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:

Текст програми:

task1.py

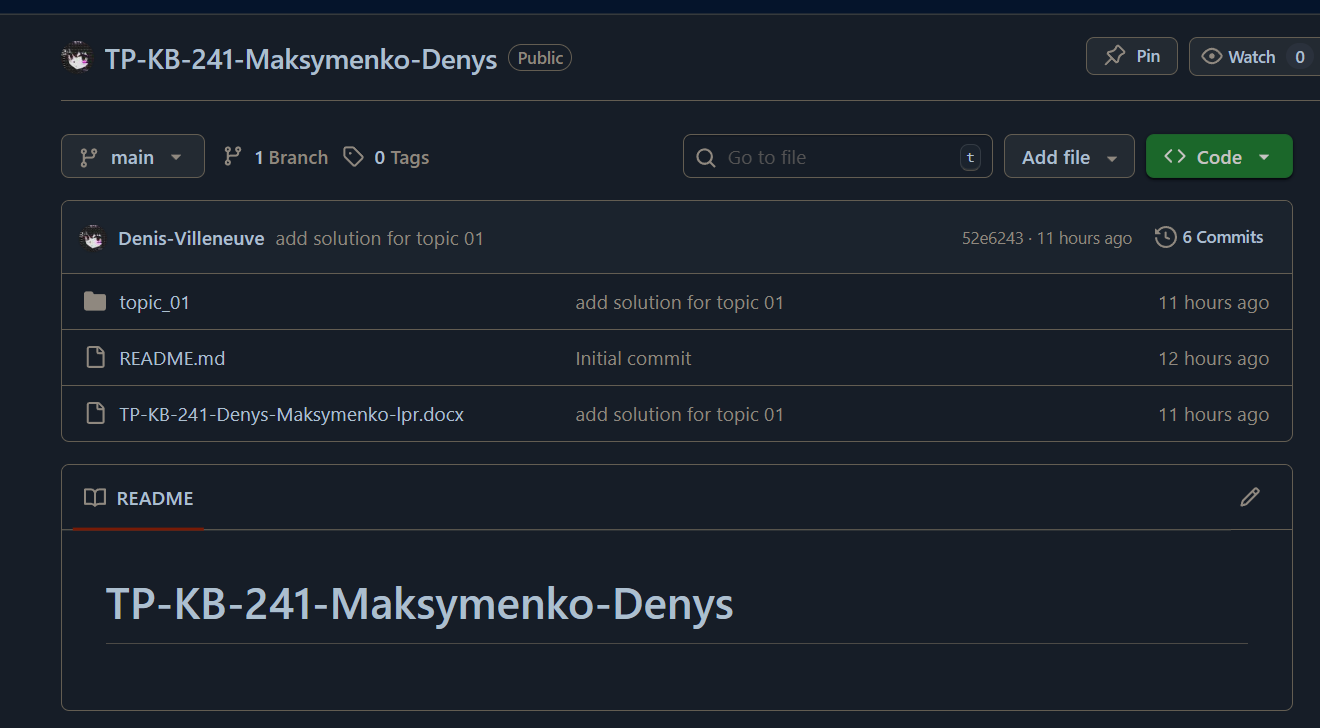
s = "abcdefg123"

s = s[:10].ljust(10)

print(s[::-1])

Посилання на github: https://github.com/Denis-Villeneuve/TP-KB-241-Maksymenko-Denys

Знімок екрану з посилання на github:



**Робота з методами рядків**

Необхідно для заданого рядка " hello world! python is GREAT " виконати перевірку роботи таких методів: strip(), capitalize(), title(), upper(), lower() .

