1、&和&&的区别？

答：&运算符有两种用法：(1)按位与；(2)逻辑与。&&运算符是短路与运算。逻辑与跟短路与的差别是非常巨大的，虽然二者都要求运算符左右两端的布尔值都是true整个表达式的值才是true。&&之所以称为短路运算是因为，如果&&左边的表达式的值是false，右边的表达式会被直接短路掉，不会进行运算。很多时候我们可能都需要用&&而不是&，例如在验证用户登录时判定用户名不是null而且不是空字符串，应当写为：username != null &&!username.equals(“”)，二者的顺序不能交换，更不能用&运算符，因为第一个条件如果不成立，根本不能进行字符串的equals比较，否则会产生NullPointerException异常。注意：逻辑或运算符（|）和短路或运算符（||）的差别也是如此。

2、Math.round(11.5) 等于多少？Math.round(-11.5)等于多少？

答：Math.round(11.5)的返回值是12，Math.round(-11.5)的返回值是-11。四舍五入的原理是在参数上加0.5然后进行下取整。

3、描述一下JVM加载class文件的原理机制？

答：JVM中类的装载是由类加载器（ClassLoader）和它的子类来实现的，Java中的类加载器是一个重要的Java运行时系统组件，它负责在运行时查找和装入类文件中的类。

由于Java的跨平台性，经过编译的Java源程序并不是一个可执行程序，而是一个或多个类文件。当Java程序需要使用某个类时，JVM会确保这个类已经被加载、连接（验证、准备和解析）和初始化。类的加载是指把类的.class文件中的数据读入到内存中，通常是创建一个字节数组读入.class文件，然后产生与所加载类对应的Class对象。加载完成后，Class对象还不完整，所以此时的类还不可用。当类被加载后就进入连接阶段，这一阶段包括验证、准备（为静态变量分配内存并设置默认的初始值）和解析（将符号引用替换为直接引用）三个步骤。最后JVM对类进行初始化，包括：1)如果类存在直接的父类并且这个类还没有被初始化，那么就先初始化父类；2)如果类中存在初始化语句，就依次执行这些初始化语句。

4、String s = new String(“xyz”);创建了几个字符串对象？

答：两个对象，一个是静态区的”xyz”，一个是用new创建在堆上的对象。

5、数据类型之间的转换：（1）如何将字符串转换为基本数据类型？（2）如何将基本数据类型转换为字符串？

答：

（1）调用基本数据类型对应的包装类中的方法parseXXX(String)或valueOf(String)即可返回相应基本类型；

（2）一种方法是将基本数据类型与空字符串（""）连接（+）即可获得其所对应的字符串；另一种方法是调用String 类中的valueOf()方法返回相应字符串。

6、怎样将GB2312编码的字符串转换为ISO-8859-1编码的字符串？

答：

代码如下所示：

String s1 = "你好";

String s2 = new String(s1.getBytes("GB2312"), "ISO-8859-1");

7、List、Map、Set三个接口存取元素时，各有什么特点？

答：List以特定索引来存取元素，可以有重复元素。Set不能存放重复元素（用对象的equals()方法来区分元素是否重复）。Map保存键值对（key-value pair）映射，映射关系可以是一对一或多对一。Set和Map容器都有基于哈希存储和排序树的两种实现版本，基于哈希存储的版本理论存取时间复杂度为O(1)，而基于排序树版本的实现在插入或删除元素时会按照元素或元素的键（key）构成排序树从而达到排序和去重的效果。

8、Java中如何实现序列化，有什么意义？

答：序列化就是一种用来处理对象流的机制，所谓对象流也就是将对象的内容进行流化。可以对流化后的对象进行读写操作，也可将流化后的对象传输于网络之间。序列化是为了解决对象流读写操作时可能引发的问题（如果不进行序列化可能会存在数据乱序的问题）。

要实现序列化，需要让一个类实现Serializable接口，该接口是一个标识性接口，标注该类对象是可被序列化的，然后使用一个输出流来构造一个对象输出流并通过writeObject(Object)方法就可以将实现对象写出（即保存其状态）；如果需要反序列化则可以用一个输入流建立对象输入流，然后通过readObject方法从流中读取对象。序列化除了能够实现对象的持久化之外，还能够用于对象的深度克隆。

9、Java中有几种类型的流？

答：字节流和字符流。字节流继承于InputStream、OutputStream，字符流继承于Reader、Writer。在java.io 包中还有许多其他的流，主要是为了提高性能和使用方便。

10、获得一个类的类信息有哪些方式？

答：

- 方法1：类型.class，例如：String.class

- 方法2：对象.getClass()，例如：”hello”.getClass()

- 方法3：Class.forName()，例如：Class.forName(“java.lang.String”)

11、ArrayList与LinkedList的区别

答：

都是实现List接口的列表，ArrayList是基于数组的数据结构，LinkedList是基于链表的数据结构，当获取特定元素时，ArrayList效率比较快，它通过数组下标即可获取，而LinkedList则需要移动指针。当存储元素与删除元素时LinkedList效率较快，只需要将指针移动指定位置增加或者删除即可，而ArrayList需要移动数据。

12、JDK、JRE、JVM关系是什么？

答：

JDK（Java Development Kit）即为Java开发工具包，包含编写Java程序所必须的编译、运行等开发工具以及JRE。开发工具如：用于编译java程序的javac命令、用于启动JVM运行java程序的java命令、用于生成文档的javadoc命令以及用于打包的jar命令等等。

JRE（Java Runtime Environment）即为Java运行环境，提供了运行Java应用程序所必须的软件环境，包含有Java虚拟机（JVM）和丰富的系统类库。系统类库即为java提前封装好的功能类，只需拿来直接使用即可，可以大大的提高开发效率。

JVM（Java Virtual Machines）即为Java虚拟机，提供了字节码文件（.class）的运行环境支持。

简单说，就是JDK包含JRE，JRE包含JVM。

13、解释内存中的栈(stack)、堆(heap)和方法区(method area)的用法。

答：

通常我们定义一个基本数据类型的变量，一个对象的引用，还有就是函数调用的现场保存都使用JVM中的栈空间；而通过new关键字和构造器创建的对象则放在堆空间，堆是垃圾收集器管理的主要区域，由于现在的垃圾收集器都采用分代收集算法，所以堆空间还可以细分为新生代和老生代，再具体一点可以分为Eden、Survivor（又可分为From Survivor和To Survivor）、Tenured；方法区和堆都是各个线程共享的内存区域，用于存储已经被JVM加载的类信息、常量、静态变量、JIT编译器编译后的代码等数据；程序中的字面量（literal）如直接书写的100、”hello”和常量都是放在常量池中，常量池是方法区的一部分，。栈空间操作起来最快但是栈很小，通常大量的对象都是放在堆空间，栈和堆的大小都可以通过JVM的启动参数来进行调整，栈空间用光了会引发StackOverflowError，而堆和常量池空间不足则会引发OutOfMemoryError。

14、阐述java的单继承+接口实现机制比多继承的优点？

15、新建一个流对象，下面哪个选项的代码是错误的？（）

A）new BufferedWriter(new FileWriter("a.txt"));

B）new BufferedReader(new FileInputStream("a.dat"));

C）new GZIPOutputStream(new FileOutputStream("a.zip"));

D）new ObjectInputStream(new FileInputStream("a.dat"));

答案：B