Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №6

з дисципліни
«Об’єктно-орієнтоване програмування»

Виконала:

студентка групи ІМ-43
Хубеджева Єлизавета Павлівна
номер у списку групи: 26

Перевірив:

Порєв В. М.

Київ 2025

Завдання

Так як варіант 26, то Ж mod 4 = 2:

- Lab6:

1. Користувач вводить значення n, Min, Max у діалоговому вікні.

2. Програма викликає програми Object2, 3 і забезпечує обмін повідомленнями для передавання та отримання інформації.

- Object2:

1. Створює вектор n дробових (double) чисел у діапазоні Min – Max

2. Показує числові значення у декількох стовпчиках та рядках у власному головному вікні

3. Записує дані в Clipboard Windows у текстовому форматі

- Object3:

1. Зчитує дані з Clipboard Windows

2. Виконує сортування масиву чисел і відображає його у декількох стовпчиках та рядках у власному головному вікні

Тексти програм:

Lab6.py:

```
import tkinter as tk
from tkinter import messagebox
import subprocess
import sys
import os
import json
import time

proc_obj2 = None
proc_obj3 = None
script_dir = os.path.dirname(os.path.abspath(__file__))
PARAMS_FILE = "params.json"
COMMAND_FILE = "command.txt"

def ensure_files_exist():
    if not os.path.exists(PARAMS_FILE):
        with open(PARAMS_FILE, "w") as f: f.write("{}")
    if not os.path.exists(COMMAND_FILE):
        with open(COMMAND_FILE, "w") as f: f.write("IDLE")

def is_running(proc):
    if proc is None:
        return False
    if proc.poll() is not None:
        return False
    return True

def start_automation(n_val, min_val, max_val):
    global proc_obj2

    print(f"--- Початок роботи: n={n_val}, range=[{min_val}, {max_val}]
---")

    data = {"n": n_val, "min": min_val, "max": max_val}

    if not is_running(proc_obj2):
        print("Запуск Object2...")
```

```
        proc_obj2 = subprocess.Popen([sys.executable,
os.path.join(script_dir, 'Object2.py')])
        time.sleep(1)
    else:
        print("Object2 вже активний.")

    try:
        with open(PARAMS_FILE, "w") as f:
            json.dump(data, f)

        with open(COMMAND_FILE, "w") as f:
            f.write("START_OBJ2")

        print("Команду START_OBJ2 надіслано.")

        root.after(500, check_obj2_finished)

    except Exception as e:
        messagebox.showerror("Помилка", f"Файлова помилка: {e}")

def check_obj2_finished():
    try:
        with open(COMMAND_FILE, "r") as f:
            status = f.read().strip()

        if status == "DONE_OBJ2":
            print("Object2 завершив генерацію.")
            run_object3_scenario()
        else:
            root.after(500, check_obj2_finished)
    except:
        root.after(500, check_obj2_finished)

def run_object3_scenario():
    global proc_obj3

    if not is_running(proc_obj3):
        print("Запуск Object3...")
        proc_obj3 = subprocess.Popen([sys.executable,
os.path.join(script_dir, 'Object3.py')])
        time.sleep(1)
```

```
    else:
        print("Object3 вже активний.")

    with open(COMMAND_FILE, "w") as f:
        f.write("START_OBJ3")

    print("Команду START_OBJ3 надіслано.")
    root.after(500, check_obj3_finished)

def check_obj3_finished():
    try:
        with open(COMMAND_FILE, "r") as f:
            status = f.read().strip()

        if status == "DONE_OBJ3":
            print("Object3 завершив роботу.")
            with open(COMMAND_FILE, "w") as f: f.write("IDLE")
        else:
            root.after(500, check_obj3_finished)
    except:
        pass

def open_input_dialog():
    dialog = tk.Toplevel(root)
    dialog.title("Введення параметрів")
    dialog.geometry("300x200")
    dialog.grab_set()

    tk.Label(dialog, text="Введіть параметри:", font=("Arial", 10,
"bold")).pack(pady=10)

    frame_d = tk.Frame(dialog)
    frame_d.pack(pady=5)

    tk.Label(frame_d, text="n:").grid(row=0, column=0, padx=5, pady=5)
    d_entry_n = tk.Entry(frame_d, width=10)
    d_entry_n.grid(row=0, column=1)
    d_entry_n.insert(0, "10")

    tk.Label(frame_d, text="Min:").grid(row=1, column=0, padx=5, pady=5)
```

```
    d_entry_min = tk.Entry(frame_d, width=10)
    d_entry_min.grid(row=1, column=1)
    d_entry_min.insert(0, "1.0")

    tk.Label(frame_d, text="Max:").grid(row=2, column=0, padx=5, pady=5)
    d_entry_max = tk.Entry(frame_d, width=10)
    d_entry_max.grid(row=2, column=1)
    d_entry_max.insert(0, "100.0")

    def on_submit():
        try:
            n = int(d_entry_n.get())
            mn = float(d_entry_min.get())
            mx = float(d_entry_max.get())
            dialog.destroy()
            start_automation(n, mn, mx)
        except ValueError:
            messagebox.showerror("Помилка", "Введіть коректні числа!")

    tk.Button(dialog, text="Виконати", command=on_submit, bg="lightblue",
width=15).pack(pady=15)

def on_closing():
    global proc_obj2, proc_obj3
    if is_running(proc_obj2): proc_obj2.terminate()
    if is_running(proc_obj3): proc_obj3.terminate()
    root.destroy()

ensure_files_exist()
root = tk.Tk()
root.title("Lab6")
root.geometry("500x300+50+50")

main_menu = tk.Menu(root)
root.config(menu=main_menu)

file_menu = tk.Menu(main_menu, tearoff=0)
main_menu.add_cascade(label="Файл", menu=file_menu)
file_menu.add_command(label="Вихід", command=on_closing)
```

```
main_menu.add_command(label="Вектор чисел", command=open_input_dialog)

root.protocol("WM_DELETE_WINDOW", on_closing)
root.mainloop()
```

