基于区块链的供应链金融平台

项目设计说明以及功能测试文档

18340159 唐瑞怡 18340234 朱莹莹 18340198 叶苗欣（排名不分先后）

一、项目设计

将供应链上的每一笔交易和应收账款单据上链，同时引入第三方可信机构来确认这些信息的交易，例如银行，物流公司等，确保交易和单据的真实性。同时，支持应收账款的转让，融资，清算等，让核心企业的信用可以传递到供应链的下游企业，减小中小企业的融资难度。

首先在智能合约中创建两个表格，assert_manage 和 transaction_record，分别存储资产账户和交易记录的信息。assert_manage 表中存储资产账户以及该资产账户的信用额度值，transaction_record 表中存储各账户之间的交易记录信息。

```
    function createTable() private {
        TableFactory tf = TableFactory(0x1001);
        //创建资产管理表
        tf.createTable("assert_manage", "account", "credit_limit");
        //创建交易记录表
        tf.createTable("transaction_record", "id", "acc1, acc2, money,
status");
    }
```

定义 event，分别为注册资产账户、转移信用额度、添加交易记录、更新交易记录（支付账单）、交易记录金额转让、删除交易记录。

```
    event RegisterEvent(int256 ret, string account, int256 credit_limit);
    event TransferEvent(int256 ret, string from, string to, int256 amount);
    event AddTransactionEvent(int256 ret, string id, string acc1, string
acc2, int256 money);
    event UpdateTransactionEvent(int256 ret, string id, int256 money);
    event SplitTransactionEvent(int256 ret, string old_id, string new_id,
string acc, int256 money);
    event RemoveTransactionEvent(int256 ret,string id);
```

功能一

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。

首先编写一个资产注册函数，可以通过该函数进行资产的注册，例如需要注册一个宝马的公司，则调用 register，参数为宝马和它的信用额度值。具体实现也较为简单，首先查询账户是否已经存在，存在则无需再次创建，修改返回值后直接返回即可；否则打开资产管理表，创建一个新的信息元组，插入到该表中。

```
    //资产注册
    function register(string account, int256 credit_limit) public
returns(int256){
```

```
        int256 ret_code = 0;
        int256 ret = 0;
        int256 temp_credit_limit = 0;
        // 查询账户是否存在
        (ret, temp_credit_limit) = select(account);
        if(ret != 0) {
            Table table = openAssetTable();

            Entry entry = table.newEntry();
            entry.set("account", account);
            entry.set("credit_limit", int256(credit_limit));
            // 插入
            int count = table.insert(account, entry);
            if (count == 1) {
                // 成功
                ret_code = 0;
            } else {
                // 失败 无权限或者其他错误
                ret_code = -2;
            }
        } else {
            // 账户已存在
            ret_code = -1;
        }
        emit RegisterEvent(ret_code, account, credit_limit);
        return ret_code;
    }
```

注册好资产账户后，如果两个资产直接发生了交易，就添加交易记录到交易记录表中。具体实现为首先查询该交易id是否已经存在，存在则无需再次创建，修改返回值后直接返回即可；否则打开交易记录表，创建一个新的信息元组，插入该中即可。这样就实现了交易上链。

```
    //添加交易记录
    function addTransaction(string id, string acc1, string acc2, int256
money) public returns(int256){
        int256 ret_code = 0;
        int256 ret = 0;
        bytes32[] memory str_list = new bytes32[](2);
        int256[] memory int_list = new int256[](3);

        // 查询交易是否存在
        (int_list, str_list) = select_transaction(id);
        if(int_list[0] != int256(0)) {
            Table table = openTransactionTable();

            Entry entry0 = table.newEntry();
            entry0.set("id", id);
            entry0.set("acc1", acc1);
            entry0.set("acc2", acc2);
            entry0.set("money", int256(money));
            entry0.set("status", int256(money));
            // 插入
            int count = table.insert(id, entry0);
            if (count == 1) {
                // 将欠款人的信用额度转移一部分给债主
                ret = transfer(acc2,acc1,money);
                // 信用额度转让失败
```

```
                if(ret != 0) {
                    ret_code = -3;
                }
                //成功
                else {
                    ret_code = 0;
                }
            }
            else {
                // 失败 无权限或者其他错误
                ret_code = -2;
            }
        }
        else {
            // 交易ID已存在
            ret_code = -1;
        }
        emit AddTransactionEvent(ret_code, id, acc1, acc2, money);
        return ret_code;
    }
```

