中山大學
SUN YAT-SEN UNIVERSITY

《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；

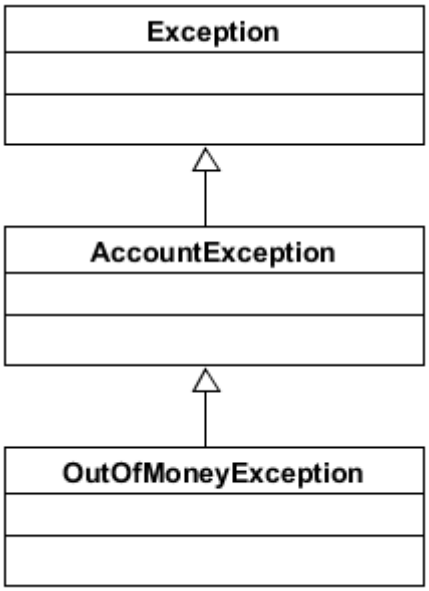
程序代码如下：

```
import java.util.ArrayList;


public class ExceptionTest {

    public static void main(String[] args) {
        ArrayList arrayList = new ArrayList();
        arrayList.add("A");
        arrayList.add("B");
        try {
            arrayList.get(3);
        }
        catch(Exception e) {
            System.out.println("An exception has occured:");
            System.out.println(e.getStackTrace());
            System.out.println(e.getMessage());
        }
    }

}
```

运行结果如下：

```
An exception has occured:
[Ljava.lang.StackTraceElement;@15db9742
Index: 3, Size: 2
```

2.



3. 不可以，去掉的后果是编译报错：

```
C:\Users\Administrator\Desktop\大三上\分布式计算\java1\accountException>javac AccountException.java
AccountException.java:8: 错误: 未报告的异常错误OutOfMoney; 必须对其进行捕获或声明以便抛出
            throw new OutOfMoney("[Balance:" + balance + " < Amount:" + amount + "]");
            ^
1 个错误
```

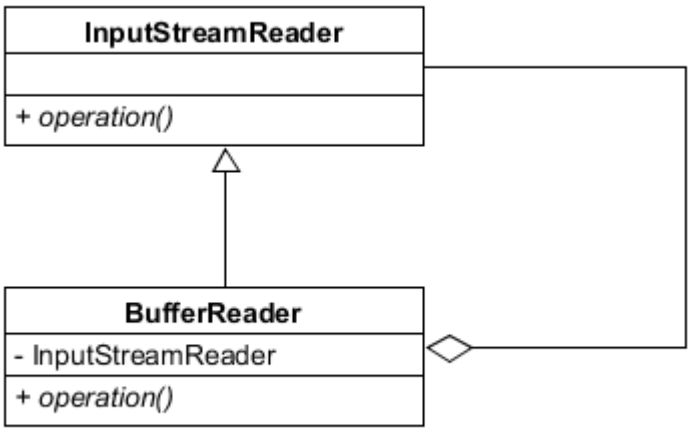
4. 会出错，出错语句为：

```
Socket server = new Socket(args[0], Integer.parseInt(args[1]));
```

5. 发送任意命令后，客户端会意外退出：

```
C:\Users\Administrator\Desktop\大三上\分布式计算\java1\TCPCommunication>java MyClient 127.0.0.1 6543

Exception in thread "main" java.net.SocketException: Connection reset
        at java.net.SocketInputStream.read(Unknown Source)
        at java.net.SocketInputStream.read(Unknown Source)
        at sun.nio.cs.StreamDecoder.readBytes(Unknown Source)
        at sun.nio.cs.StreamDecoder.implRead(Unknown Source)
        at sun.nio.cs.StreamDecoder.read(Unknown Source)
        at java.io.InputStreamReader.read(Unknown Source)
        at java.io.BufferedReader.fill(Unknown Source)
        at java.io.BufferedReader.readLine(Unknown Source)
        at java.io.BufferedReader.readLine(Unknown Source)
        at MyClient.main(MyClient.java:28)
```

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；

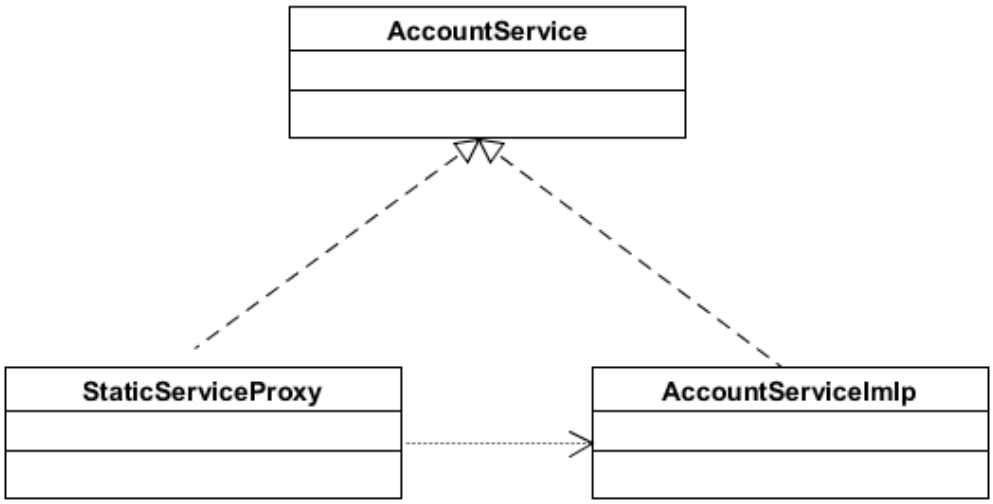
② 该函数具有“happens-before”特性，即先行发生的变化能够被所有的线程观察到。

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