Object2.py:

```
import tkinter as tk
import random
import json
import time

PARAMS_FILE = "params.json"
COMMAND_FILE = "command.txt"

def check_for_commands():
    try:
        with open(COMMAND_FILE, "r") as f:
            cmd = f.read().strip()

        if cmd == "START_OBJ2":
            perform_task()

    except Exception:
        pass

    root.after(500, check_for_commands)

def perform_task():
    try:
        with open(PARAMS_FILE, "r") as f:
            data = json.load(f)

        n = data['n']
        min_val = data['min']
        max_val = data['max']

        numbers = [round(random.uniform(min_val, max_val), 2) for _ in
range(n)]

        text_area.delete('1.0', tk.END)
        formatted_text = ""
        for i, num in enumerate(numbers):
            formatted_text += f"{num:^10}"
            if (i + 1) % 5 == 0:
                formatted_text += "\n"
```

```
            text_area.insert(tk.END, formatted_text)

            root.clipboard_clear()
            root.clipboard_append(json.dumps(numbers))
            root.update()

            with open(COMMAND_FILE, "w") as f:
                f.write("DONE_OBJ2")

        except Exception as e:
            print(f"Object2 Error: {e}")

root = tk.Tk()
root.title("Object 2")
root.geometry("450x300+560+50")

tk.Label(root, text="Object 2 (Generator)", font=("Arial", 13)).pack()
text_area = tk.Text(root, height=10, width=60, font=("Consolas", 12))
text_area.pack(pady=10)

root.after(500, check_for_commands)
root.mainloop()
```

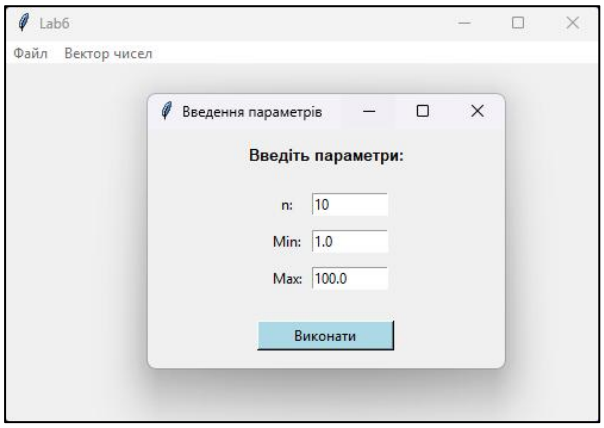
Object3.py:

