Національний університет «Львівська Політехніка»

Інститут комп’ютерних технологій, автоматики та метрології

Кафедра електронних обчислювальних машин



Звіт

Про виконання лабораторної роботи №2

### З дисципліни «Кросплатформлені засоби програмування»

**Виконав:**

студент групи КІ-305

Рудий В.В.

**Перевірив:**

Доцент кафедри ЕОМ

Іванов Ю. С.

Львів – 2023

**Тема:** Класи та пакети.

**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.Lab2;  
 • 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 розробленої програми.  
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.

**Завдання згідно варіанту:**

***Варіант: 23 Завдання: Плитка для приготування їжі***

**Виконання:**

**Код програми:**

***Lab2RudyiKI305.java:***

package pack;

import java.io.FileNotFoundException;

/\*\*

\* Class App realize example program to lab work 2

\*

\* @author Rudyi Volodymyr

\* @version 1.1

\* @since version 1.0

\*/

public class Lab2RudyiKI305 {

/\*\*

\* @param args

\* @throws FileNotFoundException

\*/

public static void main(String[] args) throws Exception {

StoveForCooking stove1 = new StoveForCooking(23232232);

stove1.getInfoLook();

stove1.repaintStoveForCooking("Red");

stove1.resizeStoveForCooking(0.6, 0.6, 0.8);

stove1.getInfoLook();

stove1.getBurnerArea();

stove1.PowerUpButtonResource();

for (int i = 0; i < 5; i++) {

stove1.powerUp();

}

for (int i = 0; i < 4; i++) {

stove1.powerDown();

}

stove1.PowerDownButtonResource();

stove1.powreMin();

stove1.powerMax();

}

}

***StoveForCooking.java:***

package pack;

import java.io.\*;

/\*\*

\* Class <code>StoveForCooking</code> implements stove for cooking

\*/

public class StoveForCooking {

private Type TypeOfStoveForCooking;

private Burner BurnerOfStoveForCooking;

private Button PowerUpButton;

private Button PowerDownButton;

private PrintWriter fout;

static int power;

/\*\*

\* Constructor

\*

\* @throws FileNotFoundException

\*/

public StoveForCooking() throws FileNotFoundException {

TypeOfStoveForCooking = new Type();

BurnerOfStoveForCooking = new Burner();

PowerUpButton = new Button();

PowerDownButton = new Button();

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

}

/\*\*

\* Constructor

\*

\* @param <code>resource</code> Button clicks resource

\* @throws FileNotFoundException

\*/

public StoveForCooking(int resource) throws FileNotFoundException {

TypeOfStoveForCooking = new Type();

BurnerOfStoveForCooking = new Burner();

PowerUpButton = new Button(resource);

PowerDownButton = new Button(resource);

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

}

}

/\*\*

\* Method change color of stove for cooking

\*/

public void repaintStoveForCooking(String uCollor) {

TypeOfStoveForCooking.setColor(uCollor);

System.out.println("New color of audio player is " + uCollor + ".");

fout.println("New color of audio player is " + uCollor + ".");

fout.flush();

}

/\*\*

\* Method change color of stove for cooking

\*/

public void resizeStoveForCooking(double uLength, double uWidth, double uHeight) {

TypeOfStoveForCooking.setSize(uLength, uWidth, uHeight);

System.out.println("New size of stove for cooking is " + uLength + "x" + uWidth + "x" + uHeight + " m.");

fout.println("New color of audio player is " + uLength + "x" + uWidth + "x" + uHeight + " .");

fout.flush();

}

/\*\*

\* Method provides information about the appearance of the stove for cooking

\*/

public void getInfoLook() {

System.out.println("Size of stove for cooking: " + TypeOfStoveForCooking.getSize() + " m^2.");

System.out.println("Weight of stove for cooking: " + TypeOfStoveForCooking.getWeight() + " kg.");

System.out.println("Color of stove for cooking: " + TypeOfStoveForCooking.getColor() + ".");

fout.println("Size of stove for cooking: " + TypeOfStoveForCooking.getSize() + " m^2.");

fout.println("Weight of stove for cooking: " + TypeOfStoveForCooking.getWeight() + " kg.");

fout.println("Color of stove for cooking: " + TypeOfStoveForCooking.getColor() + ".");

fout.flush();

}

/\*\*

\* Method provides information about the area of burner

\*/

public void getBurnerArea() {

System.out.println("Burner occupy " + BurnerOfStoveForCooking.getArea() + " cm^2 of size.");

fout.println("Burner occupy " + BurnerOfStoveForCooking.getArea() + " cm^2 of size.");

fout.flush();

