**SBTC智能合约操作使用手册**

一. 智能合约开发环境  
1. 智能合约使用Solidity语言编程.  
2. 智能合约集成开发环境，基于浏览器的IDE，在线实时编译.  
<https://remix.ethereum.org/#optimize=false&version=soljson-v0.4.19+commit.c4cbbb05.js>  
  
二.部署智能合约的流程(在regtest网络环境测试)  
1. 启动一个sbtc节点。  
2. 在IDE，使用solidity 编辑合约源码,编译获得二进制代码。  
3. 将编译好的合约部署到网络,获得合约地址。  
4. 使用sbtc-cli控制台调用已部署的合约。  
  
三.操作合约相关的rpc指令详解  
1. createcontract   
功能: 注册合约  
params:  
(1).bytecode (string,required) contract bytecode  
(2).gasLimit (numeric or string, optional) gasLimit  
(3).gasPrice (numeric or string, optional) gasPrice SBTC price per gas unit  
(4).senderaddress (string, optional) The sbtc address that will be used to create the contract.  
(5).broadcast (bool, optional, default=true) Whether to broadcast the transaction or not  
(6).changeToSender (bool, optional, default=true) Return the change to the sender  
  
Results:  
(1). txid the transaction id  
(2).sender address of the sender  
(3).hash160 ripemd-160 hash of the sender  
(4).address expected contract address  
  
eg:  
./sbtc-cli -datadir=./ createcontract   
6060604052341561000f57600080fd5b60b18061001d6000396000f300606060405260043610603f576000357c0100000000000000000000000000000000000000000000000000000000900463ffffffff168063c6888fa1146044575b600080fd5b3415604e57600080fd5b606260048080359060200190919050506078565b6040518082815260200191505060405180910390f35b60006007820290509190505600a165627a7a723058204c97a8211670e0c854e1e5461b08b5eab69804024879e2f17f3ad9f4237c560f0029 2500000 0.00000040 mjMCzCAfLmU6mxpe2MWV5PSVhGWx5dPQhN true true   
  
2.sendtocontract  
功能：使用合约  
params:  
(1).contractaddress (string, required) The contract address   
(2).datahex (string, required) data to send.  
(3).amount (numeric or string, optional)  
(4).gasLimit (numeric or string, optional) gasLimit  
(5).gasPrice (numeric or string, optional) gasPrice SBTC price per gas unit  
(6).senderaddress (string, optional) The sbtc address that will be used as sender  
(7).broadcast (bool, optional, default=true) Whether to broadcast the transaction or not  
(8).changeToSender (bool, optional, default=true) Return the change to the sender  
  
Results:  
(1). txid the transaction id  
(2).sender address of the sender  
(3).hash160 ripemd-160 hash of the sender  
eg:  
./sbtc-cli --datadir=../datadir/sbtc/testB sendtocontract 5a2e68b20e753ad70d6bc80afafa44289be9e2f3 c6888fa1000000000000000000000000000000000000000000000000000000000000000b 0 2500000 0.00000040 myJnk8kVfgeVKEVSHezUnkcwpDV1kTiksc true true   
  
3. getaccountinfo  
功能: 获取合约账户信息  
params:  
(1). address (string,required) the account address  
Results:  
(1). address the account address  
(2). balance the balance of account  
(3). storage  
(4). code

eg: ./sbtc-cli --datadir=../datadir/sbtc/testB getaccountinfo 5a2e68b20e753ad70d6bc80afafa44289be9e2f3

4. getstorage

功能：获取合约账户存储的信息

params:  
(1). address (string,required) The address to get the storage from

(2).blocknum (string, optional) Number of block to get state from

(3).index (number, optional) Zero-based index position of the storage  
Results:  
(1). storage info in json

eg: ./sbtc-cli --datadir=../datadir/sbtc/testB getstorage 5a2e68b20e753ad70d6bc80afafa44289be9e2f3

5. callcontract

功能：调用合约

params:  
(1). address (string,required) The contract address

(2).data (string, required)The data hex string

(3).address (string, optional) The sender address hex string

(4).gasLimit (numeric or string, optional) gasLimit  
Results:  
(1).address The contract address

(2).executionResult

(3).transactionReceipt

eg: ./sbtc-cli --datadir=../datadir/sbtc/testB callcontract 4a822f9f1870407bd70074436f4f2b8b23f50c9a 70a082310000000000000000000000009fa49a4a4b68cf98ce65f2adac08c98df0652567

6. listcontract

功能：列出合约信息

params:  
(1). start (numeric or string, optional) The starting account index, default 1

