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程序调试工具和分析工具。简单来说：如果你需要运行Java程序，只需安装JRE就可以了，如果你还需要编写Java程序，需要安装JDK。

1. ==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:



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 == 结构为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：



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



原来equals本质上就是==。

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



同样的，当我们进入String的equals方法，找到了答案，代码如下：



原来是String重写了Object的equals的方法，把引用比较改为了值比较。

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引用；而equals默认情况下是引用比较，只是很多类重写了equals方法，比如String、Integer等把它变成了值比较，所以一般情况下equals比较的值是否相等。

1. 两个对象的hashCode相同，则equals也一定为true，对吗？

不对，两个对象的hashCode()相同，equals()不一定true。

代码示例：



执行结果：

str1:1179395 | str2:1179395

false

代码解读：很显然“通话”和“重地”的hashCode()相同，然而equals()则为false，因为在散列表中，hashCode()相等即两个键值对的哈希值相等，然而哈希值相等，并不一定能得出键值对相等。

1. final在java中有什么作用？

●final修饰的类叫最终类，该类不能被继承。

●final修饰的方法不能被重写。

●final修饰的变量叫常量，常量必须初始化，初始化之后值就不能被修改。

1. Java中的Math.round(1.5)等于多少？

等于-1，因为在数轴上取值时，中间值（0.5）向右取整，所以正0.5是往上取整，负0.5是直接舍弃。