1、&和&&的区别？

& 位运算 && 逻辑运算

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

12 -11

3、描述一下JVM加载class文件的原理机制？

    Java中的所有类，都需要由类加载器装载到JVM中才能运行。类加载器本身也是一个类，而它的工作就是把class文件从硬盘读取到内存中。在写程序的时候，我们几乎不需要关心类的加载，因为这些都是隐式装载的，除非我们有特殊的用法，像是反射，就需要显式的加载所需要的类。

      Java类的加载是动态的，它并不会一次性将所有类全部加载后再运行，而是保证程序运行的基础类(像是基类)完全加载到jvm中，至于其他类，则在需要的时候才加载。这当然就是为了节省内存开销。

     Java的类加载器有三个，对应Java的三种类:

     Bootstrap Loader  // 负责加载系统类 (指的是内置类，像是String，对应于C#中的System类和C/C++标准库中的类)  
            |   
          - - ExtClassLoader   // 负责加载扩展类(就是继承类和实现类)  
                          |   
                      - - AppClassLoader   // 负责加载应用类(程序员自定义的类)

      三个加载器各自完成自己的工作，但它们是如何协调工作呢？哪一个类该由哪个类加载器完成呢？为了解决这个问题，Java采用了委托模型机制。

      委托模型机制的工作原理很简单：当类加载器需要加载类的时候，先请示其Parent(即上一层加载器)在其搜索路径载入，如果找不到，才在自己的搜索路径搜索该类。这样的顺序其实就是加载器层次上自顶而下的搜索，因为加载器必须保证基础类的加载。之所以是这种机制，还有一个安全上的考虑：如果某人将一个恶意的基础类加载到jvm，委托模型机制会搜索其父类加载器，显然是不可能找到的，自然就不会将该类加载进来。

      我们可以通过这样的代码来获取类加载器:

ClassLoader loader = ClassName.class.getClassLoader();

ClassLoader ParentLoader = loader.getParent();

      注意一个很重要的问题，就是Java在逻辑上并不存在BootstrapKLoader的实体！因为它是用C++编写的，所以打印其内容将会得到null。  
      前面是对类加载器的简单介绍，它的原理机制非常简单，就是下面几个步骤:

1.装载:查找和导入class文件;

2.连接:

      (1)检查:检查载入的class文件数据的正确性;

      (2)准备:为类的静态变量分配存储空间;

      (3)解析:将符号引用转换成直接引用(这一步是可选的)

3.初始化:初始化静态变量，静态代码块。

      这样的过程在程序调用类的静态成员的时候开始执行，所以静态方法main()才会成为一般程序的入口方法。类的构造器也会引发该动作

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

常量池 堆 俩个 xyz

字符串对象 栈中一个 堆中一个 指向常量池中”xyz”

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？

用各自的类 如 Integer.parseInt()

（2）如何将基本数据类型转换为字符串？

强制转化

new String（基本数据类型）

6、怎样将GB2312编码的字符串转换为ISO-8859-1编码的字符串？

s=new String(s.getBytes(“ISO-8859-1”),”ISO-8859-1”);

7、List、Map、Set三个接口存取元素时，各有什么特点？

List与Set都是单列元素的集合，它们有一个共同的父接口Collection。

Set里面不允许有重复的元素，

存元素：add方法有一个boolean的返回值，当集合中没有某个元素，此时add方法可成功加入该元素时，则返回true；当集合含有与某个元素equals相等的元素时，此时add方法无法加入该元素，返回结果为false。

取元素：没法说取第几个，只能以Iterator接口取得所有的元素，再逐一遍历各个元素。

List表示有先后顺序的集合，

存元素：多次调用add(Object)方法时，每次加入的对象按先来后到的顺序排序，也可以插队，即调用add(int index,Object)方法，就可以指定当前对象在集合中的存放位置。

取元素：方法1：Iterator接口取得所有，逐一遍历各个元素

        方法2：调用get(index i)来明确说明取第几个。

Map是双列的集合，存放用put方法:put(obj key,obj value)，每次存储时，要存储一对key/value，不能存储重复的key，这个重复的规则也是按equals比较相等。

取元素：用get(Object key)方法根据key获得相应的value。

        也可以获得所有的key的集合，还可以获得所有的value的集合，

        还可以获得key和value组合成的Map.Entry对象的集合。

List以特定次序来持有元素，可有重复元素。Set 无法拥有重复元素,内部排序。Map 保存key-value值，value可多值。

8、Java中如何实现序列化，有什么意义？

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？

字节流 字符流:

InputStream，OutputStream，Reader，Writer

10、获得一个类的类信息有哪些方式？

**第一种方式：通过类本身来获得对象**

* Class classname =  this .getClass();

**第二种方式：通过子类的实例获取父类对象**

* ClassName cn =  new  ClassName();
* UserClass = cn.getClass();
* Class SubUserClass = UserClass.getSuperclass();

**第三种方式：通过类名加.class获取对象**

* Class ForClass = \*\*.\*\*.ClassName. class ;(类在包中的路径加. class )

**第四种方式：通过类名的字符串获取对象**

* Class ForName = Class.forName( "\*\*.\*\*.ClassName" );

这种方式在jdbc中常用到

11、ArrayList与LinkedList的区别

.ArrayList是实现了基于动态数组的数据结构，LinkedList基于链表的数据结构。

12、JDK、JRE、JVM关系是什么？

JDK（Java Development Kit）是针对Java开发员的产品，是整个Java的核心，包括了Java运行环境JRE、Java工具和Java基础类库。  
Java Runtime Environment（JRE）是运行JAVA程序所必须的环境的集合，包含JVM标准实现及Java核心类库。  
JVM是Java Virtual Machine（Java虚拟机）的缩写，是整个java实现跨平台的最核心的部分，能够运行以Java语言写作的软件程序。

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

（1）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；

（2）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；

（3）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？

类的多继承有哪些缺点那：

第一，如果一个类继承多个父类，如果父类中的方法名如果相同，那么就会产生歧义。

第二，如果父类中的方法同名，子类中没有覆盖，同样会产生上面的错误。

所以，java中就没有设计类的多继承。

但是接口就设计成多继承，是因为接口可以避免上述问题：

首先，接口中的只有抽象方法和静态常量。

对于一个类实现多个接口的情况和一个接口继承多个接口的情况，因为接口只有抽象方法，具体方法只能由实现接口的类实现（也是因为实现类一定会覆盖接口中的方法），在调用的时候始终只会调用实现类（也就是子类覆盖的方法）的方法（不存在歧义），因此不存在 多继承的第二个缺点；而又因为接口只有静态的常量，但是由于静态变量是在编译期决定调用关系的，即使存在一定的冲突也会在编译时提示出错；而引用静态变量一般直接使用类名或接口名，从而避免产生歧义，因此也不存在多继承的第一个缺点。

15、新建一个流对象，下面哪个选项的代码是错误的？（ B ）

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

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

new BufferedReader(new FileReader(“a.dat”));

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

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