}

/\*\*

\* Method provides information about the count of clicks of power up button

\*/

public void PowerUpButtonResource() {

System.out.println("You can tap the power up button " + PowerUpButton.getResource() + " more times.");

fout.println("You can tap the power up button " + PowerUpButton.getResource() + " more times.");

fout.flush();

}

/\*\*

\* Method provides information about the count of clicks of power down button

\*/

public void PowerDownButtonResource() {

System.out.println("You can tap the power down button " + PowerDownButton.getResource() + " more times.");

fout.println("You can tap the power down button " + PowerDownButton.getResource() + " more times.");

fout.flush();

}

/\*\*

\* Method simulate the power up button tap

\*/

public void powerUp() {

PowerUpButton.clickButton(true);

System.out.println("You tap the power up button.");

fout.println("You tap the power up button.");

fout.flush();

if (BurnerOfStoveForCooking.getPower() > 100) {

BurnerOfStoveForCooking.setPower(100);

}

}

/\*\*

\* Method simulate the power up button tap

\*/

public void powerDown() {

PowerDownButton.clickButton(false);

System.out.println("You tap the power down button.");

fout.println("You tap the power down button.");

fout.flush();

if (BurnerOfStoveForCooking.getPower() < 0) {

BurnerOfStoveForCooking.setPower(0);

}

}

/\*\*

\* Method simulate the power down button press

\*/

public void powreMin() {

PowerDownButton.clickButton(false);

System.out.println("You press the power down button.");

fout.println("You press the power down button.");

fout.flush();

BurnerOfStoveForCooking.setPower(0);

}

/\*\*

\* Method simulate the power up button press

\*/

public void powerMax() {

PowerUpButton.clickButton(false);

System.out.println("You press the power up button.");

fout.println("You press the power up button.");

fout.flush();

BurnerOfStoveForCooking.setPower(100);

}

}

/\*\*

\* Class <code>Type</code> implements type of stove for cooking

\*

\* @author Rudyi Volodymyr

\* @version 1.0

\*/

class Type {

private String color;

private double length, width, height;

/\*\*

\* Constructor

\*/

public Type() {

color = "White";

length = 0.5;

width = 0.6;

height = 0.8;

}

/\*\*

\* Method sets new color of type of stove for cooking

\*

\* @param <code>newColor</code> The color value

\*/

public void setColor(String newColor) {

color = newColor;

}

/\*\*

\* Method sets new size of type of stove for cooking

\*

\* @param <code>newSize</code> The Size value

\*/

public void setSize(double newLength, double newWidth, double newHeight) {

length = newLength;

width = newWidth;

height = newHeight;

}

/\*\*

\* Method gets color of type of stove for cooking

\*/

public String getColor() {

return color;

}

/\*\*

\* Method gets size of type of stove for cooking in m^2

\*/

public double getSize() {

return length \* width;

}

/\*\*

\* Method gets weight of type of stove for cooking in kilograms

\*/

public double getWeight() {

return length \* width \* height \* 166.6;

}

}

/\*\*

\* Class <code>Burner</code> implements burner of stove for cooking

\*

\* @author Rudyi Volodymyr

\* @version 1.0

\*/

class Burner {

private double diametr, power;

/\*\*

\* Constructor

\*/

public Burner() {

diametr = 15;

power = 50;

}

/\*\*

\* Constructor

\*

\* @param <code>uLoudness</code> The power value

\*/

public Burner(int uLoudness) {

diametr = 15;

power = uLoudness;

}

/\*\*

\* Method gets area of dynamics of audio player in cm^2

\*/

public double getArea() {

return 3.14 \* (diametr / 2) \* (diametr / 2);

}

/\*\*

\* Method returns resource of button

\*

\*/

public double getPower() {

return power;

}

/\*\*

\* Method sets new power of burner of stove for cooking

\*

\* @param <code>newPower</code> The power value

\*/

public void setPower(double newPower) {

if (newPower < 0 || newPower > 100) {

System.out.println("Incorrectly specified power!");

} else {

power = newPower;

}

}

}

/\*\*

\* Class <code>Button</code> implements buttons of audio player

\*

\* @author Rudyi Volodymyr

\* @version 1.0

\*/

class Button {

// Button clicks resource

private int btnResource, count;

/\*\*

\* Constructor

\*/

public Button() {

btnResource = 999999999;

count = 0;

}

/\*\*

\* Constructor

\*

\* @param <code>res</code> Button clicks resource

\*/

public Button(int res) {

btnResource = res;

count = 0;

}

/\*\*

\* Method simulates Button clicking

\*/

public void clickButton(boolean type) {

if (type) {

btnResource--;

count++;

