

# 青春都一饷，忍把浮名，换了代码轻狂。

关注DirectX

随笔 - 203, 文章 - 66, 评论 - 1728, 引用 - 0

## 导航

博客园  
首 页  
新随笔  
联 系  
订 阅  
管 理

[XML](#)

< 2010年7月 >						
日	一	二	三	四	五	六
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

## 公告

昵称: zdd  
园龄: 9年4个月  
荣誉: 推荐博客  
粉丝: 846  
关注: 21  
[+加关注](#)

## 搜索

[找找看](#)

## 常用链接

## C++ 初始化列表

### 何谓初始化列表

与其他函数不同，构造函数除了有名字，参数列表和函数体之外，还可以有初始化列表，初始化列表以冒号开头，后跟一系列以逗号分隔的初始化字段。在C++中，struct和class的唯一区别是默认访问性不同，而这里我们不考虑访问性的问题，所以下面的代码都以struct来演示。

```
struct foo
{
    string name ;
    int id ;
    foo(string s, int i):name(s), id(i){} ; // 初始化列表
};
```

### 构造函数的两个执行阶段

构造函数的执行可以分成两个阶段，初始化阶段和计算阶段，初始化阶段先于计算阶段。

#### 初始化阶段

所有类类型（class type）的成员都会在初始化阶段初始化，即使该成员没有出现在构造函数的初始化列表中。

我的随笔  
我的评论  
我的参与  
最新评论  
我的标签

## 随笔分类

Android(16)  
C/C++(19)  
D3D10  
D3D11(1)  
D3D9(4)  
Demo(3)  
Direct2D(14)  
DirectInput(1)  
DirectWrite(3)  
DirectX(58)  
DXGI  
Git  
HLSL  
IOS(3)  
Math(16)  
OpenGL(1)  
Perl(1)  
PS(1)  
Shader(2)  
Swift  
Unity3D  
Vim  
XAudio2  
编程语言(9)  
代码片段(19)  
教程  
数据结构与算法(26)  
图形学(10)  
移动开发(16)  
杂(24)

## 随笔档案

## 计算阶段

一般用于执行构造函数体内的赋值操作，下面的代码定义两个结构体，其中Test1有构造函数，拷贝构造函数及赋值运算符，为的是方便查看结果。Test2是个测试类，它以Test1的对象为成员，我们看一下Test2的构造函数是怎么样执行的。



```
struct Test1
{
    Test1() // 无参构造函数
    {
        cout << "Construct Test1" << endl ;
    }

    Test1(const Test1& t1) // 拷贝构造函数
    {
        cout << "Copy constructor for Test1" << endl ;
        this->a = t1.a ;
    }

    Test1& operator = (const Test1& t1) // 赋值运算符
    {
        cout << "assignment for Test1" << endl ;
        this->a = t1.a ;
        return *this;
    }

    int a ;
};

struct Test2
{
    Test1 test1 ;
    Test2(Test1 &t1)
    {
```

2018年3月 (1)  
2017年8月 (1)  
2016年10月 (1)  
2016年9月 (1)  
2016年8月 (1)  
2016年6月 (1)  
2016年1月 (4)  
2015年12月 (1)  
2015年11月 (1)  
2015年10月 (2)  
2015年8月 (5)  
2015年7月 (1)  
2014年11月 (2)  
2014年6月 (2)  
2014年5月 (1)  
2013年9月 (1)  
2013年7月 (1)  
2013年6月 (1)  
2013年4月 (2)  
2013年3月 (3)  
2013年2月 (3)  
2012年11月 (2)  
2012年10月 (2)  
2012年9月 (5)  
2012年8月 (6)  
2012年7月 (5)  
2012年6月 (2)  
2012年5月 (2)  
2012年4月 (1)  
2011年9月 (2)  
2011年7月 (3)  
2011年6月 (2)  
2011年5月 (4)  
2011年4月 (3)  
2011年3月 (3)  
2011年2月 (4)  
2010年12月 (3)

```
        test1 = t1 ;  
    }  
};
```

### 调用代码

```
Test1 t1 ;  
Test2 t2(t1) ;
```

### 输出

```
Construct Test1  
Construct Test1  
assignment for Test1
```

解释一下，第一行输出对应调用代码中第一行，构造一个Test1对象。第二行输出对应Test2构造函数中的代码，用默认的构造函数初始化对象test1，这就是所谓的初始化阶段。第三行输出对应Test1的赋值运算符，对test1执行赋值操作，这就是所谓的计算阶段。

