区块链大作业

18340146 计算机科学与技术 宋渝杰

18340121 计算机科学与技术 刘依澜

18342052 软件工程 李研通

实验说明：

值得提前说明的是，由于我们知识和能力的不足，我们没有实现“前后端”部分，只实现了区块链“链端”部分及其应有的四个基本功能和一定的额外功能，因此报告之后的设计说明都只基于“链端”实现。

实验过程：

项目设计：

首先回顾一下我们需要实现的四个功能：

1. 实现采购商品—签发应收账款交易上链。例如车企从轮胎公司购买一批轮胎并签订应收账款单据。
2. 实现应收账款的转让上链，轮胎公司从轮毂公司购买一笔轮毂，便将于车企的应收账款单据部分转让给轮毂公司。轮毂公司可以利用这个新的单据去融资或者要求车企到期时归还钱款。
3. 利用应收账款向银行融资上链，供应链上所有可以利用应收账款单据向银行申请融资。
4. 应收账款支付结算上链，应收账款单据到期时核心企业向下游企业支付相应的欠款。

简单来说也就是实现【生成收据、转让收据、申请融资、偿还债款】四个功能

我们可以通过一份智能合约实现以上四个功能：

基本准备：

考虑到金融背景涉及**公司、银行和收据**，因此声明对应的结构体，具体如下：（智能合约为 Solidity 语言，可在其官方文档 https://solidity-cn.readthedocs.io/zh/develop/index.html 进行语法学习）

```
    struct Bill { // 收据
        string From; // 欠款人
        string To; // 收款人
        int Type; // 收据类型：1：普通收据；2：转让收据；3：融资收据；0：已偿还收据
        int Money; // 收据金额
        uint CreateDate; // 收据创建时间
        uint RepayDate; // 还债时间
    }
    struct Organization { // 公司 or 银行
        string Name; // 名称
        int Money; // 金额（余额）
        int Type; // 类型：1：公司；0：银行
    }
```

同时声明一些辅助变量，用于记录公司/银行总数、收据总数（包括已偿还收据）、用于记录所有公司/银行和所有收据的 map：

```
    int LenOrg; // 公司/银行总数
    int LenBill; // 收据总数
    mapping(string => Organization) Organizations; // 公司名 -> 公司信息
    mapping(int => Bill) Bills; // 收据id -> 收据数据
```

最后是一些辅助函数：合约的构造函数以及字符串的判断是否相等函数（Solidity 居然没有字符串 == 和 != 运算符重载，不知道它是怎么基于 C++ 的）

```
    constructor() public {
        LenOrg = 0;
        LenBill = 0;
    }

    function isEqual(string s1, string s2) public returns(bool) {
        if (bytes(s1).length != bytes(s2).length) return false;
        for (uint i=0; i<bytes(s1).length; i++) {
            if (bytes(s1)[i] != bytes(s2)[i]) return false;
        }
        return true;
    }
```

功能一：生成收据

实现的具体功能为：生成公司 From 应向公司 To 支付 Money 的一份收据

这一功能较为简单，具体算法步骤如下：

- 异常输入判断：如果收据双方不为公司类型，或者为同一公司，或者至少一方不存在，则直接返回 -1
- 正常输入情况下只需实例化收据结构体，填入相应的数据，并返回该收据的 ID

```
    function NewBill(string From, string To, int Money) public returns(int) {
        // 异常输入判断（一方不存在时Type默认值为0，也不满足以下判断）
        if (Organizations[From].Type == 1 && Organizations[To].Type == 1 &&
!isEqual(From, To)) {
            // 实例化收据：ID 从 0 开始；收据类型为“普通收据”；还款时间为了实验方便，设定为
1 分钟后
            Bills[LenBill++] = Bill(From, To, 1, Money, now, now + 60*1000);
            return LenBill-1;
        }
        return -1;
    }
```

功能二：转让收据

实现的具体功能为：从收款人为公司 From 的所有收据中转让价值 Money 的部分给公司 To

这应该是四个功能中最复杂的，需要考虑的东西也很多，具体算法步骤如下：

- 异常输入判断：和上一步一样，如果账单双方不为公司类型，或者为同一公司，或者至少一方不存在，则直接返回 -1
- 计算最大可转让收据的金额总数，转让收据时的金额数不可超过最大可转让总数
- 正式转让：遍历每一张收据，如果是收款人为公司 From 的收据：
    - 如果这张收据的金额 > Money，则这张收据的金额减少 Money，并额外生成一张价值为 Money 的收据，收款人为公司 To，欠款人为这张收据的欠款人，结束转让总过程

