

竞赛说明

一、竞赛内容

本模块包括智能合约开发、区块链应用后端功能开发和区块链应用前端页面开发，共三个任务。

二、竞赛时间

竞赛时长：共计 120 分钟。

三、注意事项

1. 参赛选手不得携带通信设备等物品进入赛场，严禁在程序及运行结果中任何位置标注竞赛队的任何信息，违反者按作弊处理。
2. 请根据大赛所提供的竞赛环境，检查所列的硬件设备、软件清单、材料清单是否齐全，计算机设备是否能正常使用。
3. 竞赛结束前，整合各试题的操作过程和结果数据，并提交完整的文档、代码、截图等竞赛结果到指定位置中。
4. 竞赛结束时，后台统一关闭答题通道，禁止将比赛所用的所有物品（包括试卷和稿纸）带离赛场。

模块 D: 区块链应用操作

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比赛平台左侧“答题报告”中对应的任务序号下。

环境说明

任务 1	智能合约相关文件所在路径	/root/Desktop/contracts
任务 2	后端 java 相关文件所在路径	/root/Desktop/EndowmentInsurance
任务 3	前端页面相关文件夹所在路径	/root/Desktop/front

注意：启动后端程序前，可使用 Navicat 链接数据库并导入桌面上的 sql 文件生成相关数据表。

任务 1: 智能合约开发

使用 Solidity 编程语言开发智能合约，包括存储合约设计开发，接口合约设计开发，最终实现区块链社保功能。针对 Solidity 智能合约进行代码检查和测试，验证函数执行结果是否与预期结果相符。

子任务 1-1: 平台合约接口编码

【任务要求】

1. 编写雇主余额充值接口 `addEmployerBalance`，判断充值账户是否合法并进行雇主余额充值，返回操作状态，实现雇主余额充值上链功能：
2. 编写办理职工养老保险接口 `handleEmployeeInsurance`，实现职工养老保险上链功能：
3. 编写社保账户转移接收或拒绝申请接口 `receiveApplication`，实现职工养老保险账户转移功能：

【操作说明】

1. 基于浏览器访问“http://127.0.0.1:5002/WeBASE-Front/”的节点前置平台来实现该任务；
2. 合约原文件位于“/root/Desktop/contracts/”目录下；
3. 在“InsurePlatform.sol”文件中进行代码编写；
4. 根据题目要求补充代码，并将补全代码截图粘贴至“答题报告”处。

子任务 1-2：社保账户合约接口编码

【任务要求】

1. 编写职工社保账户充值接口 recharge，实现社保账户缴费功能并返回当前社保账户余额；
2. 编写职工社保账户绑定雇主接口 bindEmployer，实现社保账户雇主更换功能；

【操作说明】

1. 基于浏览器访问“http://127.0.0.1:5002/WeBASE-Front/”的节点前置平台来实现该任务；
2. 合约原文件位于“/root/Desktop/contracts/”目录下；
3. 在“SecurityBureauSo.sol”文件中进行代码编写
4. 根据题目要求补充代码，并将补全代码截图粘贴至“答题报告”处。

任务 2：区块链应用后端功能开发

子任务 2-1：后端登录功能开发

【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，实现对后端系统的登录功能，并测试功能完整性，题目具体要求如下：

1. 根据用户注册信息中的用户名和账户地址从数据库中查询账户信息是否存在；
2. 登录成功向前端系统返回登录状态信息（token）；
3. 使用 postman 进行登录接口测试。

【操作说明】

1. 使用编程工具IntelliJ IDEA community Edition（位于“/root/Desktop/”目录下）
打开项目工程文件夹（位于“/root/Desktop/”目录下）；
2. 类UserServiceImpl.java所在的目录为：
“EndowmentInsurance/src/main/java/org/zh/blockchain/insure/service/impl/UserServiceImpl.java”；
3. postman接口测试工具位于“/root/Desktop/”目录下；
4. 数据库启动命令：service mysql start；
5. 根据题目要求补充代码，并将补全代码截图粘贴至“答题报告”处。

子任务 2-2：后端账户个人信息功能开发

【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，实现对后端系统的获取账户个人信息功能，并测试功能完整性，题目具体要求如下：

1. 根据账户 id 从数据库中查询账户基本信息，并判断账户是否存在；
2. 获取雇主角色的详细信息，并封装到“UserInfoVo”中；
3. 使用 postman 进行获取个人信息接口测试（“/user/user_info”）。

【操作说明】

1. 使用编程工具IntelliJ IDEA community Edition（位于“/root/Desktop/”目录下）
打开项目工程文件夹（位于“/root/Desktop/”目录下）
2. 类UserServiceImpl.java所在的目录为：
“EndowmentInsurance/src/main/java/org/zh/blockchain/insure/service/impl/UserServiceImpl.java”
3. postman接口测试工具位于“/root/Desktop/”目录下；
4. 根据题目要求补充代码，并将补全代码截图粘贴至“答题报告”处。

子任务 2-3：后端社保账户开通功能开发

【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，实现对后端系统的社保账户开通功能，并测试功能完整性，题目具体要求如下：

1. 核查职工是否已办理社保账户；

2. 将社保账户信息和银行办理记录存入对应数据库中；
3. 使用 postman 进行社保账户开通接口测试。

【操作说明】

1. 使用编程工具IntelliJ IDEA community Edition（位于“/root/Desktop/”目录下）
打开项目工程文件夹（位于“/root/Desktop/”目录下）
2. 类BankServiceImpl.java所在的目录为：
“EndowmentInsurance/src/main/java/org/zh/blockchain/insure/service/impl/BankServiceImpl.java”
3. postman接口测试工具位于“/root/Desktop/”目录下；
4. 根据题目要求补充代码，并将补全代码截图粘贴至“答题报告”处。

任务 3：区块链应用前端页面开发

子任务 3-1：注册页面前端开发

【任务要求】

请基于前端系统的开发模板，在注册页面中添加对应的代码逻辑，实现对前端系统的注册功能，并测试功能完整性；

题目的具体要求如下：

1. 注册角色的数据实现双向绑定，并通过单选选择角色；
2. 向后端系统发送用户输入的注册信息以进行注册请求，注册成功并给出相应提示，再跳转到登录页面；
3. 验证页面注册功能完整性。

【操作说明】

1. 使用位于“/root/Desktop/”目录下的visual studio code工具打开位于“/root/Desktop/”目录下的front项目文件夹，在位于“/src/views/register”目录下的index.vue文件中完成该任务；
2. 前端默认登录管理员账户：admin 密码：123456；
3. 根据题目要求补充代码，并将补全代码截图粘贴至“答题报告”处。

子任务 3-2：登录页面前端开发

【任务要求】

请基于前端系统开发模板，在登录页面中添加对应的代码逻辑，实现前端系统登录功能，并测试功能完整性：

题目的具体要求如下：

1. 用户输入的登录用户名和密码通过双向绑定存放在 `user` 中；
2. 将用户输入的登录信息发送到后端进行登录请求，并在成功后保存 `token`，登录成功并给出相应提示，再跳转到首页页面；
3. 验证页面登录功能完整性。

【操作说明】

1. 使用位于 `"/root/Desktop/"` 目录下的 `visual studio code` 工具打开位于 `"/root/Desktop/"` 目录下的 `front` 项目文件夹，在位于 `"/src/views/login"` 目录下的 `index.vue` 文件中完成该任务；
2. 前端默认登录管理员账户：`admin` 密码：`123456`；
3. 根据题目要求补充代码，并将补全代码截图粘贴至“答题报告”处。