## 为什么使用初始化列表

初始化类的成员有两种方式，一是使用初始化列表，二是在构造函数体内进行赋值操作。使用初始化列表主要是基于性能问题，对于内置类型，如int，float等，使用初始化列表和在构造函数体内初始化差别不是很大，但是对于类类型来说，最好使用初始化列表，为什么呢？由上面的测试可知，使用初始化列表少了一次调用默认构造函数的过程，这对于数据密集型的类来说，是非常高效的。同样看上面的例子，我们使用初始化列表来实现Test2的构造函数

```
struct Test2  
{  
    Test1 test1 ;
```

2010年11月 (3)  
2010年9月 (1)  
2010年8月 (11)  
2010年7月 (17)  
2010年6月 (9)  
2010年5月 (12)  
2010年4月 (2)  
2010年3月 (16)  
2010年2月 (3)  
2010年1月 (2)  
2009年12月 (2)  
2009年11月 (9)  
2009年10月 (11)  
2009年9月 (3)  
2009年8月 (1)  
2009年7月 (4)  
2009年6月 (10)  
2009年5月 (2)

## Game Engine

DX11 Tutorials  
Irrlicht  
MathWords  
Ogre

## OpenGL

## 缤纷世界

3D Controls  
3D Fractal  
chaos files  
ChaosinChinese  
Cloth Simulation  
DirectX Developer Center  
DirectX document online  
DX tutorials  
EuclideanSpace  
Fractal Video  
geometricktools

```
Test2(Test1 &t1):test1(t1){}  
}
```

使用同样的调用代码，输出结果如下。

```
Construct Test1  
Copy constructor for Test1
```

第一行输出对应 调用代码的第一行。第二行输出对应Test2的初始化列表，直接调用拷贝构造函数初始化test1，省去了调用默认构造函数的过程。所以一个好的原则是，能使用初始化列表的时候尽量使用初始化列表。

## 哪些东西必须放在初始化列表中

除了性能问题之外，有些时场合初始化列表是不可或缺的，以下几种情况时必须使用初始化列表

- 常量成员，因为常量只能初始化不能赋值，所以必须放在初始化列表里面
- 引用类型，引用必须在定义的时候初始化，并且不能重新赋值，所以也要写在初始化列表里面
- 没有默认构造函数的类类型，因为使用初始化列表可以不必调用默认构造函数来初始化，而是直接调用拷贝构造函数初始化。

对于没有默认构造函数的类，我们看一个例子。



```
struct Test1  
{  
    Test1(int a):i(a){}  
    int i ;  
};  
  
struct Test2  
{  
    Test1 test1 ;
```

Google C++ Style  
HardCode  
HyperGraph  
In Framez  
LatexEditor  
Mame  
Mandelbrot Set  
OpenGL official page  
OpenGL Tutorials  
pouet  
Ray tracing  
Rthdribl  
TechInterview  
toymaker  
W3SCHOOL  
XNA/DirectX Forum  
X-Zone  
云风的Blog

## 其他

Aogo汇编小站  
Channel 9  
Code all in one  
Emath  
Math Circle  
Math Game  
Microsoft At Home  
Microsoft At Work  
Windows forum  
中国DOS联盟

## 友情链接

LittleStart  
WW老弟

## 积分与排名

积分 - 245824  
排名 - 1007

## 最新评论

```
Test2(Test1 &t1)
{
    test1 = t1 ;
}
};
```



以上代码无法通过编译，因为Test2类中Test1 test1;需要调用默认的构造函数，但是Test1类没有无参的构造函数，但是由于Test1没有默认的构造函数，故而编译错误。正确的代码如下，使用初始化列表代替赋值操作。

```
struct Test2
{
    Test1 test1 ;
    Test2(Test1 &t1):test1(t1){}
}
```

## 成员变量的初始化顺序

成员是按照他们在类中出现的顺序进行初始化的，而不是按照他们在初始化列表出现的顺序初始化的，看代码。

```
struct foo
{
    int i ;
    int j ;
    foo(int x):i(x), j(i){}; // ok, 先初始化i, 后初始化j
};
```

再看下面的代码

```
struct foo
{
```

### 1. Re:算法-求二进制数中1的个数

完美法中妙在很好的看准了数据的一些特点：这个取膜的方式是完美法的核心，其中关键就在于输入设定为64位的数字如果我们用的是128位的话，也可以同理扩展，或者结合其它的算法。而我其实更喜欢上面那些更简单小.....

