

**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» на тему  
“Побудування програмної  
системи з множини об'єктів,  
керованих повідомленнями”**

Виконала:  
Студентка групи ІМ-33  
Пилипчук Вероніка Олексіївна  
Номер у списку групи: 18

Перевірив:  
Порєв В.М.

Київ 2024

### **Варіант завдання:**

Головний проект – програма Lab6 має бути менеджером, який керує двома іншими програмами – Object2 та Object3. Програма Lab6 повинна автоматично, без участі користувача налагоджувати співпрацю та виконувати обмін повідомленнями з програмами Object2 та Object3 для виконання потрібного завдання згідно наведеному вище алгоритму.

У кожній програмі передбачити автоматичну обробку потрібних повідомлень і переходів з одного стану в інший.

Далі вимоги щодо функціонування системи об'єктів.

1. Для початку роботи користувач програми вибирає потрібний пункт меню програми Lab6. Далі з'являється вікно діалогу, у якому потрібно ввести параметри згідно варіанту завдання. У вікні діалогу користувач натискає кнопку "Так" (або "Виконати") і на цьому місці користувача закінчується – далі він тільки спостерігає, як програма сама автоматично виконає усе, що потрібно для отримання результату. Виклик інших програм – Object2 та Object3 головна програма Lab6 повинна робити без участі користувача.

18 варіант:

Програма Lab6

Користувач вводить значення  $n$ , Min, Max у діалоговому вікні. Програма викликає програми Object2, 3 і забезпечує обмін повідомленнями для передавання та отримання інформації.

Програма Object2

Створює вектор  $n$  дробових (double) чисел у діапазоні Min – Max. Показує числові значення у декількох стовпчиках та рядках у власному головному вікні. Записує дані в Clipboard Windows у текстовому форматі.

Програма Object3

Зчитує дані з Clipboard Windows. Виконує сортування масиву чисел і відображає його у декількох стовпчиках та рядках у власному головному вікні.

2. Обмін повідомленнями та масивами даних між програмами Lab6, Object2 та Object3 повинен відбуватися автоматично, без участі користувача. Програма Lab6 повинна автоматично у певній послідовності знаходити та викликати програми Object2 та Object3.

3. У результаті одного сеансу роботи користувач повинен бачити головні вікна програм Object2 та Object3, у яких відображатимуться потрібні результати відповідно варіанту завдань. Для цього вікна програм повинні автоматично розташуватися так, щоб усі результати було видно. Програма Lab6 повинна залишатися у активному стані, щоб користувач мав можливість повторно виконати роботу.

4. Передбачити варіанти успішної роботи у випадках, коли програми Object2 та Object3 (одна або обидві) до того вже були викликані.

5. По завершенні роботи програми Lab6 повинні автоматично завершуватися і програми Object2 та Object3.

## 1. Вихідний текст головного файлу (page.tsx):

```
"use client";
```

```
import { useEffect, useState } from "react";
import { Lab6 } from "../components/lab6/Lab6";
import { Object2 } from "../components/lab6/Object2";
import { Object3 } from "../components/lab6/Object3";
```

```
const App = () => {
  const [windowType, setWindowType] = useState("");

  useEffect(() => {
    const urlParams = new URLSearchParams(window.location.search);
    if (urlParams.get("window") === "obj2") {
      setWindowType("obj2");
    } else if (urlParams.get("window") === "obj3") {
      setWindowType("obj3");
    }
  }, []);

  switch (windowType) {
    case "obj2":
      return <Object2 />;
    case "obj3":
      return <Object3 />;
    default:
      return <Lab6 />;
  }
};

export default App;
```

