

Java集合框架Map接口

Map接口概念

Map接口中常用的方法

Map接口概念

Java集合框架中的Map接口表示一种键值对（key-value）的数据结构，其中每个元素都包含一个唯一的键和对应的值。在Map中，每个键必须是唯一的，而值可以重复。Map接口提供了一些方法来实现基本的键值对操作，例如添加、查找、删除等。

Map接口的实现类可以根据其内部数据结构的不同，分为几种类型，例如HashMap、TreeMap和LinkedHashMap等。不同的实现类具有不同的特点和性能特征，应该根据实际需求来选择合适的实现类。

Map接口中常用的方法

1. put(K key, V value)：将指定的键值对添加到Map中。
2. get(Object key)：获取指定键对应的值，如果键不存在，则返回null。
3. remove(Object key)：从Map中删除指定键及对应的值。
4. containsKey(Object key)：判断Map中是否包含指定的键。
5. containsValue(Object value)：判断Map中是否包含指定的值。
6. size()：返回Map中键值对的数量。
7. keySet()：返回一个包含Map中所有键的Set集合。
8. values()：返回一个包含Map中所有值的Collection集合。
9. entrySet()：返回一个包含Map中所有键值对的Set集合。

Map接口中的键和值都可以是任意类型的，但是在使用时需要注意键的唯一性。下面是一个使用HashMap实现的简单示例，演示了如何创建Map、添加元素、遍历Map等操作：

```
1  import java.util.HashMap;
2  import java.util.Map;
3
4  public class MapExample {
5      public static void main(String[] args) {
6          // 创建一个HashMap对象
7          Map<String, Integer> map = new HashMap<>();
8
9          // 向Map中添加元素
10         map.put("apple", 1);
11         map.put("banana", 2);
12         map.put("orange", 3);
13         map.put("pear", 4);
14
15         // 输出Map大小
16         System.out.println("Map大小: " + map.size());
17
18         // 遍历Map
19         for (Map.Entry<String, Integer> entry : map.entrySet()) {
20             String fruit = entry.getKey();
21             int count = entry.getValue();
22             System.out.println(fruit + ": " + count);
23         }
24
25         // 删除一个元素
26         map.remove("banana");
27
28         // 判断Map中是否包含指定键
29         System.out.println("Map中是否包含pear: " + map.containsKey("pear"));
30
31         // 清空Map
32         map.clear();
33
34         // 判断Map是否为空
35         System.out.println("Map是否为空: " + map.isEmpty());
36     }
37 }
38
```