

# DPOS 算法实验指导书

实验课时：3 学时

实验认识 3-4 人

## 一、实验名称

DPOS 共识算法的简单实现

## 二、实验内容

理解 DPOS 算法的工作原理和流程，了解委员会成员的选举和投票过程。

## 三、实验环境

Java 开发环境（JDK）：JDK17

IDE 工具：vscode

操作系统：windows11

## 四、算法描述

DPOS（Delegated Proof of Stake）是一种共识机制，也是区块链中一种流行的共识算法。与 PoW（Proof of Work）和 PoS（Proof of Stake）不同，DPOS 采用了委托投票的方式来选举出共识节点，由选举出的共识节点来验证和打包交易，生成新的区块。

DPOS 算法的主要思想是，让网络上的持币者通过投票选举出少数几个代表节点，这些节点被授权来验证和打包交易。持币者投票的权重与其持有的货币数量成正比。由于共识节点数量很少，交易验证速度快，处理能力强，同时也保证了网络的分布式特性和安全性。

## 五、实验过程

DPOS 算法的流程如下：

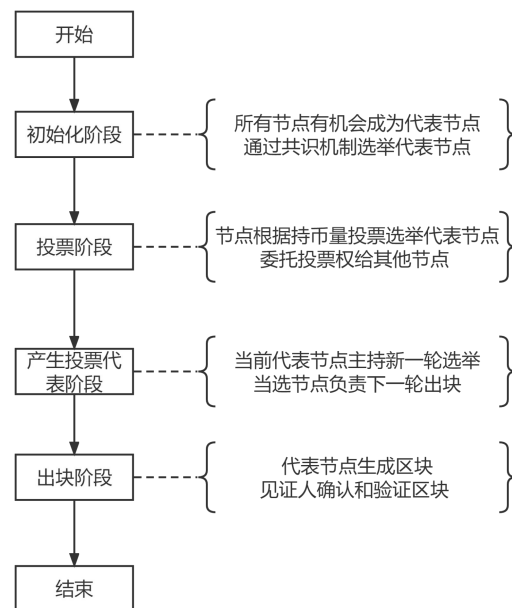


图 1：DPoS 算法流程图

1. 初始化阶段：在区块链网络中，所有的节点都有机会成为“代表”(delegate)，即参与出块的节点。在初始阶段，所有节点通过共识机制选举出一组“代表”节点。

区块链类：

```
public class Blockchain {  
    // 区块链列表  
    private static List<Blockchain> blockchainList = new ArrayList<>();  
    // 区块链的难度  
    private int difficulty;  
    // 投票列表  
    private List<Vote> voteList = new ArrayList<>();  
    // 节点列表  
    private List<Node> nodeList = new ArrayList<>();  
    // 区块列表  
    private List<Block> blockList = new ArrayList<>();  
    // 交易列表  
    private List<Transaction> transaction = new ArrayList<>();  
}
```

投票类：

```
public static class Vote {  
    private String candidate; // 候选人  
    private int voteCount; // 获得票数  
}
```

```
private int availableVotes; // 节点可用于投票的票数
```

节点类:

```
public static class Node {  
    private String address;  
    private int availableVotes; // 节点可用于投票的票数  
    private int voteCount; // 获得票数  
    private int tokenAmount; // 代币数量
```

区块类:

```
public static class Block {  
    private int index;  
    private long timestamp;  
    private List<Transaction> transactionList;  
    private List<Vote> voteList;  
    private String previousHash;  
    private String hash;  
    private int nonce;
```

交易类:

```
public static class Transaction {  
    private String from;  
    private String to;  
    private int amount;  
    private String signature;
```

创建创世区块:

```
public void createGenesisBlock() {  
    List<Transaction> transactions = new ArrayList<>();  
    List<Vote> votes = new ArrayList<>();  
    String previousHash = "";  
    int nonce = 0;  
    Block genesisBlock = new Block(0, System.currentTimeMillis(),  
transactions, votes, previousHash, nonce);  
    blockList.add(genesisBlock);  
    System.out.println("genesisBlock: " + genesisBlock.hash);  
}
```