## 2. Файл головного вікна Lab6.tsx

```
import { emit } from "@tauri-apps/api/event";
import { useState } from "react";
import { Button, Input, Modal } from "antd";

export interface GenerationParams {
  n: number;
  min: number;
  max: number;
}

export interface GeneratedData {
  numbers: number[];
}

export const Lab6 = () => {
  const [isModalOpen, setIsModalOpen] = useState(false);
  const [params, setParams] = useState<GenerationParams>({
    n: 0,
    min: 0,
    max: 0,
  });

  const showModal = () => {
    setIsModalOpen(true);
  };

  const handleOk = () => {
    emit("start-generation", params);
    setIsModalOpen(false);
  };

  const handleChange = (field: keyof GenerationParams, value: string) => {
    setParams((prev) => ({
      ...prev,
      [field]: field === "n" ? parseInt(value, 10) : parseFloat(value),
    }));
  };

  return (
    <div
      style={{
        height: "100vh",
        display: "flex",
        alignItems: "center",
        justifyContent: "center",
      }}
    >
      <Button type="primary" onClick={showModal}>
        Почати
      </Button>
      <Modal
        centered
        closable={false}
        title="Введіть дані"
        open={isModalOpen}
        okText="Так"
        onOk={handleOk}
        cancelText="Скасувати"
        onCancel={() => setIsModalOpen(false)}
      >
        <div style={{ display: "flex", flexDirection: "column", gap: 10 }}>
          <Input
            placeholder="n"
            onChange={(e) => handleChange("n", e.target.value)}
          />
        </div>
      </Modal>
    </div>
  );
};
```

```
    />
    <Input
      placeholder="min"
      onChange={(e) => handleChange("min", e.target.value)}
    />
    <Input
      placeholder="max"
      onChange={(e) => handleChange("max", e.target.value)}
    />
  </div>
</Modal>
</div>
),
};
```

### 3. Файл другого вікна програми Object2.tsx

```
import { useEffect, useState } from "react";
import { listen } from "@tauri-apps/api/event";
import { GenerationParams } from "../Lab6";
import { writeText } from "@tauri-apps/api/clipboard";

export const Object2 = () => {
  const [numbers, setNumbers] = useState<number[]>([]);

  const copyToClipboard = async (numbers: number[]) => {
    const text = numbers.join(", ");
    await writeText(text);
  };

  useEffect(() => {
    const unlisten = listen<GenerationParams>(
      "start-generation",
      ({ payload }) => {
        const { n, min, max } = payload;
        const generatedNumbers = Array.from({ length: n }, () =>
          parseFloat((Math.random() * (max - min) + min).toFixed(5))
        );

        setNumbers(generatedNumbers);
        copyToClipboard(generatedNumbers);
      }
    );

    return () => {
      unlisten.then((off) => off());
    };
  }, []);

  return (
    <div>
      <div
        style={{
          display: "flex",
          justifyContent: "center",
          fontWeight: "bold",
          color: "#1677ff",
          margin: 20,
        }}
      >
        Згенеровані числа
      </div>
      <div
        style={{
          display: "grid",
          gridTemplateColumns: "repeat(4, 1fr)",
          gap: 10,
          margin: 20,
        }}
      >
        {numbers.map((num, index) => (
          <div key={index}>{num}</div>
        ))}
      </div>
    </div>
  );
};
```

