1，程序（program）：为完成特定任务，用某种语言编写的一组指令的集合，即静态代码或静态对象
2，进程（process）：程序的一次执行过程，或正在执行的过程，动态过程
3，线程（thread）：进程可进行细化为线程，是一个程序内部的一条执行路径

程序是静态的，进程是动态的

若一个程序可同一时间执行多个线程，是支持多线程的

一，使用多线程的情况
①程序需要同时执行俩个或则多个任务
②程序需要实现一些需要等待的任务时，如用户输入，文件读写操作，网络操作，搜索等
③需要一些后台运行的程序时

二，多线程的创建和启动
java语言的jvm允许程序运行多个线程，通过java.lang.Threa类来实现

Thread类的特性：
①每个线程都是通过特定的thread对象的run（）方法来完成操作的，经常把run（）方法的主体称为线程
②通过该Thread对象的start（）方法来调用这个线程

a，创建一个继承于Thread的子类
b，重写thread中的run（）方法，方法内实现了子线程需要完成的功能
c，创建一个子类的对象
d，调用线程的start（）启动此线程，调用相应的run方法
e，一个线程只能执行一次start（）
f，不能通过thread实现类对象的run（）方法去启动一个线程

Thread.currentThread().getName()获取正在执行的线程的名字，主线程就叫main，

Thread的常用方法：
①start（）；启动线程并执行相应的run（）方法
②run（）；重写Thread类中的run（）方法，将子线程要执行的代码放入run（）方法中
③currentThread（）；静态的调取当前线程

④getName（）；获取线程的名字

⑤setName（）；设置此线程的名字

⑥yield（）；调用此方法的线程释放当前cpu的执行权，

⑦join（）；有异常，需要try/catch一下，在A线程中调用B线程的join（）方法表示当次线程执行到此方法的时候，A线程停止执行，直到B线程执行完毕

⑧isAlive（）；判断当前线程是否还存活，返回一个boolean值，如果还存活返回true否则返回false

⑨sleep(long a)；显示的让当前线程睡眠a毫秒，需要try/catch处理一下（被重写的方法不能抛出比父类更大的异常）

⑩线程通信：wait notify（）notifyAll（）

设置线程的优先级：

线程的调度：调度策略：时间片和抢占式（优先级高的线程抢占cpu）

①getPriority（）；获取当前线程的优先级

②setPriority（int a）；a大于等于1小于等于十，最大为10，最小为1，默认为5

（设置优先级只是让该线程抢占到cpu的概率变大，并不一定确定它能强到cpu的资源）

继承于Thread的匿名类对象

```
st2.start();
//继承于Thread类的匿名类的对象
new Thread(){
    public void run(){
        for(int i = 1;i <= 100;i++){
            if(i % 2 == 0){
                System.out.println(Thread.currentThread().getName() + ":
            }
        }
    }
}.start();
```