2. 投票阶段: 在下一轮次开始之前, 所有节点都可以根据自己的持币量投票选举下一轮的“代表”节点。在此过程中, 每个节点可以将自己的投票权委托给其他节点, 以提高他们当选的概率。

```
// 添加节点并随机分配代币数量  
List<Blockchain.Node> nodes = new ArrayList<>();  
Random random = new Random();
```

```

int totalTokens = 10000; // 总代币数量
for (int i = 0; i < 100; i++) {
    int tokenAmount = 1 + random.nextInt(totalTokens / 10); // 保证每个节点至少拥有1个代币
    totalTokens -= tokenAmount;
    String nodeAddress = Blockchain.HashUtil.sha256(UUID.randomUUID().toString());
    Blockchain.Node node = new Blockchain.Node(nodeAddress, tokenAmount);
    blockchain.addNode(node);
    nodes.add(node);
    // 在添加节点的同时，创建对应的投票并添加到投票列表
    Blockchain.Vote vote = new Blockchain.Vote(nodeAddress, tokenAmount, tokenAmount);
    blockchain.addVote(vote);
    System.out.println("节点已添加，节点为: " + (i + 1) + ". " + node.getAddress() + ", 代币数量为: " + node.getTokenAmount());
}
// 根据分配的代币给予节点票数
for (Blockchain.Node node : nodes) {
    int numVotes = node.getTokenAmount(); // 获取节点的代币数量
    node.addVote(numVotes); // 给节点增加票数
}
// 进行随机投票模拟
Random random1 = new Random(System.currentTimeMillis());
List<Blockchain.Vote> votes = blockchain.getVoteList();
for (Blockchain.Node node : nodes) {
    int numVotes = node.getVoteCount(); // 获取节点的票数
    for (int i = 0; i < numVotes; i++) {
        int candidateIndex = random1.nextInt(votes.size()); // 随机选择候选人索引
        Blockchain.Vote vote = votes.get(candidateIndex); // 获取对应的候选人投票
        node.vote(vote); // 节点进行投票
    }
}

```

3. 产生投票代表阶段：在每个轮次结束时，由当前的“代表”节点主持新一轮的“代表”节点选举，当选的节点将负责下一轮的出块工作。

```

// 按票数排序节点
List<Blockchain.Node> sortedNodes = Blockchain.sortNodesByVoteCount(nodes);

// 输出票数最高的30个节点

```

```
System.out.println("票数最高的 30 个节点: ");
for (int i = 0; i < 30 && i < sortedNodes.size(); i++) {
    Blockchain.Node node = sortedNodes.get(i);
    System.out.println((i + 1) + ". " + node.getAddress() + " - 票数: "
        + node.getVoteCount());
}
```

4. 出块阶段：在当前的轮次中，由被选中的“代表”节点负责生成区块，而其他节点将成为“见证人”（witness），对该区块的有效性进行确认和验证。

```
// 创建一个新的区块并添加到区块链
Blockchain.Block newBlock1 = new Blockchain.Block(1,
    System.currentTimeMillis(), new ArrayList<>(),
    blockchain.getVoteList(), blockchain.getLatestBlock().getHash(), 0);
System.out.println("等待添加区块 1: ");
blockchain.addBlock(newBlock1);
System.out.println("区块 1 已添加, 区块哈希为: " + newBlock1.getHash());
Blockchain.Block newBlock2 = new Blockchain.Block(2,
    System.currentTimeMillis(), new ArrayList<>(),
    blockchain.getVoteList(), blockchain.getLatestBlock().getHash(), 0);
System.out.println("等待添加区块 2: ");
blockchain.addBlock(newBlock2);
System.out.println("区块 2 已添加, 区块哈希为: " + newBlock2.getHash());
// 验证区块链的合法性
System.out.println("区块链的合法性为: " + blockchain.validate());
```

## 六、实验结果

生成创世区块

```
genesisBlock: zRspgzl05gIqlEVj0uAvl60j33agV1ZWPfzjJoJxFs=
创世区块已创建
```