功能二

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。

实现转让上链具体映射为将各资产账户之间欠条的拆分，车企给轮胎的应收账款的单据，也就是交易记录中的一部分金额可以转让给轮毂公司，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。

首先查询该交易记录id是否已经存在，如果存在的话，将第一个资产账户与第二个资产账户的交易的部分金额转让给第三个账户。另外也要注意一些例如第三个资产账户不存在或者原先金额小于当前要转移的金额值等一些异常问题。

```
    //欠条拆分
    function splitTransaction(string old_id, string new_id, string acc, int256 money) public returns(int256) {
        int256 ret_code = 0;
        int256 ret = 0;
        int temp = 0;
        bytes32[] memory str_list = new bytes32[](2);
        int256[] memory int_list = new int256[](3);
        string[] memory acc_list = new string[](2);
        // 查询该欠条是否存在
        (int_list, str_list) = select_transaction(old_id);

        if(int_list[0] == 0) {
            // acc不存在
            (ret, temp) = select(acc);
            if(ret != 0) {
                ret_code = -5;
                emit SplitTransactionEvent(ret_code, old_id, new_id, acc, money);
                return ret_code;
            }

            if(int_list[2] < money){    // 拆分的金额大于欠条余额
```

```
                ret_code = -2;
                emit SplitTransactionEvent(ret_code, old_id, new_id, acc, money);
                return ret_code;
            }

            // acc1先转让给acc2，然后acc2再转让给acc
            (ret,acc_list) = updateTransaction(old_id, money);
            if (ret != 0) {
                ret_code = -4;
                emit SplitTransactionEvent(ret_code, old_id, new_id, acc, money);
                return ret_code;
            }
            ret = addTransaction(new_id, acc, byte32ToString(str_list[1]), money);
            if (ret != 0) {
                ret_code = -3;
                emit SplitTransactionEvent(ret_code, old_id, new_id, acc, money);
                return ret_code;
            }

        } else {    // 拆分的欠条id不存在
            ret_code = -1;
        }

        emit SplitTransactionEvent(ret_code, old_id, new_id, acc, money);
        return ret_code;
    }
```

功能三

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。

银行是否决定同意融资可以根据资产账户的信用额度来判断，可以根据账款单据对资产账户信用额度进行修改。

信用额度转移功能的实现首先查看两个需要转移的账户是否都存在以及金额是否在合适范围内，否则报异常。如果没有异常产生，对信用额度值进行更改，然后对资产管理表中的两个账户信息都进行更新。

```
    //信用额度转移
    function transfer(string from, string to, int256 amount) public returns(int256) {
        // 查询转移资产账户信息
        int ret_code = 0;
        int256 ret = 0;
        int256 from_credit_limit = 0;
        int256 to_credit_limit = 0;

        (ret, from_credit_limit) = select(from);
        if(ret != 0) {
            ret_code = -1;
            // 转移账户不存在
            emit TransferEvent(ret_code, from, to, amount);
            return ret_code;

```

```
        }

        (ret, to_credit_limit) = select(to);
        if(ret != 0) {
            ret_code = -2;
            // 接收资产的账户不存在
            emit TransferEvent(ret_code, from, to, amount);
            return ret_code;
        }

        if(from_credit_limit < amount) {
            ret_code = -3;
            // 转移资产的账户金额不足
            emit TransferEvent(ret_code, from, to, amount);
            return ret_code;
        }

        if (to_credit_limit + amount < to_credit_limit) {
            ret_code = -4;
            // 接收账户金额溢出
            emit TransferEvent(ret_code, from, to, amount);
            return ret_code;
        }

        Table table = openAssetTable();

        Entry entry0 = table.newEntry();
        entry0.set("account", from);
        entry0.set("credit_limit", int256(from_credit_limit - amount));
        // 更新转账账户
        int count = table.update(from, entry0, table.newCondition());
        if(count != 1) {
            ret_code = -5;
            // 失败 无权限或者其他错误
            emit TransferEvent(ret_code, from, to, amount);
            return ret_code;
        }

        Entry entry1 = table.newEntry();
        entry1.set("account", to);
        entry1.set("credit_limit", int256(to_credit_limit + amount));
        // 更新接收账户
        table.update(to, entry1, table.newCondition());
        emit TransferEvent(ret_code, from, to, amount);
        return ret_code;
    }

```

功能四

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。

查询该交易记录是否存在，如果存在且金额属于合适的范围内，对交易记录进行更新，并且将信用额度返还。实现核心企业向下游企业支付相应的欠款。

```
//更新交易记录(支付欠条)

