1、请叙述在面向对象编程语言中，类和对象之间的关系。

答：

类：就是具备某些共同特征的实体的集合，它是一种抽象的数据类型，它是对所具有相同特征实体的抽象。在面向对象的程序设计语言中，类是对一类“事物”的属性与行为的抽象。

对象：就是一个真实世界中的实体，对象与实体是一一对应关系的，意思就是现实世界的每一个实体都是一个对象，所以对象是一个具体的概念。

**类是对象的一个集合，对象是类的实例。**

区别

　　1，**类是一个抽象的概念**，它不存在于现实中的时间/空间里，类只是为所有的对象定义了抽象的属性与行为。就好像“Person（人）”这个类，它虽然可以包含很多个体，但它本身不存在于现实世界上。

　　2，**对象是类的一个具体。**它是一个实实在在存在的东西。

　　3，**类是一个静态的概念，**类本身不携带任何数据。当没有为类创建任何对象时，类本身不存在于内存空间中。

　　4，**对象是一个动态的概念。**每一个对象都存在着有别于其它对象的属于自己的独特的属性和行为。对象的属性可以随着它自己的行为而发生改变。

2、请写出三个合乎规范的类名。

(知识点：类名字可由字母、下划线、数字或美元符号组成，且第一个字符不能是数字)

1、MyList

2、YouList

3、HashMapYes

3、请叙述构造方法和普通的方法之间的区别在哪里。

区别：构造方法是一种特殊的方法，它的名字必须与类名完全相同，且其为void类型，但是void必须省略不写，而且允许一个类里面允许有多个构造方法，但是不同构造方法的参数必须不同。

4、请叙述类成员变量和对象的实例变量之间的区别在哪里。

区别：

1、两个变量生命周期不同。

2、两个变量调用方式不同。成员变量只能被对象调用，类变量不仅能被对象调用，也能被类名调用。

3、别名不同。类成员变量就是静态变量，对象的实例变量是成员变量。

4、数据存储位置不同。实例变量存储在堆内存中，静态变量存储在方法区的静态区。

5、为什么修改一个对象的类成员变量，会影响其他由这个类创建的对象相应的类成员变量。

答：因为类成员变量就是静态变量，其保存在全局数据区，是被这个类的所有对象共享的变量。

6、请问如果在代码中试图为一个常量重新赋值，会出现什么错误？

答：会无法为这个常量赋值。

7、为什么类方法不允许访问一个对象的实例变量和其他的实例方法？

答：

对于类中的类方法，在该类被加载到内存时，就分配了相应的入口地址。从而类方法不仅可以被类创建的任何对象调用执行，也可以直接通过类名调用。类方法的入口地址直到程序退出才被取消。

而当类的字节码文件被加载到内存时，类的实例方法不会被分配入口地址，当该类创建对象后，类中的实例方法才分配入口地址，从而实例方法可以被类创建的任何对象调用执行。

因此如果类方法访问对象实例变量与方法时，可能实例方法并未分配入口地址，于是不允许访问。

8、请叙述在java中“按值传递”基本数据类型参数和对象数据类型参数的区别。

区别：

1）.当使用基本数据类型作为方法的形参时，在方法体中对形参的修改不会影响到实参的数值

2）.当使用引用数据类型作为方法的形参时，若在方法体中修改形参指向的数据内容，则会影响到实参的数值

**答案：在基本数据类型参数的传值中，基本数据类型的参数向该参数传递的值的级别不可以高于该参数的级别。在引用类型参数的传值中，Java的引用类型数据包括对象、数组和接口，当参数是引用类型时，“传值”传递的是变量的引用而不是变量所引用的实体。如果改变参数变量所引用的实体，就会导致原变量的实体发生同样的变化，但是如果改变参数的引用不会影响向其传值的变量的引用。**

9、请问如果通过对象数据类型的形参，在方法内部对形参所引用的实体进行改动，其那个在方法执行完毕后能保留下来吗？

答：能。

10、为什么this关键字不能出现在类方法中？

答：因为类方法属于静态方法，在程序加载时就已经有了入口地址，但是this指向的对象在程序加载时还没有入口地址，所以不能出现。（类方法可以通过类名调用，这个时候对象还没诞生，所以不能出现）

11、请叙述private访问权限和public访问权限的区别。

区别：

1、同一个类内，两类型都能访问。

2、同一个包内，不同的类，public可以访问，private不能访问。

3、不同包的继承类，public可以访问，private不能访问。

4、不同包的非继承类，public可以访问，private不能访问。

12、请问在内部类中，能定义静态成员变量吗？

答：不能。（从类名与加载时机分析）

13、请叙述在java中，包的命名惯例。

答：

包名全部由小写字母（多个单词也全部小写）；

* 如果包名包含多个层次，每个层次应该用“.”分割；
* 包名一般由倒置的域名开头，比如com.baidu；——不要有www

自定义包不能java开头；

14、Tomcat是一款著名的Servlet容器和Web服务器。它的开发站点域名为tomcat.apache.org。按照Java包的命名惯例，存放tomcat源代码的包应该叫什么名字？

