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

样题二

任

务

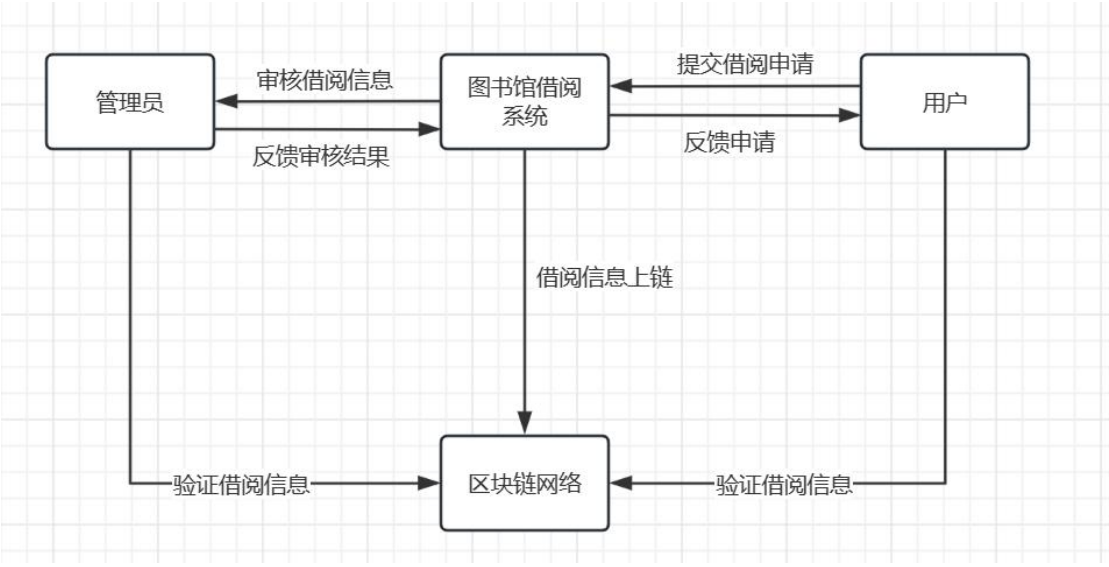
书

参赛队编号：_____

背景描述

2021 年 6 月，文化和旅游部印发《“十四五”公共文化服务体系建设规划》，提出“要以全国智慧图书馆体系建设项目为引领，运用 5G、云计算、大数据、人工智能、区块链等信息技术，搭建一套支撑智慧图书馆运行的云基础设施，打造面向未来的图书馆智慧服务体系和自有知识产权的智慧图书馆管理系统”。

基于以上背景需要我们设计一个基于区块链的智慧图书馆数字资源管理系统，将解决智慧图书馆建设进程中所遇到的数据安全难题、资源利用难题；在当今信息数字化的时代，图书馆已经演变成了数字资源的重要托管者和提供者。然而，随之而来的是对数据安全和隐私的不断担忧，以及如何更好地管理和优化数字资源的复杂性。传统的图书馆管理系统难以有效解决这些问题，因此我们需要采用更加创新的方法来满足不断增长的数字资源管理需求。



根据项目需求和团队实际情况，选择技术路线为 fisco bcos、solidity、truffle、js、Mysql、Vue、springboot、mybatis

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，将任务中设计结果、运行代码、运行结果等截图按要求进行提交。

任务 1-1：区块链产品需求分析与方案设计

【任务要求】

传统的图书管理系统需要管理员来验证和审阅，而在区块链网络的图书管理系统中也是分为管理员、用户、区块链网络图书管理注册系统（也就是区块链）。

通过对业务需求的功能分析，可以更好的服务系统的开发流程。基于应用程序的功能需求设计文档，完善功能需求设计文档，具体要求如下：

（1）根据功能需求设计文档中的功能需求表画出整体业务流程图；

（2）根据系统背景描述和需求分析，画出该系统的功能模块图。

任务 1-2：区块链系统部署与运维

子任务 1-2-1：搭建区块链系统

【任务要求】

机构 A、B、C 希望同时构建一个区块链系统，其网络端口要求如下：

请基于提供的 Linux 操作系统和相关软件包，按照要求完成区块链系统搭建，要

机构名称	节点数	群组	P2P 端口	channel 端口	rpc 端口
机构 A	1	1	30100	20100	8010
机构 B	1	1	30200	20200	8020
机构 C	2	1、2	30300	20300	8030