--南宫千寻

### 2. Re:射线和三角形的相交检测 (ray triangle intersection test)

老师您好！我想请问一定要加限制条件 $\text{Dot}(P,T) < 1$ ； $u + v \leq 1$ 。故想向您请教。...

--fredyfeng

### 3. Re:Direct2D教程（外篇）环境配置

什么情况会发生设备丢失 删掉所有设备相关资源重新创建？

--asight

### 4. Re:Direct2D教程（三）简单几何图形

谢谢楼主，请问direct2d中是否有快速批次绘制多条线段的方法，类似于d3d中的linelist和liststrip，我想要linelist。

--bodong

### 5. Re:如何使用数组参数

学习了，传引用就可以把元素个数传过去了

--rayinnight

## 阅读排行榜

```
int i ;
int j ;
foo(int x):j(x), i(j){} // i值未定义
};
```

这里i的值是未定义的，虽然j在初始化列表里面出现在i前面，但是i先于j定义，所以先初始化i，但i由j初始化，此时j尚未初始化，所以导致i的值未定义。所以，一个好的习惯是，按照成员定义的顺序进行初始化。

作者: [zdd](#)

出处: <http://www.cnblogs.com/graphics/>

本文版权归作者和博客园共有，欢迎转载，但未经作者同意必须保留此段声明，且在文章页面明显位置给出原文连接，否则保留追究法律责任的权利。

分类: [C/C++](#)

好文要顶

关注我

收藏该文



zdd

关注 - 21

粉丝 - 846

荣誉: [推荐博客](#)

[+加关注](#)

« 上一篇: [算法-求二进制数中1的个数](#)

» 下一篇: [正确使用STL-MAP中Erase函数](#)

33

0

posted on 2010-07-04 16:59 zdd 阅读(90543) 评论(41) 编辑 收藏

## 评论

1. 点到平面的距离公式 (108073)
2. C++ 初始化列表 (90543)
3. 算法-求二进制数中1的个数(87881)
4. 正则表达式-问号的四种用法(78295)
5. 字符串面试题（一）字符串逆序(69492)

### 评论排行榜

1. 程序员，请昂起你高贵的头！(364)
2. 关于数组的几道面试题 (96)
3. 几何变换详解(78)
4. 判断点是否在三角形内 (72)
5. 算法-求二进制数中1的个数(59)

### 推荐排行榜

1. 程序员，请昂起你高贵的头！(177)
2. 算法-求二进制数中1的个数(36)
3. C++ 初始化列表(33)
4. 字符串面试题（一）字符串逆序(22)
5. 关于数组的几道面试题 (20)

## #1楼 回复 引用

```
struct Test2
{
    Test1 test1 ;
    Test2(Test1 &t1)
    {
        test1 = t1 ;
    }
};
```

```
Test1 t1;
Test2 t2(t1);
```

这种情况下的结果中：“第二行输出对应Test2构造函数中的代码，用默认的构造函数初始化对象test1，这就是所谓的初始化阶段”

这句话的意思是调用Test2(Test1 &t1),然后参数传递时调用了Test1的构造函数对吗？如果是这样的话，那么改成初始化列表形式以后，输出结果中应该是在“copy structor for test1”后面还要输出一行“constructor test1”才对啊？

支持(0) 反对(0)

2012-10-18 10:17 | 钱吉

## #2楼[楼主] 回复 引用

@ DarkHorse

[引用](#)

```
struct Test2{
    Test1 test1 ;
    Test2(Test1 &t1)
```

```
{
test1 = t1 ;
}
};
Test1 t1;
Test2 t2(t1);
```

这种情况下的结果中：“第二行输出对应Test2构造函数中的代码，用默认的构造函数初始化对象test1，这就是所谓的初始化阶段”

这句话的意思是调用Test2(Test1 &t1),然后参数传递时调用了Test1的构造函数对吗？如果是这样的话，那么改成初始化列表形式以后，输出结果中应该是在“copy structor for test1”后面还要输出一行“...

