/\*EX 8 \*/

/\*1. Write a program to perform string operations using ArrayList. Write functions for the given operations.\*/

import java.util.\*;

class operations

{

ArrayList<String> str = new ArrayList<String>();

Scanner input = new Scanner(System.in);

void AppendLast()

{

String a;

System.out.print("Enter the String : ");

a = input.next();

str.add(a);

System.out.println("-------APPENDED-------");

}

void Append()

{

int i;

String a;

System.out.print("Enter the index where you want to insert the string : ");

i = input.nextInt();

System.out.print("Enter the String : ");

a = input.next();

str.add(i,a);

System.out.println("-------APPENDED-------");

}

int Search(String a)

{

return str.indexOf(a);

}

void display()

{

System.out.println("The Elements in Array List are : ");

for(int i=0;i<str.size();i++)

{

System.out.println(str.get(i));

}

}

void StartsWith()

{

char s;

System.out.print("Enter the character to find the word starting as given character : ");

s = input.next().charAt(0);

System.out.println("The element starts with "+s+" are : ");

for(int i=0;i<str.size();i++)

{

if(str.get(i).charAt(0) == s)

{

System.out.println(str.get(i));

}

}

}

void substring()

{

String a;

System.out.print("Enter the substring : ");

a = input.next();

System.out.println("The element which contains the substring "+a+" are : ");

for(int i=0;i<str.size();i++)

{

if(str.get(i).contains(a))

{

System.out.println(str.get(i));

}

}

}

void sort()

{

Collections.sort(str);

System.out.println("-------SORTED-------");

}

void remove()

{

String a;

System.out.print("Enter the String to be removed : ");

a = input.next();

int i = Search(a);

if( i == -1)

System.out.print("Array List Empty\n");

else

{

str.remove(i);

System.out.println("-------REMOVED-------");

}

}

void replace()

{

String a,b;

System.out.print("Enter the String to be replaced : ");

a = input.next();

System.out.print("Enter the new String: ");

b = input.next();

int i = Search(a);

if( i == -1)

System.out.print("Array List Empty\n");

else

{

str.set(i,b);

System.out.println("-------REPLACED-------");

}

}

void removeDuplicate()

{

ArrayList <String> arraySet = new ArrayList<String>(str.size());

for(int i=0;i<str.size();i++)

if(!arraySet.contains(str.get(i)))

arraySet.add(str.get(i));

str = arraySet;

System.out.println("-------REMOVED DUPLICATE-------");

}

}

class ArrayList1

{

public static void main(String args[])

{

Scanner scan=new Scanner(System.in);

operations s=new operations();

System.out.println("\n------------- String List Menu S-------------\n1. Append at end\n2. Insert at index\n3. Find index\n4. Display list\n5. List strings starting with a given letter\n6. List strings containing a given substring\n7. Sort the ArrayList\n8. Remove an element\n9. Replace an element\n10. Remove duplicate elements\n11. Exit");

while(true)

{

System.out.print("\nEnter the operation to be performed: ");

int op;

op=scan.nextInt();

if(op == 11)

{

System.out.println("\nExiting...");

break;

}

switch(op)

{

case 1: s.AppendLast();

break;

case 2: s.Append();

break;

case 3: System.out.print("Enter String to be searched: ");

scan.nextLine();

int index=s.Search(scan.nextLine());

System.out.println("The string is found at position "+index);

break;

case 4: s.display();

break;

case 5: s.StartsWith();

break;

case 6: s.substring();

break;

case 7: s.sort();

break;

case 8: s.remove();

break;

case 9: s.replace();

break;

case 10: s.removeDuplicate();

break;

default : System.out.println("\nInvalid Choice! Enter again.");

}

}

}

}

/\*

output :

------------- String List Menu S-------------

1. Append at end

2. Insert at index

3. Find index

4. Display list

5. List strings starting with a given letter

6. List strings containing a given substring

7. Sort the ArrayList

8. Remove an element

9. Replace an element

10. Remove duplicate elements

11. Exit

Enter the operation to be performed: 1

Enter the String : Bharath

-------APPENDED-------

Enter the operation to be performed: 1

Enter the String : AMR

-------APPENDED-------

Enter the operation to be performed: 2

Enter the index where you want to insert the string : 1

Enter the String : Kumar

-------APPENDED-------

Enter the operation to be performed: 3

Enter String to be searched: AMR

The string is found at position 2

Enter the operation to be performed: 4

The Elements in Array List are :

Bharath

Kumar

AMR

Enter the operation to be performed: 5

Enter the character to find the word starting as given character : A

The element starts with A are :

AMR

Enter the operation to be performed: 6

Enter the substring : ar

The element which contains the substring ar are :

Bharath

Kumar

Enter the operation to be performed: 7

-------SORTED-------

Enter the operation to be performed: 8

Enter the String to be removed : AMR

-------REMOVED-------

Enter the operation to be performed: 9

Enter the String to be replaced : Kumar

Enter the new String: kumar

-------REPLACED-------

Enter the operation to be performed: 10

-------REMOVED DUPLICATE-------

Enter the operation to be performed: 11

Exiting...

