第三周学习笔记

07月23日

**一：线程 进程区别**

进程是正在进行的程序，线程是进程的执行路径，一个进程里可以包含多个线程

**二：实现线程几种方式**

继承Thread类，重写run（）：实现方式很简单，只需要创建一个类去继承Thread类然后重写run方法，在main方法中调用该类实例对象的start方法即可实现多线程并发。

实现runnable接口，重写实现run（）方法和继承Thread类相似，写一个实现类，在main函数里面，将runnable实现类对象作为参数传递给一个Thread类对象，并调用其start方法即可实现多线程并发。

匿名类，重写调用一起，只用一次

**三：Thread 类中的start() run()方法的区别**

run（）方法中储存的是需要执行的程序

start（）是创建线程并调用run（）方法，是增加一个线程

若直接使用run（）方法，则仅仅是对象调用方法，不增加线程，仍是在主线程内运行，仍是单线程

**四：什么是线程安全？Vector是一个线程安全类吗？**

多个线程执行共享数据时，一个线程可能仅仅执行了一部分，另一个线程就开始执行。导致共享数据出现问题

Vector类：Vector是一个线程安全的类，其在add()等操作上添加了synchronized关键字实现同步，但是并非是绝对的线程安全类．

当进行迭代遍历时，如果在另一个线程执行add(),remove()操作，仍然会有机率抛出异常ConcurrentModificationException

且复合操作非同步，有时候需要对复合操作加锁

**五：如何在两个线程间共享数据？什么是生产消费者模型？**

接口实现类对象作为参数传递给多个Thread类对像

创建一个类去继承Thread类时，定义静态变量

消费者和生产者模型：有共享数据，可以相互唤醒和睡眠的多个线程

**六：什么是线程池？ 为什么要使用它？**

线程池是一种多任务处理形式，线程池中的线程可以重复利用

**七：多线程中调用wait() 和 sleep()方法有什么不同？**

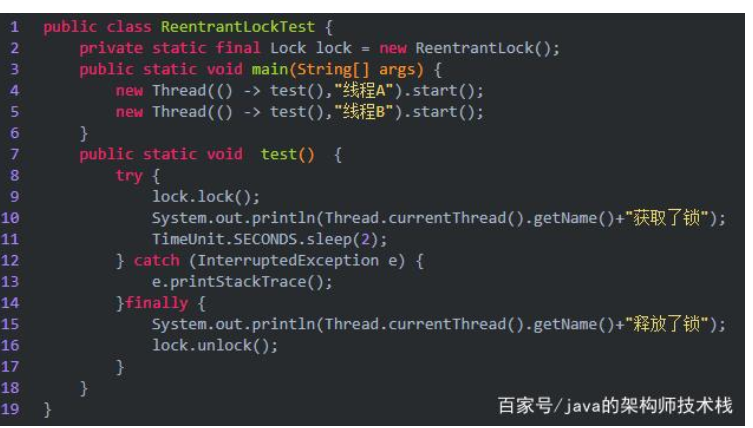
sleep（时间）自己会醒

wait（）需要等待notify（）

**八：synchronized 和** **ReentrantLock 有什么不同**

Reentrantlock 锁显现了，需要手动加锁和解锁，lock lock =new ReentrantLock（（false/ture）） false(默认) 非公平锁 随机 ，true 公平锁 等待时间长的先进行 ，有中断响应机制，可以使因相互争夺资源而无法进行的两个或多个线程中，中断某个线程，使得其他线程得以进行

