Dt : 6/10/2023

Ex-program:(Demonstrating Some methods from Set<E>)

Program : DemoSet3.java

```
package p2;
import java.util.*;
public class DemoSet3 {
     @SuppressWarnings("removal")
     public static void main(String[] args) {
        LinkedHashSet<Integer> ob1=new
LinkedHashSet<Integer>();
        LinkedHashSet<Integer> ob2=new
LinkedHashSet<Integer>();
        LinkedHashSet<Integer> ob3=new
LinkedHashSet<Integer>();
        ob1.add(new Integer(11));
        ob1.add(new Integer(12));
        ob1.add(new Integer(13));
        ob1.add(new Integer(14));
        System.out.println("ob1 : "+ob1.toString());
        ob2.add(new Integer(21));
        ob2.add(new Integer(22));
        ob2.add(new Integer(23));
        ob2.add(new Integer(24));
        System.out.println("ob2 : "+ob2.toString());
        ob3.add(new Integer(21));
        ob3.add(new Integer(22));
        System.out.println("ob3 : "+ob3.toString());
        boolean b1 = ob1.contains(new Integer(11));
        System.out.println("ob1 holding ele 11 : "+b1);
        boolean b2 = ob2.containsAll(ob3);
        System.out.println("ob2 contains all eles of ob3
: "+b2);
        ob1.addAll(ob3);
        System.out.println("ob1 : "+ob1.toString());
        ob1.remove(new Integer(11));
        System.out.println("ob1 : "+ob1.toString());
```

Venkatesh Maipathii

```
        ob1.removeAll(ob3);
        System.out.println("ob1 : "+ob1.toString());
        ob2.retainAll(ob3);
        System.out.println("ob2 : "+ob2.toString());
        ob1.clear();
        System.out.println("ob1 : "+ob1.toString());
        System.out.println("ob2 : "+ob2.toString());
        System.out.println("ob3 : "+ob3.toString());
        System.out.println("====Immutable Set
Object====");
        Set<Integer> ob4 = Set.of(20,30,40);
                        //Java8 - Version
        System.out.println("ob4 : "+ob4.toString());
        //ob4.add(new Integer(60));//Error

    }
}
```

o/p:

ob1 : [11, 12, 13, 14]

ob2 : [21, 22, 23, 24]

ob3 : [21, 22]

ob1 holding ele 11 : true

ob2 contains all eles of ob3 : true

ob1 : [11, 12, 13, 14, 21, 22]

ob1 : [12, 13, 14, 21, 22]

ob1 : [12, 13, 14]

ob2 : [21, 22]

ob1 : []

Venkatesh Maipathi

ob2 : [21, 22]

ob3 : [21, 22]

====Immutable Set Object====

ob4 : [20, 30, 40]

==========

**imp*

2.List<E>:

=>List<E> is an interface from java.util package and which organizes elements based on index values and which can hold duplicate elements.

=>The following are some important methods of List<E>:

```
public abstract int size();
public abstract boolean isEmpty();
public abstract boolean contains(java.lang.Object);
public abstract boolean add(E);
public abstract boolean remove(java.lang.Object);
public abstract boolean containsAll(java.util.Collection<?>);
public abstract boolean addAll(java.util.Collection<? extends E>);
public abstract boolean addAll(int, java.util.Collection<? extends E>);
public abstract boolean removeAll(java.util.Collection<?>);
public abstract boolean retainAll(java.util.Collection<?>);
public default void replaceAll(java.util.function.UnaryOperator<E>);
public default void sort(java.util.Comparator<? super E>);
```

Venkatesh Maipathi

public abstract void clear();

public abstract E get(int);

public abstract E set(int, E);

public abstract void add(int, E);

public abstract E remove(int);

public abstract int indexOf(java.lang.Object);

public abstract int lastIndexOf(java.lang.Object);

public abstract java.util.List<E> subList(int, int);

public static <E> java.util.List<E> of();

public abstract java.util.Iterator<E> iterator();

public abstract java.util.ListIterator<E> listIterator();

public abstract java.util.ListIterator<E> listIterator(int);

public default java.util.Spliterator<E> spliterator();

public abstract java.lang.Object[] toArray();

public abstract <T> T[] toArray(T[]);

=>The following are the implementation classes of List<E>:

(a)ArrayList<E>

(b)LinkedList<E>

(c)Vector<E>

Venkatesh Maipathii

(a)ArrayList<E>:

=>ArrayList<E> organizes elements in sequence and which is NonSynchronized class.

=>ArrayList<E> is the replacement of Array.

syntax:

ArrayList<Class_name> al = new ArrayList<Class_name>();

Ex-program:

wap to perform the following operations on Products.

1.add

2.remove

3.get

4.set

...

