进程：

应用程序是静态的，一个程序可有有多个进程，进程是动态的，即正在运行的程序。

并发执行：多个不同的程序同时运行的现象。并发执行的最小单元是线程。

线程：是进程的分支，一个进程至少有一个线程。线程分为单线程和多线程，单线程程序按调用顺序依次执行相应的操作，而多线程程序相应的操作并发执行。

extends Thread 当线程类的子类，即继承了线程类的方法，有.start（）方法，调用后可直接进入实例化的线程类的子类的对象的run（）方法里，但是因为java是单继承的机制，所以为了优化结构，将继承线程类优化成了implements Runnable实现接口，这样，子类还可以继承其原本就有的父类，但是要注意，Runnable接口中只有run方法，要是开启这个线程，让多个线程并发执行，需要将实列化的该子类对象套在线程类的构造方法里，这样创建一个线程并开启后，开启的这个线程调用的就是这个子类的run方法。

两种实现多线程方式的对比分析：

extends Thread 不能实现共享一个资源，而implements Runnable可以继承一个多个线程共享一份资源。

后台线程与前台线程：前台线程一旦结束，程序随之结束。刚开始设置的线程都是前台线程，而通过setDeamon（）方法可以设置成后台线程。

线程的生命周期及状态转换：

新建状态---就绪状态---运行状态--------死亡状态

--阻塞状态—

th.start()后该线程是进入就绪状态，等待系统的统一调度来获取cpu资源并运行。

线程的调度： 分时调度模型和抢占式调度模型

分时：平均分配各个线程使用cpu的时间

抢占式：按优先级大小顺序依次执行，优先级高的优先执行。同等优先级情况下随机执行。Main函数具有普通的优先级。

线程休眠：是Thread的静态方法sleep（），这个方法需要捕捉异常或者抛出异常

线程让步：yield（）方法。与sleep（）的不同在于sleep会阻塞当前线程一段时间，而yield是让当前线程进入就绪状态，而且只能是优先级大于等于当前线程的线程才能运行。

线程插队：join（）方法 ，插队的线程完成后，才会执行当前线程。