Министерство науки и высшего образования Российской Федерации

Рязанский государственный радиотехнический университет им. В.Ф. Уткина

Кафедра вычислительной и прикладной математики

Лабораторная работа № 6

«Организация ввода-вывода»

Выполнили:

студенты гр. 744

Акшенцева Д.А.

Гранкин А.Н.

Проверил

Доц. Коротаев А.Н.

Рязань 2020

**Задание**

На основе предыдущих лабораторных работ для своей предметной области реализуйте ввод исходных данных для своей программы (имя экземпляра класса, значения его полей и т.д.) из файла, например, in.txt. Результаты работы программы, т.е. получившиеся экземпляры классов с их параметрами необходимо сохранить в другой файл, например, out.txt.

Варианты предметной области.

1. Магазин электротоваров.

**Main.java**

package rsreu;  
import java.io.\*;  
import java.util.Scanner;  
import java.util.Arrays;  
import java.util.List;  
import java.util.logging.Level;  
import java.util.logging.Logger;  
  
  
public class Main {  
  
 public static void main(String[] args) {  
 final String DIRECTORY = "D:\\РГРТУ\\3 курс\\6 семестр\\Java\\";  
 final String FILENAME = "D:\\РГРТУ\\3 курс\\6 семестр\\Java\\in.txt";  
 // write your code here  
 //запись в файл  
 try {  
 FileOutputStream fos = new FileOutputStream(DIRECTORY+"out.txt");  
 ObjectOutputStream oos = new ObjectOutputStream(fos);  
 ProductsForConstruction productsForConstruction2 = new ProductsForConstruction("chainsaw", 45, "Product for construction",567, "battery","yes");  
 PK computer2 = new PK(56, "ASUS", 456, "30l", "no", 67, true,true, "touch");  
 KitchenAppliances blender2 = new KitchenAppliances(320, "BOSCH", 440,"5L","no",57, true, true, true );  
  
 oos.writeObject( productsForConstruction2);  
 oos.writeObject( computer2);  
 oos.writeObject(blender2);  
 oos.close();  
 fos.close();  
 } catch (FileNotFoundException ex) {  
 Logger.*getLogger*(Main.class.getName()).log(Level.*SEVERE*, null, ex);  
 }catch (IOException ex){  
 Logger.*getLogger*(Main.class.getName()).log(Level.*SEVERE*, null, ex);  
 }  
 //чтение из файла  
 try{  
 FileInputStream fis = new FileInputStream(FILENAME);  
 ObjectInputStream ois = new ObjectInputStream(fis);  
 ProductsForConstruction productsForConstruction21 =(ProductsForConstruction) ois.readObject();  
 System.*out*.println(productsForConstruction21.toString());  
 PK computer21 =(PK) ois.readObject();  
 System.*out*.println(computer21.toString());  
 KitchenAppliances blender21 = (KitchenAppliances) ois.readObject();  
 System.*out*.println(blender21.toString());  
 fis.close();  
 ois.close();  
 }catch (FileNotFoundException ex) {  
 Logger.*getLogger*(Main.class.getName()).log(Level.*SEVERE*, null, ex);  
 }catch (IOException ex){  
 Logger.*getLogger*(Main.class.getName()).log(Level.*SEVERE*, null, ex);  
 } catch (ClassNotFoundException e) {  
 e.printStackTrace();  
 }  
  
 catch(Exception ex){  
 System.*out*.println(ex.getMessage());  
 }  
 }  
}

**PK.java**

package rsreu;  
  
import java.io.Serializable;  
  
public class PK extends HomeTech implements Serializable {  
 private boolean processor;  
 private boolean typeOfPower;  
 private String typeOfControl;  
 public PK (int power,String name,int count,String volume,String integrated,int energyConsumption,  
 boolean processor,boolean defrost,String typeOfControl){  
 super(power,name,count,volume,integrated,energyConsumption);  
 this.processor = processor;  
 this.typeOfPower = typeOfPower ;  
 this.typeOfControl =typeOfControl ;  
 }  
  
 public boolean getTheGrillMode(){  
 return processor;  
 }  
  
 public void setTheGrillMode(boolean processor) {  
 this.processor = processor;  
 }  
  
 public String getTypeOfControl() {  
 return typeOfControl;  
 }  
  
 public void setTypeOfControl(String typeOfControl) {  
 this.typeOfControl = typeOfControl;  
 }  
  
 public boolean getDefrost() {  
 return typeOfPower;  
 }  
  
 public void setDefrost(boolean typeOfPowe) {  
 this.typeOfPower = typeOfPower;  
 }  
  
  
 public String toString() {  
 return new StringBuilder().append("Мощность: ").append(power).append("; Название: ").append(name).append("; Колличество: ")  
 .append(count).append(";").append("Процессор: ").append(processor).append(" Тип энергии: ").append(typeOfPower).append(" Тип управления ").append(typeOfControl).toString();  
 }  
}

**ElectricAppliances.java**

package rsreu;  
  
import java.io.Serializable;  
  
public class ElectricAppliances implements Purchase, Serializable {  
 private static final long *serialVersionUID* = 1L;  
 public int power;  
 public String name;  
 public int count;  
  
  
 ElectricAppliances(int power, String name, int count){  
  
 this.power = power;  
 this.name = name;  
 this.count = count;  
 }  
  
 private ElectricAppliances(){  
  
 }  
 public String getName() {  
 return name;  
 }  
 public void setName(String name) {  
  
 this.name = name;  
 }  
 public int getCount() {  
 return count;  
 }  
 public void setCount(int count) {  
 this.count = count;  
 }  
  
 public int getPower() {  
 return power;  
 }  
  
 public void setPower(int power) throws Exception {  
 if (power >=0){  
 this.power = power;  
 }else {  
 throw new Exception("Изменение этого значения невозможно, так как это значение не может быть отрицательным!");  
 }  
  
 }  
  
 @Override  
 public void sail() {  
 System.*out*.println("Хотите купить этот товар? ");  
 }  
  
 @Override  
 public void discount() {  
  
 }  
  
 public void description() {  
 }  
  
 public String toString() {  
 return new StringBuilder().append("Мощность: ").append(power).append("; Название: ").append(name).append("; Колличество: ")  
 .append(count).append(";").toString();  
 }  
  
  
}

**Ответы на вопросы:**

1. Что такое потоки в Java?

Поток - это абстракция, которая либо производит, либо использует информацию. Поток связывается с физическим устройством через систему ввода-вывода Java. Java работает с потоками с помощью иерархии классов, находящейся в пакете java.io.

1. Назовите 2 абстрактных суперкласса у символьных потоков?

Байтовые потоки характеризуются двумя иерархиями классов. На их вершинах находятся 2 абстрактных класса: InputStream и OutputStream. Каждый из этих абстрактных классов содержит несколько подклассов, которые работают с различными устройствами, такими как дисковые файлы, сетевые соединения и буфер обмена.

1. Что такое статический импорт?

Статический импорт является функция , представленная в языке программирования Java , что позволяет участникам (поля и методы) , которые были контекстными в пределах их контейнерного класса , как public static, для использования в Java коде без указания класса , в котором было определено поле.