Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра «Електронних обчислювальних машин»



Звіт

з лабораторної роботи № 3

з дисципліни: «Кросплатформенні засоби програмування»

на тему: «КЛАСИ ТА ПАКЕТИ »

**Виконав:**

студент групи КІ-35

Ничай В.Б.

**Прийняв:**

доцент кафедри ЕОМ

Іванов Ю. С.

   
Львів – 2022

 Мета: 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.



Текст програми

Lab3.java

package Package\_Ship;  
  
import java.io.FileNotFoundException;  
  
public class Lab3 {  
 public static void main(String []args) throws FileNotFoundException {  
 Ship ship1 = new Ship(100,100,100,100,2);  
 Ship ship2 = new Ship(100,100,100,100,4);  
 Ship ship3 = new Ship(100,100,100,100,8);  
 Ship ship4 = new Ship(100,100,100,100,11);  
 Ship ship5 = new Ship(100,100,100,100,14);  
   
 }  
  
  
}

Ship.java

package Package\_Ship;  
import java.io.\*;  
import java.util.PrimitiveIterator;  
  
*/\*\*  
 \* Constructor  
 \*/*public class Ship {  
private Engine engine1;  
private Location loc1;  
private Crew people;  
private PrintWriter fout;  
  
 Ship()throws FileNotFoundException{  
 engine1 = new Engine();  
 loc1 = new Location();  
 people = new Crew();  
  
 fout = new PrintWriter(new File("Result.txt"));  
 }  
  
 Ship(int valuePower,int valueVolume,int xLoc, int yLoc,int valueCount)throws FileNotFoundException{  
 engine1 = new Engine(valuePower,valueVolume);  
 loc1 = new Location(xLoc,yLoc);  
 people = new Crew(valueCount);  
  
 fout = new PrintWriter(new File("Result.txt"));  
 }  
 public void setShipCount(int count){  
 people.setCount(count);  
 System.*out*.println("Now there are "+ people.getCount() + "people on the ship");  
 fout.print("You change count of crew\n");  
 fout.flush();  
 }  
 public void shipStartMoving(){  
 if (people.getCount()>50){  
 System.*out*.println("Ship is going");  
 fout.print("Ship is going\n");  
 fout.flush();  
 }  
 else{  
 System.*out*.println("You don't start moving");  
 fout.print("You don't start moving\n");  
 fout.flush();  
 }  
 }  
 public int getCountOfPeople(){  
 return people.getCount();  
}  
 public void increaseCountOfPeople(int value){  
 people.increaseCount(value);  
 System.*out*.println("Now there are "+ people.getCount() + "people on the ship");  
 fout.print("You change count of crew\n");  
 fout.flush();  
 }  
 public void reduseCountOfPeople(int value){  
 people.reduseCount(value);  
 System.*out*.println("Now there are "+ people.getCount() + "people on the ship");  
 fout.print("You change count of crew\n");  
 fout.flush();  
 }  
 /////  
  
 public int getXlocationOfShip(){  
 return loc1.getXlocation();  
 }  
 public int getYlocationOfShip(){  
 return loc1.getYlocation();  
 }  
 public void setLocationOfShip(int valueX, int valueY){  
 loc1.setXLocation(valueX);  
 loc1.setYLocation(valueY);  
 }  
 public void increaseAllLocation(int valueX, int valueY){  
 loc1.increaseXLocation(valueX);  
 loc1.increaseYLocation(valueY);  
 }  
 public void reduseAllLocation(int valueX, int valueY){  
 loc1.reduseXLocation(valueX);  
 loc1.reduseYLocation(valueY);  
 }  
 public void getInfoAboutLocationOfShip(){  
 loc1.showInfoAboutLocation();  
 }  
 //  
  
 public void redusePowerOfShip(int value){  
 if (value>=engine1.getPower())  
 {  
 System.*out*.println("You enter wrong value");  
 }  
 else{  
 engine1.redusePower(value);  
 }  
 }  
 public void increasePowerOfShip(int value){  
 if (value<0)  
 {  
 System.*out*.println("You enter wrong value");  
 }  
 else{  
 engine1.increasePower(value);  
 }  
 }  
 public void showEngineOfShip(){  
 engine1.showEngine();  
 fout.print("You look info\n");  
 fout.flush();  
 }  
  
 public void AllInfo(){  
 System.*out*.println("Power of ship is " + engine1.getPower());  
 System.*out*.println("Volume of ship is "+ engine1.getVolume());  
 System.*out*.println("Count of people = " + people.getCount());  
 System.*out*.println(" X location is " + loc1.getXlocation());  
 System.*out*.println(" Y location is " + loc1.getYlocation());  
 }  
  
}

Engine.java

package Package\_Ship;  
*/\*\*  
 \* Class <code>Engine</code> implements the operation of the engine  
 \** ***@author*** *Nychai Volodymyr KI-35  
 \** ***@version*** *1.0  
 \*/*public class Engine {  
 //characteristics of engine  
 private int power;  
 private int volume;  
  
 */\*\*  
 \* Constructor  
 \*/* public Engine(){  
 this.power=100000;  
 this.volume=500;  
 }  
 */\*\*  
 \* Another Constructor  
 \** ***@param*** *power -- power of ship  
 \*/* public Engine(int power, int volume){  
 this.power=power;  
 this.volume=volume;  
 }  
  
 public int getPower(){  
 return this.power;  
 }  
 public int getVolume(){  
 return this.volume;  
 }  
 */\*\*  
 \* method reduse power  
 \** ***@param*** *value  
 \*/* public void redusePower(int value){  
 if (value>=this.power)  
 {  
 System.*out*.println("You enter wrong value");  
 }  
 this.power-=value;  
 showEngine();  
 }  
 */\*\*  
 \* method increase power  
 \*/* public void increasePower(int value){  
 if (value<=0)  
 {  
 System.*out*.println("You enter wrong value");  
 }  
 this.power+=value;  
 showEngine();  
 }  
  
 public void showEngine(){  
 System.*out*.println("Power is " + this.power);  
 System.*out*.println("Volume is " + this.volume);  
 System.*out*.print("\n");  
 } //SHOW INFO  
}

