蚂蚁集团 ANT GROUP | 蚂蚁链 ANTCHAIN

Solidity基本数据类型-数据位置

- 蚂蚁链《区块链系统开发与应用》A认证系列课程

课程
目录

01 数据位置

01 数据位置

数据位置

回顾之前的值类型和引用类型

- 我们之前讨论过值类型和引用类型：
  - 值类型：值类型的变量将始终按照值来传递；
  - 引用类型：引用类型的变量在传递时传递的是该变量的引用；
- 在处理复杂类型（占用空间超过256位的类型，常见的有字符串、数组、结构体、枚举类型）时，我们需要更加谨慎。由于拷贝这些类型变量的开销非常大，会消耗大量的 Gas，所以我们不得不考虑它的存储位置，是将它们保存在“内存”中，还是“存储”中；

数据位置

Storage

- 该存储位置存储永久数据，这意味着该数据可以被合约中的所有函数访问。可以把它视为计算机的硬盘数据，所有数据都永久存储；
- 保存在存储区(Storage)中的变量，以智能合约的状态存储，并且在函数调用之间保持持久性。与其他数据位置相比，存储区数据位置的成本较高；

```
pragma solidity ^0.4.24;

contract SolidityTest {
  uint storedData;      // 状态变量

  constructor() public {
    storedData = 10;   // 使用状态变量
  }
}
```

BaaS 平台 | BaaS 平台 | BaaS 平台

Storage location can only be given for array or struct types

蚂蚁集团 ANT GROUP | 金融科技 | 管理控制台 | 产品与服务 | 自动保存

a00e36c5 - 开放联盟链... | 编译

Solidity | 0.4.23

SolidityBasicCourse

- IdentityDemo10.sol
- IntDemo07.sol
- BoolAndEnum08.sol
- StringDemo09.sol
- DataLocation11.sol

DataLocation11.sol

```
1  pragma solidity ^0.4.20;
2
3  /**
4      状态变量详解
5  */
6  contract DataLocation11 {
7      // 对于状态变量来说，其数据位置强制指定为：storage
8      uint public storedData;
9  }
```

编译

编译详情 | 合约分析 | 调试详情 | 交易详情

Warning 28

```
Warning: This is a pre-release compiler version, please do not use it in production.
BoolAndEnum08.sol:23:23: Warning: This declaration shadows an existing declaration.
    function setSize1(Size x) public {
                      ^----^
BoolAndEnum08.sol:14:5: The shadowed declaration is here:
    bool public x = (2 < 1) && (2 == 2);  // false
    ^---------------------------------^
```

保存

08:16

在这里输入你要搜索的内容

英 17:42 2020/12/20

数据位置

Storage和 memory 补充说明

- ·在蚂蚁链平台上，局部变量和函数参数，我们只能对数组或结构体指定 Storage；
- 对于 public 类型的函数的参数我们只能使用 memory 位置；
- Storage 是静态分配的，所以我们无法将一个动态的函数参数赋值给Storage位置的变量；
- 我们可以将函数参数声明为 storage 位置， 这样 Storage 位置的局部变脸即可接收该值；

BaaS 平台 | BaaS 平台 | BaaS 平台

Storage location can only be given for array or struct types

蚂蚁集团 ANT GROUP | 金融科技 | 管理控制台 | 产品与服务

a00e36c5 - 开放联盟链... | 编译

Solidity | 0.4.23

- SolidityBasicCourse
  - IdentityDemo10.sol
  - IntDemo07.sol
  - BoolAndEnum08.sol
  - StringDemo09.sol
  - DataLocation11.sol

保存

DataLocation11.sol

```
1  pragma solidity ^0.4.20;
2
3  /**
4      状态变量详解
5  */
6  contract DataLocation11 {
7      // 对于状态变量来说，其数据位置强制指定为：storage
8      uint public storedData;
9
10     // 对于函数参数来说，其默认位置为：memory
11     function defineArray(uint a) public {
12         // 对于局部变量来说，其默认数据位置为：storage
13         uint x;
14     }
15 }
```

编译

编译详情 | 合约分析 | 调试详情 | 交易详情

Warning 28

```
Warning: This is a pre-release compiler version, please do not use it in production.
BoolAndEnum08.sol:23:23: Warning: This declaration shadows an existing declaration.
    function setSize1(Size x) public {
                      ^----^
BoolAndEnum08.sol:14:5: The shadowed declaration is here:
    bool public x = (2 < 1) && (2 == 2);  // false
    ^---------------------------------^
```

08:16

在这里输入你要搜索的内容

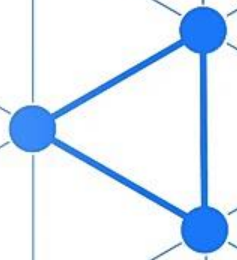
英 17:42 2020/12/20

数据位置

重要性

- 数据位置的指定非常重要，因为它们影响着赋值行为：
  - 在存储 storage 和内存 memory 之间两两赋值，或者存储 storage 向状态变量（甚至是从其它状态变量）赋值都会创建一份独立的拷贝。 然而状态变量向局部变量赋值时仅仅传递一个引用，而且这个引用总是指向状态变量，因此后者改变的同时前者也会发生改变。 另一方面，从一个内存 memory 存储的引用类型向另一个内存 memory 存储的引用类型赋值并不会创建拷贝；
- 要永久性存储，可以保存在存储区 Storage；
- 要消耗更少，局部变量可以选择使用 memory；

谢 谢



蚂蚁集团 ANT GROUP | 蚂蚁链 ANTCHAIN