// SPDX-License-Identifier: MIT

pragma solidity >=0.7.0 <0.9.0;

contract DBank

{

    address Owner;

    // We are creating the Mapping for the Adding & Transfer Amount in Account

    mapping(address=>uint)Balance;

    // constructor for the address of the Owner

    constructor()

    {

        Owner = msg.sender;

    }

    // function for adding the Ethereum in Account

    function addBalance(uint amount) public returns(uint)

    {

        // first we have to check the is it Owners Account or Not

        require(msg.sender == Owner, "Yes it is Owner Account !!");

        Balance[msg.sender] = Balance[msg.sender] + amount;

        return Balance[msg.sender];

    }

    // function for Get the Balance from an Account

    function getBalance() public view returns(uint)

    {

        return Balance[msg.sender];

    }

    // to transfer the Amount from Owner to Recipient

    function Transfer(address recipient, uint amount) public

    {

        // check the Self account is or not

        require(msg.sender != recipient, "Can't Transfer !! Self Account.");

        // check the owner has balance is available or not

        require(Balance[msg.sender] >= amount, "No, We Can't Transfer. Insufficient Balance !!");

        \_transfer(msg.sender, recipient, amount);

    }

    function \_transfer(address From, address To, uint Amount) private

    {

        Balance[From] = Balance[From] - Amount;

        Balance[To] = Balance[To] + Amount;

    }

}