项目说明

一、项目简介

本项目是一个简单的区块链系统，旨在通过学习和实践，掌握区块链的基本原理和实现方法。项目使用 Rust 语言实现，包含区块生成、区块链存储、区块添加与验证、简单的挖矿机制等功能。

二、项目结构

项目主要包含以下模块：

1. **区块模块（block.rs）**
    - 定义区块结构体，包含区块头和区块体。
    - 实现区块的生成和哈希计算。
2. **区块链模块（blockchain.rs）**
    - 定义区块链结构体，包含区块链数据存储。
    - 实现创世区块生成、区块添加与验证等功能。
3. **挖矿模块（mining.rs）**
    - 实现简单的工作量证明（Proof of Work）算法。

三、功能实现

1. 区块结构设计

定义区块结构体，包含区块头和区块体。

```rust
// filepath: blockchain/src/block.rs

use std::time::{SystemTime, UNIX_EPOCH};

#[derive(Debug, Clone)]
pub struct Block {
    pub index: u32,
    pub timestamp: u64,
    pub previous_hash: String,
    pub hash: String,
    pub data: String,
}

impl Block {
    pub fn new(index: u32, previous_hash: String, data: String) -> Self {
        let timestamp =
SystemTime::now().duration_since(UNIX_EPOCH).unwrap().as_secs();
        let hash = calculate_hash(index, timestamp, &previous_hash,
&data);
```

```
        Block {
            index,
            timestamp,
            previous_hash,
            hash,
            data,
        }
    }
}

fn calculate_hash(index: u32, timestamp: u64, previous_hash: &str, data:
&str) -> String {
    format!("{:x}", md5::compute(format!("{}{}{}{}", index, timestamp,
previous_hash, data)))
}
```

2. 创世区块生成

实现生成创世区块的函数。

```
// filepath: blockchain/src/blockchain.rs

use crate::block::Block;

pub struct Blockchain {
    pub chain: Vec<Block>,
}

impl Blockchain {
    pub fn new() -> Self {
        let genesis_block = Block::new(0, String::from("0"),
String::from("Genesis Block"));
        Blockchain {
            chain: vec![genesis_block],
        }
    }
}
```

3. 区块添加与验证

实现添加新块到区块链的函数，并验证区块的有效性。

```
// filepath: blockchain/src/blockchain.rs

impl Blockchain {
    pub fn add_block(&mut self, data: String) {
        let previous_block = self.chain.last().unwrap();
        let new_block = Block::new(previous_block.index + 1,
previous_block.hash.clone(), data);
```

```
        self.chain.push(new_block);
    }

    pub fn is_valid(&self) -> bool {
        for i in 1..self.chain.len() {
            let current_block = &self.chain[i];
            let previous_block = &self.chain[i - 1];
            if current_block.previous_hash != previous_block.hash {
                return false;
            }
            if current_block.hash != calculate_hash(current_block.index,
current_block.timestamp, &current_block.previous_hash,
&current_block.data) {
                return false;
            }
        }
        true
    }
}
```

4. 挖矿机制

实现简单的工作量证明算法。

```
// filepath: blockchain/src/mining.rs

use crate::block::Block;

impl Block {
    pub fn mine_block(&mut self, difficulty: usize) {
        let mut nonce = 0;
        loop {
            let hash = calculate_hash(self.index, self.timestamp,
&self.previous_hash, &self.data, nonce);
            if &hash[..difficulty] == "0".repeat(difficulty) {
                self.hash = hash;
                self.nonce = nonce;
                break;
            }
            nonce += 1;
        }
    }
}

fn calculate_hash(index: u32, timestamp: u64, previous_hash: &str, data:
&str, nonce: u32) -> String {
    format!("{:x}", md5::compute(format!("{}{}{}{}{}", index, timestamp,
previous_hash, data, nonce)))
}
```

