Sri Krishna College of Technology

Name :VIKRAM P Email :22cse.vikram.p@skct.edu.in
Roll no:727822TUCS253 Phone :9786253595
Branch :Sri Krishna College of Technology Department :CSE
Batch :2022-26 Degree :BE-CSE

2022_26_III_Advanced Java_CS_IRC

2022_26_java_Collections_DAY11_PAH1

Attempt : 1
Total Mark : 80
Marks Obtained : 80

Section 1 : SECTION A

1. Create a class ItemType with the following attributes.

Mark all of them as private and included appropriate getters/setters, also include constructors. And override the toString method to print details of the ItemType object.

Create a class ArrayListObjectMain and in the main method get the details of ItemTypes and store them in an ArrayList.

All class names, attribute names, and method names should be the same as specified in the problem statement.

Note: Use "%-20s%-20s%-20s" for formatting output in tabular format, display double values with 1 decimal place.

Answer

```
import java.util.*;

class ItemType
{
   String name;
   double dep;
   double cost;

   ItemType(String name,double dep,double cost)
   {
      this.name=name;
      this.dep=dep;
      this.cost=cost;
   }


}

class ArrayListMain
{
   public static void main(String[] args)
   {
      Scanner sc= new Scanner(System.in);
      {
         int n=sc.nextInt();
         sc.nextLine();
         ArrayList<ItemType> al=new ArrayList();

         for(int i=0;i<n;i++)
         {
            String name=sc.next();
            double dep=sc.nextDouble();
            double cost=sc.nextDouble();

            ItemType obj=new ItemType(name,dep,cost);
            al.add(obj);
         }
         System.out.printf("%-20s%-20s%-20s","Name","Deposit","Cost Per Day");
         Iterator<ItemType> it=al.iterator();
         System.out.println();
         while(it.hasNext())
         {
            ItemType current=it.next();
            System.out.printf("%-20s%-20s
%-20s",current.name,current.dep,current.cost);
            System.out.println();
         }
      }
   }
}
```

Status : Correct ***Marks : 10/10***

2. Create a class named Book with the following private attributes.

Include getters and setters.

Include default and parameterized constructor.

The format for the parameterized constructor is

Public Book(int id, String name, String author, String publisher, int quantity)

Create a Main class and in the main method get the details of the books and store them in the linked hash set.

Display all the book details and search for a book name in the set.

Answer

```
import java.util.*;

class Book
{
   int id;
   String name;
   String author;
   String pub;
   int quantity;

   Book(int id,String name,String author,String pub,int quantity)
   {
      this.id=id;
      this.name=name;
      this.author=author;
      this.pub=pub;
      this.quantity=quantity;
   }
}
```

```
class Main
{
    public static void main(String[] args)
    {
        Scanner sc= new Scanner(System.in);

        LinkedHashSet<Book> lk=new LinkedHashSet();

        int n=sc.nextInt();
        int f=0;
        for(int i=0;i<n;i++)
        {
            int id=sc.nextInt();
            sc.nextLine();
            String name=sc.nextLine();
            String author=sc.nextLine();
            String pub=sc.nextLine();
            int quantity=sc.nextInt();
            Book obj=new Book(id,name,author,pub,quantity);
            lk.add(obj);
        }
        Iterator<Book>it= lk.iterator();

        while(it.hasNext())
        {
            Book current=it.next();
            System.out.println(current.id+" "+current.name+" "+current.author+"
"+current.pub+" "+current.quantity);
        }
        sc.nextLine();

        String key=sc.nextLine();
        Iterator<Book> newit=lk.iterator();
        while(newit.hasNext())
        {
            Book newcurrent=newit.next();
            if(newcurrent.name.equals(key))
            {
                System.out.println(key+" is present in the set");
                f=1;
            }
        }
        if(f==0)
        {
            System.out.println(key+" is not present in the set");
        }
    }
}
```

Status : Correct ***Marks : 10/10***

3. Use remove() and isEmpty() methods of the ArrayList API and implement them in our application. let's experiment with Hall class for performing these methods.

Create a class named Hall having the following private attributes.

Include getters and Setters for the attributes

Include default and parameterized constructors.

Format for a parameterized constructor is

Hall(String name, String contactNumber, Double costPerDay,String ownerName)

The Hall class contains the following method.

