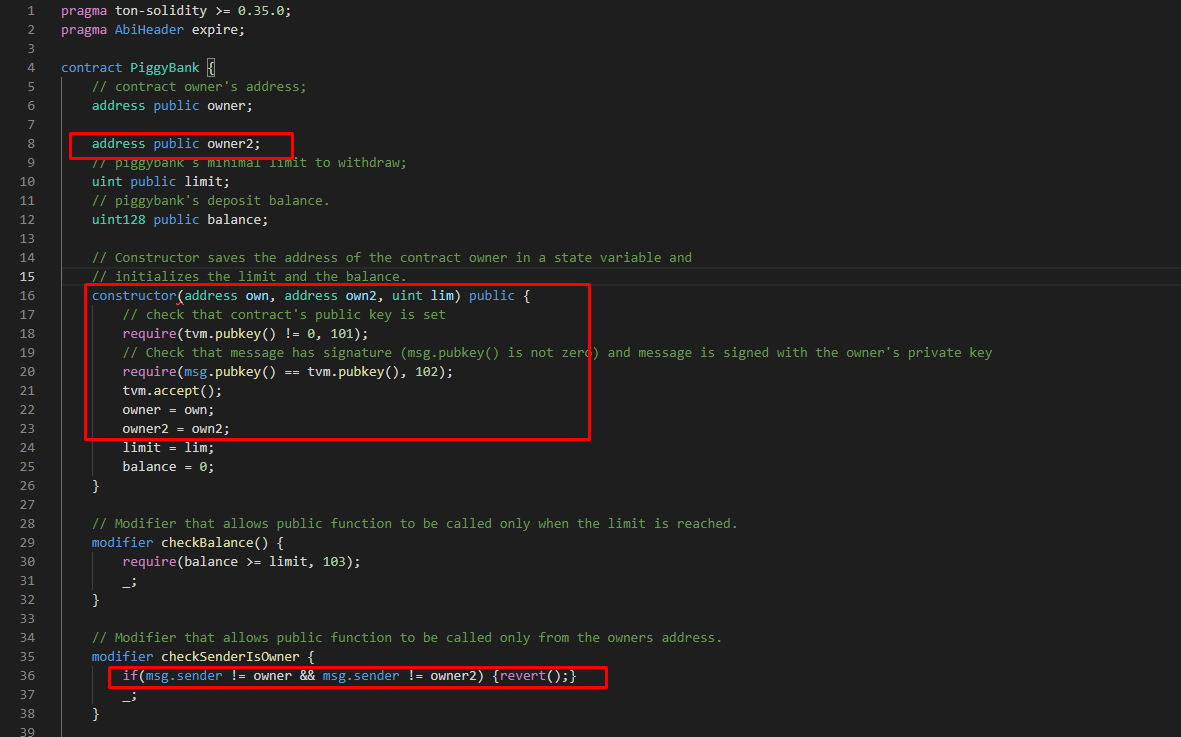
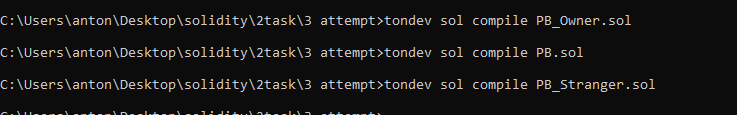
1. Редактируем код PiggyBank под двух владельцев



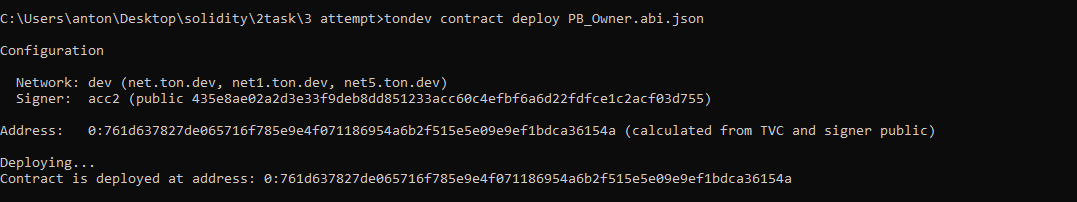
1. Компилируем все контракты



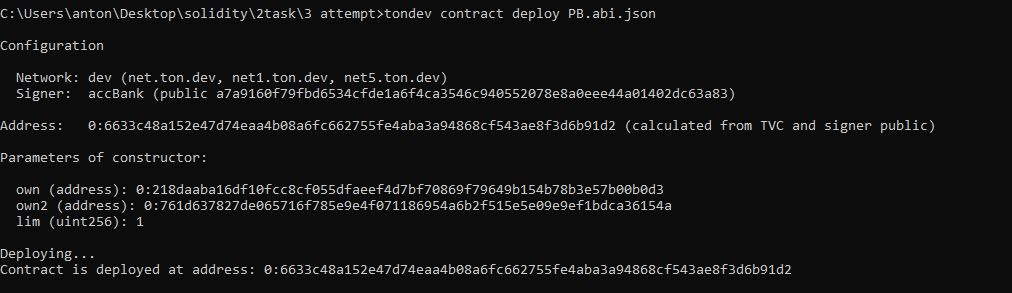
1. Деплоим 1 ownerа с адресом 0:218daaba16df10fcc8cf055dfaeef4d7bf70869f79649b154b78b3e57b00b0d3



1. Деплоим 2 ownera с адрессом 0:761d637827de065716f785e9e4f071186954a6b2f515e5e09e9ef1bdca36154a



1. Деплоим контракт банка и указываем в параметры 2 владельцев, которых до этогоо задеплоили. 0:6633c48a152e47d74eaa4b08a6fc662755fe4aba3a94868cf543ae8f3d6b91d2

