2024 年河北省职业院校技能大赛

高职组

“区块链技术应用”赛项

样题一

任

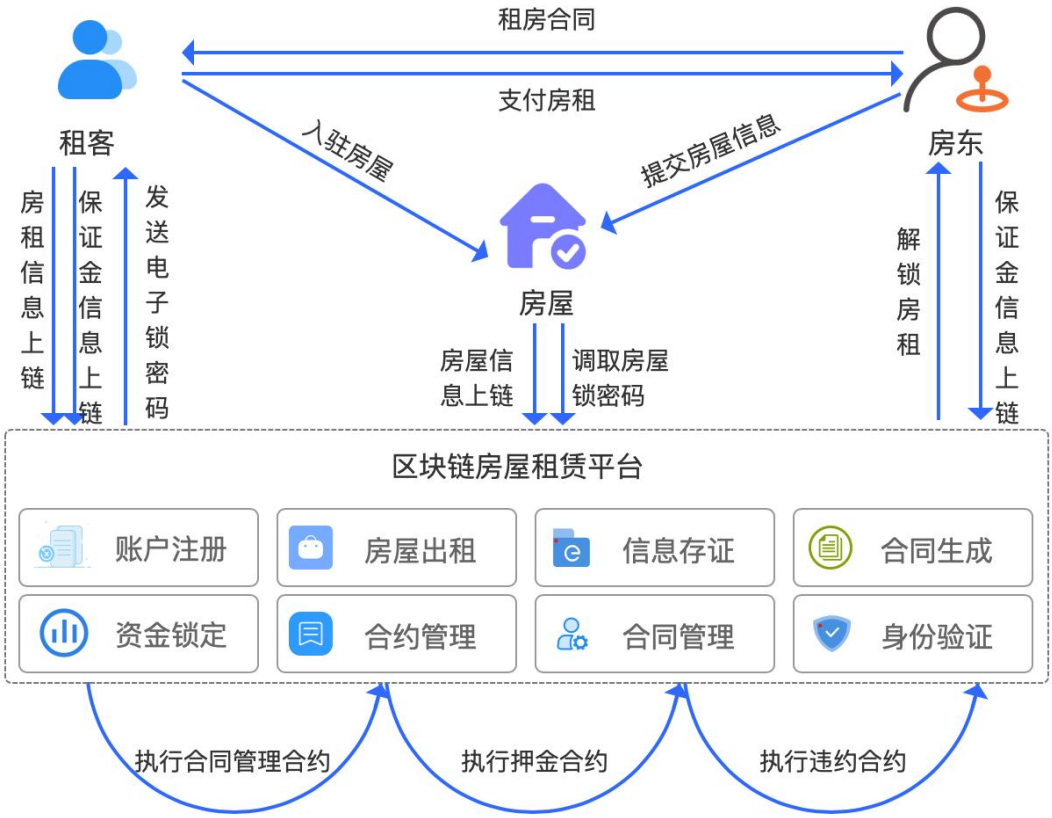
务

书

参赛队编号：_____

背景描述

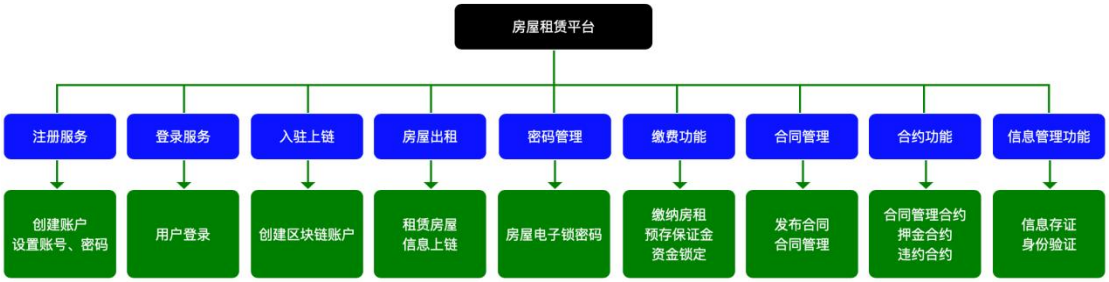
随着异地务工人员的增多，房屋租赁成为一个广阔市场；目前，现有技术中的房屋租赁是由房主发布租赁信息，租赁信息发布在房屋中介和/或租赁软件，租客获取租赁信息后，现场看房，并签订纸质的房屋租赁合同；房屋租赁费用通过中介或直接给房主；另外，后期的房租还需房主收取。