} else {

btnResource--;

count--;

}

}

/\*\*

\* Method returns resource of button

\*

\*/

public int getResource() {

return btnResource;

}

/\*\*

\* Method returns how many times the user clicked on the button at one time

\*

\* @return Button clicks

\*/

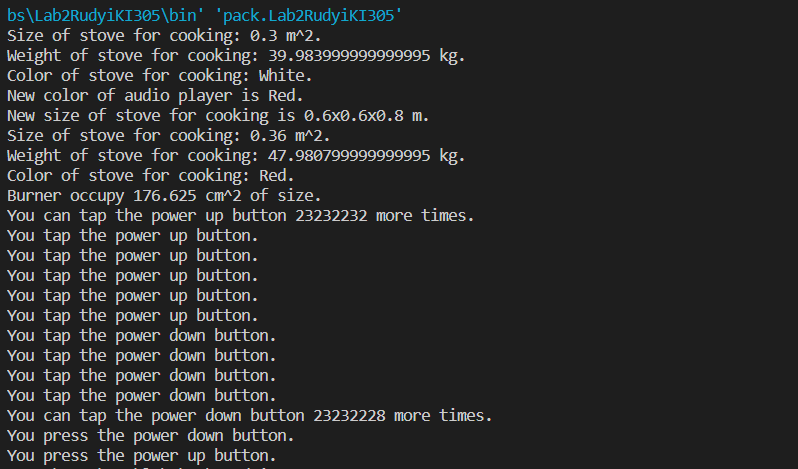
public int getCountClicks() {

return count;