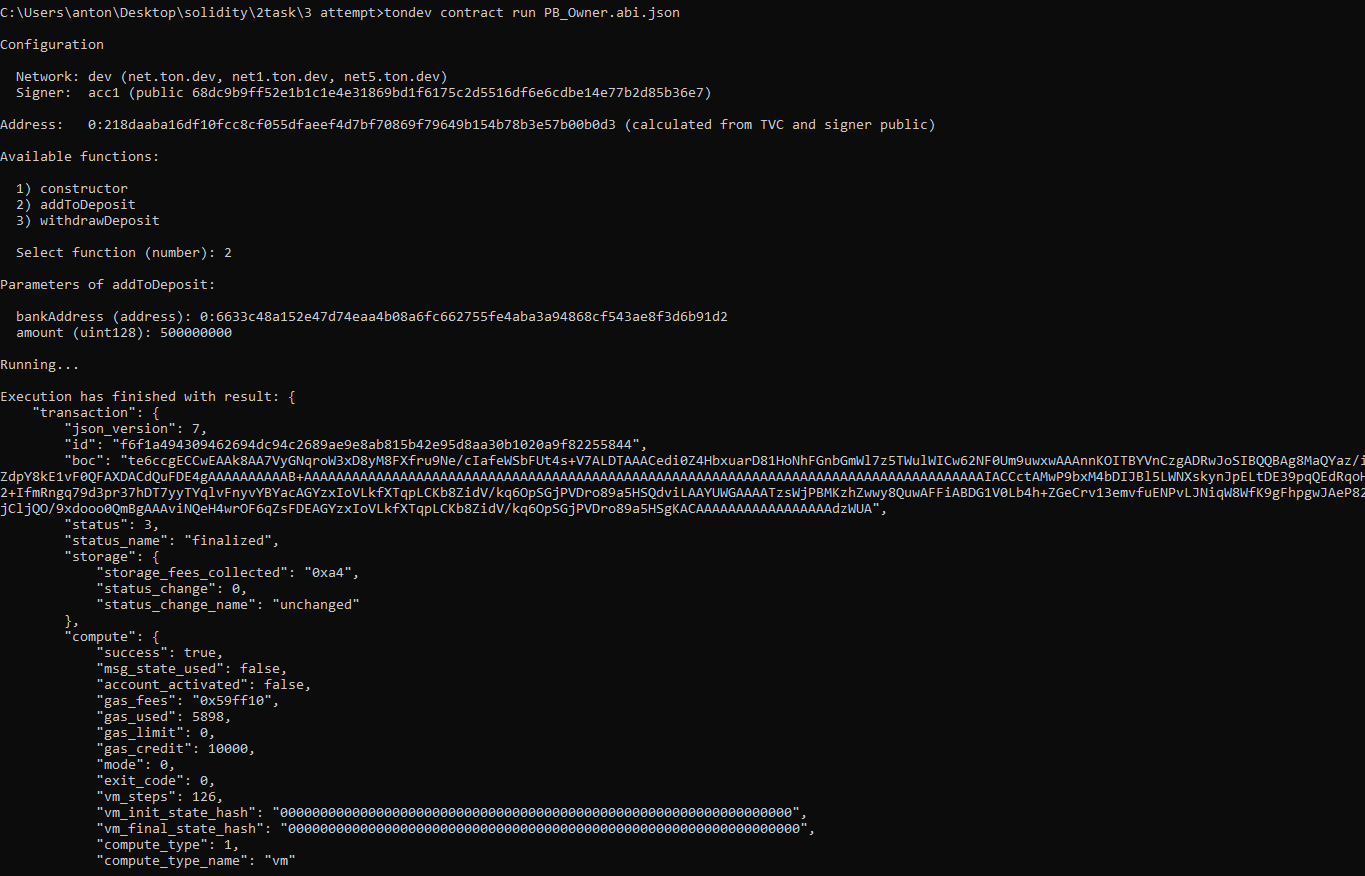


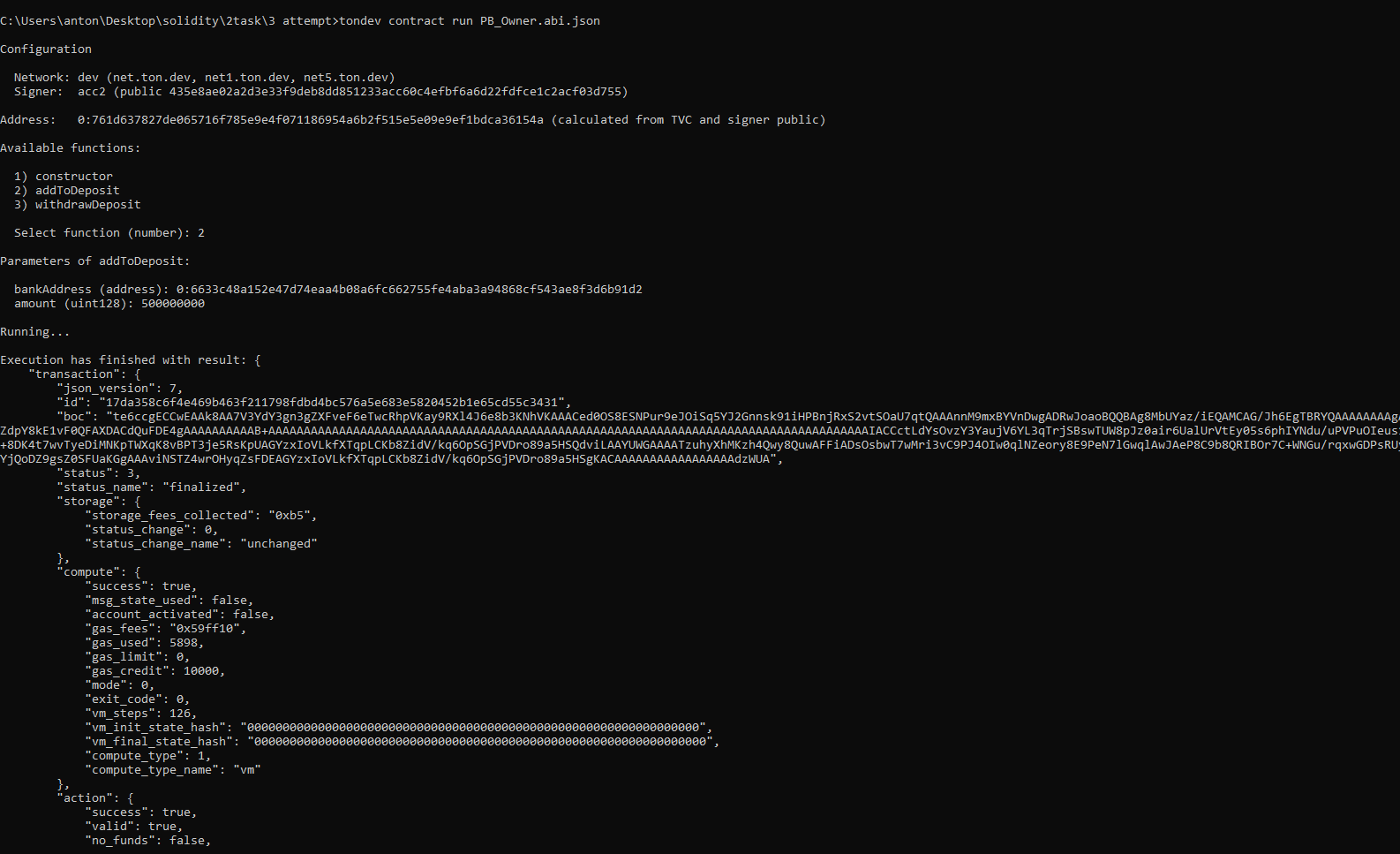
1. Деплоим обычного юзера (stranger)

