<http://www.cnblogs.com/DreamSea/archive/2012/01/11/JavaThread.html>

线程：

线程是一个程序的多个执行路径，执行调度的单位，依托于进程存在。 线程不仅可以共享进程的内存，而且还拥有一个属于自己的内存空间，这段内存空间也叫做线程栈，是在建立线程时由系统分配的，主要用来保存线程内部所使用的数据，如线程执行函数中所定义的变量。

主线程 子线程

<http://www.360doc.com/content/12/1121/20/1317564_249389214.shtml>

子线程一般是继承Thread、实现Runnable、利用Timer对象等实现的，而UI线程可以说是主线程

最开始的线程就是主线程，子线程就是被别的线程建立的线程。

所谓主线程，在Windows窗体应用程序中一般指[UI](https://www.baidu.com/s?wd=UI&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YkPyDYPyfzujF9rH7buhuW0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3En1nsPWRdPHRk)线程，这个是程序启动的时候首先创建的线程。而子线程，一般指为了完成某个特殊任务，并行于主线程的其他线程。例如让一个窗体单独开一个线程，去远程数据库中取得数据资料，下载并且保存到本地文件中。  
所以，主线程和子线程是构建于线程这个通用概念上的人为的习惯称呼，并不是什么官方的固有专用词语。

1. java.lang.Thread类

<http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/class-use/Thread.html>

二、实现java.lang.Runnable接口

Java不允许一个类继承多个类，因此，一旦Monkey类继承了Thread类，就不能再继承其他的类。为了解决这一问题，Java提供了java.lang.Runnable接口，它有一个run()方法

# java.lang.Thread 和 java.lang.Runnable的区别

thread是类  
runnable是接口  
  
实现多线程有两种方式：  
1.继承Thread，然后重写他的run方法  
2.实现Runnable接口，并实现他的run方法  
  
启动线程时也有区别  
继承Thread的类直接调用start方法即可  
实现Runnable接口要这样启动  
runab ：是实现Runnable接口类的对象  
Thread thr = new Thread(runab);   
thr.start()

三、Thread和Runnable的区别

在java中可有两种方式实现多线程，一种是继承Thread类，一种是实现Runnable接口；Thread类是在java.lang包中定义的。一个类只要继承了Thread类同时覆写了本类中的run()方法就可以实现多线程操作了，但是一个类只能继承一个父类，这是此方法的局限

继承Thread：

结果很有规律，先第一个对象执行，然后第二个对象执行，并没有相互运行。在JDK的文档中可以发现，一旦调用start()方法，则会通过JVM找到run()方法。

实现Runnable:

在实际开发中一个多线程的操作很少使用Thread类，而是通过Runnable接口完成。

Runnable定义的子类中没有start()方法，只有Thread类中才有,

**两种实现方式的区别和联系：**

在程序开发中只要是多线程肯定永远以实现Runnable接口为主，因为实现Runnable接口相比继承Thread类有如下好处：

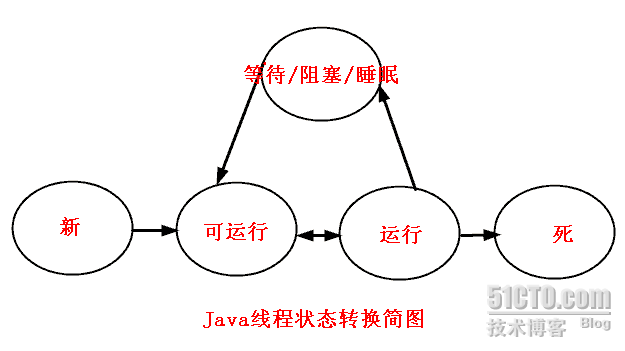
* 避免点继承的局限，一个类可以继承多个接口。
* 适合于资源的共享

public class Thread extends Object implements Runnable

发现Thread类也是Runnable接口的子类。

四、线程状态转换 （ <http://lavasoft.blog.51cto.com/62575/99153/>）

线程的状态转换是线程控制的基础。线程状态总的可分为五大状态：分别是生、死、可运行、运行、等待/阻塞。



1、新状态：线程对象已经创建，还没有在其上调用start()方法。

2、可运行状态：当线程有资格运行，但调度程序还没有把它选定为运行线程时线程所处的状态。当start()方法调用时，线程首先进入可运行状态。在线程运行之后或者从阻塞、等待或睡眠状态回来后，也返回到可运行状态。

3、运行状态：线程调度程序从可运行池中选择一个线程作为当前线程时线程所处的状态。这也是线程进入运行状态的唯一一种方式。

4、等待/阻塞/睡眠状态：这是线程有资格运行时它所处的状态。实际上这个三状态组合为一种，其共同点是：线程仍旧是活的，但是当前没有条件运行。换句话说，它是可运行的，但是如果某件事件出现，他可能返回到可运行状态。

5、死亡态：当线程的run()方法完成时就认为它死去。这个线程对象也许是活的，但是，它已经不是一个单独执行的线程。线程一旦死亡，就不能复生。 如果在一个死去的线程上调用start()方法，会抛出java.lang.IllegalThreadStateException异常。

