本程序一旦调用 stop()方法就会将 MyThread 类中的 flag 变量设置为 false，这样 run()方法就会停止运行，这种停止方式是开发中比较推荐的。

本章小结

1. 线程（thread）是指程序的运行流程。“多线程”的机制可以同时运行多个程序块，使程序运行的效率更高，也解决了传统程序设计语言无法解决的问题。
2. 如果在类里要激活线程，必须先做好两项准备：此类必须继承 Thread 类或者实现 Runnable 接口；线程的处理必须覆写 run()方法。
3. 每一个线程在其创建和消亡之前，均会处于创建、就绪、运行、阻塞、终止 5 种状态之一。
4. Thread 类里的 sleep()方法可用来控制线程的休眠状态，休眠的时间要视 sleep()里的参数而定。
5. 当多个线程对象操纵同一共享资源时，要使用 synchronized 关键字来进行资源的同步处理。
6. 线程的存在离不开进程。进程如果消失后线程一定会消失，反之如果线程消失了，进程未必会消失。
7. 对于多线程的实现，重点在于 Runnable 接口与 Thread 类启动的配合上。
8. 对于 JDK 1.5 新特性读者了解就行，知道区别就在于返回结果上即可。
9. Thread.currentThread()可以取得当前线程类对象；
10. Thread.sleep()主要是休眠，感觉上是一起休眠，但实际上是有先后顺序的。
11. 优先级越高的线程对象越有可能先执行。
12. 同步和异步的操作可以通过 synchronized 来实现。
13. 死锁是一种不定的状态。

课后习题

一、填空题

1. Java 多线程可以依靠_____、_____和_____三种方式实现。
2. 多个线程操作同一资源的时候需要注意_____，依靠_____关键字实现，实现手段是：_____和_____，过多的使用，则会出现_____问题。
3. Java 程序运行时，至少启动_____个线程，分别是_____和_____。
4. main 线程的优先级是_____。
5. 线程在生命周期中要经历五种状态，分别是_____状态、_____状态、_____状态、_____状态和_____状态。
6. Object 类提供的_____、_____和_____三个方法可以控制线程。

二、选择题

1. 线程的启动方法是（　　）。

   A. run()　　B. start()　　C. begin()　　D. accept()

2. Thread 类提供表示线程优先级的静态常量，代表普通优先级的静态常量是（　　）。

   A. MAX_PRIORITY　　B. MIN_PRIORITY

   C. NORMAL_PRIORITY　　D. NORM_PRIORITY

3. 设置线程优先级的方法是（　　）。

A. setPriority()　B. getPriority()　C. getName()　D. setName()

4. Thread 类的（　　）方法是不建议使用的。

A. stop()　B. suspend()　C. resume()　D. 全部都是

5. 下列（　　）关键字通常用来对对象加锁，从而使得对对象的访问是排他的。

A. serialize　B. transient　C. synchronized　D. static

三、判断题

1. Java 直接调用 Thread 类中的 run()方法可以启动一个线程。（　　）
2. 进程是在线程的基础之上的进一步划分。（　　）
3. Java 是多线程的编程语言。（　　）
4. 不管使用 Callable 还是 Runnable 接口实现的多线程最终都需要通过 Thread 类启动。（　　）

四、简答题

1. 简述线程两种实现方式及区别。
2. 简述死锁的产生。

五、编程题

设计四个线程对象，两个线程执行减操作，两个线程执行加操作。