Національний університет «Львівська Політехніка»

Інститут комп’ютерних технологій, автоматики та метрології

Кафедра електронних обчислювальних машин



Звіт

Про виконання лабораторної роботи №3

### З дисципліни «Кросплатформлені засоби програмування»

**Виконав:**

студент групи КІ-305

Рудий В.В.

**Перевірив:**

Доцент кафедри ЕОМ

Іванов Ю. С.

Львів – 2023

**Тема:** Спадкування та інтерфейси.

**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 №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.

**Завдання згідно варіанту:**

***Варіант: 23 Завдання: Газова плитка***

**Виконання:**

**Код програми:**

***Lab3RudyiKI305.java:***

package pack;

import java.io.FileNotFoundException;

/\*\*

\* Class App realize example program to lab work 3

\*

\* @author Rudyi Volodymyr

\* @version 1.1

\* @since version 1.0

\*/

public class Lab3RudyiKI305 {

/\*\*

\* @param args

\* @throws FileNotFoundException

\*/

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

GasStove stove1 = new GasStove(23232232, 50);

stove1.repaintStoveForCooking("Black");

stove1.getInfoLook();

stove1.getBurnerArea();

stove1.powerMax();

stove1.powerMin();

stove1.stoveHeating();

stove1.addGas(10);

stove1.stoveWorking(25);

stove1.getGas();

}

}

***StoveForCooking.java:***

package pack;

import java.io.\*;

/\*\*

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

\*/

abstract class StoveForCooking {

protected Type TypeOfStoveForCooking;

protected Burner BurnerOfStoveForCooking;

protected Button PowerUpButton;

protected Button PowerDownButton;

protected PrintWriter fout;

//static int power;

/\*\*

\* Method change color of stove for cooking

\*/

public void repaintStoveForCooking(String uCollor) {

TypeOfStoveForCooking.setColor(uCollor);

System.out.println("New color of a stove for cooking is " + uCollor + ".");

fout.println("New color of stove for cooking 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 size of stove for cooking is " + uLength + "x" + uWidth + "x" + uHeight + " .");

fout.flush();

}

abstract void getInfoLook();

/\*\*

\* 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 powerMin() {

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 stove for cooking 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 stove for cooking

\*

\* @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;

}

}

***GasStove.java:***

package pack;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.PrintWriter;

interface Heating {

void stoveHeating();

}

/\*\*

\* Class <code>GasStove</code> implements gas stove

\*/

public class GasStove extends StoveForCooking implements Heating {

/\*\*

\* Constructor

\*

\* @throws FileNotFoundException

\*/

private boolean ConnectingTheCylinder;

private int amountOfGas;

public GasStove() throws FileNotFoundException {

TypeOfStoveForCooking = new Type();

BurnerOfStoveForCooking = new Burner();

PowerUpButton = new Button();

PowerDownButton = new Button();

ConnectingTheCylinder = true;

amountOfGas = 30;

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

}

/\*\*

\* Constructor

\*

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

\* @throws FileNotFoundException

\*/

public GasStove(int resource, int uAmountOfGas) throws FileNotFoundException {

TypeOfStoveForCooking = new Type();

BurnerOfStoveForCooking = new Burner();

PowerUpButton = new Button(resource);

PowerDownButton = new Button(resource);

ConnectingTheCylinder = true;

amountOfGas = uAmountOfGas;

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

}

/\*\*

\* Method provides information about the appearance of the gas stove

\*/

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() + ".");

System.out.println("The amount of gas in the cylinder: " + amountOfGas + " L.");

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.println("The amount of gas in the cylinder: " + amountOfGas + " L.");

fout.flush();

}

/\*\*

\* Method of modeling the stove operation

\*/

public void stoveHeating() {

System.out.println("The stove is heating up.");

fout.println("The stove is heating up.");

fout.flush();

}

/\*\*

\* Method change the amount of gas

\*

\* @param <code>additionalGas</code> The value of the amount of gas

\*/

public void addGas(int additionalGas) {

amountOfGas += additionalGas;