```

```
    function updateTransaction(string id, int256 money) public
returns(int256, string[]){
        int256 ret_code = 0;
        bytes32[] memory str_list = new bytes32[](2);
        int256[] memory int_list = new int256[](3);
        string[] memory acc_list = new string[](2);
        // 查询该欠条是否存在
        (int_list, str_list) = select_transaction(id);
        acc_list[0] = byte32ToString(str_list[0]);
        acc_list[1] = byte32ToString(str_list[1]);

        if(int_list[0] == 0) { // 交易ID存在
            // 还款金额大于欠款金额
            if(int_list[2] < money){
                ret_code = -2;
                emit UpdateTransactionEvent(ret_code, id, money);
                return (ret_code, acc_list);
            }
            // 更新交易状态
            Table table = openTransactionTable();

            Entry entry0 = table.newEntry();
            entry0.set("id", id);
            entry0.set("acc1", byte32ToString(str_list[0]));
            entry0.set("acc2", byte32ToString(str_list[1]));
            entry0.set("money", int_list[1]);
            entry0.set("status", (int_list[2] - money));

            // 更新欠条
            int count = table.update(id, entry0, table.newCondition());
            if(count != 1) {
                ret_code = -3;
                // 失败 无权限或者其他错误
                emit UpdateTransactionEvent(ret_code, id, money);
                return (ret_code,acc_list);
            }

            // 信用额度返还
            int256 temp = transfer(byte32ToString(str_list[0]),
byte32ToString(str_list[1]),money);
            if(temp != 0){
                ret_code = -4 * 10 + temp;
                emit UpdateTransactionEvent(ret_code, id, money);
                return (ret_code,acc_list);
            }
            ret_code = 0;
        } else { // 交易ID不存在
            ret_code = -1;
        }
        emit UpdateTransactionEvent(ret_code, id, money);
        return (ret_code,acc_list);
    }
```

二、实验结果

以下呈现`bmw`(汽车公司)、`tire`(轮胎公司)、`lungu`(轮毂公司)、bank(银行)四方之间的交易，具体涉及的交易有借钱、还钱、显示、欠条拆分、融资等。

- 首先，部署合约Account上链， 得到合约地址：

```
[group:1]> deploy Account
transaction hash: 0x82b99ed6e0c65a5c02bff3e7277fd3904b8b49887d21de2549ea227b4db
e8f48
contract address: 0x5629ed73ecd0387d3fb00d2050359c1b026b0e0b
```

- 调用`register`函数，实现`bmw`汽车公司和`tire`轮胎公司的注册，参数为公司名字和它的信用额度值。
  - `bmw`注册信用额度为10000.

```
[group:1]> call Account 0x5629ed73ecd0387d3fb00d2050359c1b026b0e0b register bmw 10000
transaction hash: 0xce8c4aa2c3505b1fd47e799312a7db1139f64d3816acad2617583bc36c310a33
----------------------------------------------------------------------------------------
transaction status: 0x0
description: transaction executed successfully
----------------------------------------------------------------------------------------
Output
Receipt message: Success
Return message: Success
Return value: 0
----------------------------------------------------------------------------------------
Event logs
Event: {"RegisterEvent":[[0,"bmw",10000]]}
```

  - tire注册信用额度为1000.

```
[group:1]> call Account 0x5629ed73ecd0387d3fb00d2050359c1b026b0e0b register tire 1000
transaction hash: 0x32dd4cb740c65d514c60c107a9b133fa4333a06d7608cc9b791b7c2c07062acd
----------------------------------------------------------------------------------------
transaction status: 0x0
description: transaction executed successfully
----------------------------------------------------------------------------------------
Output
Receipt message: Success
Return message: Success
Return value: 0
----------------------------------------------------------------------------------------
Event logs
Event: {"RegisterEvent":[[0,"tire",1000]]}
```

2.1 功能一：签发应收账款交易上链

- 调用`addTransaction`函数，添加一笔发生在`tire`和`bmw`之间的交易，账款为100，编号为0009，并将交易记录加入到了交易记录表中：

```
[group:1]> call Account 0x5629ed73ecd0387d3fb00d2050359c1b026b0e0b addTransaction 0009 tire bmw 100
transaction hash: 0xbc5365bb4429571fae02acde105a2f7ac8509187ec6f0042a679adc1cb6fd81e
----------------------------------------------------------------------------------------
transaction status: 0x0
description: transaction executed successfully
----------------------------------------------------------------------------------------
Output
Receipt message: Success
Return message: Success
Return value: 0
----------------------------------------------------------------------------------------
Event logs
Event: {"TransferEvent":[[0,"bmw","tire",100]],"AddTransactionEvent":[[0,"0009","tire","bmw",100]]}
```

