Praktikum

Collections

Yuliana Setiowati

Politeknik Elektronika Negeri Surabaya

Prak 1 : Set

1. Bagaimana output program di bawah ini ? Jelaskan !
2. Jelaskan perbedaan Class HashSet dan TreeSet !

```
import java.util.*;

 public class SetExample {
   public static void main(String args[]) {
     Set set = new HashSet();
     set.add("Bernadine");
     set.add("Elizabeth");
     set.add("Gene");
     set.add("Elizabeth");
     set.add("Clara");
     System.out.println(set);
     Set sortedSet = new TreeSet(set);
     System.out.println(sortedSet);
   }
 }
```

Prak 2 : Set

```
1  public static void main(String[] args) {
2  Set s1 = new HashSet();
3  s1.add("Australia");
4  s1.add("Sweden");
5  s1.add("Germany");
6
7  Set s2 = new HashSet();
8  s2.add("Sweden");
9  s2.add("France");
10
11 Set union = new TreeSet(s1);
12 union.addAll(s2);
13
14 print("union", union);
15
16 Set intersect = new TreeSet(s1);
17 intersect.retainAll(s2);
18
19 print("intersection", intersect);
20 }
21
22 protected static void print(String label,
23 Collection c) {
24
25 System.out.println("--------------" + labe
26 l + "--------------");
27
28 Iterator it = c.iterator();
29 while (it.hasNext()) {
30 System.out.println(it.next());
   }
   }
```

Prak 3 : Set

- Terdapat sebuah himpunan

  A = {1,2,3,4,5}

  B = {5,6,7,8,9,10}

  Menggunakan Class yang mengimplementasikan Interface Set, dapatkah output seperti :
  - A – B
  - A ∩ B
  - A U B
  - A C B

Prak 4 : Set

- Bagaimana output program dibawah ini?Jelaskan method-method yang digunakan!

```
import java.util.Enumeration;
import java.util.Iterator;
import java.util.Hashtable;
import java.util.Collection;

public class GetCollectionOfValuesFromHashtableExample {

  public static void main(String[] args) {

    //create Hashtable object
    Hashtable ht = new Hashtable();

    //add key value pairs to Hashtable
    ht.put("1","One");
    ht.put("2","Two");
    ht.put("3","Three");

    /*
      get Collection of values contained in Hashtable using
      Collection values() method of Hashtable class
    */

    Collection c = ht.values();

    System.out.println("Values of Collection created from Hashtable are :");
    //iterate through the collection
    Iterator itr = c.iterator();
    while(itr.hasNext())
      System.out.println(itr.next());

    /*
       Please note that resultant Collection object is backed by the Hashtable.
       Any value that is removed from Collection will also be removed from
       original Hashtable object. The same is not the case with the element
       addition.
    */

    //remove One from collection
    c.remove("One");

    //print values of original values of Hashtable
    System.out.println("Hashtable values after removal from Collection are :");
    Enumeration e = ht.elements();
    while(e.hasMoreElements())
      System.out.println(e.nextElement());
  }
}
```

Prak 5 : List

- Bagaimana output program di bawah ini ? Jelaskan !
- Jelaskan mengenai ArrayList dan LinkedList ? Jelaskan perbedaan dalam memasukkan data antara kedua class tersebut!

```
import java.util.*;

 public class ListExample {
   public static void main(String args[]) {
     List list = new ArrayList();
     list.add("Bernadine");
     list.add("Elizabeth");
     list.add("Gene");
     list.add("Elizabeth");
     list.add("Clara");
     System.out.println(list);
     System.out.println("2: " + list.get(2));
     System.out.println("0: " + list.get(0));
     LinkedList queue = new LinkedList();
     queue.addFirst("Bernadine");
     queue.addFirst("Elizabeth");
     queue.addFirst("Gene");
     queue.addFirst("Elizabeth");
     queue.addFirst("Clara");
     System.out.println(queue);
     queue.removeLast();
     queue.removeLast();
     System.out.println(queue);
   }
 }
```