区块链房屋租赁平台业务流程图

现在利用区块链技术实现房屋租赁平台，将房东、房屋、租客加入到区块链网络中，将发布、租赁、合同、房租等信息存储在区块链的分布式网络中，永久有效，无法篡改。在房屋租赁场景中，房东将房屋租出后，无法有效控制房屋的使用权，比如租客未履行租赁合约相应内容时，房东无法及时有效限制租客对房屋的使用，造成租金或房屋使用权的回收困难。租赁合约履行过程中房东如果要求租客提前

搬离，也会造成租客的利益损失。现有一基于区块链的房屋租赁平台S，房东L在S中出租一套房屋，S中可以调用该房屋的电子锁E的密码。L在S中发放租房合同给租客T，默认L会发放合同，不考虑其他情况。合同中规定，房租每个月的费用是3000元，交付形式是押一付一，每个月的16号交房租。默认T会签署合同，不考虑其他情况。在T签署合同后的24小时之内，T需要在S中预存3000元保证金、3000元租金（需要按照本顺序声明），如果T未预存，交易失败。如果T预存了保证金和一个月租金，L需要在24小时内在S中预存3000元保证金，如果L未在规定时间内预存保证金，S将T的保证金与租金退回，如果L预存保证金，本交易开始执行。S自动将T的一个月房租转账给L，并将房屋E的密码发送给T。在每月16日，E会自动更换密码，如果T按时缴纳房租，S将房屋密码重新更新给T；当T到期未缴纳，S将T的保证金转账给L代替一个月的租金，S仍会将密码更新给T；如果保证金已被扣除后的月份T仍然到期未缴纳，等待补缴，T不补缴房屋密码变更会导致T进不去房屋。当未到租期L提前终止合同，S将L的保证金转给T，如果T还有保证金将保证金一并退还给T。当未到租期T提前终止合同，如果还有保证金S将T的保证金转给L，T还需赔偿L一个月租金，如果T保证金已被扣除，T需赔偿L两个月租金，S将L的保证金退还给L。默认T会赔偿租金不考虑其他情况。当合同正常到期，S将L的保证金退还，T如果还有也退还。本案例只考虑违反合同日期的情况，不考虑情况。



区块链房屋租赁平台系统架构图

模块一：区块链产品方案设计及系统运维（35 分）

选手完成本模块的任务后，将任务中软件建模图、配置文件、运行结果等截图（截图内容清晰且完整）和文字内容粘贴至作答区并提交。

任务 1-1：区块链产品需求分析与方案设计

房屋租赁平台中涉及到租客、房东、区块链租房平台、房屋等参与方，他们需要在区块链租房平台中完成账户注册、身份上链、出租房屋、生成合同等多种业务活动。通过对业务活动的功能分析，可以更好的服务系统的开发流程。基于房屋租赁平台系统架构，以区块链房屋租赁平台为背景，结合账户注册、登录服务、入驻上链、房屋出租、房租缴费等核心功能描述，撰写流程图/功能图、用例图等概要设计。

房屋租赁平台中涉及到租客、房东、区块链租房平台、房屋等参与方，他们需要在区块链租房平台中完成账户注册、身份上链、出租房屋、生成合同等多种业务活动。通过对业务活动的功能分析，可以更好的服务系统的开发流程。基于房屋租赁平台系统架构，以区块链房屋租赁平台为背景，结合账户注册、登录服务、入驻上链、房屋出租、房租缴费等核心功能描述，使用 Visio 绘制用例图、功能图、架构图等概要设计。

本任务需要依据项目背景完成需求分析与方案设计，具体要求如下：

1. 根据项目给定的背景描述和房屋租赁平台业务概览图，对房屋租赁平台进行分析，完成以下任务：

（1）编写用户群体需求分析，明确系统用户群体及其需求；

（2）绘制系统 UML 用例图，用例图中包含系统参与角色以及用例。

2. 依据给定的背景信息、房屋租赁平台业务概览图以及给出的房屋租赁平台的核心流程，使用 Visio 编制业务系统功能图；

表 1-1-1 房屋租赁平台的核心流程

发布租房合同流程	房东起草租房合同协议，填写房屋信息、租期、房租等。使用房东的私钥对租房合同进行签名并广播到区块链中进行存证
签署合同流程	房东对租房合同进行签名
缴纳房租流程	区块链房屋租赁平台节点实时房屋到期时间