五、线程调度

Java线程：线程的调度-休眠

Java线程调度是Java多线程的核心，只有良好的调度，才能充分发挥系统的性能，提高程序的执行效率。

这里要明确的一点，不管程序员怎么编写调度，只能最大限度的影响线程执行的次序，而不能做到精准控制

线程休眠的目的是使线程让出CPU的最简单的做法之一，线程休眠时候，会将CPU资源交给其他线程，以便能轮换执行，当休眠一定时间后，线程会苏醒，进入准备状态等待执行。

线程休眠的方法是Thread.sleep(long millis) 和Thread.sleep(long millis, int nanos) ，均为静态方法，那调用sleep休眠的哪个线程呢？简单说，哪个线程调用sleep，就休眠哪个线程

六、常用函数说明

**函数的定义及特点  
1) 什么是函数？**函数就是定义在类中的具有特定功能的一段独立小程序，函数也称为方法  
**2)Java中函数的格式：**修饰符 返回值类型函数名（参数类型 形式参数1，参数类型 形式参数2，。。）  
｛  执行语句;  
 return 返回值；  
｝  
返回值类型：函数运行后的结果的数据类型  
参数类型：是形式参数的数据类型  
形式参数：是一个变量，用于存储调用函数时传递给函数的实际参数  
实际参数：传递给形式参数的具体数值  
return：用于结束函数  
返回值：该值会返回给调用者  
**3) 函数的特点**a) 定义函数可以将功能代码进行封装  
b) 便于对该功能进行复用  
c) 函数只有被调用才会被执行  
d) 函数的出现提高了代码的复用性  
e) 对于函数没有具体返回值的情况，返回值类型用关键字void表示，那么该函数中的return语句如果在最后一行可以省略不写，系统会帮你自动加上。  
**注：**a) 函数中只能调用函数，不可以在函数内部定义函数。  
b) 定义函数时，函数的结果应该返回给调用者，交由调用者处理。  
c) 当函数运算后，没有具体的返回值时，这是返回值类型用一个特殊的关键字来标识该关键字就是void，void：代表的是函数没有具体返回值的情况。  
d)  当函数的返回值类型是void时，函数中的return语句可以省略不写。  
**4)  如何定义一个函数？**    函数其实就是一个功能，定义函数就是实现功能，通过两个明确来完成：  
       1）、明确该功能的运算完的结果，其实是在明确这个函数的返回值类型。  
       2）、在实现该功能的过程中是否有未知内容参与了运算，其实就是在明确这个函数的参数列表(参数类型&参数个数)。  
**5) 函数的作用：**1）、用于定义功能。  
2）、用于封装代码提高代码的复用性。  
注意：函数中只能调用函数，不能定义函数。  
**6) 主函数：**    1）、保证该类的独立运行。  
    2）、因为它是程序的入口。  
    3）、因为它在被jvm调用。  
**7) 函数定义名称是为什么呢？**答：1）、为了对该功能进行标示，方便于调用。  
    2）、为了通过名称就可以明确函数的功能，为了增加代码的阅读性。

**函数的应用**1)  两个明确  
a)  明确要定义的功能最后的结果是什么？  
b)  明确在定义该功能的过程中，是否需要未知内容参与运算

明确函数定义的格式。  
    /\*  
    修饰符返回值类型函数名(参数类型形式参数1，参数类型形式参数2，)  
    {  
           执行语句;  
           return 返回值;  
    }

**函数的重载（overload）**重载的概念：  
在同一个类中，允许存在一个以上的同名函数，只要它们的参数个数或者参数类型不同即可。  
重载的特点：  
与返回值类型无关，只看参数列表。  
重载的好处：  
方便于阅读，优化了程序设计。  
重载示例：  
返回两个整数的和  
int add(int x,int y){return x+y;}  
返回三个整数的和  
int add（int x,int y, int z）{return x+y+z;}  
返回两个小数的和  
double add(double x,double y){return x+y;}  
什么时候用重载？  
当定义的功能相同，但参与运算的未知内容不同。  
那么，这时就定义一个函数名称以表示起功能，方便阅读，而通过参数列表的不同来区分多个同名函数。

区分是否是重载

void show(int a,char b,double c){}  
下面的a,b,c,d,e,f与上述一句的区别：  
a.  
void show(int x,char y,double z){}//没有，因为和原函数一样。  
b.  
int show(int a,double c,char b){}//重载，因为参数类型不同。注意：重载和返回值类型没关系。  
c.  
void show(int a,double c,char b){}//重载，因为参数类型不同。注意：重载和返回值类型没关系。  
d.  
boolean show(int c,char b){}//重载了，因为参数个数不同。  
e.  
void show(double c){}//重载了，因为参数个数不同。  
f.  
double show(int x,char y,double z){}//没有，这个函数不可以和给定函数同时存在与一个类中。

线程函数

<http://blog.chinaunix.net/uid-26746982-id-3388663.html>

线程基本函数

http://wenku.baidu.com/link?url=T\_oV5G8Km-6Wd2tticbu4\_vTcKvATtiBHhCwLvR-EjpGZbsjOhqPBB2XK9WT\_MLnYQju5usHXbrepKiKRxzg3U1BwMqR2hEvAFy8sQjtw6G

七、常见线程名词解释  
八、线程同步  
九、线程数据传递  
十、handler  
十一、Activity.runOnUIThread(Runnable)  
十二、View.Post(Runnable)  
十三、View.PostDelayed(Runnabe,long)  
十四、AsyncTask