1.全局变量和局部变量有什么区别？实怎么实现的？操作系统和编译器是怎么知道的？

全局变量是整个程序都可访问的变量，谁都可以访问，生存期在整个程序从运行到结束(在   
程序结束时所占内存释放)，而局部变量存在于模块(子程序，函数)中，只有所在模块可以   
访问，其他模块不可直接访问，模块结束(函数调用完毕)，局部变量消失，所占据的内存   
释放。

2.String 和StringBuffer的区别

 String对项内容是不可改变的，StringBuffer是可以改变的，别且高效；

3.关键字break和continue的区别。

break用于强行退出循环，不执行循环中剩余的语句。 而continue则停止执行当前的操作，然后退回循环起始处，开始新的循环

4.静态变量和实例变量的区别？

静态变量也称作类变量，由static修饰，它只能通过类来访问。  
实例变量没有static修饰符，它只能通过实例对象来访问。  
同一类的不同的实例对象有自己的实例变量，但是它们共享同一个静态变量。当一个实例对象修改了它的实例变量时，不会影响其他的实例对象。如果一个实例对象修改了静态变量，则会影响其他的对象实例。

5.你所知道的集合类都有哪些？主要方法？

List,Set ,Map;  
List 与 Set 都是Collection接口的子接口；   
List 以特定次序来持有元素,可有重复元素.查找效率高,插入效率低.  
Set 无法拥有重复元素,内部排序,检索元素效率低,删除和插入效率高,因为插入与删除操作不会引起位置变化.  
Map并不继承自Collection接口；保存key-value值,value可多值.

6.List、Map、Set三个接口，存取元素时，各有什么特点？

List 以特定次序来持有元素,可有重复元素.查找效率高,插入效率低.  
Set 无法拥有重复元素,内部排序,查找效率低,删除和插入效率高,因为插入与删除操作不会引起位置变化.  
Map 具有键值对（key-value）的存储方式，一个键对应一个值，键必须是唯一的。

7.int 和 Integer 有什么区别

int 是基本数据类型，而Integer是int对应的包装器类，该类中提供了各种将int类型的数据与其他类型数据进行转换的方法

8.Collection 和 Collections的区别。

Collection接口是集合类的鼻祖，包含了集合操作的基本方法，其下有List、Set等接口，  
List下有 ArrayList、 LinkedList、Stack等实现类  
Set下有HashSet、TreeSet、LinkedHashSet等  
Collections是针对集合类的一个帮助类，它提供了一系列静态方法实现了对各种集合的排序，搜索和线程安全等操作。

9.数组有没有length()这个方法? String有没有length()这个方法？

数组没有length()方法，但有length属性,String有length()方法。

10.ArrayList和Vector的区别,HashMap和Hashtable的区别

ArrayList和Vector的区别:  
一、同步性:Vector是线程安全的，也就是说是同步的，而ArrayList是线程不安全的，不是同步的  
二、数据增长:当需要增长时,Vector默认增长为原来一培，而ArrayList却是原来的一半

HashMap和Hashtable的区别:  
都属于Map接口的类，实现了将惟一键映射到特定的值上。  
HashMap 类没有分类或者排序。它允许一个 null 键和多个 null 值。  
Hashtable 类似于 HashMap，但是不允许 null 键和 null 值。它也比 HashMap 慢，因为它是同步的。

11.说出ArrayList,Vector, LinkedList的存储性能和特性

ArrayList和Vector都是用数组方式存储数据,此数组元素数要大于实际的存储空间以便进行元素增加和插入操作,他们都允许直接用序号索引元素,但是插入数据元素涉及到元素移动等内存操作,所以索引数据块而插入数据慢.Vector使用了sychronized方法(线程安全),所以在性能上比ArrayList要差些.而LinkedList使用双向链表方式存储数据,按序号索引数据需要前向或后向遍历数据,但是插入数据时只需要记录前后项即可,所以插入的速度快.

