**1. 什么是Java虚拟机？为什么Java被称作是“平台无关的编程语言”？**

Java虚拟机是一种抽象的计算机，Java虚拟机在操作平台的基础上模拟实现计算机的部分功能，因此它是建立在计算机的操作系统上一种小型的具备计算机部分基本功能的计算机；Java虚拟机有自己独立的处理器、堆栈、寄存器等硬件架构，还有自己相应的指令系统，并且Java的JVM屏蔽了原机具体操作系统相关的信息和指令，因此Java程序的实现是在Java虚拟机的环境下实现的，而即便本机操作系统不一样，只要有Java虚拟机，就可以不加修改代码而在同一个计算机实现同一个Java程序，因此Java又被称作是与操作平台无关的语言。

**2. JDK和JRE的区别是什么？**

JRE全名Java Runtime Environment，顾名思义是Java程序运行时所需要的软件Java环境，因此JER是面向客户的，其中包含了Java虚拟机以及Java基础类库，便于用户运行编写好的Java程序。

JDK全名Java Development Kit，顾名思义是Java开发工具包，是编写Java程序时所需要的软件环境，是提供给编程者使用的；JDK包含了JRE，同时还包括了Java程序编译、调试和分析等工具，而JRE只能进行Java程序的执行。

综上：JDK和JRE的区别在于其面向的对象不同，JRE面向使用Java程序的用户，JDK面向编写Java程序的作者。

**3. Java支持的数据类型有哪些？什么是自动拆装箱？**

Java支持的数据类型有： int、byte、short、long、float、double、boolean、char。

Java是面向对象的编程语言，所以上述的基本类型都是一个对象，但如果直接把上述基本类型当做基本类型来处理会增加程序的复杂度和代码的长度；因此，为了解决这类问题，需要将上述的对象创建及其使用简化，引入自动拆、装箱可以解决这类问题。自动拆装箱：装即把基本的数据类型打包，拆即把基本数据类型对象的方法拆卸；使基本数据类型的创建和基本操作不用像对对象的操作一样复杂，简化了代码和程序。

**4. 接口和抽象类的区别是什么？**

抽象类是类，接口不是类；抽象类可以实现一个或多个接口，但是接口不能继承类或者抽象类；接口中的方法必须用抽象化的方法，即必须要adstract关键字修饰，而抽象类不用此关键字修饰；同时接口提供了一种Java中类同时继承几个类的方法。

**5. 什么是值传递和引用传递？**

值传递：实参把自己的值在调用方法时传递给方法的形参，即形参用实参的值初始化自己，所以值传递仅仅是吧实参的值传递给某方法，经过方法处理后，形参可能改变，但实参一定不变。

引用传递：实参把自己的存储单元的地址传递给形参，此时形参的值相当于实参的地址，因此方法对形参的操作即是对实参的操作，所以方法执行过后，形参和实参的变化是一样的，这一点类似于C\C++中的指针。

**6. Java中的两种异常类型是什么？他们有什么区别？**

不检查异常：运行时异常都是RuntimeException类及其子类,如 NullPointerException、IndexOutOfBoundsException等, 这些异常是不检查的异常, 是在程序运行的时候可能会发生的, 所以程序可以捕捉, 也可以不捕捉. 这些错误一般是由程序的逻辑错误引起的, 程序应该从逻辑角度去尽量避免，不检查异常不需要用户处理。

检查异常：检查异常是运行时异常以外的异常, 也是Exception及其子类, 这些异常从程序的角度来说是必须经过捕捉检查处理的, 否则不能通过编译. 如IOException、SQLException等，检查异常必须要用户处理。

**7. ArrayList和LinkedList有什么区别？**

ArrList需要预先估计数据的多少，而LinkList可以根据数据的多少动态的增加内存的开销；ArrList可以随机的访问指定位置的数据元素，而LinkList一般不能随机访问到指定位置的数据；ArrList在数据的中间插入或删除元素时，需要移动后续的其它元素，而LinkList的移动或删除的时间或空间的开销是固定的。

**8. Java I/O的体系结构是什么？**

I/O体系包括流式、非流式和其它类。流式包含字节流和字符流，字节流和字符流分别对应自己写（InputStream和Writer）和读（OutputStream和Reader）。非流式包括File类、RandomAccessFile类和FileDescriptor等类，来辅助完善流式的功能。其他类是文件读取部分的与安全相关的类，如：SerializablePermission类，以及与本地操作系统相关的文件系统的类，如：FileSystem类和Win32FileSystem类和WinNTFileSystem类。

**9. 如何确保N个线程可以访问N个资源同时又不导致死锁？**

死锁形成的四个必要条件分别是互斥条件、保持和请求条件、不可剥夺调教、循环等待条件。只要破坏其中任意一个条件，就可以避免死锁，但其中最简单最常用的的方法就是破环循环等待条件；即按同一顺序访问对象，加载锁，释放锁。这样确保了每个线程进行加锁取锁的顺序，即可达到N个线程可以访问N个资源而不导致死锁。

**10. 在Java中，对象什么时候可以被垃圾回收？**

在Java中，对象的销毁和回收不是实时的，当一个对象在所有Java程序内无任何引用时，会被Java中的JVM体系列为垃圾回收的对象，而JVM会定期对系统中存在的此类对象进行垃圾回收，不需要认为干预；这是在C++人为资源回收的一个提高或改进。