SubClass : Product.java

```
package p1;
public class Product extends Object
{
  public String code,name;
  public float price;
  public int qty;
  public Product(String code,String name,float
price,int qty)
  {
```

Venkatesh Maipathii

```
        this.code=code;
        this.name=name;
        this.price=price;
        this.qty=qty;
    }
    @Override
    public String toString()
    {
        return code+"\t"+name+"\t"+price+"\t"+qty;
    }
}
```

MainClass : DemoList1.java

```
package p2;

import java.util.*;

import p1.*;

public class DemoList1

{

        public static void main(String[] args)

        {

     Scanner s = new Scanner(System.in);

     try(s;){

          try {

                ArrayList<Product> al = new ArrayList<Product>();

                while(true)

                {

                      System.out.println("****Choice****");
```

Venkatesh Maipathii

```
System.out.println("\t1.add(E)"
                + "\n\t2.add(index,E)"
                + "\n\t3.remove(Object)"
                + "\n\t4.remove(index)"
                + "\n\t5.get(index)"
                + "\n\t6.set(index,E)"
                + "\n\t7.Exit");
System.out.println("Enter the Choice:");
int choice = Integer.parseInt(s.nextLine());
switch(choice)
{
case 1:
        System.out.println("===ProductDetails===");
        System.out.println("Enter the Product Code:");
        String code = s.nextLine();
        System.out.println("Enter the Product Name:");
        String name = s.nextLine();
        System.out.println("Enter the Product Price:");
        float price = Float.parseFloat(s.nextLine());
        System.out.println("Enter the Product Qty:");
        int qty = Integer.parseInt(s.nextLine());
        al.add(new Product(code,name,price,qty));
```

Venkatesh Maipathii

```
                    System.out.println("Product added....");
                    al.spliterator().forEachRemaining((k)->
                    {
                            System.out.println(k.toString());
                    });
                    break;
                case 2:
                    if(al.isEmpty()) {
                            System.out.println("List is empty...");
                    }else {
                            System.out.println("Enter the index to add
ele:");
                            int i1 = Integer.parseInt(s.nextLine());
                            if(i1>=0 && i1<al.size()) {

System.out.println("====ProductDetails===");
                                    System.out.println("Enter the Product
Code:");
                            String code1 = s.nextLine();
                            System.out.println("Enter the Product Name:");
                            String name1 = s.nextLine();
                            System.out.println("Enter the Product Price:");
                            float price1 = Float.parseFloat(s.nextLine());
```

Venkatesh Maipathii

```
                    System.out.println("Enter the Product Qty:");
                    int qty1 = Integer.parseInt(s.nextLine());
                        al.add(i1, new
Product(code1,name1,price1,qty1));
                        System.out.println("Product added by
index...");
                        al.spliterator().forEachRemaining((k)->
                    {
                         System.out.println(k.toString());
                    });
                    }else {
                        System.out.println("Invalid index...");
                    }
                }
                break;
            case 3:
                if(al.isEmpty()) {
                        System.out.println("List is empty...");
                }else {
                        System.out.println("Enter the ProdCode to
remove Product:");
                        String pC = s.nextLine();
                        boolean b = false;
```

Venkatesh Maipathi

```
Iterator<Product> it = al.iterator();
while(it.hasNext())
{
        Product pr = it.next();
        if(pC.equals(pr.code))
        {
            it.remove();
            b = true;
            break;//stop the loop
        }
}//end of loop
if(b) {
System.out.println("Product removed...");
al.spliterator().forEachRemaining((k)->
{
        System.out.println(k.toString());
});
}//end of if
else {
        System.out.println("ProductCode not
Available...");
}
```

Venkatesh Maipathii

```
                }
                break;
            case 4:
                if(al.isEmpty()) {
                    System.out.println("List is empty...");
                }else {
                    System.out.println("Enter the index to remove
the Product:");
                    int i2 = Integer.parseInt(s.nextLine());
                    if(i2>=0 && i2<al.size()){
                        al.remove(i2);
                        System.out.println("Product removed..");
                        al.spliterator().forEachRemaining((k)->
                        {
                            System.out.println(k.toString());
                        });
                    }else {
                        System.out.println("Invalid index...");
                    }
                }
                break;
            case 5:
```

Venkatesh Maipathii

```
                    if(al.isEmpty()) {
                        System.out.println("List is empty...");
                    }else {
                        System.out.println("Enter the index to get the
Product:");
                        int i3 = Integer.parseInt(s.nextLine());
                        if(i3>=0 && i3<al.size()) {
                            Product p = al.get(i3);
                            System.out.println(p.toString());
                        }else {
                            System.out.println("Invalid index...");
                        }
                    }
                    break;
                case 6:
                    if(al.isEmpty()) {
                        System.out.println("List is empty...");
                    }else {
                        System.out.println("Enter the index to set
Product:");
                        int i4 = Integer.parseInt(s.nextLine());
                        if(i4>=0 && i4<al.size()) {
```

Venkatesh Maipathii

```
System.out.println("====ProductDetails====");
                                        System.out.println("Enter the Product
Code:");
                                    String code2 = s.nextLine();
                                    System.out.println("Enter the Product Name:");
                                    String name2 = s.nextLine();
                                    System.out.println("Enter the Product Price:");
                                    float price2 = Float.parseFloat(s.nextLine());
                                    System.out.println("Enter the Product Qty:");
                                    int qty2 = Integer.parseInt(s.nextLine());
                                    al.set(i4, new
Product(code2,name2,price2,qty2));
                                    System.out.println("Product setted...");
                                    al.spliterator().forEachRemaining((k)->
                                    {
                                        System.out.println(k.toString());
                                    });
                                    }else {
                                        System.out.println("Invalid Index..");
                                    }

                                }
```

Venkatesh Maipathi

```
                    break;
                case 7:
                    System.out.println("Program Stopped...");
                    System.exit(0);
                default:
                    System.out.println("Invalid Choice...");
                }//End of switch
            }//end of loop
        }catch(Exception e) {e.printStackTrace();}
    }//end of try with resource
    }
}
```

==========

Venkatesh Maipathii