

Java 面试题

Java 基础

1. JDK 和 JRE 有什么区别？

- JDK: Java Development Kit 的简称, java 开发工具包, 提供了 java 的开发环境和运行环境。
- JRE: Java Runtime Environment 的简称, java 运行环境, 为 java 的运行提供了所需环境。

具体来说 JDK 其实包含了 JRE, 同时还包含了编译 java 源码的编译器 javac, 还包含了很多 java 程序调试和分析的工具。简单来说: 如果你需要运行 java 程序, 只需安装 JRE 就可以了, 如果你需要编写 java 程序, 需要安装 JDK。

2. == 和 equals 的区别是什么？

== 解读

对于基本类型和引用类型 == 的作用效果是不同的, 如下所示:

- 基本类型: 比较的是值是否相同;
- 引用类型: 比较的是引用是否相同;

代码示例:

```
1 String x = "string";
2 String y = "string";
3 String z = new String("string");
4 System.out.println(x==y); // true
5 System.out.println(x==z); // false
6 System.out.println(x.equals(y)); // true
7 System.out.println(x.equals(z)); // true
```

代码解读: 因为 x 和 y 指向的是同一个引用, 所以 == 也是 true, 而 new String()方法则重写开辟了内存空间, 所以 == 结果为 false, 而 equals 比较的一直是值, 所以结果都为 true。

equals 解读

equals 本质上就是 ==, 只不过 String 和 Integer 等重写了 equals 方法, 把它变成了值比较。看下面的代码就明白了。

首先来看默认情况下 equals 比较一个有相同值的对象, 代码如下:

```
1
2 class Cat {
3     public Cat(String name) {
4         this.name = name;
5     }
6
7     private String name;
```

```

8
9     public String getName() {
10         return name;
11     }
12
13     public void setName(String name) {
14         this.name = name;
15     }
16 }
17
18 Cat c1 = new Cat("王磊");
19 Cat c2 = new Cat("王磊");
20 System.out.println(c1.equals(c2)); // false

```

输出结果出乎我们的意料，竟然是 false？这是怎么回事，看了 equals 源码就知道了，源码如下：

```

1     public boolean equals(Object obj) {
2         return (this == obj);
3     }

```

原来 equals 本质上就是 ==。

那问题来了，两个相同值的 String 对象，为什么返回的是 true？代码如下：

```

1     String s1 = new String("老王");
2     String s2 = new String("老王");
3     System.out.println(s1.equals(s2)); // true

```

同样的，当我们进入 String 的 equals 方法，找到了答案，代码如下：

```

1
2     public boolean equals(Object anObject) {
3         if (this == anObject) {
4             return true;
5         }
6         if (anObject instanceof String) {
7             String anotherString = (String)anObject;
8             int n = value.length;
9             if (n == anotherString.value.length) {
10                 char v1[] = value;
11                 char v2[] = anotherString.value;
12                 int i = 0;
13                 while (n-- != 0) {
14                     if (v1[i] != v2[i])
15                         return false;
16                     i++;
17                 }
18                 return true;
19             }
20         }
21         return false;
22     }

```

原来是 String 重写了 Object 的 equals 方法，把引用比较改成了值比较。

总结： == 对于基本类型来说是值比较，对于引用类型来说是比较的是引用；而 equals 默认情况下是引用比较，只是很多类重新了 equals 方法，比如 String、Integer 等把它变成了值比较，所以一般情况下 equals 比较的是值是否相等。