- 如果这张收据的金额 <= Money，则直接修改这张收据的收款人为公司 To，并将收款总金额 Money 减少相应数值（即这张收据的金额），之后判断 Money 是否已为 0，是则结束转让总过程，否则再遍历下一张收据

```
    function TransferCount(string From, string To) public returns(int) { // 计算最大可转让收据的金额总数
        int sum = 0;
        if (Organizations[From].Type == 1 && Organizations[To].Type == 1 && !isEqual(From, To)) { // 异常输入处理在这个子函数实现
            for (int i=0; i<LenBill; i++) {
                if (isEqual(Bills[i].To, From) && !isEqual(Bills[i].From, To) && (Bills[i].Type == 1 || Bills[i].Type == 2)) { // 这张收据的欠款人不能是公司 To
                    sum += Bills[i].Money;
                }
            }
            return sum;
        }
        return -1;
    }

    function Transfer(string From, string To, int Money) public returns(int) {
        int sum = Money;
        if (TransferCount(From, To) >= Money) {
            for (int i=0; i<LenBill; i++) {
                if (isEqual(Bills[i].To, From) && (Bills[i].Type == 1 || Bills[i].Type == 2)) { // 开始转让
                    if (Bills[i].Money > sum) { // 价值大于剩余转让值
                        Bills[i].Money -= sum; // 减少对应金额
                        Bills[LenBill++] = Bill(Bills[i].From, To, 2, sum, Bills[i].CreateDate, Bills[i].RepayDate); // 生成子账单
                        sum = 0;
                    }
                    else if (!isEqual(Bills[i].From, To)) { // 小于等于
                        Bills[i].To = To; // 直接修改收款人进行转让
                        Bills[i].Type = 2; // 收据类型修改为“转让收据”
                        sum -= Bills[i].Money; // 修改剩余转让值
                    }
                    if (sum == 0) break; // 结束
                }
            }
            return 0;
        }
        return -1;
    }
```

功能三：申请融资

实现的具体功能为：将收款人为公司 From 的所有收据转让给银行 To，将收据所有金额提现

这一步也相对简单，具体算法步骤如下：

- 异常输入判断：From 需为公司类型，To 需为银行类型
- 正式融资：遍历所有收据，将收款人为公司 From 的所有收据修改收款人为银行 To，统计所有收据金额，最后加到公司 From 的账户内

```
    function Raise(string From, string To) public returns(int) {
        int sum = 0;
```

```
        if (Organizations[From].Type == 1 && Organizations[To].Type == 0) { // 异常输入处理
            for (int i=0; i<LenBill; i++) {
                if (isEqual(Bills[i].To, From) && (Bills[i].Type == 1 || Bills[i].Type == 2)) { // 正式融资
                    Bills[i].To = To; // 收款人转为银行
                    Bills[i].Type = 3; // 收据类型修改为“融资收据”
                    sum += Bills[i].Money; // 统计总金额
                }
            }
            Organizations[From].Money += sum; // 公司 From 增加相应金额
            Organizations[To].Money -= sum; // 银行 To 减少相应金额
            return sum;
        }
        return -1;
    }
```

功能四：偿还债款

实现的具体功能为：对欠款人为公司 From 的所有收据进行偿还债款

这一步也不难，具体算法步骤如下：

- 异常输入判断：From 需为公司类型
- 判断偿还总额：如果大于公司目前账户余额，则偿还失败
- 正式偿还：遍历所有收据，将欠款人为公司 From 的所有收据，修改收据类型为“已偿还收据”，同时统计收据总金额，最后从公司账户余额中扣除

```
    function RepayCount(string From) public returns(int) { // 统计偿还总额
        int sum = 0;
        if (Organizations[From].Type == 1) { // 异常输入处理在这个子函数实现
            for (int i=0; i<LenBill; i++) {
                if (isEqual(Bills[i].From, From) && Bills[i].Type != 0 && now >= Bills[i].RepayDate) {
                    sum += Bills[i].Money; // 统计总金额
                }
            }
            return sum;
        }
        return -1;
    }

    function Repay(string From) public returns(int) {
        int sum = 0;
        if (RepayCount(From) != -1 && RepayCount(From) <= Organizations[From].Money) { // 正式偿还
            for (int i=0; i<LenBill; i++) {
                if (isEqual(Bills[i].From, From) && Bills[i].Type != 0 && now >= Bills[i].RepayDate) {
                    Bills[i].Type = 0; // 修改收据类型为“已偿还收据”
                    sum += Bills[i].Money; // 统计总金额
                }
            }
            Organizations[From].Money -= sum; // 公司 From 账户扣除余额
            return sum;
        }
        return -1;
```

```
    }
```

