4.通过线程池来创建线程

使用ExecutorService、Callable、Future实现有返回结果的线程。

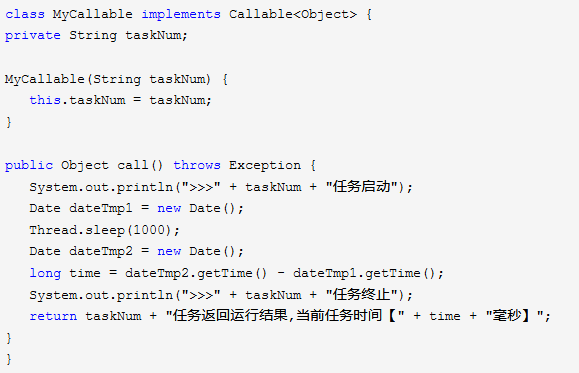
可返回值的任务必须实现Callable接口。类似的，无返回值的任务必须实现Runnable接口。

执行Callable任务后，可以获取一个Future的对象，在该对象上调用get就可以获取到Callable任务返回的Object了。

注意：get方法是阻塞的，即：线程无返回结果，get方法会一直等待。

再结合线程池接口ExecutorService就可以实现传说中有返回结果的多线程了





上述代码中Executors类，提供了一系列工厂方法用于创建线程池，返回的线程池都实现了ExecutorService接口。  
public static ExecutorService newFixedThreadPool(int nThreads)   
创建固定数目线程的线程池。  
public static ExecutorService newCachedThreadPool()   
创建一个可缓存的线程池，调用execute 将重用以前构造的线程（如果线程可用）。如果现有线程没有可用的，则创建一个新线程并添加到池中。终止并从缓存中移除那些已有 60 秒钟未被使用的线程。  
public static ExecutorService newSingleThreadExecutor()   
创建一个单线程化的Executor。  
public static ScheduledExecutorService newScheduledThreadPool(int corePoolSize)   
创建一个支持定时及周期性的任务执行的线程池，多数情况下可用来替代Timer类。