实验十 使用Java编写链码

【实验介绍】

在上一个实验中，我们对e2e案例中的合约进行了手动调用，实现了账户资本的查询以及转账操作。然后大致讲解了一下该案例中使用的合约的实现逻辑。

本次实验，要求大家使用Java实现一个与e2e案例中的合约功能相同的合约，然后进行部署、调用。

【实验要求】

1. 智能合约编写
2. Java实现
3. 合约部署与调用

【实验过程】

导入Java工程项目examplejava到Idea中，同之前的操作一样，不再赘述。

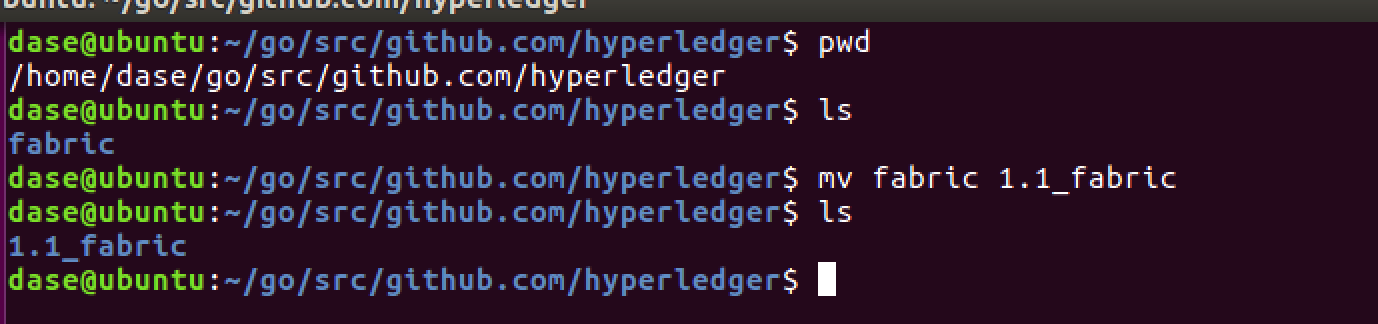
这里只需要你补全下列三个函数的逻辑，其中，该链码要实现的逻辑，与上一份文件中提供的代码文件chaincode\_example02\_e2e.go完全一致，请注意，是完全一致，所以你可以参考chaincode\_example02\_e2e.go中的代码逻辑，进行java版本的编写。最后，请把你的代码文件SimpleChaincode.java，请注意，这个文件就行，无需整个文件夹，将SimpleChaincode.java文件与你的实验报告一同提交。



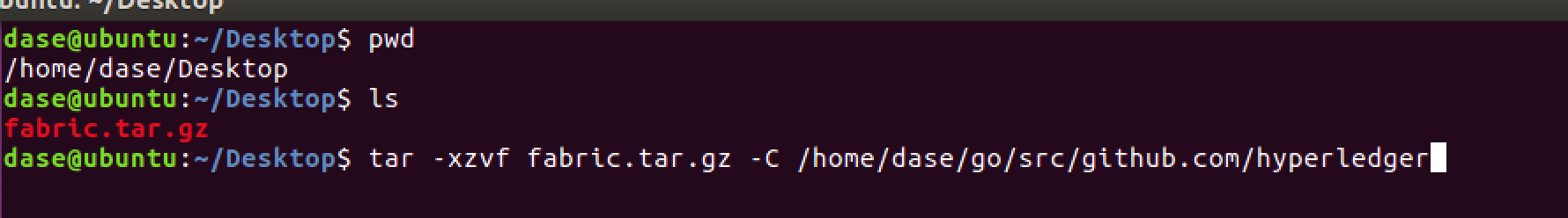
这边由于ChaincodeStub只是抽象类，且其子类实现没对外开放，故没能直接提供测试方法。你可以在完成代码后，直接将代码放入e2e案例中运行，看运行结果是否与Go版本的链码相同。

