|  |  |
| --- | --- |
| package labex\_12; | |
|  | |  | |
|  | | import java.util.Scanner; | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | | public class search\_elt { | |
|  | |  | |
|  | | /\*\* | |
|  | | \* @param args the command line arguments | |
|  | | \*/ | |
|  | | public static void main(String[] args) { | |
|  | | // TODO code application logic here | |
|  | | Scanner val=new Scanner(System.in); | |
|  | |  | |
|  | | //1...SEARCHING ELEMENT IN INTEGER DATA TYPE | |
|  | |  | |
|  | | //CREATING INTEGER ARRAY OF OBJECTS | |
|  | | System.out.println("==========INTEGER=========="); | |
|  | | System.out.println("Enter the Integer array size:"); | |
|  | | int size=val.nextInt(); | |
|  | | Integer num[]=new Integer[size]; | |
|  | | System.out.println("Enter the Integer data to create Integer array:"); | |
|  | | for(int i=0;i<size;i++) | |
|  | | { | |
|  | | num[i]=val.nextInt(); | |
|  | | } | |
|  | |  | |
|  | | //DISPLAYING ARRAY | |
|  | | System.out.println("==========ARRAY=========="); | |
|  | | for(Integer i:num) | |
|  | | System.out.println(i); | |
|  | |  | |
|  | | //SEARCH ELEMENT | |
|  | | System.out.println("Enter the data to be searched:"); | |
|  | | int elt1=val.nextInt(); | |
|  | |  | |
|  | | //TO CALL SEARCHING METHOD | |
|  | | Search<Integer,Integer> s1=new Search<>(); | |
|  | | s1.search(num, elt1); | |
|  | |  | |
|  | | //2...SEARCHING ELEMENT IN STRING DATA TYPE | |
|  | |  | |
|  | | //CREATING STRING ARRAY OF OBJECTS | |
|  | | System.out.println("==========STRING=========="); | |
|  | | System.out.println("Enter the String array size:"); | |
|  | | size=val.nextInt(); | |
|  | | String str[]=new String[size]; | |
|  | | System.out.println("Enter the String data to create String array:"); | |
|  | | for(int i=0;i<size;i++) | |
|  | | { | |
|  | | str[i]=val.next().toLowerCase(); | |
|  | |  | |
|  | | } | |
|  | |  | |
|  | | //DISPLAYING ARRAY | |
|  | | System.out.println("==========ARRAY=========="); | |
|  | | for(String i:str) | |
|  | | { | |
|  | | System.out.println(i); | |
|  | |  | |
|  | | } | |
|  | |  | |
|  | |  | |
|  | | //SEARCH ELEMENT | |
|  | | System.out.println("Enter the data to be searched:"); | |
|  | | String elt2=val.next().toLowerCase(); | |
|  | |  | |
|  | |  | |
|  | | //TO CALL SEARCHING METHOD | |
|  | | Search<String,String> s2=new Search<>(); | |
|  | | s2.search(str, elt2); | |
|  | |  | |
|  | | //2...SEARCHING ELEMENT IN CHARACTER DATA TYPE | |
|  | |  | |
|  | | //CREATING CHARACTER ARRAY OF OBJECTS | |
|  | | System.out.println("==========CHARACTER=========="); | |
|  | | System.out.println("Enter the Character array size:"); | |
|  | | size=val.nextInt(); | |
|  | | Character chars[]=new Character[size]; | |
|  | | System.out.println("Enter the Character data to create Character array:"); | |
|  | | for(int i=0;i<size;i++) | |
|  | | { | |
|  | | chars[i]=val.next().toLowerCase().charAt(0); | |
|  | |  | |
|  | | } | |
|  | |  | |
|  | | //DISPLAYING ARRAY | |
|  | | System.out.println("==========ARRAY=========="); | |
|  | | for(Character i:chars) | |
|  | | { | |
|  | | System.out.println(i); | |
|  | | } | |
|  | |  | |
|  | |  | |
|  | | //SEARCH ELEMENT | |
|  | | System.out.println("Enter the data to be searched:"); | |
|  | | char elt3=val.next().toLowerCase().charAt(0); | |
|  | |  | |
|  | | //TO CALL SEARCHING METHOD | |
|  | | Search<Character,Character> s3=new Search<>(); | |
|  | | s3.search(chars, elt3); | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | | } | |
|  | |  | |
|  | | } | |
|  | |  | |
|  | | //generic class | |
