Dt : 17/10/2023

(a)HashMap<K,V>:

=>HashMap<K,V> organizes elements without any order and which is NonSynchronized class.

(b)LinkedHashMap<K,V>:

=>LinkedHashMap<K,V> organizes elements with insertion order and which is also NonSynchronized class.

(c)TreeMap<K,V>:

=>TreeMap<K,V> organizes elements automatically in Ascending order based key-date(based on Primary-key)

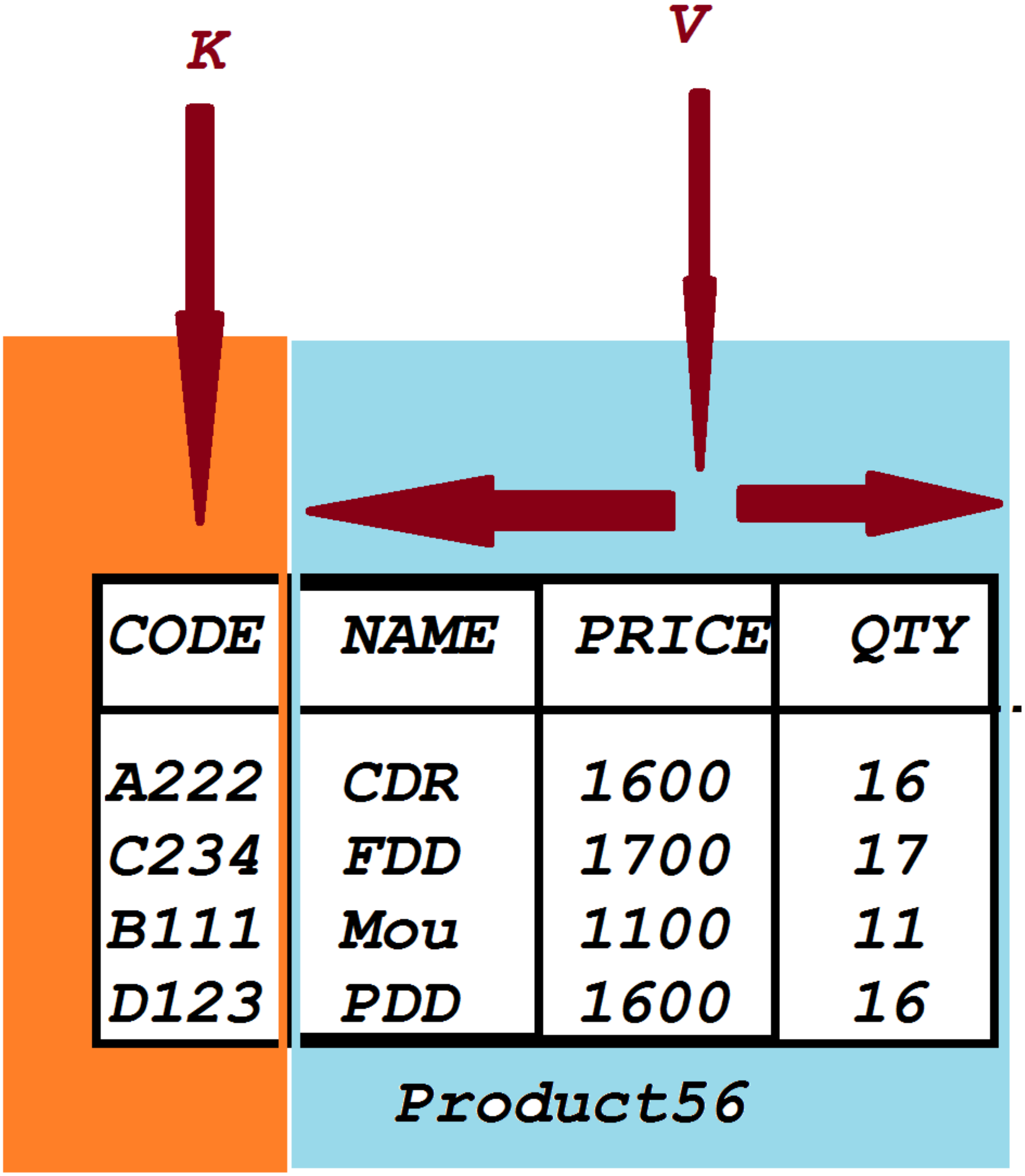
(d)Hashtable<K,V>:

=>Hashtable<K,V> organizes elements without any order,but which is synchronized class.