Prak 6 : List

- Bagaimana output program dibawah ini?Jelaskan method-method yang digunakan!

```
1  // Fig. 19.3: CollectionTest.java
2  // Using the Collection interface.
3  import java.util.List;
4  import java.util.ArrayList;
5  import java.util.Collection;
6  import java.util.Iterator;
7
8  public class CollectionTest
9  {
10    private static final String[] colors =
11       { "MAGENTA", "RED", "WHITE", "BLUE", "CYAN" };
12    private static final String[] removeColors =
13       { "RED", "WHITE", "BLUE" };
14
15    // create ArrayList, add Colors to it and manipulate it
16    public CollectionTest()
17    {
18       List< String > list = new ArrayList< String >();
19       List< String > removeList = new ArrayList< String >();
20
21       // add elements in colors array to list
22       for ( String color : colors )
23          list.add( color );
24
25       // add elements in removeColors to removeList
26       for ( String color : removeColors )
27          removeList.add( color );
28
29       System.out.println( "ArrayList: " );
30
```

```
31          // output list contents
32          for ( int count = 0; count < list.size(); count++ )
33             System.out.printf( "%s ", list.get( count ) );
34
35          // remove colors contained in removeList
36          removeColors( list, removeList );
37
38          System.out.println( "\n\nArrayList after calling removeColors: " );
39
40          // output list contents
41          for ( String color : list )
42             System.out.printf( "%s ", color );
43       } // end CollectionTest constructor
44
45       // remove colors specified in collection2 from collection1
46       private void removeColors(
47          Collection< String > collection1, Collection< String > collection2 )
48       {
49          // get iterator
50          Iterator< String > iterator = collection1.iterator();
51
52          // loop while collection has items
53          while ( iterator.hasNext() )
54
55             if ( collection2.contains( iterator.next() ) )
56                iterator.remove(); // remove current Color
57       } // end method removeColors
58
59       public static void main( String args[] )
60       {
61          new CollectionTest();
62       } // end main
63    } // end class CollectionTest
```

Prak 7 : List

- Bagaimana output program dibawah ini?Jelaskan method-method yang digunakan!

```
1  // Fig. 19.4: ListTest.java
2  // Using LinkLists.
3  import java.util.List;
4  import java.util.LinkedList;
5  import java.util.ListIterator;
6
7  public class ListTest
8  {
9     private static final String colors[] = { "black", "yellow",
10       "green", "blue", "violet", "silver" };
11    private static final String colors2[] = { "gold", "white",
12       "brown", "blue", "gray", "silver" };
13
14    // set up and manipulate LinkedList objects
15    public ListTest()
16    {
17       List< String > list1 = new LinkedList< String >();
18       List< String > list2 = new LinkedList< String >();
19
20       // add elements to list link
21       for ( String color : colors )
22          list1.add( color );
23
24       // add elements to list link2
25       for ( String color : colors2 )
26          list2.add( color );
27
28       list1.addAll( list2 ); // concatenate lists
29       list2 = null; // release resources
30       printList( list1 ); // print list1 elements
31
32       convertToUppercaseStrings( list1 ); // convert to upper case string
33       printList( list1 ); // print list1 elements
34
35       System.out.print( "\nDeleting elements 4 to 6..." );
36       removeItems( list1, 4, 7 ); // remove items 4-7 from list
37       printList( list1 ); // print list1 elements
38       printReversedList( list1 ); // print list in reverse order
39    } // end ListTest constructor
40
```

```
41      // output List contents
42      public void printList( List< String > list )
43      {
44         System.out.println( "\nlist: " );
45
46         for ( String color : list )
47            System.out.printf( "%s ", color );
48
49         System.out.println();
50      } // end method printList
51
52      // locate String objects and convert to uppercase
53      private void convertToUppercaseStrings( List< String > list )
54      {
55         ListIterator< String > iterator = list.listIterator();
56
57         while ( iterator.hasNext() )
58         {
59            String color = iterator.next();  // get item
60            iterator.set( color.toUpperCase() ); // convert to upper case
61         } // end while
62      } // end method convertToUppercaseStrings
63
64      // obtain sublist and use clear method to delete sublist items
65      private void removeItems( List< String > list, int start, int end )
66      {
67         list.subList( start, end ).clear();  // remove items
68      } // end method removeItems
69
70      // print reversed list
71      private void printReversedList( List< String > list )
72      {
73         ListIterator< String > iterator = list.listIterator( list.size() );
74
75         System.out.println( "\nReversed List:" );
76
77         // print list in reverse order
78         while ( iterator.hasPrevious() )
79            System.out.printf( "%s ", iterator.previous() );
80      } // end method printReversedList
81
82      public static void main( String args[] )
83      {
84         new ListTest();
85      } // end main
86   } // end class ListTest
```

