

# 法律声明

---

- 本课件包括：演示文稿，示例，代码，题库，视频和声音等，小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。



关注 小象学院

---

# 区块链编程: Solidity以太坊智能合约

王亮

---

# 第四课 solidity编程:智能合约实现

## 4.17 错误和异常处理

# Solidity的异常处理

---

- ❑ Solidity使用“状态恢复”来处理错误
- ❑ 有某些情况下，异常是自动抛出来的
- ❑ 抛出异常的效果是当前的执行被终止且被撤销(值的改变和帐户余额的变化都会被回退)
- ❑ Solidity暂时没有捕捉异常的方法(try...catch)

# assert/require

---

- ❑ 函数assert和require来进行条件检查，如果条件不满足则抛出异常
- ❑ assert函数通常用来测试内部错误
- ❑ require函数来检查输入变量或合约状态变量是否满足条件，以及验证调用外部合约返回值

# 异常例子

```
function sendHalf(address addr) public payable returns (uint balance) {  
    require(msg.value % 2 == 0); // require判断, 不符合则抛出异常  
    //抛出异常, 后续的代码不再执  
    uint balanceBeforeTransfer = this.balance;  
    addr.transfer(msg.value / 2);  
    //异常则后续代码不运行  
    assert(this.balance == balanceBeforeTransfer - msg.value / 2);  
    return this.balance;  
}
```

# revert/throw

---

- ❑ revert 函数可用于标记错误
- ❑ throw 同 revert 类似，已不建议使用。

# 异常例子

---

```
if (msg.sender != owner) { revert(); }  
assert(msg.sender == owner);  
require(msg.sender == owner);
```



# 常见异常

---

- ❑ 数组访问越界，或是负的序号值访问数组。
- ❑ 调用require/assert, 但参数值为false。
- ❑ .transfer()执行失败
- ❑ 如果你的public的函数在没有payable关键字时，却尝试在接收ether。
- ❑ 如果你通过消息调用一个函数，但在调用的过程中，并没有正确结束(如gas不足等)

---

# 流程演示

# 联系我们

---

## 小象学院：互联网新技术在线教育领航者

— 微信公众号：**小象学院**

