Министерство науки и высшего образования Российской Федерации

Рязанский государственный радиотехнический университет им. В.Ф. Уткина

Кафедра вычислительной и прикладной математики

Лабораторная работа № 6

«Обобщения»

Выполнили:

студенты гр. 744

Акшенцева Д.А.

Проверил

Доц. Коротаев А.Н.

Рязань 2020

**Задание**

На основе предыдущих лабораторных работ для своей предметной области на базе обычного массива (готовые массивы типа ArrayList не использовать) создать обобщённый класс DynamicArray для работы с однотипными полями экземпляров класса, для которых в лабораторной работе №6 создавался ввод и вывод в файл. Необходимо у всех экземпляров класса считать в DynamicArray значения одного поля (например, ID), вывести содержимое на экран, провести поиск дубликатов при наличии и вывести их количество. Те же самые операции необходимо произвести поочерёдно со всеми остальными полями класса. Добавлять в массив данные можно из файлов

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 notebook1 = new PK(56, "HP", 456, "30l", "no", 67, true,true, "touch");

PK notebook2 = new PK(56, "HP", 456, "no", "no", 67, true,true, "touch");

DynamicArray dynamicArray = new DynamicArray(notebook1.getName());

dynamicArray.Add(notebook2.getName());

for(int i=0; i<dynamicArray.GetCount(); i++){

System.out.println((i+1)+"-ый элемент массива: "+dynamicArray.get(i));

}

System.out.println("Количество совпадений: " + dynamicArray.FindMatch());  
 }  
}

**DynamicArray.java**

package rsreu;  
  
public class DynamicArray <T> {  
 private T element;  
 private int count = 1;  
 private int match = 0;  
 private Object[] array;  
  
 public int GetCount(){return count;}  
  
 DynamicArray(T ob){  
 this.ob = ob;  
 array = new Object[count];  
 array[0] = ob;  
 }  
  
 T get(int i) {  
 @SuppressWarnings("unchecked")  
 final T t = (T) array[i];  
 return t;  
 }  
 public void Add(T ob){  
 count++;  
 array=Insert(count, ob);  
   
 }  
 public int FindMatch(){  
 if(count==1){  
 match=0;  
 }  
 else{  
 for(int i=0;i<count-1;i++){  
 if(array[i]==array[i+1]){  
 match++;  
 }  
 }  
 }  
 return match;  
 }  
 private Object[] Insert(int count, T ob){  
 Object[] newArray = new Object[count];  
 for(int i=0;i<count-1;i++){  
   
 newArray[i]=array[i];  
 }  
 newArray[count-1]=ob;  
 return newArray;  
 }  
}

**Ответы на вопросы:**

1. Что такое обобщение?

По сути термин обобщения подразумевает параметризированные типы.

1. Что обозначает подстановочный тип <?> ?

Подстановочный аргумент обозначается как ?, и представляет из себя неизвестный тип.

1. Какие ограничения существуют у обобщений?

Подстановочные аргументы могут ограничиваться также, как и типы параметров.  
Нельзя создать массив обобщенных коллекций. Любой массив коллекций шаблонов разрешен, но является опасным с позиции безопасности типов. Нельзя создать обобщение элементарного типа.