

个人资料



laibach0304



访问：30372次

积分：562

等级：BLOG > 3

排名：千里之外

原创：21篇 转载：15篇

译文：0篇 评论：11条

文章搜索



文章分类

C# (3)

C++ (23)

C++0x (6)

Other (2)

Reading (3)

文章存档

2007年10月 (2)

2007年09月 (8)

2007年08月 (22)

2007年07月 (2)

2007年02月 (3)

阅读排行

C++ Meta Programming (1467)

boost.pool源码整理和使 (1123)

《C++0x漫谈》系列之： (934)

C++ Meta Programming (908)

boost.shared_ptr源码整 (906)

C++ Meta Programming (898)

智能指针的标准之争：B (872)

【活动】Python创意编程活动开始啦!!! CSDN日报20170428 ——《你的开发为何如此低效?》 深入浅出,带你学习 Unity

C++ Meta Programming 和 Boost MPL(1)

标签：c++ 编程 语言 struct tools yacc

2007-08-30 23:01

899人阅读

评论(1)

收藏 举报

分类：C++ (22)

目录(?)

[+]

本系列全部转载自kuibyshev.bokee.com

1. 引论

C++的发展史是一个不断吸收程序设计领域精华和不断积累充实语言特性的过程。它的创造者Stroustrup在这门新的编程语言草创之初就定下了几个基本的目标，二十年过去了，至今这些目标仍然是C++继续发展的指南针。其中他明确指出，这种语言不应强迫程序员使用单一程序设计形式[20]；就是说C++语言应该是一种“多种花样”的语言。正是因为Stroustrup这种高瞻远瞩的构想，才使得C++逐渐发展成为一门最具综合性、用途最广泛的语言，受到世界上最多的程序员喜爱。也正是因为同样的原因，人们在使用C++的过程中，探索出许多精巧优雅的编程技法，其中不少技法依赖着C++本身丰富的特性，在别种编程语言中并没有对应。例如伴随C++的模板技术而衍生的各种各样的技法，就能生动地说明这一点。

1.1. C++ 模板元编程的发展

1994年的一次C++标准委员会的会议上，Erwin Unruh提交了一个很特别的C++程序，这个程序能够产生一系列素数，但与普通程序最大的不同，运算结果的产生是在编译期完成的，为了显示这些结果，甚至故意产生了一些编译期的错误信息，也就是说，程序从未完成编译过程，却能得到结果[13]。过程如此奇异，实际上是利用了C++模板的可递归特性，或者称为“编译期递归”。

受此启发，Todd Veldhuizen在1995年的一篇文章[27]中提出了“模板元编程（Template Metaprogramming）”这个概念，不但指出模板的特化（Specialization）能力可以用于针对编译期的编程，而且提出了几种实用的控制分支结构。从此C++的编程技术分出一门旁支，C++代码拥有的编译期和运行期的两段执行能力受到人们关注。

Krzysztof Czarnecki与Ulrich W. Eisenecker在1999年出版了《Generative Programming Methods, Tools, and Applications》一书，这本书对模板元编程的贡献是实现了一个辅助库，提出了更完善的控制结构的编写方法。

2004年出版的《C++ Template Metaprogramming: Concepts, Tools, and Techniques from Boost and Beyond》一书归纳了十年来模板元编程的技术动向，成为第一本论述C++模板元编程技术的专著。

模板元编程尽管处于C++程序设计的浪尖，仍然有不少C++程序库的作者直接或间接地利用它，甚至开发出专用于模板元编程的库，当中包括著名的Boost、Loki和Blitz++等。其中Boost提供了MPL（Meta-Programming Library），用以辅助模板元编程，当中提出了许多极有创新性的概念。

目前模板元编程已经成为C++语言的一项庞大的分支，并仍然在迅速发展。许多C++程序员都十分热衷于讨论这门技术，不断提出很多新颖的概念。

1.2. C++模板元编程的概念

元编程（Metaprogramming）的前缀meta-表示“处于较高发展状态”的意思，从这个意义上讲，metaprogramming指的就是高阶的编程，即编写一种程序，使得这种程序可以把某些程序作为输入数据，生成或者操作其他的程序。一般而言，这些高阶的程序运行在编译时。因此，元编程并非一个新概念，事实上任何一种高级语言的编译器都可以视为元编程的工具，因为它们的程序必须被转换为汇编语言或者机器语言以后，才能够运行在计算机上。Lex和Yacc也是很传统的元编程工具，它们能够把一种新的语言根据规范转换成对应的C语言代码。

[一个STL风格的动态二维](#) (856)

[C++ Meta Programming](#) (855)

[C# - Ini文件类](#) (639)

评论排行

[为什么C++](#) (2)

[智能指针的标准之争：Boost vs. Loki](#) (2)

[一个STL风格的动态二维](#) (2)

[你应当如何学习C++\(以及编程\)](#) (2)

[boost.shared_ptr源码整理](#) (1)

[我看国内的C++教育以及我的建议](#) (1)

[C++ Meta Programming](#) (1)

[《C++0x漫谈》系列之：编译选项](#) (0)

[《C++0x漫谈》系列之：智能指针](#) (0)

[C#中const与readonly字](#) (0)

推荐文章

[* CSDN日报20170429 ——《程序修行从“拔刀术”到“万剑诀”》](#)

[* 抓取网易云音乐歌曲热门评论生成词云](#)

[* Android NDK开发之从环境搭建到Demo级十步流](#)

[* 个人的中小型项目前端架构浅谈](#)

[* 基于卷积神经网络\(CNN\)的中文垃圾邮件检测](#)

[* 四无年轻人如何逆袭](#)

最新评论

[C++ Meta Programming 和 Boost code_pipeline](#): 大哥，这个代码错了 template<&t; unsigned n > struct fa...

