Java多线程

什么是线程？

主要内容

- 什么是多线程
- 线程的创建
- 线程的状态和生命周期
- 线程调度
- 同步与死锁

线程的创建

- 创建一个Thread类，或者一个Thread子类的对象
- 创建一个实现Runnable接口的类的对象

Thread类

- **Thread是一个线程类，位于java.lang包下**

构造方法	说明
Thread()	创建一个线程对象
Thread(String name)	创建一个具有指定名称的线程对象
Thread(Runnable target)	创建一个基于Runnable接口实现类的线程对象
Thread(Runnable target,String name)	创建一个基于Runnable接口实现类，并且具有指定名称的线程对象。

Thread类

- **Thread类的常用方法**

方法	说明
public void run()	线程相关的代码写在该方法中，一般需要重写
public void start()	启动线程的方法
public static void sleep(long m)	线程休眠m毫秒的方法
public void join()	优先执行调用join()方法的线程

Runnable接口

- 只有一个方法run();
- Runnable是Java中用以实现线程的接口，
- 任何实现线程功能的类都必须实现该接口

线程创建

- 通过继承Thread类的方式创建线程类，重写run()方法。

线程创建

- 通过实现Runnable接口的方式创建。

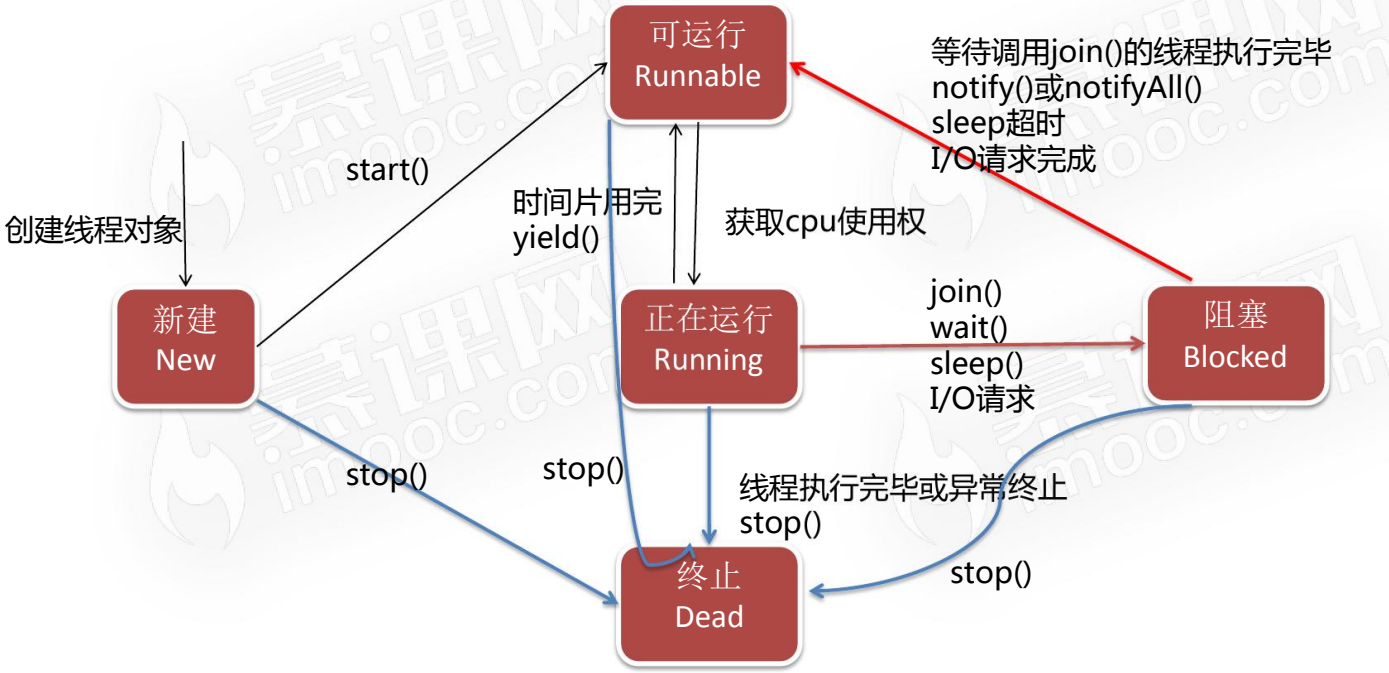
线程创建

- 为什么要实现Runnable接口？
  - Java不支持多继承
  - 不打算重写Thread类的其他方法

线程的状态

- 新建（New）
- 可运行（Runnable）
- 正在运行（Running）
- 阻塞（Blocked）
- 终止（Dead）

线程的生命周期



sleep方法应用

- Thread类的方法

```
public static void sleep(long millis)
```

- 作用：在指定的毫秒数内让正在执行的线程休眠（暂停执行）
- 参数为休眠的时间，单位是毫秒

join方法应用

- Thread类的方法

```
public final void join()
```

- 作用：等待调用该方法的线程结束后才能执行

join方法应用

- Thread类的方法

```
public final void join(long millis)
```

- 作用：等待该线程终止的最长时间为millis毫秒。
- 如果millis为0则意味着要一直等下去。

线程优先级

- Java为线程类提供了10个优先级
- 优先级可以用整数1-10表示，超过范围会抛出异常
- 主线程默认优先级为5

线程优先级

- 优先级常量
  - MAX_PRIORITY：线程的最高优先级10
  - MIN_PRIORITY：线程的最低优先级1
  - NORM_PRIORITY：线程的默认优先级5

线程优先级

- 优先级相关的方法

方法	说明
public int getPriority()	获取线程优先级的方法
public void setPriority(int newPriority)	设置线程优先级的方法

多线程运行问题

- 各个线程是通过竞争CPU时间而获得运行机会的
- 各线程什么时候得到CPU时间，占用多久，是不可预测的
- 一个正在运行着的线程在什么地方被暂停是不确定的

银行存取款问题

- 为了保证在存款或取款的时候，不允许其他线程对帐户余额进行操作
- 需要将Bank对象进行锁定
- 使用关键字synchronized实现

同步

- synchronized关键字用在
  - 成员方法
  - 静态方法
  - 语句块

```
public synchronized void saveAccount(){}
 public static synchronized void saveAccount(){}
  synchronized (obj){……}
```

线程间通信

- 问题：帐户余额不够了怎么办？
- 等待存入足够的钱后处理

线程间通信

- wait()方法：中断方法的执行，使线程等待
- notify()方法：唤醒处于等待的某一个线程，使其结束等待
- nofifyAll()方法：唤醒所有处于等待的线程，使它们结束等待

总结

- 线程的概念
- 创建线程的两种方式
- 线程的状态和生命周期
- sleep()和join()方法
- 线程的优先级
- 同步
- 线程间通信