\*/

2. Write a program to get two integer ArrayList. Perform the following operations on it.

import java.util.\*;

class ArrayListOperations

{

    ArrayList<Integer> array1 = new ArrayList<Integer>();

    ArrayList<Integer> array2 = new ArrayList<Integer>();

    public void addElement1(int data)

    {

        array1.add(data);

    }

    public void addElement2(int data)

    {

        array2.add(data);

    }

    public void mergeLists(ArrayList<Integer> mergeArr)

    {

        mergeArr.addAll(array1);

        mergeArr.addAll(array2);

        System.out.println("\nThe Lists are merged successfully.");

        System.out.println(mergeArr);

    }

    public void listUnion(ArrayList<Integer> arraySet)

    {

        arraySet.addAll(array1);

        for(int i=0;i<array2.size();i++)

            if(!arraySet.contains(array2.get(i)))

                arraySet.add(array2.get(i));

        System.out.print("\nThe union of the two lists is : ");

        System.out.println(arraySet);

    }

    public void listIntersection(ArrayList<Integer> intersecArr)

    {

        intersecArr.addAll(array1);

        intersecArr.retainAll(array2);

        System.out.println("\nThe intersection of the two lists is : ");

        System.out.println(intersecArr);

    }

}

class ArrayListTwo

{

    public static void main(String args[])

    {

        Scanner scan = new Scanner(System.in);

        ArrayListOperations i = new ArrayListOperations();

        while(true)

        {

            System.out.println("\n--------> Integer Array List <--------\n1. Add elements to Array 1\n2. Add elements to Array 2\n3. Merge the two lists\n4. Union of the two lists\n5. Intersection of the two lists\n6. Exit");

            int op;

            System.out.print("Enter operation: ");

            op=scan.nextInt();

            if(op == 6)

            {

                System.out.println("\nExiting...");

                break;

            }

            ArrayList<Integer> union=new ArrayList<Integer>();

            ArrayList<Integer> merge=new ArrayList<Integer>();

            ArrayList<Integer> intersection=new ArrayList<Integer>();

            switch(op)

            {

                case 1: System.out.print("\nEnter the element: ");

                        int data = scan.nextInt();

                        i.addElement1(data);

                        break;

                case 2: System.out.print("\nEnter the element: ");

                        data = scan.nextInt();

                        i.addElement2(data);

                        break;

                case 3: i.mergeLists(merge);

                        break;

                case 4: i.listUnion(union);

                        break;

                case 5: i.listIntersection(intersection);

                        break;

                default: System.out.println("\nInvalid option! Enter again.");

            }

        }

        System.out.println();

    }

}

/\* Output:

--------> Integer Array List <--------

1. Add elements to Array 1

2. Add elements to Array 2

3. Merge the two lists

4. Union of the two lists

5. Intersection of the two lists

6. Exit

Enter operation: 1

Enter the element: 1

--------> Integer Array List <--------

1. Add elements to Array 1

2. Add elements to Array 2

3. Merge the two lists

4. Union of the two lists

5. Intersection of the two lists

6. Exit

Enter operation: 1

Enter the element: 3

--------> Integer Array List <--------

1. Add elements to Array 1

2. Add elements to Array 2

3. Merge the two lists

4. Union of the two lists

5. Intersection of the two lists

6. Exit

Enter operation: 1

Enter the element: 5

--------> Integer Array List <--------

1. Add elements to Array 1

2. Add elements to Array 2

3. Merge the two lists

4. Union of the two lists

5. Intersection of the two lists

6. Exit

Enter operation: 2

Enter the element: 2

--------> Integer Array List <--------

1. Add elements to Array 1

2. Add elements to Array 2

3. Merge the two lists

4. Union of the two lists

5. Intersection of the two lists

6. Exit

Enter operation: 2

Enter the element: 4

--------> Integer Array List <--------

1. Add elements to Array 1

2. Add elements to Array 2

3. Merge the two lists

4. Union of the two lists

5. Intersection of the two lists

6. Exit

Enter operation: 2

Enter the element: 6

--------> Integer Array List <--------

1. Add elements to Array 1

2. Add elements to Array 2

3. Merge the two lists

4. Union of the two lists

5. Intersection of the two lists

6. Exit

Enter operation: 3

The Lists are merged successfully.

[1, 3, 5, 2, 4, 6]

--------> Integer Array List <--------

1. Add elements to Array 1

2. Add elements to Array 2

3. Merge the two lists

4. Union of the two lists

5. Intersection of the two lists

6. Exit

Enter operation: 1

Enter the element: 2

--------> Integer Array List <--------

1. Add elements to Array 1

2. Add elements to Array 2

3. Merge the two lists

4. Union of the two lists

5. Intersection of the two lists

6. Exit

Enter operation: 2

Enter the element: 3

--------> Integer Array List <--------

1. Add elements to Array 1

2. Add elements to Array 2

3. Merge the two lists

4. Union of the two lists

5. Intersection of the two lists

6. Exit

Enter operation: 4

The union of the two lists is : [1, 3, 5, 2, 4, 6]

--------> Integer Array List <--------

1. Add elements to Array 1

2. Add elements to Array 2

3. Merge the two lists

4. Union of the two lists

5. Intersection of the two lists

6. Exit

Enter operation: 5

The intersection of the two lists is :

[3, 2]

--------> Integer Array List <--------

1. Add elements to Array 1

2. Add elements to Array 2

3. Merge the two lists

4. Union of the two lists

5. Intersection of the two lists

6. Exit

Enter operation: 6

Exiting...\*/