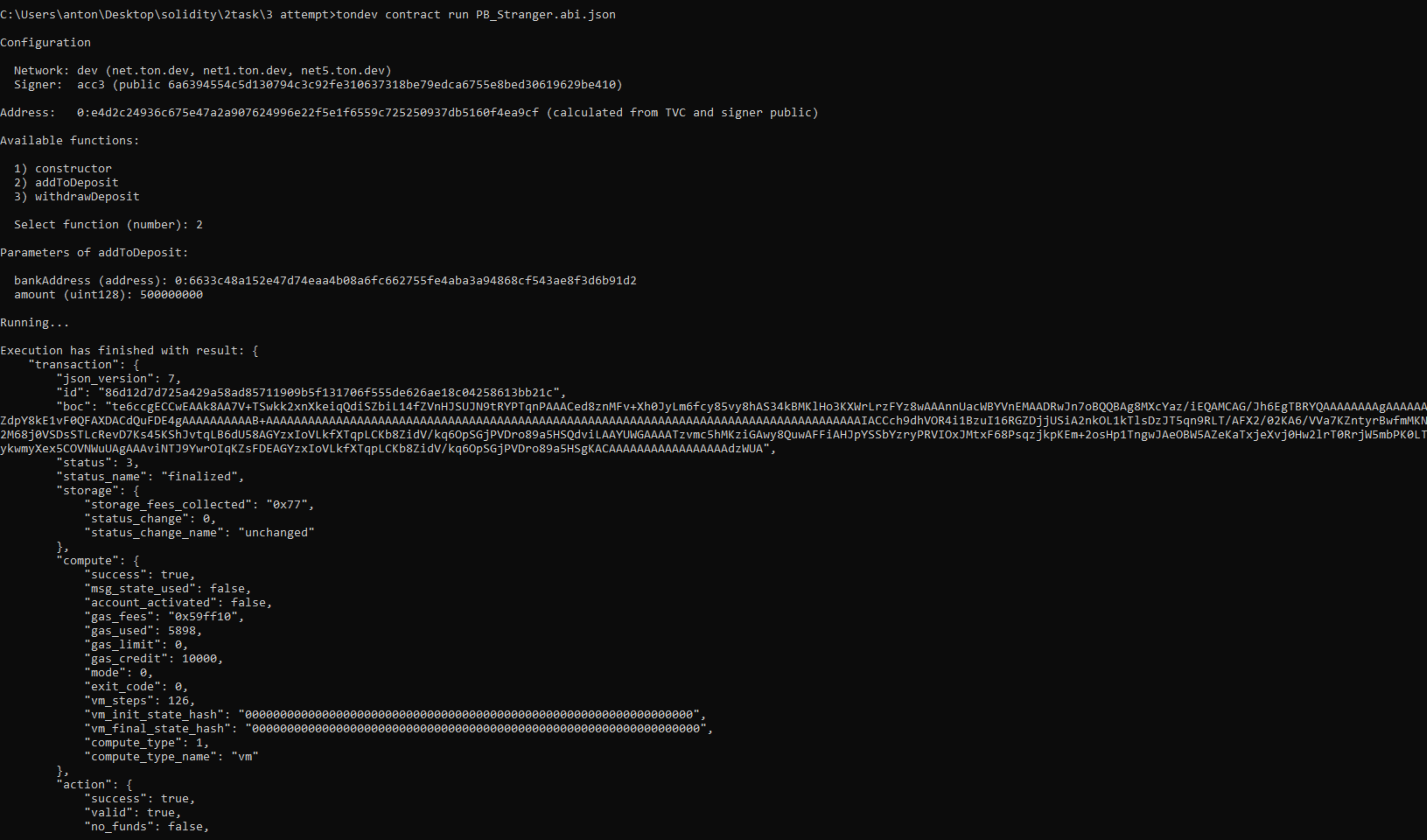
0:e4d2c24936c675e47a2a907624996e22f5e1f6559c725250937db5160f4ea9cf



