

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

<https://github.com/Danylo-Tykhonov/TP-Danylo-Tykhonov-K6-242>

Звіт до Теми №1

Функції та змінні

Текст програми:

```
def reverse_string(s: str) -> str:
    return s[::-1]

print(reverse_string("Hello!"))
```

1)

```
peremenaya = "  hello world  "

print("strip():", peremenaya.strip())
print("capitalize():", peremenaya.capitalize())
print("title():", peremenaya.title())
print("upper():", peremenaya.upper())
print("lower():", peremenaya.lower())
```

2)

```
def discriminant(a: float, b: float, c: float) -> float:
    return b**2 - 4*a*c

print(discriminant(1, -3, 2))
```

3)

Звіт до Теми №2

Умовний перехід

```

1  import math
2
3  def discriminant(a: float, b: float, c: float) -> float:
4      return b**2 - 4*a*c
5
6  def korni(a: float, b: float, c: float):
7      D = discriminant(a, b, c)
8      if D > 0:
9          x1 = (-b + math.sqrt(D)) / (2*a)
10         x2 = (-b - math.sqrt(D)) / (2*a)
11         return x1, x2
12     elif D == 0:
13         x = -b / (2*a)
14         return x,
15     else:
16         return None
17
18 a, b, c = 1, -3, 2
19 roots = korni(a, b, c)
20 print("Корні рівняння:", roots)
21

```

```

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 "Ділення на 0"

def calculator(x, y, op):
    if op == "+":
        return add(x, y)
    elif op == "-":
        return subtract(x, y)
    elif op == "*":
        return multiply(x, y)
    elif op == "/":
        return divide(x, y)

print(calculator(5, 2, "+"))
print(calculator(5, 2, "/"))
print(calculator(5, 2, "-"))
print(calculator(5, 2, "*"))

```

2)

```
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 "Неизвестная операция"
27
28 print(calculator(5, 2, "*"))
29 print(calculator(5, 0, "/"))
30 print(calculator(5, 2, "+"))
31 print(calculator(5, 2, "-"))
32
```

3)

Звіт до Теми №3

Цикли

```

pr3 > taskipy > -
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 "Деление 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 "Неизвестная операция!"
27
28 - def res():
29     | print("Калькулятор запущен. Введите 'exit' для завершения.")
30     | while True:
31     |     a = input("Введите первое число: ")
32     |     if a.lower() == "exit":
33     |         print("Выход из программы.")
34     |         break
35
36     |     b = input("Введите второе число: ")
37     |     if b.lower() == "exit":
38     |         print("Выход из программы.")
39     |         break
40
41     |     op = input("Введите операцию (+, -, *, /): ")
42     |     if op.lower() == "exit":
43     |         print("Выход из программы.")
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

```

1)

2)

```

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

```

Звіт до Теми №4
Виняткові ситуації


```

pr4 > 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     try:
12         result = x / y
13         return result
14     except ZeroDivisionError:
15         return "Помилка: ділення на нуль!"
16
17 def calculator(x, y, op):
18     try:
19         match op:
20             case "+":
21                 return add(x, y)
22             case "-":
23                 return subtract(x, y)
24             case "*":
25                 return multiply(x, y)
26             case "/":
27                 return divide(x, y)
28             case _:
29                 return "Невідома операція!"
30     except Exception as e:
31         return f"Сталася помилка: {e}"
32
33 def run_calculator():
34     print("Калькулятор запущено. Введи 'exit' щоб завершити.")
35     while True:
36         try:
37             a = input("Введи перше число: ")
38             if a.lower() == "exit":
39                 print("Вихід з програми.")
40                 break
41
42             b = input("Введи друге число: ")
43             if b.lower() == "exit":
44                 print("Вихід з програми.")
45                 break
46
47             op = input("Введи операцію (+, -, *, /): ")
48             if op.lower() == "exit":

```

1)

```

47         op = input("Введи операцію (+, -, *, /): ")
48         if op.lower() == "exit":
49             print("Вихід з програми.")
50             break
51
52         a = float(a)
53         b = float(b)
54
55         result = calculator(a, b, op)
56         print(f"Результат: {result}")
57
58     except ValueError:
59         print("Помилка: введено не число!")
60     except Exception as e:
61         print(f"Невідома помилка: {e}")
62
63 run_calculator()
64

