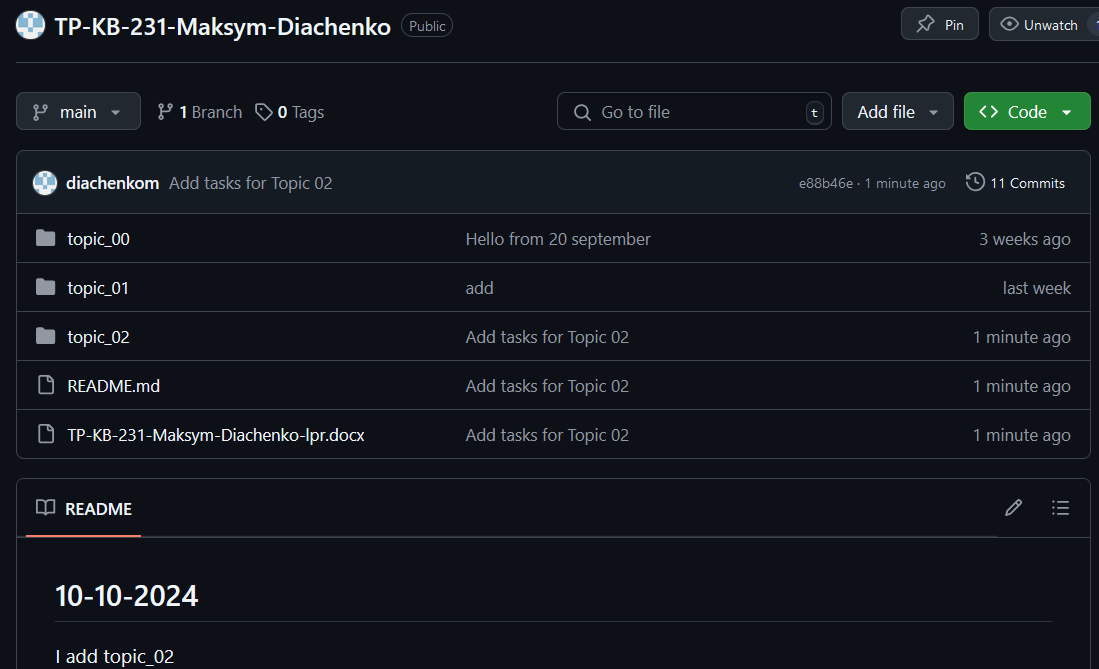
            print("Неправильний код операції.")

calculator()

Посилання на github:

<https://github.com/diachenkom/TP-KB-231-Maksym-Diachenko>

Знімок екрану з посилання на github:



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

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.

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

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

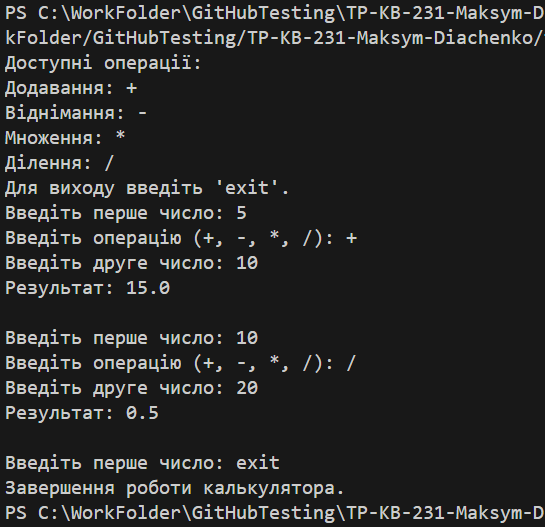
4) Маючи відсортований список, написати функцію пошуку позиції для вставки нового елементу в список.

**Програма калькулятор з постійними запитами на введення нових даних та операцій**

Необхідно написати програму калькулятор з постійними запитами на введення нових даних та операцій.

Хід виконання завдання:

У цьому завданні я, використовуючи цикл while, написав програму калькулятор з постійними запитами на введення нових даних та операцій.