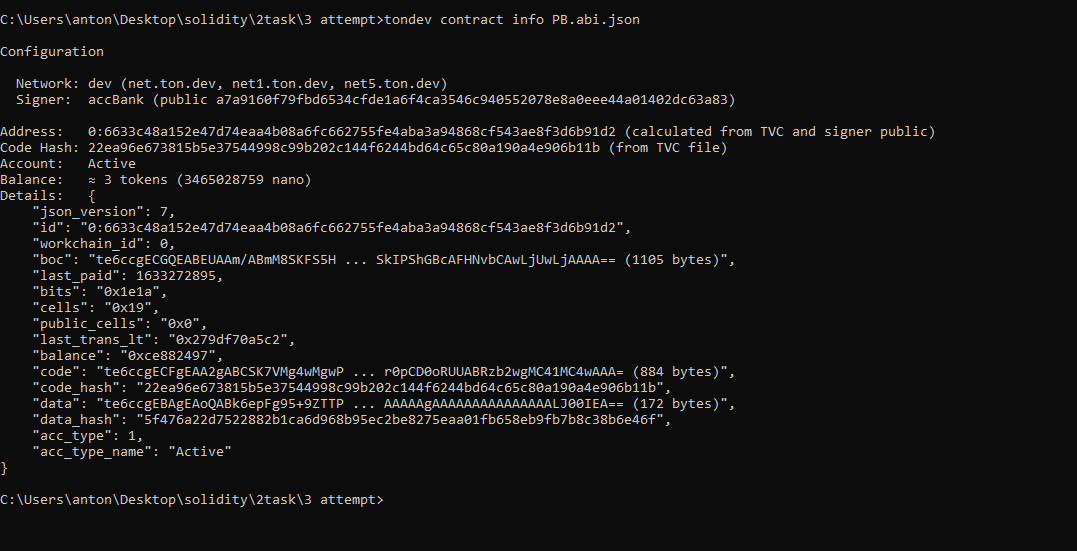
1. Переводим по половине токена с каждого аккаунта