```

Звіт до Теми №5

Бібліотеки

1)

```

import random

def game():
    options = ["rock", "scissor", "paper"]
    user_choice = input("Введіть свій вибір (rock, scissor, paper): ").lower()

    if user_choice not in options:
        print("Невірне значення")
        return

    computer_choice = random.choice(options)
    print(f"Комп'ютер обрав: {computer_choice}")

    if user_choice == computer_choice:
        print("Нічия")
    elif (user_choice == "stone" and computer_choice == "scissor") or \
         (user_choice == "scissor" and computer_choice == "paper") or \
         (user_choice == "paper" and computer_choice == "rock"):
        print("Ви перемогли")
    else:
        print("Комп'ютер переміг")

if __name__ == "__main__":
    game()

```

2)


```

import requests

def get_exchange_rate(currency_code):
    url = f"https://bank.gov.ua/NBUStatService/v1/statdirectory/exchange?valcode={currency_code}&json"
    response = requests.get(url)
    if response.status_code == 200:
        data = response.json()
        if data:
            return data[0]['rate']
    print("Не вдалося отримати курс валют")
    return None

def convert():
    supported = ["EUR", "USD", "PLN"]
    currency = input("Введіть валюту (EUR, USD, PLN): ").upper()

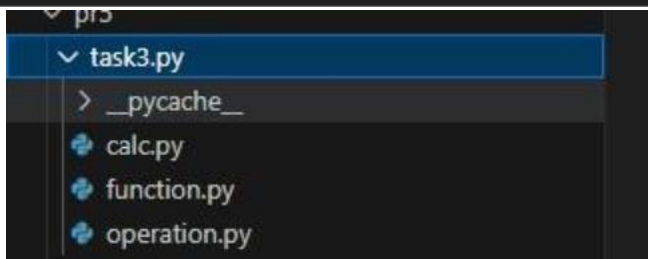
    if currency not in supported:
        print("Ця валюта не підтримується")
        return

    try:
        amount = float(input(f"Введіть кількість {currency}: "))
    except ValueError:
        print("Невірне число")
        return

    rate = get_exchange_rate(currency)
    if rate:
        uah = amount * rate
        print(f"{amount} {currency} = {uah:.2f} UAH за курсом {rate:.2f}")

if __name__ == "__main__":
    convert()

```



3)

```
or5 > task3.py > calc.py
1  from operation import perform_operation
2
3  if __name__ == "__main__":
4      perform_operation()
5
```

```
r5 > task3.py > function.py > divide
1  def add(a, b):
2      return a + b
3
4  def subtract(a, b):
5      return a - b
6
7  def multiply(a, b):
8      return a * b
9
10 def divide(a, b):
11     if b == 0:
12         return "Ділення на нуль неможливе"
13     return a / b
14
```

```
5 > task3.py > operation.py > ...
1  from function import add, subtract, multiply, divide
2
3  def numbers():
4      a = float(input("Введіть перше число: "))
5      b = float(input("Введіть друге число: "))
6      return a, b
7
8  def perform_operation():
9      print("Оберіть операцію: +, -, *, /")
10     operation = input("Операція: ")
11
12     a, b = numbers()
13
14     if operation == '+':
15         print(f"Результат: {add(a, b)}")
16     elif operation == '-':
17         print(f"Результат: {subtract(a, b)}")
18     elif operation == '*':
19         print(f"Результат: {multiply(a, b)}")
20     elif operation == '/':
21         print(f"Результат: {divide(a, b)}")
22     else:
23         print("Невідома операція")
24
```

Звіт до Теми №6

Робота з файлами

1)

The image shows two screenshots of the Visual Studio Code editor. The top screenshot displays the `calc.py` file, which imports functions from `function.py` and `operation.py`, and uses `logger.py` for logging. The bottom screenshot displays the `function.py` file, which defines the `add`, `subtract`, `multiply`, and `divide` functions. The Explorer sidebar on the left of both screenshots shows a project structure with folders `pr3`, `pr4`, `pr5`, and `pr6`, each containing `task1.py`, `task2.py`, and `task3.py` files. The `pr6` folder also contains `task4.py`, `operation.py`, `logger.py`, `log.txt`, and `__pycache__`. The `calc.py` file is highlighted in the top screenshot, and the `function.py` file is highlighted in the bottom screenshot.

```
1 from function import add, subtract, multiply, divide
2 from operation import get_numbers, get_operation
3 from logger import write_log
4
5 def main():
6     a, b = get_numbers()
7     op = get_operation()
8
9     if op == '+':
10         result = add(a, b)
11     elif op == '-':
12         result = subtract(a, b)
13     elif op == '*':
14         result = multiply(a, b)
15     elif op == '/':
16         result = divide(a, b)
17     else:
18         print("Невідома операція")
19         return
20
21     write_log(a, b, op, result)
22     print(result)
23
24
25 if __name__ == "__main__":
26     main()
27
```

```
1 def add(a, b):
2     return a + b
3
4 def subtract(a, b):
5     return a - b
6
7 def multiply(a, b):
8     return a * b
9
10 def divide(a, b):
11     if b == 0:
12         return "Ділення на нуль неможливе"
13     return a / b
14
```