#### 4. Файл третього вікна програми Object3.tsx

```
import { useEffect, useState } from "react";
import { readText } from "@tauri-apps/api/clipboard";

export const Object3 = () => {
  const [sortedNumbers, setSortedNumbers] = useState<number[]>([]);

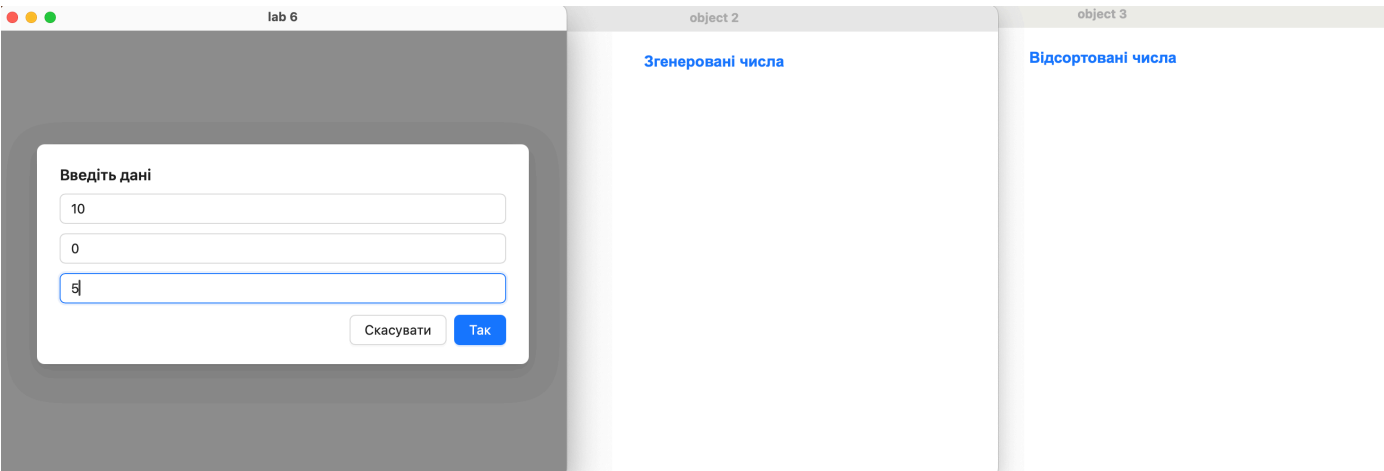
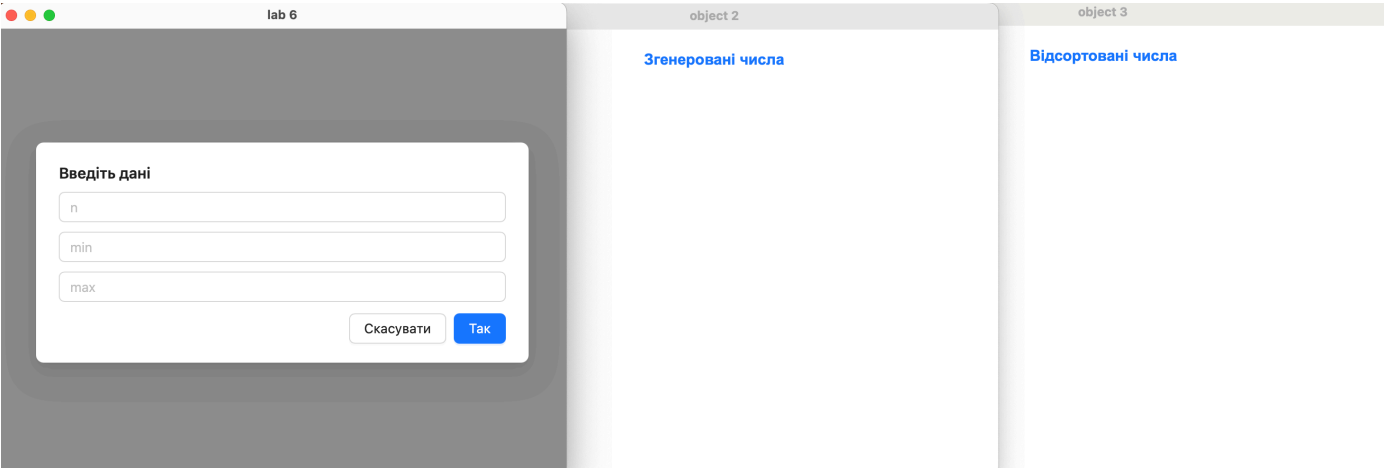
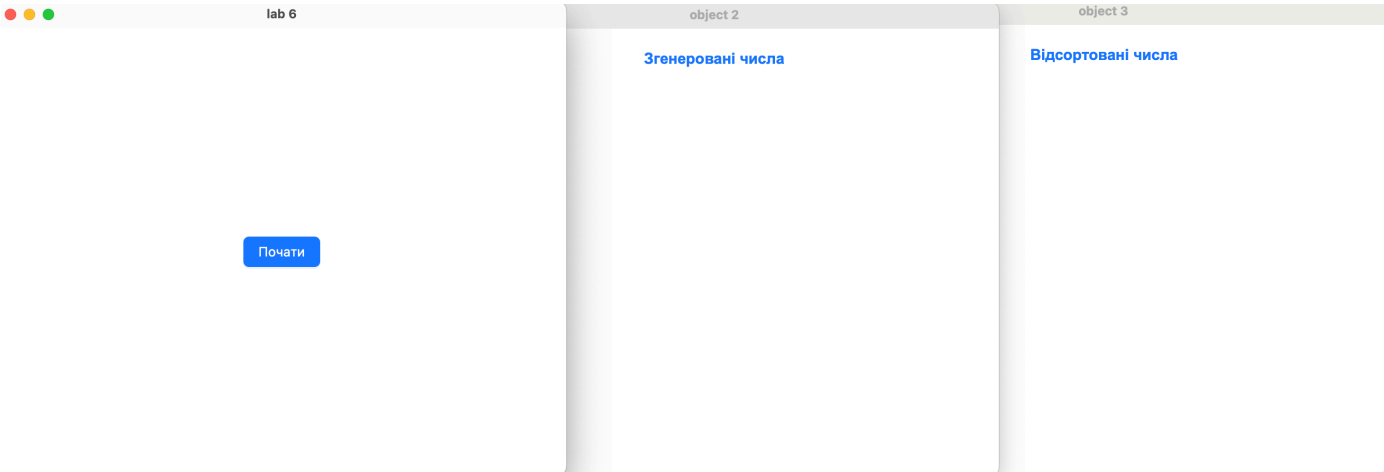
  const fetchFromClipboard = async () => {
    try {
      const text = await readText();
      if (text) {
        const numbers = text
          .split(",")
          .map((num) => parseFloat(num.trim()))
          .filter((num) => !isNaN(num));
        return numbers;
      }
    } catch (error) {
      console.error(error);
    }
    return [];
  };

  useEffect(() => {
    const interval = setInterval(() => {
      fetchFromClipboard().then((numbers) => {
        const sorted = [...numbers].sort((a, b) => a - b);
        setSortedNumbers(sorted);
      });
    });

    return () => clearInterval(interval);
  }, []);

  return (
    <div>
      <div
        style={{
          display: "flex",
          justifyContent: "center",
          fontWeight: "bold",
          color: "#1677ff",
          margin: 20,
        }}
      >
        Відсортовані числа
      </div>
      <div
        style={{
          display: "grid",
          gridTemplateColumns: "repeat(4, 1fr)",
          gap: 10,
          margin: 20,
        }}
      >
        {sortedNumbers.map((num, index) => (
          <div key={index}>{num}</div>
        ))}
      </div>
    </div>
  );
};
```