生成 100 个节点，并将 10000 个代币随机分配给这 100 个节点

```
节点已添加, 节点为: 1. dodhtduHeRSyqVKIK+BWcjawmuSqrWzT0P810y6so8s=, 代币数量为: 401
节点已添加, 节点为: 2. WM1kwDugXNT9BRW9kkNq9ev0UW/hnZathrHDw0TzAo=, 代币数量为: 180
节点已添加, 节点为: 3. D3fQrRqRwgmDfEvcBhW/Q5GUL43bj/dDSV/ElgSC94Q=, 代币数量为: 373
节点已添加, 节点为: 4. 95oeZUsb/p/ECWcrTXNXLHUJgZuKeDp/0aQNPzf6A1=, 代币数量为: 445
节点已添加, 节点为: 5. NnV59+EoQhmuSE3BV0M98jNq08Z541aEYyI6oMpcU=, 代币数量为: 851
节点已添加, 节点为: 6. Pq0LXh90oP6ngmX3cRA2L6jf2F60HK6zpfKM8cF4=, 代币数量为: 24
节点已添加, 节点为: 7. 2weS3b783CW5J3gJ1eagVv080NR6w2mmq/eUUh/b/tc=, 代币数量为: 722
节点已添加, 节点为: 8. G4y4rR/6b3c1ScF60xZ6VFLW/A6mmUaA0AR0UQ=, 代币数量为: 52
节点已添加, 节点为: 9. g/LmF6nyHXB8fAX8BuYeQlDlUMPL1KWY7d880gbw=, 代币数量为: 556
节点已添加, 节点为: 10. TK3MtT0Geh8dAT2UKLXW3xv0ToQjtvIXep8tXF0FSw=, 代币数量为: 361
节点已添加, 节点为: 11. UKWzF8xN0vnm0Fdmkxos2eMLFIb7f2EYpD06a7+ch8=, 代币数量为: 446
节点已添加, 节点为: 12. bu0Jukm+888XKbZnTC+2UeUcA2UqcsIACF2E3TCj4FA=, 代币数量为: 298
节点已添加, 节点为: 13. aWbWCW/Lw6ENhT80pYPAJj/gfeNpR+tpEzGNUU4rfo=, 代币数量为: 519
节点已添加, 节点为: 14. Sth5EY7ZjehG5vAKQBaL3d2QZ3k1tQ13o4Gg392Hoc=, 代币数量为: 71
节点已添加, 节点为: 15. /t7w/xiv87T+sHsduE6FK5A1NwZ6+2qH4+Nvij3Q1A=, 代币数量为: 419
节点已添加, 节点为: 16. M1A280wjYhp4V3P68XzjwQcJFaMm+DcaUqJ4way4c=, 代币数量为: 84
节点已添加, 节点为: 17. e3X1i0oPj8cc170Cq682BNY2d/3YvNk081NfMNS9=, 代币数量为: 132
节点已添加, 节点为: 18. fCV0NNPvhlLD0ZstH70dyne/EY9HtF8h9e0q0dSC=, 代币数量为: 362
节点已添加, 节点为: 19. p163K4v+CCl8ndFXg87MYcaQthbb3gE1exJChwB2Uw=, 代币数量为: 6
节点已添加, 节点为: 20. RwfRRR0W0131686Rlta19zT2Zv31P2w8eMd0uK8Q1I8=, 代币数量为: 364
节点已添加, 节点为: 21. W8oX0u1JWS177NDU2Q7LkRvYsffFuzCN194oXQ7N4ps=, 代币数量为: 290
节点已添加, 节点为: 22. 6m3ZbGBN8eVnjsKje3gc0FPLRhjHdXnYo81JymcDRw=, 代币数量为: 161
节点已添加, 节点为: 23. y59BchQZme2h6vAvfAFgcZbK/4EILBxVw5bonPwD18=, 代币数量为: 154
节点已添加, 节点为: 24. 34fzK1WB5KXK9L1K2vjIeVnc+008g15ff+FcZfubkQ=, 代币数量为: 61
