Міністерство освіти і науки України

Національний університет „Львівська політехніка”



**Звіт**

з лабораторної роботи №1

з дисципліни: “Кросплатформні засоби

програмування”

на тему: “СПАДКУВАННЯ ТА ІНТЕРФЕЙСИ”

Варіант 14

Виконав: ст. гр. КІ-301

Матвіїшин М.З

Прийняв:

Майдан М.В

Львів – 2023

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

**ТЕОРЕТИЧНІ ВІДОМОСТІ**

Спадкування Спадкування в ООП призначене для розширення функціональності існуючих класів шляхом утворення нових класів на базі вже існуючих. У Java реалізована однокоренева архітектура класів згідно якої всі класи мають єдиного спільного предка (кореневий клас в ієрархії класів) – клас Object. Решта класів мови Java утворюються шляхом успадковування даного класу. Будь-яке спадкування у мові Java є відкритим, при цьому аналогів захищеному і приватному спадкуванню мови С++ не існує. На відміну від С++ у Java можливе спадкування лише одного базового класу (множинне спадкування відсутнє).

Поліморфізм

Механізм поліморфізму забезпечує можливість присвоєння об’єктним змінним суперкласу об’єктів похідних класів та звертання з-під цих змінних до перевизначених у підкласі членів суперкласу. У Java всі об’єктні змінні є поліморфними. Поліморфізм реалізується за допомогою механізму динамічного (пізнього) зв’язування, який полягає у тому, що вибір методу, який необхідно викликати, відбувається не на етапі компіляції, а під час виконання програми.

Приведення об’єктних типів

Приведення типів у Java відбувається вказуванням у круглих дужках перед змінною, яку необхідно привести до іншого типу, типу до якого її необхідно привести.

Абстрактні класи

Абстрактні класи призначені бути основою для розробки ієрархій класів та не дозволяють створювати об’єкти свого класу. Вони реалізуються за допомогою ключового слова abstract. На відміну від звичайних класів абстрактні класи можуть містити абстрактні методи (а можуть і не містити). Абстрактні методи – це методи, що 6 оголошені з використанням ключового слова abstract і не місять тіла. Розширюючи абстрактний клас можна залишити деякі або всі методи невизначеними.

**ЗАВДАННЯ**

1. Написати та налагодити програму на мові Java, що розширює клас, що реалізований у

лабораторній роботі №2, для реалізації предметної області заданої варіантом.

Суперклас, що реалізований у лабораторній роботі №2, зробити абстрактним.

Розроблений підклас має забезпечувати механізми свого коректного функціонування

та реалізовувати мінімум один інтерфейс. Програма має розміщуватися в пакеті

Група.Прізвище.Lab3 та володіти коментарями, які дозволять автоматично

згенерувати документацію до розробленого пакету.

2. Автоматично згенерувати документацію до розробленого пакету.

3. Завантажити код на GitHub згідно методичних вказівок по роботі з GitHub.

4. Скласти звіт про виконану роботу з приведенням тексту програми, результату її

виконання та фрагменту згенерованої документації та завантажити його у ВНС.

5. Дати відповідь на контрольні запитання.

**Хід роботи**

Мій варіант 14 – Телевізор з тюнером

Лістинг програми

TV.java

package MatviishynLB3;

import java.io.\*;

// Інтерфейс для відображення інформації

interface LB3 {

void displayInfo();

}

// Абстрактний клас електронного пристрою

abstract class ElectronicDevice implements LB3 {

private boolean isOn;

public ElectronicDevice() {

isOn = false;

}

public void turnOn() {

isOn = true;

}

public void turnOff() {

isOn = false;

}

public boolean isOn() {

return isOn;

}

*@Override*

public abstract void displayInfo();

public abstract void dispose();

}

// Клас екрану

class Screen {

private boolean isOn;

public Screen() {

isOn = false;

}

public void turnOn() {

isOn = true;

}

public void turnOff() {

isOn = false;

}

public boolean isOn() {

return isOn;

}

}

// Клас тюнера

class Tuner extends ElectronicDevice {

private String[] tvChannels;

private int currentChannel;

public Tuner() {

tvChannels = new String[]{

"1+1",

"National Geographic",

"Novyy",

"Films",

"HBO",

"BBC News",

"Cartoon Network",

"History Channel",

"Food Network",

"Netflix"

};

currentChannel = 0;

}

public void changeChannel(int newChannel) {

if (newChannel >= 0 && newChannel < tvChannels.length) {

currentChannel = newChannel;

System.***out***.println("TV channel is changed");

getCurrentChannel();

} else {

System.***out***.println("Invalid channel number");

}

}

public String getCurrentChannel() {

String currentChannelName = tvChannels[currentChannel];

System.***out***.println("Current TV channel: " + currentChannelName);

return currentChannelName;

}

*@Override*

public void displayInfo() {

System.***out***.println("Tuner is " + (isOn() ? "on" : "off"));

getCurrentChannel();

}

*@Override*

public void dispose() {

System.***out***.println("Tuner is being disposed.");

}

}

// Клас телевізора

class TV extends ElectronicDevice {

private Screen screen;

private Tuner tuner;

private int currentVolume;

private PrintWriter fout;

public TV() throws FileNotFoundException {

screen = new Screen();

tuner = new Tuner();

currentVolume = 50;

try {

fout = new PrintWriter(new File("TVLog.txt"));

} catch (FileNotFoundException e) {

System.***err***.println("Could not create TVLog.txt");

}

}

public void changeChannel(int newChannel) {

tuner.changeChannel(newChannel);

System.***out***.println("Channel is changed");

}

public void adjustVolume(int delta) {

currentVolume += delta;

if (currentVolume < 0) {

currentVolume = 0;

} else if (currentVolume > 100) {

currentVolume = 100;

}

System.***out***.println("Volume is changed");

getCurrentVolume();

}

public int getCurrentVolume() {

System.***out***.println("Current volume: " + currentVolume);

return currentVolume;

}

*@Override*

public void displayInfo() {

System.***out***.println("TV is " + (isOn() ? "on" : "off"));

tuner.displayInfo();

}

*@Override*

public void dispose() {

tuner.dispose();

if (fout != null) {

fout.close();

}

}

}

MatviishynLB.java

package MatviishynLB3;

import java.io.FileNotFoundException;

public class MatviishynLB {

public static void main(String[] args) {

try {

TV tv = new TV();

tv.turnOn();

tv.displayInfo();

tv.changeChannel(2);

tv.adjustVolume(10);

tv.displayInfo();

tv.turnOff();

tv.displayInfo();

tv.dispose();

} catch (FileNotFoundException e) {

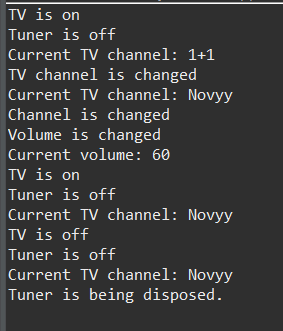
e.printStackTrace();

}

}

}

**Результат виконання програми**



**Висновок**

На цій лабораторній роботі я ознайомився та навчився використовувати спадкування та інтерфейси на практиці використовуючи мову Java.