****

**《分布式计算》**

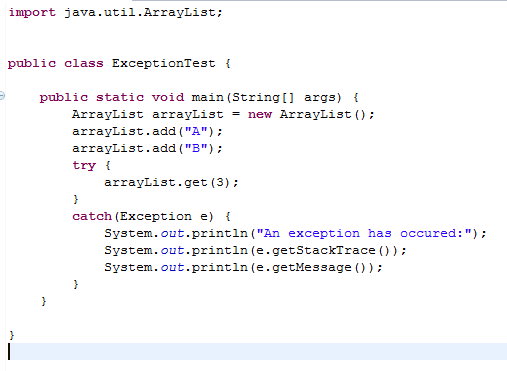
JAVA进阶与Socket通讯

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **学院名称** | **：** | 数据科学与计算机学院 | | | | | |
| **成员** | **：** | 陈伟宸 | | | | | |
| **时间** | **：** | 2016 | 年 | 9 | 月 | 4 | 日 |

1. RuntimeException称为运行时异常，是不检查异常（Unchecked Exception）的一种；

它的子类包括 NullPointerException，IndexOutOfBoundsException等；

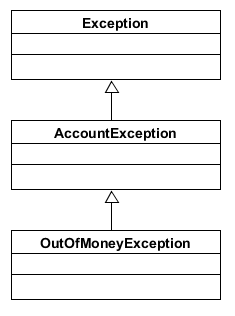
程序代码如下：



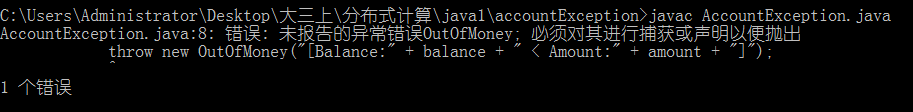
运行结果如下：



2.



3. 不可以，去掉的后果是编译报错：



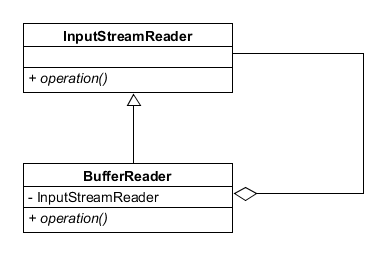
4. 会出错，出错语句为：



5. 发送任意命令后，客户端会意外退出：



6.



7. 不能，因为服务器只有一个线程，在有客户端连接时一直阻塞在while块中，同一时间只能处理一个客户端的请求。

8. 进程是一种重量级任务，而线程则是一种轻量级任务。线程与进程之间的主要区别在于：每一进程占有独立的地址空间，包括代码、数据及其他资源，而一个进程中的多个线程可共享该进程的这些空间；进程之间的通信（简称IPC）开销较大且受到诸多限制，必须具有明确的对象或操作接口并采用统一的通信协议，而线程之间则可通过共享的公共数据区进行通信，开销较少且比较简单；进程间的切换开销也较大，而线程间的切换开销较小。

9.

|  |  |  |
| --- | --- | --- |
|  | 优点 | 缺点 |
| 实现Runnable接口 | 避开了单继承的局限  适用于资源共享 | 编写难度较大 |
| 继承Thread类 | 易于使用 | 不能再继承其他类 |

10. 使用了synchronized 声明的函数具有以下特征：

① 保证在同一时刻最多只有一个线程对一个对象执行这个函数；

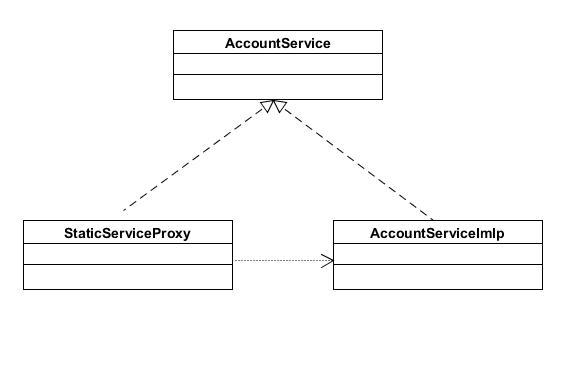
② 该函数具有“happens-before”特性，即先行发生的变化能够被所有的线程观察到。

11. 不能。指针指向的对象的地址在不同的程序实例中时不确定的，指针值没有意义。

12.

13. instanceof是一个操作符，用来判断第一个操作数是否为第二个操作数的实例，得到的结果是一个布尔值。而反射机制是在运行状态中，对于任意一个类，都能够知道这个类的所有属性和方法；对于任意一个对象，都能够调用它的任意一个方法和属性。

14.



当客户端发送消息时，StaticServiceProxy尝试连接服务器，并向服务器发送getAccount命令，等待服务器的响应，并将响应返回给客户端，完成一次通信。

15. 静态代理是由程序员创建或由特定工具自动生成源代码，再对其编译，在程序运行前，代理类的.class文件已经存在了；而动态代理是在程序运行时，运用发射机制动态创建而成。

小结

1. 学习了使用JAVA实现服务器的方式，如单线程，实现Runnable的接口的多线程和继承Thread的多线程

2. 学习了客户端与服务器之间的交互细节，如远程过程调用、静态代理、动态代理

3. 学习了JAVA中的序列化与反序列化

4. 学习了使用Jackson对JAVA对象进行序列化和反序列化

5. 学习了Decorator Pattern的原理及实现方式

6. 复习了线程间的同步问题

7. 重温了安装虚拟机的过程

8. 重温了对虚拟机网络的配置