12.抽象类和接口的区别，什么时候用？

简单来说，   
接口是公开的，里面不能有私有的方法或变量，是用于让别人使用的，而抽象类是可以有私有方法或私有变量的，   
另外，实现接口的一定要实现接口里定义的所有方法，而实现抽象类可以有选择地重写需要用到的方法，一般的应用里，最顶级的是接口，然后是抽象类实现接口，最后才到具体类实现。   
还有，接口可以实现多重继承，而一个类只能继承一个超类，但可以通过继承多个接口实现多重继承，接口还有标识（里面没有任何方法，如Remote接口）和数据共享（里面的变量全是常量）的作用

13.面向对象的基本特征有哪些方面

1. 抽象：　  
   　抽象就是忽略一个主题中与当前目标无关的那些方面，以便更充分地注意与当前目标有关的方面。抽象并不打算了解全部问题，而只是选择其中的一部分，暂时不用部分细节。抽象包括两个方面，一是过程抽象，二是数据抽象。  
   2.继承：  
   　继承是一种联结类的层次模型，并且允许和鼓励类的重用，它提供了一种明确表述共性的方法。对象的一个新类可以从现有的类中派生，这个过程称为类继承。新类继承了原始类的特性，新类称为原始类的派生类（子类），而原始类称为新类的基类（父类）。派生类可以从它的基类那里继承方法和实例变量，并且类可以修改或增加新的方法使之更适合特殊的需要。  
   3.封装：  
   封装是把过程和数据包围起来，对数据的访问只能通过已定义的界面。面向对象计算始于这个基本概念，即现实世界可以被描绘成一系列完全自治、封装的对象，这些对象通过一个受保护的接口访问其他对象。  
   4. 多态性：  
   　多态性是指允许不同类的对象对同一消息作出响应。多态性包括参数化多态性和包含多态性。多态性语言具有灵活、抽象、行为共享、代码共享的优势，很好的解决了应用程序函数同名问题。

14.final, finally, finalize的区别。

**final 用于声明属性，方法和类，分别表示属性不可变，方法不可覆盖，类不可继承。**

**finally是异常处理语句结构的一部分，表示总是执行。**

**finalize是Object类的一个方法，在垃圾收集器执行的时候会调用被回收对象的此方法，**

**可以覆盖此方法提供垃圾收集时的其他资源回收，例如关闭文件等。**

15.GC是什么? 为什么要有GC? (基础)。垃圾回收器可以马上回收内存吗？有什么办法主动通知虚拟机进行垃圾回收？

GC是垃圾收集器。  
  
当一个对象的引用丢失后，即没有地方再引用了该对象，这个对象将成为垃圾。但这样的对象仍在内存中，占据内存资源，虚拟机会按一定的方式启动垃圾收集器来回收这些对象占据的内存。  
如果想通过程序去回收内存，可以通过System.gc()或Runtime.getRuntime().gc()方法通知虚拟机进行垃圾回收，但并不一定马上就会执行垃圾回收，也就是不能强制立即执行垃圾回收，只能通知垃圾回收器有空的话赶紧去收拾一下。

16.Overload和Override的区别。Overload的方法是否可以改变返回值的类型?

方法的重写Overriding和重载Overloading是Java多态性的不同表现。  
重写Overriding是父类与子类之间多态性的一种表现，重载Overloading是一个类中多态性的一种表现。

如果在子类中定义某方法与其父类有相同的名称和参数，我们说该方法被重写 (Overriding)。子类的对象使用这个方法时，将调用子类中的定义，对它而言，父类中的定义如同被“屏蔽”了。

如果在一个类中定义了多个同名的方法，它们或有不同的参数个数或有不同的参数类型，则称为方法的重载(Overloading)。Overloaded的方法是可以改变返回值的类型。

17.error和exception有什么区别?

error 表示恢复不是不可能,但很困难的情况下的一种严重问题。比如说内存溢,网络故障等。不可能指望程序能处理的情况。   
  
exception 表示一种设计或实现问题。也就是说，它表示如果程序运行正常，就不会发生的情况。程序可以捕获到这些异常，并进行处理。

18.运行时异常与一般异常有何异同

一般异常表示程序运行过程中可能出现的非正常状态,如读取文件时找不到文件，这类异常必须捕获  
  
运行时异常表示虚拟机的通常操作中可能遇到的异常，是一种常见运行错误。如空指针、除以零，这里异常可以不需要捕获

19.说出一些常用的类，包,接口，请各举5个

常用包：  
java.lang、java.util、java.io、java.sql、java.net、javax.swing、java.text  
  
常用类：  
StringBuffer、Math、Random、InputStream、OutputStream、ObjectInputSteam、BufferedReader、SimpleDateFormat、NumberFormat、Socket、ServerSocket、Connection、Statement、ResultSet  
  
