Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра МКС

Звіт

З лабораторної роботи №3

З дисципліни: «Кросплатформні засоби програмування»

На тему:

«КЛАСИ ТА ПАКЕТИ»

Виконав:
ст.гр. КІ-35

Куденчук Владислав

Пр
ийняв:

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. Відеоплеєр

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;
    }
}
```

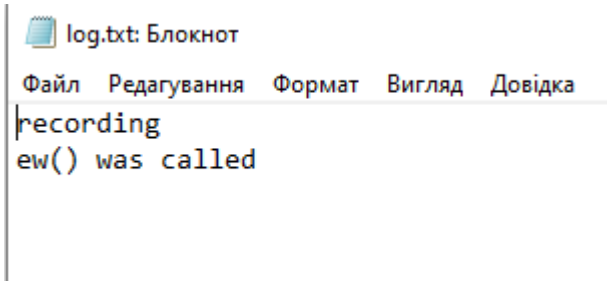
Class Sound

```
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;
    }
}
```

Console:

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

Text file:



Відповіді на контрольні запитання

1. Синтаксис визначення класу.
   [public] class НазваКласу

   {

   [конструктори] [методи] [поля]

   }

2. Синтаксис визначення методу.
   [СпецифікаторДоступу] [static] [final] Тип назваМетоду([параметри]) [throws класи]

   {

   [Тіло методу] [return [значення]];

   }

3. Синтаксис оголошення поля.
   [СпецифікаторДоступу] [static] [final] Тип НазваПоля [= ПочатковеЗначення];

4. Як оголосити та ініціалізувати константне поле?
   Приклад оголошення поля: private int i;

   Приклад оголошення константного поля: private final int i;

5. Які є способи ініціалізації полів?
   - явно при оголошені поля класу;
   - у статичному блоці ініціалізації

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