

**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 №1
з дисципліни
«Об'єктно-орієнтоване програмування»

Виконала:

студентка групи ІМ-43
Хубеджева Єлизавета Павлівна
номер у списку групи: 28

Перевірив:

Порєв В. М.

Завдання

Необхідно створити два пункти меню – "Робота1" та "Робота2".

1. Потрібно, щоб при виборі пункту меню "Робота1" виконувалося щось згідно варіанту V_1 , причому V_1 обчислюється за формулою

$$V_1 = Ж \bmod 4.$$

2. Запрограмувати також, щоб при виборі пункту меню "Робота2" виконувалося щось згідно варіанту V_2 :

$$V_2 = (Ж+1) \bmod 4.$$

Так, як номер у списку групи 28, то:

$$V_1 = 28 \bmod 4 = 0;$$

$$V_2 = (28+1) \bmod 4 = 1.$$

Завдання для $V_1 = 0$:

Вікно діалогу для вводу тексту, яке має стрічку вводу (Edit Control) та дві кнопки: [Так] і [Відміна]. Якщо ввести рядок тексту і натиснути [Так], то у головному вікні повинен відображатися текст, що був введений.

Завдання для $V_2 = 1$:

Вікно діалогу з повзуном горизонтального скролінгу (Horizontal scroll Bar) та дві кнопки: [Так] і [Відміна]. Рухаючи повзунок скролінгу користувач вводить число у діапазоні від 1 до 100. Після натискування кнопки [Так] вибране число буде відображатися у головному вікні.

Тексти головних файлів програми

Index.js:

```
const { app, BrowserWindow, Menu } = require("electron/main");
const path = require("node:path");
const { textField } = require("./module1");
const { numSlider } = require("./module2");

let mainwin;
exports.mainwin = mainwin;

const createWindow = () => {
  mainwin = new BrowserWindow({
    width: 1000,
    height: 600,
    minWidth: 300,
    minHeight: 300,
    webPreferences: {
      nodeIntegration: true,
      contextIsolation: false,
    },
  });
};

mainwin.loadFile(path.join(__dirname, "index.html"));

mainwin.on("closed", () => {
  mainwin = null;
});
};

app.whenReady().then(() => {
  createWindow();

  app.on("activate", () => {
    if (BrowserWindow.getAllWindows().length === 0) createWindow();
  });
});

const menu = [
  {
```

```

    label: "File",
    submenu: [{ label: "Exit", role: "quit" }],
  },
  {
    label: "Actions",
    submenu: [
      {
        label: "work 1",
        click: () => {
          textField(mainwin);
        },
      },
      {
        label: "work 2",
        click: async () => {
          const result = await numSlider(mainwin);
          if (result !== null) {
            mainwin.webContents.send("userNum", result);
          }
        },
      },
    ],
  },
],
{
  label: "Help",
  submenu: [{ label: "About", role: "about" }],
},
];
Menu.setApplicationMenu(Menu.buildFromTemplate(menu));

app.on("window-all-closed", () => {
  if (process.platform !== "darwin") app.quit();
});

```

Index.html:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="stylesheet" href="style.css" />
    <title>Laboratory work 1. Khubedzheva Elizabeth</title>
  </head>
  <body>
    <h1>Hello user!</h1>
    <p id="output-text"></p>
    <p id="output-number"></p>
    <script>
      const { ipcRenderer } = require("electron");
      ipcRenderer.on("userText", (event, text) => {
        document.getElementById("output-text").textContent =
          text !== "" ? "Entered text: " + text : "No text is entered";
      });

      ipcRenderer.on("userNum", (event, number) => {
        document.getElementById("output-number").textContent =
          "Selected number: " + number;
      });
    </script>
  </body>
</html>
```

Тексти модульних файлів програми

Module1.js:

```
const prompt = require("electron-prompt");

const textField = (window) => {
  return prompt({
    width: 500,
    height: 250,
    title: "Text field",
    label: "Enter your text",
    value: "text",
    inputAttrs: {
      type: "text",
    },
    type: "input",
  })
  .then((userText) => {
    if (userText !== null) {
      window.webContents.send("userText", userText);
    }
  })
  .catch(console.error);
};

module.exports = { textField };
```

Module2.js:

