BCT-Practical - 3

CODE

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract Bank {
    mapping(address => uint256) private userAccount;
    mapping(address => bool) private userExists;

    function createAccount() public payable returns (string memory) {
        require(!userExists[msg.sender], "Account already exists!");
        userAccount[msg.sender] = msg.value;
        userExists[msg.sender] = true;
        return "Account created successfully!";
    }

    function deposit() public payable returns (string memory) {
        require(userExists[msg.sender], "Account not found!");
        require(msg.value > 0, "Deposit amount must be greater than 0");
        userAccount[msg.sender] += msg.value;
        return "Amount deposited successfully!";
    }

    function withdraw(uint256 amount) public returns (string memory) {
        require(userExists[msg.sender], "Account not found!");
        require(amount > 0, "Withdraw amount must be greater than 0");
        require(userAccount[msg.sender] >= amount, "Insufficient balance");

        userAccount[msg.sender] -= amount;
        payable(msg.sender).transfer(amount);
        return "Amount withdrawn successfully!";
    }

    function getBalance() public view returns (uint256) {
        require(userExists[msg.sender], "Account not found!");
        return userAccount[msg.sender];
    }

    function hasAccount() public view returns (bool) {
        return userExists[msg.sender];
    }
}
```