3. 按照基础层、合约层、接口层以及应用层的结构来设计区块链系统的架构，其中在基础层需指明需要的节点、名称、协议、存储等信息，使用 Visio 绘制系统架构图。

任务 1-2：区块链系统部署与运维

围绕区块链房屋租赁平台部署与运维需求，进行项目相关系统、节点以及管理工具的部署工作。通过监控工具完成对网络、节点服务的监控。最终利用业务需求规范，完成系统日志、网络参数、节点服务等系统结构的维护，具体要求如下：

1. 根据参数与端口设置要求，部署区块链系统并验证；
2. 根据参数与端口设置要求，部署区块链网络管理平台并验证；
3. 基于区块链系统相关管理平台，按照任务指南实施系统运维工作并验证；
4. 基于区块链系统相关监管工具，按照任务指南对区块链系统进行监管。

子任务 1-2-1： 搭建区块链网络并验证

基于给定服务器环境以及软件（地址“/root/tools”），搭建单机、单机构、两群组、五节点的区块链系统并验证，具体工作内容如下：

机构	群组	节点	P2P 端口	channel 端口	rpc 端口
agency	group1	node0、node1、node2	30200	20200	8020
agency	group2	node3、node4	30300	20300	8030

区块链网络搭建信息表

（1）根据区块链网络搭建信息表编写 ipconf 配置文件搭建区块链网络，ipconf 文件内容和创建过程的输出结果截图保存；

（2）通过命令验证区块链节点进程运行状况，结果截图；

（3）通过命令验证区块链节点(node1)连接节点数和共识状态日志输出，结果截图。

子任务 1-2-2：搭建区块链控制台并验证

基于给定服务器环境以及软件（地址“/root/tools”），搭建区块链控制台并开展相关运维工作，具体工作内容如下：

（1）配置控制台，管理相关证书并启动，控制台启动结果截图；

（2）使用控制台部署 HelloWorld 智能合约，命令和结果截图；

（3）使用控制台完成 HelloWorld 智能合约的 set 与 get 操作，命令和结果截图；

（4）将控制台从 group1 切换到 group2，命令和结果截图。

子任务 1-2-3：区块链账户权限控制

基于已完成的区块链系统与控制台搭建工作,开展区块链账户权限管理等运维工作，具体内容如下：

（1）运行脚本创建三个新的账户(格式为 pem),使用账户 1(account1)指定群组 1 登录控制台,添加账户 1(account1)、账户 2(account2)和账户 3(account3)为委员并验证，过程和结果截图保存；

（2）修改账户 1 的票数为 2 并验证，修改投票阈值为 75%并验证，结果截图保存；

（3）撤销账户 3(account3)的委员权限，设置账户 3 为运维角色并验证，过程和结果截图保存。

子任务 1-2-4：区块链网络运维

根据任务描述要求，完成网络配置与管理运维操作，具体内容如下：

（1）设置区块链系统黑名单，将 node3 设为黑名单禁止并通过控制台验证 node3 的 Peers，配置文件和验证结果截图；

（2）通过给定工具（地址/root/tools）完成新节点（node5）创建，将新节点(node5)接入群组 1 和群组 2 并参加共识,通过日志信息验证新节点(node5)已经接入群组 1 和 2，过程和验证结果截图保存；

（3）在群组 1 中将 node1 设置为观察节点，验证结果并截图。

任务 1-3：区块链系统测试

基于 WeBASE 的部署脚本完成 WeBASE 环境搭建以及搭建结果验证，最后将执行结果截图保存。

(1) 实现 WeBASE 平台部署，访问 WeBASE 管理平台首页，截图保存；

(2) 使用 WeBASE-Sign 进行对数据 E7ADBEE5908D 进行签名，结果截图；

(3) 使用 WeBASE-Front 查询机器历史性能信息，结果截图。

序号	中文	参数名	类型	必填	说明
1	开始日期	beginDate	LocalDateTime	是	
2	结束日期	endDate	LocalDateTime	是	
3	对比开始日期	contrastBeginDate	LocalDateTime	否	
4	对比结束日期	contrastEndDate	LocalDateTime	否	
5	间隔	gap	int	否	默认为 1

