Міністерство освіти і науки України НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра ЕОМ



Звіт

з лабораторної роботи № 3

з дисципліни «Кросплатформенні засоби програмування» на тему: «Cпадкування та інтерфейси»

Виконав: ст. гр. КІ-302 Козак О.П.

Перевірив: викладач

Майдан М.В.

**Мета роботи:** ознайомитися з спадкуванням та інтерфейсами у мові Java.

# Завдання:

1. Написати та налагодити програму на мові Java, що розширює клас, що реалізований у лабораторній роботі №3, для реалізації предметної області

заданої варіантом. Суперклас, що реалізований у лабораторній роботі №3, зробити абстрактним. Розроблений підклас має забезпечувати механізми свого коректного функціонування та реалізовувати мінімум один інтерфейс.

Програма має розміщуватися в пакеті Група.Прізвище.Lab4 та володіти коментарями, які дозволять автоматично згенерувати документацію до розробленого пакету.

1. Автоматично згенерувати документацію до розробленого пакету.
2. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації.
3. Дати відповідь на контрольні запитання.

**Варіант:** 10

# Виконання завдання:

*Лістинг 1*

abstract class House {

    address: string;

    numberOfFloors: number;

    hasGarden: boolean;

    constructor() {

        this.address = "No information";

        this.numberOfFloors = 0;

        this.hasGarden = false;

    }

    setAddress(address: string) {

        this.address = address;

    }

    setNumberOfFloors(numberOfFloors: number) {

        this.numberOfFloors = numberOfFloors;

    }

    setHasGarden(hasGarden: boolean) {

        this.hasGarden = hasGarden;

    }

    abstract displayDetails(): void;

}

class OfficeCenter extends House {

    officeSpace: number;

    meetingRoom: boolean;

    numberOfDesks: number;

    hasProjector: boolean;

    hasWhiteboard: boolean;

    constructor(address: string, numberOfFloors: number, officeSpace: number, meetingRoom: boolean, numberOfDesks: number, projector: boolean, whiteboard: boolean) {

        super();

        this.officeSpace = officeSpace;

        this.meetingRoom = meetingRoom;

        this.numberOfDesks = numberOfDesks;

        this.hasProjector = projector;

        this.hasWhiteboard = whiteboard;

    }

    setOfficeSpace(officeSpace: number) {

        this.officeSpace = officeSpace;

    }

    equipMeetingRoom(meetingRoom: boolean, projector: boolean, whiteboard: boolean) {

        this.meetingRoom = true;

        this.hasProjector = projector;

        this.hasWhiteboard = whiteboard;

    }

    addDesks(desksToAdd: number) {

        this.numberOfDesks = desksToAdd;

    }

    getNumberOfDesks() {

        return this.numberOfDesks;

    }

    hasProjectorEquipment() {

        return this.hasProjector;

    }

    hasWhiteboardEquipment() {

        return this.hasWhiteboard;

    }

    displayDetails() {

        let officeCenterDetails = document.getElementById("officeCenterDetails");

        if (officeCenterDetails) {

            officeCenterDetails.innerHTML = "<h2>Деталі офісного центру</h2>";

            officeCenterDetails.innerHTML += "<p>Адреса: " + this.address + "</p>";

            officeCenterDetails.innerHTML += "<p>Кількість поверхів: " + this.numberOfFloors + "</p>";

            officeCenterDetails.innerHTML += "<p>Площа офісу (кв. м): " + this.officeSpace + "</p>";

            officeCenterDetails.innerHTML += "<p>Має кімнату для зборів: " + (this.meetingRoom ? "Так" : "Ні") + "</p>";

            officeCenterDetails.innerHTML += "<p>Кількість робочих столів: " + this.getNumberOfDesks() + "</p>";

            officeCenterDetails.innerHTML += "<p>Має проектор: " + (this.hasProjectorEquipment() ? "Так" : "Ні") + "</p>";

            officeCenterDetails.innerHTML += "<p>Має маркерну дошку: " + (this.hasWhiteboardEquipment() ? "Так" : "Ні") + "</p>";

        }

    }

}

let officeCenter = new OfficeCenter("", 0, 0, false, 0, false, false);

function updateOfficeCenter() {

    let address = document.getElementById("address") as HTMLInputElement;

    let floors = parseInt((document.getElementById("floors") as HTMLInputElement).value);

    let officeSpace = parseInt((document.getElementById("officeSpace") as HTMLInputElement).value);

    let meetingRoom = (document.getElementById("meetingRoom") as HTMLInputElement).checked;

    let desks = parseInt((document.getElementById("desks") as HTMLInputElement).value);

    let projector = (document.getElementById("projector") as HTMLInputElement).checked;

    let whiteboard = (document.getElementById("whiteboard") as HTMLInputElement).checked;

    officeCenter.setAddress(address.value);

    officeCenter.setNumberOfFloors(floors);

    officeCenter.setOfficeSpace(officeSpace);

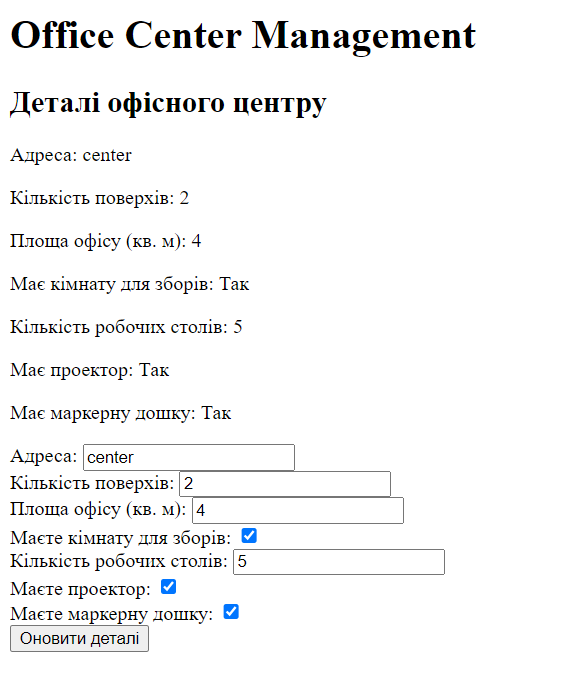
    officeCenter.equipMeetingRoom(meetingRoom, projector, whiteboard);

    officeCenter.addDesks(desks);

    officeCenter.displayDetails();

}

Результат роботи програми:



**Висновок:** у ході виконання лабораторної роботи було вивчено поняття спадкування та інтерфейсів у мові Javascript. Було розроблено дочірній клас та імплементовано у ньому оголошені dispalyDetails.