

entrySet()

将Map集合转换为存储元素类型为Map的Set集合

```
HashMap<String, Integer> map = new HashMap<String, Integer>();
for (HashMap.Entry<String, Integer> entry : dict.entrySet()) {
    System.out.println(entry.getKey() + " is in class " + entry.getValue());
}
```

keySet()

map.keySet()以Set集合的形式返回Map集合中所有的键对象Key

```
HashMap<String, Integer> map = new HashMap<String, Integer>();
for (int mapKey : map.keySet())
    {
        System.out.println( "key:" + mapKey + "value:" + map.get(mapKey));
    }
```

Iterator

Map集合中所有键对象转换为Set单列集合，接着将包含键对象的Set集合转换为Iterator接口对象，然后遍历Map集合中所有的键，在根据键获取相应的值

```
HashMap<String,Integer> map = new HashMap <>();
Set<String> keySet = map.keySet();
Iterator<String> iterator = keySet.iterator();
while (iterator.hasNext()) {
    String key = iterator.next();
    System.out.println( "key:" + key + "value:" + map.get(key));
}
```

forEach(BiConsumer action)

与Collection结合遍历类似，在JDK8中也根据Lambda表达式特性新增了一个forEach(BiConsumer action)方法来遍历Map集合

```
HashMap<String,Integer> map = new HashMap <>();
map.forEach((key,value) -> System.out.println(key + ":" + value));
```

values

通过这个方法可以直接获取Map中存储所有值的Collection集合

```
HashMap<String,Integer> map = new HashMap <>();
Collection values = map.values();
values.forEach(v -> System.out.println(v));
```