你干嘛把第一段代码的描述用到第二段代码上呢？仔细看看第二段代码的描述，初始化列表不调用默认的构造函数。

支持(0) 反对(0)

2012-10-18 10:37 | zdd

### #3楼 回复 引用

sorry，俺是新手，没描述清楚。我主要想请教两个问题：

1、如果是这种情况：

```
struct Test2
{
Test1 test1 ;
Test2(Test1 &t1)
{
}
};
```

执行Test1 t1;Test2 t2(t1)，会执行两次Test1的无参构造过程：



```
Construct Test1  
Construct Test1
```

那么第二次执行无参构造函数是由什么引起的？

2、换成初始化列表后，为什么不执行默认构造函数了，本质是什么？

谢谢！！

支持(0) 反对(0)

2012-10-18 11:26 | 钱吉

## #4楼[楼主] 回复 引用

@ DarkHorse

我文中已经写得很清楚啦，每一行代码的输出我都有解释，为什么你还是看不懂呢？

支持(0) 反对(0)

2012-10-18 17:05 | zdd

## #5楼 回复 引用

不错

支持(0) 反对(0)

2012-10-22 16:24 | 恒月美剑

## #6楼[楼主] 回复 引用

@ 枫雁

[引用](#)

不错

欢迎多提意见.

支持(0) 反对(0)

2012-10-25 09:32 | zdd

## #7楼 回复 引用

楼主讲的详细明了，排版很好，值得我学习啊~一个疑问：为什么你都用的struct呢？用类class不是更普世一些吗

支持(0) 反对(0)

2012-11-17 20:04 | 小比丘

## #8楼[楼主] 回复 引用

@ 小比丘

[引用](#)

楼主讲的详细明了，排版很好，值得我学习啊~一个疑问：为什么你都用的struct呢？用类class不是更普世一些吗

感谢您的提醒，因为在C++中，struct和class唯一的区别就是struct的所有成员默认都是public的，所以如果不考虑成员的可访问性，这两者是没有区别的，于是为了简化，我就使用了struct。正如你所说，使用class会更好一些。

支持(1) 反对(0)

2012-11-18 01:48 | zdd

## #9楼 回复 引用

@ zdd

基本上是这样的，楼主可以将这段说明放在文章开头的部分，以示提醒。

支持(0) 反对(0)

2012-11-18 12:56 | 小比丘

## #10楼[楼主] 回复 引用

@ 小比丘

[引用](#)

@zdd基本上是这样的，楼主可以将这段说明放在文章开头的部分，以示提醒。

已经加在第一段末尾，感谢提醒，欢迎继续提意见。

支持(0) 反对(0)

2012-11-19 12:26 | zdd

## #11楼 回复 引用

写得非常好，之前我也想过写关于构造函数的很完全的应用介绍，但是一直没敢动笔。学习了

支持(0) 反对(0)

2013-01-09 20:03 | Matrix\_R

## #12楼[楼主] 回复 引用

@ Matrix\_R

[引用](#)

写得非常好，之前我也想过写关于构造函数的很完全的应用介绍，但是一直没敢动笔。学习了

过奖了，C++博大精深，还有很长的路要走！

支持(0) 反对(0)

2013-01-10 14:19 | zdd

### #13楼 回复 引用

笔记：

- 1.能使用初始化列表的时候尽量使用初始化列表。
- 2.按照成员定义的顺序进行初始化。

菜鸟受教了，谢谢博主~

支持(0) 反对(0)

2013-01-13 09:36 | Lucien23

### #14楼 回复 引用

美文，受教。

支持(0) 反对(0)

2013-08-15 16:40 | 小小舒格

### #15楼[楼主] 回复 引用

@ 小小舒格

[引用](#)

美文，受教。

感谢支持！

支持(0) 反对(0)

2013-08-15 16:54 | zdd

## #16楼 回复 引用

学习了，不错！

支持(0) 反对(0)

2013-09-09 14:14 | chenjt3533

## #17楼[楼主] 回复 引用

@\_ chenjt3533

[引用](#)

学习了，不错！

感谢支持！

支持(0) 反对(0)

2013-09-09 17:04 | zdd

## #18楼 回复 引用

楼主你好，除了构造函数有初始化列表外，子类继承也有初始化列表，那个是用来初始化基类的，但我分不清写法，能指点一下吗

支持(0) 反对(0)