(2). maxDisplay (numeric or string, optional) Max accounts to list, default 20

Results:  
(1). contract address and balance in json

eg: ./sbtc-cli --datadir=../datadir/sbtc/testB listcontract

7. gettransactionreceipt

功能：获取交易单信息

params:  
(1). hash (string, required) The transaction hash

Results:  
(1). log info in json

eg: ./sbtc-cli --datadir=../datadir/sbtc/testB gettransactionreceipt fdbce276a79b818af4321f1e2fb609f72d4e519e6562aba68a5160e0d9cbd4ad

8. searchlog

功能：查询log信息

params:  
(1). fromBlock (numeric,required) The number of the earliest block

(2).toBlock (string, required) The number of the latest block

(3).address (string, optional) An address or a list of addresses to only get logs from particular account

(4). topics (string, optional) An array of values from which at least one must appear in the log entries

(5). minconf (uint, optional, default=0) Minimal number of confirmations before a log is returned  
Results:  
(1).log info in json

eg: ./sbtc-cli --datadir=../datadir/sbtc/testB searchlogs 1 400

四. 启动一个sbtc节点

./sbtd --datadir=../datadir/sbtc/testB

五. 在IDE上，用solidity编写合约代码。

pragma solidity ^0.4.17;

contract test {

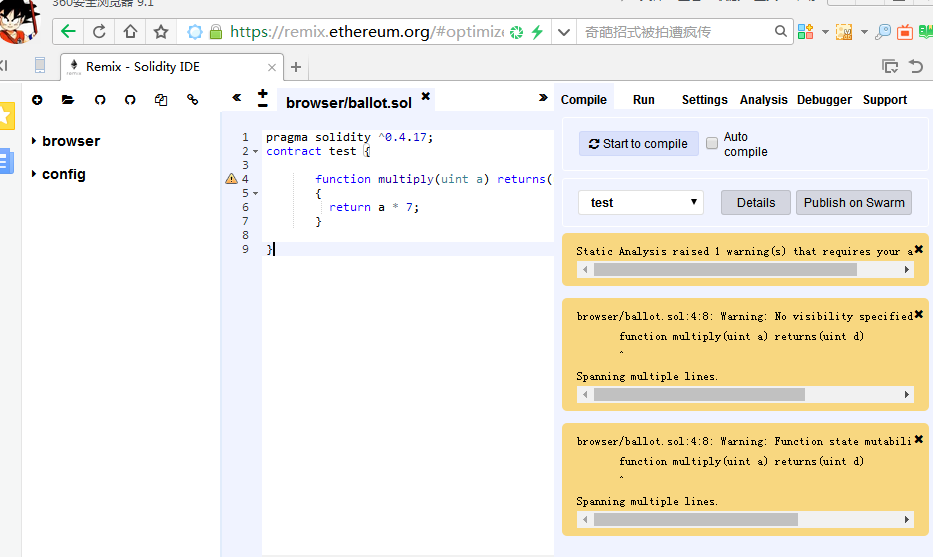
function multiply(uint a) returns(uint d)

{

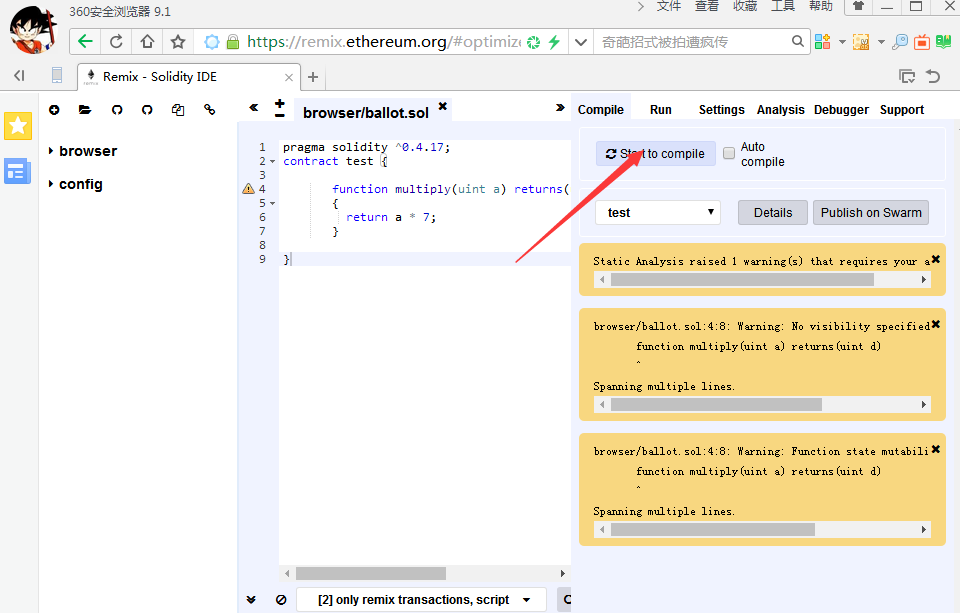
return a \* 7;