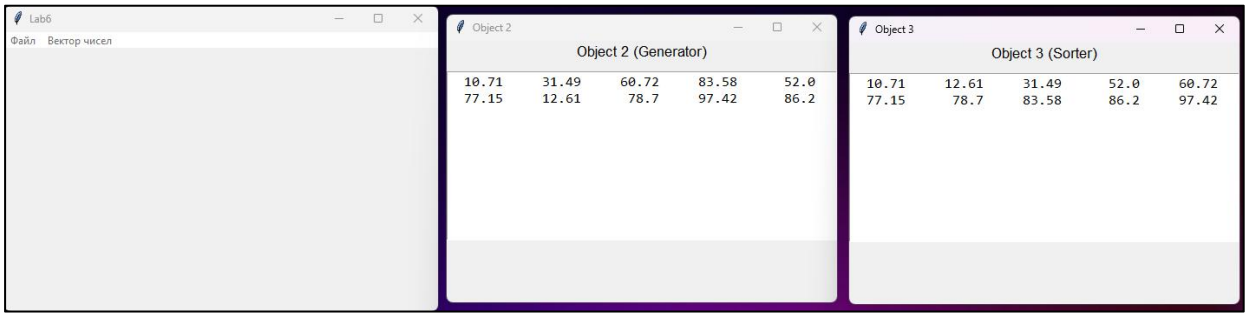
```
import tkinter as tk
import json

COMMAND_FILE = "command.txt"

def check_for_commands():
    try:
        with open(COMMAND_FILE, "r") as f:
            cmd = f.read().strip()

        if cmd == "START_OBJ3":
            perform_sort()

    except Exception:
        pass

    root.after(500, check_for_commands)

def perform_sort():
    try:
        content = root.clipboard_get()
        numbers = json.loads(content)

        numbers.sort()

        text_area.delete('1.0', tk.END)
        formatted_text = ""
        for i, num in enumerate(numbers):
            formatted_text += f"{num:^10}"
            if (i + 1) % 5 == 0:
                formatted_text += "\n"
        text_area.insert(tk.END, formatted_text)

        with open(COMMAND_FILE, "w") as f:
            f.write("DONE_OBJ3")

    except Exception as e:
        print(f"Object3 Error: {e}")
```

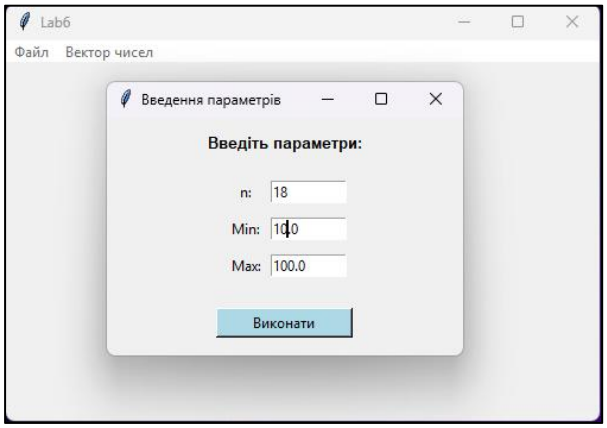
```
root = tk.Tk()
root.title("Object 3")
root.geometry("450x300+1020+50")

tk.Label(root, text="Object 3 (Sorter)", font=("Arial", 13)).pack()
text_area = tk.Text(root, height=10, width=60, font=("Consolas", 12))
text_area.pack(pady=10)

root.after(500, check_for_commands)
root.mainloop()
```

Скріншоти:


Lab6
Файл
Вектор чисел


Lab6
Файл
Вектор чисел
Введення параметрів
Введіть параметри:
n:
10
Min:
1.0
Max:
100.0
Виконати


