**Лабораторна робота №3**

Основи роботи із TypeScript

Мета: Навчитись розробляти веб-додатки за допомогою мови програмування TypeScript.

**Завдання**

Рефакторинг завданнь з попередніх лабораторних робіт на мову програмування TypeScript.

**Хід роботи**

1. **Завантажуємо пакет TypeScript**

Виконуємо у терміналі команду **npm install -g typescript**



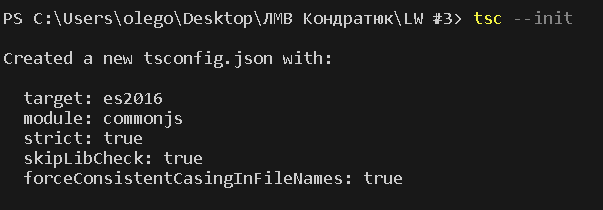
Далі перевіряємо версію typeScript командою tsc -v



Версію виводить – значить пакет було встановлено.

1. **Створюємо файл конфігурації**

Файл конфігурації дозволяє нам встановлювати налаштування для TypeScript, визначати інструменти, версії пакету. Щоб створити цей файл – потрібно виконати команду **tsc –init** у терміналі середовища розробки.



Після цього, в директорії проєкту у нас повинен з’явитися файл конфігурації – **tsconfig.json**

1. **Компілюємо TypeScript**

Спочатку створюємо в директорії проєкту файл **script.ts,** розширення ts – typescript. Та додамо у нього для прикладу, перше завдання з першої лабораторної: «Дано величину А, що виражає обсяг в байтах. Перекласти А в більш великі обсяги інформації»:

function convertBytes(): void {

    const inputBytesElement: HTMLInputElement | null = document.getElementById("input-bytes") as HTMLInputElement;

    const selectedUnitElement: HTMLSelectElement | null = document.getElementById("conversion-unit") as HTMLSelectElement;

    const resultElement: HTMLElement | null = document.getElementById("result");

    if (inputBytesElement === null || selectedUnitElement === null || resultElement === null) {

        return;

    }

    var inputBytes: number = parseFloat(inputBytesElement.value);

    var selectedUnit: string = selectedUnitElement.value;

    var result: string = "";

    if (selectedUnit === "kb") {

        result = (inputBytes / 1024) + " Кб";

    } else if (selectedUnit === "mb") {

        result = (inputBytes / 1024 / 1024) + " Мб";

    } else if (selectedUnit === "gb") {

        result = (inputBytes / 1024 / 1024 / 1024) + " Гб";

    } else {

        result = "Не вдалося прочитати міру величини в яку здійснюється перетворення.";

    }

    resultElement.innerText = "Результат: " + result;

}

Як бачимо, у TypeScript при оголошенні методу або функції, в дужках вказується перелік параметрів з їх типом даних, а також тип даних, який повертає фукнція.

Тепер якщо ми запустимо проєкт, у нас всеодно не буде працювати скрипт, тому що для початку його потрібно компілювати ts файл у формат js. Для цього виконуємо команду **tsc ‘імя файлу’:**



Після цього у нас повинен з’явитися файл **script.js** який потрібно буде під’єднати до HTML сторінки.

1. **Здійснюємо рефакторинг завдань з 1 та 2 лабораторних робіт, додаючи типи даних до змінних, функцій.**

## #1 Дана величина А, що виражає обсяг інформації в байтах. Перекласти А в більш великі одиниці виміру інформації.

function convertBytes(): void {

    const inputBytesElement: HTMLInputElement | null = document.getElementById("input-bytes") as HTMLInputElement;

    const selectedUnitElement: HTMLSelectElement | null = document.getElementById("conversion-unit") as HTMLSelectElement;

    const resultElement: HTMLElement | null = document.getElementById("result");

    if (inputBytesElement === null || selectedUnitElement === null || resultElement === null) {

        return;

    }

    var inputBytes: number = parseFloat(inputBytesElement.value);

    var selectedUnit: string = selectedUnitElement.value;

    var result: string = "";

    if (selectedUnit === "kb") {

        result = (inputBytes / 1024) + " Кб";

