# 法律声明

- 本课件包括：演示文稿，示例，代码，题库，视频和声音等，小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。

关注 小象学院

# 区块链编程：Solidity以太坊智能合约

王亮

# 第四课 solidity编程:智能合约实现

## 4.11 小型实战-课程积分

# 项目需求

- 为活跃课程的氛围，促进同学之间的交流，决定建立课程积分体系：

1. 每个同学都能领100个课程积分
2. 为别的同学答疑解惑，可以从对方那里获得一定的课程积分
3. 课程结束后，积分最高的同学获得小红花

# “古典”设计

## 基于sql数据库设计实现:

用户表(User)

| 字段 | 内容 |
| --- | --- |
| userId | 用户id |
| email | 用户email |
| … | |

用户积分表(UserCoin)

| 字段 | 内容 |
| --- | --- |
| userId | 用户id |
| coin | 积分数量 |
| … | |

用户积分交易表(UserTrade)

| 字段 | 内容 |
| --- | --- |
| userId | 发送方用户id |
| toUserId | 接收方用户id |
| coin | 发送数量 |
| … | |

# “古典”设计的不足

- 发放积分方式不透明，需要课程讲师信誉背书
- 积分交易数据不透明，可能出现刷分现象
- 服务器可能挂了……

# Dapp设计的优势

- 去信任。代码透明，积分体系和发放形式完全公开，无需讲师信用背书。
- 去中心。数据透明，交易数据公共可读，无法随意篡改，公共监督。不用担心机器崩溃。
- 隐私保护。暂不需要。

# Dapp数据结构设计

## 用状态变量实现数据结构

用户表(User)

```
struct User {
    address userId; //用户ic
}

User[] userList;
```

用户积分表(UserCoin)

```
struct UserCoin {
    address userId;//用户id
    uint coin; //积分数量
}

UserCoin[] userCoinList;
```

用户积分交易表(UserTrade)

```
struct UserTrade {
    address userId;//发送方user id
    address toUserId; //接收方user id
    uint coin; //发送数量
}

UserTrade[] userTradeList;
```

# Dapp数据结构设计改进原则

- 基本思路同“SQL数据映射到KV数据库”
- 根据查询/索引需求，在基本数据基础上，建立mapping
- 为节省存储，去掉冗余数据

# 功能需求1

- 每个同学都能领100个课程积分
- 函数 getCoin()
- 功能：由合约为调用者发送100个积分
- 查询需求：只能领一次，需要根据合约和调用者地址，查询对应的交易记录。

# 功能需求2

- 同学之间可以互送积分
- 函数 sendCoin()
- 功能：调用者向一个地址发送积分
- 查询需求：一个用户地址拥有的积分

# Dapp数据结构改进

用户积分交易

```
struct UserTrade {
    address userId;//发送方user id
    address toUserId; //接收方user id
    uint coin; //发送数量
}

UserTrade[] userTradeList;
```

查询需求：根据“合约和调用者地址”，查询交易

```
mapping (address => mapping(address=>UserTrade[])) trans;
```

去除冗余数据

```
mapping (address => mapping(address=>uint[])) trans;
```

# Dapp数据结构设计结果

用户

```
address[] userList;
mapping (address=>uint8) userDict;
```

用户积分(address为key)

```
mapping (address => uint) balances;
```

用户积分交易表(address + address为key)

```
mapping (address => mapping(address=>uint[])) trans;
```

# 流程演示

# 联系我们

## 小象学院：互联网新技术在线教育领航者

- 微信公众号：**小象学院**