

Міністерство освіти і науки України
Національний університет «Львівська політехніка»

Кафедра ЕОМ



Звіт

до лабораторної роботи № 4

з дисципліни: «Кросплатформні засоби програмування»

«Спадкування та інтерфейси»

Варіант - 17

Виконав: ст. гр. КІ-34

Панасюк М.Т.

Прийняв:

Іванов Ю

Мета: 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.
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.

Варіант завдання:

17. Відеомагнітофон

Код програми:

File Main.java

```
package lab3;

import java.util.ArrayList;

public class Main {
    public static void main(String[] argc) throws Exception{
        VideoPlayer videoPlayer = new VideoPlayer();

        videoPlayer.turnOn();

        ArrayList<Frame> video = new ArrayList<Frame>();

        for (int i = 0; i < 40; i++) {
            ArrayList<Pixel> pixels = new ArrayList<Pixel>();

            for(int j = 0; j < 50; j++) {
                pixels.add(new Pixel((int) ( Math.random() * 255 ),(int) ( Math.random() * 255 ),(int) ( Math.random() * 255
            ));
            }

            video.add(new Frame(pixels));
        }

        videoPlayer.writeIntoMemory(video);
    }
}
```

```

        videoPlayer.playVideo();

        videoPlayer.turnOff();

        videoPlayer.dispose();
    }
}

```

File VideoRecorder.java

```

package lab4;

import java.io.FileNotFoundException;
import java.util.ArrayList;

interface IVideoRecorder {
    void record();
}

public class VideoRecorder extends VideoPlayer {

    public VideoRecorder() throws FileNotFoundException {
    }

    /**
     * Write video to memory
     * @param video
     */
    public void record(ArrayList<Frame> video) {
        video.forEach((frame -> {
            super.memory.write(frame);
        }));
    }

    /**
     * Method for video play
     * @throws Exception
     */
    public void play() throws Exception {
        super.playVideo();
    }
}

```

File VideoPlayer.java

```

package lab3;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.concurrent.atomic.AtomicInteger;

```

```

public class VideoPlayer {

    private PrintWriter fout;
    private final String FILE = "lab3.txt";
    private boolean isOn = false;

    private Memory memory;

    public VideoPlayer() throws FileNotFoundException {
        fout = new PrintWriter(new File(FILE));
        memory = new Memory();
    }
    /**
     * Write video to memory
     * @param frames
     */
    public void writeIntoMemory(ArrayList<Frame> frames) {
        frames.forEach((frame -> {
            memory.write(frame);
        }));
    }
    /**
     * Return frame by index from memory
     * @param index - frame index
     */
    public Frame readFromMemory(int index) throws Exception{

        if(index >= memory.getSize()) {
            throw new Exception("Wrong index");
        }

        return memory.read(index);
    }
    /**
     * Method for video play
     * @throws Exception
     */
    public void playVideo() throws Exception{
        if(!isOn) {
            throw new Exception("You need to turn on the video player");
        }

        memory.getContent().forEach((frame -> {
            ShowFrame(frame);
        }));
    }
    /**
     * Method show frame
     * @param frame
     */
    public void ShowFrame(Frame frame) {
        AtomicInteger i = new AtomicInteger();
        frame.getContent().forEach((pixel) -> {
            if(i.get() == frame.getFRAME_WIDTH()) {
                i.set(0);
                printMessage("\n");
            }else {
                printMessage(pixel.getInfo());
                i.getAndIncrement();
            }
        });
    }
    /**
     * Method turns on the VideoPlayer

```

```

    */
    public void turnOn() {
        if (isOn) {
            return;
        }
        isOn = true;
        printMessage("VideoPlayer is turned on");
    }
    /**
     * Method turns off the VideoPlayer
     */
    public void turnOff()
    {
        if(isOn)
        {
            isOn = false;
            printMessage("VideoPlayer is turned off");
        }
    }
    public void dispose()
    {
        fout.flush();
        fout.close();
    }

    private void printlnMessage(String message)
    {
        fout.println(message);
        System.out.println(message);
    }
    private void printMessage(String message)
    {
        fout.print(message);
        System.out.print(message);
    }
}

```

File Pixel.java

```

package lab3;

public class Pixel {
    public int red;
    public int green;
    public int blue;

    public Pixel(int r, int g, int b) {
        this.red = r;
        this.green = g;
        this.blue = b;
    }
    /**
     * Returns info about the pixel
     */
    public String getInfo() {
        return "(" + red + "," + green + "," + blue + ")";
    }
}