    } else if (selectedUnit === "mb") {

        result = (inputBytes / 1024 / 1024) + " Мб";

    } else if (selectedUnit === "gb") {

        result = (inputBytes / 1024 / 1024 / 1024) + " Гб";

    } else {

        result = "Не вдалося прочитати міру величини в яку здійснюється перетворення.";

    }

    resultElement.innerText = "Результат: " + result;}

## #2 Знайти першу цифру числа.

function getFirstDigit(): void {

    const resultElement: HTMLElement | null = document.getElementById("result-2");

    const inputValueElement: HTMLInputElement | null = document.getElementById("input-number") as HTMLInputElement;

    if (resultElement === null || inputValueElement === null) {

        return;

    }

    var inputValue: string = inputValueElement.value;

    var firstDigit: string = "";

    if (inputValue.length > 0) {

        if (inputValue.charAt(0) === '-' || inputValue.charAt(0) === '.') {

            firstDigit = inputValue.charAt(1);

        } else {

            firstDigit = inputValue.charAt(0);

        }

        resultElement.innerText = "Перша цифра: " + firstDigit;

    } else {

        resultElement.innerText = "Не введено число.";

    }

}

## #3 Надрукувати найдовше слово з тексту.

function getLongestWord(): void {

    const inputTextElement: HTMLInputElement | null = document.getElementById("input-text") as HTMLInputElement;

    const resultElement: HTMLElement | null = document.getElementById("result-3");

    if (inputTextElement === null || resultElement === null) {

        return;

    }

    var words: string[] = inputTextElement.value.split(" ");

    var longestWord: string = "";

    var maxLength: number = 0;

    for (let word of words) {

        if (word.length > maxLength) {

            longestWord = word;

            maxLength = word.length;

        }

    }

    resultElement.innerText = "Найдовше слово: " + longestWord;

}

## #4 Виведіть в консоль підрядок з 6 по 20 символ включно;

function getSubstring(startPosition: number, endPosition: number): void {

    const resultElement: HTMLElement | null = document.getElementById("result-4");

    const inputText: HTMLInputElement | null = document.getElementById("input-text2") as HTMLInputElement;

    if (inputText === null || resultElement === null) {

        return;

    }

    var substring: string = inputText.value.substring(startPosition - 1, endPosition);

    resultElement.innerText = "Підрядок з " + startPosition + " по " + endPosition + " символ: " + substring;

}

## #5 Знайти максимальний елемент масиву і його номер за умови, що всі елементи масиву різні.

function generateMatrix(size : number) : number[][] {

    var matrix : number[][] = [];

    var usedNumbers : number[] = [];

    for (let i = 0; i < size; i++) {

        let row : number[] = [];

        for (let j = 0; j < size; j++) {

            let randomNumber : number;

            do {

                randomNumber = Math.floor(Math.random() \* (size \* size));

            } while (usedNumbers.indexOf(randomNumber) !== -1);

            usedNumbers.push(randomNumber);

            row.push(randomNumber);

        }

        matrix.push(row);

    }

    return matrix;

}

function getMatrix(): void {

    const matrixSizeInput: HTMLInputElement | null = document.getElementById("matrix-size") as HTMLInputElement;

    const matrixElement: HTMLElement | null = document.getElementById("result-5-matrix");

    const maxElement: HTMLElement | null = document.getElementById("result-5-max");

    if (matrixSizeInput === null || matrixElement === null || maxElement === null) {

        return;

    }

    let matrixSize: number = parseInt(matrixSizeInput.value);

    if (matrixSize > 0) {

        let matrix: number[][] = generateMatrix(matrixSize);

        let matrixString: string = "Згенерована матриця:\n";

        for (let i = 0; i < matrix.length; i++) {

            matrixString += matrix[i].join('\t') + '\n';

        }

        matrixElement.innerText = matrixString;

        let max: number = matrix[0][0];

        let rowIndex: number = 0;

        let columnIndex: number = 0;

        for (let i = 0; i < matrix.length; i++) {

            for (let j = 0; j < matrix[i].length; j++) {

                if (matrix[i][j] > max) {

                    max = matrix[i][j];

                    rowIndex = i;

                    columnIndex = j;