|  | | class Search<T,S> | |
|  | | { | |
|  | | T[] array; | |
|  | | S value; | |
|  | | boolean flag=false; | |
|  | |  | |
|  | | //generic method for all data types | |
|  | | public <T,S>void search(T[] array,S value) | |
|  | | { | |
|  | | for(T i:array) | |
|  | | { | |
|  | | if(value.equals(i)) | |
|  | | { | |
|  | | flag=true; | |
|  | | break; | |
|  | | } | |
|  | | } | |
|  | |  | |
|  | | if(flag) | |
|  | | System.out.println("Searched element '"+value+"' found /\*(@ \_ @)\*/"); | |
|  | | else | |
|  | | System.out.println("Searched element not found {=\_=}"); | |
|  | | } | |
|  | | } | |

|  |  |
| --- | --- |
| package labex\_12; | |
|  | |  | |
|  | | import java.util.Arrays; | |
|  | | import java.util.List; | |
|  | | import java.util.Scanner; | |
|  | |  | |
|  | | /\*\* | |
|  | | \* | |
|  | | \* @author | |
|  | | \*/ | |
|  | | public class generic\_methods { | |
|  | |  | |
|  | | /\*\* | |
|  | | \* @param args the command line arguments | |
|  | | \*/ | |
|  | | public static void main(String[] args) { | |
|  | | // TODO code application logic here | |
|  | | Scanner val=new Scanner(System.in); | |
|  | |  | |
|  | | //CREATING INTEGER ARRAY OF OBJECTS | |
|  | | System.out.println("==========INTEGER=========="); | |
|  | | System.out.println("Enter the Integer array size:"); | |
|  | | int size=val.nextInt(); | |
|  | | Integer num[]=new Integer[size]; | |
|  | | System.out.println("Enter the Integer data to create Integer array:"); | |
|  | | for(int i=0;i<size;i++) | |
|  | | { | |
|  | | num[i]=val.nextInt(); | |
|  | | } | |
|  | |  | |
|  | | //GENERIC METHODS CALLING | |
|  | | prop\_1<Integer> obj1=new prop\_1<>(); | |
|  | | System.out.println("\n>>>>>>>>>COUNTING NO. OF ODD INTEGERS>>>>>>>>>"); | |
|  | | obj1.count\_of\_odd\_int(num); | |
|  | |  | |
|  | | prop\_2\_3<Integer> obj2=new prop\_2\_3<>(); | |
|  | | System.out.println("\n>>>>>>>>>INTERCHANGING ELEMENTS IN 2 DIFFERENT POSITIONS>>>>>>>>>"); | |
|  | | System.out.println("Enter the two positions whose elements are to be interchanged:"); | |
|  | | int m=val.nextInt(); | |
|  | | int n=val.nextInt(); | |
|  | | obj2.swap(num, m, n); | |
|  | |  | |
|  | | System.out.println("\n>>>>>>>>>FINDING MAXIMAL ELEMENT>>>>>>>>>"); | |
|  | | System.out.println("Enter the Start and End index for the range in which Maximal element is to be found:"); | |
|  | | m=val.nextInt(); | |
|  | | n=val.nextInt(); | |
|  | | obj2.maximal(num, m, n); | |
|  | |  | |
|  | | //CREATING STRING ARRAY OF OBJECTS | |
|  | | System.out.println("\n-\*-\*-\*-\*-\*-\*-\*--\*-\*-\*-\*-\*-\*-\*--\*-\*-\*-\*-\*-\*-\*--\*-\*-\*-\*-\*-\*-\*-\n==========STRING=========="); | |
|  | | System.out.println("Enter the String array size:"); | |
|  | | size=val.nextInt(); | |
|  | | String str[]=new String[size]; | |
|  | | System.out.println("Enter the String data to create String array:"); | |
|  | | for(int i=0;i<size;i++) | |
|  | | { | |
|  | | str[i]=val.next(); | |
|  | |  | |
|  | | } | |
|  | |  | |
|  | | //GENERIC METHODS CALLING | |
|  | | prop\_2\_3<String> obj3=new prop\_2\_3<>(); | |
|  | | System.out.println("\n>>>>>>>>>INTERCHANGING ELEMENTS IN 2 DIFFERENT POSITIONS>>>>>>>>>"); | |
|  | | System.out.println("Enter the two positions whose elements are to be interchanged:"); | |
|  | | m=val.nextInt(); | |
|  | | n=val.nextInt(); | |
|  | | obj3.swap(str, m, n); | |
|  | |  | |
|  | | System.out.println("\n>>>>>>>>>FINDING MAXIMAL ELEMENT>>>>>>>>>"); | |
|  | | System.out.println("Enter the Start and End index for the range in which Maximal element is to be found:"); | |