Скріншоти виконання програми:





object 2				object 3			
Згенеровані числа				Відсортовані числа			
2.08695	2.81562	3.95653	1.44323	0.16065	0.91457	1.04713	1.44323
0.16065	0.91457	4.01847	3.70004	2.08695	2.81562	3.70004	3.93886
3.93886	1.04713			3.95653	4.01847		

lab 6				object 3			
Введіть дані				Відсортовані числа			
20				0.16065	0.91457	1.04713	1.44323
0				2.08695	2.81562	3.70004	3.93886
3				3.95653	4.01847		
Скасувати Так							

object 2				object 3			
Згенеровані числа				Відсортовані числа			
2.47237	2.401	0.56934	0.31348	0.03086	0.31348	0.48188	0.56934
1.86295	1.97845	2.67042	0.48188	0.579	0.91697	1.55782	1.63778
2.48449	1.63778	0.03086	1.87286	1.86295	1.87286	1.97845	2.0254
1.55782	2.78888	2.0254	2.14047	2.14047	2.401	2.47237	2.47986
0.91697	2.56411	0.579	2.47986	2.48449	2.56411	2.67042	2.78888

**Висновки:**

Моя лабораторна робота виконана із використанням бібліотек для створення користувацьких інтерфейсів на Typescript із використанням технологій Clipboard з ані фреймворку Taui.

Спробувала виконати поставлену задачу згідно зі своїм варіантом. Створила нові модулі вікон та систематизувала “спілкування” між ними.