}

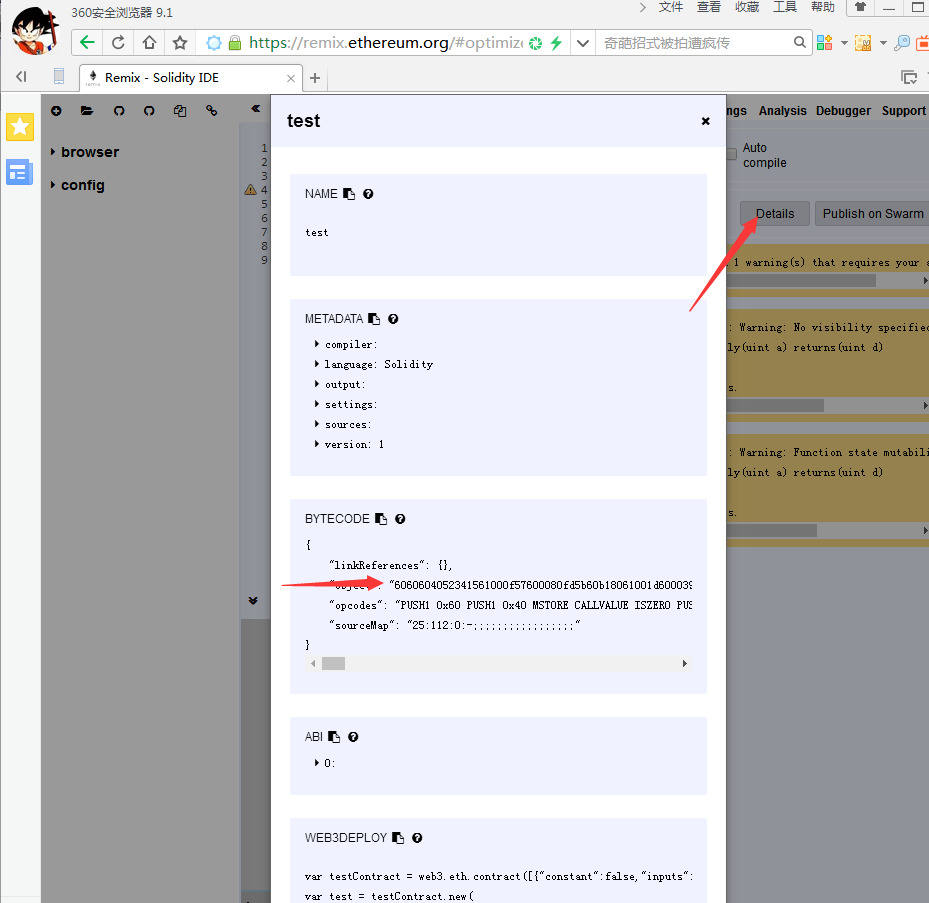
}



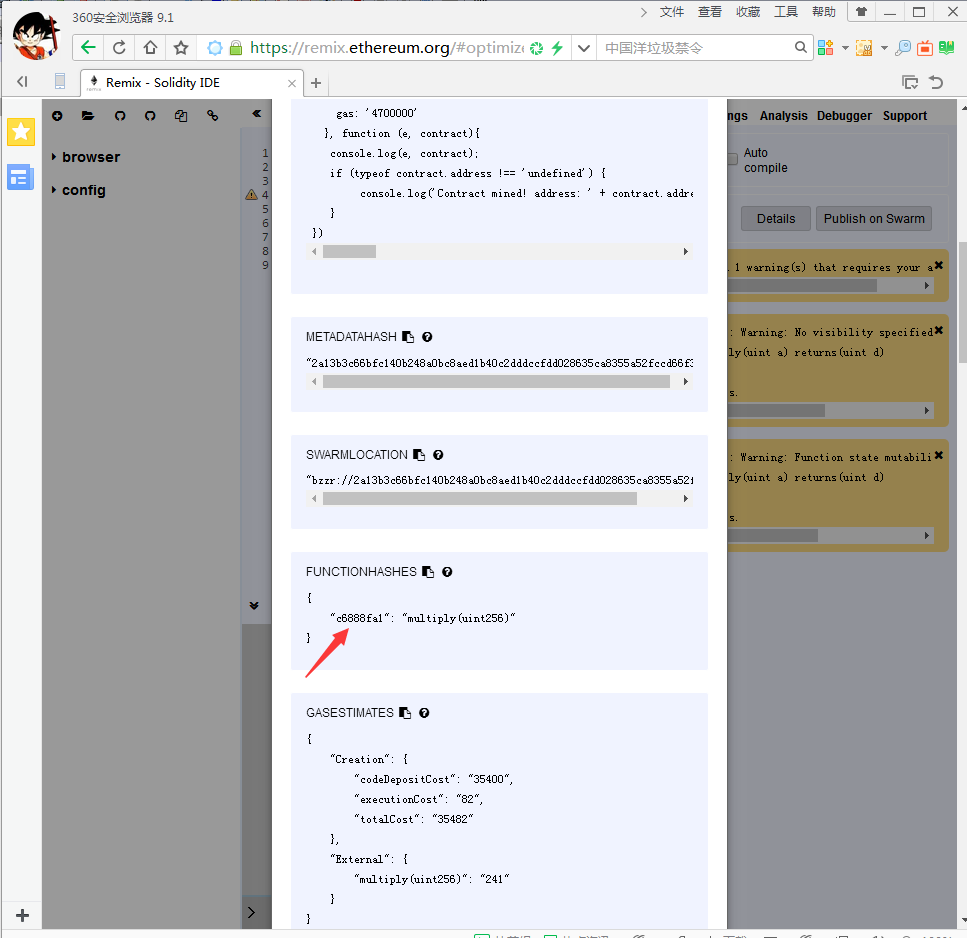
六. 在IDE上，编译源码。



获得编译后的二进制代码



获得ABI接口对应合约函数编码：



七. 使用sbtc-cli控制台部署合约代码。

1. 产生一个新地址

./sbtc-cli -datadir=/datadir/sbtc/testA getnewaddress test11

mk1Dgi3wiaS1CQxg3TXeTXegFTQPRGdzdE

./sbtc-cli -datadir=/datadir/sbtc/testA getnewaddress test22

n2KqYksoabWydPhPDSTwqnLiWDT2Nt99LY

2.挖矿101次

./sbtc-cli -datadir=/datadir/sbtc/testA generate 101

3.给新地址转钱

./sbtc-cli -datadir=/datadir/sbtc/testA sendtoaddress mk1Dgi3wiaS1CQxg3TXeTXegFTQPRGdzdE 20

./sbtc-cli -datadir=/datadir/sbtc/testA sendtoaddress

n2KqYksoabWydPhPDSTwqnLiWDT2Nt99LY 20

4.挖矿一次

./sbtc-cli -datadir=/datadir/sbtc/testA generate 1

5.部署合约

./sbtc-cli -datadir=/datadir/sbtc/testA createcontract 606060405260043610603f576000357c0100000000000000000000000000000000000000000000000000000000900463ffffffff168063c6888fa1146044575b600080fd5b3415604e57600080fd5b606260048080359060200190919050506078565b6040518082815260200191505060405180910390f35b60006007820290509190505600a165627a7a723058202a13b3c66bfc140b248a0bc8aed1b40c2dddccfdd028635ca8355a52fccd66f30029

2500000 0.00000040

mk1Dgi3wiaS1CQxg3TXeTXegFTQPRGdzdE true true

{

"txid": "0a7b4a36cf8c08c5995f4b97ce36af154afd6664dfea1b6cec4a75e627516fc6",