|  | | m=val.nextInt(); | |
|  | | n=val.nextInt(); | |
|  | | obj3.maximal(str, m, n); | |
|  | |  | |
|  | | //CREATING CHARACTER ARRAY OF OBJECTS | |
|  | | System.out.println("\n-\*-\*-\*-\*-\*-\*-\*--\*-\*-\*-\*-\*-\*-\*--\*-\*-\*-\*-\*-\*-\*--\*-\*-\*-\*-\*-\*-\*-\n==========CHARACTER=========="); | |
|  | | System.out.println("Enter the Character array size:"); | |
|  | | size=val.nextInt(); | |
|  | | Character chars[]=new Character[size]; | |
|  | | System.out.println("Enter the Character data to create Character array:"); | |
|  | | for(int i=0;i<size;i++) | |
|  | | { | |
|  | | chars[i]=val.next().charAt(0); | |
|  | |  | |
|  | | } | |
|  | |  | |
|  | | //GENERIC METHODS CALLING | |
|  | | prop\_2\_3<Character> obj4=new prop\_2\_3<>(); | |
|  | | System.out.println("\n>>>>>>>>>INTERCHANGING ELEMENTS IN 2 DIFFERENT POSITIONS>>>>>>>>>"); | |
|  | | System.out.println("Enter the two positions whose elements are to be interchanged:"); | |
|  | | m=val.nextInt(); | |
|  | | n=val.nextInt(); | |
|  | | obj4.swap(chars, m, n); | |
|  | |  | |
|  | | System.out.println("\n>>>>>>>>>FINDING MAXIMAL ELEMENT>>>>>>>>>"); | |
|  | | System.out.println("Enter the Start and End index for the range in which Maximal element is to be found:"); | |
|  | | m=val.nextInt(); | |
|  | | n=val.nextInt(); | |
|  | | obj4.maximal(chars, m, n); | |
|  | |  | |
|  | | } | |
|  | |  | |
|  | | } | |
|  | |  | |
|  | | //BOUNDED CLASS TO FIND COUNT OF ODD INTEGERS | |
|  | | class prop\_1<I extends Number> | |
|  | | { | |
|  | | int count=0; | |
|  | |  | |
|  | | //GENERIC METHOD TO FIND COUNT OF ODD INTEGERS | |
|  | | public <I extends Number> void count\_of\_odd\_int(I[] numarr) | |
|  | | { | |
|  | | for(I i:numarr) | |
|  | | { | |
|  | | if(i.intValue()%2 != 0) | |
|  | | count++; | |
|  | | } | |
|  | | System.out.println("Count of Odd Integers= "+count); | |
|  | | } | |
|  | | } | |
|  | |  | |
|  | | //GENERIC CLASS | |
|  | | class prop\_2\_3<V> | |
|  | | { | |
|  | | //GENERIC METHOD TO INTERCHANGING ELEMENTS IN 2 DIFFERENT POSITIONS | |
|  | | public <V> void swap(V[] array,int index\_1,int index\_2) | |
|  | | { | |
|  | | System.out.println("Interchanging elements in Position '"+index\_1+"' & Position '"+index\_2+"' ........\nProcessing --------(=\_=)-------"); | |
|  | | for(int i=0;i<array.length;i++) | |
|  | | { | |
|  | | for(int j=0;j<array.length;j++) | |
|  | | { | |
|  | | if(index\_1==i && index\_2==j) | |
|  | | { | |
|  | | V temp=array[index\_1]; | |
|  | | array[index\_1]=array[index\_2]; | |
|  | | array[index\_2]=temp; | |
|  | | break; | |
|  | | } | |
|  | | } | |
|  | | } | |
|  | |  | |
|  | | System.out.println("\nInterchanged!!! (^\_^) ........\n=====INTERCHANGED ARRAY====="); | |
|  | | for (V ele: array) { | |
|  | | System.out.println(ele); | |
|  | | } | |
|  | |  | |
|  | | } | |
|  | |  | |
|  | | //GENERIC METHOD TO FINDING MAXIMAL ELEMENT | |
|  | | public <V extends Comparable> void maximal(V[] array,int begin,int end) | |
|  | | { | |
|  | | System.out.println("Finding Maximal element in the range ["+begin+","+end+"] ........\nProcessing --------(=\_=)-------"); | |
|  | |  | |
|  | | V max=array[begin]; | |
|  | | System.out.println("Elements in the range ["+begin+","+end+"]:"); | |
|  | | for(int i=begin;i<end;i++) | |
|  | | { | |
|  | | System.out.println(array[i]); | |
|  | | if(array[i].compareTo(max) > 0) | |
|  | | max=array[i]; | |
|  | | } | |
|  | | System.out.println("Found Maximal element!! \*($\_$)\*\nMaximal element is: "+max); | |
|  | | } | |
|  | | } | |