DB Table : Product56

CODE	NAME	PRICE	QTY
A222	CDR	1600	16

C234	*FDD*	*1700*	*17*
B111	*Mou*	*1100*	*11*
D123	*PDD*	*1600*	*16*



CODE	NAME	PRICE	QTY
A222	CDR	1600	16
C234	FDD	1700	17
B111	Mou	1100	11
D123	PDD	1600	16

Note:

=>In the processing of organizing database table data,we must construct

User-defined class declared with variables equal to Database table Values.

p1 : ProductValues.java

```
package p1;
public class ProductValues extends Object
{
  public String name;
  public float price;
  public int qty;
  public ProductValues(String name,float price,int qty)
{
      this.name=name;
      this.price=price;
      this.qty=qty;
  }
  @Override
  public String toString() {
      return name+"\t"+price+"\t"+qty;
  }
}
```

p2 : DemoMap1.java(MainClass)

```
package p2;

import java.util.*;

import p1.*;

public class DemoMap1 {

    public static void main(String[] args) {

  Scanner s = new Scanner(System.in);

  try(s;){
```

```
try {
    Map<String,ProductValues> ob=null;
    String nm=null;
    while(true) {
        System.out.println("****Choice****");
        System.out.println("\t1.HashMap"
                + "\n\t2.LinkedHashMap"
                + "\n\t3.TreeMap"
                + "\n\t4.Hashtable"
                + "\n\t5.Exit");
        System.out.println("Enter the Choice:");
        switch(Integer.parseInt(s.nextLine())) {
        case 1:
            ob = new HashMap<String,ProductValues>();
            nm="HashMap";
            break;
        case 2:
            ob = new LinkedHashMap<String,ProductValues>();
            nm="LinkedHashMap";
            break;
        case 3:
            ob = new TreeMap<String,ProductValues>();
        nm="TreeMap";
        break;
case 4:
        ob = new Hashtable<String,ProductValues>();
        nm="Hashtable";
        break;
case 5:
        System.out.println("Operation Stopped..");
        System.exit(0);
default:
        System.out.println("Invalid Choice...");
        continue;
}//end of switch
System.out.println("---date from "+nm+"----");
ob.put(new String("A222"),
                new ProductValues("CDR",1600,16));
ob.put(new String("C234 "),
                new ProductValues("FDD",1700,17));
ob.put(new String("B111"),
                new ProductValues("Mou",1100,11));
ob.put(new String("D123"),
                new ProductValues("PDD",1600,16));
```

```
ob.forEach((p,q)->
{
        System.out.println(p.toString()+"\t"+q.toString());
});
System.out.println("Perform Operations on "+nm);
xyz:while(true) {
        System.out.println("-----Choice-----");
System.out.println("\t1.put(K,V)"
              + "\n\t2.remove(Object)"
              + "\n\t3.get(Object)"
              + "\n\t4.keySet()"
              + "\n\t5.values()"
              + "\n\t6.Exit");
System.out.println("Enter the Choice:");
switch(Integer.parseInt(s.nextLine())) {
case 1:
        System.out.println("***Add Product Details***");
        System.out.println("Enter the ProdCode:");
        String pC = s.nextLine();
        System.out.println("Enter the ProdName:");
        String pN = s.nextLine();
        System.out.println("Enter the ProdPrice:");
```

```
                float pP = Float.parseFloat(s.nextLine());
                System.out.println("Enter the ProdQty:");
                int pQ = Integer.parseInt(s.nextLine());
                ob.put(new String(pC),
                        new ProductValues(pN,pP,pQ));
                System.out.println("Product Details added to "+nm+"
Successfully...");
                System.out.println("----Details----");
                ob.forEach((p,q)->
                {

System.out.println(p.toString()+"\t"+q.toString());
                });
                break;
            case 2:
                System.out.println("Enter the Key to remove Product
Details...");
                String pC2 = s.nextLine();
                if(ob.containsKey(pC2)) {
                    ob.remove(pC2);
                    System.out.println("Product details removed
Successfully...");
                    System.out.println("----Details----");
                    ob.forEach((p,q)->
                {

System.out.println(p.toString()+"\t"+q.toString());
                });
                }else {
                    System.out.println("Invalid ProdCode...");
                }
                break;
            case 3:
                System.out.println("Enter the Key to display Product
deatils...");
                String pC3 = s.nextLine();
                if(ob.containsKey(pC3)) {
                    ProductValues pv = ob.get(pC3);
                    System.out.println(pv.toString());
                }else {
                    System.out.println("Invalid ProdCode....");
                }
                break;
            case 4:
                if(ob.isEmpty()) {
```

```
            System.out.println("Map is empty...");
        }else {
            Set<String> ks = ob.keySet();
            System.out.println("====Key-data====");
            ks.forEach((a)->
            {
                System.out.println(a.toString());
            });
        }
        break;
    case 5:
        if(ob.isEmpty()) {
            System.out.println("Map is empty...");
        }else {
            Collection<ProductValues> pv = ob.values();
            System.out.println("=====Values-data====");
            pv.forEach((b)->
            {
                System.out.println(b.toString());
            });
        }
        break;
```

```
                case 6:
                    System.out.println("Operations Stopped ob "+nm);
                    break xyz;//Stopping Inner Loop
                default:
                    System.out.println("Invalid Choice...");
                }//end of switch
                }//end of Inner loop
            }//end of Outer loop
        }catch(Exception e) {e.printStackTrace();}
  }//end of try with resource
    }
}
```