  输出Success并返回0，表示借钱成功。

- 调用`updateTransaction`函数，实现支付，即针对0009的交易，`bmw`还款100给`tire`：

```
[group:1]> call Account 0x5629ed73ecd0387d3fb00d2050359c1b026b0e0b updateTransaction 0009 100
transaction hash: 0x4d3a23038ba912dc4f7c62781429cae9165aee3a101cc50cc2b01697d694fc09
---------------------------------------------------------------------------------------------
transaction status: 0x0
description: transaction executed successfully
---------------------------------------------------------------------------------------------
Output
Receipt message: Success
Return message: Success
Return value: 0
---------------------------------------------------------------------------------------------
Event logs
Event: {"UpdateTransactionEvent":[[0,"0009",100]],"TransferEvent":[[0,"tire","bmw",100]]}
```

此时，输出0009号交易记录的状态：

```
[group:1]> call Account 0x5629ed73ecd0387d3fb00d2050359c1b026b0e0b select_transaction 0009
---------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
---------------------------------------------------------------------------------------------
Return values:
[
    [
        0,
        100,
        0
    ],
    [
        "tire\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u000
0\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000",
        "bmw\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000
\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000"
    ]
]
---------------------------------------------------------------------------------------------
```

发现发生交易的双方是`tire`和`bmw`，应还金额（第二行）为100，待还金额（第三行）为0，表示还款成功。

2.2 功能二：实现应收账款的转让上链

- 首先，添加一笔新的交易记录：`bmw`向`tire`借款100，交易编号为0101

```
[group:1]> call Account 0x5629ed73ecd0387d3fb00d2050359c1b026b0e0b addTransaction 0101 tire bmw 100
transaction hash: 0xcbaab9318f13af49d1f81a472d665cf0976b7d68483c0040e412968aaa4d28f3
---------------------------------------------------------------------------------------------
transaction status: 0x0
description: transaction executed successfully
---------------------------------------------------------------------------------------------
Output
Receipt message: Success
Return message: Success
Return value: 0
---------------------------------------------------------------------------------------------
Event logs
Event: {"TransferEvent":[[0,"bmw","tire",100]],"AddTransactionEvent":[[0,"0101","tire","bmw",100]]}
```

此时，调用`select_transaction`函数，查看编号为0101的交易记录：

```
[group:1]> call Account 0x5629ed73ecd0387d3fb00d2050359c1b026b0e0b select_transaction 0101
---------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
---------------------------------------------------------------------------------------------
Return values:
[
    [
        0,
        100,
        100
    ],
    [
        "tire\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u000
0\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000",
        "bmw\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000
\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000"
    ]
]
---------------------------------------------------------------------------------------------
```

可以看到，交易双方为`tire`和`bmw`，应收账款=待收账款=100.

- 注册轮毂公司`lungu`，注册信用额度为1000：

```
[group:1]> call Account 0x5629ed73ecd0387d3fb00d2050359c1b026b0e0b register lungu 1000
transaction hash: 0x0f84eed5fbb121b0373789dfcb78c17c8eabebbf5a1bbfead8775ea235c4e663
-------------------------------------------------------------------------------------------
transaction status: 0x0
description: transaction executed successfully
-------------------------------------------------------------------------------------------
Output
Receipt message: Success
Return message: Success
Return value: 0
-------------------------------------------------------------------------------------------
Event logs
Event: {"RegisterEvent":[[0,"lungu",1000]]}
```

- 调用`splitTransaction`函数，实现应收账款的转让，将原来的0101交易记录中，`tire`轮胎公司的应收账款100中的50转让给轮毂公司`lungu`，转让的交易编号为0110：

```
[group:1]> call Account 0x5629ed73ecd0387d3fb00d2050359c1b026b0e0b splitTransaction 0101 0110 lungu 50
transaction hash: 0xbf3546a82bf52edeefc048b48688ecc2f5f4f91adb1d1307e4756a435d22b4d3
-------------------------------------------------------------------------------------------
transaction status: 0x0
description: transaction executed successfully
-------------------------------------------------------------------------------------------
Output
Receipt message: Success
Return message: Success
Return value: 0
-------------------------------------------------------------------------------------------
Event logs
Event: {"UpdateTransactionEvent":[[0,"0101",50]],"SplitTransactionEvent":[[0,"0101","0110","lungu",50]],"TransferEvent":
[[0,"tire","bmw",50],[0,"bmw","lungu",50]],"AddTransactionEvent":[[0,"0110","lungu","bmw",50]]}
```

