 

**实 验 报 告**

|  |  |
| --- | --- |
| **课程名称：** | **《区块链技术与应用》** |
| **学生姓名：** |  |
| **学生学号：** |  |
| **学生专业：** | **软件工程** |
| **授课教师：** | **许可** |
| **开课学期：** | **2024-2025-1** |

**华南理工大学软件学院**

**2024 年 9月**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **实验地点：** | B7后座 1楼 | 138房 | | |
| **实验日期与时间：** | 2024年9月28日1-4节 | | | |
| **实验教师：** | 郭 锐、许可 | |  |  |
| **总体批改意见：** |  | | | |

**实验得分：**

|  |  |  |  |
| --- | --- | --- | --- |
| 实验1评分 | 实验2评分 | 实验3评分 | 总分 |
| /40 | /40 | /20 | /100 |

(\*\*\*\*\*\*注意：本页内容学生请勿改动\*\*\*\*\*\*\*\*)

**实验一 区块链理解与实践（40分）**

**准备实验环境（8分）**

1. 你实验用的操作系统是 。（2分）
2. 你的实验环境中是否已经安装必须使用的依赖（如curl、openssl、wget、openjdk8）？若无，请给出你的安装步骤。（2分）
3. 所安装的Java版本是： （请提供java -version的运行截图）（2分）
4. 你的实验环境中是否已经安装了VS Code或者IntelliJ IDEA等必要的开发IDE？若无，请给出你的安装步骤。（2分）

**在命令行搭建FISCO BCOS区块链（15分）**

1. 你创建的工作目录是： （请填写绝对路径），创建并进入该目录需要哪些命令？（2分）
2. 你是如何下载build\_chain.sh脚本的？使用了哪些命令？（5分）
3. 你指定搭建区块链时的IP端口是什么？（3分）
4. 使用build\_chain.sh脚本以后，命令行输出了什么？请简述。（3分）
5. 你使用了哪些命令启动了区块链？请简述你是如何检查区块链启动正常的。（2分）

**使用控制台部署调用HelloWorld合约（12分）**

1. 使用哪些命令下载控制台？（2分）
2. 你是怎么配置控制台的？请截图并描述。（3分）
3. 请尝试描述以下HelloWorld合约所实现的功能。（3分）

pragma solidity >=0.6.10 <=0.8.26;

contract HelloWorld {

string name;

constructor() public {

name = "Hello, World!";

}

function get() public view returns (string memory) {

return name;

}

function set(string memory n) public {

name = n;

}

}

1. 请提供步骤截图，描述如何使用控制台部署HelloWorld合约、调用HelloWorld合约set接口。在调用set接口时，要求填入自己的学号。（4分）

**PBFT共识算法容错模拟实践（5分）**

1. 通过逐渐停掉其中1个节点、2个节点，观察是否还能成功部署HelloWorld合约的截图，在控制台中输入getSyncStatus命令查看节点数量。（3分）
2. 尝试分析上一步停掉部分节点后，操作成功/失败的原因（2分）

**实验二 使用智能合约与区块链交互（40分）**

**准备Solidity开发环境（10分）**

搭建Solidity开发环境有以下几种方式：

* 使用Remix 在线/本地搭建的方式，链接：[在线](https://remix.ethereum.org/)、[本地](https://github.com/ethereum/remix-project)
* 使用VSCode + 插件的方式
* 使用IntelliJ IDEA + 插件的方式

1. 你使用的是什么方式搭建的？请截图描述步骤。（8分）
2. 你使用的Solidity版本是什么版本？（2分）

**实现ERC20标准的合约ERC20Test（25分）**

1. 根据ERC20标准的接口定义，编写智能合约，合约名为ERC20Test，要求实现mint、transfer、approve、transferFrom、balanceOf接口（15分）

参考内容链接：<https://ethereum.org/zh/developers/docs/standards/tokens/erc-20/#body>

|  |
| --- |
| ERC-20（以太坊意见征求 20）由 Fabian Vogelsteller 提出于 2015 年 11 月。这是一个能实现智能合约中代币的应用程序接口标准。  ERC-20 的功能示例包括：   * 将代币从一个帐户转到另一个帐户 * 获取帐户的当前代币余额 * 获取网络上可用代币的总供应量 * 批准一个帐户中一定的代币金额由第三方帐户使用 |

如果智能合约实施了下列方法和事件，它可以被称为 ERC-20 代币合约，一旦部署，将负责跟踪以太坊上创建的代币。

|  |
| --- |
| // 接口定义  function name() public view returns (string)  function symbol() public view returns (string)  function decimals() public view returns (uint8)  function totalSupply() public view returns (uint256)  function balanceOf(address \_owner) public view returns (uint256 balance)  function transfer(address \_to, uint256 \_value) public returns (bool success)  function transferFrom(address \_from, address \_to, uint256 \_value) public returns (bool success)  function approve(address \_spender, uint256 \_value) public returns (bool success)  function allowance(address \_owner, address \_spender) public view returns (uint256 remaining)  // 事件  event Transfer(address indexed \_from, address indexed \_to, uint256 \_value)  event Approval(address indexed \_owner, address indexed \_spender, uint256 \_value) |

1. 在合约中使用event、struct、mapping、require、modifier等语言特性。（10分，每个特性2分）

**在FISCO BCOS上部署调用合约ERC20Test（5分）**

在控制台中部署ERC20Test合约，调用合约的mint接口，给自己账号中发100个资产，并使用balanceOf接口查看结果（5分）

**实验三 在区块链上构建应用系统（20分）**

**将实验2的智能合约编译成Java接口文件（10分）**

参考文档链接，将实验2的智能合约编译成Java接口文件。（10分）

参考链接：[生成智能合约的Java接口文件](https://fisco-bcos-doc.readthedocs.io/zh-cn/latest/docs/sdk/java_sdk/contracts_to_java.html)

**使用Spring-boot-crud框架构建服务调用区块链（8分）**

参考demo项目：[FISCO-BCOS/spring-boot-crud](https://github.com/FISCO-BCOS/spring-boot-crud)

1. 根据demo教程指导，编译构建智能合约项目。请给出智能合约项目创建完成后，项目结构的截图，要求展开其中的src目录。（4分）
2. 简要说明项目中的代码项目结构，说明每一级目录代表的功能。（4分）

**在命令行发起HTTP请求调用服务接口（2分）**

根据demo教程指导，编译项目，并启动服务。在终端中或者其他工具发起HTTP请求到服务中，截图展示结果。

**Extra Bonus（非必做题）：仿造Spring-boot-crud的写法构建ERC20Test合约的Java服务（10分）**

1. 使用本实验第一步骤生成的Java 接口文件ERC20Test.java，新增到spring-boot-crud项目中contract文件夹。（1分）
2. 实现ERC20TestClient类，实现5个调用合约的基本接口：mint、transfer、approve、transferFrom、balanceOf。将这个类新增到项目client文件夹。（2分）
3. 实现ERC20TestController类，用于控制HTTP请求接口调用到ERC20TestClient的方法。将这个类新增到项目controller文件夹。（5分）
4. 尝试编译启动服务，并发起HTTP请求到新增接口中。（2分）