四、运行结果

1. 创世区块生成

运行代码生成创世区块，输出结果如下：

```
Block {
    header: BlockHeader {
        nonce: 0,
        time: 1741505102,
        bits: 553713663,
        txs_hash:
"fb8cde2d50d0d48bce64c96c507194823a5b6dcb46c90034fda8f0f6b0dc6656",
        pre_hash:
"22caaf24ef0aea3522c13d133912d2b722caaf24ef0aea3522c13d133912d2b7",
    },
    tranxs: [
        Transaction {
            nonce: 0,
            amount: 0,
            fee: 0,
            from: "0x0000",
            to: "0x0000",
            sign: "创世区块",
            hash:
"fb8cde2d50d0d48bce64c96c507194823a5b6dcb46c90034fda8f0f6b0dc6656",
        },
    ],
    hash:
"12619cb2400ec66dfd39496f806b83045600b978b8292e628d4473e25577e9df",
}
```

2. 添加新块

添加新块到区块链，输出结果如下：

```
Block {
    header: BlockHeader {
        nonce: 0,
        time: 1741505102,
        bits: 553713663,
        txs_hash:
"20c4a3b94c760d6a9f0bd9b2cd0dcb683cd6ad57d8c8fd3287df2df2eb5c1d3c",
        pre_hash:
"12619cb2400ec66dfd39496f806b83045600b978b8292e628d4473e25577e9df",
    },
    tranxs: [
        Transaction {
            nonce: 0,
            amount: 0,
            fee: 0,
            from: "0x0000",
            to: "0x1b2d",
            sign: "0x0000 -> 0x1b2d: 50 btc",
            hash:
"9a1cd23a5b0ecb6326621ae93c218e8db4499283386ce6b6008d496d469364c2",
        },
        Transaction {
            nonce: 1,
            amount: 9,
            fee: 1,
            from: "0xabcd",
            to: "0xabce",
            sign: "0xabcd -> 0xabce: 9 btc",
            hash:
"53d5cc1ac2b19555233eb2a9e8fdb62fb4a19387127f5f3b45b195edaa568305",
        },
        Transaction {
            nonce: 2,
            amount: 5,
            fee: 1,
            from: "0xabcd",
            to: "0xabce",
            sign: "0xabcd -> 0xabce: 5 btc",
            hash:
"84fa895cb4f0d21d555e66366af23648987fd81da539d99cb4f6810558863c2d",
        },
    ],
    hash:
"20448006a3c983acf305acc3b8f4e446d812add995704883cc48b42dda33a2cf",
}
```

3. 挖矿机制

挖矿机制的运行结果如下：

```
-------------------------Mine Info----------------------------
Start mining ...
Produced a new block!
New produced block saved!

Start mining ...
Produce a new block!

New produced block saved!

Start mining ...
Produce a new block!

New produced block saved!
```

4. 矿工信息

矿工信息如下：

```
-------------------------Miner Info------------------------------
Miner {
    name: "anonymous",
    balance: 204,
    address: "0x1b2d",
}
```

5. 账户信息

账户信息如下：

```
-------------------------Account Info----------------------------
Account {
    nonce: 2,
    balance: 84,
    name: "Kim",
    address: "0xabcd",
    hash:
"43a313f7687f1bb31ba1511c0fcad924a0fb72241ec8699e38f4b2f590f2e972",
}
Account {
    nonce: 4,
    balance: 103,
    name: "Tom",
    address: "0xabce",
    hash:
"82f22a084728d1a408d0e6d2153968bb62eb7701f7d646485cd739d1f69cd94f",
}
Account {
    nonce: 2,
    balance: 109,
    name: "Jim",
    address: "0xabcf",
    hash:
"2ba65fbbb28012d25a4287ca3099e73b9a0d832e648ab1fbc94cd54b57ea231c",
}
```