2013-10-17 21:34 | 放作夥

## #19楼[楼主] 回复 引用

@ 放作夥

[引用](#)

楼主你好，除了构造函数有初始化列表外，子类继承也有初始化列表，那个是用来初始化基类的，但我分不清写法，能指点一下吗

可以为基类设置一个构造函数，然后在派生类中调用基类的构造函数来初始化。借用[stackoverflow](#)上的一段代码。

```
1  class Parent
2  {
3  protected:
4      Parent( const std::string& something ) : something( something )
5      {}
6
7      std::string something;
8  }
9
10 class Child : public Parent
11 {
12 private:
13     Child() : Parent("Hello, World!")
14     {
15     }
16 }
```

支持(1) 反对(0)

2013-10-18 09:17 | zdd

## #20楼 回复 引用

楼主你好：

成员初始化列表初始化的成员的顺序取决于成员变量在类中的声明顺序，对于只有两个成员变量的情况，您讲解的很正确，但是成员变量增加到三个的时候，我对于编译器的行为就不能理解了。如：

```
class Data{
public:
    int i;
    int j;
    int k;
    Data(int a):j(a),k(i),i(i)
    {
    }
};
```

这种情况下，i,j,k应该都是a的值，但是我的编译器输出却不是，请您讲解一下

支持(0) 反对(0)

2013-12-13 10:13 | 李承芳

## #21楼[楼主] 回复 引用

@ 李承芳

### 引用

楼主你好：成员初始化列表初始化的成员的顺序取决于成员变量在类中的声明顺序，对于只有两个成员变量的情况，您讲解的很正确，但是成员变量增加到三个的时候，我对于编译器的行为就不能理解了。如：

```
class Data{
public:
int i;
int j;
int k;
Data(int a):j(a),k(i),i(i)
{
}
};
```

这种情况下，i,j,k应该都是a的值，但是我的编译器输出却不是，请您讲解一下

你的写法有问题，按照你的写法，只有j的值等于a的值，k和i的值都是未定义的。

支持(0) 反对(0)

2013-12-13 13:22 | zdd

## #22楼 回复 引用

@ zdd

你好，我觉得成员初始化列表是按照成员变量声明的顺序去初始化成员变量的，也就是说，在我的代码中，i先被赋值a，然后j被i赋值，k被i赋值，所以应该全是a的值，不知道我的理解哪里错了，希望你讲解一下。尤其是成员初始化列表这方面，我还不是太清楚

支持(0) 反对(0)

2013-12-15 09:34 | 李承芳



## #23楼[楼主] 回复 引用

@ 李承芳

[引用](#)

@zdd你好，我觉得成员初始化列表是按照成员变量声明的顺序去初始化成员变量的，也就是说，在我的代码中，i先被赋值a，然后j被i赋值，k被i赋值，所以应该全是a的值，不知道我的理解哪里错了，希望你讲解一下。尤其是成员初始化列表这方面，我还不是太清楚

说的没错，可是你仔细看看你的代码，i是用a初始化的么？你用i初始化i自己。当然是未定义的值。

支持(0) 反对(0)

2013-12-15 21:38 | zdd

## #24楼 回复 引用

@ zdd

我似乎明白了，就是说成员的初始化顺序不是按照初始化列表给的参数顺序，而是按照成员变量在类中的声明顺序来初始化的，也就是说，如果成员初始化列表中的参数合理，初始化列表的顺序是可以随意打乱的。不论我把i放在初始化列表的哪一个位置，i都是第一个被初始化的，其值的大小取决于参数。明白了，谢谢楼主

支持(0) 反对(0)

2013-12-17 08:19 | 李承芳

## #25楼[楼主] 回复 引用

@ 李承芳

[引用](#)

@zdd我似乎明白了，就是说成员的初始化顺序不是按照初始化列表给的参数顺序，而是按照成员变量在类中的声明顺序来初始化的，也就是说，如果成员初始化列表中的参数合理，初始化列表的顺序是可以随意打乱的。不论我把i放在初始化列表的哪一个位置，i都是第一个被初始化的，其值的大小取决于参数。明白了，谢谢楼主

就是这样，不必客气。

支持(0) 反对(0)