}

/\*\*

\* Method how the stove works

\*/

public void stoveWorking(int time) {

if (ConnectingTheCylinder) {

System.out.println("The stove is working " + time +" min.");

fout.println("The stove is working " + time +" min.");

} else {

System.out.println("The stove is NOT working.");

fout.println("The stove is NOT working.");

}

amountOfGas -= (time/10);

fout.flush();

}

/\*\*

\* Method print amount of gas

\*/

public void getGas() {

System.out.println("Amount of gas in the cylinder: " + amountOfGas + " L.");

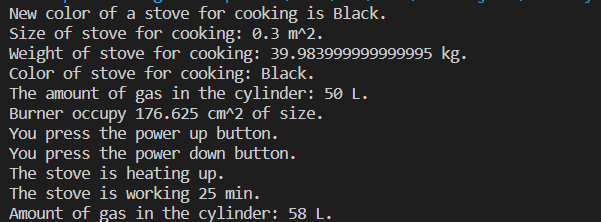
fout.println("Amount of gas in the cylinder: " + amountOfGas + " L.");

fout.flush();

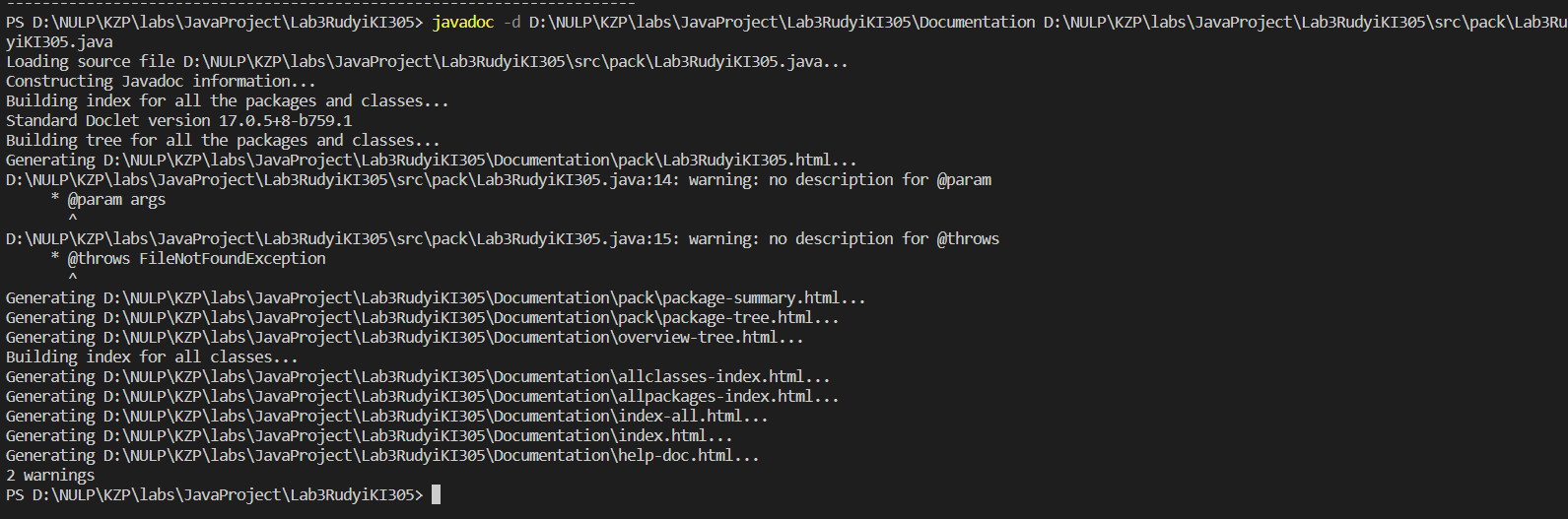
}

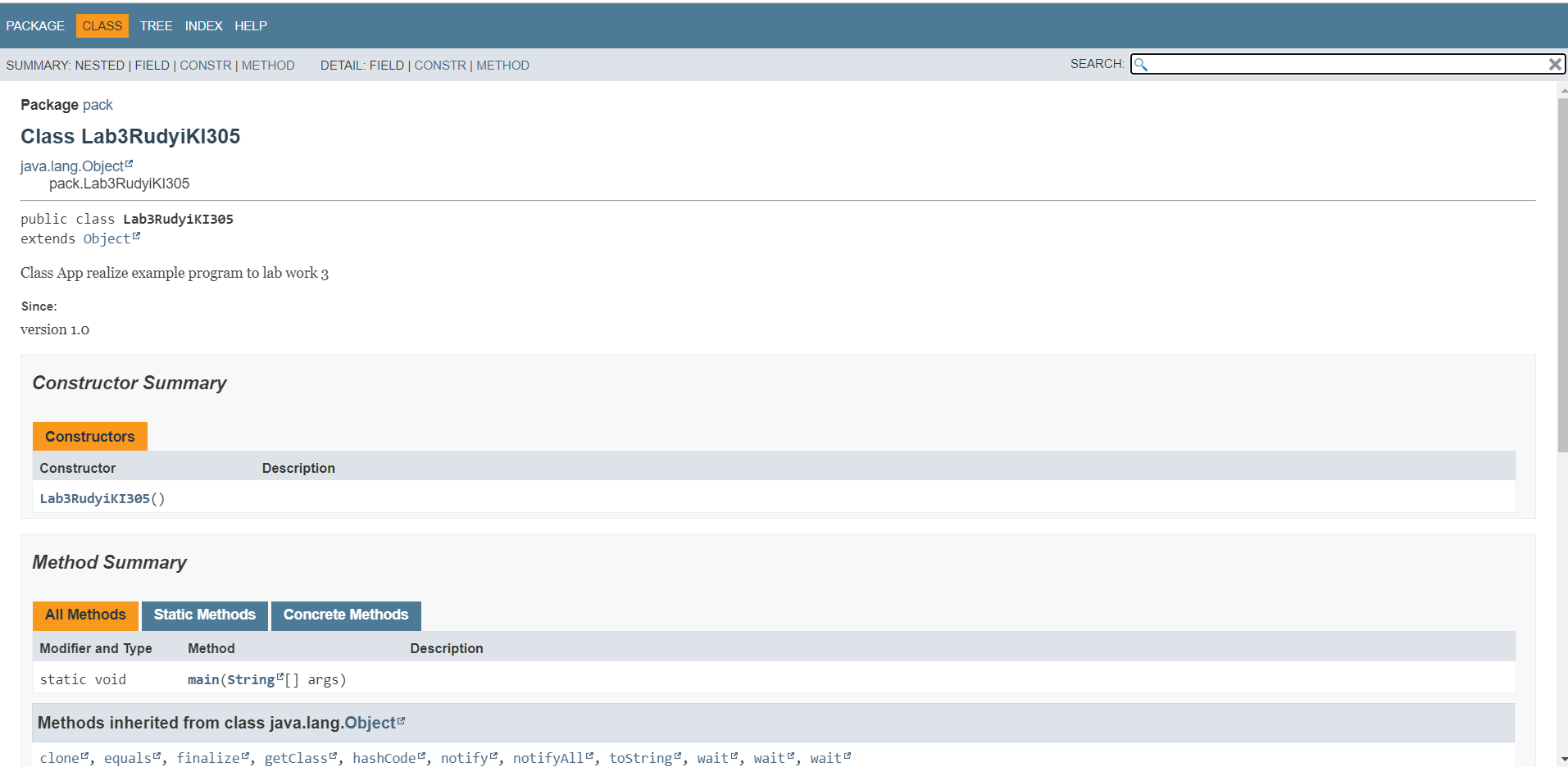
}

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



**Фрагмент генерації документації:**

****

**Висновок:** на цій лабораторній роботі я ознайомився з спадкуванням та інтерфейсами у мові Java. Перетворив попередній клас плита для приготування їжі у абстрактний і реалізував деякі його методи у новоствореному класі газова плиат. Також створив інтерфейс «Розігрів» і реалізував його метод в газовій плиті також.