额外功能：

为了便于测试，新添加如下功能：

注册公司/银行：输入公司/银行名和账户余额，程序将其加入 mapping 表：

```
    function Register(string memory Name, int Money, int Type) public{
        Organizations[Name] = Organization(Name, Money, Type);
        LenOrg++;
    }
```

输出公司/银行和收据：通过公司/银行的名字/收据的 ID，输出对应公司/银行/收据的信息：

```
    function GetCompany(string memory Name) public view returns(string memory,
int, int) {
        return (Organizations[Name].Name, Organizations[Name].Money,
Organizations[Name].Type);
    }

    function GetBill(int id) public view returns(string memory, string memory,
int, int, uint, uint) {
        return (Bills[id].From, Bills[id].To, Bills[id].Type, Bills[id].Money,
Bills[id].CreateDate, Bills[id].RepayDate);
    }
```

功能测试：

预备工作：

首先启动上个实验生成的区块链和控制台：

```
bash nodes/127.0.0.1/start_all.sh
cd ~/fisco/console && bash start.sh
```

部署智能合约：

```
deploy syj
```

```
[group:1]> deploy syj
transaction hash: 0x1b97db9a5313d2278b98cd5dd95cc17e9390bb04ae3a635cf3fae81b82d6fe64
contract address: 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d
```

注册几个公司和一个银行：

```
call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d Register com1 10000 1
call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d Register com2 10000 1
call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d Register com3 10000 1
call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d Register bank 10000000 0
```

```
[group:1]> call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d Register com1 10000 1
transaction hash: 0xe019b67b3581b23cb1adb7a88d7f59511ee74528e6e8a0c16490446d685bd61a
---------------------------------------------------------------------------------------------
transaction status: 0x0
description: transaction executed successfully
---------------------------------------------------------------------------------------------
Output
Receipt message: Success
Return message: Success
Return value: 0
---------------------------------------------------------------------------------------------
Event logs
Event: {}

[group:1]> call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d Register com2 10000 1
transaction hash: 0xf989a5115078f7534817b3eadae0c077b7b5fab459882bad507b733a67c10cf7
---------------------------------------------------------------------------------------------
transaction status: 0x0
description: transaction executed successfully
---------------------------------------------------------------------------------------------
Output
Receipt message: Success
Return message: Success
Return value: 0
---------------------------------------------------------------------------------------------
Event logs
Event: {}
```

```
[group:1]> call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d Register com3 10000 1
transaction hash: 0xe0d716b82d38d2136fbfb41d3fc4d93dfcc1a00f2a81f562ba62494132b1c6a5
---------------------------------------------------------------------------------------------
transaction status: 0x0
description: transaction executed successfully
---------------------------------------------------------------------------------------------
Output
Receipt message: Success
Return message: Success
Return value: 0
---------------------------------------------------------------------------------------------
Event logs
Event: {}

[group:1]> call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d Register bank 10000000 0
transaction hash: 0x2cd3bae1f8b68ead268b3826921580800834f2c244297314dd0def7fe178e24e
---------------------------------------------------------------------------------------------
transaction status: 0x0
description: transaction executed successfully
---------------------------------------------------------------------------------------------
Output
Receipt message: Success
Return message: Success
Return value: 0
---------------------------------------------------------------------------------------------
Event logs
Event: {}
```

显示公司 com1 和银行 bank 信息：

```
call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetCompany com1
call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetCompany bank
```

```
[group:1]> call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetCompany com1
---------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
---------------------------------------------------------------------------------------------
Return values:
[
    "com1",
    10000,
    1
]
---------------------------------------------------------------------------------------------

[group:1]> call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetCompany bank
---------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
---------------------------------------------------------------------------------------------
Return values:
[
    "bank",
    10000000,
    0
]
---------------------------------------------------------------------------------------------
```

