Лабораторна робота №4

Налаштування середовища та базові смарт-контракти Ethereum

Шостак Юліан 41ІСТ

1. У середовищі Remix IDE створено новий файл SimpleBank.sol.

2. Реалізовано контракт SimpleBank, який включає такі компоненти:

- змінні owner, balances, registered, totalBankBalance, userAddresses;

- модифікатори onlyOwner() та isRegistered();

- функції register(), deposit(), withdraw(), transfer(), getMyBalance(), getTotalBalance(), getAllUsersBalance().

Лістинг  
// SPDX-License-Identifier: MIT  
pragma solidity ^0.8.0;  
  
contract SimpleBank {  
 address public owner;  
 mapping(address => uint256) public balances;  
 mapping(address => bool) public registered;  
 uint256 public totalBankBalance;  
 address[] private userAddresses;  
  
 struct UserBalance {  
 address user;  
 uint256 balance;  
 }  
  
 constructor() {  
 owner = msg.sender;  
 }  
  
 modifier onlyOwner() {  
 require(msg.sender == owner, "Only owner can perform this action");  
 \_;  
 }  
  
 modifier isRegistered() {  
 require(registered[msg.sender], "User is not registered");  
 \_;  
 }  
  
 function register() public {  
 require(!registered[msg.sender], "Already registered");  
 registered[msg.sender] = true;  
 userAddresses.push(msg.sender);  
 }  
  
 function deposit() public payable isRegistered {  
 require(msg.value > 0, "Deposit amount must be greater than zero");  
 balances[msg.sender] += msg.value;  
 totalBankBalance += msg.value;  
 }  
  
 function getMyBalance() public view isRegistered returns (uint256) {  
 return balances[msg.sender];  
 }  
  
 function withdraw(uint256 \_amount) public isRegistered {  
 require(balances[msg.sender] >= \_amount, "Insufficient balance");  
 balances[msg.sender] -= \_amount;  
 totalBankBalance -= \_amount;  
 payable(msg.sender).transfer(\_amount);  
 }  
  
 function transfer(address \_to, uint256 \_amount) public isRegistered {  
 require(registered[\_to], "Recipient not registered");  
 require(balances[msg.sender] >= \_amount, "Insufficient balance");  
 balances[msg.sender] -= \_amount;  
 balances[\_to] += \_amount;  
 }  
  
 function getTotalBalance() public view onlyOwner returns (uint256) {  
 return totalBankBalance;  
 }  
  
 function getAllUsersBalance() public view onlyOwner returns (UserBalance[] memory) {  
 UserBalance[] memory users = new UserBalance[](userAddresses.length);  
 for (uint256 i = 0; i < userAddresses.length; i++) {  
 users[i] = UserBalance(userAddresses[i], balances[userAddresses[i]]);  
 }  
 return users;  
 }  
}

Результати виконання

|  |  |  |  |
| --- | --- | --- | --- |
| № | Функція | Дія | Результат |
| 1 | register() | Реєстрація користувача | Успішно |
| 2 | deposit() | Внесення 1 ETH | Баланс оновлено |
| 3 | withdraw() | Зняття 0.5 ETH | Баланс зменшився |
| 4 | transfer() | Переказ 0.1 ETH іншому користувачу | Успішно |
| 5 | getTotalBalance() | Перевірка загального балансу | Відображено суму банку |
| 6 | getAllUsersBalance() | Перегляд усіх користувачів | Відображено список |