Хід виконання завдання:

Текст програми:

task2.py

s = "   hello world! python is GREAT   "

print(s.strip())

print(s.capitalize())

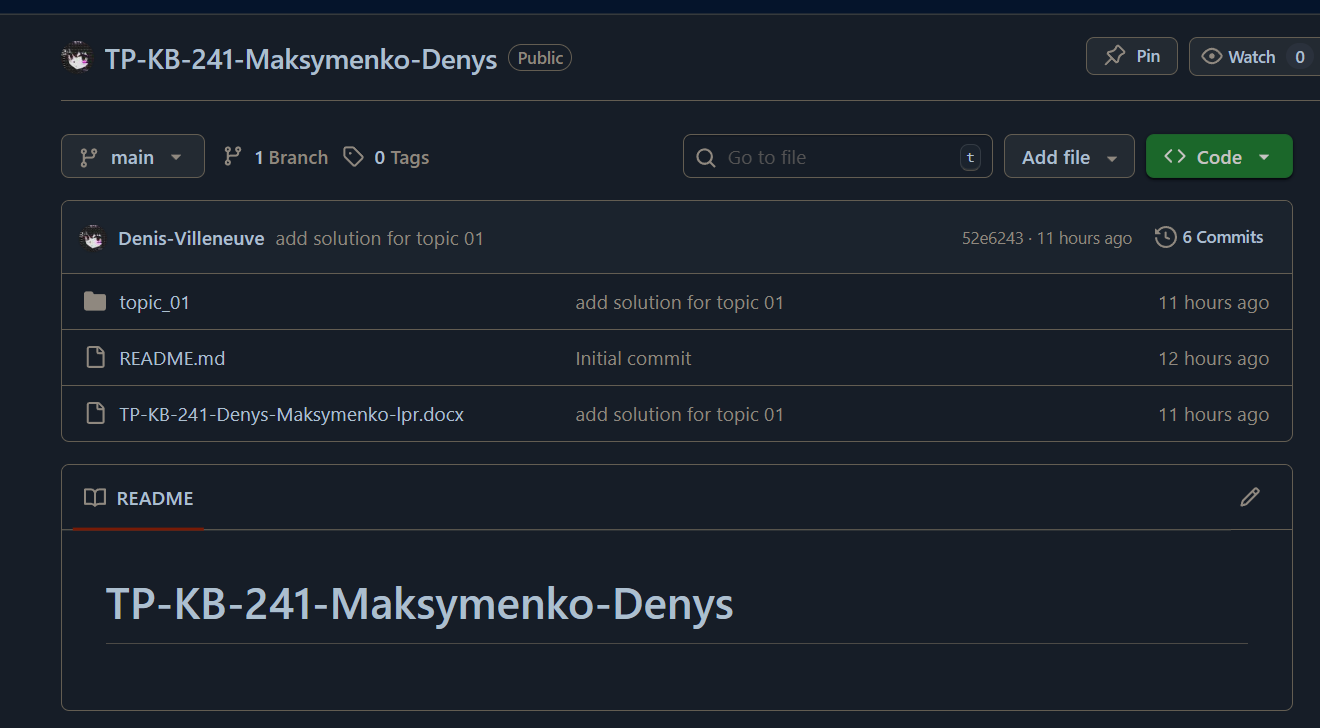
print(s.title())

print(s.upper())

print(s.lower())

Посилання на github: https://github.com/Denis-Villeneuve/TP-KB-241-Maksymenko-Denys

Знімок екрану з посилання на github:



**Розв’язування квадратного рівняння**

Необхідно розробити програму для знаходження коренів квадратного рівняння.

Хід виконання завдання:

Текст програми:

task3.py

import math

a, b, c = 1, -3, 2

d = b\*\*2 - 4\*a\*c

if d < 0:

    print("No real roots")

elif d == 0:

    print(-b / (2\*a))

else:

