Java HashMap: How to get key from value?

If your data structure has ***many-to-one*** mapping between keys and values **you should iterate over entries and pick all suitable keys**:

public static <T, E> Set<T> getKeysByValue(Map<T, E> map, E value) {

Set<T> keys = new HashSet<T>();

for (Entry<T, E> entry : map.entrySet()) {

**//判断当前entry是否含有value**

if (Objects.equals(value, entry.getValue())) {

**//通过含有value的entry得到对应的key**

keys.add(entry.getKey());

}

}

return keys;

}

In case of ***one-to-one*** relationship, you can **just return the first matched key**:

public static <T, E> T getKeyByValue(Map<T, E> map, E value) {

for (Entry<T, E> entry : map.entrySet()) {

if (Objects.equals(value, entry.getValue())) {

**//一找到就return**

return entry.getKey();

}

}

return null;

}

In Java 8: 函数式编程

public static <T, E> Set<T> getKeysByValue(Map<T, E> map, E value) {

return map.entrySet()

.stream()

.filter(entry -> Objects.equals(entry.getValue(), value))

.map(Map.Entry::getKey)

.collect(Collectors.toSet());

}

Also, for Guava users, [BiMap](http://docs.guava-libraries.googlecode.com/git/javadoc/com/google/common/collect/BiMap.html) may be useful. For example:

BiMap<Token, Character> tokenToChar =

ImmutableBiMap.of(Token.LEFT\_BRACKET, '[', Token.LEFT\_PARENTHESIS, '(');

Token token = tokenToChar.inverse().get('(');

Character c = tokenToChar.get(token);