                }

            }

        }

        maxElement.innerText = "Максимальний елемент: " + max + "\nЙого координати: (" + (rowIndex + 1) + ", " + (columnIndex + 1) + ")";

    } else {

        matrixElement.innerText = "Введіть коректний розмір матриці.";

        maxElement.innerText = "";

    }

}

## #6 Є цілочисельна матриця A (k, l). Перевірити, чи є в ній елементи, рівні нулю. Якщо є, знайти номер першого з них.

function generateArray(size: number): number[][] {

    let array: number[][] = [];

    for (let i = 0; i < size; i++) {

        let row: number[] = [];

        for (let j = 0; j < size; j++) {

            let randomNumber: number = Math.floor(Math.random() \* size);

            row.push(randomNumber);

        }

        array.push(row);

    }

    return array;

}

function getArray(): void {

    const arraySizeInput: HTMLInputElement | null = document.getElementById("array-size") as HTMLInputElement;

    const arrayElement: HTMLElement | null = document.getElementById("result-6-array");

    const zeroIndexElement: HTMLElement | null = document.getElementById("result-6-zero");

    if (arraySizeInput === null || arrayElement === null || zeroIndexElement === null) {

        return;

    }

    let arraySize: number = parseInt(arraySizeInput.value);

    if (arraySize > 0) {

        let array: number[][] = generateArray(arraySize);

        let arrayString: string = "Згенерований масив:\n";

        for (let i = 0; i < array.length; i++) {

            arrayString += array[i].join('\t') + '\n';

        }

        arrayElement.innerText = arrayString;

        let zeroIndex: number = -1;

        let zeroY: number = -1;

        for (let i = 0; i < arraySize; i++) {

            for (let j = 0; j < arraySize; j++) {

                if (array[i][j] === 0) {

                    zeroIndex = i;

                    zeroY = j;

                    break;

                }

            }

            if (zeroIndex !== -1 && zeroY !== -1) {

                break;

            }

        }

        if (zeroIndex !== -1) {

            zeroIndexElement.innerText = "Перший елемент, що рівний нулю, має координати: " + (zeroIndex + 1) + "; " + (zeroY + 1);

        } else {

            zeroIndexElement.innerText = "У масиві немає елементів, що рівні нулю.";

        }

    } else {

        arrayElement.innerText = "Введіть коректний розмір масиву.";

        zeroIndexElement.innerText = "";

    }

}

## #7 Список співробітників підприємства містить наступні відомості: ПІБ, номер відділу, номер лабораторії, освіта, посада, рік народження. Скласти список співробітників заданого відділу, розділивши співробітників по лабораторіях.

class LabWorker {

    surname: string;

    name: string;

    patronymic: string;

    yearBirth: number;

    viddilNumber: number;

    labNumber: number;

    osvita: string;

    posada: string;

    constructor(

        surname: string,

        name: string,

        patronymic: string,

        yearBirth: number,

        viddilNumber: number,

        labNumber: number,

        osvita: string,

        posada: string

    ) {

        this.surname = surname;

        this.name = name;

        this.patronymic = patronymic;

        this.yearBirth = yearBirth;

        this.viddilNumber = viddilNumber;

        this.labNumber = labNumber;

        this.osvita = osvita;

        this.posada = posada;

    }

}

class EmployeeCollection {

    employees: LabWorker[];

    constructor() {

        this.employees = [];

    }

    addWorker(worker: LabWorker): void {

        this.employees.push(worker);

    }

    getWorkersByDepartment(viddilNumber: number): LabWorker[] {

        return this.employees.filter((worker) => worker.viddilNumber === viddilNumber);