Prak 8 : List

- Buatlah program menggunakan LinkedList, sehingga output seperti berikut:

```
C:\winod\collection>javac
LinkedListExample.java

C:\winod\collection>java
LinkedListExample
Linked List Example!
Linked list data: 11 22 33 44
Linked list size: 4
Adding data at 1st location: 55
Now the list contain: 55 11 22 33
44
Now the size of list: 5
Adding data at last location: 66
Now the list contain: 55 11 22 33
44 66
Now the size of list: 6
Adding data at 3rd location: 55
Now the list contain: 55 11 99 22
33 44 66
Now the size of list: 7
```

```
First data: 55
Last data: 66
Data at 4th position: 22
Data removed from 1st location:
55
Now the list contain: 11 99 22 33
44 66
Now the size of list: 6
Data removed from last location:
66
Now the list contain: 11 99 22 33
44
Now the size of list: 5
Data removed from 2nd location:
99
Now the list contain: 11 22 33 44
Now the size of list: 4
Linked list is empty
```

Prak 9 : Map

- Bagaimana output program dibawah ini?Jelaskan method-method yang digunakan!

```
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import java.util.Set;

public class MapExample {

    public static void main(String[] args) {

        Map<Object,String> mp=new HashMap<Object, String>();

        // adding or set elements in Map by put method key and value pair
        mp.put(new Integer(2), "Two");
        mp.put(new Integer(1), "One");
        mp.put(new Integer(3), "Three");
        mp.put(new Integer(4), "Four");
```

```
        //Get Map in Set interface to get key and value
        Set s=mp.entrySet();

        //Move next key and value of Map by iterator
        Iterator it=s.iterator();

        while(it.hasNext())
        {
            // key=value separator this by Map.Entry to get key and value
            Map.Entry m =(Map.Entry)it.next();

            // getKey is used to get key of Map
            int key=(Integer)m.getKey();

            // getValue is used to get value of key in Map
            String value=(String)m.getValue();

            System.out.println("Key :"+key+"  Value :"+value);
        }
    }
}
```

Prak 10 : Map

- Bagaimana output program dibawah ini?Jelaskan method-method yang digunakan!

```
import java.util.*;

public class TreeMapExample{
public static void main(String[] args) {
System.out.println("Tree Map Example!\n");
TreeMap <Integer, String>tMap = new TreeMap<Integer, String>();
//Addding data to a tree map
tMap.put(1, "Sunday");
tMap.put(2, "Monday");
tMap.put(3, "Tuesday");
tMap.put(4, "Wednesday");
tMap.put(5, "Thursday");
tMap.put(6, "Friday");
tMap.put(7, "Saturday");
//Rerieving all keys
System.out.println("Keys of tree map: " + tMap.keySet());
//Rerieving all values
System.out.println("Values of tree map: " + tMap.values());
```

```
//Rerieving the value from key with key number 5
System.out.println("Key: 5 value: " + tMap.get(5)+ "\n");
//Rerieving the First key and its value
System.out.println("First key: " + tMap.firstKey() + " Value: "
+ tMap.get(tMap.firstKey()) + "\n");
//Rerieving the Last key and value
System.out.println("Last key: " + tMap.lastKey() + " Value: "
+ tMap.get(tMap.lastKey()) + "\n");
//Removing the first key and value
System.out.println("Removing first data: "
+ tMap.remove(tMap.firstKey()));
System.out.println("Now the tree map Keys: " + tMap.keySet());
System.out.println("Now the tree map contain: "
+ tMap.values() + "\n");
//Removing the last key and value
System.out.println("Removing last data: "
+ tMap.remove(tMap.lastKey()));
System.out.println("Now the tree map Keys: " + tMap.keySet());
System.out.println("Now the tree map contain: " + tMap.values())
}
}
```

Prak 11 : Map

- Bagaimana output program di bawah ini ? Jelaskan method-method yang digunakan !
- Jelaskan mengenai HashMap dan TreeMap! Adakah perbedaan dalam class tersebut !

```
import java.util.*;

public class MapExample {
  public static void main(String args[]) {
    Map map = new HashMap();
    Integer ONE = new Integer(1);
    for (int i=0, n=args.length; i<n; i++) {
      String key = args[i];
      Integer frequency = (Integer)map.get(key);
      if (frequency == null) {
        frequency = ONE;
      } else {
        int value = frequency.intValue();
        frequency = new Integer(value + 1);
      }
      map.put(key, frequency);
    }
    System.out.println(map);
    Map sortedMap = new TreeMap(map);
    System.out.println(sortedMap);
  }
}
```