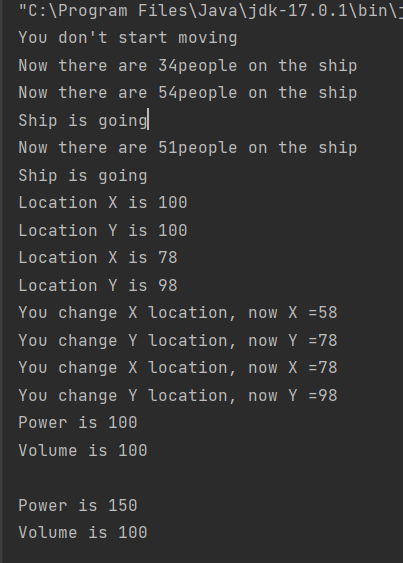
Location.java

package Package\_Ship;  
  
import java.util.IllegalFormatCodePointException;  
*/\*\*  
 \* Class <code>Location</code> implements the operation of the engine  
 \** ***@author*** *Nychai Volodymyr KI-35  
 \** ***@version*** *1.0  
 \*/*public class Location {  
 private int x,y;  
  
 */\*\*  
 \* Constructor  
 \*/* Location(){  
 this.x=0;  
 this.y=0;  
 }  
 */\*\*  
 \* Another Constructor  
 \*/* Location(int xValue, int yValue){  
 this.x=xValue;  
 this.y=yValue;  
 }  
  
 */\*\*  
 \* getXlocation - method return value x  
 \** ***@return*** *\*/* public int getXlocation(){  
 return this.x;  
 }  
 */\*\*  
 \* getYlocation - method return value y  
 \** ***@return*** *\*/* public int getYlocation(){  
 return this.y;  
 }  
 public void setXLocation(int valueX){  
 this.x=valueX;  
 }  
 public void setYLocation(int valueY){  
 this.y=valueY;  
 }  
 */\*\*  
 \* Print info about location  
 \*/* public void showInfoAboutLocation(){  
 System.*out*.println("Location X is " + getXlocation());  
 System.*out*.println("Location Y is " + getYlocation());  
 }  
 */\*\*  
 \* method increase X  
 \*/* public void increaseXLocation(int valueX){  
 if (valueX<0){  
 System.*out*.println("You entered wrong value");  
 }  
 else{  
 this.x+=valueX;  
 System.*out*.println("You change X location, now X =" + getXlocation());  
 }  
  
 }  
 */\*\*  
 \* method increase Y  
 \*/* public void increaseYLocation(int valueY){  
 if (valueY<0){  
 System.*out*.println("You entered wrong value");  
 }  
 else{  
 this.y+=valueY;  
 System.*out*.println("You change Y location, now Y =" + getYlocation());  
 }  
  
 }  
 */\*\*  
 \* method reduse Y  
 \*/* public void reduseYLocation(int valueY){  
 if (valueY>this.y){  
 System.*out*.println("You entered wrong value");  
 }  
 else{  
 this.y-=valueY;  
 System.*out*.println("You change Y location, now Y =" + getYlocation());  
 }  
 }  
 */\*\*  
 \* method reduse X  
 \*/* public void reduseXLocation(int valueX){  
 if (valueX>this.x){  
 System.*out*.println("You entered wrong value");  
 }  
 else{  
 this.x-=valueX;  
 System.*out*.println("You change X location, now X =" + getXlocation());  
 }  
 }  
}

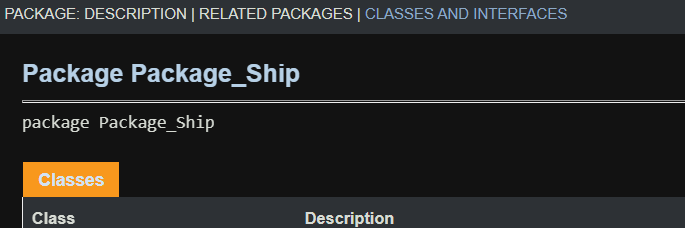
Crew.java

package Package\_Ship;  
//екіпаж  
public class Crew {  
 private int count;  
  
 */\*\*  
 \* Constructor  
 \*/* Crew(){  
 this.count=0;  
 }  
 Crew(int count){  
 this.count=count;  
 }  
 */\*\*  
 \* method print info about move  
 \*/* public void startMoving(){  
 if (count>50){  
 System.*out*.println("Ship is going");  
 }  
 else{  
 System.*out*.println("You don't start moving");  
 }  
 }  
 public int getCount(){  
 return count;  
 }  
 public void setCount(int count){  
 this.count+=count;  
 }  
  
 public void increaseCount(int value){  
 if (value<0){  
 System.*out*.println("You entered wrong number");  
 }  
 else {  
 this.count+=value;  
 }  
  
 }  
 public void reduseCount(int value){  
 if (value>this.count){  
 System.*out*.println("There aren't that many people on the ship");  
 }  
 else {  
 this.count-=value;  
 }  
  
 }  
  
}

Результат роботи програми



**Фрагмент згенерованої документації**



Контрольні питання

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]];

}

Висновок: на даній лабораторній роботі я ознайомився з поняттям пакетів. Створив робочу програму, що пов’язує у собі декілька класів. Навчився створювати поля одного класу за допомогою об’єктів іншого класу. Компіляція пройшла успішно, дані також були записані у файл.