第13章 多线程

主要内容

1. Java中的多线程实现技术
2. 多线程的管理

多线程

- 多线程

  程序中同时存在几个执行体，按照几条不同的执行路线共同工作，独立完成各自的功能而互不干扰。

1 多线程实现技术

- 线程的生命周期

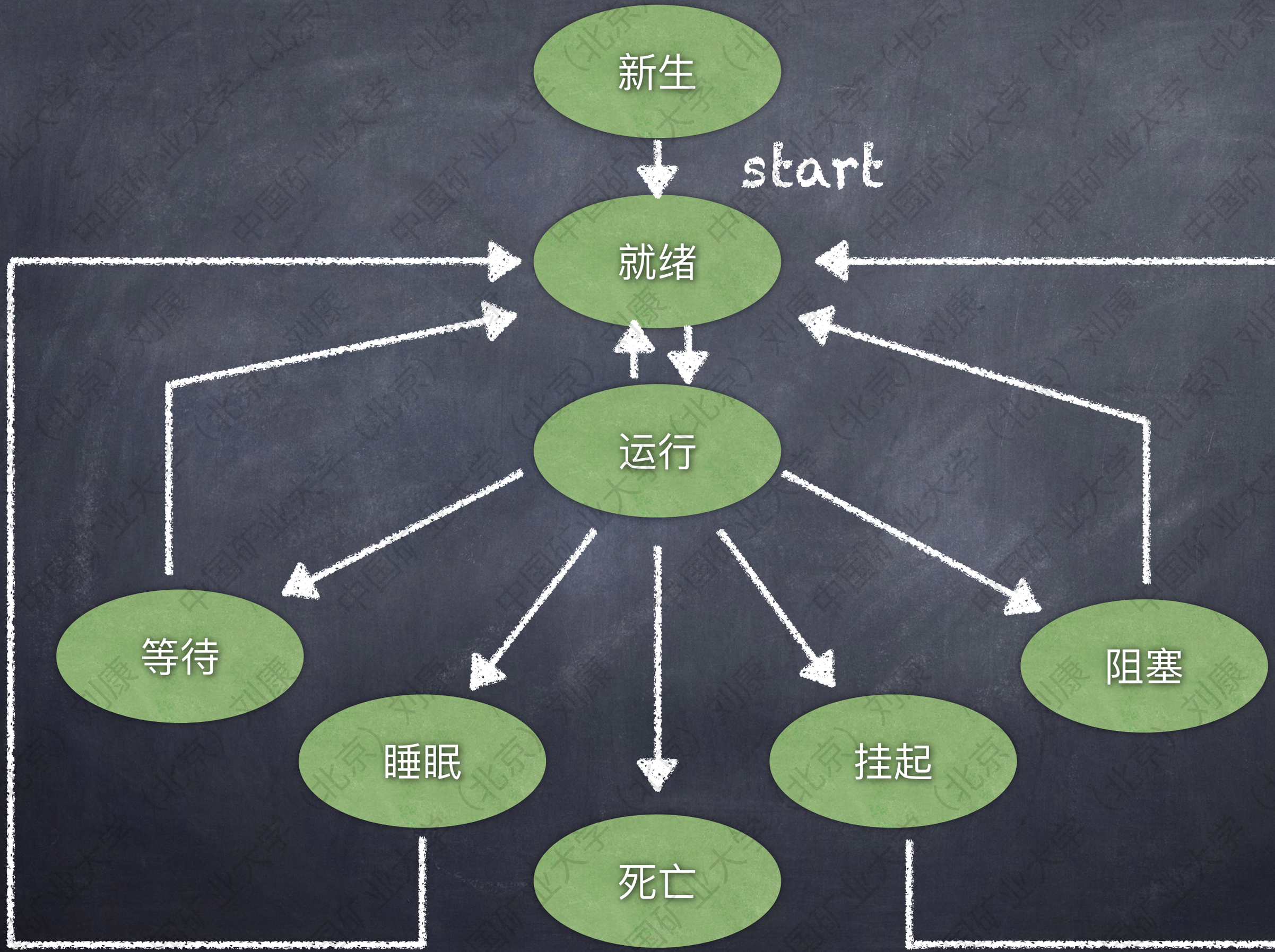
  每个Java程序都有一个缺省的主线程。

  对于Application，主线程是main( )方法执行的线索。

  对于Applet，主线程指挥浏览器加载并执行Java小程序。

- Java语言使用Thread类及其子类的对象来表示线程。

- 新建的线程在它的一个完整的生命周期中通常要经历新生、就绪、运行、阻塞和死亡等五种状态。


新生
start
就绪
运行
等待
睡眠
死亡
挂起
阻塞

1.1 Thread类的方法

- Thread类(线程类)是java.lang包中的一个专门用来创建线程和对线程进行操作的类。

- Java在Thread类中定义了许多方法，帮助我们运用和处理线程。这些方法可分为四组：

  (1) 构造方法。用于创建用户的线程对象，书中表13.1列出了Thread类的构造方法。

(2) run()方法。定义用户线程所要执行的操作。

(3) 改变线程状态的方法。如start()、sleep()、stop()、suspend( )、resume()、yield()和wait()方法等。

(4)其他方法。如setPriority()、setName()