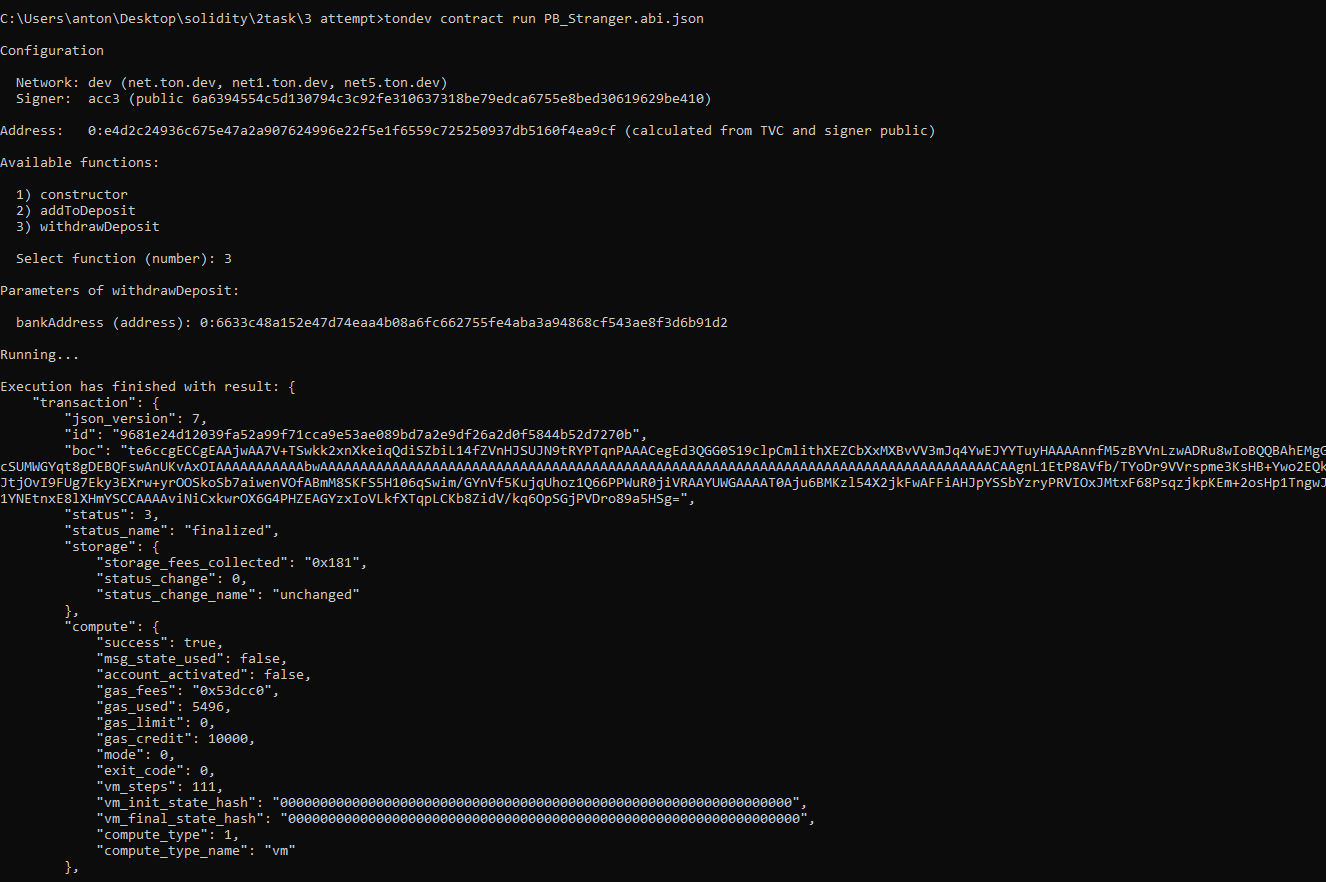


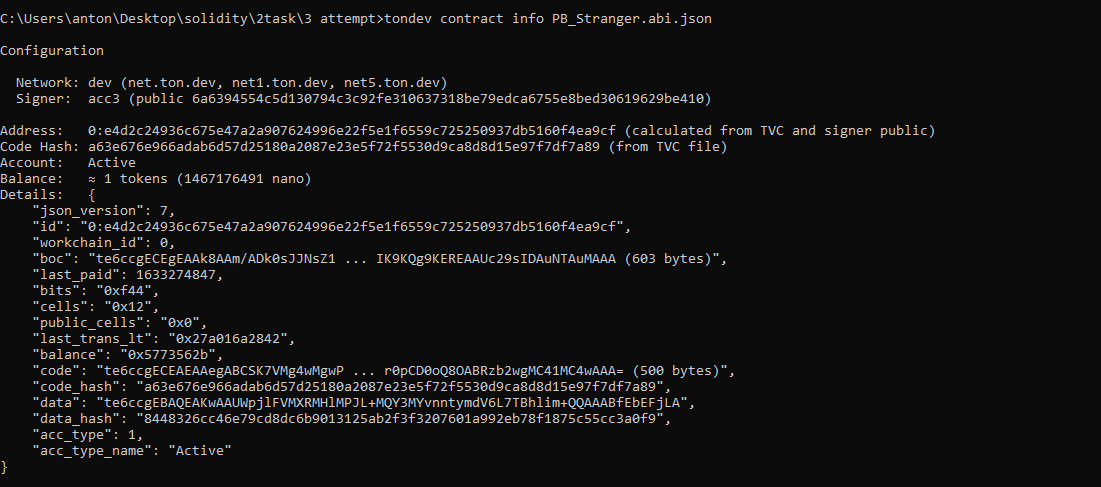


1. Проверяем баланс банка (изначально было 2 монетки на балансе, то есть 1.5 монетки зачислилось, все верно)

