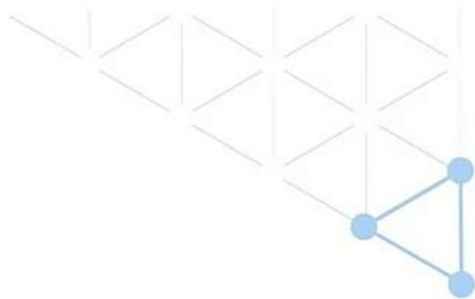


Solidity 控制结构-循环语句

- 蚂蚁链《区块链系统开发与应用》A认证系列课程

课程 目标

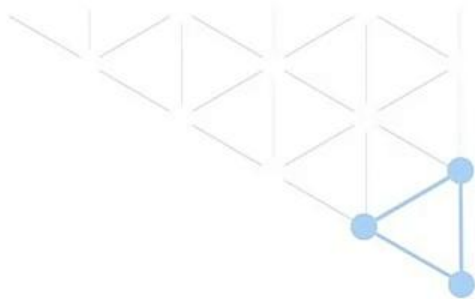
- 了解循环结构



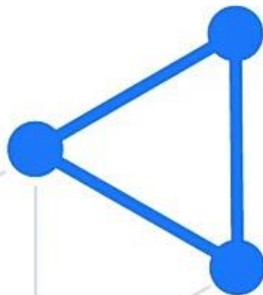
课程 目录

01

循环结构



01 循环结构



循环语句

定义

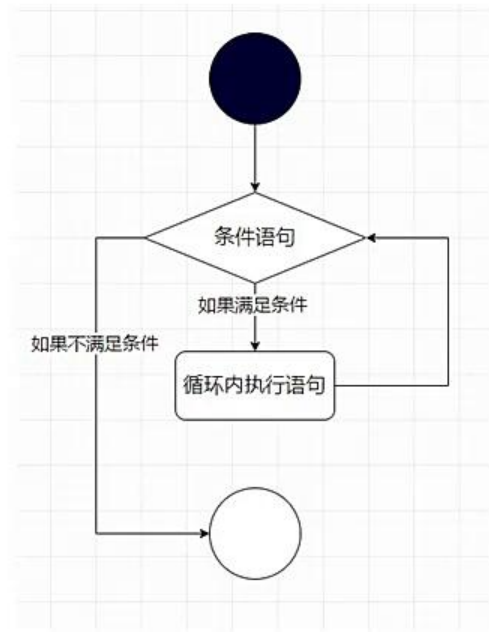
- 类似于其他高级编程语言，Solidity 语言也为我们提供了如下的三个循环结构：
 - while
 - do ... while
 - For
- 可以使用循环来完成迭代操作，也可以完成数组的遍历操作；
- 我们可以使用循环控制语句 break 和 continue 来控制循环；

循环语句

while 循环

- 使用 while 关键字来开始循环;
- 如果表达式为真，则执行 while 循环中的代码块，否则，不执行该代码块;
- while 循环的流程图如右侧所示，while 循环在 Solidity 中的语法如下图所示;

```
while (表达式) {  
    // 如果表达式为真，则执行该代码块操作  
    // 如果表达式为假，则不执行该代码块操作  
}
```





a00e36c5 - 开放联盟链... 编译

ControlDemo19

字节码 部署合约

0x608060405234801561001057600080fd5b5060d68061001f6000396000f300608060405260043610603f576000357c010000000000

合约接口说明 (ABI) 已部署合约

[{"constant": false, "inputs": [{"name": "x", "type": "uint256"}], "name": "accumulator", "outputs": [{"name": "", "type": "uint256"}]}

controldemo100

合约ID: 0xe62077dff5ca809705099ce57b2...

TX Hash: 0x752374839205e49b92957d166...

function accumulator 调用合约

tx hash
0xd593b8f5f960d1ff9b96ffcd228ddb...

input
x(uint256): 10

output
uint256: 55

log

保存

ControlDemo20.sol X

```
1 pragma solidity ^0.4.20;
2
3 // 循环语句
4 contract ControlDemo19 {
5     // while 循环: 累加
6     function accumulator(uint x) public returns(uint) {
7         uint sum = 0;
8
9         while(x > 0) {
10             sum = sum + x;
11             x = x - 1;
12         }
13
14         return sum;
15     }
16 }
```

