

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
1) ...
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, "-"))
```

3)

Звіт до Теми №3

Цикли

```

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 "Недостаточно операндов"
27
28 def res():
29     print("Калкулятор запущен. Введіть 'exit' для завершення.")
30     while True:
31         a = input("Введіть перше число: ")
32         if a.lower() == "exit":
33             print("Вихід з програми.")
34             break
35
36         b = input("Введіть друге число: ")
37         if b.lower() == "exit":
38             print("Вихід з програми.")
39             break
40
41         op = input("Введіть операцію: (+, -, *, /): ")
42         if op.lower() == "exit":
43             print("Вихід з програми.")
44             break
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)

```
prj > task2.py > 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 v 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

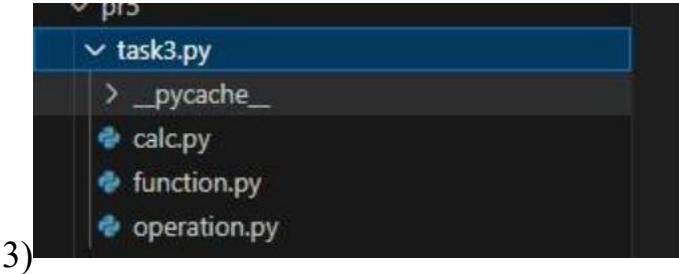
def convert():
    supported = ["EUR", "USD", "PLN"]
    currency = input("Введіть валюту (EUR, USD, PLN): ").upper()

    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)

```
r5 > 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
```

```
> 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 screenshot displays two instances of the Visual Studio Code (VS Code) interface, likely from a split screen or a comparison view.

Top Editor: The title bar says "calc.py ...\\task1". The code implements a simple calculator:

```
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()
```

Bottom Editor: The title bar says "function.py ...\\task1". The code defines basic arithmetic functions:

```
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
```

The left sidebar shows a file tree for a project named "TP-DANYLO-TYKHONOV-K6-242". Both panes show "OPEN EDITORS 2 unsaved" in their respective title bars.

The screenshot shows a code editor interface with two tabs open, each displaying a Python script. The left tab contains the `logger.py` script, and the right tab contains the `operation.py` script. Both scripts are part of a project structure under `TP-DANYLO-TYKHONOV-K6-242`.

logger.py (Top Tab)

```
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")
```

operation.py (Bottom Tab)

```
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 (+, -, *, /): ")
```

The screenshot shows a code editor interface with a dark theme. On the left is the Explorer sidebar displaying a project structure with several folders (pr3, pr4, pr5, pr6) and files (task1.py, task2.py, task3.py, calc.py, function.py, operation.py, log.txt, logger.py, README.md). In the center, the code editor displays the content of `task2.py`. The code defines a list of student data and a function to sort it by name or grade.

```
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': 'Serhi', '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")
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")
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)
```