Create a driver class called Main. In the Main method, obtain input from the console If the list is empty and a remove action is performed display "The list is empty" and terminate. Display the Hall details by iterating the Hall List and calling the display() method after removing action.

Hint: Use isEmpty() and remove() methods of ArrayList api.

Use System.out.printf("%-20s%-20s%-20s%-20s") for displaying the Hall details in tabular form.

Note: Strictly adhere to the Object-Oriented specifications given in the problem statement.

All class names, attribute names, and method names should be the same as specified in the problem statement.

Answer

```
import java.util.*;

class Hall
{
   String name;
   String num;
   double cost;
   String ownname;

   Hall(String name,String num,double cost,String ownname)
   {
      this.name=name;
      this.num=num;
      this.cost=cost;
      this.ownname=ownname;
   }

   static void display(int ind,ArrayList al)
   {
      System.out.printf("%-20s%-20s%-20s%-20s","Name","Contact
Number","CostperDay","Owner Name");
      System.out.println();

      al.remove(ind);
      Iterator<Hall>it=al.iterator();
      while(it.hasNext())
      {
         Hall c=it.next();
         System.out.printf("%-20s%-20s%-20s
%-20s",c.name,c.num,c.cost,c.ownname);
         System.out.println();
      }
   }
}

class Main
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      int n=sc.nextInt();
```

```
        sc.nextLine();
        ArrayList<Hall> al=new ArrayList();
        for(int i=0;i<n;i++)
        {
           String name=sc.nextLine();
           String num=sc.nextLine();
           double cost=sc.nextDouble();
           sc.nextLine();
           String ownname=sc.nextLine();
           Hall obj=new Hall(name,num,cost,ownname);
           al.add(obj);
        }
        int ind=sc.nextInt();
        if(al.isEmpty())
        {
           System.out.println("The list is empty");
        }
        else
        {
           Hall.display(ind,al);
        }
    }
}
```

Status : Correct ***Marks : 10/10***

4. Equals method - Collections

Let's dive deep into Set and explore its inbuilt functions. We usually perform equals operations to compare objects. Now try the same feature here. Experiment with the equals() method in this problem.

Create Main class. Obtain two sets of numbers and check whether they are the same or not.

Note:

Strictly adhere to the Object-Oriented specifications given in the problem statement. All class names, attribute names, and method names should be the same as specified in the problem statement.

Answer

```
// You are using Java
import java.util.*;

class Main
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);

      int n=sc.nextInt();
      Set<Integer> st1=new HashSet<Integer>();
      for(int i=0;i<n;i++)
      {
         st1.add(sc.nextInt());
      }

      Set<Integer> st2=new HashSet<Integer>();
      for(int i=0;i<n;i++)
      {
         st2.add(sc.nextInt());
      }
      if(st1.equals(st2))
         System.out.println("Both sets are same");
      else
         System.out.println("Both sets are different");

   }
}
```

Status : Correct ***Marks : 10/10***

5. Input a positive integer N (N > 0), input N strings, and sort the strings in place in the order of increasing length. Print the sorted strings using ArrayList as an implementation of the List interface for storing the individual strings.

Answer

```
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Collections;
```

```
class Main
{
   public static void main(String[] args)
   {
      Scanner sc= new Scanner(System.in);

      int n=sc.nextInt();
      sc.nextLine();
      ArrayList<String> al=new ArrayList<String>();

      for(int i=0;i<n;i++)
      {
         String s=sc.next();
         al.add(s);
      }

      Collections.sort(al,(s1,s2)->s1.length()-s2.length());
      System.out.print(al);
   }
}
```

Status : Correct ***Marks : 10/10***

6. Playing cards during travel is a fun-filled experience. For this game, they wanted to collect all four unique symbols. Can you help these guys to collect unique symbols from a set of cards?

Create Card class with attributes symbol and number. From our main method collect each card details (symbol and number) from the user. Collect all these cards in a set, since the set is used to store unique values or objects. Cards need to be compared with each other to identify whether both the cards are the same symbol. For this, we need to implement equals method to tell whether both the cards are same or not.

Once we collect all four different symbols display the first occurrence of card details in alphabetical order.

