

第三周学习笔记

07月23日

一：线程 进程区别

1. 进程在操作系统中可以独立运行，作为资源分配的基本单位，表示运行中的程序；而线程是进程中的一个实例，作为系统调度和分派的基本单位，是进程中的一段序列，能够完成一个进程的功能。
2. 同一个进程可以包含多个线程，一个进程至少包含一个线程，一个线程只能存在于一个进程中。
3. 同一个进程下的所有线程能够共享进程下的资源。
4. 线程是轻量级的进程，它的创建和销毁所需要的时间比进程小得多，所有操作系统的执行功能都是通过创建线程去完成的
5. 线程在执行时是同步和互斥的，因为他们共享一个进程下的资源

二：实现线程几种方式

1. 定义子类继承于 Thread 类，创建子类对象并且调用 start 方法
2. 定义子类实现 Runnable 接口，重写 run 函数，创建 Thread 对象和线程任务对象，将线程任务对象传入线程对象中，并且调用线程对象的 start 方法

三：Thread 类中的 start() run() 方法的区别

调用 `start` 方法真正实现了多线程，使线程处于可运行的状态，一旦线程获取时间片，就会执行 `run` 方法，当 `run` 方法结束，此县城也结束。而 `run` 方法只是类的一个普通方法而已，如果直接调用 `run`，并没有开启一个新的线程

四：什么是线程安全？`Vector` 是一个线程安全类吗？

不同的线程对对共享内存进行读写操作的时候由线程的切换导致的安全问题。`Vector` 不是一个线程安全类，最好用 `arraylist`

五：如何在两个线程间共享数据？什么是生产者消费者模型？

将数据抽象成 1 个类，并且将对这个数据的操作作为这个类的方法，并且使用同步

生产者和消费者在同一时间段内共用同一个存储空间，生产者往存储空间中添加产品，消费者从存储空间中取走产品，当存储空间为空时，消费者阻塞，当存储空间满时，生产者阻塞。

六：什么是线程池？为什么要使用它？

一种多线程处理形式，处理过程中将任务添加到队列，然后在创建线程后自动启动这些任务。

(1) 降低系统资源消耗，通过重用已存在的线程，降低线程创建和销毁造成的消耗；

(2) 提高系统响应速度，当有任务到达时，通过复用已存在的线程，无需

等待新线程的创建便能立即执行；

(3) 方便线程并发数的管控。因为线程若是无限制的创建，可能会导致内存占用过多而产生 OOM，并且会造成 CPU 过度切换（CPU 切换线程是有时间成本的（需要保持当前执行线程的现场，并恢复要执行线程的现场））。

(4) 提供更强大的功能，延时定时线程池。

七：多线程中调用 `wait()` 和 `sleep()` 方法有什么不同？

`wait` 可以指定时间，也可以不指定；但是 `sleep` 必须要指定时间

在同步中，对于 CPU 的执行权和所的处理不同。`wait` 释放执行权释放锁；`sleep` 在释放执行权的同时释放执行资格，但是不释放锁

八：`synchronized` 和 `ReentrantLock` 有什么不同？

`Synchronized` 使用的锁是隐式锁，可以自己定义锁也可以有同步函数自己定义，但是 `ReentrantLock` 使用的锁是显式锁，通过 `lock` 与 `unlock` 进行加锁减锁，并可以创建多个条件对象，更为方便

九：语法知识：

1、`throws` 和 `throw` 的区别？（毕9-10）

`throws` 用在函数上，而 `throw` 用在函数内

throws 抛出的是异常类，并且可以抛出多个。throw 抛出的是异常对象

2、 什么情况下不能再函数上抛出（声明）异常（毕10-3）

当函数是重写父类或者接口中的函数的时候，如果上一级函数没有抛出异常，则再覆盖的时候不可以抛出异常

3、 线程和进程的区别？（毕11-1）

（1）进程在操作系统中可以独立运行，作为资源分配的基本单位，表示运行中的程序；而线程是进程中的一个实例，作为系统调度和分派的基本单位，是进程中的一段序列，能够完成一个进程的功能。

（2）同一个进程可以包含多个线程，一个进程至少包含一个线程，一个线程只能存在于一个进程中。

（3）同一个进程下的所有线程能够共享进程下的资源。

（4）线程是轻量级的进程，他的创建和销毁所需要的时间比进程小得多，所有操作系统的执行功能都是通过创建线程去完成的

（5）线程在执行时是同步和互斥的，因为他们共享一个进程下的资源

4、 创建线程（继承 Thread）后调用 t.start();和 t.run();的区别？（毕11-3）

调用 start 可以开启一个线程并且运行 run 函数,但是 run 方法只是一个普通的函数而已,不会开启线程

5、 继承和实现两种方式实现线程的区别 ? (毕11-8)

实现解决的 java 不允许许多继承的弊端,并且用面向对象的思想将线程任务代码封装成了对象

6、 synchronized 和 lock 这两中方式的区别 ? (毕12-6)

第一种使用的是隐式锁,第二种使用的显式锁对象,第一种只能有一个监视器而第二种能够有多个监视器即条件对象。

7、 stop 方法和 interrupt 方法的区别 ? (毕12-7)

stop 是将线程强行停止,会破坏对象;而 interrupt 是安全的,它的作用是改变线程的状态,对当前状态进行解除,从冻结状态恢复到运行状态,并且抛出异常

8、下面两句代码的区别 ? (毕12-1)

```
String str1 = "abc";
```

JVM 首先会到字符串常量池中检查是否存在“abc”,如果不存在,则创建对象,并将地址返回给 str1,否则直接返回

```
String str2 = new String("abc");
```

在栈区中创建 str2变量,在堆区中创建 abc 对象,并将对象的地址赋给 str2

9、写出下面语句的运行结果 ?

```
System.out.println("abdlghl".lastIndexOf("dlgh"));
```

2

10、数组中的获取长度和字符串中获取长度的方法有什么区别？
(12-2)

数组是 `length`，字符串是 `length()`

`length()` 方法返回的正是字符数组 `value` 的长度 (`length`)，`value` 本身是 `private` 的，因此很有必要为 `String` 类提供一个 `public` 级别的方法来供外部访问字符串的长度

11、写出下面语句的输出结果？

```
System.out.println("abcdefgh".substring(2, 6));
```

cdef

12、`StringBuffer` 和 `String` 数组都是容器，那他们的区别是？
(毕13-10)

13、`StringBuilder` 和 `StringBuffer` 的区别 (毕13-12)

14、写出下面语句的输出结果？(毕13-13)

```
int x = Integer.parseInt("3c", 16);
```

```
int xx = x + 15;
```

```
System.out.println(xx);
```

75

15、写出下面语句的输出结果？（毕13-14）

```
Integer x1 = 127;  
  
Integer x2 = 127;  
  
Integer x3 = 128;  
  
Integer x4 = new Integer(127);  
  
System.out.println(x1==x2);  
  
System.out.println(x1==x3);  
  
System.out.println(x1==x4);  
  
System.out.println(x1.equals(x4));
```

true

false

false

true