

**综合性、设计性实验报告**

姓名\_\_\_\_\_\_\_\_ 学号\_\_\_\_\_\_\_\_\_\_\_\_

专业 软件工程 班级2017级 软件 班

实验课程名称 前沿软件讲座

指导教师及职称 黎明 副高

开课学期 2019 至 2020 学年 第一学期

上课时间 第3次至第4次实验

湖南科技学院教务处编

1. **实验设计方案**

|  |  |  |
| --- | --- | --- |
| 实验名称： 积分智能合约实验 | | 实验时间：第3次至第4次实验 |
| 小组合作： 是○ 否○● | 成绩： | 教师评语： |
| 1、实验目的：  (1)、掌握智能合约的基本概念。  (2)、掌握智能合约的设计方法。  (3)、掌握智能合约的安全编程方法。  (4)、掌握私密信息的设计方法 | | |
| 2、实验设备及材料：  MAC一体机，操作系统 MacOSX，开发平台 Remix或者ethreum平台 | | |
| 实验内容：  (1)、完成银行积分的基本设计，确定功能清单  (2)、确定商户，客户，银行对象的设计  完成相关代码设计  测试相关代码，并注意隐私数据的测试  启动并部署智能合约  调用智能合约相关函数，并实现功能  测试整个代码 | | |
| 4、实验方法步骤：  (1) 分析解题思路  编代码-》创对象-》测试  (2) 编写源程序  **pragma solidity ^0.4.0;**  **contract Score{**  **address owner;//合约的拥有者银行**  **uint issuedScoreAmount;//银行已经发行的积分总数**  **uint settledScoreAmount;//银行已经清算的积分总数**  **struct Customer{**  **address customerAddr;//客户address**  **bytes32 password;//客户密码**  **uint scoreAmount;//积分余额**  **bytes32[] buyGoods;//购买的商品数组**  **}**  **struct Good{**  **bytes32 goodId;//商品Id**  **uint price;//价格**  **address belong;//商品属于那个商户**  **}**  **struct Merchant{**  **address merchantAddr;//商户 address**  **bytes32 password;//商户密码**  **uint scoreAmount;//积分余额**  **bytes32[] sellGoods;//发布的商品数组**  **}**  **mapping (address=>Customer) customer;//根据客户address查找**  **mapping (bytes32=>Good) good;//根据商品Id查找该件商品**  **mapping (address=>Merchant) merchant;//根据商户de的 address查找**  **address[] customers;//已经注册的客户数组**  **bytes32[] goods;//已经上线的商品数组**  **address[] merchants;//已经上线的商品数组**  **//增加权限控制 ，某些方法只能由合约的创建者调用**  **modifier onlyOwner(){**  **if(msg.sender!=owner) throw;**  **\_;**  **}**  **//构造函数**  **function Score(){**  **owner = msg.sender;**  **}**  **//返回合约调用者地址**  **function getOwner() constant returns(address){**  **return owner;**  **}**  **//注册一个客户**  **event NewCustomer(address sender,bool isScuccess,string message);**  **function newCustomer(address \_customerAddr,string \_password){**  **//判断是否已经注册**  **if(!isCustomerAlreadyRegister(\_customerAddr)){**  **//未注册**  **customer[\_customerAddr].customerAddr = \_customerAddr;**  **customer[\_customerAddr].password = stringToBytes32(\_password);**  **customers.push(\_customerAddr);**  **NewCustomer(msg.sender, true,"注册成功 ");**  **return;**  **}else{**  **NewCustomer(msg.sender,false,"该账户已经注册");**  **return;**  **}**  **}**  **//注册一个商户**  **event NewMerchant(address sender,bool isScuccess,string message);**  **function newMerchant(address \_merchantAddr,string \_password){**  **//判断是否已经注册**  **if(!isMerhantAlreadyRegister(\_merchantAddr)){**  **merchant[\_merchantAddr].merchantAddr = \_merchantAddr;**  **merchant[\_merchantAddr].password = stringToBytes32(\_password);**  **merchants.push(\_merchantAddr);**  **NewMerchant(msg.sender, true,"注册成功 ");**  **return;**  **}else{**  **NewMerchant(msg.sender,false,"该账户已经注册");**  **return;**  **}**  **}**  **//判断一个客户是否已经注册**  **function isCustomerAlreadyRegister(address \_customerAddr)internal returns(bool){**  **for(uint i=0;i<customers.length;i++){**  **if(customers[i]==\_customerAddr){**  **return true;**  **}**  **}**  **return false;**  **}**  **//判断一个商户 是否已经注册**  **function isMerhantAlreadyRegister(address \_merchantAddr)internal returns(bool){**  **for(uint i=0;i<merchants.length;i++){**  **if(merchants[i]==\_merchantAddr){**  **return true;**  **}**  **}**  **return false;**  **}**  **//登录 ，查询用户密码**  **function getCustomerPassword(address \_customerAddr)constant returns(bool,string){**  **//先判断该用户是否注册**  **if(isCustomerAlreadyRegister(\_customerAddr)){**  **bytes32 pwd = customer[\_customerAddr].password;**  **return (true,bytes32ToString(pwd));**  **}else{**  **return(false,"");**  **}**  **}**  **//登录 ，查询商户 密码**  **function getMerchantPassword(address \_merchantAddr)constant returns(bool,string){**  **//先判断该是否注册**  **if(isMerhantAlreadyRegister(\_merchantAddr)){**  **bytes32 pwd = merchant[\_merchantAddr].password;**  **return (true,bytes32ToString(pwd));**  **}else{**  **return(false,"");**  **}**  **}**  **//修改客户密码**  **event UpdateCustomer(address sender,bool isScuccess,string message);**  **function updateCustomer(address \_customerAddr,string \_password){**  **for(uint i=0;i<customers.length;i++){**  **if(customers[i]==\_customerAddr){**  **customer[\_customerAddr].password = stringToBytes32(\_password);**  **return;**  **}**  **}**  **}**  **//银行发送机分给客户，只能被银行调用，且只能发给客户**  **event SendScoreToCustomer(address sender,string message);**  **function sendScoreToCustomer(address \_receiver,uint \_amount){**  **if(isCustomerAlreadyRegister(\_receiver)){**  **//已经注册**  **issuedScoreAmount += \_amount;**  **customer[\_receiver].scoreAmount += \_amount;**  **SendScoreToCustomer(msg.sender,"发行积分成功");**  **return;**  **}else{**  **//还没注册**  **SendScoreToCustomer(msg.sender,"该账户未注册，发行积分失败");**  **return;**  **}**  **}**  **//根据客户address查找余额**  **function getScoreWithCustomerAddr(address customerAddr)constant returns(uint){**  **return customer[customerAddr].scoreAmount;**  **}**  **//两个账户转移积分，任意两个账户之间都可以转移 \_senderType 0表示客户，1表示商户**  **event TransferScoreToAnother(address sender,string message);**  **function transferScoreToAnother(uint \_senderType,address \_sender,address \_receiver,uint \_amount){**  **string memory message;**  **if(!isCustomerAlreadyRegister(\_receiver) && !isMerhantAlreadyRegister(\_receiver)){**  **//目的账号不存在**  **TransferScoreToAnother(msg.sender,"目前账户不存在，请确认后再转移");**  **return;**  **}**  **if(\_senderType == 0){**  **//客户转移**  **if(customer[\_sender].scoreAmount >= \_amount){**  **customer[\_sender].scoreAmount -=\_amount;**  **if(isCustomerAlreadyRegister(\_receiver)){**  **customer[\_receiver].scoreAmount += \_amount;**  **}else{**  **merchant[\_receiver].scoreAmount +=\_amount;**  **}**  **TransferScoreToAnother(msg.sender,"积分转让成功！");**  **return;**  **}else{**  **TransferScoreToAnother(msg.sender,"你的积分余额不足，转让失败 ");**  **return;**  **}**  **}else{**  **//商户转让**  **if(merchant[\_sender].scoreAmount>=\_amount){**  **merchant[\_sender].scoreAmount -=\_amount;**  **if(isCustomerAlreadyRegister(\_receiver)){**  **customer[\_receiver].scoreAmount += \_amount;**  **}else{**  **merchant[\_receiver].scoreAmount +=\_amount;**  **}**  **TransferScoreToAnother(msg.sender,"积分转让成功！");**  **return;**  **}**  **}**  **}**  **//银行查找已经发行的积分总数**  **function getIssuedScoreAmount()constant returns(uint){**  **return issuedScoreAmount;**  **}**  **//银行查找已经清算的积分总数**  **function getSettledScoreAmount()constant returns(uint){**  **return settledScoreAmount;**  **}**  **//商户 添加一件商品**  **event AddGood(address sender,bool isScuccess,string message);**  **function addGood(address \_merchantAddr,string \_goodId,uint \_price){**  **bytes32 tempId = stringToBytes32(\_goodId);**  **//首先判断该商品ID是否已经存在**  **if(!isGoodAlreadyAdd(tempId)){**  **good[tempId].goodId = tempId;**  **good[tempId].price = \_price;**  **good[tempId].belong = \_merchantAddr;**  **goods.push(tempId);**  **merchant[\_merchantAddr].sellGoods.push(tempId);**  **AddGood(msg.sender,true,"添加商品成功 ");**  **return;**  **}else{**  **AddGood(msg.sender,false,"该商品已经添加 ");**  **return;**  **}**  **}**  **//用户用积分购买一件商品**  **event BuyGood(address sender,bool isSuccess,string message);**  **function buyGood(address \_customerAddr,string \_goodId){**  **//判断输入的商品ID是否存在**  **bytes32 tempId = stringToBytes32(\_goodId);**  **if(isGoodAlreadyAdd(tempId)){**  **if(customer[\_customerAddr].scoreAmount < good[tempId].price){**  **BuyGood(msg.sender,false,"余额不足，兑换商品失败 ");**  **return;**  **}else{**  **customer[\_customerAddr].scoreAmount -= good[tempId].price;**  **//对应的商品增加相应的yue余额**  **customer[\_customerAddr].buyGoods.push(tempId);**  **BuyGood(msg.sender,true,"购买商品成功 ");**  **return;**  **}**  **}else{**  **BuyGood(msg.sender,true,"该商品未发布");**  **return;**  **}**  **}**  **//判断一个商品是否已经创建**  **function isGoodAlreadyAdd(bytes32 \_tempId)internal returns(bool){**  **for(uint i=0;i<goods.length;i++){**  **if(goods[i]==\_tempId){**  **return true;**  **}**  **}**  **return false;**  **}**  **/// string类型转化为bytes32型转**  **function stringToBytes32(string memory source) constant internal returns(bytes32 result){**  **assembly{**  **result := mload(add(source,32))**  **}**  **}**  **/// bytes32类型转化为string型转**  **function bytes32ToString(bytes32 x) constant internal returns(string){**  **bytes memory bytesString = new bytes(32);**  **uint charCount = 0 ;**  **for(uint j = 0 ; j<32;j++){**  **byte char = byte(bytes32(uint(x) \*2 \*\*(8\*j)));**  **if(char !=0){**  **bytesString[charCount] = char;**  **charCount++;**  **}**  **}**  **bytes memory bytesStringTrimmed = new bytes(charCount);**  **for(j=0;j<charCount;j++){**  **bytesStringTrimmed[j]=bytesString[j];**  **}**  **return string(bytesStringTrimmed);**  **}**  **}**  (3) 结果截图  因为刚开始余额为0所以购买会失败：  将顾客余额设置成100，购买成功：  C:\Users\Administrator\AppData\Local\Microsoft\Windows\INetCache\Content.Word\2019-12-02 09-52-45屏幕截图.png  余额剩下99：  (4) 结果分析  试验成功，实验时地址不知道哪里弄，遇到了一些问题，但解决了。 | | |