**多线程**

**实现多线程的两种方式**

1. **继承Thread类**

A: 自定义类继承Thread类

B:在自定义类中重写run()方法

C:创建自定义类对象

D:启动线程对象start()

1. **自定义类实现Runnable接口（常用）**

A: 自定义类实现该接口

B:在自定义类中重写run()方法

C:创建自定义类对象my

D: 创建Thread对象th,有参构造Thread th=new Thread(my)

E:启动线程对象start()

**获取线程对象的名称**

public final String getName() 获取线程的名称

public static Thread currentThread() 返回当前正在执行的线程

**设置线程对象的名称**

public final void setName(String name) 设置线程的名称

**线程休眠**

public **static** void sleep(long millis) 不释放锁，wait()方法释放锁

**线程加入**

public final void join() 该线程先执行，在启动之后设置

**线程礼让**

public **static** void yield() 暂停当前正在执行的线程对象，并执行其他线程

**守护线程**

public final void setDaemon(boolean on) 当on为true时该线程为守护线程，当on为false时该线程为用户线程（默认为false）。当用户线程结束时，守护线程只能执行一会就结束，在启动之前设置

**中断线程**

public void interrupt() 把线程的状态终止，并抛出一个InterruptedException异常（常用于结束线程的休眠）

**同步解决线程安全问题**

**同步代码块**

synchronized(锁对象){} 锁对象可以是任何对象

**同步方法public synchronized void**

锁对象是this

**静态同步方法**

锁对象是当前类的字节码文件对象(类名.class)

**等待唤醒机制**

**Object类的三个方法：通过锁对象调用**

wait() 线程等待，退出同步，进入阻塞状态，唤醒后从中断处继续执行

notify() 唤醒单个线程

notifyAll() 唤醒所有线程