

# 没有指针的Java

2020年2月29日 星期六 下午10:57

1. 引用实质就是指针, 但受控、安全

- 全查空
- 没有指针运算
- 不能访问没有引用到的内存
- 自动回收垃圾

2. C Java

- 传地址 对象
- \*(&t5) 指针运算 数组 args[5]
- 函数指针 接口、Lambda表达式
- 指向结点的指针 对象的引用

3. Java Native Interface (JNI)

允许与其它语言代码交互

4. "==" 基本类型 { 不同数值类型: 转换后比较  
浮点数  
boolean 不可与 int 比

```
Integer i = new Integer(10);
Integer j = new Integer(10);
System.out.println(i==j); //false, 因为对象是两个

Integer m = 10;
Integer n = 10;
System.out.println(m==n); //true, 因为对象有缓存

Integer p = 200;
Integer q = 200;
System.out.println(p==q); //false, 因为对象是两个
```

引用类型

- 枚举类型
  - 内部进行了惟一实例化, 所以可以直接判断
- 引用对象
  - 是直接看两个引用是否一样
  - 如果要判断内容是否一样, 则要重写equals方法
  - 如果重写equals方法, 则最好重写 hashCode()方法

String 判内容用 equals 不用 ==  
对常量特殊处理

但是字符串常量 (String literal) 及字符串常量会进行内部化 (interned) 相同的字符串常量是 == 的

例

```
String hello = "Hello", lo = "lo";
System.out.println( hello == "Hello"); //true
System.out.println( Other.hello == hello ); //true

System.out.println( hello == ("Hel"+"lo") ); //true
System.out.println( hello == ("Hel"+lo) ); //false

System.out.println( hello == new String("Hello")); //false
System.out.println( hello == ("Hel"+lo).intern()); //true
```