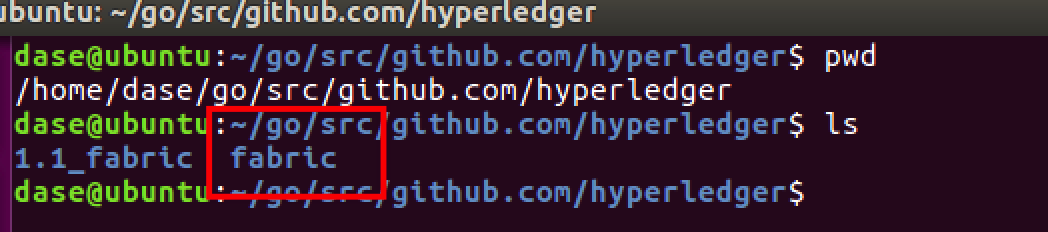
另外，由于之前提供的Fabric版本是1.1，而Java链码在1.3才支持，所以我们这边需要使用1.3版本的fabric。具体的使用方法请根据以下步骤进行：

进入到路径/home/dase/go/src/github.com/hyperledger中，可以看到我们之前使用的fabric文件夹，可以直接rm -rf fabric将其删除，也可以换个目录名，后续不再使用这个版本了，如下所示：

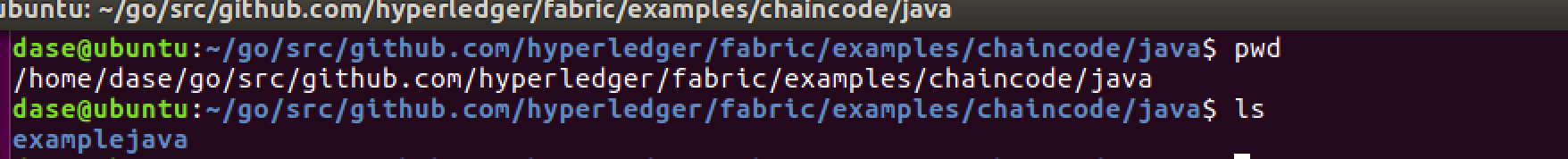


然后将1.3版本的fabric解压缩到路径/home/dase/go/src/github.com/hyperledger中，比如当前该压缩包在桌面上，进行如下操作：

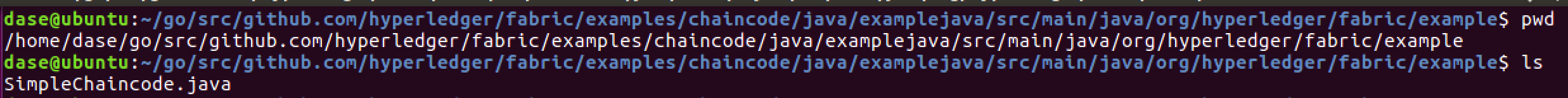




然后进入到路径/home/dase/go/src/github.com/hyperledger/fabric/examples/chaincode/java中，可以看到有一个examplejava的文件夹，和我提供的java工程项目名称一样：



最后，你补全java代码后，需要将你的代码文件SimpleChaincode.java，放至以下路径中，请勿自行修改文件名，也不要删除、移动examplejava文件夹下的内容，也请严格按照下图的路径进行放置，否则无法成功运行：



根据上述步骤放置SimpleChaincode.java文件后，就可以进行e2e案例的运行了，但是在运行之前，需要完成几个步骤，首先，如果你之前有用1.1版本运行过./network\_setup.sh up，请务必确认运行./network\_setup.sh down，以删除相关的容器，然后使用以下命令删除原先的镜像：

docker rmi -f $(docker images -qa)



最后，进入到e2e文件夹中，执行./network\_setup.sh up，进行案例的运行，如果没有明显的报错，那么说明你的链码实现是正确的。其中，运行时间会比较长，第一次运行的话，还需要重新拉去镜像，因此建议在校园网下进行操作。然后后续的输出，也会明显比之前慢一点，没有明显的报错的话耐心等一下。