synchronized 修饰代码块或者修饰函数，自动加锁或者解锁，无中断响应机制

****

**九：语法知识：**

1. throws和throw的区别？（毕9-10）

Throw：

作用在方法内，表示抛出具体异常，由方法体内的语句处理。

具体向外抛出的动作，所以它抛出的是一个异常实体类。若执行了Throw一定是抛出了某种异常。

Throws：

作用在方法的声明上，表示如果抛出异常，则由该方法的调用者来进行异常处理。

主要的声明这个方法会抛出会抛出某种类型的异常，让它的使用者知道获的异常的类型。

出现异常是一种可能性，但不一定会发生异常。

1）作用不同：

throw用于程序员自行产生并抛出异常；

throws用于声明在该方法内抛出了异常

2） 使用的位置不同：

throw位于方法体内部，可以作为单独语句使用；

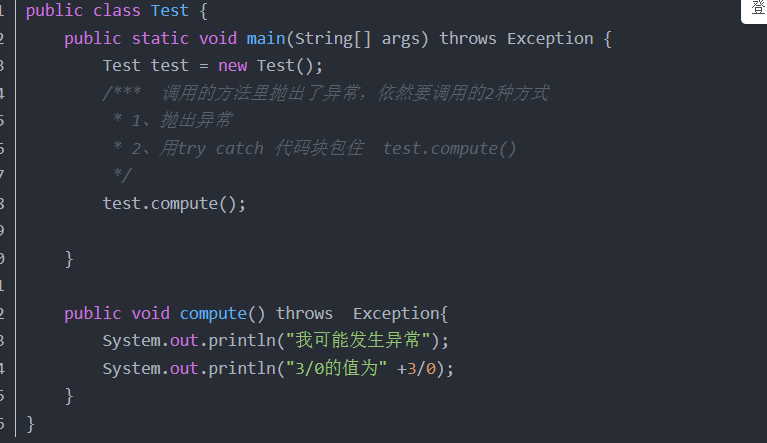
throws必须跟在方法参数列表的后面，不能单独使用。

3）内容不同：

throw抛出一个异常对象，且只能是一个；

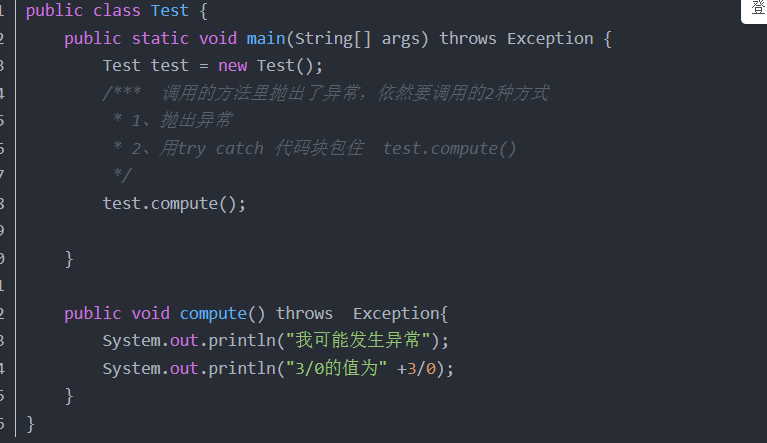
throws后面跟异常类，而且可以有多个

方法名后面跟上 throws Exception 证明这个方法可能发生异常，在别处使用就必须也抛出异常或者用try catch处理



throw 语句不能单独使用必须配合 try catch ，或者throws

情况一：用try catch 包住报错的那一行



情况二：配合throws 使用



2、什么情况下不能再函数上抛出（声明）异常（毕10-3）

7、stop方法和interrupt方法的区别？（毕12-7）

stop方法不建议使用，可能导致线程安全问题

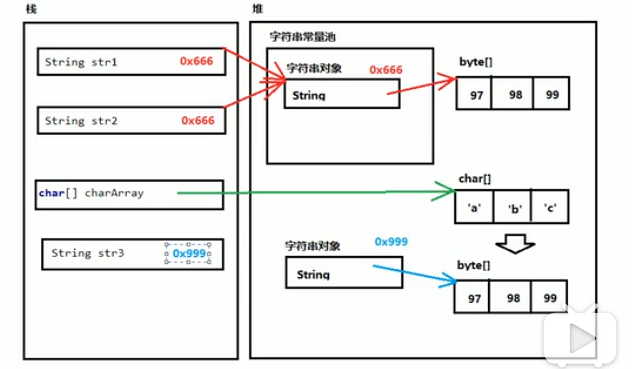
interrupt 方法不会出现强制终止的情况，他会抛出一个异常，线程是安全的

8、下面两句代码的区别？（毕12-1）

String str1 = “abc”;

String str2 = new String(“abc”);

第一个在堆的量池中，第二个是new 之后在堆中



9、写出下面语句的运行结果？

System.out.println(“abdlghl”.lastIndexOf(“dlgh”));

* indexOf 和 lastIndexOf 都是索引文件
* indexOf 是查某个指定的字符串在字符串首次出现的位置（索引值）（从左往右）
* lastIndexOf 是查某个指定的字符串在字符串最后一次出现的位置（索引值）（从右往左）

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）

字符串长度是字符串里bai字符的个数，String中的.length()就可以获得。

数组是一种链表结构，他的长zhi度是数组中包dao含多少个数组元素，用.size()获得。元素可以是各种类型，包括String。

11、写出下面语句的输出结果？

System.out.println(“abcdefgh”.substring(2,6));

bcdef错 cdef 对

substring() 方法用于提取字符串中介于两个指定下标之间的字符。

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

String是一个不可变的字符序列，而stringbuffer是一个可变的字符序列。

内存方面：String是在常量区开辟空间，比较耗费内存，而stringbuffer是一个字符串缓冲区，当字符串的大小没有超过缓冲区的容量时，可以调用之前的字符串，不用开辟新的空间

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

首先String、StringBuffer、StringBuilder在JDK中都被定义为final类，这意味着他们不可以被继承。

2）String最常见，与StringBuffer相比，String的性能较差，因为对String类型进行改变的时候都会重新生成一个新的String对象，这在字符串拼接操作时很明显，因此内容经常改变的字符串不应该使用String，如果不考虑多线程，则应使用StringBuilder。

3）StringBuffer生成一个对象后，在进行字符串拼接操作时，调用append方法即可，不会产生新的对象，仅对对象本身进行操作，性能比String要高。另外StringBuffer是线程安全的，因此适合在多线程中使用，也正因为如此，速度跟StringBuilder相比会比较慢。

4）StringBuilder的使用方法跟StringBuffer类似，但其是非线程安全的，因此一般常用于单线程，效率比StringBuffer高。

14、写出下面语句的输出结果？(毕13-13)

int x = Integer.parseInt("3c",16);3\*16+12=60

int xx = x + 15;60+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));

ture false false ture