"sender": " mk1Dgi3wiaS1CQxg3TXeTXegFTQPRGdzdE ",

"hash160": "cca7396738473b259ad7ceaffa7840bef65b32d1",

"address": "60bb3de2baf33e5bd9692588e82afeca077b9dac"

}

执行成功后获得合约地址：

60bb3de2baf33e5bd9692588e82afeca077b9dac

6. 挖矿一次

./sbtc-cli -datadir=/datadir/sbtc/testA generate 1

八. 使用sbtc-cli控制台执行合约交易。

7.发送sendtoaddress 执行合约交易

./sbtc-cli -datadir=/datadir/sbtc/testA sendtocontract 60bb3de2baf33e5bd9692588e82afeca077b9dac c6888fa1000000000000000000000000000000000000000000000000000000000000000b 0 2500000 0.00000040 n2KqYksoabWydPhPDSTwqnLiWDT2Nt99LY

true true

c6888fa1： //ABI 定义的 multiply函数编码

000000000000000000000000000000000000000000000000000000000000000b: // 合约里的函数multiply 入参

合约交易正常运行,mout输出正确结果:

000000000000000000000000000000000000000000000000000000000000004d

8. 挖矿一次

./sbtc-cli -datadir=/datadir/sbtc/testA generate 1

9.调用callcontract 命令调用合约里的multiply函数

./sbtc-cli -datadir=/datadir/sbtc/testA callcontract 60bb3de2baf33e5bd9692588e82afeca077b9dac c6888fa1000000000000000000000000000000000000000000000000000000000000000b

c6888fa1： //ABI 定义的 multiply函数编码

000000000000000000000000000000000000000000000000000000000000000b: // 合约里的函数multiply 入参

执行完后输出：

000000000000000000000000000000000000000000000000000000000000004d