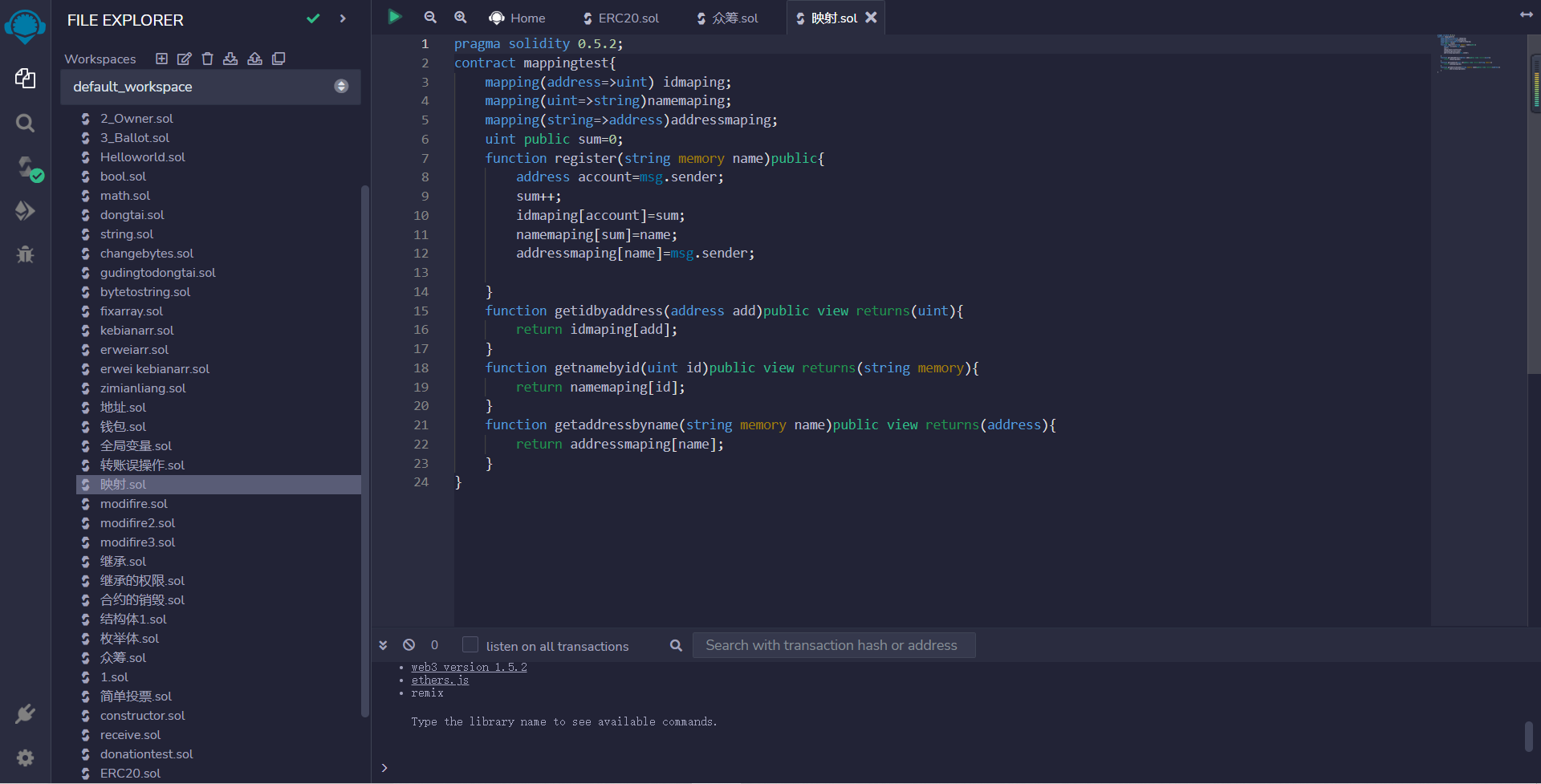
# 关于solidity的学习：

1. 映射：



映射用mapping表示，具体使用格式：mapping（string/uint/address/../=>string/uint/address/mapping/../)命名；

简单来说就是一个一一对应的关系，对solidity数据存储有非常大的作用

1. ERC20:

具体代码：

// SPDX-License-Identifier: MIT

pragma solidity ^ 0.8.17;

interface IERC20{

    event Transfer(address indexed from,address indexed to,uint256 value);

    event Approval(address indexed owner,address indexed spender,uint256 value);

    function totalSupply()external view returns(uint256);

    function balanceOf(address account)external view returns(uint256);

    function allowance(address owner,address spender)external view returns(uint256);

    function approve(address spender,uint256 amount)external  returns(bool);

    function transfer(address to,uint256 amount)external returns(bool);

    function transferFrom(address from,address to,uint256 account)external returns(bool);