EXPLORER

TP-DANYLO-TYKHONOV-K6-242

- pr3
 - task2.py
 - task3.py
 - task4.py
- pr4
 - task1.py
- pr5
 - task3.py
 - task1.py
 - task2.py
- pr6
 - task1
 - __pycache__
 - calc.py
 - function.py
 - log.txt
 - logger.py
 - operation.py
 - task2.py

pr6 > task1 > logger.py > write_log

```
1 from datetime import datetime
2
3 def write_log(a, b, oper, result):
4     with open("log.txt", "a", encoding="utf-8") as file:
5         time = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
6         file.write(f"{time} | {a} {oper} {b} = {result}\n")
```

EXPLORER

TP-DANYLO-TYKHONOV-K6-242

- pr3
 - task2.py
 - task3.py
 - task4.py
- pr4
 - task1.py
- pr5
 - task3.py
 - task1.py
 - task2.py
- pr6
 - task1
 - __pycache__
 - calc.py
 - function.py
 - log.txt
 - logger.py
 - operation.py
 - task2.py

pr6 > task1 > operation.py > ...

```
1 def get_numbers():
2     a = float(input("Введіть перше число: "))
3     b = float(input("Введіть друге число: "))
4     return a, b
5
6 def get_operation():
7     return input("Оберіть операцію (+, -, *, /): ")
8
```

The image shows a VS Code editor window with a Python script named `task2.py` open. The script defines a list of student dictionaries, `students_data`, and a function `sort_dicts` that sorts this list based on a user-provided column name. The script also includes a `while` loop to prompt the user for the column name.

```
pr6 > task2.py > ...
1  students_data = [
2      {'name': 'Daria', 'grade': 73},
3      {'name': 'Ivan', 'grade': 89},
4      {'name': 'Svitlana', 'grade': 94},
5      {'name': 'Taras', 'grade': 58},
6      {'name': 'Roman', 'grade': 81},
7      {'name': 'Yulia', 'grade': 100},
8      {'name': 'Bohdan', 'grade': 66},
9      {'name': 'Iryna', 'grade': 77},
10     {'name': 'Serhii', 'grade': 91},
11     {'name': 'Nadia', 'grade': 84}
12 ]
13
14 while True:
15     c = input("Select the column by which you want to sort information about\n")
16     if c == "name" or c == "grade":
17         break
18     else:
19         print("\nYou entered something wrong in the name of the column to be sorted\n")
20
21 def sort_dicts(list, name_col):
22     sorted_dict = sorted(list, key=lambda dict: dict[name_col])
23     if name_col == "grade":
24         print("\nSorted by grade\n")
25         sorted_dict.reverse()
26         for col in sorted_dict:
27             print(f'{col["name"]} {col["grade"]}')
28     else:
29         print("\nSorted by name\n")
30         for col in sorted_dict:
31             print(f'{col["name"]} {col["grade"]}')
32
33 sort_dicts(students_data, c)
```

The Explorer sidebar on the left shows the project structure, including folders `pr3`, `pr4`, `pr5`, and `pr6`, and files like `task1.py`, `task2.py`, `task3.py`, `task4.py`, `operation.py`, `calc.py`, `function.py`, `log.txt`, and `logger.py`. The OPEN EDITORS sidebar shows that `task2.py` is the active file.

Звіт до Теми №7

Об'єктно-орієнтоване програмування

1)

```

1  class Student:
2      def __init__(self, name, age):
3          self.name = name
4          self.age = age
5
6      def __str__(self):
7          return f'{self.name} {self.age}'
8
9
10 class StudentData:
11     def __init__(self, students):
12         self.students = students
13
14     def sort_students(self, column):
15         if column == "name":
16             sorted_students = sorted(self.students, key=lambda student: student.name)
17             print("\nSorted by name:\n")
18
19             elif column == "age":
20                 sorted_students = sorted(self.students, key=lambda student: student.age)
21                 print("\nSorted by age:\n")
22
23             else:
24                 print("\nIncorrect column. Try again\n")
25                 return
26
27             for student in sorted_students:
28                 print(student)
29
30
31
32 students = [
33     Student('Daria', 18),
34     Student('Oleh', 19),
35     Student('Sofia', 20),
36     Student('Mark', 22),
37     Student('Iryna', 21),
38     Student('Tim', 23),
39     Student('Helena', 19),
40     Student('Leon', 24),
41     Student('Karina', 20),