常用接口：  
Cloneable、Runnable、Serializable、List、Set、DataSource

20.接口和内部类、抽象类的特性

接口的特性：   
不是类，不能用new运算符实例化一个接口。   
虽然不能构造接口对象，却能声明接口变量。   
接口变量必须引用实现了接口的类对象。   
可以使用instanceof检查一个对象是否实现了某个接口。   
接口可以扩展。即可以interface1 extends interface2。   
接口可以包含常量。   
接口中的方法和域被自动设为public static final   
一个类，可以实现多个接口。   
  
内部类的特性：   
内部类方法可以访问该类定义所在的作用域中的数据，包括私有数据。   
内部类可以对同一个包中的其实类隐藏起来。   
匿名内部类比较便捷。   
只有内部类可以是私有类   
  
抽象类的特性：   
抽象类可以没有抽象方法。   
包含一个或多个抽象方法的类，必须被声明为抽象的。   
抽象类可以包含具数据我具体方法。   
同interface：抽象类不能被实例化，但可以定义一个抽象类的变量，这个变量只能引用非抽象了类的对象。   
与interface不同，一个类只能扩展一个抽象类。

**21.处理字符串时，操作符==和equal()方法的区别**

字符串属于对象；  
在对象比较中= = 比较的是内存对象，即八种基本数据类型，也就是是不是同一个对象；  
而 equals比较的 内容是否相等

**22.是否可以从一个static方法内部发出对非static方法的调用？**

不可以.  
  
从类的加载顺序上来说,静态方法或静态变量是最先被加载到内存的，此时内存中还没有非静态的方法,所以无法调用;  
  
使用静态方法时候,要注意以下几点:  
a. 在静态方法中职能直接调用同类中的其他静态方法或静态变量,而不能直接方法非静态的方法或变量;  
b.静态方法不能以任何方法引用this或supper关键字

23.线程有哪几个基本状态？它们之间如何转化？简述线程的生命周期。

新建状态，可运行状态，运行状态，阻碍状态，终止状态。对线程调用各种控制方法，就使线程从一种状态转换到另一种状态。线程的生命周期从新建开始，在可运行、运行和其他阻碍中循环，在可运行、运行、对象锁阻塞、等待阻塞中循环，最终在运行后run()方法结束后终止。

24.如何在Java程序中实现多线程？

1：通过Thread类的构造方法

2：通过实现Runnable接口(implements)创建线程、

3：通过继承(extends )Thread类创建线程

25.分别写出2个你熟悉字节流，字符类，包装流的类名称  
字节流：InputStream,OutputStream

字符流：Reader,writer

包装流：DataInputStream,DataOutputStream

26.Java数据库操作基本流程：

取得数据库连接 、执行sql语句、处理执行结果、释放数据库连接

1. 通过驱动程序管理器取得连接接口，注册一个驱动
2. 建立一个数据库的连接Connection
3. 创建一个陈述对象Statement
4. 通过陈述对象执行sql语句，并返回结果集ResultSet
5. 处理该结果集
6. 关闭Statement
7. 关闭连接接口

27.简述基于TCP及UDP套接字通信的主要区别

**1.TCP:可靠,传输大小无限制,但是需要时间来建立连接,差错控制开销大；主要用于文本、程序等可靠性高的数据传输**

**2.UDP:不可靠,传输大小限制在64K以下,不需要建立连接,差错控制开销小；主要用于类似图像、声音、网络游戏、视频会议等等对可靠性要求没有那么高的业务进行传输**

28.介绍一下DatagramPacket的常用构造方法

　　1.　**DatagramPacket( byte buf [ ] , int length)**

**2.DatagramPacket( byte buf [ ] , int length , InetAddress addr, int port)**

3. **DatagramPacket( byte buf [ ] , int offset , int length)**

4. **DatagramPacket( byte buf [ ] , int offset, int length , InetAddress addr , int port)**

其中，buf中存放数据报数据，length为数据报中数据的长度，addr和port旨明目的地址，offset指明了数据报的位移量。

29.请说明try、catch、throw、throws和finally各自的作用?

try : 用于监控可能发生异常的代码块

catch : 当try中的代码块发生异常，就交给catch处理，try和catch是配对使用

throw ： 手动引发异常

throws ： 抛出异常，当我们不想处理某个异常时，可以将它往外抛。

finally ： 不管异常发生与否，finally都会执行，通过用该代码块处理资源释放。

30.java跨平台的原理?

