

# HW7

## # lesson7

### ## 作业

- 搭建Clarity环境, 跑通简单的智能合约并将过程截图提交到screenshot文件夹中
- 分析token.clar代码, 将带有注释的token.clar代码提交到screenshot文件夹中
- 回答思考题, 将思考题答案提交到screenshot文件夹中
- 文件名格式为 "学号-BNS域名"。例如 "000-gavin.id"

### ## 思考题目

- 题目一: 根据今天对于智能合约的讲解, 你认为智能合约可以解决哪些现有互联网无法解决的问题? 又会带来哪些问题?
- 题目二: 前六节课的主要内容均为Blockstack V1的架构, 本节课为Blockstack V2架构中的一个核心内容, 请问你认为V1与V2将如何结合在一起呢?
- 题目三: 如果将本节课的内容应用去中心化留言板中, 你认为整个留言板的流程图会有什么变化? 会添加哪些功能? (建议画出流程图讲解)

# 1. 搭建Clarity环境, 跑通简单的智能合约并将过程截图提交到screenshot文件夹中

```
Transaction executed and committed. Returned: 0x68656c6c6f576f726c64
root@c1110048f31a:/src/blockstack-core/sample-programs# vim store.clar
root@c1110048f31a:/src/blockstack-core/sample-programs# clarity-cli check store.clar /data/db
Error (line 4, column 1): defining 'value' conflicts with previous value.
(define-map store ((key (buff 32))) ((value (buff 32))))

roroot@c1110048f31a:/src/blockstack-core/sample-programs# clarity-cli check storclar /data/db
Checks passed.
root@c1110048f31a:/src/blockstack-core/sample-programs# clarity-cli launch $DEMO_ADDRESS.store store.clar /data/db
Contract initialized!
root@c1110048f31a:/src/blockstack-core/sample-programs# clarity-cli execute /data/db $DEMO_ADDRESS.store get-value $DEMO_ADDRESS
Transaction executed and committed. Returned: u0
root@c1110048f31a:/src/blockstack-core/sample-programs# clarity-cli execute /data/db $DEMO_ADDRESS.store set-value $DEMO_ADDRESS u2
Transaction execution error:
Unchecked(UndefinedFunction("set-value"))
root@c1110048f31a:/src/blockstack-core/sample-programs# clarity-cli execute /data/db $DEMO_ADDRESS.store set $DEMO_ADDRESS u2
Transaction executed and committed. Returned: true
root@c1110048f31a:/src/blockstack-core/sample-programs# clarity-cli execute /data/db $DEMO_ADDRESS.store get-value $DEMO_ADDRESS
Transaction executed and committed. Returned: u1
root@c1110048f31a:/src/blockstack-core/sample-programs# clarity-cli execute /data/db $DEMO_ADDRESS.store get-value $DEMO_ADDRESS
Transaction executed and committed. Returned: u1
root@c1110048f31a:/src/blockstack-core/sample-programs# vim kv-store.clar
root@c1110048f31a:/src/blockstack-core/sample-programs# clarity-cli check kv-store.clar /data/db
Checks passed.
root@c1110048f31a:/src/blockstack-core/sample-programs#
```

# DB中可以看到已经发布的智能合约资讯

New Database

Open Database

Write Changes

Revert Changes

Open Project

Save Project

Attach Database

Close Database

Database Structure

Browse Data

Edit Pragmas

Execute SQL

Table: 

data\_table

New Record

Delete Record

	key	value
	Filter	Filter
35	a0f30135120c3...	17
36	9cce01e576e6a...	{'block_height':18,'block_time':1586854132,'block_header_hash':{...
37	c5ca3bec28ca7...	18
38	f00708bc8c904...	{'block_height':19,'block_time':1586854732,'block_header_hash':{...
39	8010221d074b...	19
40	51106fb0d71ad...	{'block_height':20,'block_time':1586855332,'block_header_hash':{...
41	4ef4a540b05d6...	20
42	a2c14c1fc3af43...	{'contract_identifier':{'issuer':[22,1239,254,102,161,75,105,58,188...
43	bbdf56f5123c8...	{'contract_context':{'contract_identifier':{'issuer':[22,1239,254,102,...
44	f68a47dd584a1...	{'contract_identifier':{'issuer':[22,1239,254,102,161,75,105,58,188...
45	a4f49f87e4b70...	{'value_type':'UIntType'}
46	7734512bc903...	{'type':'u128','value':'0'}
47	4ffafd5671f458...	{'contract_context':{'contract_identifier':{'issuer':[22,1239,254,102,...
48	af7a82156b6ce...	{'type':'u128','value':'1'}

