

Міністерство освіти і науки України  
Національний університет «Львівська політехніка»  
Кафедра МКС



## **Звіт**

З лабораторної роботи №4

З дисципліни: «Кросплатформні засоби програмування»

На тему:

«Інтерфейси»

Виконав:  
ст.гр. КІ-35

Куденчук Владислав

Прийняв:  
Іванов Ю. С.

**Львів 2022**

Мета: ознайомитися з процесом розробки класів та пакетів мовою Java.

### ЗАВДАННЯ

1. Написати та налагодити програму на мові Java, що реалізує у вигляді класу предметну область згідно варіанту. Програма має задовольняти наступним вимогам:
  - програма має розміщуватися в пакеті `Група.Прізвище.Lab3`;
  - клас має містити мінімум 3 поля, що є об'єктами класів, які описують складові частини предметної області;
  - клас має містити кілька конструкторів та мінімум 10 методів;
  - для тестування і демонстрації роботи розробленого класу розробити клас-драйвер;
  - методи класу мають вести протокол своєї діяльності, що записується у файл;
  - розробити механізм коректного завершення роботи з файлом (не надіятися на метод `finalize()`);
  - програма має володіти коментарями, які дозволять автоматично згенерувати документацію до розробленого пакету.
2. Автоматично згенерувати документацію до розробленого пакету.
3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації.
4. Дати відповідь на контрольні запитання.

## 17. Відеомагнітофон

Виконання:

Код:

```
package KI_35_Kudenchuk;

public class main {
    /**
     * @author Kudenchuk Vladyslav
     * @version 1.0
     * /
     /** @throws "FileNotFoundException"
     * @return void
     * Creation of a variable for classes made, using its methods
     */
    public static void main(String[] args) {
        VideoRecorder vr1 = new VideoRecorder(new Sound(20),new Screen(7));
        VideoRecorder vr2 = new VideoRecorder(new Sound(20),new Screen(6.7));
        VideoRecorder vr3 = new VideoRecorder(new Sound(15),new Screen(6.3));
        VideoPlayer vp1 = new VideoPlayer(new Sound(20),new Screen(6.7));
        VideoPlayer vp2 = new VideoPlayer(new Sound(20),new Screen(7));
        VideoPlayer vp3 = new VideoPlayer(new Sound(17),new Screen(6.5));
        vp1.play();
        vp2.review();
        vp3.setIx(6.6);
        System.out.println(vp3.br.area);

        vr1.record();
    }
}
```

## Class VideoPlayer

```
package KI_35_Kudenchuk;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.PrintWriter;

public class VideoPlayer{

    /**
     * Creation of a file to write information
     */
    private static final File dataFile = new File("log.txt");
    PrintWriter pw;
    {
        try {
            pw = new PrintWriter(dataFile);
        } catch (FileNotFoundException e) {
            throw new RuntimeException(e);
        }
    }

    /**
     * Creation of new objects
     */
    Sound kt = new Sound();
    Screen br = new Screen();

    /**
     * "Putting" Screen and Sound into a box
     * @param kt
     * @param ix
     */
    public VideoPlayer(Sound kt, Screen br) {
        this.kt = kt;
        this.br = br;
    }

    /**
     * Gets Sound variables
     * @return kt
     */
    public Sound getKt() {
        return kt;
    }

    /**
     * Sets Sound variables
     * @param kt
     */
    public void setKt(Sound kt) {
```

```

        this.kt = kt;
    }
    /**
     * Gets Sound variables
     * @return br
     */
    public Screen getIx() {
        return br;
    }
    /**
     * Sets Screen variables
     * @param d
     */
    public void setIx(double d) {
        this.br.area = d;
    }
    /**
     * Gets Sound variables
     * @throws FileNotFoundException
     */
    public VideoPlayer() throws FileNotFoundException {
    }

    public void play() {
        System.out.println("VideoPlayer sounds...");
    }

    public void nextSong() {
        System.out.println("start playing next song");
    }

    /**
     * Methods that print information into file
     * @param message
     * @throws FileNotFoundException
     */
    protected void logActivity(String message) throws FileNotFoundException
    {
        pw.println(message);
        pw.flush();
    }

    /**
     * Method for display of showing and printing info in file
     *
     * @throws FileNotFoundException
     */
    public void show() throws FileNotFoundException {
        System.out.println("showing screen");
        try {
            logActivity("method show() was called");
        } catch (FileNotFoundException e) {
            throw new RuntimeException(e);
        }
    }
    /**
     * Method for notification of reviewing and printing info in file