Answer

```
// You are using Java
import java.util.*;
```

```
class Card
{
    char symbol;
    int num;
    public void setter(char symbol, int num)
    {
        this.symbol=symbol;
        this.num=num;
    }
}
class CardComparator implements Comparator<Card>
{
    public int compare(Card o1, Card o2)
    {

         return Character.compare(o1.symbol,o2.symbol);

    }
}
class Main
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        Set<Card>t=new TreeSet<Card>(new CardComparator());
        int c=0;
        while(sc.hasNext())
        {
            Card obj=new Card();
            char symbol=sc.next().charAt(0);
            int num=sc.nextInt();
            obj.setter(symbol,num);
            t.add(obj);
            c++;

        }
        System.out.println("Four symbols gathered in "+c+" cards");
        System.out.println("Cards in Set are: ");
        Iterator<Card>it=t.iterator();
        while(it.hasNext())
        {
            Card newob=it.next();
         System.out.println(newob.symbol+" "+newob.num);
      }
   }
}
```

Status : Correct ***Marks : 10/10***

7. Create a class Student with the following private attributes.

rno, mark1, mark2,mark3 as integername as Stringavg as DoubleIncluded appropriate getters/setters, also include constructors. And override the toString method to print details of the Student object. Include a method to calculate the average.

Create a main class ArrayListSort and get roll number, name, and marks of 3 subjects from the user using ArrayList and calculate average. Sort the students based on average and display.

Format: "%-20s%-20s%-20s".

Answer

```
// You are using Java
import java.util.*;
class ComparatorDemo implements Comparator<Student>
{
   public int compare(Student o1, Student o2)
   {
      return Double.compare(o1.getteravg(),o2.getteravg());
   }
}
class Student
{
   private int rno;
   private int m1;
   private int m2;
   private int m3;
   private String name;
   private double avg;
   public void setter(int rno, int m1, int m2, int m3,String name, double avg)
   {
      this.rno=rno;
```

```
        this.m1=m1;
        this.m2=m2;
        this.m3=m3;
        this.name=name;
        this.avg=avg;
    }
    public int getterrno()
    {
        return rno;
    }
    public int getterm1()
    {
        return m1;
    }
    public int getterm2()
    {
        return m2;
    }
    public int getterm3()
    {
        return m3;
    }
    public String gettername()
    {
        return name;
    }
    public double getteravg()
    {
        return avg;
    }
}
class Main
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        ArrayList<Student>al=new ArrayList<Student>();
        for(int i=0;i<n;i++)
        {
            Student obj=new Student();
            int rno=sc.nextInt();
            sc.nextLine();
            String name=sc.next();
            int m1=sc.nextInt();
            int m2=sc.nextInt();
            int m3=sc.nextInt();
            double avg=(m1+m2+m3)/3;
            obj.setter(rno,m1,m2,m3,name,avg);
            al.add(obj);
        }
        Comparator newobj=new ComparatorDemo();
        Collections.sort(al,newobj);
        System.out.printf("%-20s%-20s%-20s","Roll No","Name","Average");
        System.out.println();
        Iterator<Student>it=al.iterator();
        while(it.hasNext())
        {
            Student newit=it.next();
            System.out.printf("%-20s%-20s
%-20s",newit.getterrno(),newit.gettername(),newit.getteravg());
            System.out.println();
        }

    }
}
```

Status : Correct ***Marks : 10/10***

8. Using Java Library ArrayList as a List Interface implementation, input N integers from standard input and add to the list only if they form an increasing sequence.

Take a number, N > 0 as inputAccept N integers as input Add the number to the list only if it forms an increasing sequence else ignorePrint the list

Answer

```
// You are using Java
import java.util.*;

class Main
{
    public static void addele(ArrayList<Integer> al1,int num)
    {
        if((al1.isEmpty()) || num>al1.get(al1.size()-1))
        {
            al1.add(num);
        }
    }

    public static void main(String[] arg)
    {
        Scanner sc= new Scanner(System.in);

        int no=sc.nextInt();
        ArrayList<Integer> al=new ArrayList();
        ArrayList<Integer> al1=new ArrayList();

        for(int i=1;i<=no;i++)
        {
            al.add(sc.nextInt());
        }
        for(int i=0;i<al.size();i++)
        {
            addele(al1,al.get(i));
        }
        System.out.println(al1.toString());
    }
}
```

Status : Correct ***Marks : 10/10***