```

```
32  ▾ students = [  
33      Student('Daria', 18),  
34      Student('Oleh', 19),  
35      Student('Sofia', 20),  
36      Student('Mark', 22),  
37      Student('Iryna', 21),  
38      Student('Tim', 23),  
39      Student('Helena', 19),  
40      Student('Leon', 24),  
41      Student('Karina', 20),  
42      Student('Robert', 26)  
43  ]  
44  
45  ▾ while True:  
46      column = input("Choose the column (name, age): ")  
47  ▾ if column in ("name", "age"):  
48      |     break  
49  ▾ else:  
50      |     print("\nSomething went wrong\n")  
51  
52  student_data = StudentData(students)  
53  student_data.sort_students[column]
```

2)

```
ask3.py pr7 U × calc.py ...\task4 U × function.py ...\task4 U operation.py ...\task4 U . ▶ 🔍 📄 ...

pr7 > task4 > calc.py > ...
1  import operation
2  import function
3  print("Calculator Program")
4  print("Enter 'q' at any time to leave\n")
5
6  while True:
7      a = operation.getNumber("Enter the first number: \n")
8      if a == 'Q' or a == 'q':
9          operation.log('q')
10         print("You left the program")
11         break
12
13     b = operation.getNumber("Enter the second number: \n")
14     if b == 'Q' or b == 'q':
15         operation.log('q')
16         print("You left the program")
17         break
18
19     op = operation.getOperations("Choose an operation (+, -, *, /): \n")
20     if op == 'Q' or op == 'q':
21         operation.log('q')
22         print("You left the program")
23         break
24
25     if op == '+':
26         result = function.plus(a, b)
27         operation.log(a, b, op, result)
28         print(f"Answer: {result}")
29     elif op == '-':
30         result = function.minus(a, b)
31         operation.log(a, b, op, result)
32         print(f"Answer: {result}")
33     elif op == '*':
34         result = function.multiply(a, b)
35         operation.log(a, b, op, result)
36         print(f"Answer: {result}")
37     elif op == '/':
38         result = function.divide(a, b)
39         operation.log(a, b, op, result)
40         print(f"Answer: {result}")
41
42     * / + - * % % %
```

```
task3.py pr7 U  calc.py ...\task4 U  function.py ...\task4 U X  operation.py ...\task4 U  . ▶ 🔍 📄 ...
pr7 > task4 > function.py > divide
1  def plus(a: int | float, b: int | float):
2      return float(a) + float(b)
3
4  def minus(a: int | float, b: int | float):
5      return float(a) - float(b)
6
7  def multiply(a: int | float, b: int | float):
8      return float(a) * float(b)
9
10 def divide(a: int | float, b: int | float):
11     try:
12         return float(a) / float(b)
13     except ZeroDivisionError:
14         return "Error: Division by zero is not allowed."
```



```
ask3.py pr7 U  calc.py ...\task4 U  function.py ...\task4 U  operation.py ...\task4 U  ×  .  ▷  🔍  □  ...

pr7 > task4 > operation.py > getNumber
1  import datetime
2
3  def getNumber(prompt):
4      while True:
5          number = input(prompt)
6          if number == 'Q' or number == 'q':
7              return 'q'
8          else:
9              try:
10                 return float(number)
11             except ValueError:
12                 print("You entered something wrong! Please try again!")
13                 log("err_num")
14
15  def getOperations(prompt):
16      while True:
17          op = input(prompt)
18          if op == 'Q' or op == 'q' or op == '+' or op == '-' or op == '*' or op == '/':
19              return op
20          else:
21              print("Error!!! You entered something wrong! No operation can be done")
22              log("err_op")
23
24  def log(a=None, b=None, op=None, result=None):
25      time = datetime.datetime.now().strftime('%d.%m.%Y %H:%M:%S')
26
27      if a == "err_num":
28          logging = f"\n{time} User entered letters or other characters except numbers"
29          with open('calc.log', 'a') as logs:
30              logs.write(logging)
31
32      elif a == "err_op":
33          logging = f"\n{time} User entered some characters that are different from operators"
34          with open('calc.log', 'a') as logs:
35              logs.write(logging)
36
37      elif a == "q" or a == "Q":
38          logging = f"\n{time} User left the program!"
39          with open('calc.log', 'a') as logs:
40              logs.write(logging)
41
```

```
42  ✓ elif a == "0":  
43      logging = f"\n{time}  User left the program!"  
44  ✓      with open('calc.log', 'a') as logs:  
45          logs.write(logging)  
46  
47  ✓ else:  
48      logging = f"\n{time}  User entered value of a = {a} and value of b = {b}"  
49  ✓      with open('calc.log', 'a') as logs:  
50          logs.write(logging)
```