

Dt : 19/10/2023

define Iterable<E>?

=>Iterable<E> is an interface from java.lang package and which is Parent Interface of Collection<E>.

=>This Iterable<E> interface will provide the following methods to perform iteration process:

public abstract java.util.Iterator<T> iterator();

public default void forEach(java.util.function.Consumer<? super T>);

public default java.util.Spliterator<T> spliterator();

iterator() : method is used to create implementation object for Iterator<E> interface and the object will hold the reference of Collection<E> Object.

forEach() : forEach() method is used to retrieve elements directly from Collection<E> and Map<k,V> objects.

spliterator<T> : method is used to create implementation object for Spliterator<T> interface the object will hold the reference of Array-Object or Collection<E>-Object

=====

faq:

define Cursor Statements?

=>The statements which are used to retrieve elements from Collection<E>

Objects are known as Cursor Statements.

=>The following are some important Cursor statements used on Collection<E>

Objects:

(a)Iterator<E> - Used on Collection<E>-Objects

(b)ListIterator<E> - Used only on List<E>-Objects

(c)Enumeration<E> - Used on Vector<E>-Objects

(d)Spliterator<T> - Used in Array-Objects and Collection<E>-Objects

=====

faq:

define BiPredicate<T,U>?

=>BiPredicate<T,U> is a Functional Interface from java.util.function package introduced by Java8 version and which is used to perform conditional operation on Map<K,V> Objects.

Structure of BiPredicate<T,U>:

```
public interface java.util.function.BiPredicate<T, U>  
{  
  
    public abstract boolean test(T, U);  
  
    ...  
  
}
```

syntax of LambdaExpression:

BiPredicate<T,U> bp = (T,U)->

```
{  
...  
};
```

Ex-program :

(Display Product-details which are having price<=1000 and qty<=10)

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 : DemoMap3.java(MainClass)

package p2;

import p1.*;

import java.util.*;

import java.util.function.*;

public class DemoMap3

{

public static void main(String[] args)

{

LinkedHashMap<String,ProductValues> ob =

new LinkedHashMap<String,ProductValues>();

ob.put(new String("A222"),

new ProductValues("CDR",1600,16));

ob.put(new String("C234 "),

new ProductValues("FDD",700,17));

ob.put(new String("B111"),

new ProductValues("Mou",1100,1));

ob.put(new String("D123"),

new ProductValues("PDD",600,8));

System.out.println("*Data from Map****");***

ob.forEach((p,q)->

{

```
        System.out.println(p.toString()+"\t"+q.toString());
    });
```

```
//Predicate for Product-Price
```

```
BiPredicate<String,ProductValues> bp1 = (x,y)->
```

```
{
    if(y.price<=1000) return true;
    else return false;
};
```

```
BiPredicate<String,ProductValues> bp2 = (x,y)->
```

```
{
    if(y.qty<=10) return true;
    else return false;
};
```

```
System.out.println("****Products price<=1000****");
```

```
ob.forEach((p,q)->
```

```
{
    if(bp1.test(p, q))
    {
        System.out.println(p.toString()+"\t"+q.toString());
    }
}
```

```

    });

    System.out.println("****Products qty<=10****");

    ob.forEach((p,q)->
    {

        if(bp2.test(p, q))

        {

            System.out.println(p.toString()+"\t"+q.toString());

        }

    });

}

```

o/p:

******Data from Map******

A222 CDR 1600.0 16

C234 FDD 700.0 17

B111 Mou 1100.0 1

D123 PDD 600.0 8

******Products price<=1000******

C234 FDD 700.0 17

D123 PDD 600.0 8

******Products qty<=10******

B111 Mou 1100.0 1

D123 PDD 600.0 8

=====

Venkatesh Maipathii