Lab6
Файл
Вектор чисел
Object 2
Object 2 (Generator)
10.71 31.49 60.72 83.58 52.0
77.15 12.61 78.7 97.42 86.2
Object 3
Object 3 (Sorter)
10.71 12.61 31.49 52.0 60.72
77.15 78.7 83.58 86.2 97.42


Lab6
Файл
Вектор чисел
Введення параметрів
Введіть параметри:
n:
18
Min:
100
Max:
100.0
Виконати

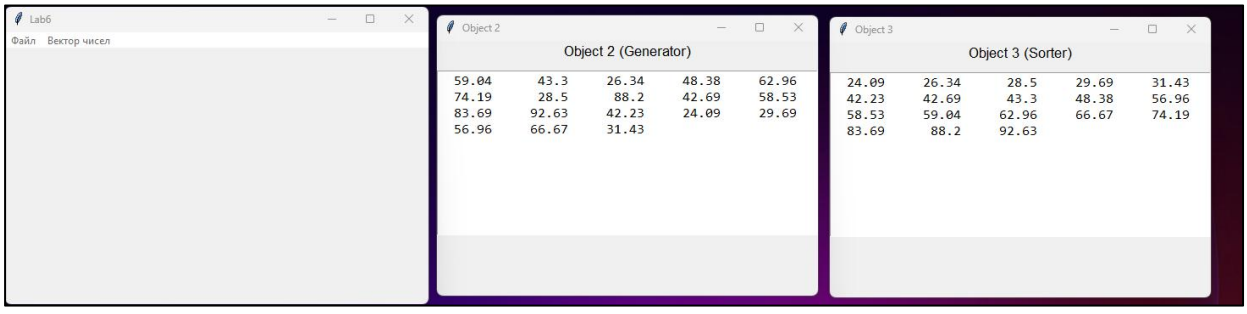
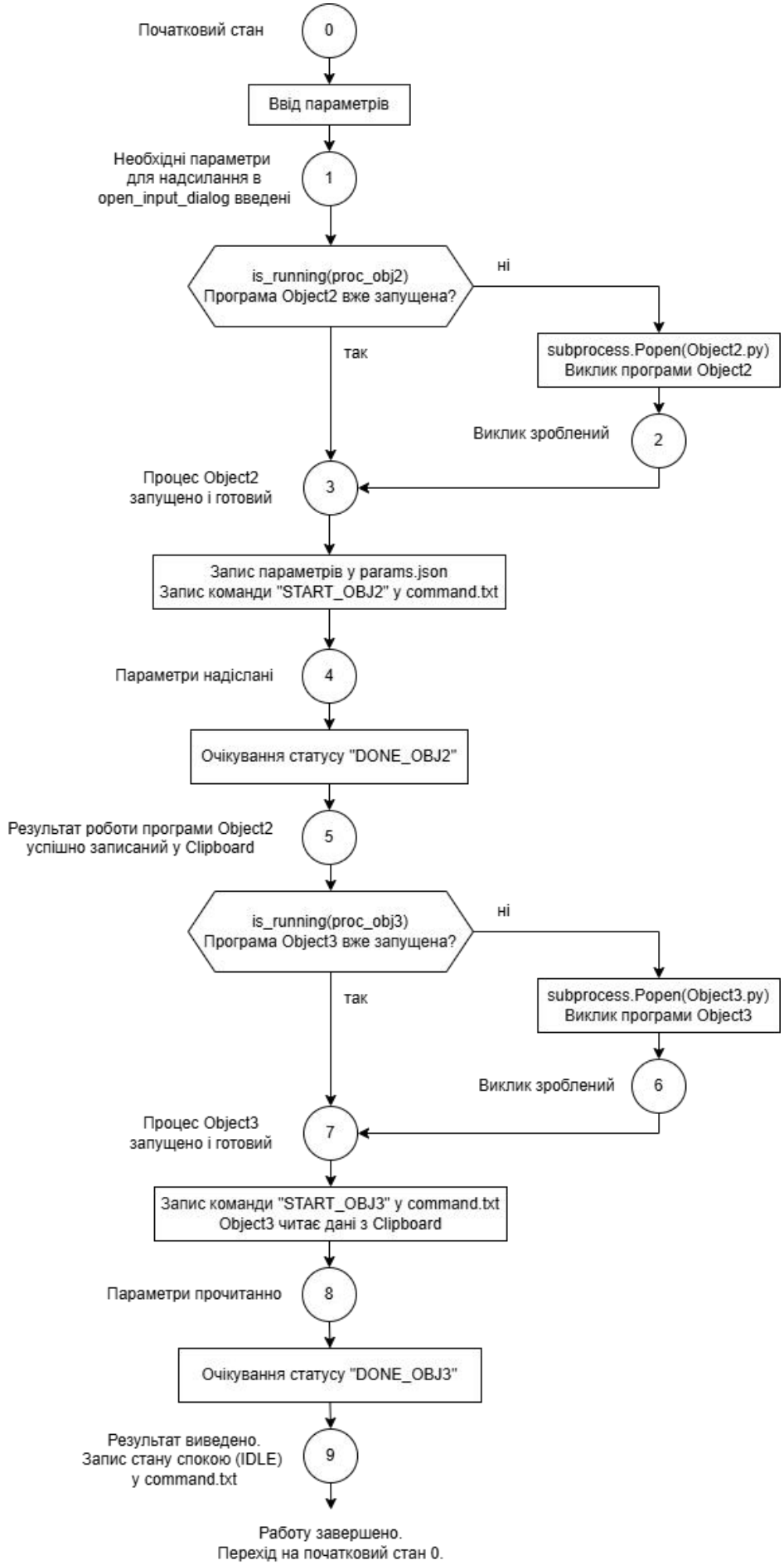

