# Solidity

## 基本内容

(1）为什么要学习solidity

Solidity 是一门为实现智能合约而创建的**高级编程语言**。其设计的目的是**能在以太坊虚拟机（EVM）上运行**。 Solidity 是**静态类型语言，支持继承、库和复杂的用户定义类型**。其被广泛运用在智能合约的部署上，因此很适合作为区块链工程专业学生的我们学习，特别有关Dapp等新型技术的学习。

（2）solidity源文件构成

solidity主要由三部分构成：

**pragma 指令**：指定该代码编写版本

（Solidity ^0.6.8 以上版本要求引入 SPDX 许可证：即**// SPDX-License-Identifier: MIT** SPDX标准规定了别人是否拥有商业开发，学习使用等权利。MIT 规定了其他人随便用该代码，但出问题不负责。）

**contract智能合约定义：**智能合约的主要内容

**导入文件：**由import语句引入，同java相似

（2）solidity注释格式：

**单行注释：**//之后到行尾的文本，都被看作注释，编译器忽略此内容

**块注释：**/\* 与 \*/ 之间的文本被看作注释， 编译器忽略此内容

（3）solidity数据类型：

①solidity无null和undefined概念

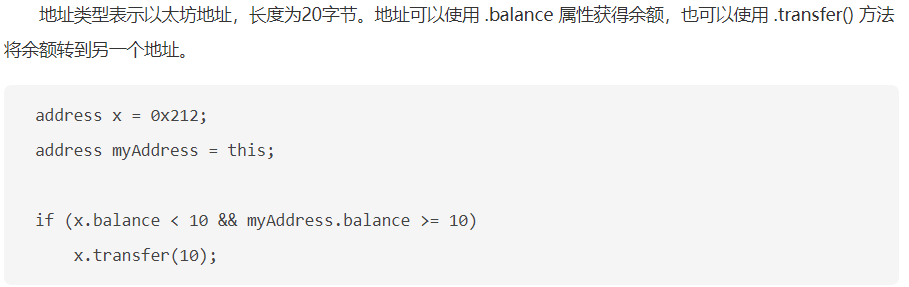
②数据类型分为值类型和引用类型

**值类型**：**布尔型**（bool，存在true和false两种可能值）

**整型**



**地址类型**



**枚举型**

solidity 的枚举类型 enum 是一种用户自定义类型，用于表示多种状态。

枚举类型 enum 内部就是一个自定义的整型，默认的类型为unit8，当枚举数足够多时，它会自动变成unint16。

枚举类型 enum可以与整数进行显式转换，但不能进行隐式转换。显示转换会在运行时检查数值范围，如果不匹配，将会引起异常。

枚举类型 enum 至少应该有一名成员。

**函数型**

函数型有以下特点：

• 函数可以赋值给函数型变量

• 函数型的函数参数用于将函数传递给另外一个函数

• 函数型的函数参数也可用于从另外一个函数中返回一个函数

• 两个函数型的参数类型、返回类型、内部/外部属性都相同时，可以向状态可变性限制条件更松的方向隐式转换 pure函数可以转换为view和non-payable函数 view函数可以转换为non-payable函数 payable函数可以转换为non-payable函数 • 调用未初始化或者已经被delete的函数变量将会报错panic error

**引用类型：**

同java相同，其引用类型也包括数组型（bytes,string），结构体型(struct)，还有映射型(mapping)

## ****Solidity相关：****

**运算符：**（同java等语言差不多）

算术运算符：+（加）-（减）\*（乘）/（除）%（取模）++（自增）--（自减）

**比较运算符**：

==（相等） ！=（不等） >(大于) <（小于） >= (大于等于) <=（小于等于）

可配合if等语句做条件判断：

如if（i==0）{

return “0”;

}如果i==0则返回0值

### ****逻辑运算符****

&&（如果两个操作数都非零，则条件为真）

||(如果这两个操作数中有一个非零，则条件为真)

! (反转操作数的逻辑状态。如果条件为真，则逻辑非操作将使其为假)

位运算符：

<<(左移位) >>(右位移) & (位与) | (位或) ^ (位异或) ~ (位非)

### ****Solidity条件语句：****

条件表达式可为上面提到的 关系表达式 、逻辑表达式，甚至是数值表达式

if (条件表达式) {

被执行语句(如果条件为真)

}

if (条件表达式) {

被执行语句(如果条件为真)

} else {

被执行语句(如果条件为假)

}

if (条件表达式 1) {

被执行语句(如果条件 1 为真)

} else if (条件表达式 2) {

被执行语句(如果条件 2 为真)

} else if (条件表达式 3) {

被执行语句(如果条件 3 为真)

} else {

被执行语句(如果所有条件为假)

}

### ****Solidity事件（event）：****

在 Solidity 代码中，使用 event 关键字来定义一个事件 Event，语法如下：

event EventName(xxx);