求如下：

（1）在“/root/tools/”目录下构建区块链节点，组网配置文件以“ipconf”命名；

（2）在“/root/tools/”目录下启动区块链节点；

（3）在“/root/tools/”目录下查看节点（node0）的进程运行情况、节点连接状态和共识状态（要求节点状态输出信息不少于三条）。

子任务 1-2-2：控制台配置与使用

【任务要求】

控制台时区块链系统重要的交互工具，它通过 Java SDK 与区块链节点建立连接，实现区块链节点数据读写的访问请求。基于已搭建的区块链系统，为区块链系统（node0）配置控制台，使用控制台与区块链节点进行简单交互，具体要求如下：

（1）在/root/tools/目录下配置控制台相关证书，并启动控制台；

（2）通过控制台查询节点版本信息；

（3）使用控制台部署 HelloWorld 合约；

（4）使用控制台分别完成 HelloWorld 合约的 set（参数为 FISCO BCOS）与 get 操作；

（5）查询部署合约的区块信息；

（6）将交易执行允许消耗的最大 gas 数设置为 500000000，并验证。

子任务 1-2-3：区块链系统扩容

【任务要求】

采用群组架构的网络中，根据业务场景的不同，可存在多个不同的账本，区块链节点可以根据业务关系选择群组加入，参与到对应账本的数据共享和共识过程中。基于已完成的区块链系统，现为机构 A 扩容一个新节点（newnode），端口号依次为 30400,20400,8040，具体内容如下：

（1）在“/root/tools/”目录下生成新节点(newnode)证书；

（2）在“/root/tools/”目录下准备新节点(newnode)配置文件；

（3）在“/root/tools/”目录下启动新节点，并查看新节点的连接数；

（4）将新节点作为观察节点加入 group1 当中，并验证是否加入成功。

子任务 1-2-4：区块链可视化平台搭建

【任务要求】

WeBASE 是在区块链应用和区块链节点之间搭建的一套通用组件，能够提高区块链应用的开发效率。为可视化分析链上数据和实时监控区块链系统，搭建区块链可视化平台 WeBASE，具体要求如下：

（1）配置 MySQL，将 root 用户身份验证设置为“mysql_native_password”，并设置密码为“123456”；

（2）修改配置文件，配置 WeBASE 子系统数据库存储，基于已搭建的区块链系统部署 WeBASE 可视化平台；

（3）启动可视化平台，并检查各子系统进程；

（4）WeBASE 管理平台的初始账户为 admin，密码为 Abcd1234。

任务 1-3：区块链系统测试

子任务 1-3-1：区块链系统压力测试

【任务要求】

Caliper 是一个通用区块链性能测试工具，能够方便地对接多种区块链平台模拟合约转账及增删改查的压力测试，并输出的可视化性能测试报告。使用 Caliper 测试工具通过调用 HelloWorld 合约，新建一条默认链（单机 4 节点）来进行区块链系统进行压力测试，具体要求如下：