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 "Помилка: ділення на нуль не можливе"

    return x / y

def calculator():

    print("Доступні операції:")

    print("Додавання: +")

    print("Віднімання: -")

    print("Множення: \*")

    print("Ділення: /")

    print("Для виходу введіть 'exit'.")

    while True:

        # Запит першого числа

        num1 = input("Введіть перше число: ")

        if num1.lower() in ['exit']:

            print("Завершення роботи калькулятора.")

            break

        try:

            num1 = float(num1)

        except ValueError:

            print("Некоректне число. Спробуйте ще раз.")

            continue

        # Запит операції

        operation = input("Введіть операцію (+, -, \*, /): ")

        if operation.lower() in ['exit']:

            print("Завершення роботи калькулятора.")

            break

        if operation not in ['+', '-', '\*', '/']:

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

            continue

        # Запит другого числа

        num2 = input("Введіть друге число: ")

        if num2.lower() in ['exit']:

            print("Завершення роботи калькулятора.")

            break

        try:

            num2 = float(num2)

        except ValueError:

            print("Некоректне число. Спробуйте ще раз.")

            continue

        # Виконання обчислення

        if operation == '+':

            result = add(num1, num2)

        elif operation == '-':

            result = subtract(num1, num2)

        elif operation == '\*':

            result = multiply(num1, num2)

        elif operation == '/':

            result = divide(num1, num2)

        print(f"Результат: {result}\n")

calculator()

**Програма тестування функцій списків**

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

Хід виконання завдання:

У цьому завданні я написав програму тестування функцій списків extend(), append(), insert(id, val), remove(val), clear(), sort(), reverse(), copy() та з’ясував призначення кожної із цих функцій.

Функція append() виконує додавання елементу в кінець списку.

Функція extend() виконує додавання всіх елементів одного списку до другого.

Функція remove() виконує видалення заданого елементу списку.

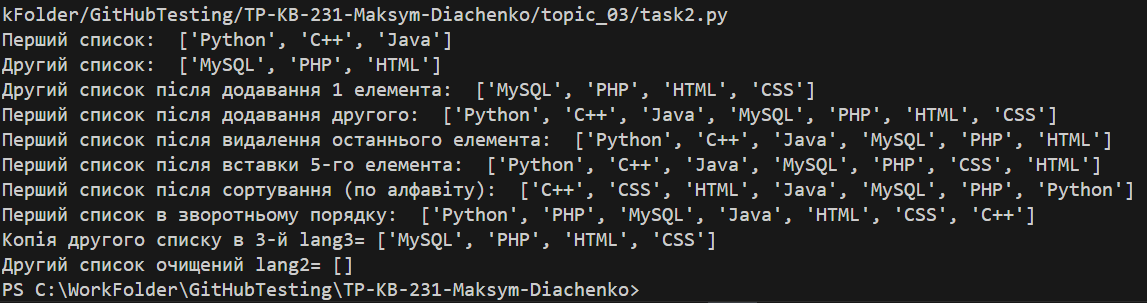
Функція insert() виконує вставку заданого елемента списку у вказану позицію.

Функція sort() виконує сортування всіх елементів списку за алфавітом.

Функція reverse() виконує сортування в зворотньому порядку всіх елементів списку.

Функція copy() робить копію списку.

Функція clear() видаляє всі елементи зі списку.



Текст програми:

lang1 = ["Python", "C++", "Java"]

lang2 = ["MySQL", "PHP", "HTML"]

print("Перший список: ", lang1)

print("Другий список: ", lang2)

# Додавання елементу в кінець списку

lang2.append("CSS")

print("Другий список після додавання 1 елемента: ", lang2)

# Додавання всіх елементів одного списку до другого

lang1.extend(lang2)

print("Перший список після додавання другого: ", lang1)

# Видалення останнього елементу списку

lang1.remove("CSS")

print("Перший список після видалення останнього елемента: ", lang1)

# Вставка передостаннього елементу списку

lang1.insert(5,"CSS")

print("Перший список після вставки 5-го елемента: ", lang1)

# Сортування всіх елементів списку

lang1.sort()

print("Перший список після сортування (по алфавіту): ", lang1)

# Реверс всіх елементів списку

lang1.reverse()

print("Перший список в зворотньому порядку: ", lang1)

# Копія 2-го списку в 3-й список

lang3 = lang2.copy()

print("Копія другого списку в 3-й lang3=", lang3)

# Очищення 2-го списку

lang2.clear()

print("Другий список очищений lang2=", lang2)

**Програма тестування функцій словників**

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

Хід виконання завдання:

У цьому завданні я написав програму тестування функцій словників: update(), del(), clear(), keys(), values(), items() та з’ясував призначення кожної із цих функцій.