节点已添加, 节点为: 25. nfhuc4/IfmArQxV1uxsTMfPdy2Lkqcu665T5v1IKL4=, 代币数量为: 262
节点已添加, 节点为: 26. 7WgJ121jaVqu+4f/Wu0u6F4Cb3S81KdYp96j79Y5Cw=, 代币数量为: 115
节点已添加, 节点为: 27. 2JNjFK+FXh0d4VXkvTcJocb1R04FwR2A7+fc6P4d=, 代币数量为: 34
节点已添加, 节点为: 79. 95LH0g8Hy+VveUwBZTaeI0wsX+pk6JKE2R73huHvL=, 代币数量为: 6
节点已添加, 节点为: 80. FCF0V6Hsj10EjKo4YWS4sXU0XIozhMo61hed3gns=, 代币数量为: 5
节点已添加, 节点为: 81. PDCNfYqAGVJeAdz9tX0LhhRY8YHwgpQCRKhQW+gro=, 代币数量为: 4
节点已添加, 节点为: 82. 3o7P1ZBxvWpJn9v/CXSerW98cfsbq+ns1+8Lnnn8AEI=, 代币数量为: 8
节点已添加, 节点为: 83. L3JmTEygE3GL+fjNRU71mw6QJ3AH0U0aWaux0Inak4=, 代币数量为: 5
节点已添加, 节点为: 84. SzV9Uu67Zs1XDs4Iq9Izd635Y2Kv0Jcopd1SXG6bT8=, 代币数量为: 2
节点已添加, 节点为: 85. 7bMGpj1fTfTfJUsd01+AVcnQDzZrrPBWIE+fUkvfMoA=, 代币数量为: 2
节点已添加, 节点为: 86. DmyNVhnpaEXjnk5rM9PR71EycTFeb5twmCvobZAQ0I=, 代币数量为: 5
节点已添加, 节点为: 87. 0czFrsC3z0MQTWs4Lucus+9UvU8HrHcbR140IKt0u5A=, 代币数量为: 5
节点已添加, 节点为: 88. RvusD81PH2+yWPRX8XoeGncRhmh+WTAl1VLDsonq1kM=, 代币数量为: 4
节点已添加, 节点为: 89. 8uM3kNWqZf/kpTvz1BvAxaVPxYlTv50AnoWRDzSSug=, 代币数量为: 4
节点已添加, 节点为: 90. 0M1oWGHqZ4Z6k5XpsI5CpEdw/U6MeKLRcNUNYQpD9Sc=, 代币数量为: 1
节点已添加, 节点为: 91. 8NJMLCKKdyk0wsRVUnxg4IA26qfTsQYT4AaBg3Py6xc=, 代币数量为: 5
节点已添加, 节点为: 92. d0gEBZDkKd+U/Z2Y5WK64A1j3Nyw3Lj22L+N3R8Hdw=, 代币数量为: 3
节点已添加, 节点为: 93. FxYvo3w35oCFI4T1W8+JbcuLSL3aHzTm6LxbiVcYqDY=, 代币数量为: 3
节点已添加, 节点为: 94. ZH6gZWBPyJakWcp7q/LosFDf3bmNQBqRecRZeJn1bA=, 代币数量为: 3
节点已添加, 节点为: 95. 38tSRF0mw+VR6Y5ldptf6yfcP89/Zmhn70MptM4Lio9=, 代币数量为: 2
节点已添加, 节点为: 96. UknPJdK06cU+P55LeWp25F6cFmbt73JfFPBkbTBwrfI=, 代币数量为: 3
节点已添加, 节点为: 97. gMpI+eyZv5/3Q0ewQ3m6s++jh9cStLbPHB5zmXdDEC8=, 代币数量为: 3
节点已添加, 节点为: 98. +89H7a0tpd6eb1FR7b5L85Qsq10c+I0Wo+0pnnqIViu8=, 代币数量为: 2
节点已添加, 节点为: 99. 1ns8F2L0Lo6FawEo713uHWSucKFYU5jxKymAsIdE=, 代币数量为: 1
节点已添加, 节点为: 100. 7cJmKPKimJn8xo1UpuYQJkaW+qY1XkyXc1d4TVnmS8=, 代币数量为: 2
```

