

笔记总链接:http://bbs.itheima.com/thread-200600-1-1.html

6、集合

6.2 集合类 6.2.5 Map、HashMap、TreeMap

Map:一次添加一对元素, Collection一次添加一个元素。

常用方法:

Map也称为双列集合, Collection集合称为单列集合。 其实Map集合中存储的就是键值对。 map集合中必须保证键的唯一性。 P.S. 关于泛型相关的知识请参看Java高级知识笔记: http://bbs.itheima.com/thread-107670-1-1.html。

1、添加

value put(key,value):返回前一个和key关联的值,如果没有返回null。 2、删除

void clear():清空map集合。 value remove(Object key):根据指定的key删除这个键值对。

3、判断 boolean containsKey(key);

boolean containsValue(value); boolean isEmpty();

4、获取

value get(key):通过键获取值,如果没有该键返回null。 当然可以通过返回null,来判断是否包含指定键。 int size():获取键值对个数。

示例1: import java.util.HashMap; 01. 02. import java.util.Map; 03. 04. public class MapDemo{ public static void main(String[] args){ 05. Map<Integer,String> map = new HashMap<Integer,String>(); 06.

method(map); 07. } 08. 09. public static void method(Map<Integer,String> map){ //学号和姓名 10.

//添加元素 11. System. out.println(map.put(8,"旺财")); 12. System. out.println(map.put(8,"小强")); 13. 14. System. out.println(map); 15. map.put(2, "张三"); 16. map.put(7, "赵六"); 17. System. out.println(map); 18. 19. //删除 20. System. out.println("remove:" + map.remove(2)); 21. 22. //判断 23. System. out.println("containsKey:" + map.containsKey(7)); 24.

25. //获取 26. System. out.println("get:" + map.get(7)); 27. } 28. 29. 复制代码 运行结果: ■ 管理员: C:\Windows\system32\cmd.exe D:∖code∖day18>javac MapDemo.java D:\code\day18>java MapDemo null 旺财 _7=赵六,8=小强> (2=张二, remove:张 containsKey:true get:赵六 获取Map集合元素并打印方式一: 示例2: 01. import java.util.HashMap; import java.util.Iterator; 02.

03. import java.util.Map; 04. import java.util.Set; 05. 06. public class MapDemo{ public static void main(String[] args){ 07. 08. Map<Integer,String> map = new HashMap<Integer,String>(); method(map); 09. 10. 11. public static void method(Map<Integer,String> map){ 13. map.put(8, "王五"); map.put(2, "赵六"); 14. map.put(7, "小强"); 15. map.put(6, "旺财"); 16. 17. //取出map中的所有元素。 18. //原理,通过keySet方法获取map中所有的键所在的set集合,在通过set的迭代器获取到每一个键。 19. //再对每一个键通过map集合的get方法获取其对应的值即可。 20. 21. Set<Integer> keySet = map.keySet(); 22. 23. Iterator<Integer> it = keySet.iterator();

while(it.hasNext()){ 25. 26. Integer key = it.next(); 27. String value = map.get(key); System.out.println(key + ":" + value); 28. 29. } 30. } } 31. 复制代码 运行结果: - - X 画 管理员: C:\Windows\system32\cmd.exe D:\code\day18>javac MapDemo.java

D:\code\day18>java MapDemo

map.put(7, "小强");

map.put(6, "旺财");

*/

通过Map转成Set就可以迭代。

找到了另一个方法, entrySet。

while(it.hasNext()){

import java.util.HashMap;

import java.util.Iterator;

public static void main(String[] args){

method(map);

map.put(8, "王五");

map.put(2, "赵六");

map.put(7, "小强");

map.put(6, "旺财");

while(it.hasNext()){

import java.util.Map;

public class MapDemo{

15.

16. 17. 18.

19.

20.

22. 23.

24.

25. 26.

27.

28.

29.

30.

02. 03.

04.

05. 06.

07.

08.

09.

10. 11.

12.

13.

14.

15. 16.

17. 18.

19. 20.

21.

22. 23. 24.

}

运行结果:

复制代码

Map常用的子类:

爾 管理员: C:\Windows\system32\cmd.exe

this.name = name;

this.name = name;

public String getName(){

return this .name;

public void setAge(int age){

this.age = age;

public int getAge(){

return this .age;

public void setName(String name){

this.age = age;

12.

13.

14. 15. 16.

17.

18. 19.

20. 21.

22. 23.

24.

25. 26. 27.

28.

29.

30.

34.

38. 39.

44.

45.

46.

48.

49.

54.

56.

59.

60.

61.

63.

64.

65.

66.

68. 69.

70. 71.

73.

74.

75.

76. 77.