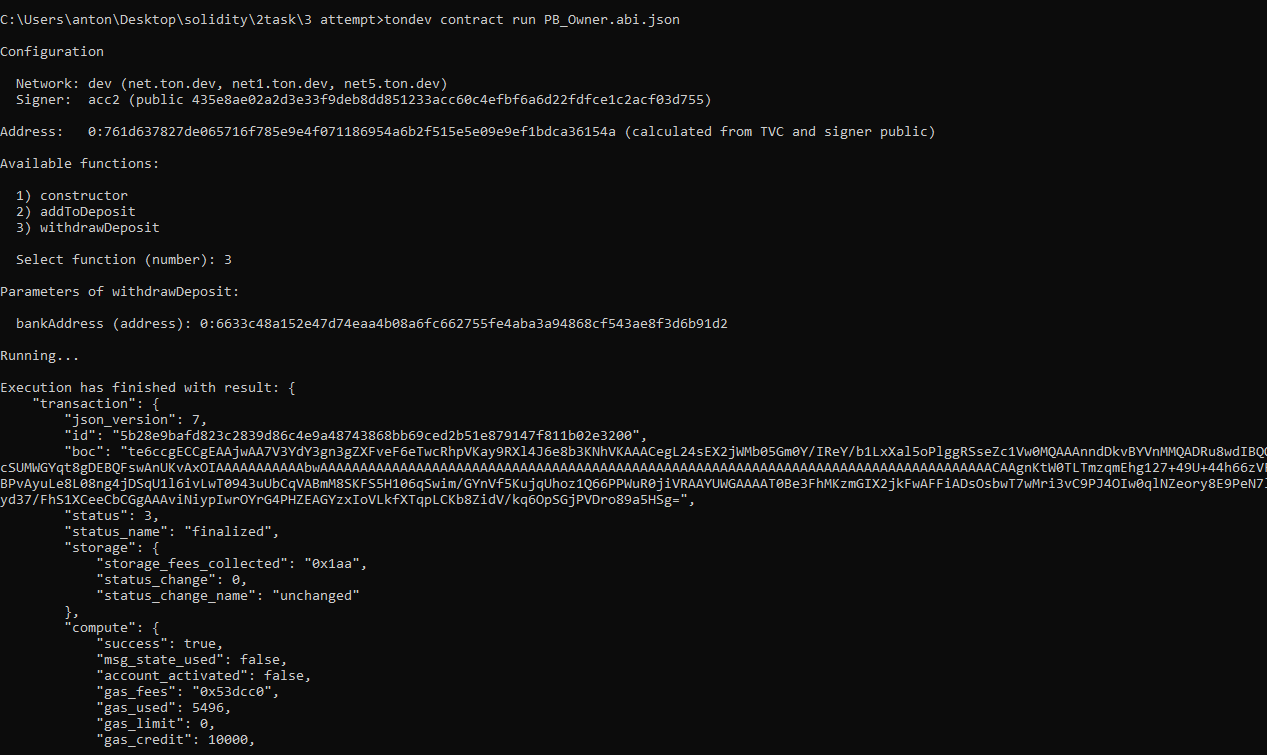


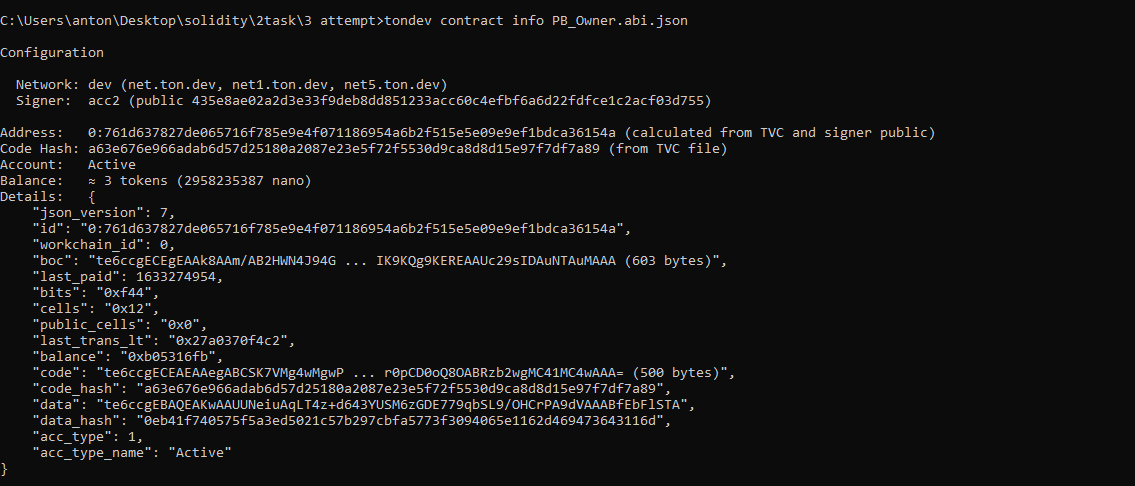
1. Пробуем снять баланс банка с разных аккаунтов

Stranger  


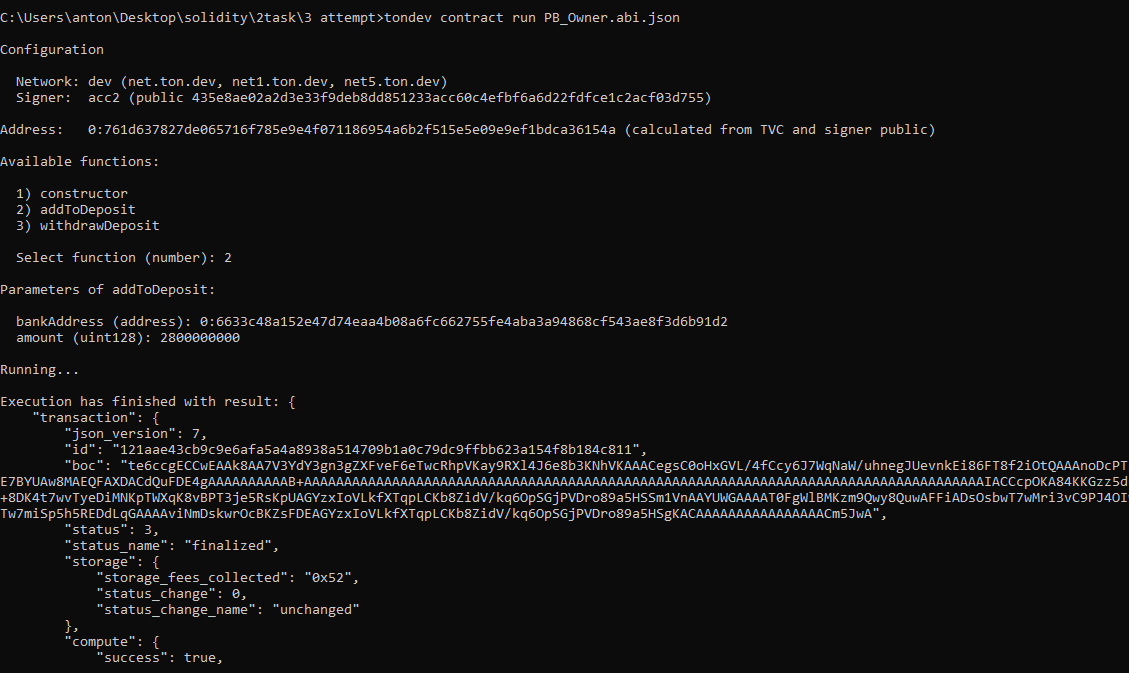
На аккаунте так же 1.5 монеты, как и до выполнения операции (т.к. изначально было 2 монеты, пол монеты перевел в банк)