6. 区块信息

```
-------------------------Block Info------------------------------
Block {
    header: BlockHeader {
        nonce: 0,
        time: 1741505102,
        bits: 553713663,
        txs_hash:
"fb8cde2d50d0d48bce64c96c507194823a5b6dcb46c90034fda8f0f6b0dc6656",
        pre_hash:
"22caaf24ef0aea3522c13d133912d2b722caaf24ef0aea3522c13d133912d2b7",
    },
    tranxs: [
        Transaction {
            nonce: 0,
            amount: 0,
            fee: 0,
            from: "0x0000",
            to: "0x0000",
            sign: "创世区块",
            hash:
"fb8cde2d50d0d48bce64c96c507194823a5b6dcb46c90034fda8f0f6b0dc6656",
        },
    ],
    hash:
"12619cb2400ec66dfd39496f806b83045600b978b8292e628d4473e25577e9df",
}
Block {
    header: BlockHeader {
        nonce: 0,
        time: 1741505102,
        bits: 553713663,
        txs_hash:
"20c4a3b94c760d6a9f0bd9b2cd0dcb683cd6ad57d8c8fd3287df2df2eb5c1d3c",
        pre_hash:
"12619cb2400ec66dfd39496f806b83045600b978b8292e628d4473e25577e9df",
    },
    tranxs: [
        Transaction {
            nonce: 0,
            amount: 0,
            fee: 0,
            from: "0x0000",
            to: "0x1b2d",
            sign: "0x0000 -> 0x1b2d: 50 btc",
            hash:
"9a1cd23a5b0ecb6326621ae93c218e8db4499283386ce6b6008d496d469364c2",
        },
        Transaction {
            nonce: 1,
            amount: 9,
            fee: 1,
            from: "0xabcd",
            to: "0xabce",
            sign: "0xabcd -> 0xabce: 9 btc",
            hash:
"53d5cc1ac2b19555233eb2a9e8fdb62fb4a19387127f5f3b45b195edaa568305",
        },
        Transaction {
            nonce: 2,
            amount: 5,
            fee: 1,
            from: "0xabcd",
            to: "0xabce",
            sign: "0xabcd -> 0xabce: 5 btc",
            hash:
"84fa895cb4f0d21d555e66366af23648987fd81da539d99cb4f6810558863c2d",
        },
    ],
    hash:
"20448006a3c983acf305acc3b8f4e446d812add995704883cc48b42dda33a2cf",
}
Block {
    header: BlockHeader {
        nonce: 0,
        time: 1741505105,
        bits: 553713663,
        txs_hash:
"dcb88666af447af1833b20e871f0a8f024b369b88111a2b315758c5e841539f7",
        pre_hash:
"20448006a3c983acf305acc3b8f4e446d812add995704883cc48b42dda33a2cf",
    },
    tranxs: [
        Transaction {
            nonce: 0,
            amount: 0,
            fee: 0,
            from: "0x0000",
            to: "0x1b2d",
            sign: "0x0000 -> 0x1b2d: 50 btc",
            hash:
"9a1cd23a5b0ecb6326621ae93c218e8db4499283386ce6b6008d496d469364c2",
        },
        Transaction {
            nonce: 3,
            amount: 6,
            fee: 1,
            from: "0xabce",
            to: "0xabcf",
            sign: "0xabce -> 0xabcf: 6 btc",
            hash:
"edb391ee92448903f7b1f26b8e23e1cc4b7a77d78e9f5e70fec7ebedb7670052",
        },
        Transaction {
            nonce: 4,
            amount: 3,
            fee: 1,
            from: "0xabce",
            to: "0xabcf",
            sign: "0xabce -> 0xabcf: 3 btc",
            hash:
"cde1e971fab799a9b3ca6adc150c4c3c3b472ab8f2185f3d1985179b4d30f3b2",
        },
    ],
    hash:
"63b5881e71df1057dfe54d7a85fd02736beffd80407cdf35c4452804919bc427",
}
```

区块信息如下：

五、项目总结

本项目通过 Rust 语言实现了一个简单的区块链系统，包含区块生成、区块链存储、区块添加与验证、简单的挖矿机制等功能。通过本项目的实践，掌握了区块链的基本原理和实现方法，为进一步研究和开发区块链应用打下了基础。

六、参考资料

1. 区块链教程
2. 区块链学习路线
3. Go 实现的 demo
4. B 站区块链项目实战