

『0020』 - 众筹（CrowdFunding）智能合约（Smart Contract）综合案例

孔壹学院：国内区块链职业教育领先品牌

作者：黎跃春，区块链、高可用架构工程师

微信：liyc1215 QQ群：348924182 博客：<http://liyuechun.org>

结构体和字典综合案例

下面的案例是一个 集资 合约的案例，里面有两个角色，一个是投资人 Funder，也就是 出资者。另一个角色是运动员 Campaign，被赞助者。一个 Funder 可以给多个 Campaign 赞助，一个 Campaign 也可以被多个 Funder 赞助。

完整合约：

```
pragma solidity ^0.4.4;

contract CrowdFunding {

    // 定义一个`Funder`结构体类型，用于表示出资人，其中有出资人的钱包地址和他一共出资的总额度。
    struct Funder {
        address addr; // 出资人地址
        uint amount; // 出资总额
    }

    // 定义一个表示存储运动员相关信息的结构体
    struct Campaign {
        address beneficiary; // 受益人钱包地址
        uint fundingGoal; // 需要赞助的总额度
        uint numFunders; // 有多少人赞助
        uint amount; // 已赞助的总金额
        mapping (uint => Funder) funders; // 按照索引存储出资人信息
    }

    uint numCampaigns; // 统计运动员(被赞助人)数量
    mapping (uint => Campaign) campaigns; // 以键值对的形式存储被赞助人的信息
```

```

// 新增一个`Campaign`对象, 需要传入受益人的地址和需要筹资的总额
function newCampaign(address beneficiary, uint goal) public returns (uint campaignID) {
    campaignID = numCampaigns++; // 计数+1
    // 创建一个`Campaign`对象, 并存储到`campaigns`里面
    campaigns[campaignID] = Campaign(beneficiary, goal, 0, 0);
}

// 通过campaignID给某个Campaign对象赞助
function contribute(uint campaignID) public payable {
    Campaign storage c = campaigns[campaignID]; // 通过campaignID获取campaignID对应的Campaign对象
    c.funders[c.numFunders++] = Funder({addr: msg.sender, amount: msg.value}); // 存储投资者信息
    c.amount += msg.value; // 计算收到的总款
    c.beneficiary.transfer(msg.value);
}

// 检查某个campaignID编号的受益人集资是否达标, 不达标返回false, 否则返回true
function checkGoalReached(uint campaignID) public returns (bool reached)
{
    Campaign storage c = campaigns[campaignID];
    if (c.amount < c.fundingGoal)
        return false;
    return true;
}
}

```

技术交流

- 区块链技术交流QQ群: 348924182
- 「区块链部落」官方公众号



长按，识别二维码，加关注