Java Programming For Web Applications. 22

CSA·0985 , Assingment·03.

12/08/24

Name : S. Anitha.

Reg No : 192324040.

Dept : B.Tech (AI & DS)

Date: 12/08/24.

Day: Monday.

10/20

Collection and objects:

Single unit of object. collection from work provides many interfaces and classes.

List

Array List

Linked List.

List:

```
Public class main
{
  Public static void main (string[] ,args)
   {
        obj. add ("one");
        obj. add ("Two");
        obj. add ("Three");
        obj. add (1000);
        obj. add (1000);
      System. out. println (" Arrays list:"+ obj);
   }
}
```

Arraylist:

```
        import java.util.list
        class main
        {
         Public static void main (string[ ] ,args)
          {
           List < integer> number = new arraylist <>( );
                  number. add (1);
                  number. add (2);
                  number.add (3);
            system. out. println ("List' + number);
          int get number = number.get (2);
         system. out. println("element at index 2:" + get number);
           numbers. remove(1);
         system. out . println ("List after removal" + number);
          numbers set (1,4);
        system. out. println ("List after update:" + number);
        system. out. println (" Iterating through the list:");
    for (int  number : number)
        {
          system.out. println (numbers + " ");
         }
          system. out. println( ),
        }
    }
```

```
List = [1,2,3]
elements at index 2:3
List after removal :[1,3]
List after update : [1,4]
Iterating through the list : 1 4
```

Linked list:

```
import java.util.list;
import java.util.linkedlist;
class main
{
  Public static void main (string[] args)
  {
   List < string > numbers = new linked list <>();
     numbers.add ("Apple");
     numbers.add ("orange");
     numbers.add ("mango");
 string number = number.get (2),
 System.out.println ("Allowed element" + numbers);
 int index = numbers.index + ("Apple");
 system..out.println ("pos of 2 is" + index);
   numbers.set (2, "banana");
 system.out.println ("updated list:" + number);
numbers. remove ("orange");
system. out. println ("final list");
    for (string fruit : number);
      {
          system. out. println (fruits);
      }
    }
  }
```

Output:

```
Accessed element : Mango
pos of 'apple' is :0
updated list : [apple, orange, banana].
final list : apple, banana, grape pineapple.
```

Vector:

```
Import . java . util. Iterator
import java. util . vector
class main
  {
    Public static void main (string[] args)
    {
        vector < string > fruits = new vector <>();
         fruits. add ("Apple");
         fruits. add ("orange");
         fruits. add ("mango");
```

```
system. out. println ("vector1" +fruits);
   string element : fruitget (2);
 system. out. println ("element at index 2:" element);
   fruits.add [index & element, "banana");
 system.out. printh ("vector" ; "fruits")
  vector < string : Indianfruits : new vector <>();
    Indianfruit .add All (fruits);
  system.out. println ("vector"); + Indianfruits);
   Iterate < string > Iterate : indianfruits. Iterators();
     system. out. println ("vector");
   Iterate < string > iterate : indianfruits. iterate();
     while (iterate. hasnext ());
   {
     system. out. println (iterate. next ());
       system.out. println (",");
```

Sort and reverse:

```
    import java.util. arrays
    import java. util. collections

   class Main
   {
     Public static void main (string[] ,args)
     {
       first < string : fruits = new linked list <>() ;
             fruits. add ("Apple");
             fruits. add ("orange");
             fruits.add ("Mango");
             fruits.add ("Grape");
     system. out .println ("ori list" + fruits);
     collection .sort (fruits);
    system.out. println ("Rev list" + fruits);
    collection. sort (fruits);
   system . out . println (fruits. collection . reverseorder());
    collection. sort (fruits. collection .reverseorder());
   system. out. println ("sort in des order" + fruits);
   system. out. println ("fruits in the basket");
   for (int i=0 ; i< fruits . size(); i++)
     {
        system. out. println (fruits. get(i));
     }
```

stack
Queue
dequeue.

Stack :

```
import java. util. stack;
Public class fruitstack
{
    Public static void main (string[] ,args)
    {
        stack < string > fruitstack = new stack <>();
        fruit stack. push ("Apple");
        fruitstack. push ("Banana");
        fruitstack. push ("cherry");
        system. out. println ("stack:")
        while (! fruitstack. isempty())
        {
          system. out. println (fruitstack. pop());
        }
    }
}
```

```
Stack: Cherry
       Banana
       Apple.
```

```
system. out. println (" fruits in the basket (in reverse order)
    for (int i= fruits. size( )-1 ; i>=0; i++)
    {
        system. out. println ( fruits. get (i));
    }
  }
}
```

Output:

```
    ori list: [apple, orange , mango, grape].
   sort list: [apple, orange, mango, grape].
    Rev list: [orange, mango, grape, apple].
sort in asc order: [apple, grape, mango, orange].
sort in des order: [orange, mango, grape, apple].

fruits in the basket → orange
                       mango
                       Grape
                       Apple .
```

Dequeue:

```
import java.util.Arraydequeue;
import java.util.dequeue;
Public class fruits dequeue
 {
   Public static void main (string[] args)
   {
     dequeue<string> fruit dequeue = new arraydequeue<>();
       fruit dequeue.add fruit ("Mango");
       fruit dequeue.add last ("Peach");
       fruit dequeue.add first ("Kiwi");
     system.out.println ("dequeue:");
     while (! fruitdequeue.is empty())
     {
       System.out.println (fruit dequeue.poll first())
     }
   }
 }
```

Output:

```
dequeue : Kiwi
          Mango
          Peach.
```

It is an interface include methods of.

Collections interface.

```
import java. util. map;
Import java. util. Hashmap;
class main {
        Public static void (string[] args)
      {
       map: Integer, string > fruits = new map<>();
       fruits = new Hashmap();
        fruits. put (1, "Apple")
        fruits. put (2,"orange").
       system. out. println ("fruits". Emptyset ().
         boolean vale = (fruits remove (2," orange").
       system. out. println (" Avail in basket": value)
```