在 Solidity 代码中，使用 emit 关键字来触发一个事件 Event，语法如下：

emit EventName(xxx);

Xxx中填需要记录的数据，需要注意要做到数据的对应与匹配。

Eg：event WithdrawTo(address indexed sender,address indexed \_toAddress, uint256 wad);

function withdrawTo(address payable \_toAddress, uint256 wad)public payable{

require(balanceOf[msg.sender] >= wad);

balanceOf[msg.sender] -= wad;

transferFrom(msg.sender,\_toAddress,wad);

\_toAddress.transfer(wad);

emit WithdrawTo(msg.sender,\_toAddress, wad);

}

**Indexed：**被indexed标记过的参数可以通过链外进行搜索查询。

普惠金融：

随着金融科技的发展，数字普惠金融已逐渐成为了实现普惠金融的重要模式，切实解决了普惠金融在社会实践中存在的各种问题，推动了金融市场的发展。

近年来，在政策引领和推动下，金融机构着力从多方面采取措施破解普惠金融发展难题，改进普惠群体金融服务，不断提升普惠金融服务的可得性和满意度，在助企纾困、乡村振兴和保障⺠生等方面发挥了积极作用。WeLab的联合创始人兼CEO⻰沛智(SimonLoong)介绍说，自2013年在香港创办以来，公司现在已经拥有千余名员工，在香港、内地和印尼都有业务布局。基于在香港市场积累的能力，公司能够进入像印尼这样的市场迅速开展新业务，服务当地年轻的用戶群，并帮助当地推动普惠金融的发展。

而数字普惠金融是数字技术与普惠金融的有机结合，是指在数字技术的支持下通过金融服务促进普惠金融的行动。作为面向社会各阶层提供金融服务的重要形式，普惠金融的成本可负担性一直备受关注，而以大数据、云计算、区块链以及人工智能为主要代表的数字技术，能够有效降低普惠金融的风险识别成本、数据处理成本以及传统经营成本，并经过不断迭代升级最终促成移动支付、网上理财、小微信贷以及互联网保险等一系列数字普惠金融的相关应用。普惠金融同区块链，web3等项目的有机结合可使普惠金融走向更开放和具体，走出原来普惠金融不透明，不下放的误区。

近期香港金融周中，中国银保监会副主席肖远企先生的演讲中指出“近年来我国金融机构注重加强金融科技应用，不断提高金融可获得性，持续降低服务成本，重点覆盖金融服务薄弱和空白领域，突破物理网点和地域限制，触达更多的普惠金融客群。2021年内地小微企业互联网流动资金贷款余额同比增⻓47%，远高于一般贷款增速；互联网财产保险保费同比增⻓8%，较财产险行业整体保费增速高出7个百分点。目前内地银行通过线上渠道发生的业务交易笔数占各类渠道比重已超过90%。”更说明了我国对数字普惠金融的重视。而在11月6日举行的第五届中国国际进口博览会上，中国银行也正式发布了“I·SMART—中银数字普惠金融服务+”，着重“SMART”五个方面全面提升普惠金融服务质效。

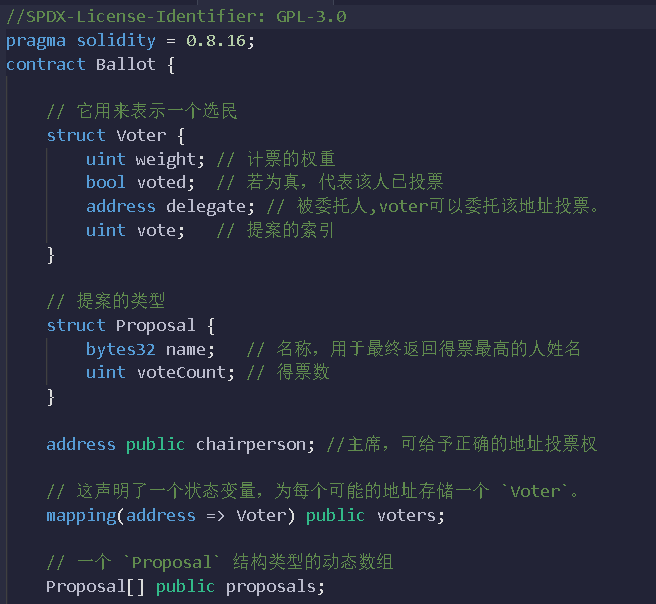
## solidity投票合约的编写

#### 基本思路

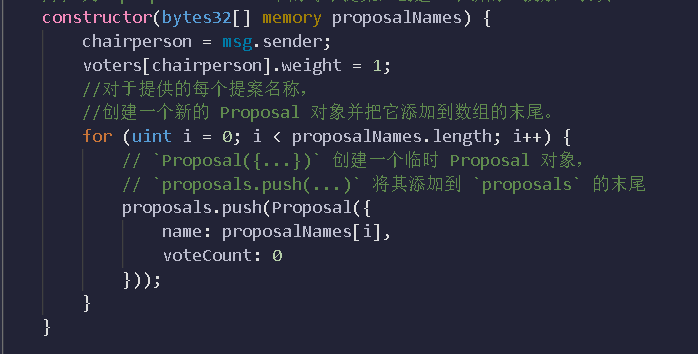
电子投票的主要问题是如何将投票权分配给正确的人员以及如何防止被操纵,以及如何安全进行委托投票，同时使得投票又是自动和透明的。