编译 编译详情 合约分析 调试详情 交易详情

Filter

> txhash: 0x83ad5ebda196f3d6a6789583df1b109308d611552c8f39f25a9e2e86cf47b7b7

> txhash: 0xb953cf98192ec79f58ea3b92c6cb733de04d7966d6b4f80abf170952b45c74c7

> txhash: 0x0050ee5b53eff9e55421165f71b2f88cd1a144fe1d0becd62239d8db886aafcb

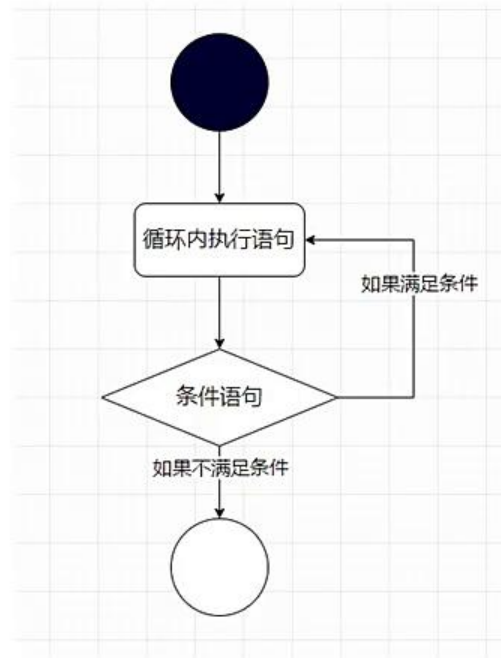
> txhash: 0x786bacd657d1155bfee08cbe9ed7e68a2965c05f51d2d49f33fd3153e1e20af9

循环语句

do ... while 循环

- do ... while 循环的表达式写在 while 关键字后面;
- do ... while 循环, 无论表达式是否为真, 都会首先执行一次代码块;
- 如果表达式为真, 则继续执行代码块; 否则, 不执行;
- Do ... while 循环的流程图如右侧所示, 在 Solidity 中的语法如下图所示;

```
do {  
    // 如果表达式为真, 则执行该代码块操作  
    // 如果表达式为假, 则不执行改代码块操作  
} while (表达式);
```





a00e36c5 - 开放联盟链... **编译**

字节码 **部署合约**

0x608060405234801561001057600080fd5b50610167806100206000396000f30060806040526004361061004c576000357c01000000

合约接口说明 (ABI) **已部署合约**

```
[{"constant": false, "inputs": [{"name": "x", "type": "int256"}], "name": "accumulator2", "outputs": [{"name": "", "type": "int256"}], "type": "function"}]
```

controldemo107

合约ID: 0x501872a33939af822d739b86233...

TX Hash: 0x0bfe7cfd96f4c3a48f454958e0c...

function accumulator2 **调用合约**

tx hash

0xf642a116bfb0349fba2feb4eb165...

ControlDemo20.sol X

```
17 }
18
19 // do ... while 循环: 累加
20 function accumulator2(int x) public returns(int) {
21     int sum = 0;
22
23     if (x <= 0) {
24         return 0;
25     }
26
27     // do ... while 循环会首先执行一次代码块, 然后再进行条件判断:
28     do {
29         sum = sum + x;
30         x = x - 1; // 退出条件一定不能忘!
31     } while(x > 0); // 不要忘记分号;
32
33     return sum;
34 }
```

编译

编译详情 合约分析 调试详情 **交易详情**

0x0bfe7cfd96f4c3a48f454958e0c22490a359f659caf2e6dbdff57820d94da7ec

output

int256: 0

log

function accumulator **调用合约**

- > txhash: 0x83ad5ebda196f3d6a6789583df1b109308d611552c8f39f25a9e2e86cf47b7b7
- > txhash: 0xb953cf98192ec79f58ea3b92c6cb733de04d7966d6b4f80abf170952b45c74c7
- > txhash: 0x0050ee5b53eff9e55421165f71b2f88cd1a144fe1d0becd62239d8db886aafcb
- > txhash: 0x786bacd657d1155bf0e08cbe9ed7e68a2965c05f51d2d49f33fd3153e1e20af9

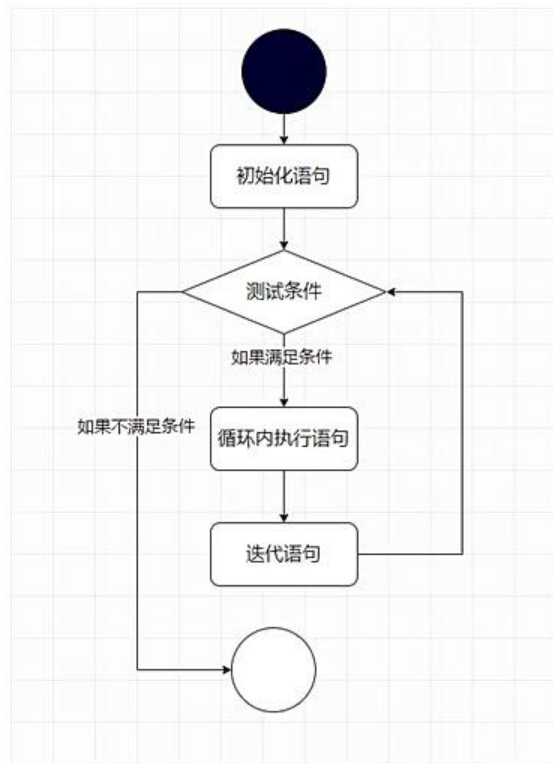
保存

循环语句

for 循环

- for 循环包含三个部分：
 - 初始化语句：首先会声明并初始化一个值；
 - 测试条件：紧接着判断该值是否满足测试条件，如果满足测试条件，则执行代码块，否则，不执行；
 - 迭代语句：如果测试条件满足，在执行完代码块之后，会执行一次迭代语句，然后再次进入到测试条件的判断中；
- for 循环的流程图如右侧所示，在 Solidity 中的语法如下图；

```
// 首先执行初始化语句；  
// 然后判断测试条件是否为真，若为真，则执行代码块内容，否则，不执行；  
for (初始化语句; 测试条件; 迭代语句) {  
    // 代码块  
    // 代码块执行一次之后，执行迭代语句；  
    // 然后再次判断测试条件是否为真，如果为真，则继续执行代码块，否则，不执行；  
}
```