书中表13.2列出Thread类的后三种方法。

1.2 继承Thread类创建线程

- 在Java中创建Thread类的子类：

  1.声明子类的构造方法；

  2.自定义的run()方法覆盖Thread类的run()方法，即将自定义程序块写入run()方法中。

1.3 Runnable接口创建线程

- Runnable接口只有run()，用户新建线程的操作由这个方法来决定。
- run()方法必须由实现此接口的类来实现。定义run()方法后，当用户程序需要建立新线程时，在实现run()方法的类中创建系统类Thread的对象，从而继承用户实现的run()方法。

2 多线程管理

2.1 线程调度

在单CPU的计算机上运行多线程程序，或当线程数多于处理机的数目时，存在多个线程争用CPU的情况，这就需要提供一种机制来合理地分配CPU，使多个线程有条不紊、互不干扰地工作。

调 度

- 在Java运行系统中，由线程调度器对线程按优先级进行调度。当有多个线程在同一时刻处于就绪状态时，线程调度器选择优先级最高的线程运行。

- 如果发生下列情况之一，调度器终止此线程运行：

(1) 本线程调用yield( )方法，放弃对CPU的占有权；

(2) 本线程调用了sleep( )方法，使线程进入睡眠状态；

(3) 本线程由于I/O操作而进入阻塞状态；

(4)其它具有更高优先级的线程从睡眠状态被唤醒，或I/O操作完成而返回就绪状态。

- Java的线程调度算法分为两种：

(1) 优先抢占式

当线程的优先级不同时，为保证优先级最高的线程先运行而采用优先抢占式调度算法，即优先级高的线程优先抢占CPU。

(2) 轮转调度

当若干个线程具有相同的优先级时，则采用队列轮转调度算法，即当一个线程运行结束时，选择调度该优先队列中排在最前面的线程运行。

2.2 线程优先级

- 在Java系统中，运行的每个线程都有优先级。设置优先级是为了在多线程环境中便于系统对线程进行调度。

- Java线程的优先级是一个在1～10之间的正整数，数值越大，优先级越高。未设定优先级的线程其优先级取缺省值为5。

- Java线程的优先级设置遵从下述原则：

(1)线程创建时，子线程继承父线程的优先级；

(2)线程创建后，可在程序中通过调用setPriority()方法改变线程的优先级；

- Java线程的优先级设置遵从下述原则：

(3)线程的优先级是1~10之间的正整数，可用标识符常量

MIN_PRIORITY表示优先级为1，

NORM_PRIORITY表示优先级为5，

MAX_PRIORITY表示优先级为10。

其他级别的优先级既可以直接用正整数来设置，也可用标识符常量与常数相加。

```
setPriority(Thread.NORM_PRIORITY + 3);
```

2.3 线程同步

- 由于Java支持多线程，具有并发功能，大大提高了计算机处理能力。
- 但当两个或两个以上的线程需要共享同一资源时，线程之间的执行次序就需要协调，并且线程占用这一资源时，其他线程只能等待。

2.3 线程同步

- 在程序设计中，可用两个线程分别代表生产者和消费者，可将货架视为任意时刻只允许一个线程访问的临界资源。

- 为了不发生混乱，规定：

  1.当生产者往货架上放置货物时，不允许消费者取货物

  2.当消费者从货架上取货物时，不允许生产者放货物

线程同步

2.3 线程同步

- Java系统中，临界区程序段使用关键字synchronized标识。

synchronized (expression) statement

- 临界区程序通过监控器的系统软件管理，一般为自定义实现的类。

- 当执行临界区程序段时，监控器将这段程序加锁，此时，该线程占用临界资源，直到程序执行完，才释放锁。

2.4 线程组

- Java系统的每个线程都属于某一个线程组。
- 采用线程组，可对多个线程进行集中管理，可同时启动、挂起或终止一个线程组中的全部线程。
- Java系统提供ThreadGroup类实现对线程组的管理功能。