机器历史性能查询参数表

模块二：智能合约开发与测试（30 分）

选手完成本模块的任务后，将任务中设计结果、运行代码、运行结果等截图（截图内容清晰且完整）粘贴至作答区并提交。

任务 2-1：智能合约设计

根据区块链房屋租赁平台产品需求分析和设计文档的描述，绘制智能合约 UML 时序图，编写该区块链产品的智能合约功能需求文档，具体要求如下：

1. 绘制区块链房屋租赁平台的智能合约 UML 时序图；

2. 结合区块链房屋租赁平台项目背景、概要设计、需求分析和功能设计等，编写区块链房屋租赁平台的智能合约功能需求文档。

任务 2-2：智能合约开发

使用 Solidity 语言进行智能合约开发，根据需求功能介绍在待补充源码中完成程序接口功能的编码，解决代码错误和警告，正确编译合约，功能调试正确，运行合约进行业务功能的验证，下列子任务中的合约编码表示合约中对应接口功能开发。

子任务 2-2-1：合同管理功能

根据需求功能介绍在待补充源码中完成合同管理功能的编码，解决代码错误和警告，正确编译合约，功能调试正确，运行合约中的房东签署合同、租金支付接口功能。

（1）编写房东签署合同接口，完成本合同位置只允许房东签署，通过合同中的信息生成租赁合同的链上哈希，触发协议签署合同的功能，其中合同中的信息包括房东链上账户、租客链上账户、租赁开始时间、租赁结束时间、月租金额、押金金额、交租时间，代码及调用结果截图保存；

表 2-2-1 Agreement 实体说明

名称	说明
landlord	房东的账户
tenant	租客的账户
rentAmount	每月租金金额（默认 3000）
depositAmount	押金金额（默认 3000）
rentDueDate	租金到期日（默认 16）
leaseStartDate	租赁开始日期
leaseEndDate	租赁结束日期
leaseDuration	租赁期限

（2）编写租金支付接口，完成只允许租客支付租金的规则，检查支付的租金金额是否正确，触发记录租金支付情况的功能，代码及调用结果截图保存。

子任务 2-2-2：违约管理功能

根据需求功能介绍在待补充源码中完成违约管理功能的编码,解决代码错误和警告，正确编译合约，功能调试正确，运行合约中的房东终止合同、租客终止合同接口功能。

（1）编写房东终止合同接口，实现房东终止合同判断，如果租客已经终止合同则合同无效，如果合同有效，对合同终止状态进行标记，将剩余押金退还给租客的功能，代码及调用结果截图保存；

表 2-2-2 Rental 实体说明

名称	说明
gracePeriod	宽限期
terminationFee	终止合同费用
contractActive	合同是否有效
landlordTerminated	房东是否终止合同
tenantTerminated	租客是否终止合同
tenantDefaulted	租客是否违约
securityReturned	押金是否已退还

（2）编写租客终止合同接口，实现租客终止合同判断，如果房东已经终止合同则合同无效，如果合同有效，对合同终止状态进行标记，将剩余押金退还给房东的功能,其中字段包括房东地址、租客地址、租金、押金、合同开始日期、合同结束日期、宽限期、终止合同费用、合同是否有效、房东是否终止合同、租客是否终止合同、租客是否违约、押金是否已退还，代码截图保存。

子任务 2-2-3：押金管理功能

根据需求功能介绍在待补充源码中完成押金管理功能的编码,解决代码错误和警告，正确编译合约，功能调试正确，运行合约中的租客缴纳押金情况查询、房东收取押金情况查询接口功能。

（1）编写房东收取押金情况查询接口，实现房东是否已收到押金的功能，代码及调用结果截图保存。

任务 2-3：智能合约测试

子任务 2-3-1：基于 Web 前置平台的合约测试

1．解决代码错误和警告，正确编译所有合约并部署合约，成功获取部署的合约地址和 abi，智能合约地址截图，abi 文件命名为【智能合约.abi】并保存至客户端桌面【工位号文件夹】下。

