Collection :  
ArrayList :

package day6;

import java.util.ArrayList;

public class task2 {

public static void main(String args[])

{

ArrayList l1 = new ArrayList(10);

ArrayList l2 = new ArrayList(20);

System.***out***.println(l1); // empty --> []

System.***out***.println(l2); // empty --> []

System.***out***.println(l1.size()); // size --> 0

System.***out***.println(l2.size()); // size --> 0

l1.add(1); // add to --> l1

l1.add(2); // add to --> l1

System.***out***.println(l1.size()); // size --> 2

System.***out***.println(l1); // print l1 --> [1,2]

l2.addAll(l1); // add the all element of the l1 to the l2

System.***out***.println(l2.size());// size --> 2

System.***out***.println(l2); // print l1 --> [1,2]

ArrayList l3 = new ArrayList(l2); // copy the l2 list to l3

System.***out***.println(l3); // print --> [1,2]

l3.add(l2); // add the l2 list

System.***out***.println(l3); // it print[1,2,[1,2]]

}

}

[]

[]

0

0

2

[1, 2]

2

[1, 2]

[1, 2]

[1, 2, [1, 2]]

Array list and vector uses the marker interface for random access .

package day6;

import java.util.ArrayList;

public class task3 {

public static void main(String args[])

{

ArrayList l1 = new ArrayList(10);

ArrayList l2 = new ArrayList(20);

l1.add(1);

l1.add(2);

l2.addAll(l1);

l2.add(0,"hari"); // set the index of the added element

System.***out***.println(l2);

l2.remove(2); // remove the 2 nd index of the array

System.***out***.println(l2);

l2.retainAll(l1);

System.***out***.println(l2);

}

}

[hari, 1, 2]

[hari, 1]

[1]

Vector :

package day6;

import java.util.ArrayList;

import java.util.Vector;

public class task3 {

public static void main(String args[])

{

Vector l1 = new Vector();

//ArrayList l2 = new ArrayList(20);

l1.add(1);

l1.add(2);

l1.add(4);

l1.add('g');

System.***out***.println(l1);

}

}

|  |  |
| --- | --- |
| ARRAY LIST | VECTOR |
| Every method are non synchronized | Every method are synchronized |
| Relatively performance are Performance high | Relative performance are low |
| Not a legacy class | Legacy class(old) |

List – synchronized :

package day6;

import java.util.\*;

public class task3 {

public static void main(String args[])

{

Vector l1 = new Vector();

ArrayList l2 = new ArrayList(20);

l1.add(1);

l1.add(2);

l1.add(4);

l1.add('g');

List l4 = Collections.*synchronizedList*(l2);

System.***out***.println(l1);

}

}

[1, 2, 4, g]

For each loop and ArrayList<String> 🡪 generic data .

package day6;

import java.util.ArrayList;

import java.util.Vector;

public class task4 {

public static void main(String args[])

{

//Vector l1 = new Vector();

ArrayList<String> l1 = new ArrayList<String>();

l1.add("hello");

l1.add("every");

l1.add("one");

l1.add("welcome");

for(String a:l1)

{

System.***out***.println(a);

}

hello

every

one

welcome

l1.add(53);

if we add the integer to the l1 the error will occur because the ArrayList are mention in <String> generic data

The method add(int, String) in the type ArrayList<String> is not applicable for the arguments (int)

**ITERATION :**

import java.util.ArrayList;

public class task6 {

public static void main(String args[])

{

ArrayList l1 = new ArrayList();

l1.add(10);

l1.add(30);

l1.add(30);

l1.add(40);

for(Object a : l1)

{

System.***out***.println(l1);

}

[10, 30, 30, 40]

[10, 30, 30, 40]

[10, 30, 30, 40]

[10, 30, 30, 40]

LinkedList :

package day6;

import java.util.LinkedList;

public class task5 {

public static void main(String args[])

{

LinkedList l1 = new LinkedList();

l1.add("hari");

l1.add("hello");

l1.add("every");

l1.add("one");

l1.add("welcome");

System.***out***.println(l1);

System.***out***.println(l1.getFirst());

System.***out***.println(l1.getLast());

System.***out***.println(l1.removeFirst());

System.***out***.println(l1.removeLast());

//l1.addFirst("first");

System.***out***.println(l1);

l1.set(1,"1 index" ); // replace the 1st index value but add only append the value

System.***out***.println(l1);

l1.add(3," 0 th index");

System.***out***.println(l1);

l1.set(0,"1 index" );

System.***out***.println(l1);

hari

welcome

[hello, every, one]

[hello, 1 index, one]

[hello, 1 index, one, 0 th index]

[1 index, 1 index, one, 0 th index]

HashSet

import java.util.HashSet;

public class task7 {

public static void main(String args[])

{

HashSet l1 = new HashSet();

l1.add("hari");

l1.add("hello");

l1.add("every");

l1.add("one");

l1.add("welcome");

l1.add("welcome"); // if the duplication is added to the hashSet it simply ignore

System.***out***.println(l1); // if print the set it can not maintain the insertion order

}

}

[hari, one, hello, every, welcome]

LINKED HASH SET

package day6;

import java.util.HashSet;

import java.util.LinkedHashSet;

public class task7 {

public static void main(String args[])

{

LinkedHashSet l1 = new LinkedHashSet();

l1.add("hari");

l1.add("hello");

l1.add("every");

l1.add("one");

l1.add("welcome");

l1.add("welcome"); // if the duplication is added to the hashSet it simply ignore

System.***out***.println(l1); // if print the set it maintain the insertion order

[hari, hello, every, one, welcome]

**SORTEDSET** :

Tree Set :

package day6;

import java.util.HashSet;

import java.util.LinkedHashSet;

import java.util.TreeSet;

public class task8 {

public static void main(String args[])

{

TreeSet l1 = new TreeSet();

l1.add("hari");

l1.add("hello");

l1.add("every");

l1.add("one");

l1.add("welcome");

l1.add("welcome"); // if the duplication is added to the hashSet it simply ignore

System.***out***.println(l1); // if print the set it can not main the insertion order

}

}

maintain the order

Insertion of tree set :

* Insert k
* Insert I 🡪 compare 🡪 -ve so this is left side
* Insert n 🡪 compare 🡪 +ve so right side
* Insert g 🡪 compare 🡪 -ve so lest side of the I
* 