MAP :

|package day8;

import java.util.HashMap;

public class task1 {

public static void main(String args[])

{

HashMap h = new HashMap();

h.put("tea", 15);

h.put("coffee", 30);

h.put("dosa" , 50);

h.put("idli",60);

System.***out***.println(h);

h.put("birr", 15);

System.***out***.println(h.put("bir", 15)); // the key value is not exist so it return the null

System.***out***.println(h);

h.put("tea", 99);

System.***out***.println(h);

}

}

{tea=15, dosa=50, coffee=30, idli=60}

null

{tea=15, dosa=50, coffee=30, idli=60, birr=15, bir=15}

{tea=99, dosa=50, coffee=30, idli=60, birr=15, bir=15}

Keyset() and values() method :

package day8;

import java.util.Collection;

import java.util.HashMap;

import java.util.Set;

public class task2 {

public static void main(String args[])

{

HashMap h = new HashMap();

h.put("tea", 15);

h.put("coffee", 30);

h.put("dosa" , 50);

h.put("idli",60);

h.put("biryani", 200);

Set keySet = h.keySet(); // key only in key set

System.***out***.println(keySet);

Collection c = h.values(); // use the collection class for value retrival because if using the

System.***out***.println(c); // set the the duplicate keys are not allowed

}

}

[biryani, tea, dosa, coffee, idli]

[200, 15, 50, 30, 60]

entrySet() method

public class task2 {

public static void main(String args[])

{

HashMap h = new HashMap();

h.put("tea", 15);

h.put("coffee", 30);

h.put("dosa" , 50);

h.put("idli",60);

h.put("biryani", 200);

Set entry = h.entrySet(); // get the key=value pair in the form of set

System.***out***.println(entry);

}

[biryani=200, tea=15, dosa=50, coffee=30, idli=60]

Iterator :

package day8;

import java.util.Collection;

import java.util.HashMap;

import java.util.Iterator;

import java.util.Set;

public class task2 {

public static void main(String args[])

{

HashMap h = new HashMap();

h.put("tea", 15);

h.put("coffee", 30);

h.put("dosa" , 50);

h.put("idli",60);

h.put("biryani", 200);

Set entry = h.entrySet(); // get the key=value pair in the form of set

System.***out***.println(entry);

Iterator it = entry.iterator();

while(it.hasNext())

{

System.***out***.println(it.next());

}

}

}

biryani=200

tea=15

dosa=50

coffee=30

idli=60

in this program to change the value of the entire map element :

package day8;

import java.util.Collection;

import java.util.HashMap;

import java.util.Iterator;

import java.util.Map;

import java.util.Set;

public class task3 {

public static void main(String args[])

{

HashMap h = new HashMap();

h.put("tea", 15);

h.put("coffee", 30);

h.put("dosa" , 50);

h.put("idli",60);

h.put("biryani", 200);

Set entry = h.entrySet();

System.***out***.println(entry);

Iterator it = entry.iterator();

while(it.hasNext())

{

Map.Entry mentry = (Map.Entry)it.next();

if(mentry.getKey().equals("tea"))

{

mentry.setValue(20);

}

if(mentry.getValue().equals(200))

{

mentry.setValue(250);

}

//System.out.println(mentry);

Integer val = (Integer)mentry.getValue();

mentry.setValue(val+25);

System.***out***.println(mentry);

}

System.***out***.println(h);

}

}

[biryani=200, tea=15, dosa=50, coffee=30, idli=60]

biryani=275

tea=45

dosa=75

coffee=55

idli=85

{biryani=275, tea=45, dosa=75, coffee=55, idli=85}

LINKED HASH MAP:

import java.util.HashMap;

import java.util.Iterator;

import java.util.LinkedHashMap;

import java.util.Map;

import java.util.Set;

public class task5 {

public static void main(String args[])

{

HashMap h = new HashMap();

h.put("tea", 15);

h.put("coffee", 30);

h.put("dosa" , 50);

h.put("idli",60); // does not maintain the insertion order

h.put("biryani", 200);

Set entry = h.entrySet();

System.***out***.println(entry);

Iterator it = entry.iterator();

LinkedHashMap l = new LinkedHashMap();

l.put("tea", 15);

l.put("coffee", 30); // its maintain the insertion order

l.put("dosa" , 50);

l.put("idli",60);

l.put("biryani", 200);