子任务 2-3-2：漏洞测试

分析下面漏洞智能合约,使用 WeBASE 进行漏洞复现,修复漏洞并使用 WeBASE 进行验证。

如下有问题的合约代码：

```
contract SimpleERC20 {
    address public owner;
    uint public total;
    mapping(address => uint) private balances;

    event Mint(address,uint);

    constructor() public {
        owner = msg.sender;
    }
```

```
        //bytes(mint_d22vi9okr4w(address))                                                    =
6D696E745F6432327669396F6B72347728616464726573732 9
        function mint_d22vi9okr4w(address _account) public {
            require(msg.sender == owner);
            require(balances[_account]+1000 > balances[_account] && total+1000 > total);
            balances[_account] +=1000;
            total +=1000;
            emit Mint(_account,balances[_account]);
        }

    //transfer(address,address,uint256)                                                       =
7472616E736665722861646472657373 2C616464726573732C75696E7432353629
    function transfer(address from,address to,uint amount) public {
            require(msg.sender==owner);
            require(balances[from] >= amount && balances[to] + amount > balances[to]);
            balances[from] -=amount;
            balances[to] +=amount;
        }

        function getBalance(address _account) public view returns(uint) {
            return balances[_account];
        }
    }

    contract factoryERC20{
        address public owner;

        constructor() public {
            owner = msg.sender;
        }

        function mint(address _token,address _account)public returns(bool){
            require(msg.sender == owner);
            (bool success, ) = address(_token).call(abi.encodePacked(bytes4(0x00000000),
abi.encode(_account)));
        }

        function createERC20() public returns (address){
            SimpleERC20 erc = new SimpleERC20();
            return address(erc);
        }

        function transfer(bytes memory _method ,address _token,address _to, uint _ammount)
public returns(bool){
```

```
                (bool success, ) =
address(_token).call(abi.encodePacked(bytes4(keccak256(abi.encodePacked(_method))),
abi.encode(msg.sender, _to, _ammount)));
                return success;
        }

        function getBalance(address _token,address _account) public returns(uint){
                SimpleERC20 erc = SimpleERC20(_token);
                return erc.getBalance(_account);
        }
    }
```

（1）分析智能合约中存在问题，并说明导致漏洞的原因及其危害；

（2）通过 WeBASE 调用智能合约，复现智能合约中存在的漏洞，结果截图；

（3）修复智能合约漏洞并测试，修复结果和测试结果截图。

模块三：区块链应用系统开发（30 分）

选手完成本模块的任务后，将任务中添加代码、Web 页面、运行结果等截图粘贴至作答区并提交。。

任务 3-1：区块链应用前端功能开发

在 user.vue 完成区块链应用系统的构建、服务器端（后端）与 Web 端（前端）的接口的联调。要求如下：

1. 使用 VsCode 工具，按照押金详情原型图的长度、宽度、行高、间距、文字样式、颜色等，完成押金详情页面的样式开发，将 Web 页面和代码截图保存；

2. 使用 VsCode 工具，完成 Vue 调用押金查询接口 API，获取接口返回的租房人、房屋位置、楼号、押金金额、收款人、收款日期信息，填充至 Vue 页面中，将 Web 页面和代码截图保存。

任务 3-2：区块链应用后端功能开发

子任务 3-2-1：区块链网络环境启动

区块链应用系统开发需要区块链底层网络进行支撑，完成和链上数据进行交互，与节点建立链接，完成链上信息查询。要求如下：

（1）使用 IntelliJ IDEA 工具，打开（BlockController.java）文件，在查询区块链信息接口中,使用 Java-SDK 获取区块链的最新高度和最新交易 Hash，并将结果按十进制的整数和字符串类型返回，将代码和结果截图保存。

子任务 3-2-2：编写签署房屋租赁合同合约接口和数据库设计

（1）根据“签署房屋租赁合同合约”中的字段，在 Java 项目中声明实体类（LeaseContract），将声明代码结果截图保存；

- ◆ 房东的账户地址（landlord）
- ◆ 租客的账户地址（tenant）
- ◆ 每月租金金额（monthlyRent）
- ◆ 押金金额（depositAmount）
- ◆ 租金到期日（rentDueDate）
- ◆ 租赁开始日期（leaseStartDate）
- ◆ 租赁结束日期（leaseEndDate）

（2）请打开 Navicat Premium 客户端，并连接到数据库，根据第一步中声明的实体类，创建一个名为" lease_contract "的数据库表，包含以下字段，并给出建表语句：

- ◆ 房东的账户地址（landlord）
- ◆ 租客的账户地址（tenant）
- ◆ 每月租金金额（monthly_rent）
- ◆ 押金金额（deposit_amount）
- ◆ 租金到期日（rent_due_date）
- ◆ 租赁开始日期（lease_start_date）
- ◆ 租赁结束日期（lease_end_date）

子任务 3-2-3：编写调用签署合同接口

已将押金管理合约部署至区块链，完成调用房东签署合同合约接口，要求如下：

（1）接受从 Web 端接收对应各种参数；

（2）调用智能合约 API，返回调用结果信息传递给前端页面,返回信息包括（massage:“创建成功”）以及交易成功后的返回信息如 input，transactionHash 等；

（3）签署合同成功后，对合同信息进行解析，并通过数据库依赖包（mysql-connector-java-bin.jar）存储到数据库中；

（4）使用 postman 测试功能完整性，测试参数和结果截图，调用房东签署合同合约接口部分的代码截图。