COMP6451 T1 2022

Assignment 2

Haozhe Chen

z5142012

**Part 1: Contract structure overview**

This assignment can be divided into two parts of contract:

1. Share contract use ERC20 standard interface
2. 图示

   描述已自动生成Fundraising contract which is used to collected ether and grant shares (tokens)

**[Round 0] Initial**

In this round, beneficiary (aka NeverPay company), will deploy the fundraising contract (NeverPayFundraising.sol) into the blockchain. In the fundraising contract’s constructor, the share contract (ERC20NeverPayToken.sol) will be initiated.

Before this round end, the fundraising contract should keep 10000 tokens.

**[Round 1] Bid**

In this round, investors are able to call bid function to make a blinded bid. The parameter should be a hash value = keccak256(share, value, nonce).

**[Round 2] Reveal**

In this round, investor will call reveal function to open their bid in round1 and pay to the contract. The parameters are shares, value and nonce.

If hash value of these three parameters is equal to bid record in round1 and amount paid is larger than price \* shares, this reveal is successful. Otherwise, this reveal is failed.

**[After Round 2] Issue**

This round is used to let every investor to get their shares and refunds. Beneficiary can also call this function to get ETH collected during the last round (only successful bid).

The issue function will firstly sort every bid (reveal successfully in round2), then the successful bid’s owner will get the shares, failed bid’s owner will get the refunds.

**Part 2a: Data model (Token contract)**

|  |  |  |
| --- | --- | --- |
| **Token contract: ERC20NeverPayToken.sol** | | |
| **Property** | **Type** | **Description** |
| name | uint256 | Token’s name |
| decimals | uint8 | How many decimals to show |
| symbol | string | Token’s identifier |
| balances | mapping  address => uint256 | user\_adress => balances  User’s balances |
| totalSupply | uint | Token’s total supply |

The token contract is followed by ERC20 token interface. Every users’ balance is stored by a mapping structure explicitly.

|  |  |  |
| --- | --- | --- |
| **Function: transfer** | | |
| **Property** | **Type** | **Description** |
| \_to | address | Receiver’s address |
| \_value | uint256 | Transfer amount |
| **Returntype** | **Description** | |
| bool | Transfer successfully or failed | |

This function is used to let sender to transfer money to certain address.

|  |  |  |
| --- | --- | --- |
| **Function: transferFrom** | | |
| **Property** | **Type** | **Description** |
| \_from | address | Sender’s address |
| \_to | address | Receiver’s address |
| \_value | uint256 | Transfer amount |
| **Returntype** | **Description** | |
| bool | Transfer successfully or failed | |

This function is also used to transfer money. However, this function should be called by receiver to initiate a transaction from a certain address. Which also need sender’s to approve this transaction with approve function.

|  |  |  |
| --- | --- | --- |
| **Function: approve** | | |
| **Property** | **Type** | **Description** |
| \_spender | address | Receiver’s address |
| \_value | uint256 | Approve transfer amount |
| **Returntype** | **Description** | |
| bool | Approve successfully or failed | |

This function should be called by sender to approve amount of money transferred by certain address. In case of receiver ‘stole’ money from sender (can initiate transfer with any amount from any address).

|  |  |  |
| --- | --- | --- |
| **Function: balanceOf** | | |
| **Property** | **Type** | **Description** |
| \_owner | address | Owner’s address |
| **Returntype** | **Description** | |
| uint256 | Owner’s balance | |

This function is used to get the balance of certain address.

|  |  |  |
| --- | --- | --- |
| **Function: allowance** | | |
| **Property** | **Type** | **Description** |
| \_owner | address | Sender’s address |
| \_spender | address | Receiver’s address |
| **Returntype** | **Description** | |
| uint256 | Value approved by sender to transfer. | |

This function is used to check the amount of value approved by sender to transferred to receiver.

**Part 2b: Data model (Fundraising contract)**

|  |  |  |
| --- | --- | --- |
| **Token contract: ERC20NeverPayToken.sol** | | |
| **Property** | **Type** | **Description** |
| name | uint256 | Token’s name |
| decimals | uint8 | How many decimals to show |
| symbol | string | Token’s identifier |
| balances | mapping  address => uint256 | user\_adress => balances  User’s balances |
| totalSupply | uint | Token’s total supply |

The token contract is followed by ERC20 token interface. Every users’ balance is stored by a mapping structure explicitly.