78.

public static void main(String[] args){

hm.put(new Student("lisi",38),"北京");

hm.put(new Student("zhaoliu" ,24),"上海");

hm.put(new Student("xiaoqiang" ,31),"沈阳");

hm.put(new Student("wangcai" ,28),"大连"); hm.put(new Student("zhaoliu" ,24),"铁岭");

Iterator < Student > it = hm.keySet().iterator();

while(it.hasNext()){

國 管理员: C:\Windows\system32\cmd.exe

D:\code\day18>java HashMapDemo xiaoqiang:31---沈阳 wangcai:28---大连

键有了判断依据, HashMap中的值就被覆盖。

import java.util.Comparator;

import java.util.Iterator;

import java.util.TreeMap;

public class HashMapDemo{

public static void main(String[] args){

tm.put(new Student("lisi",38),"北京");

tm.put(new Student("zhaoliu" ,24),"上海");

tm.put(new Student("xiaoqiang",31),"沈阳");

tm.put(new Student("wangcai",28),"大连");

tm.put(new Student("zhaoliu" ,24),"铁岭");

Map.Entry<Student,String> me = it.next();

Student key = me.getKey();

String value = me.getValue();

while(it.hasNext()){

}

}

class ComparatorByName implements Comparator<Student>{

int temp = s1.getName().compareTo(s2.getName());

//将学生对象和学生的归属地通过键与值存储到map集合中

Iterator<Map.Entry<Student,String>> it = tm.entrySet().iterator();

System.out.println(key.getName() + ":" + key.getAge() + "---" + value);

TreeMap<Student,String> tm = new TreeMap<Student,String>(new ComparatorByName());

return temp == 0?s1.getAge() - s2.getAge():temp;

public int compare(Student s1,Student s2){

import java.util.Map;

D:\code\day18>javac HashMapDemo.java

}

}

复制代码

wangcai:28 lisi:38---北京 ::-:24---铁岭

}

运行结果:

P.S.

示例6:

01.

02.

03.

04. 05.

06.

07.

08.

09. 10. 11.

12.

13. 14.

15.

16. 17.

18.

19.

20.

21.

22. 23.

24.

26.

27.

28.

29.

31.

32.

}

Student key = it.next();

String value = hm.get(key);

//将学生对象和学生的归属地通过键与值存储到map集合中

HashMap<Student,String> hm = new HashMap<Student,String>();

System.out.println(key.getName() + ":" + key.getAge() + "---" + value);

}

}

}

D:\code\day18>java MapDemo

获取Map集合元素并打印方式二: 示例3: import java.util.HashMap; 01. import java.util.Iterator; 02. 03. import java.util.Map; import java.util.Set; 04. 05. public class MapDemo{ 06. public static void main(String[] args){ 07. Map<Integer,String> map = new HashMap<Integer,String>(); 08. method(map); 09. } 10. 11. public static void method(Map<Integer,String> map){ 12. 13. map.put(8, "王五"); map.put(2, "赵六"); 14.

该方法将键和值的映射关系作为对象存储到了Set集合中,而这个映射关系的类型就是Map.Entry类型

Set < Map. Entry < Integer, String >> entry Set = map.entry Set();

Iterator<Map.Entry<Integer,String>> it = entrySet.iterator();

Map.Entry<Integer,String> me = it.next();

Integer key = me.getKey();

String value = me.getValue();

System. out.println(key + ":" + value); 31. } 32. 33. 34. 复制代码 运行结果: - - X ■ 管理员: C:\Windows\system32\cmd.exe D:\code\day18>javac MapDemo.java D:\code\day18>java MapDemo 获取Map集合元素并打印方式三: 示例4: import java.util.Collection;

Map<Integer,String> map = new HashMap<Integer,String>();

public static void method(Map<Integer,String> map){

Collection<String> values = map.values();

Iterator<String> it = values.iterator();

|--Hashtable:内部结构是哈希表,是同步的。不允许null作为键,null作为值。

|--Properties:用来存储键值对型的配置文件的信息,可以和IO技术相结合。

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

- 0 X

|--HashMap:内部结构式哈希表,不是同步的。允许null作为键,null作为值。 |--TreeMap:内部结构式二叉树,不是同步的。可以对Map结合中的键进行排序。 hashSet实现Set接口,由哈希表(实际上是一个HashMap实例)支持。 示例5: 01. import java.util.HashMap; import java.util.Iterator; 02. 03. class Student { 04. private String name; 05. 06. private int age; 07. public Student(){ 08. } 09. 10. 11. public Student(String name,int age){

public int hashCode() { 32. final int prime = 31; 33. int result = 1; result = prime * result + age; 35. result = prime * result + ((name == null) ? 0 : name.hashCode()); 37. return result; } public boolean equals(Object obj) { 40. if (this == obj) 42. return true; if (obj == null)43. return false; if (getClass() != obj.getClass()) return false; Student other = (Student) obj; 47. if (age != other.age) return false; if (name == null) { 50. if (other.name != null) 52. return false; 53. } else if (!name.equals(other.name)) return false; return true; 55. } 57. 58. public class HashMapDemo{

} 33. 复制代码 运行结果: _ D X 画 管理员: C:\Windows\system32\cmd.exe D:∖code∖day18>javac HashMapDemo.java lisi:38wangcai:28xiaoqiang:31zhaoliu:24--~END~ ~爱上海,爱黑马~