

第三周学习笔记

07月23日

一：线程 进程区别

1. 进程

进程是程序资源分配的最小单元，是程序在系统上的一次动态执行、运行过程，它经历了从代码加载、执行到执行完毕的一个完整过程，是一个能独立运行的单位，它有自己的独立资源和私有的地址空间；

2. 线程

线程是程序执行的最小单元，是进程的一个实体，是 cpu 调度和分派的基本单位，它是比进程更小的能独立运行的基本单位；

3. 总结

线程在进程当中，进程标识一个完整程序，所以进程之间无明显联系；线程直属于 cpu，是进程中的基本单位，所以线程之间可以共享内存

二：实现线程几种方式

1. Thread 类

通过继承 Thread 类，重写 Thread 的 run () 方法，将线程运行的逻辑放入其中；

2. Runnable 接口

实现 Runnable 接口，实例化 Thread 类；

三：Thread 类中的 start() run() 方法的区别

区别：

Run 在原来的线程中调用，start 启动新的线程；start 内部调用 run 方法。

四：什么是线程安全？Vector 是一个线程安全类吗？

1. 线程安全：

多线程编程时的计算机程序代码的一个概念，线程安全的代码会通过同步机制保证各个线程都可以正确且正常的执行，因此如果单线程每次运行的结果与多线程运行时的结果一致，且

其它的变量的值也与预期一致时便是线程安全的；

2. Vector:

Vector 通过同步的方法，使多个线程不可能同时访问 Vector 中的数据，只能一个一个访问，不会出现数据混乱的现象，所以线程安全；

五：如何在两个线程间共享数据？什么是生产者消费者模型？

1. 线程间的数据共享：

如果每个线程执行相同的代码，那么可以使用 Runnable 对象，Runnable 里面的数据共享；

如果每个线程执行的代码不同，则需要两个 Runnable，通过内部类方法，将共享数据写在内部类中，提供每个线程不同的方法，构建实例，通过构造方法传递的对象，进行数据共享；或者通过外部类方法，将共享数据提供在外部类的成员变量中，提供外部类的不用线程的方法，在 Runnable 中调用外部类的方法或者变量，以此实现数据共享。

2. 消费者模型

在一个系统中存在生产者、消费者两种角色，通过内存缓冲区进行通信，生产者生产数据，消费者将数据做成产品；生产者生产数据到缓冲区中，消费者从缓冲区中取数据。如果缓冲区已经满了，则生产者线程阻塞。如果缓冲区为空，那么消费者线程阻塞

六：什么是线程池？为什么要使用它？

1. 线程池：

一种线程使用模式；

线程池维护着多个线程，等待着监督管理者分配可并发执行的任务。这避免了在处理短时间任务时创建与销毁线程的代价。线程池不仅能够保证内核的充分利用，还能防止过分调度。可用线程数量应该取决于可用的并发处理器、处理器内核、内存、网络 sockets 等的数量。

2. 线程池优点：

线程池可以通过重用已存在的线程降低系统资源消耗，降低线程创建和销毁造成的消耗；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；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))。
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？

1. Wait ()

Object 类的方法，可以中断线程的运行，使本线程等待，暂时让出 CPU 的使用权，并允许其他线程使用这个同步方法。其他线程如果在使用这个同步方法时不需要等待，那么它使用完这个方法的同时，应该用 notifyAll() 方法通知所有由于使用了这个同步方法而处于等待的线程结束等待，曾中断的线程就会从刚才中断处继续执行这个同步方法（并不是立马执行，而是结束等待），并遵循“先中断先继续”的原则。

2. sleep ()

Thread 类的方法，线程通过调用该方法，进入休眠状态主动让出 CPU，从而 CPU 可以执行其他的线程。经过 sleep 指定的时间后，CPU 回到这个线程上继续往下执行。；

3. 区别：

sleep() 方法的主要作用是让线程暂停一段时间，时间一到则自动恢复，不涉及线程间的通信，也不会释放锁，可以放在任何地方使用，但 sleep() 方法必须捕获异常；

调用 wait() 方法后，线程会释放掉它所占用的锁，从而使线程所在对象中的其他 synchronized 数据可被别的线程使用，用于线程间的通信，必须放在同步控制方法或者同步语句块中使用，不需要捕获异常。