想法是为每一个投票表决创建一份合约，为每个选项提供简称。然后作为合约的创造者（主席）,将给予每个独立的地址投票权。地址后面的人可以选择自己投票或者委托给他们信任的人来投票。在投票结束后通过函数返回获得最多投票的提案。

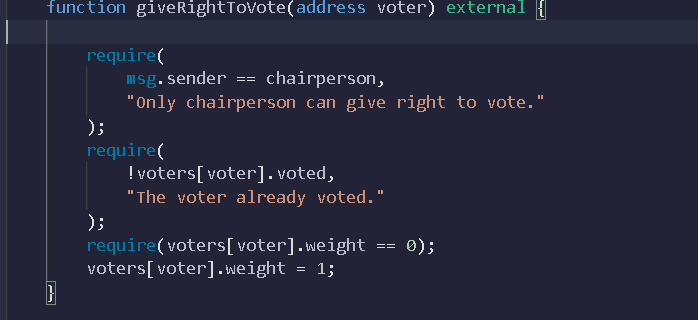
#### 代码实现



构造函数，为proposalNames的每个提案，创建一个新的投票表决



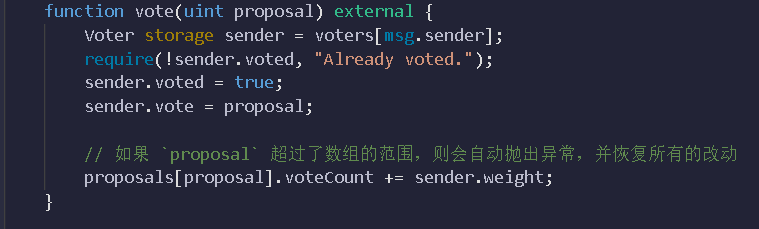
授权 `voter` 对这个（投票）表决进行投票，只有 `chairperson` 可以调用该函数。



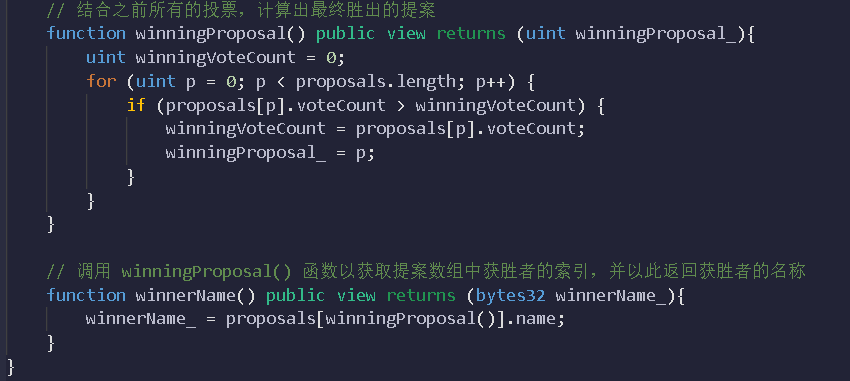
委托的实现



投票



结果



参考solidity文档

<https://learnblockchain.cn/docs/solidity/solidity-by-example.html>

### solidity中特殊函数的学习

#### receive接受以太函数

##### 定义

1.solidity 接收函数 receive 没有参数、没有返回值。

2.solidity 向合约转账，发送 Eth，就会执行 receive 函数。

3.如果没有定义接收函数 receive，就会执行 fallback 函数。

##### 声明语法

receive() external payable{

}

#### Fallback回退函数

##### 定义

1.回退函数是一个不接受任何参数也不返回任何值的特殊函数

2.如果在对合约的调用中，没有与给定的函数标识符匹配时，回退函数会被调用.

3.每当合约接受到以太币时，且没有receive函数时，回退函数会被调用

4.一个合约中最多有一个回退函数

##### 声明语法

fallback() external [payable]

回退函数没有function关键字，必须是external可见性(仅允许被外部合约调用)，如果回退函数需要接受以太币，则标记为payable关键字。

#### 区别

Receive函数只有在合约转账时调用，而Fallback函数除了在合约转账时调用，在合约没有函数匹配或需要向合约发送附加数据时,也调用Fallback函数.

#### 调用流程