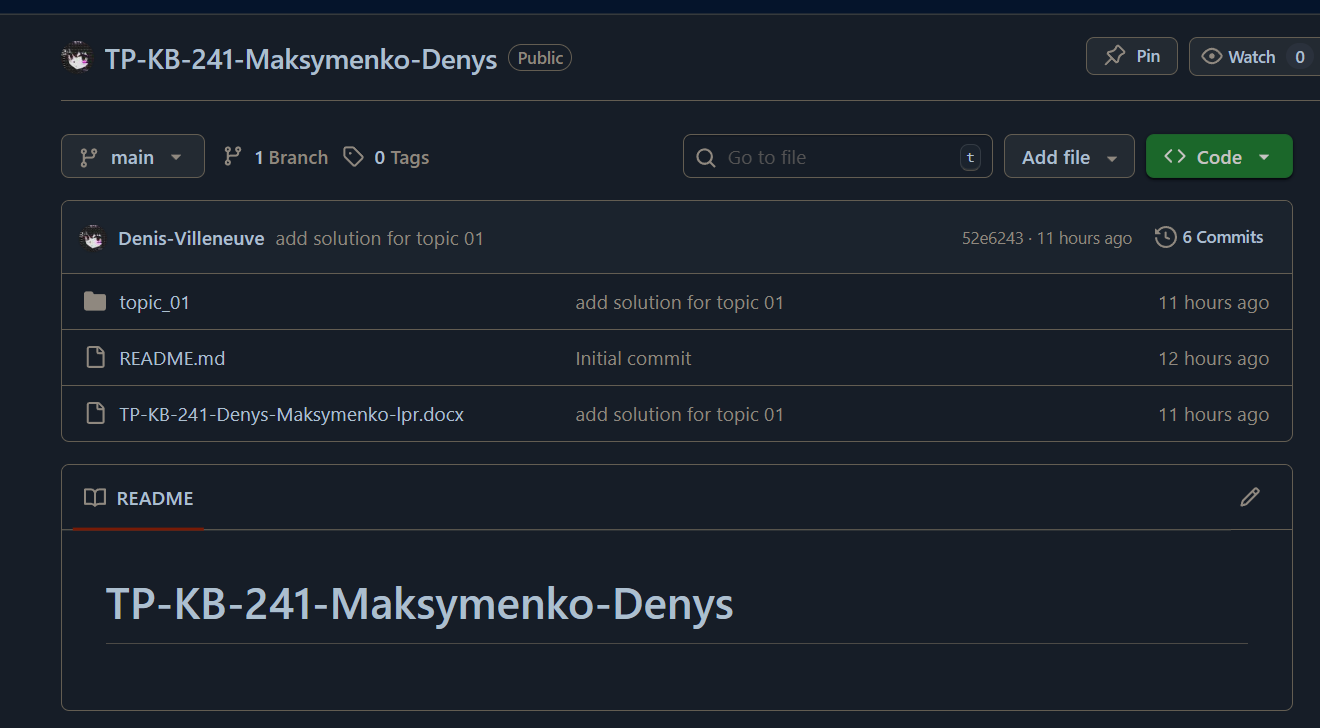
    x1 = (-b + math.sqrt(d)) / (2\*a)

    x2 = (-b - math.sqrt(d)) / (2\*a)

    print(x1, x2)

Посилання на github: https://github.com/Denis-Villeneuve/TP-KB-241-Maksymenko-Denys

Знімок екрану з посилання на github:



Звіт до Теми №2 Функції та змінні

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

Завдання для самостійного виконання:

1. Написати функцію пошуку коренів квадратного рівняння використовуючи функцію розрахунку дискримінанту з попередньої теми та умовні переходи.

Текст програми:

task1.py

import math

def discriminant(a, b, c):

return b\*\*2 - 4\*a\*c

def quadrat(a, b, c):

d = discriminant(a, b, c)

if d > 0:

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

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

return f"Два корені: x1 = {x1}, x2 = {x2}"

elif d == 0:

x = -b / (2\*a)

return f"Один корінь: x = {x}"

else:

return "Коренів немає (дискримінант < 0)"

a = float(input("Введіть a: "))

b = float(input("Введіть b: "))

c = float(input("Введіть c: "))

print(quadrat(a, b, c))

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

Текст програми:

task1.py

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 "Ділення на нуль неможливе"

def calculator\_if(a, b, op):

if op == "+":

return add(a, b)

elif op == "-":

return subtract(a, b)

elif op == "\*":

return multiply(a, b)

elif op == "/":

return divide(a, b)

else:

return "Невідома операція"

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

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

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

print("Результвт:", calculator\_if(a, b, op))

1. Написати програму калькулятор використовуючи **match** конструкцію. Кожна операція має бути виконана в окремій функції.