然后，调用`select_transaction`函数，查看编号为0101和0110的交易记录：

- 在0101交易记录中，发现交易双方为`tire`和`bmw`，表示`bmw`应还`tire`账款为100，待还账款为50，表示账款从`tire`转出成功。

```
[group:1]> call Account 0x5629ed73ecd0387d3fb00d2050359c1b026b0e0b select_transaction 0101
-------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
-------------------------------------------------------------------------------------------
Return values:
[
    [
        0,
        100,
        50
    ],
    [
        "tire\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u000
0\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000",
        "bmw\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000
\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000"
    ]
]
-------------------------------------------------------------------------------------------
```

- 在0110交易记录中，发现交易双方为`lungu`和`bmw`，表示`bmw`应还`lungu`账款为50，待还账款为50，表示账款转入到`lungu`成功。

```
[group:1]> call Account 0x5629ed73ecd0387d3fb00d2050359c1b026b0e0b select_transaction 0110
-------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
-------------------------------------------------------------------------------------------
Return values:
[
    [
        0,
        50,
        50
    ],
    [
        "lungu\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u00
00\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000",
        "bmw\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000
\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000"
    ]
]
-------------------------------------------------------------------------------------------
```

2.3 功能三：利用应收账款向银行融资上链

- 注册银行`bank`，注册信用额度为1000,000,000：

```
[group:1]> call Account 0x5629ed73ecd0387d3fb00d2050359c1b026b0e0b register bank 1000000000
transaction hash: 0x27e5432bc586005e444221cbe261e58a609735066ddf4338e0e262f62bd761d2
-------------------------------------------------------------------------------------------
transaction status: 0x0
description: transaction executed successfully
-------------------------------------------------------------------------------------------
Output
Receipt message: Success
Return message: Success
Return value: -1
-------------------------------------------------------------------------------------------
Event logs
Event: {"RegisterEvent":[[-1,"bank",1000000000]]}
```

注意：此时返回的值为`-1`，但不意味着注册失败，只是之前测试的时候已经注册过`bank`了（不影响后序实验）

- 调用`splitTransaction`函数，实现`tire`可以利用和`bmw`之间的应收账款（编号为0101）向银行申请融资，融资金额为10，此交易编号为0111.

```
[group:1]> call Account 0x5629ed73ecd0387d3fb00d2050359c1b026b0e0b splitTransaction 0101 0111 bank 10
transaction hash: 0x70be0ebc8d2775707f864b36dd1bbed4600bb0f66533c7134cc560b2622f8d14
-------------------------------------------------------------------------------------------
transaction status: 0x0
description: transaction executed successfully
-------------------------------------------------------------------------------------------
Output
Receipt message: Success
Return message: Success
Return value: 0
-------------------------------------------------------------------------------------------
Event logs
Event: {"UpdateTransactionEvent":[[0,"0101",10]],"SplitTransactionEvent":[[0,"0101","0111","bank",10]],"TransferEvent":[
[0,"tire","bmw",10],[0,"bmw","bank",10]],"AddTransactionEvent":[[0,"0111","bank","bmw",10]]}
```

- 调用`select_transaction`函数，查看0101交易记录：

```
[group:1]> call Account 0x5629ed73ecd0387d3fb00d2050359c1b026b0e0b select_transaction 0101
-------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
-------------------------------------------------------------------------------------------
Return values:
[
    [
        0,
        100,
        40
    ],
    [
        "tire\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u000
0\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000",
        "bmw\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000
\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000"
    ]
]
-------------------------------------------------------------------------------------------
```

此时，发现交易双方依旧为`tire`和`bmw`，但待还账款从50变为40。这是由于`tire`利用此交易向银行发起融资，相当于`tire`从`bank`那里提前获得了`bmw`待还账款中的10，因此`bmw`待还`tire`的款项减少为40，减少的10的还款接收方为`bank`。

- 调用`select_transaction`函数，查看0111交易记录：

```
[group:1]> call Account 0x5629ed73ecd0387d3fb00d2050359c1b026b0e0b select_transaction 0111
-------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
-------------------------------------------------------------------------------------------
Return values:
[
    [
        0,
        10,
        10
    ],
    [
        "bank\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u000
0\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000",
        "bmw\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000
\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000"
    ]
]
-------------------------------------------------------------------------------------------
```