```
const { ipcMain, Browserwindow } = require("electron");
const path = require("path");

const numSlider = (window) => {
  return new Promise((resolve) => {
    const modal = new Browserwindow({
      width: 600,
      height: 250,
      parent: window,
      modal: true,
      resizable: false,
      autoHideMenuBar: true,
      webPreferences: {
        nodeIntegration: true,
        contextIsolation: false,
      },
    });
    modal.loadFile(path.join(__dirname, "module2.html"));
    modal.once("ready-to-show", () => modal.show());

    const onValue = (event, value) => {
      ipcMain.removeListener("slider-canceled", onCancel);
      resolve(value);
      if (!modal.isDestroyed()) modal.close();
    };

    const onCancel = () => {
      ipcMain.removeListener("slider-value-selected", onValue);
      resolve(null);
      if (!modal.isDestroyed()) modal.close();
    };

    ipcMain.once("slider-value-selected", onValue);
    ipcMain.once("slider-canceled", onCancel);
  });
};

module.exports = { numSlider };
```

Module2.html:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Select Number</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <div id="container">
      <p>Choose a number (1-100)</p>
      <div>
        <input id="slider" type="range" min="1" max="100" value="50" />
        <span id="value">50</span>
      </div>
      <div id="controls">
        <button id="cancel">Cancel</button>
        <button id="ok">OK</button>
      </div>
    </div>
    <script>
      const { ipcRenderer } = require("electron");
      const slider = document.getElementById("slider");
      const display = document.getElementById("value");
      const okBtn = document.getElementById("ok");
      const cancelBtn = document.getElementById("cancel");

      slider.addEventListener("input", () => {
        display.textContent = slider.value;
      });

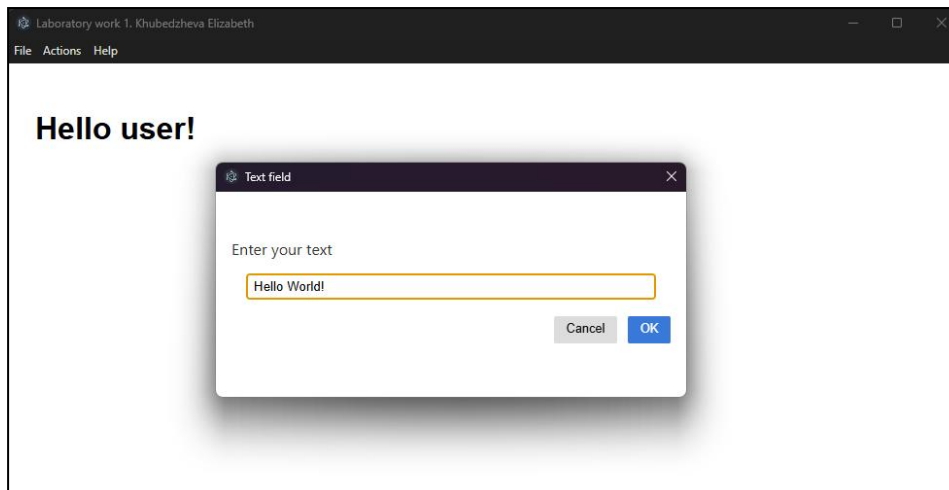
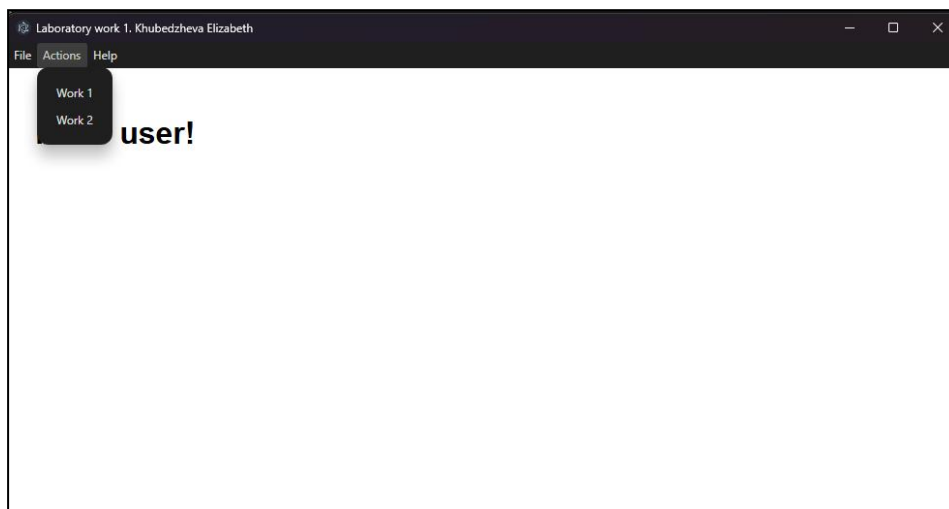
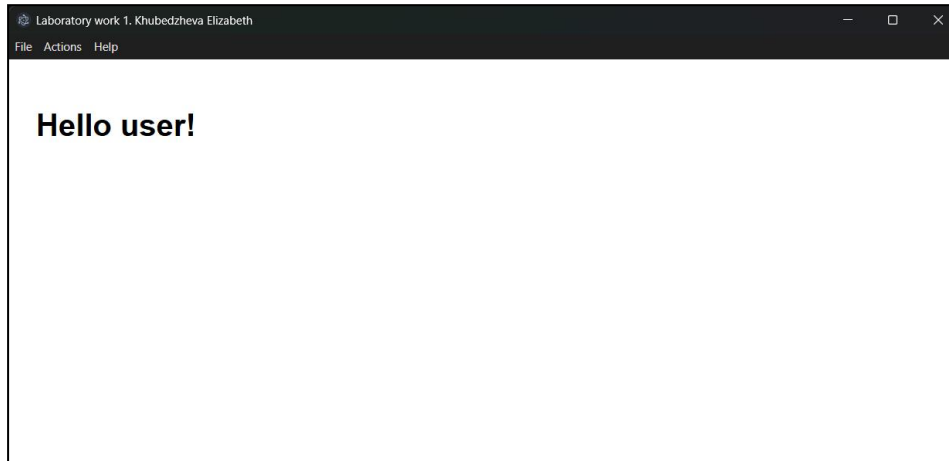
      okBtn.addEventListener("click", () => {
        ipcRenderer.send("slider-value-selected", Number(slider.value));
      });

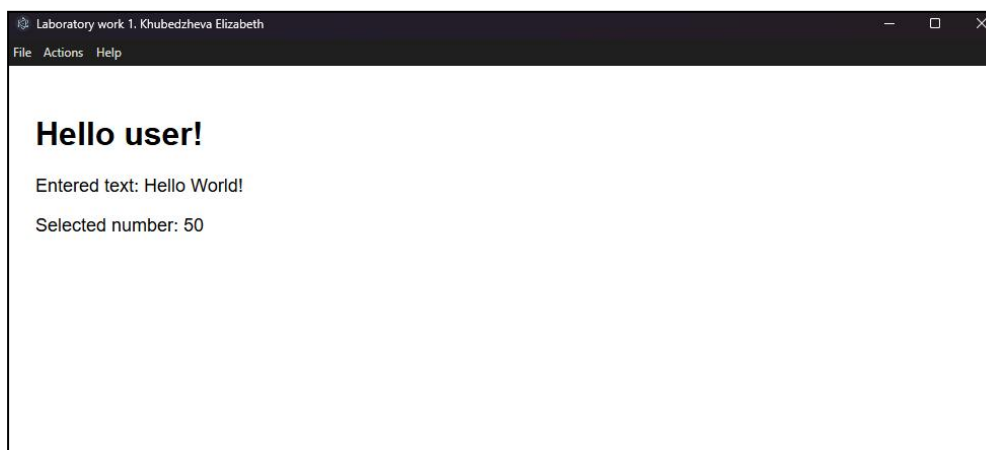
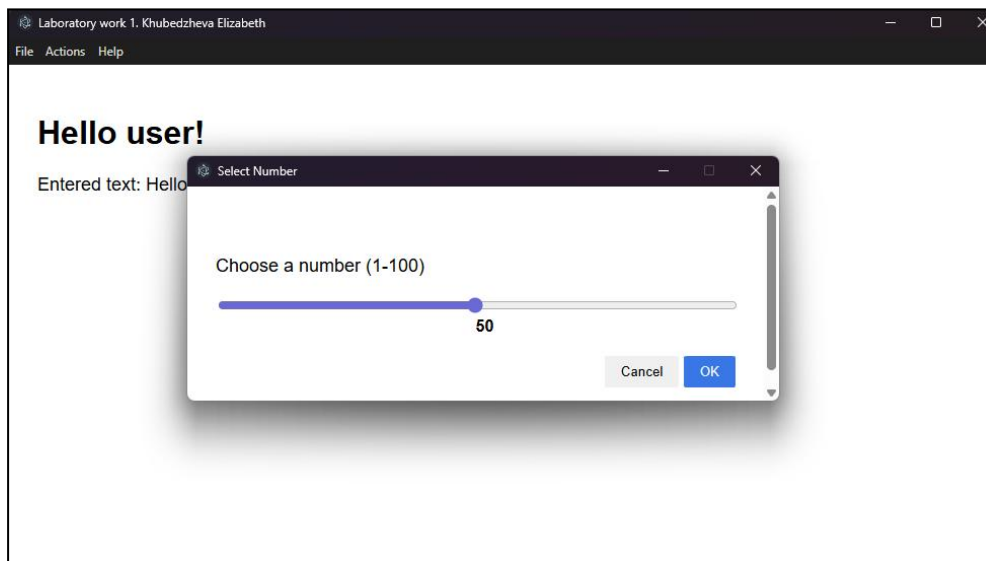
      cancelBtn.addEventListener("click", () => {
        ipcRenderer.send("slider-canceled");
      });
    </script>
```