Функція update () виконує оновлення елементу словника.

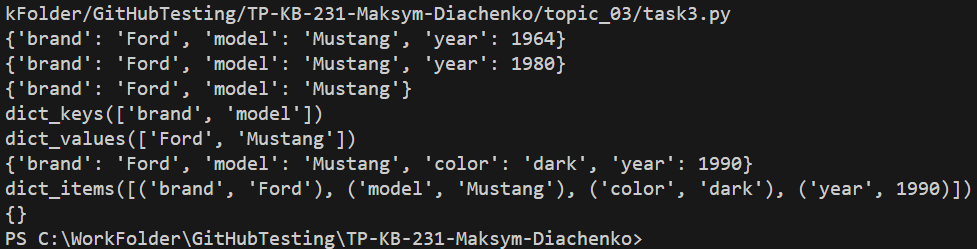
Функція del () виконує видалення елементу словника за вказаним ключем.

Функція clear () виконує очищення всіх елементів словника.

Функція keys () виконує виведення на екран списку усіх ключів словника.

Функція values () виконує виведення на екран списку усіх значень словника.

Функція items () виконує виведення на екран списку усіх елементів словника.



Текст програми:

# Словник для прикладу

my\_dict = {

  "brand": "Ford",

  "model": "Mustang",

  "year": 1964

}

print(my\_dict)

# Оновлення елемента словника

my\_dict.update({"year": 1980})

print(my\_dict)

# Видалення елемента словника по вказаному ключу

del my\_dict["year"]

print(my\_dict)

# Список усіх ключів словника

my\_keys = my\_dict.keys()

print(my\_keys)

# Список усіх значень словника

my\_values = my\_dict.values()

print(my\_values)

# Додати новi елемент до словника

my\_dict["color"] = "dark"

my\_dict["year"] = 1990

print(my\_dict)

# Отримати список усіх елементів словника

my\_items=my\_dict.items()

print(my\_items)

# Очищення усіх елементів словника

my\_dict.clear()

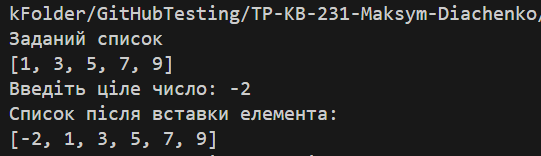
print(my\_dict)

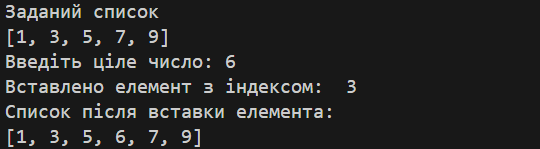
**Функція пошуку позиції для вставки нового елементу в список**

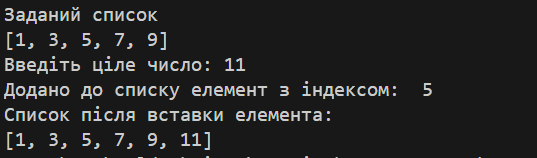
Необхідно маючи відсортований список, написати функцію пошуку позиції для вставки нового елементу в список.

Хід виконання завдання:

У цьому завданні я написав функцію пошуку позиції для вставки нового елементу у відсортований список.







Текст програми:

list1 = [1, 3, 5, 7, 9]

print("Заданий список")

print(list1)

num = int(input("Введіть ціле число: "))

for i in range(0, len(list1)-1):

# Якщо число менше 0-го елементу списку

    if num < list1[i]:

        list1.insert(0, num)

        print("Вставлено елемент з індексом: 0")

        break

# Якщо число рівне одному з елементів

    if num == list1[i]:

        list1.insert(i, num)

        print("Вставлено елемент з індексом: ",i)

        break

# Якщо число в діапазоні між 2-ма сусідніми елем-ми

    if (list1[i] < num) and (num < list1[i+1]):

        list1.insert(i+1, num)

        print("Вставлено елемент з індексом: ",i+1)

        break

else:

    list1.append(num)

    print("Додано до списку елемент з індексом: ",len(list1)-1)

print("Cписок після вставки елемента:")

print(list1)

Посилання на github:

<https://github.com/diachenkom/TP-KB-231-Maksym-Diachenko>

Знімок екрану з посилання на github: