Міністерство освіти і науки України

Національний університет «Львівська політехніка»



ЗВІТ

Про виконання лабораторної роботи №3

З дисципліни “Кросплатформні засоби програмування ”

Варіант-25

**Виконала:**

студ. групи КІ-35

Янчинська Н. Р.

**Прийняв:**

Іванов Ю. С.

Львів 2022

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:
2. програма має розміщуватися в пакеті Група.Прізвище.Lab3;
3. клас має містити мінімум 3 поля, що є об’єктами класів, які описують складові частини предметної області;
4. клас має містити кілька конструкторів та мінімум 10 методів;
5. для тестування і демонстрації роботи розробленого класу розробити клас-драйвер;
6. методи класу мають вести протокол своєї діяльності, що записується у файл;
7. розробити механізм коректного завершення роботи з файлом (не надіятися на метод finalize());
8. програма має володіти коментарями, які дозволять автоматично згенерувати документацію до розробленого пакету.
9. Автоматично згенерувати документацію до розробленого пакету.
10. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації.
11. Дати відповідь на контрольні запитання.

ВАРІАНТИ ЗАВДАННЬ

25. Кондиціонер

Код програми:

AirConditioner.java

package KI35.Yanchynska.Lab3;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.PrintWriter;

/\*\*

\* Class <code>AirConditioner</code> implements air conditioner

\*

\* @author Natalia Yanchynska

\* @version 1.0

\*/

public class AirConditioner {

private PrintWriter fout;

private SwitchButton switchButton;

private Temperature temperature;

private Fan fan;

private Mode mode;

// AirConditioner brand name value

private String brand;

/\*\*

\* Constructor

\*

\* @param

\* <code>brand</code> AirConditioner brand name value

\* @throws FileNotFoundException

\*/

public AirConditioner(String brandN)throws FileNotFoundException {

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

switchButton=new SwitchButton();

fan= new Fan();

temperature=new Temperature();

mode=Mode.DEFAULT;

brand=brandN;

}

/\*\*

\* Constructor

\*

\* @param

\* <code>temperatureValue</code> Temperature value

\* @throws FileNotFoundException

\*/

public AirConditioner(int initialTempValue)throws FileNotFoundException{

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

switchButton=new SwitchButton();

fan= new Fan();

temperature=new Temperature(initialTempValue);

mode=Mode.DEFAULT;

}

/\*\*

\* Method simulates turning on air conditioner

\*/

public void turnOn(){

switchButton.pressOnButton();

fan.startFan();

System.out.println("Air conditioner starts working!\n");

fout.printf("Initial temperature: "+temperature.getTemperatureV()+" C"+"\n");

fout.print("Fan start's spinning! Status: " + fan.isSpinned() + " with "+fan.getSpeedMode().name()+" mode"+"\n");

fout.print("Button "+switchButton.getButtonMode().name()+" is pressed"+"\n");

fout.flush();

}

/\*\*

\* Method simulates turning off air conditioner

\*/

public void turnOff(){

switchButton.pressOffButton();

fan.finishFan();

System.out.println("Air conditioner finished working!\n");

fout.print("Fan finished spinning! Status: " + fan.isSpinned() +"\n");

fout.print("Button "+switchButton.getButtonMode().name()+" is pressed"+"\n");

fout.flush();

}

/\*\*

\* Method simulates changing fan speed mode

\*/

public void changeFanSpeedMode(SpeedMode speedMode){

fan.changeFan(speedMode);

fout.printf("Fan speed mode changed to "+fan.getSpeedMode()+"\n");

}

/\*\*

\* Method simulates increasing temperature value

\*/

public void increaseTemperature(int increaseV){

temperature.increaseTemperature(increaseV);

fout.printf("Temperature value increased to "+temperature.getTemperatureV()+"\n");

}

/\*\*

\* Method simulates reduction temperature value

\*/

public void reductTemperature(int reductV){

temperature.reductTemperature(reductV);

fout.printf("Temperature value reducted to "+temperature.getTemperatureV()+"\n");

}

/\*\*

\* Method returns info about Temperature value

\*

\* @return Temperature value

\*/

public int getTemperatureV() {

return temperature.getTemperatureV();

}

/\*\*

\* Method returns SwitchButton mode value

\*

\* @return SwitchButton mode value

\*/

public String getButtonModeV() {

return switchButton.getButtonMode().name();

}

/\*\*

\* Method simulates choosing mode of air conditioner

\*/

public void chooseMode(Mode modeV){

mode=modeV;

fout.printf("Air conditioner stars working in "+mode.name()+" mode!"+"\n");

}

/\*\*

\* Method sets air conditioner brand name

\*/

public void setBrandName(String brandN){

brand=brandN;

}

/\*\*

\* Method returns AirConditioner brand name

\*

\* @return AirConditioner brand name

\*/

public String getBrandName() {

return brand;

}

/\*\*

\* Method releases used recourses

\*/

public void dispose() {

fout.close();

}

}

AirConditionerApp.java

package KI35.Yanchynska.Lab3;

import java.io.FileNotFoundException;

/\*\*

\* AirConditionerApp class implements main method for AirConditioner

\* class possibilities demonstration

\*

\* @author Natalia Yanchynska

\* @version 1.0

\*/

public class AirConditionerApp {

/\*\*

\* @param args

\* @throws FileNotFoundException

\*/

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

AirConditioner airConditioner = new AirConditioner("Phillips");

AirConditioner airConditioner1 = new AirConditioner(8);

airConditioner.turnOn();

airConditioner.chooseMode(Mode.DRY);

airConditioner.changeFanSpeedMode(SpeedMode.MAX);

airConditioner.increaseTemperature(12);

airConditioner.reductTemperature(5);

System.out.println("Current temperature: "+airConditioner.getTemperatureV()+" C");

System.out.println("Brand name of air conditioner : "+airConditioner.getBrandName());

airConditioner.turnOff();

airConditioner.dispose();

airConditioner1.turnOn();

airConditioner1.chooseMode(Mode.HEAT);

airConditioner1.changeFanSpeedMode(SpeedMode.LOW);

airConditioner.increaseTemperature(6);

airConditioner.reductTemperature(10);

System.out.println("Current temperature: "+airConditioner1.getTemperatureV()+" C");

airConditioner1.setBrandName("Samsung");

System.out.println("Brand name of air conditioner : "+airConditioner1.getBrandName());

airConditioner.turnOff();

airConditioner1.dispose();

}

}

ButtonMode.java

package KI35.Yanchynska.Lab3;

public enum ButtonMode {

ON,OFF

}

Fan.java

package KI35.Yanchynska.Lab3;

/\*\*

\* Class <code>Fan</code> implements fan

\*

\* @author Natalia Yanchynska

\* @version 1.0

\*/

public class Fan {

// Fan mode value

private SpeedMode speedMode;

// Fan mode

private boolean isSpinned;

/\*\*

\* Constructor

\*/

public Fan() {

isSpinned = false;

speedMode = SpeedMode.INITIAL;

}

/\*\*

\* Method simulates Fan start's working

\*/

public void startFan(){

isSpinned=true;

}

/\*\*

\* Method simulates Fan finishes working

\*/

public void finishFan(){

isSpinned=false;

}

/\*\*

\* Method returns Fan speed mode value

\* @return Fan speed mode value

\* <code>Fan.SpeedMode</code>

\*/

public SpeedMode getSpeedMode(){

return speedMode;

}

/\*\*

\* Method simulates changing Fan speed mode value

\*/

public void changeFan(SpeedMode speedModeValue){

speedMode=speedModeValue;

}

/\*\*

\* Method returns Fan mode value

\* @return The Fan mode

\*/

public boolean isSpinned() {

return isSpinned;

}

}

Mode.java

package KI35.Yanchynska.Lab3;

public enum Mode {

DEFAULT,HEAT,FAN,DRY,AUTO

}

SpeedMode.java

package KI35.Yanchynska.Lab3;

public enum SpeedMode {

INITIAL,LOW,MEDIUM,MAX

}

SwitchButton.java

package KI35.Yanchynska.Lab3;

/\*\*

\* Class <code>SwitchButton</code> implements switch button

\*

\* @author Natalia Yanchynska

\* @version 1.0

\*/

public class SwitchButton {

// Switch button mode

private boolean isPressed;

// Switch button mode value

private ButtonMode buttonMode;

/\*\*

\* Constructor

\*/

public SwitchButton(){

isPressed=false;

buttonMode=ButtonMode.OFF;

}

/\*\*

\* Method simulates pressing button to on mode

\*/

public void pressOnButton(){

isPressed=true;

buttonMode=ButtonMode.ON;

}

/\*\*

\* Method simulates pressing button to off mode

\*/

public void pressOffButton(){

isPressed=true;

buttonMode=ButtonMode.OFF;

}

/\*\*

\* Method returns the SwitchButton mode

\* @return The SwitchButton mode

\*/

public ButtonMode getButtonMode() {

return buttonMode;

}

}

Temperature.java

package KI35.Yanchynska.Lab3;

/\*\*

\* Class <code>Temperature</code> implements temperature

\*

\* @author Natalia Yanchynska

\* @version 1.0

\*/

public class Temperature {

// Temperature value

private int temperatureValue;

/\*\*

\* Constructor

\*/

public Temperature(){

temperatureValue=(int)(Math.random() \* 25 + 5);

}

/\*\*

\* Constructor

\* @param

\* <code>tValue</code> Temperature temperature value

\*/

public Temperature(int tValue){

temperatureValue=tValue;

}

/\*\*

\* Method simulates increasing Temperature value

\*/

public void increaseTemperature(int increasedValue){

temperatureValue+=increasedValue;

}

/\*\*

\* Method simulates reduction Temperature value

\*/

public void reductTemperature(int reductedValue){

temperatureValue-=reductedValue;

}

/\*\*

\* Method returns the Temperature value

\* @return The Temperature value

\*/

public int getTemperatureV() {

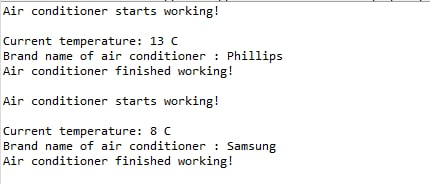
return temperatureValue;

}

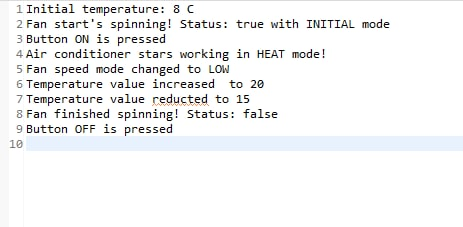
}

Результат виконання програми:

Консоль:



Текстовий файл:



КОНТРОЛЬНІ ПИТАННЯ

Синтаксис визначення класу.

[public] class НазваКласу

{

[конструктори]

[методи]

[поля]

}

Синтаксис визначення методу.

[СпецифікаторДоступу] [static] [final] Тип назваМетоду([параметри]) [throws класи]

{

[Тіло методу]

[return [значення]];

}

Висновок:

На цій лабораторній роботі я ознайомилася з процесом розробки класів та пакетів мовою Java.