    }

}

const employees : EmployeeCollection = new EmployeeCollection();

const viddilNumbers : number[] = [];

function addEmployee(): void {

    const surnameInput : HTMLInputElement | null = document.getElementById("surname") as HTMLInputElement;

    const nameInput : HTMLInputElement | null = document.getElementById("name") as HTMLInputElement;

    const patronymicInput : HTMLInputElement | null = document.getElementById("patronymic") as HTMLInputElement;

    const yearBirthInput : HTMLInputElement | null = document.getElementById("yearBirth") as HTMLInputElement;

    const viddilNumberInput  : HTMLInputElement | null= document.getElementById("viddilNumber") as HTMLInputElement;

    const labNumberInput : HTMLInputElement | null = document.getElementById("labNumber") as HTMLInputElement;

    const osvitaInput : HTMLInputElement | null = document.getElementById("osvita") as HTMLInputElement;

    const posadaInput : HTMLInputElement | null = document.getElementById("posada") as HTMLInputElement;

    if (

        surnameInput &&

        nameInput &&

        patronymicInput &&

        yearBirthInput &&

        viddilNumberInput &&

        labNumberInput &&

        osvitaInput &&

        posadaInput

    ) {

        let surname: string = surnameInput.value;

        let name: string = nameInput.value;

        let patronymic: string = patronymicInput.value;

        let yearBirth: number | null = parseInt(yearBirthInput.value);

        let viddilNumber: number | null = parseInt(viddilNumberInput.value);

        let labNumber: number | null = parseInt(labNumberInput.value);

        let osvita: string = osvitaInput.value;

        let posada: string = posadaInput.value;

        let worker : LabWorker = new LabWorker(surname, name, patronymic, yearBirth, viddilNumber, labNumber, osvita, posada);

        addEmployeeToCollection(worker);

    }

}

function addEmployeeToCollection(worker: LabWorker): void {

    if (viddilNumbers.indexOf(worker.viddilNumber)===-1) {

        const select : HTMLSelectElement = document.getElementById("departmentSelect") as HTMLSelectElement;

        const option : HTMLOptionElement = document.createElement("option");

        option.value = worker.viddilNumber.toString();

        option.textContent = "Відділ " + worker.viddilNumber;

        select.appendChild(option);

        viddilNumbers.push(worker.viddilNumber);

    }

    employees.addWorker(worker);

    clearInputFields();

}

function clearInputFields(): void {

    (document.getElementById("surname") as HTMLInputElement).value = "";

    (document.getElementById("name") as HTMLInputElement).value = "";

    (document.getElementById("patronymic") as HTMLInputElement).value = "";

    (document.getElementById("yearBirth") as HTMLInputElement).value = "";

    (document.getElementById("viddilNumber") as HTMLInputElement).value = "";

    (document.getElementById("labNumber") as HTMLInputElement).value = "";

    (document.getElementById("osvita") as HTMLInputElement).value = "";

    (document.getElementById("posada") as HTMLInputElement).value = "";

}

function displayEmployees(): void {

    const selectedDepartmentHTML: HTMLSelectElement | null = document.getElementById("departmentSelect") as HTMLSelectElement;

    const employeeList: HTMLElement | null = document.getElementById("employeeList");

    if (employeeList === null || selectedDepartmentHTML === null) {

        return;

    }

    let selectedDepartment: number = parseInt(selectedDepartmentHTML.value);

    employeeList.innerHTML = "";

    let result = employees.getWorkersByDepartment(selectedDepartment);

    let added: LabWorker[] = [];

    let previousLabNumber: number | null = null;

    result.forEach((worker) => {

        if (worker.viddilNumber === selectedDepartment) {

            added.push(worker);

        }

    });

    added.sort((a, b) => a.labNumber - b.labNumber);

    let separator: HTMLHRElement = document.createElement("hr");

    added.forEach((worker) => {

        let listItem: HTMLLIElement = document.createElement("li");

        if (worker.labNumber !== previousLabNumber) {

            employeeList.appendChild(separator);

            previousLabNumber = worker.labNumber;

        }

        listItem.textContent = `${worker.surname} ${worker.name} ${worker.patronymic} - відділ: ${worker.viddilNumber}, лабораторія: ${worker.labNumber}`;

        employeeList.appendChild(listItem);