35 - 48 of 48

Go to:

1

Edit Database Cell

Mode: 

JSON

Import

Export

Set as NULL

1

168,147,113,184,225,63,115,52,125]],"name":"store"},"private\_function\_...

Type of data currently in cell: Valid JSON  
545 char(s)

Apply

Remote

Identity

Name	Commit	Last modified	Size
------	--------	---------------	------

SQL Log

Plot

DB Schema

Remote

## 2- 分析token.clar代码, 将带有注释的token.clar代码提交到screenshot文件夹中

```
1 复杂合约参考
2 token.clar
3 ; tokens是个mapping, key: account, value: balance
4 (define-map tokens ((account principal)) ((balance uint)))
5
6 ; 定义private函数get-balance, 根据account取得此account之balance
7 (define-private (get-balance (account principal))
8   (default-to u0 (get balance (map-get? tokens (tuple (account account))))))
9
10 ; 定义private函数token-credit!, amount是>0时 principal的balance增加amount值,
    如果 amount <=0 则报错, 用在函数token-transfer与mint!中。
11 (define-private (token-credit! (account principal) (amount uint))
12   (if (<= amount u0)
13     (err "must move positive balance")
14     (let ((current-amount (get-balance account)))
15       (begin
16         (map-set tokens (tuple (account account))
17           (tuple (balance (+ amount current-amount))))
18         (ok amount)))))
```

```
19 |  
20 ; 定义public函数token-transfer, tx-sender之balance减少, principal之balance增加  
21 (define-public (token-transfer (to principal) (amount uint))  
22   (let ((balance (get-balance tx-sender)))  
23     (if (or (> amount balance) (<= amount u0))  
24       (err "must transfer positive balance and possess funds")  
25       (begin  
26         (map-set tokens (tuple (account tx-sender))  
27           (tuple (balance (- balance amount))))  
28         (token-credit! to amount))))))  
29  
30 ; 定义public函数mint! 让tx-sender的balance增加amount值  
31 (define-public (mint! (amount uint))  
32   (let ((balance (get-balance tx-sender)))  
33     (token-credit! tx-sender amount)))
```

### 3. 思考题目

题目一：根据今天对于智能合约的讲解，你认为智能合约可以解决哪些现有互联网无法解决的问题？又会带来哪些问题？

解决的问题：将中心化的database存储，改为去中心化的存储，此外，可以完全做到流程共有化，自动化，不可篡改，无需中间人服务

带来哪些问题：由于需要给予参与共识机制的所有节点一定的奖励，所以每一笔去中心化的存储都会有一定的成本，此外由于处理共识机制(挖矿)需要时间，TPS预计也会比中心化的存储低一些

题目二:前六节课的主要内容均为Blockstack V1的架构,本节课为Blockstack V2架构中的一个核心内容,请问你认为V1与V2将如何结合在一起呢?

V1主要强调的是DID与user的data存储与隐私与user的定向授权,群组授权, V2主要强调的是smart contract, 可能可以结合的点会是, 将定向授权与智能合约结合在一起, 例如:当A付给B某token后, B才将授权码交给A, 可以在生成邀请码的时候通过智能合约setvalue, 然后B想getvalue需要转移token, 目前还需要官方提供javascript端如何与lisp端结合, 最后形成一个完整的dapp。



题目三:如果将本节课的内容应用在去中心化留言板中,你认为整个留言板的流程图会有什么变化?会添加哪些功能?(建议画出流程图讲解)

可以添加若是某user付了token之后,才可以收到某留言群组邀请码的功能。