编译

ControlDemo20.sol X

JAMES E. HANCOCK AND JAMES L. WATSON

部審記登

ControlDemo20.sol

ControlDemo19

字节码 8

部属合約

```
0x608060405234801561001057600080fd5b
506101dd806100206000396000f300608060
405260043610610057576000357c0100000
```

合约接口说明 (ABI)

已部署合约

```
[ { "constant": false, "inputs": [ { "name": "x",
" type": "uint256" } ], "name":
"accumulator03", "outputs": [ { "name": "",
```

controldemo108

合约ID:0x2d35a56743e739839a07bf2e384...

TX Hash:0xc0cabbe705014973e2affaf4c9d...

```
function accumulator03
```

雇用合约

tx hash

0x68bd6be2d200c2022dd450ed716...

```
x(uint256): 3
```

保存

```

24         return 0;
25     }
26
27     // do ... while 循环会首先执行一次代码块，然后再进行条件判断。
28     do {
29         sum = sum + x;
30         x = x - 1; // 退出条件一定不能忘！
31     } while(x > 0); // 不要忘记分号；
32
33     return sum;
34 }
35
36 // for 循环：累加
37 function accumulator03(uint x) public returns(uint) {
38     // for(初始化语句; 测试条件; 循环退出语句){}
39     for (uint sum = 0; x > 0; x = x - 1) {
40         sum = sum + x;
41     }
42 }

```

100

[编译详情](#)

合约分析

调试详情

交易详情

```
> txhash: 0x83ad5ebda196f3d6a6789583df1b109308d611552c8f39f25a9e2e86cf47b7b7
```

```
> txhash: 0xb953cf98192ec79f58ea3b92c6cb733de04d7966d6b4f80abf170952b45c74c7
```

```
> txhash: 0x0050ee5b53eff9e55421165f71b2f88cd1a144fe1d0becd62239d8db886aafcb
```

```
> txhash: 0x786bacd657d1155bfee08cbe9ed7e68a2965c05f51d2d49f33fd3153e1e20af9
```

08:14



a00e36c5 - 开放联盟链... 编译

ControlDemo20.sol

ControlDemo19

字节码 部署合约

0x608060405234801561001057600080fd5b50610261806100206000396000f300608060405260043610610062576000357c0100000

合约接口说明 (ABI) 已部署合约

[{"constant": false, "inputs": [{"name": "x", "type": "int256"}], "name": "accumulator4", "outputs": [{"name": "", "type": "int256"}]}

controldemo111

合约ID: 0x4eb58e18541e747bcea7aad4d83...

TX Hash: 0x787a4554919451b2527052657...

function accumulator4 调用合约

tx hash

0x158d27824c0b221134ab14b5c134...

input

x(int256): 5

output

int256: 8

保存

ControlDemo20.sol X

```
43
44     return sum;
45 }
46
47 // break 和 continue
48 function accumulator4(int x) public returns(int) {
49     int sum = 0;
50
51     while(x > 0) {
52         // 实际上我们计算的是 (x-1) 的累加结果
53         x = x - 1;
54
55         // break 会退出整个循环，后续循环不会再执行
56         // continue 会推出本次循环，后续循环还是会执行
57         if (x == 2) {
58             continue;
59         }
60
61         sum = sum + x;
62     }
63 }
```

编译

编译详情 合约分析 调试详情 交易详情

Filter

- > txhash: 0x83ad5ebda196f3d6a6789583df1b109308d611552c8f39f25a9e2e86cf47b7b7
- > txhash: 0xb953cf98192ec79f58ea3b92c6cb733de04d7966d6b4f80abf170952b45c74c7
- > txhash: 0x0050ee5b53eff9e55421165f71b2f88cd1a144fe1d0becd62239d8db886aafcb
- > txhash: 0x786bacd657d1155bfee08cbe9ed7e68a2965c05f51d2d49f33fd3153e1e20af9

循环语句

注意事项

- 由于 do ... while 循环会在不判断表达式的前提下直接执行一次代码块，所以如果不注意代码块中的逻辑，会经常出现错误。因此，对于循环语句，我们推荐使用 while 循环和 for 循环；
- 注意不要出现死循环，否则合约调用会出错；

谢谢



蚂蚁集团
ANT GROUP



蚂蚁链
ANTCHAIN