}

contract ERC20 is IERC20{

    uint256 \_totaSupply;

    mapping(address=>uint256)\_balance;

    string \_name;

    string \_symbol;

    address \_owner;

    modifier onlyOwner(){

        require(\_owner==msg.sender,"only owner can access this function");

        \_;

    }

    constructor(string memory name\_,string memory symbol\_){

        \_name=name\_;

        \_symbol=symbol\_;

        \_owner=msg.sender;

        \_balance[msg.sender]=10000;

        \_totaSupply=10000;

    }

    function mint(address account,uint256 amount)public onlyOwner{

        require(account!=address(0),"mint to address 0");

        \_totaSupply+=amount;

        \_balance[account]+=amount;

        emit Transfer(address(0),account,amount);

    }

    function burn(address account,uint256 amount)public onlyOwner{

        require(account!=address(0),"mint to address 0");

        \_totaSupply-=amount;

        uint256 accountbalance=\_balance[account];

        require(accountbalance>=amount,"no more  token to burn");

        \_balance[account]=accountbalance-amount;

        emit Transfer(account,address(0),amount);

    }

    function name()public view returns(string memory){

        return \_name;

    }

    function symbol()public view returns(string memory){

        return \_symbol;

    }

    function decimals()public pure returns(uint256){

        return 18;

    }

    function totalSupply()public view returns(uint256){

        return \_totaSupply;

    }

    function balanceOf(address account)public view returns(uint256){

        return \_balance[account];

    }

    function transfer(address to,uint256 amount)public returns(bool){

        uint256 myBalance=\_balance[msg.sender];

        require(myBalance>=amount,"no money to transfer");

        require(to!=address(0),"transfer to address 0");

        \_balance[msg.sender]=myBalance-amount;

        \_balance[to]=\_balance[to]+amount;

        emit Transfer(msg.sender,to,amount);

        return true;

    }

    mapping(address=>mapping(address=>uint256))\_allowance;

    function allowance(address owner,address spender)public view returns(uint256){

        return \_allowance[owner][spender];

    }

    function approve(address spender,uint256 amount)public  returns(bool){

        \_allowance[msg.sender][spender]=amount;

        emit Approval(msg.sender,spender,amount);

        return true;

    }

    function transferFrom(address from,address to,uint256 account)public returns(bool){

          uint256 myallowance=\_allowance[from][msg.sender];

          require(myallowance>=account,"error:myallowance<account");

          \_allowance[from][msg.sender]=myallowance-account;

          emit Approval(from,msg.sender,myallowance-account);

          uint256 frombalance=\_balance[from];

          require(frombalance>=account,"error:frombalance<account");

          \_balance[from]=frombalance-account;

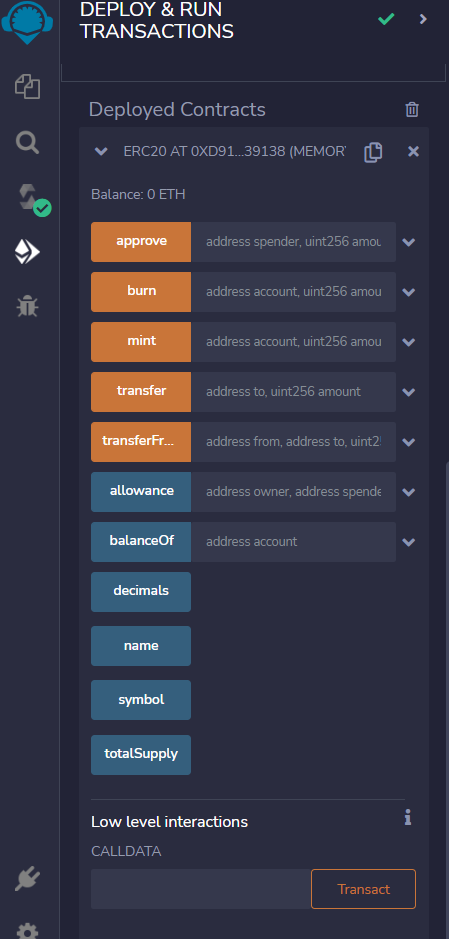
          \_balance[to]+=account;

          emit Transfer(from,to,account);

          return true;

      }

}

部署成功时： 

简单来说，erc20就是一种代币标准，该代码实现了铸造代币，销毁代币，转账，委托给别人等一系类方法，其中值得注意的是，当我们需要铸造一种代币的时候，我们需要设置铸造和销毁代币的权力必须是我们自己，erc20实现这套接口方法时，所有钱包就可以直接兼容，比如jaxx,mew,intoken等。