```java
package cn.sxt.collection;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.HashSet;
import java.util.Iterator;
import java.util.List;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Set;
/**
 * 测试迭代器遍历List，Set，Map
 * @author 江
 *
 */
public class TestIterator {
    public static void main(String[] args) {
        testIteratorList();
        testIteratorSet();
        testIteratorMap();
    }

    public static void testIteratorList() {
        List<String> list=new ArrayList<>();
        list.add("aa");
        list.add("bb");
        System.out.println(list);

        //使用iterator遍历list
        for(Iterator<String> iter=list.iterator();iter.hasNext();) {
```

```java
                    String temp=iter.next();
                    System.out.println(temp);
            }
    }


    public static void testIteratorSet() {
            Set<String> set=new HashSet<>();
            set.add("aa");
            set.add("bb");
            System.out.println(set);

            //使用iterator遍历set
            for(Iterator<String>
iter=set.iterator();iter.hasNext();) {
                        String temp=iter.next();
                        System.out.println(temp);
            }
    }


    public static void testIteratorMap() {
            Map<Integer,String> map=new HashMap<>();
            map.put(100,"张三");
            map.put(200,"李四");
            map.put(300,"王五");

            //第一种遍历Map的方式
            Set<Entry<Integer,String>> set=map.entrySet();
            for(Iterator<Entry<Integer,String>>
iter=set.iterator();iter.hasNext();) {
                        Entry temp=iter.next();
```

```java
                    System.out.println(temp.getKey()+"---
"+temp.getValue());
                }


            //第二种遍历Map的方式
            Set<Integer> set2=map.keySet();
            for(Iterator<Integer>
iter=set2.iterator();iter.hasNext();) {
                    Integer temp=iter.next();
                    System.out.println(temp+"---
"+map.get(temp));
                }
        }
}
```

【示例9-15】遍历List方法一：普通for循环

```java
for(int i=0;i<list.size();i++){//list为集合的对象名
    String temp = (String)list.get(i);
    System.out.println(temp);
}
```

【示例9-16】遍历List方法二：增强for循环(使用泛型!)

```java
for (String temp : list) {
System.out.println(temp);
}
```

【示例9-17】遍历List方法三：使用Iterator迭代器(1)

```java
for(Iterator iter= list.iterator();iter.hasNext();){
    String temp = (String)iter.next();
    System.out.println(temp);
}
```

【示例9-18】遍历List方法四：使用Iterator迭代器(2)

```java
Iterator  iter =list.iterator();
while(iter.hasNext()){
    Object  obj  = iter.next();
    iter.remove();//如果要遍历时，删除集合中的元素，建议使用这种方式!
    System.out.println(obj);
}
```

【示例9-19】遍历Set方法一：增强for循环

```java
for(String temp:set){
System.out.println(temp);
}
```

【示例9-20】遍历Set方法二：使用Iterator迭代器

```
1    Iterator  iter =list.iterator();
2    while(iter.hasNext()){
3         Object  obj =  iter.next();
4         iter.remove();//如果要遍历时，删除集合中的元素，建议使用这种方式!
5         System.out.println(obj);
6    }
```

【示例9-19】遍历Set方法一：增强for循环

```
1    for(String temp:set){
2    System.out.println(temp);
3    }
```

【示例9-20】遍历Set方法二：使用Iterator迭代器

```
1    for(Iterator iter = set.iterator();iter.hasNext();){
2         String temp = (String)iter.next();
3         System.out.println(temp);
4    }
```

【示例9-21】遍历Map方法一：根据key获取value

```
1    Map<Integer, Man> maps = new HashMap<Integer, Man>();
2    Set<Integer>  keySet =  maps.keySet();
3    for(Integer id : keySet){
4    System.out.println(maps.get(id).name);
5    }
```

【示例9-22】遍历Map方法二：使用entrySet

```
1    Set<Entry<Integer, Man>>  ss = maps.entrySet();
2    for (Iterator iterator = ss.iterator(); iterator.hasNext();) {
3         Entry e = (Entry) iterator.next();
4         System.out.println(e.getKey()+"--"+e.getValue());
```

分享到： ⭐ 🅱 🔗 ➕   0 评论   ✏️