

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

Звіт до Теми №1

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.

Текст програми:

1)

```
pr3 > task1.py > ...
1  def add(x, y):
2      return x + y
3
4  def subtract(x, y):
5      return x - y
6
7  def multiply(x, y):
8      return x * y
9
10 def divide(x, y):
11     if y == 0:
12         return "Ділення на 0!"
13     return x / y
14
15 def calculator(x, y, op):
16     match op:
17         case "+":
18             return add(x, y)
19         case "-":
20             return subtract(x, y)
21         case "*":
22             return multiply(x, y)
23         case "/":
24             return divide(x, y)
25         case _:
26             return "Невідома операція!"
27
28 def res():
29     print("Калькулятор запущено. Введи 'exit' щоб завершити.")
30     while True:
31         a = input("Введи перше число: ")
32         if a.lower() == "exit":
33             print("Вихід з програми.")
34             break
35
36         b = input("Введи друге число: ")
37         if b.lower() == "exit":
38             print("Вихід з програми.")
39             break
40
41         op = input("Введи операцію (+, -, *, /): ")
42         if op.lower() == "exit":
43             print("Вихід з програми.")
44             break
45
46         try:
47             a = float(a)
48             b = float(b)
```

```
45
46         try:
47             a = float(a)
48             b = float(b)
49             result = calculator(a, b, op)
50             print(f"Результат: {result}")
51         except ValueError:
52             print("Помилка!")
53
54     res()
55
```

2)

```
pr3 7 task2.py 7 test_list_functions
1  def test_list_functions():
2      nums = [3, 1, 4]
3      print("Початковий список:", nums)
4
5      nums.append(5)
6      print("append(5):", nums)
7
8      nums.extend([7, 8])
9      print("extend([7,8]):", nums)
10
11     nums.insert(1, 9)
12     print("insert(1,9):", nums)
13
14     nums.remove(4)
15     print("remove(4):", nums)
16
17     copy_nums = nums.copy()
18     print("copy():", copy_nums)
19
20     nums.sort()
21     print("sort():", nums)
22
23     nums.reverse()
24     print("reverse():", nums)
25
26     nums.clear()
27     print("clear():", nums)
28
29     test_list_functions()
30
```

3)

```

1  def test_dict_functions():
2      print("\nТестування функцій словників")
3      student = {"name": "Danylo", "age": 18}
4      print("Початковий словник:", student)
5
6      student.update({"grade": "A"})
7      print("update({'grade': 'A'}):", student)
8
9      del student["age"]
10     print("del age:", student)
11
12     print("keys():", list(student.keys()))
13     print("values():", list(student.values()))
14     print("items():", list(student.items()))
15
16     student.clear()
17     print("clear():", student)
18
19     test_dict_functions()

```

4)

```

1  def find_insert_position(sorted_list, new_element):
2      for i in range(len(sorted_list)):
3          if new_element < sorted_list[i]:
4              return i
5      return len(sorted_list)
6
7  def test_insert_position():
8      nums = [1, 3, 5, 7, 9]
9      print("\nВідсортований список:", nums)
10     x = int(input("Введи новий елемент: "))
11     pos = find_insert_position(nums, x)
12     print(f"Елемент {x} слід вставити на позицію {pos}")
13
14     test_insert_position()
15

```

Посилання на github: <https://github.com/Danylo-Tykhonov/TP-Danylo-Tykhonov-K6-242?tab=readme-ov-file#>

Знімок екрану з посилання на github:

TP-Danylo-Tykhonov-K6-242Public

Pin

Watch0

Fork0

Star0

main1 Branch0 Tags

Go to file

Add file

Code

About

Danylo-Tykhonovupdate README046c530 · now10 Commits

pr1	rename lab1 to pr1	last month
pr2	add solution for pr2	3 weeks ago
pr3	add solution for p3	4 minutes ago
README.md	update README	now
TP-KB-242-Tykhonov-Danylo pr1.pdf	Rename TP-KB-242-Tykhonov-Danylo lab1.pdf to TP-KB-242...	last month
TP-KB-242-Tykhonov-Danylo pr2.pdf	pdf file	3 weeks ago

README

2025-09-17

add folder topic\_01 add solution for task1,2,3

2025-09-28

add folder pr\_02 add solution for task1,2,3

2025-10-16

add folder pr\_03 add solution for task1,2,3,4

TP-Danylo-Tykhonov-K6-242

cd

No description, website, or topics provided.

Readme

Activity

0 stars

0 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Languages

Python100.0%

Suggested workflows

Based on your tech stack

SLSA Generic generator

Configure

Generates SLSA3 provenance for your existing release workflows

Python Package using Anaconda

Configure

Create and test a Python package on multiple Python versions using Anaconda