Map接口下的方法

Map接口下的方法

Map中的元素以"键值对"(Key-Value)的形式出现

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

当Key值不存在时向Map中添加元素（键值对）

Key是唯一的，value是可重覆的

当Key存在时，修改键对应的值

添加方法：

```
map.put(1001, "张三");
map.put(1008, "李四");
map.put(1003, "王麻子");
```

判断是否包含某个key

```
System.out.println(map.containsKey(1001));
System.out.println(map.containsKey(1002));
```

判断是否包含value

```
System.out.println(map.containsValue("王麻子"));
System.out.println(map.containsValue("小名"));
```

根据key获取value

```
System.out.println(map.get(1001));
System.out.println(map.get(1002));
```

获取map中的所有key

```
System.out.println(map.keySet());
```

获取map中所有的value

```
System.out.println(map.values());
```

获取所有的键值对

```
System.out.println(map.entrySet());
```

根据key值删除元素

```
map.remove(1001);
map.remove(1002);
System.out.println(map);
System.out.println(map.size());
```

遍历map

```
for (Integer key : map.keySet()) {
    System.out.printf("%d - %s\n", key, map.get(key));
}

for (Iterator<Integer> iterator = map.keySet().iterator(); iterator.hasNext(); ) {
    Integer key = iterator.next();
    System.out.printf("%d - %s\n", key, map.get(key));
}

for (Map.Entry<Integer, String> entry : map.entrySet()) {
    Integer key = entry.getKey();
    String value = entry.getValue();
    System.out.printf("%d - %s\n", key, map.get(key));
}

for (Iterator<Map.Entry<Integer, String>> iterator = map.entrySet().iterator();
iterator.hasNext(); ) {
    Map.Entry<Integer, String> entry = iterator.next();
    Integer key = entry.getKey();
    String value = entry.getValue();
    System.out.printf("%d - %s\n", key, map.get(key));
}
```