**Lab Session 01: LISTS USING ARRAYS**

|  |
| --- |
| **CODE:**  package lab.pkg01;  import java.util.\*;  public class Lab01 {  public static void main(String[] args) {  // String Arrays that are to be used for creating MyArrayList  String[] L1 = {"Abu Dhabi", "Dubai", "Sharjah", "Ajman"};  String[] L2 = {"Fujara", "Dubai", "Ras Alkhaima", "AlAin"};    // Two MyArrayLists, list1 and list2 are created  MyArrayList list1 = new MyArrayList(L1);  MyArrayList list2 = new MyArrayList(L2);    // addAll() function is being called  System.out.println("1. Invoke list1.addAll(list2), and displays list1 and list2");  list1.addAll(list2);  System.out.println("\n\tList1: "+list1.toString());  System.out.println("\n\tList2: "+list2.toString());    // Recreate list1 and list2 with same initial values  list1 = new MyArrayList(L1);  list2 = new MyArrayList(L2);    // removeAll() function is being called  System.out.println("\n2. Invoke list1.removeAll(list2), and displays list1 and list2");  list1.removeAll(list2);  System.out.println("\n\tList1: "+list1.toString());  System.out.println("\n\tList2: "+list2.toString());    // Recreate list1 and list2 with same initial values  list1 = new MyArrayList(L1);  list2 = new MyArrayList(L2);    // retainALL() function is being called  System.out.println("\n2. Invoke list1.retainAll(list2), and displays list1 and list2");  list1.retainAll(list2);  System.out.println("\n\tList1: "+list1.toString());  System.out.println("\n\tList2: "+list2.toString());  }  }  class MyArrayList{  static int size=0; // size counts the number of elements in MyList  List<String> MyList = new ArrayList<>();    /\* Constructor of MyArrayList is initialized which will create MyArrayList with initial values \*/  MyArrayList(String[] list){  for(String i:list){  MyList.add(i);  size++;  }  }    /\* Checks whether MyList contains the string s, and return the positions where string s occurs in MyList \*/  public ArrayList<Integer> contains(String e){  boolean b = false;  ArrayList<Integer> positions = new ArrayList<>(); /\* arraylist to save the poistions where string s occurs in MyList \*/  for (int i=0; i<size; i++){  if(MyList.get(i).equals(e)){  positions.add(i);  b = true;  }  }  return positions;  }    // returns MyList array of the object by which it is called.  public List<String> returnList(){  return MyList;  }    // adds the elements of MyArrayList which are not present in MyList  public boolean addAll(MyArrayList e){  boolean isChanged = false;  List<String> list = e.returnList();  for(String s:list){  if(!MyList.contains(s)) {  MyList.add(s);  isChanged = true;  }  }  return isChanged; // return true if MyList is updated, else returns false.  }    // retruns true if MyList contains values that are present in MyArrayList e, else False.  public boolean removeAll(MyArrayList e){  boolean isChanged = false;  List<String> list = e.returnList();  for(String s: list){  if(MyList.contains(s)) {  MyList.remove(s);  isChanged = true;  }  }  return isChanged ;  }    // give the values that are common in both lists and overwrite MyList with that value.  public void retainAll(MyArrayList e){  List<String> list = e.returnList();  MyList.retainAll(list);  }    // displays MyList  @Override  public String toString(){  String str="";  for (String s:MyList){  str = s+", "+str;  }  return str;  }  } |
| **SCREENSHOT** |