

java认证:多线程技术在J2ME中的应用总结Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/594/2021_2022_java_E8_AE_A4_E8_AF_81_c104_594715.htm 我们知道，在操作系统级别上软件的运行一般都是以进程为单位，而在每个进程的运行过程中允许同时并发执行多个不同线程，这就使得一个程序能同时执行不同的操作。使用多线程的目的是为了最大限度地利用计算机CPU资源。JAVA程序字节码最终是在JVM虚拟机下运行的，同一虚拟机进程中的不同操作都是通过多线程来运行的。在JAVA虚拟机中，线程常用有单线程和多线程，单线程指程序执行过程只是一个有效操作的序列，不同操作都有着明确的先后顺序.而多线程允许同时进行着不同的操作，这些不同的操作同时并发进行着，并由CPU时钟频率根据不同的调度方式对他们进行执行调度。在JAVA语言中提供了丰富的多线程操纵接口，提供了各类不同的线程实现方法供我们选择，功能非常强大。在手机软件设计中，由于同样需要执行网络连接(基于HTTP的高级Internet协议通讯)、UI调度等待、UI显示幻化、游戏控制等操作需要通过后台的数据运算或UI不断更新等操作。因此在J2ME中，KVM虚拟机也提供了功能强大的多线程API，使我们同样能在J2ME中实现线程的并发运算。在J2ME中，主要有以下三种方法实现多线程。

一、继承Thread类(java.lang.Thread) 通过编写线程类继承Thread类并重写Thread类中的run()方法实现线程，当线程对象被运行时候将会自动执行run方法中的实体内容，从而开辟一个单独的线程并运行起来。如：

```
public class ThreadSimple extends Thread{ public ThreadSimple() { //constructor } public void
```

`run() { //run code entity }` } 线程实例使用，直接创建对象并调用`start()`方法即可运行线程。 `new ThreadSimple(). start()`.当执行`start`方法时候，将会自动运行`run`方法，但是执行`start`方法时候只做了一件事，就是将线程转化为可执行状态，然后等待操作系统进行调度并运行，因此无法保证线程能立即启动。在JAVA中，`Thread`类实现了`Runnable`接口，因此`run`方法是通过实现接口`Runnable`中的抽象方法。

二、直接实现Runnable多线程接口(`java.lang.Runnable`)

线程接口`Runnable`中只有一个抽象方法`run`，通过实现`Runnable`接口中的方法的类即可创建出有多线程特征的对象，但该对象并无法使其启动线程，需要作为参数并借助`Thread`的构造方法构造创建对象并调用`start`方法对线程进行启动。如：`public class RunnableSimple implements Runnable{ public RunnableSimple() { //constructor } public void run(){ //run code entity } }` 实现类型的对象使用：`RunnableSimple rs = new RunnableSimple(). new Thread(rs).start()`. 由此可见，以上两种方法都是通过`Thread`的`start`来启动线程的，实际上所有的线程操作都是封装在`Thread`这个类中，由`Thread`对象调用各种接口来控制线程。

J2ME中线程中主要方法：`void setPriority(int newPriority)`，设置线程优先级，在操作系统中线程的调度是不确定性的，可以通过该方法设置相应线程的优先级别。 `static void sleep(long millis)`，线程中静态方法，用于让线程进入休眠状态，执行该方法将会让线程在指定时间`millis`毫秒内休眠。 `void start()`，使现在进入可执行状态。 `void run()`，线程执行主体。 `void join()`，等待该线程终止。 `boolean isAlive()`，用于判断线程是否出于Alive状态。 `static void yield()`，尽量让其他

线程先执行。三、使用任务组合实现多线程 在J2ME中，同样具有JAVA中的任务处理组合类，他们分别为Timer和TimerTask，可以使用他们实现多线程，简单说就是定时实现任务。Timer是JAVA中的一个定时器，可以实现在某一时间做某件事或者在某一时间段做某些事，分别通过方法schedule(TimerTask tt, long millis)和schedule(TimerTask tt, long start, long off)。TimerTask是一个任务类，通过继承该类并覆盖方法run即可创建一个任务。如：

```
public class TimerTaskS extends TimerTask{
    public TimerTaskS(){
        //constructor
    }
    public void run(){
        //run code entity
    }
}
```

任务调用：

```
Timer timer = new Timer(); //3秒钟后执行任务
timer.schedule(new TimerTaskS(), 3000); //3秒钟后执行任务并且之后每5秒钟执行一次
timer.schedule(new TimerTaskS(), 3000, 5000);
```

由此可见在使用计时任务可以达到实现线程的效果，分别执行不同的并发操作，通过Timer类对象来操作TimerTask对象，通过schedule方法来计时执行任务，在结束任务的时候，通常使用cancel()来实现。通常情况下，在J2ME软件中我们通过手机按键来触发一系列相应的操作，在程序响应处理过程中较多会涉及网络操作、数据存储等相对消耗时间和资源的操作，而这些操作往往需要一定的时间才能完成，因此在处理按键响应过程中通常我们需要建立线程处理，避免程序出现死机现象。

```
public void commandAction(Command c, Displayable s) {
    if(c==do1Com){ //创建实现接口线程
        new Thread(new RunnableSimple()).start();
    }
    else if(c==do2Com){ //创建继承Thread线程
        new ThreadSimple().start();
    }
    else{ //创建任务线程
        new
    }
}
```

Timer().schedule(new TimerTaskS(),3000,20). } } 更多优质资料尽在百考试题论坛 百考试题在线题库 java认证更多详细资料 100Test 下载频道开通 , 各类考试题目直接下载。详细请访问 www.100test.com