// SPDX-License-Identifier: GPL-3.0

pragma solidity >=0.7.0 <0.9.0;

import "./sellFlat2.sol";

contract arenda is Flats{

    struct rent{

        address owner;

        uint idFlat;

        uint price;

        uint term;

        address user;

        bool statusRent; // снимает ли кто-то квартиру

        bool statuseAnswer; // закрыта ли эта заявка(отказ предоставлять квартиру)

    }

    rent[] rents;

    //Создание объявления об аренде

    function createRent(uint idFlat, uint price, uint term)public{

        require(bazaOwners[msg.sender][idFlat].statuseOwner == true, "You not owner");

        require(bazaOwners[msg.sender][idFlat].statusRent == false, "you have already rent");

        rents.push(rent(msg.sender, idFlat, price, term, address(0), false,false));

    }

    //Отклик на предложение

    function requestRent(uint idRent)public payable{

        require(msg.sender != rents[idRent].owner, "you cannot rent the flat to yourself");

        require(rents[idRent].statusRent == false, "apartment already rented");

        require(rents[idRent].statuseAnswer == false, "Application closed");

        require(rents[idRent].price == msg.value, "you have low money");

        rents[idRent].user = msg.sender;

    }

    //Соглашение хозяина

    function answerUser(uint idRent, bool answer)public{

        require(rents[idRent].owner == msg.sender,"You not owner");

        require(rents[idRent].user != address(0), "You has not buyer");

        require(rents[idRent].statusRent == false, "You already agreed");

        if(answer){

            rents[idRent].statusRent = true;

            payable(rents[idRent].owner).transfer(rents[idRent].price);

        }

        else{

            rents[idRent].user = address(0);

            payable(rents[idRent].user).transfer(rents[idRent].price);

        }

    }

    //Отказ от аренды

    function dpropRent (uint idRent)public payable{

        require(rents[idRent].owner == msg.sender,"You not owner");

        require(rents[idRent].statuseAnswer == false, "Application closed");

        require(rents[idRent].term > nowMoment, "little time has passed");

        if(rents[idRent].user != address(0)){

            payable(rents[idRent].user).transfer(rents[idRent].price);

        }

        else{

         rents[idRent].statuseAnswer = true;

        }

    }

    //Отмена аренды квартиры

    function cancelRent(uint idRent) public payable{

        require(rents[idRent].user == msg.sender,"You not buyer");

        require(rents[idRent].statusRent == false, "Application closed");

        require(rents[idRent].statuseAnswer == true, "Status is not true");

        payable(msg.sender).transfer(rents[idRent].price);

    }

}