

# 『0003』 - Solidity合约结构（初识别状态变量、局部变量、构造函数、析构函数）

孔壹学院：国内区块链职业教育领先品牌

作者：黎跃春，区块链、高可用架构工程师

微信：liyc1215 QQ群：348924182 博客：<http://liyuechun.org>

## 什么是合约？

在区块链上运行的程序，通常称为 智能合约（Smart Contract）📄。所以通常会把写 区块链程序 改称 写智能合约。

简单点来讲，合约就是运行在区块链上的一段程序。

## 一个完整的合约

```
pragma solidity ^0.4.4;

contract Counter {

    uint count = 0;
    address owner;

    function Counter() {
        owner = msg.sender;
    }

    function increment() public {
        uint step = 10;
        if (owner == msg.sender) {
            count = count + step;
        }
    }

    function getCount() constant returns (uint) {
        return count;
    }

    function kill() {
```

```
        if (owner == msg.sender) {
            selfdestruct(owner);
        }
    }
}
```

## 版本声明

```
pragma solidity ^0.4.4;
```

`pragma solidity` 代表 `solidity` 版本声明，`0.4.4` 代表 `solidity` 版本，`^` 表示向上兼容，`^0.4.4` 表示 `solidity` 的版本在 `0.4.4 ~ 0.5.0` (不包含`0.5.0`) 的版本都可以对上面的合约代码进行编译，`0.4.5` , `0.4.8` 等等可以用来修复前面的 `solidity` 存在的一些 `bug` 。

## 合约声明

`contract` 是合约声明的关键字，`Counter` 是合约名字，`contract Counter` 就是声明一个 `Counter` 合约。

`contract` 相当于其他语言中的 `class`，`Counter` 相当于类名，`contract Counter` 相当于 `class Counter extends Contract`。

## 状态变量

```
uint count = 0;
address owner;
```

`count` 和 `owner` 就是状态变量，合约中的状态变量相当于 类 中的属性变量。

## 构造函数（Constructor）

`function Counter()` 函数名和合约名相同时，此函数是合约的构造函数，当合约对象创建时，会先调用构造函数对相关数据进行初始化处理。

## 成员函数

`function increment() public` 和 `function getCount() constant returns (uint)` 都是 `Counter` 合约的成员函数，成员函数在iOS里面叫做方法、行为，合约实例可以调用成员函数

处理相关操作。当调用 `increment()` 函数时，会让 状态变量`count` 增加 `step` 。当调用 `getCount()` 时会得到状态变量 `count` 的值。

## 本地变量

```
function increment() public {
    uint step = 10;
    if (owner == msg.sender) {
        count = count + step;
    }
}
```

`increment()` 方法中声明的 `step` 就是局部变量。局部变量只在离它最近的 `{}` 内容使用。

## 析构函数（`selfdestruct`）

析构函数 和 构造函数 对应，构造函数是初始化数据，而析构函数是销毁数据。在 `counter` 合约中，当我们手动调用 `kill` 函数时，就会调用 `selfdestruct(owner)` 销毁当前合约。

## 技术交流

- 区块链技术交流QQ群：348924182
- 「区块链部落」官方公众号



长按，识别二维码，加关注