    });

    employeeList.appendChild(separator);

}

addEmployeeToCollection(new LabWorker("Прізвище1", "Ім'я1", "По-батькові1", 1990, 1, 101, "Вища", "Посада1"));

addEmployeeToCollection(new LabWorker("Прізвище2", "Ім'я2", "По-батькові2", 1995, 1, 101, "Вища", "Посада2"));

addEmployeeToCollection(new LabWorker("Прізвище3", "Ім'я3", "По-батькові3", 1985, 2, 102, "Вища", "Посада3"));

addEmployeeToCollection(new LabWorker("Прізвище4", "Ім'я4", "По-батькові5", 1985, 2, 103, "Вища", "Посада3"));

addEmployeeToCollection(new LabWorker("Прізвище5", "Ім'я5", "По-батькові5", 1985, 2, 102, "Вища", "Посада3"));

addEmployeeToCollection(new LabWorker("Прізвище6", "Ім'я6", "По-батькові6", 1985, 2, 104, "Вища", "Посада3"));

displayEmployees();

**#8 Відділ кадрів (людина, звання, посада, штатний розклад)**

class Person {

    name: string;

    surname: string;

    birthdate: string;

    titleName: string;

    positionName: string;

    constructor(name: string, surname: string, birthdate: string, titleName: string, positionName: string) {

        this.name = name;

        this.surname = surname;

        this.birthdate = birthdate;

        this.titleName = titleName;

        this.positionName = positionName;

    }

}

class HRDepartment {

    people: Person[] = [];

    addPerson(person: Person): void {

        this.people.push(person);

    }

    getSchedule(): { totalPeople: number, uniquePositions: string[], uniqueTitles: string[] } {

        let totalPeople: number = this.people.length;

        let uniquePositions: string[] = [];

        let uniqueTitles: string[] = [];

        this.people.forEach(person => {

            if (uniquePositions.indexOf(person.positionName)===-1) {

                uniquePositions.push(person.positionName);

            }

            if (uniqueTitles.indexOf(person.titleName)===-1) {

                uniqueTitles.push(person.titleName);

            }

        });

        return {

            totalPeople,

            uniquePositions,

            uniqueTitles

        };

    }

}

const hr : HRDepartment = new HRDepartment();

const person1 : Person = new Person("Іван", "Петров", "1990-01-15", "Магістр", "Менеджер");

const person2 : Person = new Person("Марія", "Іванова", "1985-05-20", "Бакалавр","Програміст");

hr.addPerson(person1);

hr.addPerson(person2);

function addPerson(): void {

    let nameInput: HTMLInputElement | null = document.getElementById("personName") as HTMLInputElement;

    let surnameInput: HTMLInputElement | null = document.getElementById("personSurname") as HTMLInputElement;

    let birthdateInput: HTMLInputElement | null = document.getElementById("personBirthdate") as HTMLInputElement;

    let titleNameInput: HTMLInputElement | null = document.getElementById("titleName") as HTMLInputElement;

    let positionNameInput: HTMLInputElement | null = document.getElementById("positionName") as HTMLInputElement;

    if (nameInput && surnameInput && birthdateInput && titleNameInput && positionNameInput) {

        let name: string = nameInput.value;

        let surname: string = surnameInput.value;

        let birthdate: string = birthdateInput.value;

        let titleName: string = titleNameInput.value;

        let positionName: string = positionNameInput.value;

        let person: Person = new Person(name, surname, birthdate, titleName, positionName);

        hr.addPerson(person);

        clearInputFields2();

        alert("Людина додана до відділу кадрів.");

    } else {

        alert("Вхідні дані не можуть бути нульовими.");

    }

}

function clearInputFields2(): void {

    const nameInput: HTMLInputElement | null = document.getElementById("personName") as HTMLInputElement;

    const surnameInput: HTMLInputElement | null = document.getElementById("personSurname") as HTMLInputElement;

    const birthdateInput: HTMLInputElement | null = document.getElementById("personBirthdate") as HTMLInputElement;

    const titleNameInput: HTMLInputElement | null = document.getElementById("titleName") as HTMLInputElement;

    const positionNameInput: HTMLInputElement | null = document.getElementById("positionName") as HTMLInputElement;

    if (nameInput && surnameInput && birthdateInput && titleNameInput && positionNameInput) {

        nameInput.value = "";

        surnameInput.value = "";

        birthdateInput.value = "";

        titleNameInput.value = "";

        positionNameInput.value = "";

