1. 暂停线程

（一） 使用方法

可以使用suspend()方法来暂停线程，使用resume()方法来恢复线程

MyThread thread = new MyThread();

thread.start();

thread.suspend();

Thread.sleep(1000);

thread.resume();

（二） 缺点分析

suspend()暂停线程，如果使用不当时，很容易造成公共的同步对象的独占，使得其他线程无法访问公共的同步对象。同时，也会出现数据不同步的情况。

1. yield方法

yield()方法的作用是放弃当前的CPU资源，让其他的任务占用CPU的执行时间，但是这种放弃不确定的，即可能刚放弃，又获得了时间片。

1. 线程划分优先级
2. 概述

线程是有优先级别的，优先级别高的线程能够得到更多的CPU资源，能够优先执行任务。但是这种优先执行任务并不一定表示高优先级别任务就会先完成，CPU的资源供给还是存在着随机性。高优先级的线程大部分会优先执行，但是少部分依然是延后执行的。

1. 优先级的继承特性

如A线程启动B线程，那么B线程的优先级与A线程是一样的。

1. 线程优先级

在Java中，线程优先级一共划分为了1-10这10个等级，如果小于1或者大于10，则JDK会跑出 IllegalArgumentException异常。

1. 设置优先级

setPriority()方法能够设置线程的优先级，如下所示：

thread.setPriority(6)

1. 获取优先级

getPriority()方法能够获取线程的优先级，如下所示：

thread.getPriority();

1. 守护线程
2. 概述

Java中有两种线程，一种是用户线程，另一种是守护进程。守护进程用于守护着非守护线程，即只要有非守护线程存在，那么守护线程就会一直存在着，因此守护线程的作用就是为非守护线程的运行提供便利服务的，最典型的应用即是垃圾回收期（GC）。

1. 开启守护线程

开启一个线程，通过setDaemon(true)方法设置该线程为守护线程即可，

thread.setDaemon(true);