

# 第28章 确定的表达式 求值顺序

《现代C++语言核心特性解析》 谢丙堃

# 表达式求值顺序的不确定性

- 例子

```
void f2() {  
    std::string s = "but I have heard it works even if you don't believe in it";  
    s.replace(0, 4, "")  
    .replace(s.find("even"), 4, "only")  
    .replace(s.find(" don't"), 6, "");  
    assert(s == "I have heard it works only if you believe in it"); // OK  
}
```

# 表达式求值顺序的不确定性

- 理想的顺序 vs 实际顺序

```
1. replace(0, 4, "")
2. tmp1 = find("even")
3. replace(tmp1, 4, "only")
4. tmp2 = find(" don't")
5. replace(tmp2, 6, "")
```

```
1. tmp1 = find("even")
2. tmp2 = find(" don't")
3. replace(0, 4, "")
4. replace(tmp1, 4, "only")
5. replace(tmp2, 6, "")
```

# 表达式求值顺序详解

- 函数表达式一定会在函数的参数之前求值

```
foo(a, b, c);
```

- 后缀表达式和移位运算符求值总是从左往右的

```
E1[E2]
```

```
E1.E2
```

```
E1.*E2
```

```
E1->*E2
```

```
E1<<E2
```

```
E1>>E2
```

# 表达式求值顺序详解

- 赋值表达式求值总是从右往左的

`E1=E2`

`E1+=E2`

`E1-=E2`

`E1*=E2`

`E1/=E2`

...

- `new`表达式的内存分配总是会优先于构造函数中参数的求值
- 重载运算符的表达式的求值顺序应由与之相应内置运算符的求值顺序确定



感谢您的观看  
欢迎关注