    } else {

        alert("Деякі поля вводу відсутні.");

    }

}

function displayHRInfo(): void {

    const hrInfo: HTMLElement | null = document.getElementById("hrInfo");

    if (hrInfo) {

        hrInfo.innerHTML = "";

        hrInfo.appendChild(createList("Люди:", hr.people));

        hrInfo.appendChild(createScheduleList("Штатний розклад:", hr.getSchedule()));

    } else {

        alert("Не було знайдено контейнер для відображення даних.");

    }

}

function createScheduleList(header: string, schedule: { totalPeople: number, uniquePositions: string[], uniqueTitles: string[] }): HTMLUListElement {

    const list: HTMLUListElement = document.createElement("ul");

    const headerElement: HTMLElement = document.createElement("strong");

    headerElement.textContent = header;

    list.appendChild(headerElement);

    const totalPeopleItem: HTMLLIElement = document.createElement("li");

    totalPeopleItem.textContent = "Загальна кількість людей: " + schedule.totalPeople;

    list.appendChild(totalPeopleItem);

    const uniquePositionsItem: HTMLLIElement = document.createElement("li");

    uniquePositionsItem.textContent = "Унікальні посади: " + schedule.uniquePositions.join(", ");

    list.appendChild(uniquePositionsItem);

    const uniqueTitlesItem: HTMLLIElement = document.createElement("li");

    uniqueTitlesItem.textContent = "Унікальні звання: " + schedule.uniqueTitles.join(", ");

    list.appendChild(uniqueTitlesItem);

    return list;

}

function createList(header: string, people: Person[]): HTMLUListElement {

    const list: HTMLUListElement = document.createElement("ul");

    const headerElement: HTMLElement = document.createElement("strong");

    headerElement.textContent = header;

    list.appendChild(headerElement);

    for (let i = 0; i < people.length; i++) {

        const person: Person = people[i];

        const listItem: HTMLLIElement = document.createElement("li");

        const personInfo: HTMLUListElement = document.createElement("ul");

        const nameItem: HTMLLIElement = document.createElement("li");

        nameItem.textContent = "Ім'я: " + person.name;

        personInfo.appendChild(nameItem);

        const surnameItem: HTMLLIElement = document.createElement("li");

        surnameItem.textContent = "Прізвище: " + person.surname;

        personInfo.appendChild(surnameItem);

        const birthdateItem: HTMLLIElement = document.createElement("li");

        birthdateItem.textContent = "Дата народження: " + person.birthdate;

        personInfo.appendChild(birthdateItem);

        const titleItem: HTMLLIElement = document.createElement("li");

        titleItem.textContent = "Звання: " + person.titleName;

        personInfo.appendChild(titleItem);

        const positionItem: HTMLLIElement = document.createElement("li");

        positionItem.textContent = "Посада: " + person.positionName;

        personInfo.appendChild(positionItem);

        listItem.appendChild(personInfo);

        list.appendChild(listItem);