2 owner

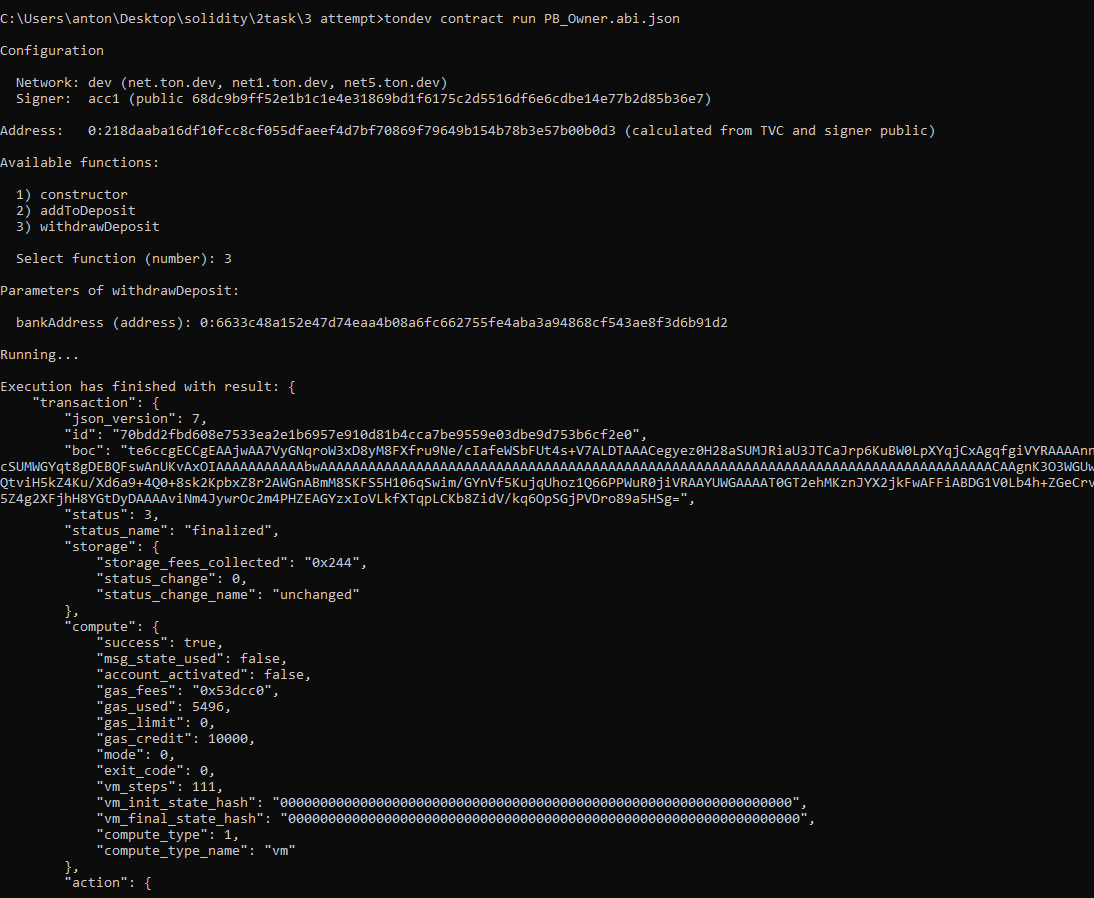


На аккаунте стало 3 монеты, значит операция прошла успешно (т.к. изначально было 2 монеты, пол монеты перевел в банк и потом снял еще 1.5 монеты, итого 3)

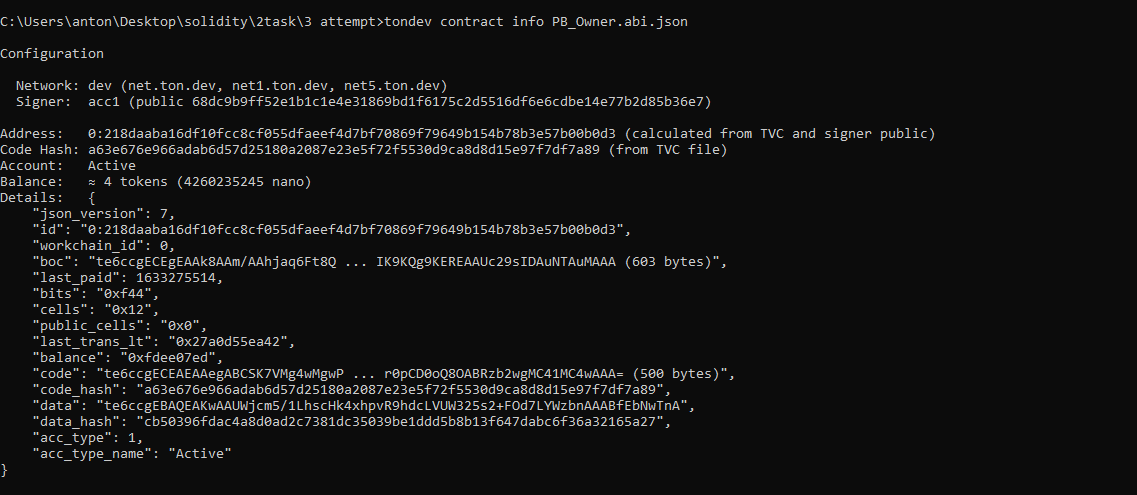
Переводим деньги обратно в банк, чтобы проверить операцию с первым владельцем (2.8 монеты)



1 owner



На аккаунте стало 4.2 монеты, значит все прошло успешно (т.к. изначально было 2 монеты, потом пол монеты было отправлено в банк и затем получено 2.8 монеты итого ~4.3)



Код:

PiggyBank (PB.sol)

pragma ton-solidity >= 0.35.0;

pragma AbiHeader expire;

contract PiggyBank {

    // contract owner's address;

    address public owner;

    address public owner2;

    // piggybank's minimal limit to withdraw;

    uint public limit;

    // piggybank's deposit balance.

    uint128 public balance;

    // Constructor saves the address of the contract owner in a state variable and

    // initializes the limit and the balance.

    constructor(address own, address own2, uint lim) public {

        // check that contract's public key is set

        require(tvm.pubkey() != 0, 101);

        // Check that message has signature (msg.pubkey() is not zero) and message is signed with the owner's private key

        require(msg.pubkey() == tvm.pubkey(), 102);

        tvm.accept();

        owner = own;

        owner2 = own2;

        limit = lim;

        balance = 0;

    }

    // Modifier that allows public function to be called only when the limit is reached.

    modifier checkBalance() {

        require(balance >= limit, 103);

        \_;

    }

    // Modifier that allows public function to be called only from the owners address.

    modifier checkSenderIsOwner {

        if(msg.sender != owner && msg.sender != owner2) {revert();}

        \_;

    }

    // Function that can be called by anyone.

    function deposit() public {

        balance += uint128(msg.value);

    }

    // Function that can be called only by the owner after reaching the limit.

    function withdraw() public checkSenderIsOwner checkBalance {

        tvm.accept();

        msg.sender.transfer(balance);

        balance = 0;

    }

}

PiggyBank Owner (PB\_Owner.sol)

pragma ton-solidity >= 0.35.0;

pragma AbiHeader expire;

import "PB.sol";

// This contract describes the owner of PiggyBank who can add to deposit and withdraw deposit.