Java编译器可以把源程序转换为字节码文件，通过Java虚拟机运行，而且Java虚拟机可以安装在不同的操作系统上，而且Java程序具有一次编译，到处运行的原理

31.TreeSet和hashSet的区别？

TreeSet基于平衡数的方式存放数据，HashSet基于列表式的方式存放数据.在选择的时候，如果顺序重要，则可以选择TreeSet，操作性能和时间效率很重要的话，则选择HashSet。

32.Java单例模式详解？

单例模式分为三种：懒汉式单例，饿汉式单例，登记式单例。

特点：单例类只有一个实例

单例类必须自己创建自己的唯一实例

单例类必须给所有其他对象提供这一实例

定义一个私有的构造函数

定义一个类本身的静态变量

创建一个实例化对象的函数getInstance

33.线程安全工作机制原理？

如果你的代码所在的进程中有多个线程在同时运行，而这些线程可能会同时运行这段代码。如果每次运行结果和单线程运行结果是一样的，而且其他的变量的值也和预期的是一样的，就是线程安全的。

34.线程的同步，如何实现线程的同步？

当多个线程共享一个资源时，可能会产生线程安全问题，需要通过线程安全问题来解决。

同步就是使用一个同步对象来标识并发访问同一资源的代码，需要进入此代码，必须获得同步对象。同步对象可以理解为锁，要进入同步代码，必须获得同步对象。一个线程获得同步对象进入同步代码后，其他线程无法获得同步对象而无法进入，等执行完同步代码后，该线程自动让出同步对象，其他线程就可以抢占同步对象，谁抢到谁就可以进入，一般线程优先级高，等待时间长的线程容易抢到。

35.什么是Java序列化，如何实现Java序列化？

序列化就是一种用来处理对象流的机制，所谓对象流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。序列化是为了解决在对对象流进行读写操作的是所以引发的问题。

36.swing模型中由哪三部分组成，工作原理是什么？

事件对象，事件源，事件监听器。原理：当一个事件发生时，如点击鼠标、点击按钮、在文本区输入、系统会将事件的相关信息封装成事件对象，并将它送到一个至多个监听器那里。在这种方案中，监听器简单的等待，直到它收到一个事件。一旦事件被接受，监听器将处理这个事件，处理完后返回。

37. 简述程序、进程和线程之间的关系？什么是多线程程序？

程序是一段静态的代码，它是应用软件执行的蓝本。进程是程序的一次动态执行过程，它对应了从代码加载、执行到执行完毕的一个完整过程。这个过程也是进程本身从产生、发展、到消亡的过程。线程是比进程更小的单位。一个进程在其执行过程中，可以产生多个线程，形成多个执行流。每个执行流即每个线程也有它自身的产生、存在和消亡的过程，也是一个动态的概念。多线程程序是指一个程序中包含多个执行流。  
38.Vector与TreeSet在存储结构和存取性能上的区别

TreeSet是基于平衡数的方式存放数据，在选择的时候，如果顺序重要，则可以选择TreeSet，它可以对Set集合中的元素进行排序（自然循序）。Vector是线程同步的，安全的，默认增长为原来的一倍。

39.什么是线程安全问题？什么情况下容易产生线程安全问题？线程安全问题如何解决？

1，如果你的代码所在的进程中有多个线程在同时运行，而这些线程可能会同时运行这段代码。如果每次运行结果和单线程运行结果是一样的，而且其他的变量里的值也和预期是一样的，就是线程安全的。

,2，当两个线程在争夺同一个资源时，容易产生互斥锁，这就产生了线程安全问题。

3. 可以使用同步代码块去解决，或者使用同步函数（同步函数就是使用synchronized修饰一个函数 ） 来解决

40. 什么是垃圾，java垃圾回收的工作原理

当一个对象的引用丢失后，即没有地方再引用了该对象，这个对象将成为垃圾。但这样的对象仍在内存中，占据内存资源，虚拟机会按一定的方式启动垃圾收集器来回收这些对象占据的内存。

如果想通过程序去回收内存，可以通过系统的垃圾处理器（System.gc）或Runtime.getRuntime().gc()方法通知虚拟机进行垃圾回收，但并不一定马上就会执行垃圾回收，也就是不能强制立即执行垃圾回收，只能通知垃圾回收器有空的话赶紧去收拾一下。