（1）编写核心测试用例代码 get.js 和 set.js（参数可自定义）；

（2）设置交易发送数量为 1000，交易发送速率为 100；

（3）查看可视化测试报告。

子任务 1-3-2：智能合约安全漏洞测试

【任务要求】

智能合约漏洞事件严重威胁着区块链生态安全，一旦智能合约部署到区块链上，就很难甚至无法进行修补，常见的合约漏洞有整数溢出、重入攻击、访问控制等。

基于给定的智能合约代码，使用 truffle 完成智能合约安全漏洞测试，具体要求如下：

（1）分析智能合约可能面临的安全威胁问题；

（2）在“Blocker.sol”合约中编写攻击合约，需先在构造函数中初始化 Blocker 合约实例，并提供 2 个方法，分别对漏洞合约中的两个竞猜方法进行攻击；

（3）在“1_test.js”中编写合约测试脚本，要求在每次测试前部署一个新的合约，并对合约攻击进行测试。

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、运行结果等截图粘贴至作答区并提交。

任务 2-1：智能合约设计

【任务要求】

根据区块链图书管理系统功能需求设计文档的描述，画出业务系统的时序图。具体要求如下：

（1）请根据该项目智能合约的设计梳理出业务逻辑的时序图

任务 2-2：智能合约开发

子任务 2-2-1：图书信息（Book）合约接口编码

【任务要求】

使用 Solidity 语言完成图书信息智能合约开发，具体要求如下：

（1）编写图书信息实体接口，完成图书信息初始化，实现图书信息的上链功能：

名称	类型	说明
name	string	图书名称
author	string	图书作者
number	string	图书编号
currentState	states	图书状态

records	address[]	图书借阅记录

```
contract Book{
    enum states {Library, loaned}//图书的当前状态，在库中、已借出
    //①图书名称
    //②图书作者
    //③图书编号
    //④图书状态
    //⑤图书借阅记录，保存所有借过本书的用户地址信息
}
```

（2）编写添加图书借阅记录接口，将借书人的地址记录到合约中：

```
    function addRecords(address _borrower) public {
        //①
    }
```

子任务 2-2-2：学生信息（Student）合约接口编码

【任务要求】

使用 Solidity 语言完成学生信息智能合约开发，具体要求如下：

（1）编写添加学生借阅记录接口，将图书地址添加到合约中：

```
    function addRecords(address _book) public {
        //①
    }
```

子任务 2-2-3：图书管理员（Librarian）合约接口编码

【任务要求】

使用 Solidity 语言完成图书管理员智能合约开发，具体要求如下：

（1）编写创建新图书的接口，完成新图书的创建以及合约状态变量的修改：

```
    function createBook(string memory _name, string memory _author, string memory _number) public returns(address){
        Book book = ①;
        //②
```

```
        //③

        return ④;
    }
```

（2）编写删除图书的接口，根据图书编号删除对应的图书信息，并将记录中后续的图书信息往前补位：

```
    function deleteBook(uint256 _bookNumber) public {
        //①
        //②
        for (uint256 i = ③; i < ④; i++){
            //⑤
        }
        //⑥
    }
```

（3）编写查询学生借阅记录的接口，根据学生 id，查询该学生借阅过的所有图书的信息：

```
    function getStudentRecords(string memory _id) public view
returns(string[] memory, string[] memory, string[] memory, address[] memory){
        require(address(students[_id]) != address(0), "无效的学生地址");

        address[] memory addr = ①;
        string[] memory name = new string[](addr.length);
        string[] memory author = new string[](addr.length);
        string[] memory number = new string[](addr.length);

        //循环遍历借阅数组，将图书信息写入对应数组
        for(uint i = 0; i < addr.length; i++){
            Book book = ②;
            //③
        }

        return (name, author, number, addr);
```

```
}
```

（4）编写借阅图书的接口，学生从库中借走图书，完成相应合约状态变量的修改：

```
function Borrowing(string memory ①, uint256 ②) public {
    Book book = books[_bookNumber];
    Student student = students[_id];
    require(③ < 5, "已借阅 5 本图书，不能继续借阅");
    require(_bookNumber < bookNumber, "无效的图书编号");
    //④

    //⑤
    //⑥
    //⑦
    //⑧

    //⑨
}
```

