Exercise 8.16 solution: IntegerSet.java IntegerSet class definition

Posted on April 19, 2012 by iJava2Admin

Here is the main method for the Integerset program:

```
// Exercise 8.16 solution: IntegerSet.java
// IntegerSet class definition
import javax.swing.*;
public class IntegerSet {
private final int SETSIZE = 101;
private boolean[] set;

// no-argument constructor
public IntegerSet()
{
set = new boolean[ SETSIZE ];
}

// constructor creates a set from array of integers
public IntegerSet( int array[] )
{
set = new boolean[ SETSIZE ];

for( int i = 0; i < array.length; i++ )
insertElement( array[ i ] );
}

// input a set from the user
public void inputSet()
{
int number;
```

```
// get a set of integers from user
do {
number = Integer.parseInt( JOptionPane.showInputDialog(
"Enter an integer (-1 to terminate input):" ) );

if ( validEntry( number ) )
set[ number ] = true;

} while ( number != -1 );
}

// return string representation of set
public String toSetString()
{
int x = 1;
boolean empty = true; // assume set is empty
String setString = "{ ";

// get set elements
for ( int count = 0; count < 101; ++count ) {
if ( set[ count ] ) {
setString += count + " ";
empty = false; // set is not empty
++x;
}
}

// empty set
if ( empty )
setString += "---"; // display an empty set

setString += " }";
```

```
return setString;

} // end method toSetString
// returns the union of two sets
public IntegerSet union( IntegerSet integerSet )
{
IntegerSet temp = new IntegerSet();

for ( int count = 0; count < 101; ++count )
if ( set[ count ] || integerSet.set[ count ] )
temp.set[ count ] = true;

return temp;
}

// returns the intersection of two sets
public IntegerSet intersection( IntegerSet integerSet )
{
IntegerSet temp = new IntegerSet();

for ( int count = 0; count < 101; ++count )
if ( set[ count ] && integerSet.set[ count ] )
temp.set[ count ] = true;

return temp;
}

// insert a new integer into this set
public void insertElement( int insertInteger )
{
if ( validEntry( insertInteger ) )
```

```
set[ insertInteger ] = true;
}

// remove an integer from this set
public void deleteElement( int deleteInteger )
{
if ( validEntry( deleteInteger ) )
set[ deleteInteger ] = false;
}

// determine if two sets are equal
public boolean isEqualTo( IntegerSet integerSet )
{
for ( int count = 0; count < 101; ++count )
if ( set[ count ] != integerSet.set[ count ] )
return false; // sets are not-equal

return true; // sets are equal
}

// determine if input is valid
public boolean validEntry( int input )
{
return input >= 0 && input <= 100;
}
} // end class IntegerSet
```

Solved Create class IntegerSet. Each IntegerSet object can hold integers in the range 0–100.

Posted on April 19, 2012 by iJava2Admin

Create class IntegerSet. Each IntegerSet object can hold integers in the range 0–100.

The set is represented by an array of booleans. Array element a[i] is true if integer i is in the set.

Array element a[j] is false if integer j is not in the set. The no-argument constructor initializes the

Java array to the “empty set” (i.e., a set whose array representation contains all false values).

Provide the following methods: Method union creates a third set that is the set-theoretic union

of two existing sets (i.e., an element of the third set’s array is set to true if that element is true in

either or both of the existing sets; otherwise, the element of the third set is set to false). Method

intersection creates a third set which is the set-theoretic intersection of two existing sets (i.e., an

element of the third set’s array is set to false if that element is false in either or both of the existing

sets; otherwise, the element of the third set is set to true). Method insertElement inserts a

new integer k into a set (by setting a[k] to true). Method deleteElement deletes integer m (by

setting a[m] to false). Method toSetString returns a string containing a set as a list of numbers

separated by spaces. Include only those elements that are present in the set. Use — to represent an

empty set. Method isEqualTo determines whether two sets are equal. Write a program to test class