进行随机投票并将节点按获得的票数排序，得出排名前 30 的节点

票数最高的30个节点：

```
1. NnV59+EoQhmeuSEJBV0M98jNq00Z541aEYyI6oMpncU= - 票数：851
2. 2weS3b703KCW5J8Jg1eaGYvoB0NRGw2mmq/eUUhb/tc= - 票数：722
3. g/lmFGnyMSXBsfVAX8uVe+QLDTuMP1IKWY7d88Dgbvw= - 票数：556
4. aMWbCW/lwGENhqT80pYPAjJ/gfeNpR+tPEzGNUU4rfo= - 票数：519
5. UMWzF8xN0vvm0Fdwkdxos2eMLFIb7f2EYpD06a7+ch0= - 票数：446
6. 95oeZusb/b/ECWtn7KXo1lHUJgZUKedp/0aQMPzf6AI= - 票数：445
7. /t7w/xiv87T+sWHsduE6FK5AiNwZ6+2qH4+Nvijq3IA= - 票数：419
8. dodhtduHeRSygKVIK+BWcjawmuSqrWzT0P810yGso8s= - 票数：401
9. D3fQRqRwgBmDfEvc0hW/Q5GUL4JbJ/dDSv/ElgSC94Q= - 票数：373
10. RWfRRRW0i3lGBGRUta1l9zT2ZYb31P2wBsMDuKBQJI8= - 票数：364
11. fCV0NhPYbhILD6ZsTHt70dyxe/Ey9WIafScHrQvD4S4= - 票数：362
12. TK3MnT0Geh8EdATzUKlXW3xv0ToQjtvIXep8tXF0FSw= - 票数：361
13. bUoJuKm+88BxKbZnTC+2UeUcA2UqcsiACf2E3TCj4FA= - 票数：298
14. Wa0xQuiJWSi77hDU2Q7LYkRvYsfFuzCNl94qXQ7N4ps= - 票数：290
15. nfhuc4/IFmKARQXv1uxsTMfPYd2LkqcuG65T5vIiKL4= - 票数：262
16. C3R09Tj46wa2hXsUEg5schGD/uQXaWWcGRvT8kNRhzM= - 票数：181
17. WM1kwW0ugxNT9BRW9kkNq9ev0UW/hnZathrHDw0TzAo= - 票数：180
18. 6m3ZbGBN8eVnjsKje3gc0fPLRVhjHdXnYo0iJymcDRw= - 票数：161
19. nd8rqnXkr/UftFwUvdUiylT7mvGmUkQ85Gf7H3EcAzg= - 票数：159
20. jVo74pymUA/VP2JRZM5fngmNzHhX7UVxCmvCuLuwD64= - 票数：159
21. y59BchQZmeZhGvAvfAFgcZbk/4EILbMxvW5bonPwD18= - 票数：154
22. e3xi1o0oPj8Ccl7CQ4p8zBNYZd/jYyNk0HihNfmN500= - 票数：132
23. 7WgV121j6aVqu+F/wU6uF64Cb330iKqWYpg6j79YsCw= - 票数：115
24. DcYnWKK0h701v2F/w7PTA1C+cmK9qvU1HA+D+i6RfvUw= - 票数：104
```

最后进行出块并验证合法性

等待添加区块1：

Block mined: 000W5R05W+ZY1ST3sMwMz1XKL1+w7p6bE640qwkrzLY=

区块 1已添加，区块哈希为：000W5R05W+ZY1ST3sMwMz1XKL1+w7p6bE640qwkrzLY=

等待添加区块2：

Block mined: 000hFDksh28Ds9BiwnwDCQ9jYZAig0ZuwHL99iXzkmU=

区块 2已添加，区块哈希为：000hFDksh28Ds9BiwnwDCQ9jYZAig0ZuwHL99iXzkmU=

区块链的合法性为：true