（5）编写归还图书的接口，，学生归还图书，完成相应操作的上链：

```
function returnBook(string memory _id, uint256 _bookNumber) public {
    Book book = books[_bookNumber];
    require(_bookNumber < bookNumber, "无效的图书编号");
    require(address(students[_id]) != address(0), "无效的学生");
    require(book.getStatus() != Book.states.Library, "当前图书已在库中，无法归还");
    require(① == address(students[_id]), "错误的图书借阅人");

    //②
    //③

    //④
}
```

子任务 2-2-4：智能合约编译与部署

【任务要求】

（1）解决代码错误和警告，正确编译并部署合约，成功获取部署的合约地址和 abi；

（2）调用图书管理智能合约的接口，完整验证业务流程。

任务 2-3：智能合约测试

【任务要求】

编写智能合约单元测试代码并完成合约功能测试、性能测试，具体要求如下：

1.补全 createBook 和 createStudent 方法

基于 VSCODE 加载的测试项目，补全位于 test 文件夹中 Book.js 文件，添加测试用例，测试智能合约的 createBook 和 createStudent 方法。

2.测试 Borrowing 和 returnBook 方法

基于 VSCODE 加载的测试项目，补全位于 test 文件夹中 Book.js 文件，添加测试用例，测试智能合约的 Borrowing 和 returnBook 方法。

3.测试 getBookRecords 和 getStudentRecords 方法

基于 VSCODE 加载的测试项目，补全位于 test 文件夹中 Book.js 文件，添加测试用例，测试智能合约的 getBookRecords 和 getStudentRecords 方法。

4.运行测试脚本，完成项目测试

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，将任务中设计结果、运行代码、运行结果等截图粘贴至作答区并提交。

