## 什么是多线程：

每个程序为一个进程，一个进程含有多个线程（至少一个线程），线程控制着进程的执行。

以迅雷为例，每个下载任务都是多个线程同时进行下载的。

## 多线程的意义：

多线程使得一个进程可以同时执行多个任务。

## 创建新执行线程有三种方法。

## 一种方法是将类声明为 Thread 的子类。该子类应重写 Thread 类的 run 方法。

1 定义一个类继承Thread。

2 重写run方法。

3 创建子类对象，就是创建线程对象。

4 调用start方法，开启线程并让线程执行，同时还会告诉jvm去调用run方法。

**第二种方法是 使用实现接口 Runnable 的对象创建一个线程时，启动该线程将导致在独立执行的线程中调用对象的 run 方法。**

1、定义类实现Runnable接口。

2、覆盖接口中的run方法。。

3、创建Thread类的对象

4、将Runnable接口的子类对象作为参数传递给Thread类的构造函数。

5、调用Thread类的start方法开启线程。

## 第三种方法是实现Callable接口（Callable接口解决了实行run方法无返回值的缺点）

1、定义类实现Callable接口。V类型为你所需返回值的类型

2、覆盖接口中的call方法。

3、创建线程池对象：

ExecutorService es = Executors.newFixedThreadPool(2);

4、调用submit方法将任务提交，然后获取Future对象

5、使用Future的get()方法获取返回值

## 获取线程的名字：

1. 每个线程,都有自己的名字
2. 运行方法main线程,名字就是"main"
3. 其他新键的线程也有名字,默认 "Thread-0","Thread-1"
4. 如何获取线程名：

4.1Thread：

NameThread nt = new NameThread();

nt.setName("旺财");

nt.start();

由于NameThread是Thread的子类，所以NameThread可以在run中直接使用getName()来获得线程名

4.2Runnable：

MyRunnable mr=**new** MyRunnable();

Thread thread=**new** Thread(mr);

thread.setName("Runnable线程名");

thread.start();

由于MyRunnable只是一个接口的实现类，所以MyRunnable只能在run方法中使用Thread.currentThread().getName()来获得当前运行的MyRunnable的线程名

4.3获取主线程名（main）或者获取当前运行的线程名

Thread.currentThread().getName()

## 什么是线程池：

线程池可以说是一个线程类型的集合。

**好处：**如果你要创建很多线程来完成一个任务，多次创建Thread对象很容易使得资源浪费，而线程池里的线程是不死亡的（当然是在不使用shutdown()方法的情况下，因为shutdown()方法直接把线程池给消灭了），可以使得线程重复利用。

**使用步骤：**

1. 创建线程池对象：

ExecutorService es = Executors.newFixedThreadPool(2);2表示连接翅中有两个线程可以使用。

1. 多次使用使用submit()方法来调用线程任务
2. 如果想消除线程池可以用shutdown()方法

**注意：如果任务多于线程池中的线程数，则多余的任务只能等在运行中线程空闲才能在运行余下的任务。**

## 如何向任务中动态传递值:



