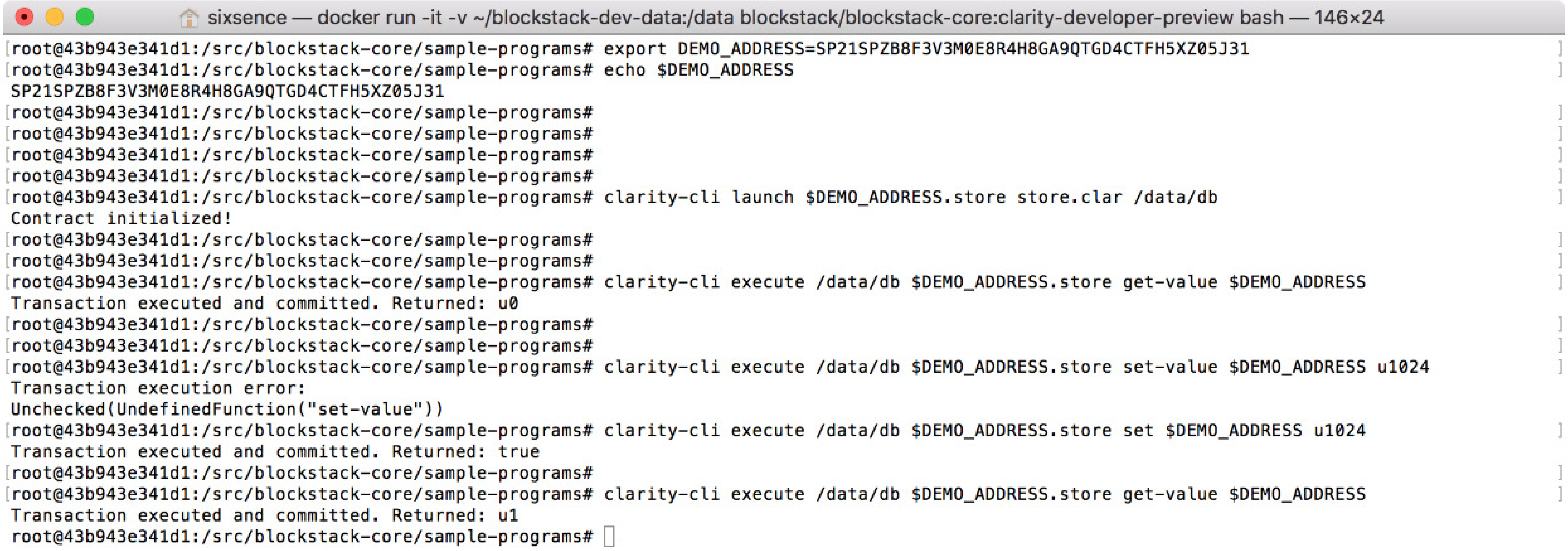
store.clar 截图


sixsence — docker run -it -v ~/blockstack-dev-data:/data blockstack/blockstack-core:clarity-developer-preview bash — 146×24

```
[root@43b943e341d1:/src/blockstack-core/sample-programs# export DEMO_ADDRESS=SP21SPZB8F3V3M0E8R4H8GA9QTGD4CTFH5XZ05J31
[root@43b943e341d1:/src/blockstack-core/sample-programs# echo $DEMO_ADDRESS
 SP21SPZB8F3V3M0E8R4H8GA9QTGD4CTFH5XZ05J31
[root@43b943e341d1:/src/blockstack-core/sample-programs#
[root@43b943e341d1:/src/blockstack-core/sample-programs#
[root@43b943e341d1:/src/blockstack-core/sample-programs#
[root@43b943e341d1:/src/blockstack-core/sample-programs#
[root@43b943e341d1:/src/blockstack-core/sample-programs# clarity-cli launch $DEMO_ADDRESS.store store.clar /data/db
 Contract initialized!
[root@43b943e341d1:/src/blockstack-core/sample-programs#
[root@43b943e341d1:/src/blockstack-core/sample-programs#
[root@43b943e341d1:/src/blockstack-core/sample-programs# clarity-cli execute /data/db $DEMO_ADDRESS.store get-value $DEMO_ADDRESS
 Transaction executed and committed. Returned: u0
[root@43b943e341d1:/src/blockstack-core/sample-programs#
[root@43b943e341d1:/src/blockstack-core/sample-programs#
[root@43b943e341d1:/src/blockstack-core/sample-programs# clarity-cli execute /data/db $DEMO_ADDRESS.store set-value $DEMO_ADDRESS u1024
 Transaction execution error:
 Unchecked(UndefinedFunction("set-value"))
[root@43b943e341d1:/src/blockstack-core/sample-programs# clarity-cli execute /data/db $DEMO_ADDRESS.store set $DEMO_ADDRESS u1024
 Transaction executed and committed. Returned: true
[root@43b943e341d1:/src/blockstack-core/sample-programs#
[root@43b943e341d1:/src/blockstack-core/sample-programs# clarity-cli execute /data/db $DEMO_ADDRESS.store get-value $DEMO_ADDRESS
 Transaction executed and committed. Returned: u1
 root@43b943e341d1:/src/blockstack-core/sample-programs# 
```

token.clar代码分析

```
(define-map tokens ((account principal)) ((balance uint))) ; 定义tokens的一个map,用于记录账户与余额
(define-private (get-balance (account principal)) ; 定义查询余额的函数,当查不到账号数据时,返回余额为0
  (default-to u0 (get balance (map-get? tokens (tuple (account account))))))

(define-private (token-credit! (account principal) (amount uint)) ; 定义发币函数
  (if (<= amount u0)
      (err "must move positive balance") ; 当发币金额小于0时, 提示错误信息
      (let ((current-amount (get-balance account))) ; 把当前账户的余额赋值给current-amount, 同时更新当前账户的余额加上要发币的金额
        (begin
          (map-set tokens (tuple (account account))
                      (tuple (balance (+ amount current-amount))))
          (ok amount)))))

(define-public (token-transfer (to principal) (amount uint)) ; 定义转移token, 为to参数对应的账户转移amount额度
  (let ((balance (get-balance tx-sender))) ; 把当前账户的余额赋值给balance参数
    (if (or (> amount balance) (<= amount u0))
        (err "must transfer positive balance and possess funds") ; 当 amount > balance 或者 amount <= 0时, 提示错误信息
        (begin
          (map-set tokens (tuple (account tx-sender))
                      (tuple (balance (- balance amount)))) ; 更新当前账户余额减少amount额度
          (token-credit! to amount))))) ;同时给to参数对应的账户发币amount额度

(define-public (mint! (amount uint)) ; 定义铸币
   (let ((balance (get-balance tx-sender))) ; 把当前账户的余额赋值给balance参数
     (token-credit! tx-sender amount))) 为当前账户发币amount额度

(token-credit! 'SZ2J6ZY48GV1EZ5V2V5RB9MP66SW86PYKKQ9H6DPR u10000)
(token-credit! 'SM2J6ZY48GV1EZ5V2V5RB9MP66SW86PYKKQVX8X0G u300)
```

思考题

- 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？

  > 现在可以想到的就是审计行业要求核心数据进行智能合约,规避数据作假.
  > 以及核心资产数字版权
  > 至于会带来哪些新的问题,目前还没有什么想法

- 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？

  > 我的理解是,会提供一个地域实现逻辑协议,让以此为基础的应用之间创建智能合约,确保是在一个协议下.
  > 而用户在授权过程中,也相当于是在与应用创建授权的智能合约,以保证用户核心数据的保密性.

- 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）

  > 因为目前对于智能合约的理解还比较浅显,所以暂时还没有什么想法.