发现交易双方是`bank`和`bmw`，应还款项=待还款项=10，说明`tire`融资成功。

2.4 功能四：应收账款支付结算上链

- 调用`updateTransaction`函数，对编号为0101的交易记录实现还款，金额为20，即`bmw`向`tire`还款20：

```
[group:1]> call Account 0x5629ed73ecd0387d3fb00d2050359c1b026b0e0b updateTransaction 0101 20
transaction hash: 0xcef14c5741b14f1096a00dac1bcc8e78ab02abb25dbd89251b2d96739a971f45
---------------------------------------------------------------------------------------------
transaction status: 0x0
description: transaction executed successfully
---------------------------------------------------------------------------------------------
Output
Receipt message: Success
Return message: Success
Return value: 0
---------------------------------------------------------------------------------------------
Event logs
Event: {"UpdateTransactionEvent":[[0,"0101",20]],"TransferEvent":[[0,"tire","bmw",20]]}
```

此时，调用`select_transaction`函数，查看此交易的最新状态：发现待还账款从40减少到20，说明还款成功。

```
[group:1]> call Account 0x5629ed73ecd0387d3fb00d2050359c1b026b0e0b select_transaction 0101
---------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
---------------------------------------------------------------------------------------------
Return values:
[
    [
        0,
        100,
        20
    ],
    [
        "tire\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u000
0\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000",
        "bmw\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000
\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000"
    ]
]
---------------------------------------------------------------------------------------------
```

- 调用`updateTransaction`函数，对编号为0110的交易记录实现还款，金额为50，即`bmw`向下游企业`lungu`还款50：

```
[group:1]> call Account 0x5629ed73ecd0387d3fb00d2050359c1b026b0e0b updateTransaction 0110 50
transaction hash: 0x36ca7ed67d9d00344ce59b450403dedc6a215b315ab25a7f657d22ed63e30fac
---------------------------------------------------------------------------------------------
transaction status: 0x0
description: transaction executed successfully
---------------------------------------------------------------------------------------------
Output
Receipt message: Success
Return message: Success
Return value: 0
---------------------------------------------------------------------------------------------
Event logs
Event: {"UpdateTransactionEvent":[[0,"0110",50]],"TransferEvent":[[0,"lungu","bmw",50]]}
```

此时，调用`select_transaction`函数，查看编号为0110的交易记录：

```
[group:1]> call Account 0x5629ed73ecd0387d3fb00d2050359c1b026b0e0b select_transaction 0110
---------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
---------------------------------------------------------------------------------------------
Return values:
[
    [
        0,
        50,
        0
    ],
    [
        "lungu\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u00
00\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000",
        "bmw\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000
\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000"
    ]
]
---------------------------------------------------------------------------------------------
```

发现交易双方`lungu`和`bmw`之间的账款已还清，待还账款从50减少到0，表明还款成功。

- 调用`updateTransaction`函数，对编号为0111的交易记录实现还款，金额为10，即`bmw`向`bank`还款10：

  ```
  [group:1]> call Account 0x5629ed73ecd0387d3fb00d2050359c1b026b0e0b updateTransaction 0111 10
  transaction hash: 0x411f52c925aa53255d9811739c4fa0c77956febf0ff1a4c504ad30861c053877
  ---------------------------------------------------------------------------------------------
  transaction status: 0x0
  description: transaction executed successfully
  ---------------------------------------------------------------------------------------------
  Output
  Receipt message: Success
  Return message: Success
  Return value: 0
  ---------------------------------------------------------------------------------------------
  Event logs
  Event: {"UpdateTransactionEvent":[[0,"0111",10]],"TransferEvent":[[0,"bank","bmw",10]]}
  ```

  此时，调用`select_transaction`函数，查看编号为0111的交易记录：

  ```
  [group:1]> call Account 0x5629ed73ecd0387d3fb00d2050359c1b026b0e0b select_transaction 0111
  ---------------------------------------------------------------------------------------------
  Return code: 0
  description: transaction executed successfully
  Return message: Success
  ---------------------------------------------------------------------------------------------
  Return values:
  [
      [
          0,
          10,
          0
      ],
      [
          "bank\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u000
  0\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000",
          "bmw\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000
  \u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000"
      ]
  ]
  ---------------------------------------------------------------------------------------------
  ```

  发现交易双方`bank`和`bmw`之间的账款已还清，待还账款从10减少到0，表明还款成功。