测试功能一：生成收据

生成 com1 对 com2、com1 对 com3 的两份价值均为 1000 的收据，账单 ID 分别为 0 和 1

```
call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d NewBill com1 com2 1000
call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d NewBill com1 com3 1000
```

```
[group:1]> call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d NewBill com1 com2 1000
transaction hash: 0xaf8bd7f9a198ddd8bd1089720f7dfc925521901199c87239172c8038da4a1b56
---------------------------------------------------------------------------------------------
transaction status: 0x0
description: transaction executed successfully
---------------------------------------------------------------------------------------------
Output
Receipt message: Success
Return message: Success
Return value: 0
---------------------------------------------------------------------------------------------
Event logs
Event: {}

[group:1]> call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d NewBill com1 com3 1000
transaction hash: 0xc11685bb20a91a29c7592794dba182d5ee962b569b0cb43d6821f29e6e88cf02
---------------------------------------------------------------------------------------------
transaction status: 0x0
description: transaction executed successfully
---------------------------------------------------------------------------------------------
Output
Receipt message: Success
Return message: Success
Return value: 1
---------------------------------------------------------------------------------------------
Event logs
Event: {}
```

查看这两份收据：可以看出收据生成成功且信息无误（需要注意的是，收据生成时间和结算时间以毫秒为单位）

```
call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetBill 0
call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetBill 1
```

```
[group:1]> call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetBill 0
---------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
---------------------------------------------------------------------------------------------
Return values:
[
    "com1",
    "com2",
    1,
    1000,
    1607488845485,
    1607488905485
]
---------------------------------------------------------------------------------------------

[group:1]> call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetBill 1
---------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
---------------------------------------------------------------------------------------------
Return values:
[
    "com1",
    "com3",
    1,
    1000,
    1607488849084,
    1607488909084
]
---------------------------------------------------------------------------------------------
```

测试功能二：转让收据

简单情况：将收款人为 com2 的收据中转让价值为 1000 的部分给 com3：

```
call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d Transfer com2 com3 1000
```

```
[group:1]> call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d Transfer com2 com3 1000
transaction hash: 0x6627e466dc6b492d7fc8dd9adb4e513ff8d4e8b60c5893e16ccd63fd137125d2
---------------------------------------------------------------------------------------------
transaction status: 0x0
description: transaction executed successfully
---------------------------------------------------------------------------------------------
Output
Receipt message: Success
Return message: Success
Return value: 0
---------------------------------------------------------------------------------------------
Event logs
Event: {}
```

检查结果：（根据算法原理，结果应该为：收据 ID 为 0 的收据收款人改为 com3，且收据类型改为“转让收据”）

```
call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetBill 0
```

```
[group:1]> call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetBill 0
---------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
---------------------------------------------------------------------------------------------
Return values:
[
    "com1",
    "com3",
    2,
    1000,
    1607488845485,
    1607488905485
]
---------------------------------------------------------------------------------------------
```

复杂情况：将收款人为 com3 的收据中转让价值为 1500 的部分给 com2：

```
call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d Transfer com3 com2 1500
```

```
[group:1]> call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d Transfer com3 com2 1500
transaction hash: 0xecb3bcbc6f691ffad3caa618a3ac855a6496c52b83137680dc22703a81b603b5
---------------------------------------------------------------------------------------------
transaction status: 0x0
description: transaction executed successfully
---------------------------------------------------------------------------------------------
Output
Receipt message: Success
Return message: Success
Return value: 0
---------------------------------------------------------------------------------------------
Event logs
Event: {}
```

检查结果：（根据算法原理，结果应该为：收据 ID 为 0 的收据收款人再改回 com2；收据 ID 为 1 的收据价值减少 500，且收据类型保持“正常收据”；新生成收据 ID 为 2 的收据，欠款人为收据 ID 为 1 的收据的欠款人，收款人为 com2，价值为 500，时间和收据 ID 为 1 的收据一样）

```
call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetBill 0
call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetBill 1
call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetBill 2
```

```
[group:1]> call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetBill 0
---------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
---------------------------------------------------------------------------------------------
Return values:
[
    "com1",
    "com2",
    2,
    1000,
    1607488845485,
    1607488905485
]
---------------------------------------------------------------------------------------------

[group:1]> call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetBill 1
---------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
---------------------------------------------------------------------------------------------
Return values:
[
    "com1",
    "com3",
    1,
    500,
    1607488849084,
    1607488909084
]
---------------------------------------------------------------------------------------------

[group:1]> call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetBill 2
---------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
---------------------------------------------------------------------------------------------
Return values:
[
    "com1",
    "com2",
    2,
    500,
    1607488849084,
    1607488909084
]
---------------------------------------------------------------------------------------------
```

测试功能三：申请融资

首先查看一下公司 com3 和银行 bank 目前的资金情况：

```
call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetCompany com3
call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetCompany bank
```

```
[group:1]> call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetCompany com3
---------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
---------------------------------------------------------------------------------------------
Return values:
[
    "com3",
    10000,
    1
]
---------------------------------------------------------------------------------------------

[group:1]> call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetCompany bank
---------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
---------------------------------------------------------------------------------------------
Return values:
[
    "bank",
    10000000,
    0
]
---------------------------------------------------------------------------------------------
```

