05-07-2024

Sub-interface

COLLECTION FRAMEWORK OR COLLECTIONS =>

List -> Arraylist, vector, stack, linkedlist

Queue -> Abstract Queue (abstract class)

Priority Queue

Set -> Abstract Set (abstract class)

Hash set

↓

sorted set -> Tree set, linkedHashset

Map -> Hash Table, Hash Map

↓

Sorted Map -> Tree Map, linked Hash Map

Iterator

↓

listiterator

LIST:- It is a linear Data - structure elements are stored in preserved order.

Duplicates are allowed.

-> package sreeja;   List. java

```
import java.util.Arraylist;
public class Demo
{
  public static void main (String ar[])
  {
    Arraylist<string> a = new Arraylist<string>();
    a.add("hai"); a.add("hello"); a.add("chello");
    System.out.println(a);
     a.add(1,"bye");
     a.remove(2);
    System.out.println(a);
  }
}
```

O/p:- [hai, hello, chello]
2
  [hai, bye, chello]

VECTOR :- It is a legacy class.

Vectorr . java

Same as above code

```
public stati
{
   Vector < String > a = new Vector < String > ( );
 }
}
```

STACK :- LIFO order Last In First out .

It is a sub class of Vector ( ) . It operates on the principle of LIFO.

peek ( ) -> TOP most element.

stackk . java

```
-> package sreeja;
import java . util . stack;  [illegible]
public class Demo
 {
   public static void main ( String ar [ ] )
   {
     stack < String > a = new stack < String > ( );
     a . push ( " hello " ); a . push ( " hai " ); a . push ( " bye " );
     System . out . println ( a );
     System . out . println ( a . peek ( ) );
     System . out . println ( a . pop ( ) );
     System . out . println ( a . isEmpty ( ) );
   }
}
```

O/P :-

```
[hello, hai, bye]
bye
bye
false
```

peek ( ) ;- last element   pop ( ) :- LIFO

LINKEDLIST :-

Linked_listt.java

```
→ package sreeja;
  import java.util.ArrayList;
  public class Demo
  {
     public static void main (String ar[])
     {
        LinkedList <String> l = new LinkedList<String>();
        l.add ("hai"); l.add ("bye"); l.add(1, "challo");
        l.addFirst ("siit"); l.addLast (" satvick");
        System.out.println (l);
     }
  }
```

QUEUE :- It is a linear data structure which follows the principle of FIFO order.   FIFO → First in First out

Queue.java

```
→ package sreeja;
  import java.util.ArrayList;
  public class Demo
  { public static void main (String ar[])
    {
      PriorityQueue <String> P = new PriorityQueue<String>();
      P.add("10"); P.add ("20"); P.add ("5");
      System.out.println (P);
      System.out.println (P.poll());
      System.out.println (P.poll());
  } }
```

```
o/p:-[10, 20,5]
      10
      20
```