2013-12-17 09:07 | zdd

## #26楼[楼主] 回复 引用

@ 李承芳

[引用](#)

@zdd我似乎明白了，就是说成员的初始化顺序不是按照初始化列表给的参数顺序，而是按照成员变量在类中的声明顺序来初始化的，也就是说，如果成员初始化列表中的参数合理，初始化列表的顺序是可以随意打乱的。不论我把i放在初始化列表的哪一个位置，i都是第一个被初始化的，其值的大小取决于参数。明白了，谢谢楼主

所以一个好的编程习惯是，保持成员变量的初始化顺序和他们定义的顺序一致。这样就不会导致混乱了。

支持(0) 反对(0)

2013-12-17 09:09 | zdd

## #27楼 回复 引用

```
struct Test1
```

```
{
```

```
Test1(int a):i(a){}
```

```
int i ;  
};  
  
struct Test2  
{  
    Test1 test1 ;  
    Test2(Test1 &t1)  
    {  
        test1 = t1 ;  
    }  
};
```

以上代码无法通过编译，因为Test2的构造函数中test1 = t1这一行实际上分成两步执行。

1. 调用Test1的默认构造函数来初始化test1
2. 调用Test1的赋值运算符给test1赋值

但是由于Test1没有默认的构造函数，所谓第一步无法执行，故而编译错误。正确的代码如下，使用初始化列表代替赋值操作。

```
struct Test2  
{  
    Test1 test1 ;  
    Test2(Test1 &t1):test1(t1){}  
}
```

改造后的代码也是编译不过的吧，Test1 test1;这个声明是需要调用Test1的无参构造函数的，但是Test1已经没有无参构造函数了，所以这个声明就是编译不过的

支持(0) 反对(0)

2014-01-15 20:35 | 41062947

## #28楼[楼主] 回复 引用

@ 41062947

[引用](#)

```
struct Test1{
    Test1(int a):i(a){}
    int i ;
};
```

```
struct Test2
{
    Test1 test1 ;
    Test2(Test1 &t1)
    {
        test1 = t1 ;
    }
};
```

以上代码无法通过编译，因为Test2的构造函数中test1 = t1这一行实际上分成两步执行。

1. 调用Test1的默认构造函数来初始化test1

## 2. 调用Test1的赋值运算符给test1赋值

但是由于Test1没有默认的构造函数，所谓第一步无法执行，故而编译错误。正确的代码如下，使用初...

建议你先试试再说。

支持(0) 反对(0)

2014-01-16 15:38 | zdd

## #29楼 回复 引用

@ zdd

[引用](#)

@41062947

引用struct Test1

```
{  
Test1(int a):i(a){}  
int i ;  
};
```

struct Test2

```
{  
Test1 test1 ;  
Test2(Test1 &t1)  
{  
test1 = t1 ;  
}  
};
```

以上代码无法通过编译，因为Test2的构造函数中test1 = t1这一行实际上分成两步执行。

1. 调用Test1的默认构造函数来初始化test1

2. 调用Test1的赋值运算符给test1赋值

但是由于Test1没有默认的构造函数，所谓第一步无法执行，故而编译错误。正确的代码如下，使用初...  
建议你先试试再说。

我试过了，能编译通过，但是为什么能通过呢？

支持(0) 反对(0)

2014-01-16 15:50 | 41062947

## #30楼[楼主] 回复 引用

@ 41062947

[引用](#)

@zdd

[引用](#)

引用@41062947

引用struct Test1

```
{  
Test1(int a):i(a){}  
int i ;  
};
```

```
struct Test2
{
    Test1 test1 ;
    Test2(Test1 &t1)
    {
        test1 = t1 ;
    }
};
```

以上代码无法通过编译，因为Test2的构造函数中test1 = t1这一行实际上分成两步执行。

1. 调用Test1的默认构造函数来初始化test1
2. 调用Test1的赋值运算符给test1赋值

但是由于Test1没有默认的构造函数，所谓第一步无法执行，故而编译错误。正确的代码如下，使用初...  
建...