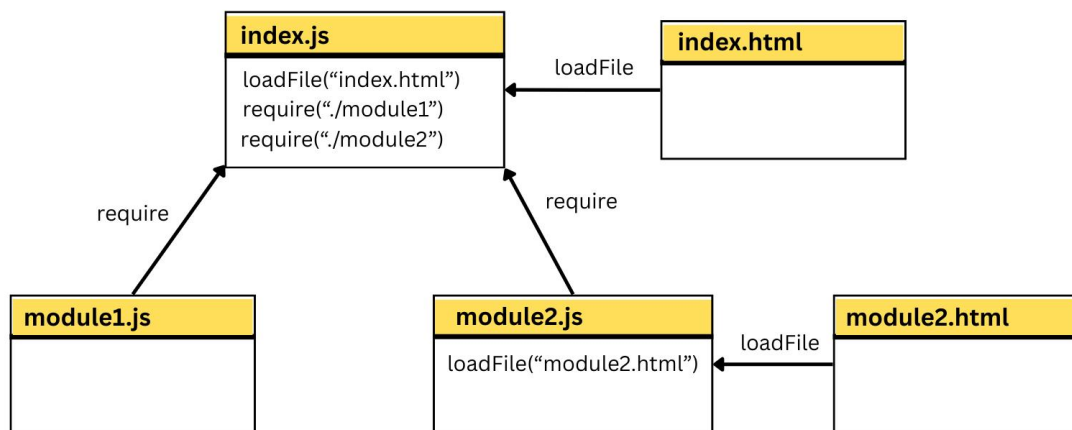
```
</body>  
</html>
```

Скріншоти:





Ієрархія файлів проєкту:



Висновок

В ході виконання лабораторної роботи було створено програму на базі *Electron* - фреймворку для створення крос-платформених настільних застосунків за допомогою веб-технологій (JavaScript, HTML та CSS).

Проект демонструє роботу з користувацьким вводом: 1) поле для введення тексту (*Work1*), 2) горизонтальний повзунок для вибору числа (*Work2*); отримані значення відображаються у головному вікні.

Архітектура програми побудована модульно: головний файл *index.js* виступає як точка входу та підключає допоміжні модулі, які експортують конкретні функції (*textFile*, *numSlider*), модулі не виконують UI під час імпорту, а лише ініціюють діалоги за викликом, що забезпечує розподіл відповідальностей і полегшує тестування. Передача даних між *main*- і *renderer*-процесами організована через механізм IPC: після підтвердження вводу модулі пересилають результати у головне вікно за допомогою *mainWindow.webContents.send*, а рендерер отримує їх через *ipcRenderer.on* і відображає в інтерфейсі.

У роботі продемонстровано правильну семантику роботи з файлами: модулі імпортуються (*require*) у *main*-процесі, а HTML-файли не імпортуються як модулі - вони завантажуються у вікна (*loadFile*). Такий підхід спрощує підтримку та розширення функціоналу застосунку, зберігаючи безпеку та чистоту архітектури.