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。

1. String属于基础的数据类型吗？

String不属于基础类型，基础类型有8种：byte、boolean、char、short、int、float、long、double，而String属于对象。

1. Java中操作字符串都有哪些类？它们之间有什么区别？

操作字符串的类有：String、StringBuffer、StringBuilder。

String和StringBuffer、StringBuilder的区别在于String声明的是不可变的对象，每次操作都会生成新的String对象，然后将指针指向新的String对象，而StringBuffer、StringBuilder可以在原有对象的基础上进行操作，所以在经常改变字符串内容的情况下最好不要使用String。

StringBuffer和StringBuilder的最大区别在于，StringBuffer是线程安全的，而StringBuilder是非线程安全的，但StringBuilder的性能却高于StringBuffer，，所以在单线程环境下推荐使用StringBuilder，多线程环境下推荐使用StringBuffer。

1. String str = “1” 与String str= new String(“i”)一样吗？

不一样，因为内存的分配方式不一样。String str=”i”的方式，Java虚拟机会将其分配到常量池中；而String str = new String(“i”)则会被分到堆内存中。

1. 如何将字符串反转？

使用StringBuilder 或者StringBuffer的reverse()方法。

示例代码：

//StringBuffer reverse

StringBuffer stringBuffer = new StringBuffer();

stringBuffer.append(“abcdefg”);

System.out.println(stringBuffer.reverse());//gfedcba

//StringBuilder reverse

StringBuilder stringBuilder = new StringBuilder();

stringBuilder.append(“abcdefg”);

System.out.println(stringBuilder.reverse());//gfedcba

1. String类的常用方法有哪些？

•indexOf():返回指定字符的索引。

•charAt():返回指定索引处的字符。

•replace():字符串替换。

•trim():去除字符串两端空白。

•split():分割字符串，返回一个分割后的字符串数组。

•getBytes():返回字符串的byte类型数组。

•length():返回字符串的长度。

•toLowerCase():将字符串转换成小写字母。

•toUpperCase():将字符串转换成大写字母。

•substring():截取字符串。

•equals():字符串比较。

1. 抽象类必须有抽象方法吗？

不需要，抽象类不一定非要有抽象方法。

示例代码：

abstract class Cat{

public static void sayHi(){

System.out.println(“hi~”)

}

}

上面代码，抽象类并没有抽象方法但完全可以运行。

1. 普通类和抽象类有哪些区别？

•普通类不能包含抽象方法，抽象类可以包含抽象方法。

•抽象类不能直接示例话，普通类可以直接实例化。

1. 抽象类能使用final修饰吗？

不能，定义抽象类就是让其他类继承的，如果定义为final该类就不能被继承，这样彼此就会产生矛盾，所以final不能修饰抽象类，如下图所示，编辑器也会提示错误信息。

1. 接口和抽象类有什么区别？

•实现：抽象类的子类使用extends来继承；接口必须使用implements来实现接口。

•构造函数：抽象类可以有构造函数；接口不能有。

•实现数量：类可以实现多个接口；但是只能继承一个抽象类。

•访问修饰符：接口中的方法默认使用public修饰；抽象类中的方法可以是任意访问修饰符。

1. Java中IO流分为几种？

按功能来分：输入流（input）和输出流（write）。

按类型来分：字节流和字符流。

字节流和字符流的区别是：字节流按8位传输以字节为单位输入输出数据，字符流按16位传输以字符为单位输入输出数据。