Assignment - 3

21

Java Programming For Web Application.

CSA-0985

Name : S. Monesh.
Reg No : 192324040.
Dept : B-Tech (AI & DS).

Date : 12/08/24.
Day : Monday.

12/08/24

10/10

Collection & 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 (10000);
    system. out. println ("Arrays list:"+ obj);
    }
}
```

Array List:

```
        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  getnumber = 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. printh ("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 ("vector :" + fruits);
  string element = fruitget (2);
 system . out. println ("elem at index 2:" element;
  fruits . add [ index of element, "banana");
 system. add . println ("vector " ; "Monesh"
vector<string = ~~element~~ Indianfruits = new vector<>();
  Indianfruit . add All (fruits );
  system . out. println ("vector"); + Indianfruit);

  Iterate < string > iterate = indianfruits . iterator ( );
  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 ("New list" + fruits);
 collection . sort (fruits);
 system. out. sort (fruits. collection. reverse order ());
 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));
  }
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.
```

→ stack .

→ Queue .

→ dequeue .

stack :

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

```
stack :   cherry .
          banana.
          Apple.
```

Queue :

```
import java . util . linkedlist ;
import java . util . Queue;
public class fruitQueue
   {
   public static void main (string [] args)
    {
      Queue < string > fruitqueue = new linkedlist <>();
       fruitqueue . add ("orange");
       fruitqueue . add ("pineapple");
       fruitqueue . add ("Grapes");
       system . out . println ("Queue");
       while (! fruitqueue . isempty ())
        {
         system . out . println (fruitqueue . poll ());
        }
     }
   }

   Queue  →  orange.
             Pineapple.
             Grapes.
```

Dequeue :

```
        import   java. util. Arraydequeue;
        import   java. util. dequeue;
        public  class  fruit dequeue
          {
            public static  void  main (string[] args)
             {
              dequeue < string > fruitdequeue = new  arradequeue <>();
                fruit dequeue .add first ("Mango");
                fruit  dequeue . addlast ("Peach");
                fruit dequeue . addfirst ("kiwi");
              system . out . println ("dequeue");
              while ( ! fruitdequeue  . is empty())
                {
                system. out. println( fruit dequeue. poll first ());
                }
             }
          }
```

Output:

```
        dequeue  :  kiwi.
                    Mango.
                    Peach .
```

Map interface:

It is an interface include methods of collection interface.

key, value.

```
import java. util .map;
import . java . util . Hashmap
class main
  {
   public static void main ( string [] . args)
     {
      map< Integer ~~string~~ >string = new map<>();
      fruits = new hashmap();
      fruits . put (1, " Apple");
      fruits . put (1, "orange");
      fruits . put (3, " Mango");
    system .out. println ("Map' + fruits );
    system. out. println ( "keys:" + fruits . Keyset ());
    system . out . println ("values" : + fruit.values ());
    system . out . println ("fruits :" + fruits . emptyset ());
    boolean value = (fruits . remove (2, "orange"));
    system . out. println (" Rem value :" + value);
    system . out . println ( " new map' + fruits);
boolean value 1 = fruits. containskey(3);
system. out. println ("Avail in the basket:" +value);
}
}
```

Priority Queue:

```
import java. util. priority Queue;
import java. util. queue;
public class fruit priorityqueue
{ public static void main (string[], args)
  {
   Queue < string > fruitpriorityqueue = new priorityqueue<>();
   fruit priority queue. add ("strawberry");
   fruitpriority queue. add ("Blueberry");
   fruitpriority queue. add ("Raspberry");
   fruitpriority queue. add ("Apple");
  system. out. println ("Priority queue:");
  while (! fruitpriority queue. isempty());
   {
     system. out. println (fruitpriorityqueue. poll());
   }
  }
}
```

12/07/24

Priority queue: Apple, Blueberry, Raspberry, strawberry