System.***out***.println(l);

}

}

[biryani=200, tea=15, dosa=50, coffee=30, idli=60]

{tea=15, coffee=30, dosa=50, idli=60, biryani=200}

TREE MAP :

import java.util.TreeMap;

public class task5 {

public static void main(String args[])

{

TreeMap t = new TreeMap();

t.put(1,"tea");

t.put(2,"coffee");

t.put(4,"dosa" );

t.put(3,"idli");

System.***out***.println(t);

}

}

{1=tea, 2=coffee, 3=idli, 4=dosa}

The treeMap always compare the key and sort it . If the next element is string key it raise the error

t.put("6",1);

System.***out***.println(t);

Exception in thread "main" java.lang.ClassCastException:

Comparator for customization sort

package day8;

import java.util.Comparator;

import java.util.TreeMap;

class MyComparator implements Comparator{

public int compare(Object ob1 , Object ob2)

{

String s1 = (String)ob1;

String s2 = (String)ob2;

return s2.compareTo(s1);

}

}

public class task5 {

public static void main(String args[])

{

TreeMap t = new TreeMap(new MyComparator());

t.put("tea",1);

t.put("coffee",2);

t.put("dosa",3 );

t.put("idli",4);

System.***out***.println(t);

//System.out.println();

{tea=1, idli=4, dosa=3, coffee=2}

REGEX :

package day8;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class task6 {

public static void main(String args[])

{

int c=0;

Pattern p = Pattern.*compile*("aab");

Matcher m = p.matcher("aababbaababab");

while(m.find())

{

c++;

System.***out***.println(m.start()+ "--> " + m.end()+"-->"+m.group());

}

System.***out***.println(c);

}

}

0--> 3-->aab

6--> 9-->aab

2

package day8;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class task6 {

public static void main(String args[])

{

int c=0;

Pattern p = Pattern.*compile*("[^a-z A-Z 0-9]");

Matcher m = p.matcher("a\*z@123.com");

while(m.find())

{

c++;

System.***out***.println(m.start()+ "--> " + m.end()+"-->"+m.group());

}

System.***out***.println(c);

}

}

1--> 2-->\*

3--> 4-->@

7--> 8-->.

3

PRE DEFINE CHARACTER

🡪\s 🡪\s 🡪 \d 🡪\D 🡪 \w 🡪\W 🡪 . (it print the any character)

package day8;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class task6 {

public static void main(String args[])

{

int c=0;

Pattern p = Pattern.*compile*("\s");

Matcher m = p.matcher("java training session");

while(m.find())

{

c++;

System.***out***.println(m.start()+ "--> " + m.end()+"-->"+m.group());

}

System.***out***.println(c);

}

}

4--> 5--> (space)

13--> 14--> (space)

2

QUANTIFIER :

Example : abababaaab

|  |  |  |  |
| --- | --- | --- | --- |
| a | a+ | a\* | a? |
| 0--> 1-->a  2--> 3-->a  4--> 5-->a  6--> 7-->a  7--> 8-->a  8--> 9-->a  6 | 0--> 1-->a  2--> 3-->a  4--> 5-->a  6--> 9-->aaa  4 | 0--> 1-->a  1--> 1-->  2--> 3-->a  3--> 3-->  4--> 5-->a  5--> 5-->  6--> 9-->aaa  9--> 9-->  10--> 10-->  9 | 0--> 1-->a  1--> 1-->  2--> 3-->a  3--> 3-->  4--> 5-->a  5--> 5-->  6--> 7-->a  7--> 8-->a  8--> 9-->a  9--> 9-->  10--> 10-->  11 |

SPLIT():

Split based on digit :

package day8;

import java.util.regex.Pattern;

public class task7 {

public static void main(String args[])

{

Pattern p = Pattern.*compile*("\\d");

String[] s = p.split("this5is3java6programming");

for(String str:s)

{

System.***out***.println(str);

}}

}

this

is

java

programming

split based on space :

package day8;

import java.util.regex.Pattern;

public class task7 {

public static void main(String args[])

{

Pattern p = Pattern.*compile*("\s");

String[] s = p.split("this is java programming");

for(String str:s)

{

System.***out***.println(str);

}}

}

this

is

java

programming