因为使用了初始化列表就不调用构造函数了，而是调用拷贝构造函数呀。

支持(0) 反对(0)

2014-06-24 17:01 | zdd

## #31楼 回复 引用

以上代码无法通过编译，因为Test2的构造函数中test1 = t1这一行实际上分成两步执行。

1. 调用Test1的默认构造函数来初始化test1

## 2. 调用Test1的赋值运算符给test1赋值

楼主，以上描述是否有误？这里并非 `test1 = t1` 这一行分两步执行，而是在这之前编译器就需要调用默认构造函数（因为没有使用初始化列表）来构造`test1`，因为`Test1`没有默认构造函数所以报错。

这也就是你前面提到的初始化阶段，实际上编译器的工作是在用户代码执行之前（也就是函数体当中的代码之前）完成类类型成员的构造，这时因没有默认构造函数而报错，因此还没有执行到`test1 = t1`这一句，应该说`test1`的构造和`test1 = t1`是在两个阶段进行的，而非`test1 = t1`分两步，它只有一步就是调用合成赋值操作符。

假如将`test1 = t1`这一句注释，同样会报相同错误，这也说明错误发生在这一句之前的构造阶段。

《深度探索c++对象模型中》第二章中有关于构造函数较为详细的论述，它指出编译器会在一些情况下扩展我们编写的构造函数，比如编译器会在构造函数函数体中的代码前加入构造类类型成员的代码（调用类类型成员的构造函数）。

初学乍练，从书上识得一些理论，还望指教！

支持(2) 反对(0)

2014-10-03 19:12 | bicying

## #32楼 回复 引用

谢谢 很受教啊

支持(0) 反对(0)

2014-10-08 10:05 | liyusufu

## #33楼[楼主] 回复 引用

@ liyusufu

感谢支持。

支持(0) 反对(0)



---

2014-11-24 10:06 | zdd

### #34楼[楼主] 回复 引用

@ bicying

非常感谢，你是正确的。

支持(0) 反对(0)

---

2014-11-24 10:07 | zdd

### #35楼 回复 引用

楼主，看了这篇文章，我怎么感觉c++各种坑啊。

支持(0) 反对(0)

---

2015-01-05 10:26 | xvmoon

### #36楼[楼主] 回复 引用

@ xvmoon

C++本来就是各种坑啊。

支持(0) 反对(0)

---

2015-01-07 14:51 | zdd

### #37楼 回复 引用

写的很好啊！解答了我不少疑问。

支持(0) 反对(0)

2015-07-14 17:00 | itfanr

## #38楼 回复 引用

@ bicying

### 引用

以上代码无法通过编译，因为Test2的构造函数中test1 = t1这一行实际上分成两步执行。

1. 调用Test1的默认构造函数来初始化test1

2. 调用Test1的赋值运算符给test1赋值

楼主，以上描述是否有误？这里并非 test1 = t1 这一行分两步执行，而是在这之前编译器就需要调用默认构造函数（因为没有使用初始化列表）来构造test1，因为Test1没有默认构造函数所以报错。

这也就是你前面提到的初始化阶段，实际上编译器的工作是在用户代码执行之前（也就是函数体当中的代码之前）完成类类型成员的构造，这时因没有默认构造函数而报错，因此还没有执行到test1...

我也觉得是这样啊，不是很懂的说

支持(0) 反对(0)

2016-03-17 11:22 | 你已在我心里

## #39楼[楼主] 回复 引用

@ bicying

你说的对呀，感谢指正！

我稍后改一下。

支持(0) 反对(0)

2016-03-17 22:44 | zdd

## #40楼[楼主] 回复 引用

@ 你已在我心里

引用

@bicying

引用

引用以上代码无法通过编译，因为Test2的构造函数中test1 = t1这一行实际上分成两步执行。

1. 调用Test1的默认构造函数来初始化test1
2. 调用Test1的赋值运算符给test1赋值

楼主，以上描述是否有误？这里并非 test1 = t1 这一行分两步执行，而是在这之前编译器就需要调用默认构造函数（因为没有使用初始化列表）来构造test1，因为Test1没有默认构造函数所以报错。

这也就是你前面提到的初始化阶段，实际上编译器的工作是在用户代码执行之前（也就是函数体当中的代码之前）完成类类型成员的构造，这时因没有默认构造函数...

说的对，的确是这样的。我改一下。

支持(1) 反对(0)

2016-03-17 22:45 | zdd

## #41楼 回复 引用

31楼说的对。

我就觉得这里有问题，自己也试了，确实不行。看评论找了半天，看到有一样的了。