任务 3-1：区块链应用前端开发

子任务 3-1-1：图书管理系统库存管理界面开发

【任务要求】

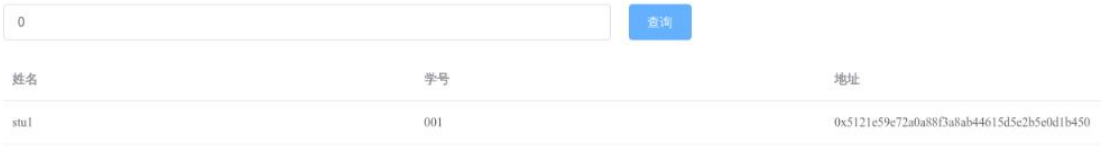
实现图书库存管理功能，具体展示内容有序号，书名，作者，当前借阅状态等

子任务 3-1-2：图书管理系统借阅模块前端开发

【任务要求】

结合给定区块链业务需求和前端页面示例，完成前端首页页面，具有如下功能：

1. 根据输入的图书编号，查询相应图书的借阅历史信息；



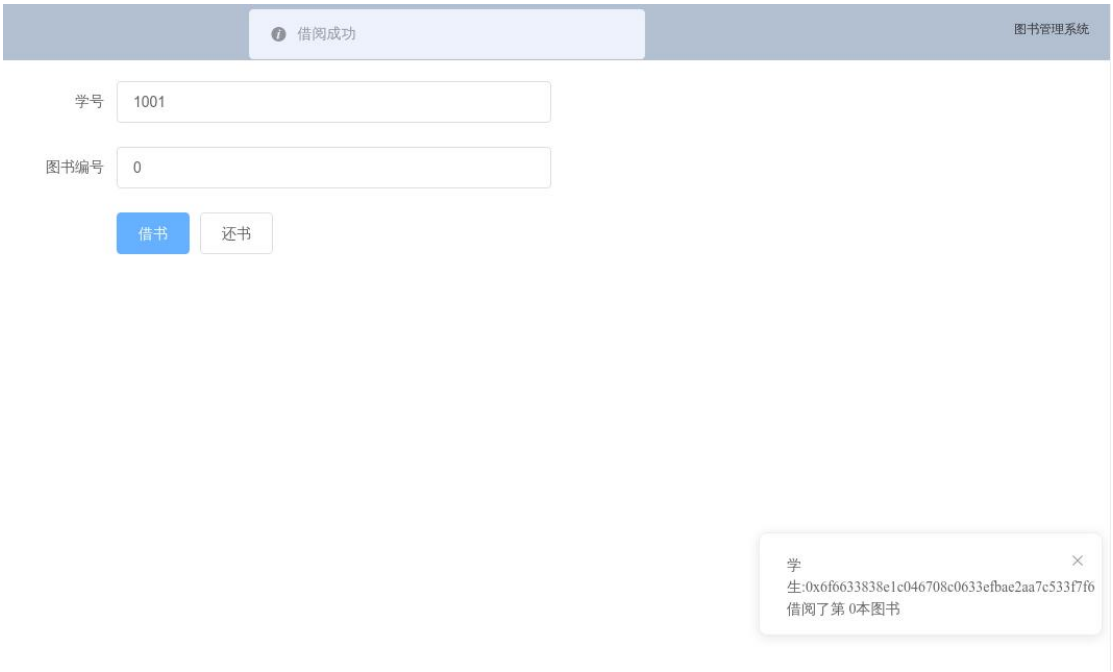
2. 根据输入的学生编号，查询相应借阅历史信息；

子任务 3-1-3：图书管理系统借阅消息通知功能开发

【任务要求】

结合给定区块链业务需求和前端页面示例，完成前端页面消息通知功能，具有如下功能：

1. 使用 websocket 与后端建立连接，即时展示借阅图书通知信息；



任务 3-2：区块链应用后端开发

使用 Java-SDK 与区块链进行交互，将 Solidity 智能合约转译为可供 Java 调用的文件，实现区块链编程。

子任务 3-2-1：创建学生

【任务要求】

使用 Java 语言编写后端代码进行交互，创建学生信息，要求如下：
1. 函数成功创建学生信息；
2. 函数有调用异常处理，若合约交互失败需返回失败原因。

子任务 3-2-2：创建图书

【任务要求】

使用 Java 语言编写后端代码进行交互，创建图书信息，要求如下：

1. 函数成功创建图书信息；
2. 函数有调用异常处理，若合约交互失败需返回失败原因。

子任务 3-2-3：图书列表

【任务要求】

开发图书管理系统中的接口，根据系统中原有的代码补充 getAllBookMessage 接口，实现全部图书信息的查询；

子任务 3-2-4：借阅图书

【任务要求】

开发图书管理系统中的接口，根据系统中原有的代码补充 borrowing 接口，实现对图书的借阅；

子任务 3-2-5：归还图书

【任务要求】

开发图书管理系统中的接口，根据系统中原有的代码补充 returnBook 接口，实现对图书的归还；

子任务 3-2-6：监听图书管理系统借阅事件

【任务要求】

开发图书管理系统中的接口，根据系统中原有的代码补充监听借阅事件接口，实现对借阅图书事件的实时监听,要求只监听借阅事件,对事件消息进行解码处理；

借阅事件定义为：event borrowingBookEvent(address stuAddr, uint256 bookNumber)；

子任务 3-2-7：持久化借阅事件信息

【任务要求】

基于后端系统的开发模板，补充对应的代码逻辑，完成后端代码逻辑，实现对借阅事件数据的持久化，要求包含事件名称，事件参数，创建事件三个字段；

子任务 3-2-8：实时借阅消息通知

【任务要求】

开发图书管理系统中的接口，根据系统中原有的代码补充监听长连接功能接口，实现对借阅事件的实时播报，要求管理订阅的客户端，对客户端发送借阅通知消息；

子任务 3-2-9：完整验证业务功能流程

【任务要求】

联合前端页面，依次验证 图书管理-借阅图书-借阅消息通知-借阅记录 业务功能流程