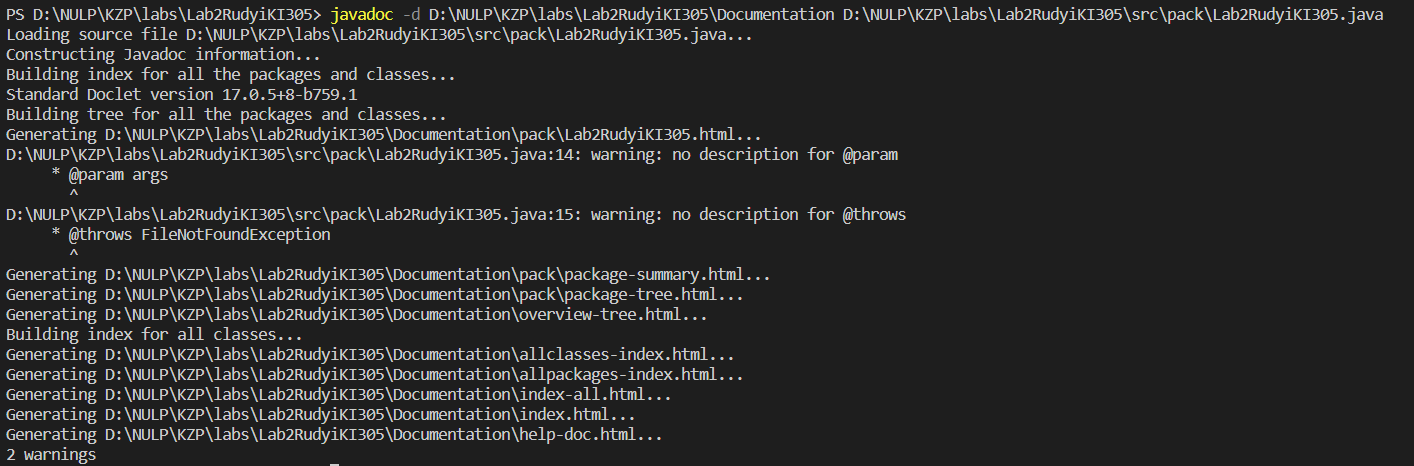
}

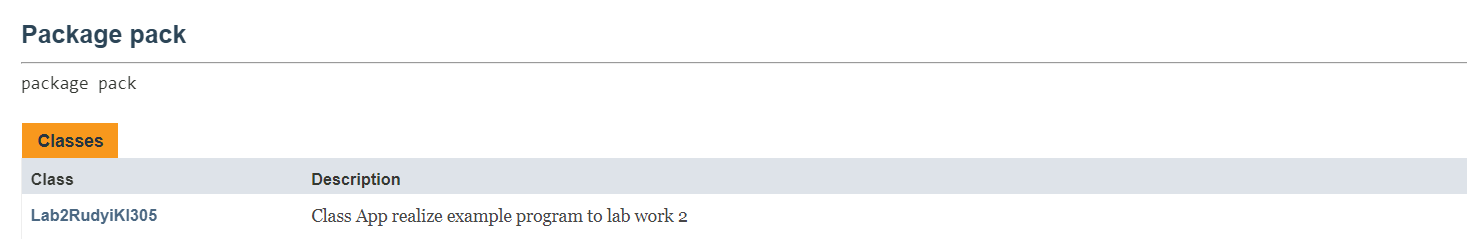
}

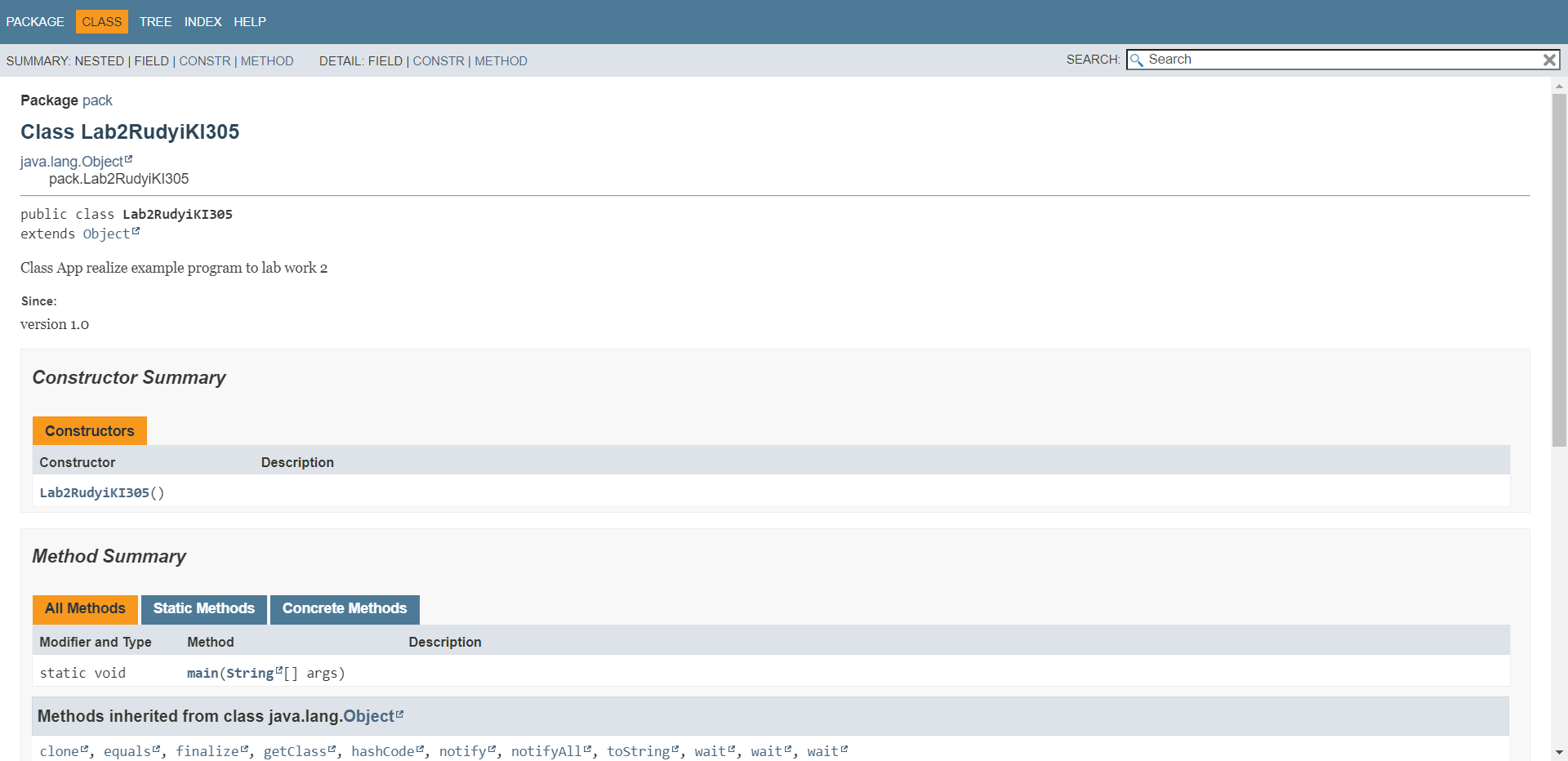
**Результат роботи програми у консолі та файлі:**



**Фрагмент генерації документації:**

****

****

****

**Висновок:** на цій лабораторній роботі я ознайомитися з процесом розробки класів та пакетів мовою Java.