[为什么C++](#): don: 总觉得刘某就是针对linus在说。

[为什么C++](#): don: C++肯定还是有用的，这点毫无疑问

[我看国内的C++教育以及我的建议](#): coolage31: 多态与虚拟是其核心，这也是所有面向对象的核心，这也是所有面向对象的核心。你的心得总结写的很好~但这两者也真的很难吃透现...

[一个STL风格的动态二维数组](#): laibach0304: 恩，没错 完全原创

[一个STL风格的动态二维数组](#): jxlczjp77: 老兄这时你自己写的吗？看着恐怖，这么长.^_^

[智能指针的标准之争：Boost vs. laibach0304](#): 呵呵,欢迎光临. 我也准备尝试新的东西了,现在还在观望中,但是C++我还是会坚持下去的. 不提马光...

[智能指针的标准之争：Boost vs. guokeno1](#): 我来也! 不错,老兄你研究的东西估计马光志抓狂也弄明白了..... 我现在不怎么用C++了,...

[你应当如何学习C++\(以及编程\)](#): laibach0304: 呵呵，转载的文章，没仔细排版了，见谅

[你应当如何学习C++\(以及编程\)](#): BillEasy: 看起来太累.字体大小格式应该整理下。

C++强大的模板机制赋予了模板在编译时的运算能力。例如以下的代码计算n！，借助于模板的特化能力形成递归，所有运算都在编译时完成：

```
#include
template< unsigned n >
struct factorial
{
    static const unsigned value = n * factorial < 0 > ::value;
};
template<>
struct factorial<0>
{
    static const unsigned value = 1;
};
int main()
{
    std::cout<<factorial < 6 > ::value<<std::endl; 6!="720
}
```

在这个例子里面，factorial在传统意义上是一个类模板，但用元编程的观点去看待，则可个元函数（Metafunction）。这个元函数接受一个整型的模板参数，它的返回值是元函数的一个静态成员。注意元函数与普通的函数很不一样，一方面，元函数往往需要定义一个以上的特化版本作为终结条件，另一方面，元函数的返回值需要显式地提取出来。这个简单的程序似乎仅仅是玩弄技巧，但正如1994年Erwin Unruh所演示的那个载入史册的素数生成程序，它们其实都只是引起人们注意的冰山一角，为了深入研究模板元编程的能力、使用范围和局限性，必须清楚了解其背后的机理。

顶

0

踩

0

上一篇

智能指针的标准之争：Boost vs. Loki

下一篇

C++ Meta Programming 和 Boost MPL(2)

我的同类文章

C++ (22)

• boost.tuple源码整理和使用...

2007-10-07

阅读 417

• 泛型归并排序

2007-09-18

阅读 303

• C++ Meta Programming 和 ...

2007-08-30

阅读 908

• C++ Meta Programming 和 ...

2007-08-30

阅读 856

• 我看国内的C++教育以及我...

2007-08-28

阅读 602

• 泛型插入排序

2007-09-18

阅读 350

• 为什么C++

2007-09-18

阅读 438

• C++ Meta Programming 和 ...

2007-08-30

阅读 1467

• 智能指针的标准之争：Boost...

2007-08-28

阅读 872

• 泛型快速排序

2007-08-28

阅读 448

更多文章

关闭



猜你在找

C++程序设计

深入浅出C++程序设计（基础篇）

《C语言/C++学习指南》Linux开发篇

深入实践Boost1 获取配置选项方法c++

推荐使用C++ TR1和boost库

C++知识点---explicitTR1Boost

其他人的blog

[刘未鹏|C++的罗浮宫](#)[马维达](#)[荣耀](#)[yacht](#)[电池管理系统](#)[特色结婚照](#)[开发一个app](#)[免费结婚的游](#)[性价比手机排](#)[android睡眠](#)[清扫机](#)[查看评论](#)1楼 [code_pipeline](#) 2007-11-09 16:48发表

大哥，这个代码错了

```
template< unsigned n >
```

```
struct factorial
```

```
{
```

```
static const unsigned value = n * factorial < 0 > ::value;
```

```
};
```

```
#####
```

```
应该是value = n * factorial < n-1 > ::value;
```

[您还没有登录,请\[登录\]或\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

[全部主题](#) [Hadoop](#) [AWS](#) [移动游戏](#) [Java](#) [Android](#) [iOS](#) [Swift](#) [智能硬件](#) [Docker](#) [OpenStack](#)
[VPN](#) [Spark](#) [ERP](#) [IE10](#) [Eclipse](#) [CRM](#) [JavaScript](#) [数据库](#) [Ubuntu](#) [NFC](#) [WAP](#) [jQuery](#)
[BI](#) [HTML5](#) [Spring](#) [Apache](#) [.NET](#) [API](#) [HTML](#) [SDK](#) [IIS](#) [Fedora](#) [XML](#) [LBS](#) [Unity](#)
[Splashtop](#) [UML](#) [components](#) [Windows Mobile](#) [Rails](#) [QEMU](#) [KDE](#) [Cassandra](#) [CloudStack](#) [FTC](#)
[coremail](#) [OPhone](#) [CouchBase](#) [云计算](#) [iOS6](#) [Rackspace](#) [Web App](#) [SpringSide](#) [Maemo](#)
[Compuware](#) [大数据](#) [aptech](#) [Perl](#) [Tornado](#) [Ruby](#) [Hibernate](#) [ThinkPHP](#) [HBase](#) [Pure](#) [Solr](#)
[Angular](#) [Cloud Foundry](#) [Redis](#) [Scala](#) [Django](#) [Bootstrap](#)

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)[网站客服](#) [杂志客服](#) [微博客服](#) [webmaster@csdn.net](#) 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 |

江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2016, CSDN.NET, All Rights Reserved

关闭