contract Owner {

    constructor() public {

        // check that contract's public key is set

        require(tvm.pubkey() != 0, 101);

        // Check that message has signature (msg.pubkey() is not zero) and message is signed with the owner's private key

        require(msg.pubkey() == tvm.pubkey(), 102);

        tvm.accept();

    }

    modifier onlyOwner {

        // Check that message was signed with contracts key.

        require(msg.pubkey() == tvm.pubkey(), 102);

        \_;

    }

    // Function to deposit money to piggy bank.

    function addToDeposit(PiggyBank bankAddress, uint128 amount) public pure onlyOwner {

        tvm.accept();

        bankAddress.deposit{value: amount}();

    }

    // Function to withdraw money from piggy bank.

    function withdrawDeposit(PiggyBank bankAddress) public pure onlyOwner {

        tvm.accept();

        bankAddress.withdraw();

    }

}

PiggyBank Stranget (PB\_Stranger.sol)

pragma ton-solidity >= 0.35.0;

pragma AbiHeader expire;

import "PB.sol";

// This contract describes the Stranger who can add to deposit of PiggyBank but can't withdraw deposit.

contract Stranger {

    constructor() public {

        // check that contract's public key is set

        require(tvm.pubkey() != 0, 101);

        // Check that message has signature (msg.pubkey() is not zero) and message is signed with the owner's private key

        require(msg.pubkey() == tvm.pubkey(), 102);

        tvm.accept();

    }

    modifier onlyOwner {

        // Check that message was signed with contracts key.

        require(msg.pubkey() == tvm.pubkey(), 102);

        \_;

    }

    // Function to deposit money to piggy bank.

    function addToDeposit(PiggyBank bankAddress, uint128 amount) public pure onlyOwner {

        tvm.accept();

        bankAddress.deposit{value: amount}();

    }

    // Function to withdraw money from piggy bank.

    function withdrawDeposit(PiggyBank bankAddress) public pure onlyOwner {

        tvm.accept();

        bankAddress.withdraw();

    }

}

Remix ide test data

Owner1 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4

Owner2 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2

Stranger 0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c

Bankacc 0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db

Tonos-cli

1. Tondev tonos-cli intstall
2. tonos-cli genphrase

tonos-cli getkeypair file.json “seed”

owner1: "young noise uniform hand when glimpse naive alarm notable acoustic another capable"

Raw address: -1:7e92dcc6d0c8d159605c4c4260f1c0e708e277d8ad46e300ced6fbd85f79fed9

testnet:

Non-bounceable address (for init): 0f9-ktzG0MjRWWBcTEJg8cDnCOJ32K1G4wDO1vvYX3n-2QLW

Bounceable address (for later access): kf9-ktzG0MjRWWBcTEJg8cDnCOJ32K1G4wDO1vvYX3n-2V8T

mainnet:

Non-bounceable address (for init): Uf9-ktzG0MjRWWBcTEJg8cDnCOJ32K1G4wDO1vvYX3n-2blc

Bounceable address (for later access): Ef9-ktzG0MjRWWBcTEJg8cDnCOJ32K1G4wDO1vvYX3n-2eSZ

Succeeded

owner2: "blind midnight coin clerk argue swim erosion ill assist skirt convince metal"

Config: default

Input arguments:

tvc: PB\_Owner.tvc

wc: -1

keys: key\_owner2.json

init\_data: None

is\_update\_tvc: None

Raw address: -1:b1394e14bde71975782d4e0402257520a954b4fa84bbb88c159ac00b4ff176a8

testnet:

Non-bounceable address (for init): 0f-xOU4UvecZdXgtTgQCJXUgqVS0-oS7uIwVmsALT\_F2qO1A

Bounceable address (for later access): kf-xOU4UvecZdXgtTgQCJXUgqVS0-oS7uIwVmsALT\_F2qLCF

mainnet:

Non-bounceable address (for init): Uf-xOU4UvecZdXgtTgQCJXUgqVS0-oS7uIwVmsALT\_F2qFbK

Bounceable address (for later access): Ef-xOU4UvecZdXgtTgQCJXUgqVS0-oS7uIwVmsALT\_F2qAsP

Succeeded

stranger: "tail wait exact donor october able risk suffer project galaxy health speed"

Config: default

Input arguments:

tvc: PB\_Stranger.tvc

wc: -1

keys: key\_stranger.json

init\_data: None

is\_update\_tvc: None

Raw address: -1:a720fcb246f8b1f213595941367f021d20757f04be8fe6226909bf68b9370b36

testnet:

Non-bounceable address (for init): 0f-nIPyyRvix8hNZWUE2fwIdIHV\_BL6P5iJpCb9ouTcLNnw\_

Bounceable address (for later access): kf-nIPyyRvix8hNZWUE2fwIdIHV\_BL6P5iJpCb9ouTcLNiH6

mainnet:

Non-bounceable address (for init): Uf-nIPyyRvix8hNZWUE2fwIdIHV\_BL6P5iJpCb9ouTcLNse1

Bounceable address (for later access): Ef-nIPyyRvix8hNZWUE2fwIdIHV\_BL6P5iJpCb9ouTcLNppw

Succeeded

Bank :

1. Деплоим все контракты по порядку

tonos-cli deploy PB\_Owner.tvc "{}" --abi PB\_Owner.abi.json --sign key\_owner1.json --wc -1

tonos-cli deploy PB\_Owner.tvc "{}" --abi PB\_Owner.abi.json --sign key\_owner2.json --wc -1

tonos-cli deploy PB\_Stranger.tvc "{}" --abi PB\_Stranger.abi.json --sign key\_stranger.json --wc -1

и сам банк

tonos-cli deploy PB.tvc params.json --abi PB.abi.json --sign key\_bank.json --wc -1