# 扩展java.lang.Thread类

## 注意

1、start()方法调用后不会立即执行多线程代码，而是使得该线程变为可运行态。什么时候运行由操作系统决定。

2、start()方法重复调用会出现java.lang. IllegalThreadStateException异常

3、main也是一个线程。两个线程：垃圾回收和main

# 实现java.lang.Runnable接口

## 优势

1、适合多个相同的程序代码的线程去处理同一个资源

2、可以避免单继承

3、线程池只能放实现Runnable或callable类的线程

# 线程

## 状态转换



1、新建状态：new

2、就绪状态：start()

3、运行态：获取CPU，执行代码

4、阻塞状态：放弃CPU，暂时停止运行

（一）等待同步：wait（等待池、释放锁）

（二）同步阻塞：同步锁被占用（放入锁池）

（三）其他阻塞sleep()、join()、I/O请求。（不释放持有锁）

## 线程调度

1、线程优先级

2、线程睡眠：Thread.sleep(1000)。允许较低优先级的线程获得运行机会

3、线程等待：Object类中的wait()（会释放占有的锁）。直到其他线程调用notify和notifyAll（相当于wait(0)，释放锁），且synchronized(){}块结束。

4、线程让步：Thread.yield()。让当前线程回到可运行状态，以允许具有相同优先级的其他线程获得运行机会（可能又是当前线程抢到）

5、线程加入：join()。主线程需要等待子线程的终止

6、线程唤醒：notify()

# 各种线程概念

1、主线程：main

2、当前线程：Thread.currentThread()

3、后台线程：也称守护线程，是否随主线程的结束而结束（GC）

4、前台线程：接受后天线程服务的线程