现在对公司 com3 向 bank 进行融资：

```
call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d Raise com3 bank
```

```
[group:1]> call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d Raise com3 bank
transaction hash: 0x9a16d92d820addf6655479177340d782d96f7ca8d2ba273104c856185611733c
---------------------------------------------------------------------------------------------
transaction status: 0x0
description: transaction executed successfully
---------------------------------------------------------------------------------------------
Output
Receipt message: Success
Return message: Success
Return value: 500
---------------------------------------------------------------------------------------------
Event logs
Event: {}
```

检查结果：（根据算法原理，收款人为公司 com3 的收据为收据 1，价值为 500，融资后该账单收款人改为银行 bank，公司 com3 余额 +500，银行 bank 余额 -500）

```
call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetBill 1
call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetCompany com3
call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetCompany bank
```

```
[group:1]> call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetBill 1
---------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
---------------------------------------------------------------------------------------------
Return values:
[
    "com1",
    "bank",
    3,
    500,
    1607488849084,
    1607488909084
]
---------------------------------------------------------------------------------------------

[group:1]> call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetCompany com3
---------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
---------------------------------------------------------------------------------------------
Return values:
[
    "com3",
    10500,
    1
]
---------------------------------------------------------------------------------------------

[group:1]> call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetCompany bank
---------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
---------------------------------------------------------------------------------------------
Return values:
[
    "bank",
    9999500,
    0
]
---------------------------------------------------------------------------------------------
```

测试功能四：偿还债款

首先查看一下公司 com1 目前的资金情况：

```
call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetCompany com1
```

```
[group:1]> call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetCompany com1
---------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
---------------------------------------------------------------------------------------------
Return values:
[
    "com1",
    10000,
    1
]
---------------------------------------------------------------------------------------------
```

之后公司 com1 偿还所有的收据：

```
call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d Repay com1
```

```
[group:1]> call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d Repay com1
transaction hash: 0x44d4432721c603688458e9983a7812eed505b75cbcf0b468f383dd542811ab98
---------------------------------------------------------------------------------------------
transaction status: 0x0
description: transaction executed successfully
---------------------------------------------------------------------------------------------
Output
Receipt message: Success
Return message: Success
Return value: 2000
---------------------------------------------------------------------------------------------
Event logs
Event: {}
```

检查结果：（根据算法原理，欠款人为公司 com1 的收据为收据 0、1、2，价值一共为 2000，偿还后这些收据类型修改为“已偿还收据”，公司 com1 余额 -2000）

```
call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetBill 0
call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetBill 1
call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetBill 2
call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetCompany com1
```

```
[group:1]> call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetBill 0
---------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
---------------------------------------------------------------------------------------------
Return values:
[
    "com1",
    "com2",
    0,
    1000,
    1607488845485,
    1607488905485
]
---------------------------------------------------------------------------------------------

[group:1]> call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetBill 1
---------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
---------------------------------------------------------------------------------------------
Return values:
[
    "com1",
    "bank",
    0,
    500,
    1607488849084,
    1607488909084
]
---------------------------------------------------------------------------------------------

[group:1]> call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetBill 2
---------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
---------------------------------------------------------------------------------------------
Return values:
[
    "com1",
    "com2",
    0,
    500,
    1607488849084,
    1607488909084
]
---------------------------------------------------------------------------------------------
```

```
[group:1]> call syj 0x29a16a7075e6371858fcfcc943d0ef1a53ce037d GetCompany com1
---------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
---------------------------------------------------------------------------------------------
Return values:
[
    "com1",
    8000,
    1
]
---------------------------------------------------------------------------------------------
```

项目分工：

宋渝杰：实现“生成收据、转让收据、注册和输出”功能

刘依澜：实现“申请融资”功能、进行功能调试和测试

李研通：实现“偿还债款”功能、实验报告和视频录制

项目心得：

关于大作业的前端和后端的实现，虽然我们在期末考试完很快就开始了学习，但是大家都是零基础，只能通过上网自己查询资料，然后安装环境来缓慢进行。通过自学，我们对写前端页面有了较好的掌握，但是最终还是没能彻底搞明白前后端间以及后端和链端间的数据交互问题，最后只能以失败告终。虽然未能在提交日期前实现，但我们一致认为前后端的学习是十分重要的，因此我们决定在之后的寒假生活中会继续进行这方面的学习。