o/p:

```
****Choice****
    1.HashMap
    2.LinkedHashMap
    3.TreeMap
    4.Hashtable
    5.Exit
Enter the Choice:
2
---date from LinkedHashMap----
```

```
A222  CDR   1600.0        16
C234  FDD   1700.0        17
B111  Mou   1100.0        11
D123  PDD   1600.0        16
Perform Operations on LinkedHashMap
-----Choice-----
          1.put(K,V)
          2.remove(Object)
          3.get(Object)
          4.keySet()
          5.values()
          6.Exit
Enter the Choice:
5
=====Values-data====
CDR   1600.0        16
FDD   1700.0        17
Mou   1100.0        11
PDD   1600.0        16
-----Choice-----
          1.put(K,V)
          2.remove(Object)
```

```
        3.get(Object)

        4.keySet()

        5.values()

        6.Exit

Enter the Choice:

6

Operations Stopped ob LinkedHashMap

****Choice****

        1.HashMap

        2.LinkedHashMap

        3.TreeMap

        4.Hashtable

        5.Exit

Enter the Choice:

8

Invalid Choice...

****Choice****

        1.HashMap

        2.LinkedHashMap

        3.TreeMap

        4.Hashtable

        5.Exit
```

```
Enter the Choice:
3
---date from TreeMap----
A222  CDR   1600.0        16
B111  Mou   1100.0        11
C234  FDD   1700.0        17
D123  PDD   1600.0        16
Perform Operations on TreeMap
-----Choice-----
        1.put(K,V)
        2.remove(Object)
        3.get(Object)
        4.keySet()
        5.values()
        6.Exit
Enter the Choice:
4
====Key-data====
A222
B111
C234
D123
```

```
-----Choice-----

        1.put(K,V)

        2.remove(Object)

        3.get(Object)

        4.keySet()

        5.values()

        6.Exit

Enter the Choice:

6

Operations Stopped ob TreeMap

****Choice****

        1.HashMap

        2.LinkedHashMap

        3.TreeMap

        4.Hashtable

        5.Exit

Enter the Choice:

5

Operation Stopped..

================================================================
```