Lab6
Файл
Вектор чисел
Object 2
Object 2 (Generator)
59.04 43.3 26.34 48.38 62.96
74.19 28.5 88.2 42.69 58.53
83.69 92.63 42.23 24.09 29.69
56.96 66.67 31.43
Object 3
Object 3 (Sorter)
24.09 26.34 28.5 29.69 31.43
42.23 42.69 43.3 48.38 56.96
58.53 59.04 62.96 66.67 74.19
83.69 88.2 92.63

Схема послідовності надсилання-обробки повідомлень:



Висновок

В ході даної лабораторної роботи було запрограмовано програмну систему з множини окремих незалежних програм, керованих повідомленнями. В результаті було створено 3 незалежні програми, які автоматично взаємодіють між собою.

Файл з основною програмою має назву Lab6.py. В ньому реалізовано головне вікно менеджера з меню, в якому можна обрати пункт "Вектор чисел" для виклику діалогового вікна введення даних.

У діалоговому вікні є три поля для вводу параметрів: n, Min та Max. При натисканні на кнопку "Виконати", програма зчитує введені параметри та запускає автоматичну послідовність дій. Далі програма Lab6 викликає програму Object2, передаючи їй параметри через файл params.json та команду "START_OBJ2" через файл command.txt.

Програма Object2 постійно моніторить файл command.txt. При отриманні команди "START_OBJ2" вона зчитує параметри з файлу params.json, генерує вектор n дробових чисел у заданому діапазоні та відображає їх у власному головному вікні у декількох стовпчиках та рядках. Для цього використовується текстове поле з форматуванням, де кожні 5 чисел формують новий рядок. Після завершення роботи програма копіює дані у буфер обміну у форматі JSON та надсилає повідомлення "DONE_OBJ2" назад до Lab6.

Коли Lab6 отримує повідомлення про завершення роботи Object2, вона автоматично викликає програму Object3 та надсилає їй команду "START_OBJ3" через файл command.txt. Програма Object3 отримує дані з буферу обміну, виконує сортування масиву чисел та відображає відсортований вектор у власному головному вікні у декількох стовпчиках та рядках. Після цього вона надсилає повідомлення "DONE_OBJ3" про завершення роботи.

Усі програми автоматично розташовуються на екрані так, щоб користувач міг одночасно бачити всі результати. При закритті головної програми Lab6 автоматично закриваються також програми Object2 та Object3, що забезпечує коректне завершення роботи всієї системи.

Таким чином, було реалізовано повноцінну систему з трьох незалежних програм, які обмінюються повідомленнями та автоматично виконують поставлене завдання без участі користувача після початкового введення параметрів.