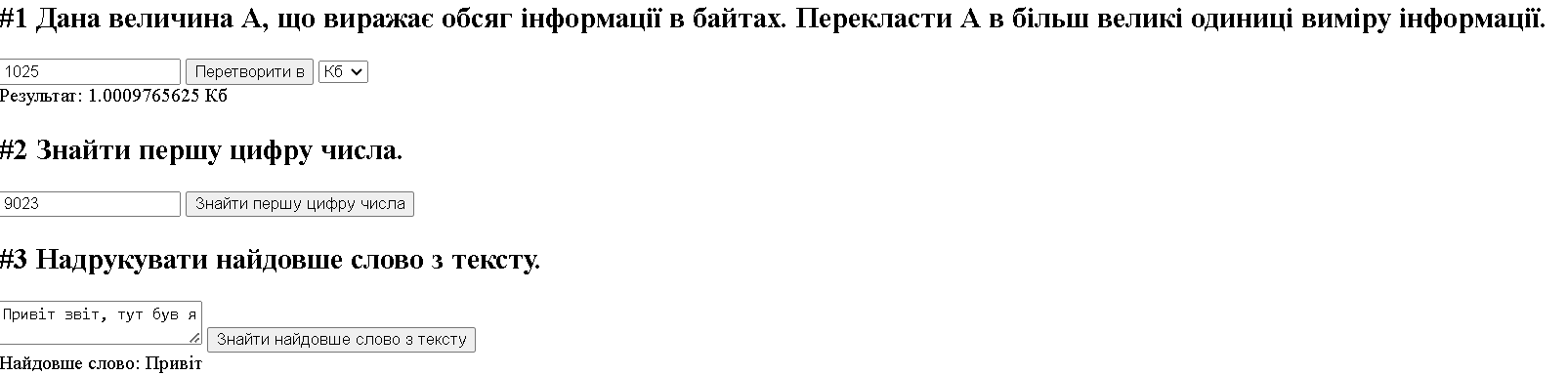
    }

    return list;

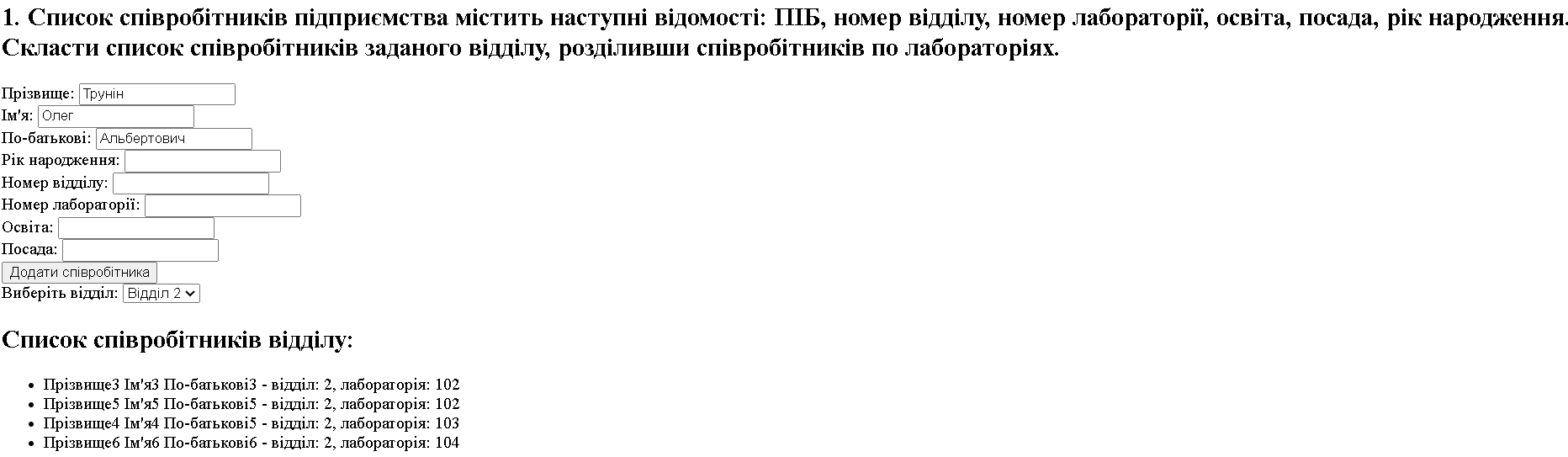
}

1. Тестуємо проєкт

Перевіряємо роботу функцій на сторінці, на прикладі перших трьох завдань:



Також перевіряємо роботу класів в 7 завданні:



**Висновок:** Я навчився розробляти веб-додатки на мові програмування TypeScript, навчився використовувати принципи ООП та класи, при розробці додатків, а також навчився користуватися базовими конструкціями мови TypeScript.