IntegerSet. Instantiate several IntegerSet objects. Test that all your methods work properly.

```
// Exercise 8.16 solution: IntegerSetTest.java
2 // Program that tests IntegerSet
3 import javax.swing.*;
45
public class IntegerSetTest {
6 private IntegerSet set1, set2, set3, set4;
7 private String output;
89
// no-argument constructor
10 public IntegerSetTest()
11 {
12 set1 = new IntegerSet();
13 set2 = new IntegerSet();
14 output = "";
15 }
16
17 public void start()
18 {
19 // initialize two sets
20 JOptionPane.showMessageDialog( null, "Input Set A" );
21 set1.inputSet();
22 JOptionPane.showMessageDialog( null, "Input Set B" );
23 set2.inputSet();
24
25 // get union and intersection of two sets
26 set3 = set1.union( set2 );
27 set4 = set1.intersection( set2 );
28
```

```
29 // prepare output
30 output += "Set A contains elements:\n" + set1.toSetString() +
31 "\n\nSet B contains elements:\n" + set2.toSetString() +
32 "\n\nUnion of Set A and Set B contains elements:\n" +
33 set3.toSetString() + "\n\nIntersection of Set A and Set B" +
34 " contains elements:\n" + set4.toSetString();
35
36 // test whether two sets are equal
37 if ( set1.isEqualTo( set2 ) )
38 output += "\n\nSet A is equal to set B\n";
39 else
40 output += "\n\nSet A is not equal to set B\n";
41
42 // test insert and delete
43 output += "\nInserting 77 into set A...\n";
44 set1.insertElement( 77 );
45 output += "Set A now contains elements:\n" + set1.toSetString();
46
47 output += "\nDeleting 77 from set A...\n";
48 set1.deleteElement( 77 );
49 output += "Set A now contains elements:\n" + set1.toSetString();
50
51 // test constructor
52 int intArray[] = { 25, 67, 2, 9, 99, 105, 45, -5, 100, 1 };
IntegerSet set5 = new IntegerSet( intArray );
54
55 output += "\n\nNew Set contains elements:\n" + set5.toSetString();
56
57 JOptionPane.showMessageDialog( null, output );
58
59 } // end method start
60
```

```
61 public static void main( String[] args )
62 {
63 IntegerSetTest test = new IntegerSetTest();
64 test.start();
65 System.exit( 0 );
66 }
67
68 } // end class IntegerSetTest
```

```
public class IntegerSet
{
    public boolean set[] = new boolean[100];

    public IntegerSet()
    {
        for(int i = 0; i < set.length; i++)
        {
            set[i] = false;
        }
    }
    public IntegerSet(int s[])
    {
        for(int i = 0; i < set.length; i++)
        {
            set[i] = false;
        }
        for(int i = 0; i < s.length; i++)
        {
            if(s[i] >= 0 && s[i] <= 99)
            {
                set[s[i]] = true;
            }
            else
            {
                System.out.printf("Incorrect set element %d passed\n", s[i]);
            }
        }
    }

    public void insertElement(int x)
    {
        if(x >= 0 && x <= 99)
        {
            set[x] = true;
        }
        else
        {
            System.out.printf("Incorrect set element %d passed to insert
method\n", x);
        }
    }

    public void deleteElement(int x)
    {
        if(x >= 0 && x <= 99)
        {
            set[x] = false;
        }
        else
        {
            System.out.printf("Incorrect set element %d passed to delete
method\n", x);
        }
    }
```

```
    public IntegerSet union(IntegerSet s)
    {
        IntegerSet union = new IntegerSet();
        for(int i = 0; i < set.length; i++)
        {
            if(set[i] == true || s.set[i] == true)
            {
                union.insertElement(i);
            }
        }
        return union;
    }

    public IntegerSet intersection(IntegerSet s)
    {
        IntegerSet intersection = new IntegerSet();
        for(int i = 0; i < set.length; i++)
        {
            if(set[i] == true && s.set[i] == true)
            {
                intersection.insertElement(i);
            }
        }
        return intersection;
    }

    public String toString()
    {
        String setAsString = "{ ";
        for(int i = 0; i < set.length; i++)
        {
            if(set[i] == true)
            {
                setAsString = setAsString + i + " ";
            }
        }
        setAsString = setAsString + "}";
        return setAsString;
    }


}
```

```
import java.util.Scanner;
public class IntegerSetTest
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the size of the First Set: ");
        int setSize = input.nextInt();
        IntegerSet setA = new IntegerSet();
        for(int i = 0; i < setSize; i++)
        {
            System.out.printf("Enter the %d element of the First Set: ",
i+1);
            int element = input.nextInt();
            setA.insertElement(element);
        }
        System.out.println();
        System.out.print("Enter the size of the Second Set: ");
        setSize = input.nextInt();
        int[] intArray = new int[setSize];
        for(int i = 0; i < setSize; i++)
        {
            System.out.printf("Enter the %d element of the First Set: ",
i+1);
            intArray[i] = input.nextInt();
        }
        IntegerSet setB = new IntegerSet(intArray);
        System.out.println();
        System.out.print("Enter Element you want to delete from Set A: ");
        int deleteEle = input.nextInt();
        setA.deleteElement(deleteEle);
        System.out.println("Set - A = " + setA);
        System.out.println("Set - B = " + setB);
        System.out.println("A Union B = " + setA.union(setB));
        System.out.println("A Intersection B = " + setA.intersection(setB));


    }
}
```