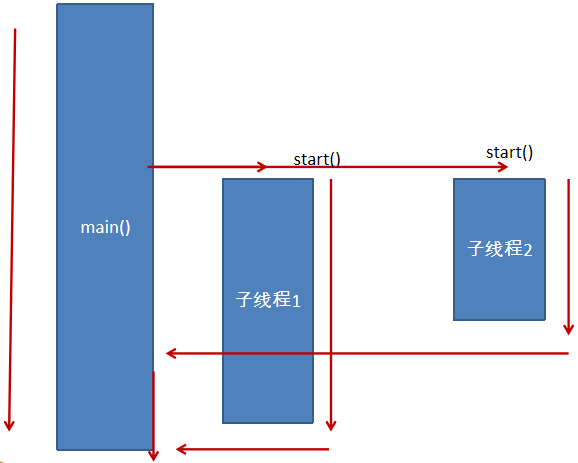
* **线程的创建和启动**
* Java语言的JVM允许程序运行多个线程，它通过**java.lang.Thread**类来实现。
* Thread类的特性：
* 每个线程都是通过某个特定Thread对象的run()方法来完成操作的，经常把run()方法的主题称为线程体。
* mt子线程的创建和启动过程：



* **Thread类**
* **Thread()**：创建新的Thread对象。
* **Thread(String threadname)**：创建线程并指定线程实例名。
* **Thread(Runnable target)**：指定创建线程的目标对象，它实现了Runnable接口中的run方法。
* **Thread(Runnable target, String name)**：创建新的Thread对象。
* **创建线程的两种方式**
* 继承Thread类：
* 定义子类继承Thread类。
* **子类中重写Thread类中的run方法**。
* 创建Thread子类对象，即创建了线程对象。
* 调用线程对象start方法：启动线程，调用run方法。
* 一个线程只能执行一次。
* 不能通过Thread实现类对象的run()方法去启动一个线程。

public class TestThread {  
  
 public static void main(String[] args) {  
 //3）创建一个Thread的子类对象  
 SubThread st = new SubThread();  
 //4）调用线程的start()方法，启动此线程：调用响应的run()方法  
 st.start();  
  
 //主线程做同样的事  
 for (int i = 0; i <= 100; i++) {  
 System.***out***.println(Thread.currentThread().getName() + ":" + i);  
 }  
 }  
  
}  
  
/\*\*  
 \* 1）创建一个继承于Thread的子类  
 \*/  
class SubThread extends Thread {  
  
 /\*\*  
 \* 2）重写Thread类的run()方法，方法内实现此子线程要完成的功能。  
 \*/  
 @Override  
 public void run() {  
 for (int i = 0; i <= 100; i++) {  
 System.***out***.println(Thread.currentThread().getName() + ":" + i);  
 }  
 }  
}



* 实现Runnable接口：
* 定义子类，实现Runnable接口。
* **子类中重写Runnable接口中的run方法**。
* 通过Thread类含参构造器创建线程对象。
* **将Runnable接口的子类对象作为实际参数传递给Thread类的构造方法中**。
* 调用Thread类的start方法：开启线程，调用Runnable子类接口的run方法。

public class TestThread2 {  
  
 public static void main(String[] args) {  
 PrintNum1 p = new PrintNum1();  
 //要想启动一个多线程，必须调用start()方法。  
 Thread t1 = new Thread(p);  
 //启动线程，执行Thread()对象生成时构造器形参的对象的run()方法。  
 t1.start();  
  
 //再创建一个线程  
 Thread t2= new Thread(p);  
 t2.start();  
 }  
}  
  
class PrintNum1 implements Runnable {  
  
 @Override  
 public void run() {  
 //子线程执行的代码  
 for (int i = 1; i <= 100; i++) {  
 if (i % 2 == 0) {  
 System.***out***.println(Thread.currentThread().getName() + ":" + i);  
 }  
 }  
 }  
}

* 继承方式和实现方式的联系与区别

public class Thread extends Object implements Runnable

* 区别：
* 继承Thread：线程代码存放Thread子类run方法中。
* 实现Runnable：线程代码存在接口的子类的run方法。
* 实现方法的好处：
* **避免了单继承的局限性**。
* **多个线程可以共享同一个接口实现类的对象，非常适合多个相同线程来处理同一份资源**。