Текст програми:

task1.py

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 "Ділення на нуль неможливе"

def calculator\_match(a, b, op):

match op:

case "+":

return add(a, b)

case "-":

return subtract(a, b)

case "\*":

return multiply(a, b)

case "/":

return divide(a, b)

case \_:

return "Невідома операція"

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

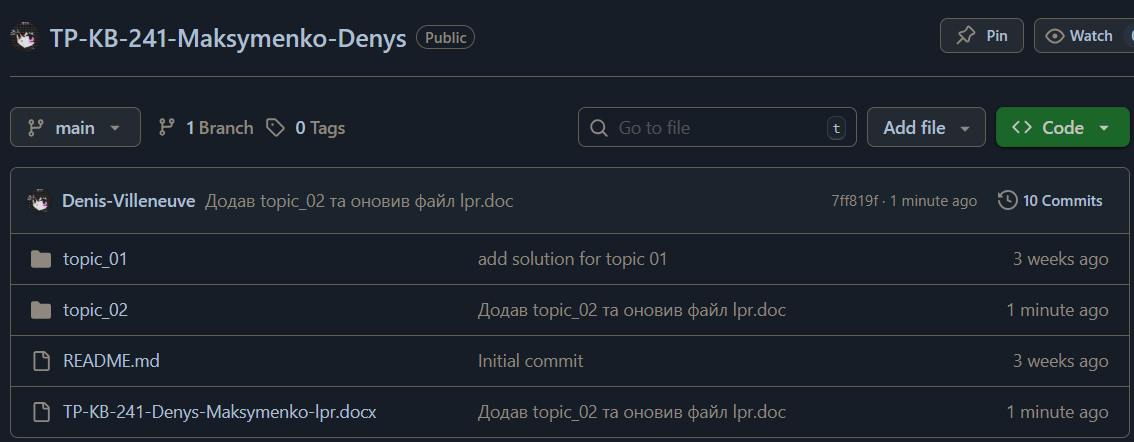
b = float(input("Введіть друге число: "))

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

print("Результат:", calculator\_match(a, b, op))

Посилання на github: https://github.com/Denis-Villeneuve/TP-KB-241-Maksymenko-Denys

Знімок екрану з посилання на github:



Звіт до Теми №3 Функції та змінні

Під час виконання практичного завдання до Теми №3 було надано варіанти рішення до наступних задач:

Завдання для самостійного виконання:

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.

Текст програми:

task1.py

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 "Ділення на нуль неможливе"

def calculator(a, b, op):

if op == "+":

return add(a, b)

elif op == "-":

return subtract(a, b)

elif op == "\*":

return multiply(a, b)

elif op == "/":

return divide(a, b)

else:

return "Невідома операція"

def main():

while True:

command = input("Введіть 'exit' для завершення або Enter для продовження: ")

if command == "exit":

print("Завершення роботи калькулятора.")

break

try:

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

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

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

print("Результат:", calculator(a, b, op))

except ValueError:

print("Введено некоректне число!")

main()

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

Текст програми:

Task2.py

def test\_list\_functions():

lst = [1, 2, 3]

print("Початковий список:", lst)

lst.extend([4, 5])

print("extend:", lst)

lst.append(6)

print("append:", lst)

lst.insert(2, 99)

print("insert(2, 99):", lst)

lst.remove(99)

print("remove(99):", lst)

lst.sort()

print("sort:", lst)

lst.reverse()

print("reverse:", lst)

copy\_lst = lst.copy()

print("copy:", copy\_lst)

lst.clear()

print("clear:", lst)

test\_list\_functions()

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

Текст програми:

Task3.py

def test\_dict\_functions():

d = {'a': 1, 'b': 2}

print("Початковий словник:", d)

d.update({'c': 3})

print("update({'c': 3}):", d)

del d['a']

print("del d['a']:", d)

print("keys():", list(d.keys()))

print("values():", list(d.values()))

print("items():", list(d.items()))

d.clear()

print("clear():", d)

test\_dict\_functions()

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

Текст програми:

Task4.py

def find\_insert\_position(sorted\_list, value):

for i, elem in enumerate(sorted\_list):

if value < elem:

return i

return len(sorted\_list)

sorted\_list = [1, 3, 5, 7, 9]

value = 6

position = find\_insert\_position(sorted\_list, value)

print(f"Позиція для вставки {value}: {position}")