```

File Memory.java

```
package lab3;

import java.util.ArrayList;

public class Memory {

    private ArrayList<Frame> content;

    public Memory() {
        content = new ArrayList<Frame>();
    }
    /**
     * Write frame to memory
     * @param frame
     */
    public void write(Frame frame) {
        content.add(frame);
    }
    /**
     * Read frame from memory
     */
    public Frame read(int index) {
        return content.get(index);
    }
    /**
     * Returns content size
     */
    public int getSize() {
        return content.size();
    }
    /**
     * Returns content
     */
    public ArrayList<Frame> getContent() {
        return content;
    }
}
```

File Frame.java

```
package lab3;

import java.util.ArrayList;

public class Frame {

    private ArrayList<Pixel> content;
    private int FRAME_WIDTH = 5;

    public Frame(ArrayList<Pixel> content) {
        this.content = content;
    }
    /**
     * Returns frame width
     */
    public int getFRAME_WIDTH() {
```

```

    return FRAME_WIDTH;
}
/**
 * Returns frame content
 */
public ArrayList<Pixel> getContent() {
    return content;
}
}

```

Результат виконання програми:

```

(187;179;189)(179;14;21)(223;148;82)(77;98;99)(11;222;80)
(108;37;54)(65;1;95)(164;175;176)(183;242;86)(22;212;230)
(82;15;127)(111;248;117)(75;78;129)(161;38;152)(54;52;44)
(245;172;246)(33;83;184)(127;189;73)(3;43;237)(46;157;124)
(140;9;123)(35;7;97)(98;162;58)(74;54;247)(189;27;135)(30;168;252)(179;100;96)
(38;54;244)(117;162;74)(127;167;118)(213;124;201)(194;176;146)
(56;183;76)(59;65;206)(89;91;82)(252;154;8)(124;185;228)
(1;33;91)(72;75;7)(209;169;179)(127;72;129)(60;65;178)
(181;35;7)(38;235;66)(232;210;190)(220;116;190)(174;120;182)
(147;216;208)(117;187;58)(151;225;73)(253;30;101)(134;109;73)
(242;215;104)(67;147;212)(184;47;44)(177;192;170)(0;240;246)
(235;36;212)(9;77;96)(82;193;57)(50;70;4)(128;241;69)
(253;137;123)(175;42;129)(51;62;174)(239;152;171)(1;117;14)(246;151;24)(145;155;46)
(40;148;253)(17;36;62)(16;104;55)(240;192;239)(206;127;183)
(193;95;2)(12;112;192)(243;223;186)(209;128;222)(48;166;43)
(107;9;215)(127;146;34)(94;103;27)(50;70;22)(197;206;187)
(242;254;1)(146;1;155)(45;184;100)(139;32;241)(121;2;7)
(151;167;200)(129;174;70)(34;52;16)(111;251;83)(242;234;78)
(156;207;149)(190;95;9)(117;160;185)(76;173;47)(200;42;101)
(213;9;19)(246;90;181)(129;17;140)(197;164;107)(60;86;124)
(73;192;138)(173;209;141)(212;65;37)(223;82;187)(178;156;200)(32;226;38)(61;92;123)
(71;78;138)(112;60;81)(185;247;124)(183;60;249)(41;39;232)
(198;243;51)(246;57;134)(23;250;16)(200;43;146)(202;155;197)
(79;226;200)(108;231;63)(182;59;36)(74;199;249)(186;166;97)
(24;193;236)(173;88;48)(6;156;84)(192;57;184)(132;67;126)
(166;37;132)(186;227;139)(113;185;7)(42;80;213)(135;220;152)
(119;243;27)(15;121;33)(203;234;247)(66;146;8)(92;246;219)
(77;184;6)(201;73;143)(248;169;173)(148;126;158)(170;134;154)
(135;244;135)(96;225;232)VideoPlayer is turned off
Process finished with exit code 0

```

Constructor Summary

Constructors

Constructor	Description
<code>Memory()</code>	

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
<code>ArrayList<Frame></code>	<code>getContent()</code>	Returns content
<code>int</code>	<code>getSize()</code>	Returns content size
<code>Frame</code>	<code>read(int index)</code>	Read frame from memory
<code>void</code>	<code>write(Frame frame)</code>	Write frame to memory

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Висновок: на цій лабораторній роботі я ознайомився з спадкуванням та інтерфейсами у мові Java.