八：synchronized 和 ReentrantLock 有什么不同？

1. synchronized

Java 语言的关键字，可用来给对象和方法或者代码块加锁，当它锁定一个方法或者一个代码块的时候，同一时刻最多只有一个线程执行这段代码。当两个并发线程访问同一个对象 object 中的这个加锁同步代码块时，一个时间内只能有一个线程得到执行。另一个线程必须等待当前线程执行完这个代码块以后才能执行该代码块。然而，当一个线程访问 object 的一个加锁代码块时，另一个线程仍可以访问该 object 中的非加锁代码块。；

2. ReentrantLock

ReentrantLock 类，实现了 Lock 接口，是一种可重入的独占锁，具有与使用 synchronized 相同的一些基本行为和语义，但功能

更强大。ReentrantLock 内部通过内部类实现了 AQS 框架的 API 来实现独占锁的功能；

3. 区别：

Synchronized 是 java 语言的关键字，是原生语法层面的互斥，需要 jvm 实现，使用比较方便简洁；

ReentrantLock 是 JDK 1.5 之后提供的 API 层面的互斥锁，需要 lock() 和 unlock() 方法配合 try/finally 语句块来完成，锁更灵活；

优化以后的 Synchronized 的性能与 ReentrantLock 差不多。

九：语法知识：

1、 throws 和 throw 的区别？（毕9-10）

答：

throw 是用作可执行语句，明确的抛出一个异常；throws 是用作声明语句，标示一个方法可能会抛出的异常。

2、 什么情况下不能再函数上抛出（声明）异常（毕10-3）

答：

如果父类或接口的方法中没有异常抛出，那么子类在覆盖方法时，也不可以抛出异常。

3、 线程和进程的区别？（毕11-1）

答：

线程在进程当中，进程标识一个完整程序；线程直属于 cpu，是进程中的基本单位、控制单元；线程控制进程程序的进行。

4、 创建线程(继承 Thread)后调用 t.start(); 和 t.run();

的区别？（毕11-3）

答：

调用 start 方法方可启动线程，而 run 方法只是 thread 的一个普通方法调用，在主线程里执行，程序中依然只有主线程这一个线程，其程序执行路径还是只有一条，还是要顺序执行，还是要等待 run 方法体执行完毕后才可继续执行下面的代码。把需要并行处理的代码放在 run() 方法中，start() 方法启动线程将自动调用 run() 方法

5、 继承和实现两种方式实现线程的区别？（毕11-8）

答：

单继承使得以继承方法实现线程后不能继承其它的类，而实现方法实现线程可以继承其它类；

实现方法可以访问同一变量，继承方法不能，但继承方法使用线程更方便，可以轻易获取线程 id、名字、状态等；

继承方法可以使用 thread 类中生产者、消费者模式的方式操作同一变量。

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

答：

Synchronized 是 java 语言的关键字，是原生语法层面的互斥，需要 jvm 实现，使用比较方便简洁；

Lock 是一个类，通过这个类可以实现同步访问，同时 Lock 提供了很多 Synchronized 做不到的功能；

采用 synchronized 不需要用户去手动释放锁，当 synchronized 方法或者 synchronized 代码块执行完之后，系统会自动让线程释放对锁的占用；而 Lock 则必须要用户去手动释放锁，如果没有主动释放锁，就有可能导致出现死锁现象。

7、 stop 方法和 interrupt 方法的区别？（毕12-7）

答：

Stop 直接强制终止线程；

Interrupted 可以使线程进入异常中对线程做清尾操作。

8、 下面两句代码的区别？（毕12-1）

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

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

答：

使用的内存不同，前者使用栈内存，后者使用堆内存；

9、 写出下面语句的运行结果？

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

答：

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）

答：

数组中获取的长度是数组元素的个数，获取方式：`array.size()`
字符串中获取长度为字符的个数，获取方法：`string.length()`

11、 写出下面语句的输出结果；

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

答：

（从下标为 `start=2` 的 `c` 读到 `end=6` 的 `g`，不包含 `g`）
`cdef`

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

答：

`String`：生成常量，只能通过 '+' 在末尾通过连接增加字符；
`Stringbuffer`：生成储存字符串变量的储存空间即变量，增删改查方便且线程安全，常用的方法有 `append`、`deleteCharAt`、`delete`、`insert`、`reverse`、`setCharAt`、`trimToSize`、`length`、`replace`、`toString`。

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

答：

继承于同一父类 `AbstractStringBuilder`, `StringBuilder` 提出是为了加快 `StringBuffer` 在单线程中运行的速率, 所有该类未保证线程安全但有速度优势

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);
```

答：

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));
```

答：

True False False True