中断

让线程提前结束

1.stop 强制性终止 这个是没法控制。

2.发出interrupted信号 注意，这个信号的接受线程，不能处于睡眠等阻塞状态

中断的是正在抢夺时间分片的线程

3.通过设置标志位 可以使用等待的阻塞的状态

# 优先级

就是给线程的时间分片多少问题，跟操作系统有关。

# 线程类型

精灵 游戏 小精灵

英雄带着一个小精灵

守护精灵 守护进程

把当前定义的线程作为一个精灵了，守护线程

必须有一个英雄 主体线程

在线程运行的时候，精灵就存在，当线程结束了，精灵就不工作，结束了。

守护线程

守护用户线程的。

# 线程同步

主要作用就是同时操作，并不是同步，而是异步的，并发的，不是串行的。

为何又需要同步呢？

其中有一个数据共享

数据 每个线程都会操作的数据，共享数据

data=100

t1 -- 99 t2-- 98 t3-- 97

超卖的情况

直白描述 给你时间分片 睡觉，浪费了。

优化过程 睡觉了，让出使用cpu权利。立马释放，把资源给被的线程。如果cpu又来了，发现你还在睡觉，还是释放资源。给别的线程。

多线程对共享数据的操作，但是这个数据在操作 到时候可能有误差，所以对每个线程操作这个数据的时候，需要同步处理，当一个线程进来之后，其他线程不能访问这个资源，等到这个线程中的同步指令结束了，别的线程才有机会得到这个资源进行操作。

同步，就是加了一把锁，互斥锁

锁 是对象上的锁

对象锁

每个对象都有一把锁

阻塞 sleep 睡一会会自己醒的。

但是，对象锁 ，其它任何线程都进入阻塞。

释放锁的时候

如果不释放了呢？永远进入阻塞

死锁 永远进入阻塞

都是进入阻塞，前者是当前线程阻塞，后者是让别的线程阻塞。

悲观锁，乐观锁

没法控制这把锁，是jvm控制，代码块结束，释放锁。如果遇到特殊情况，如果发生异常，处理一一直等待等。永久锁住

乐观锁 ，可以通过程序任意控制，枷锁，解锁

数据库中也有悲观锁和乐观锁

默认情况，如果使用

乐观锁，不管位置，拿这个锁也好，都可以通过程序控制