```

```

        * @throws FileNotFoundException
        */
    public void review(){
        System.out.println("reviewing case");
        try {
            logActivity("method review() was called");
        } catch (FileNotFoundException e) {
            throw new RuntimeException(e);
        }
    }
}

```

## Class Screen

```

package KI_35_Kudenchuk;

public class Screen {
    /**
     * @param area of Screen
     */
    double area =0;

    /**
     * Constructor of Screen
     * @param area of Screen
     */
    public Screen(double area) {
        this.area = area;
    }
    public Screen() {

    }
    /**
     * Gets the area of an Screen
     */
    public void mainScreen() {
        System.out.println("returned to main screen");
    }

    public double getArea() {
        return area;
    }

    /**
     * Sets the area of an Screen
     * @param area of Screen
     */
    public void setArea(double area) {
        this.area = area;
    }
}

```

## Class Sound

```
package KI_35_Kudenchuk;

public class Sound {
    /**
     * @param loudness of Sound
     */
    double loudness;

    /**
     * Constructor of Sound
     * @param loudness
     */
    public Sound(double loudness) {
        this.loudness = loudness;
    }

    public Sound() {
    }

    public void checkSound() {
        System.out.println("all is working fine");
    }

    /**
     *
     * @return loudness of Sound
     */
    public double getLoudness() {
        return loudness;
    }

    /**
     * Sets the loudness of Sound
     * @param loudness
     */
    public void setLoudness(double loudness) {
        this.loudness = loudness;
    }
}
```

## Class VideoRecorder

```
package KI_35_Kudenchuk;

import java.io.FileNotFoundException;

/**
 * Creation of a variable for classes made, using its methods
 * @author Kachur Vladyslav
 * @version 1.0
 */

public class VideoRecorder extends VideoPlayer implements Record {
```

```

        VideoPlayer a = new VideoPlayer(new Sound(35.5), new Screen(20));

    /**
     * Making default instruments military
     * @param hk Hook
     * @param ix IceAxe
     */
    public VideoRecorder(Sound hk, Screen ix) {
        super(hk, ix);
    }

    /**
     * Default constructor of MilitaryAlpinistInventory
     * @throws FileNotFoundException
     */
    public VideoRecorder() throws FileNotFoundException {
        super();
    }

    /**
     * Method hide
     */
    @Override
    public void record() {
        try {
            a.logActivity("recording");
        } catch (FileNotFoundException e) {
            throw new RuntimeException(e);
        }
        System.out.println("recording");
    }
}

```

## Interface Record

```

package KI_35_Kudenchuk;

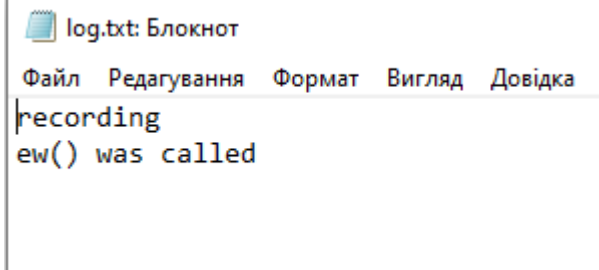
public interface Record {
    void record();
}

```

Console:

```
VideoPlayer sounds...  
reviewing case  
6.6  
recording
```

Text file:



log.txt: Блокнот

Файл Редагування Формат Вигляд Довідка

```
recording  
ew() was called
```

### Відповіді на контрольні запитання

1. Синтаксис реалізації спадкування.  
class Підклас extends Суперклас  
  
{  
  
Додаткові поля і методи  
  
}
2. Що таке суперклас та підклас?  
В термінах мови Java базовий клас найчастіше називається суперкласом, а похідний клас – підкласом.
3. Як звернутися до членів суперкласу з підкласу?  
public НазваПідкласу([параметри])  
  
{  
  
super([параметри]);  
  
оператори конструктора підкласу  
  
}

**Висновок:** ознайомився з процесом розробки класів та пакетів мовою Java.