答：org.apache.tomcat

15、import java.util.\* 和 import java.util.Scanner有啥区别？

答：

1、“importjava.util.\*;”表示的是把util这个包下的全部类导入到程序中；而“importjava.util.Scanner;”表示的是只把util包下的Scanner类导入到程序中。

2、导入“importjava.util.\*;”后程序中后面如果要用到util包里面的别的类都不需要再重新导入包了，而如果导入“importjava.util.Scanner;”的话，程序中后面如果要用到util包的别的类，都要再次导入这个类对应的包才能使用。

3、\*表示的是通配符，包括util的所有类，而Scanner表示的是一个具体的类。

16、什么是线程？什么是多线程？应用程序中的多线程有什么用？

答：

线程是比进程更小的执行单位。一个进程可以有多个线程。

多线程就是在单个程序中同时运行多个线程来完成不同的工作。

多线程用处：

1.使用线程可以把占据时间长的程序中的任务放到后台去处理

2.用户界面更加吸引人,这样比如用户点击了一个按钮去触发某件事件的处理,可以弹出一个进度条来显示处理的进度

3. 多线程可以使得整个环境能够异步执行，这有助于防止浪费CPU时钟周期从而提高效率。

4.在一些等待的任务实现上如用户输入,文件读取和网络收发数据等,线程就比较有用了.

17、java为线程机制提供了什么类与接口？

类：Thread类以及其子类

接口：Runnable接口

**答案：Runnable接口与Thread类，Runnable接口声明run（）方法，约定线程的执行方法；Thread类提供线程实例。调用Thread类的start（）可启动一个线程。**

18、编写一个线程，其任务是让一个字符串从屏幕左侧向右移动，当所有字符都消失后，字符串重新从左边出现并继续向右移动。

19、线程有哪5种基本状态？它们之间是如何转化的？

1、新建，当Thread类或其子类的对象被声明并创建时，新生的线程对象处于新建状态，此时已经有了相应的内存空间和其他资源。

2、运行，新建线程后，线程调用start（）方法通知JVM，则线程开始运行类中的run（）方法。

3、中断，线程在运行时，如果jvm将cpu资源切换给其他线程，或者线程执行了sleep（）方法，或者执行了wait()方法，或者进入某个阻塞状态，都会产生中断状态。

4、死亡，当线程运行完run（）方法后，其完成了他全部的工作或者提前强制性终止run（）方法，于是释放分配给线程对象的内存也叫作死亡。

**答案：线程有“新建”、“就绪”、“阻塞”、“运行”、“死亡”5个基本状态。线程对象被创建时进入“新建”状态；程序执行语句启动这个线程后，它进入线程队列排列等待CPU时间片，称为“就绪”状态；等CPU时间和其他资源齐备时就进入“运行”状态；线程运行完毕或者被强制终止则进入“死亡”状态。上述线程各状态之间的转换就构成了线程的基本生命周期。**

20、线程的方法sleep（）与wait（）有什么区别？

区别：

1.原理不同。sleep()方法是来自Thread类的静态方法，是线程用来控制自身流程的，它会使某个线程阻塞一段时间，等计时时间一到，此线程就会自动“苏醒”。而wait()方法是来自Object类的方法，用于线程间的通信，它也可以使线程阻塞，不过要等其他线程调用了notify()或者notifyAll()方法该线程才能苏醒。

2.对锁处理机制不同。sleep()方法的主要作用是让线程暂停一段时间，并不会释放掉自己持有的锁，而调用wait()方法的线程，会释放掉它所占有的锁。

3.使用区域不同。sleep()方法可以在任何代码块中的任何位置使用，而wait()方法必须要在同步方法或者同步块中使用。

4.是否捕获异常不同。sleep()必须要捕获异常，而wait()不需要捕获异常。

**答案：sleep方法和wait方法都可以用来放弃CPU一定的时间，不同点在于如果线程持有某个对象的监视器，sleep方法不会放弃这个对象的监视器，wait方法会放弃这个对象的监视器。**

21、什么是线程调度？java的线程调度采用什么策略？

答：

一般线程调度模式分为两种——抢占式调度和协同式调度。

抢占式调度指的是每条线程执行的时间、线程的切换都由系统控制，系统控制指的是在系统某种运行机制下，可能每条线程都分同样的执行时间片，也可能是某些线程执行的时间片较长，甚至某些线程得不到执行的时间片。在这种机制下，一个线程的堵塞不会导致整个进程堵塞。

协同式调度指某一线程执行完后主动通知系统切换到另一线程上执行，这种模式就像接力赛一样，一个人跑完自己的路程就把接力棒交接给下一个人，下个人继续往下跑。线程的执行时间由线程本身控制，线程切换可以预知，不存在多线程同步问题，但它有一个致命弱点：如果一个线程编写有问题，运行到一半就一直堵塞，那么可能导致整个系统崩溃。

**答案：安排线程队列中CPU时间的线程被称为线程调度。Java的线程调度采用优先级基础上的“先到先服务”策略。**