<<Imperfect C++中文版>>

图书基本信息

书名：<<Imperfect C++中文版>>

13位ISBN编号：9787115277978

10位ISBN编号：7115277974

出版时间：2012-6-19

出版时间：人民邮电出版社

作者：[美]Matthew Wilson

页数：591

字数：875000

译者：荣耀,刘未鹏

前言

<<Imperfect C++中文版>>

内容概要

即便是C++阵营里最忠实的信徒，也不得不承认：C++语言并不完美。
实际上，世界上也没有完美的编程语言。

如何克服C++类型系统的不足？
在C++中，如何利用约束、契约和断言来实施软件设计？
如何处理被C++标准所忽略的动态库、静态对象以及线程等有关的问题？
隐式转换有何替代方案？
《Imperfect C++中文版》将为你一一解答这些问题。
针对C++的每一个不完美之处，《Imperfect C++中文版》都具体地分析原因，并探讨实用的解决方案。
书中也不乏许多作者创新的、你从未听说过或使用的技术，但这些确实能帮助你成为C++方面的专家。

《Imperfect C++中文版》适合有一定经验的C++程序员和项目经理阅读，也适合对C++编程的一些专门或高级话题感兴趣的读者参考。

作者简介

作者：（美国）Matthew Wilson 译者：荣耀 刘未鹏Matthew Wilson是一名软件开发顾问，也是STLSoft库的创建者。
他为双月刊C/C++ Users Journal撰写关于将C/C++与其他语言和技术进行整合的专栏文章，同时也是C++ Experts Forum在线专栏作家。
Wilson有十余年C++开发经验。
荣耀，南京师范大学教师。
他是一名C++讲师和研究者，译有《C++必知必会》、《C++ Templates全览》以及《C++ Template Metaprogramming中文版》（中文繁体版）等，并在期刊杂志上发表过多篇文章。
他原任电力自动化研究院工程师与项目经理，是数个企业级信息系统项目负责人。
刘未鹏，南京大学计算机系硕士毕业，现就职于微软亚洲研究院创新工程中心。

<<Imperfect C++中文版>>

书籍目录

章节摘录

版权页： 插图： 如果你没有任何COM背景的话，上面的代码对你来说可能会相当陌生，但事实上它是相当直观的。
它之所以能够工作是因为从根本上说类也只不过是个结构，并且，如果该类定义了虚函数（或者派生自某个定义了虚函数的类）的话，那么其实例将会包含一个隐藏的vptr（指向vtable的指针）。
vptr指向一个表，该表中包含了指向该类的所有虚函数的指针，因而被称为虚函数表（vtable）。
一般情况下，同一个类的所有实例共享同一份vtable。
在上面的C代码中，vtable是以结构IObjectVTable来表现的，它包含了指向SetName（）和GetName（）函数的指针，这种函数指针跟其他的函数指针没什么两样，只不过其第一个参数总是指向该接口的指针，即C++中的this指针。
struct IObject中只有一个成员vtable，它是一个指针，指向该接口的虚函数表，即structIObjectVTable的一个实例。
正如我前面说过的，我们的6个Win32编译器在这个布局方面完全一致。
或许Win32编译器支持这种布局并非巧合，因为这种布局正是COM所使用的，而Win32编译器要想得到流行的话必须支持COM。
尽管我们承认这当然是一个明显而高效的对象布局模型，然而确实存在着一些编译器不这么做。
一个不支持这种布局的Win32编译器就是GCC2.95（GCC3.2支持）。
经过一些“不足为外人道”的手法，探知GCC2.95使用的vtable布局是像这样的：v1、v3和v4的值是0，因此我猜测这与pack问题有关，v2看起来是指向某个地址跟SetName（）和GetName（）很靠近的函数，但我不知道这个函数的精确特征。
如此“特立独行”的并非仅GCC2.95一家，Sun的C++编译器1使用类似下面的这种布局：所以我们现在有了一个不太理想的选择。
选择之一是采用这种部分的解决方案，至少Win32上的现代编译器看起来都支持这种方案。
如果在其他平台上试一下的话，可能也会发现类似的一致布局的情况，对此我们可以采取同样的立场。
且慢！
我们是不完美主义的实践者，并且这种做法远没达到预期的程度。
我们需要寻找一种彻底的解决方案。
8.1.2动态操纵虚函数表 在我们试图设计出一个完全可移植的方案之前，我暂且充当一次不负责任的角色，向你展示一些“奇技淫巧”，借此你可以对C++对象布局动点手术。
可能你会觉得这里讲的这些东西太高深，并担心是否必须自己定义这些C虚函数表。
通常你不必这么做，C++编译器会为你打理好一切。
干预任何C++编译器生成的类的虚函数表都不是明智之举，因为你对它的任何干预都会在整进程的生命期中反映在该类的所有实例身上。
1但是在c里面你完全可以操纵你自己的虚函数表，这种能力在某些非常罕见的场合下可以被用来改变对象的运行期特征（行为）。
我并不建议你采取这种做法，它的主要作用在于可以学习c++实现相关的机理，但是作为用在产品软件中的技术却不大合适。
不过，知道它是怎么做的倒也不算坏事。
基本上，这可以分为3个步骤。
首先，你必须分配自己的虚函数表。
这在C里面实现，只不过是简单地分配一份可用于保存虚函数表内容的内存块而已。
其次，你得从一个有效的对象中将虚函数表拷贝出来，这也是用c来实现的，只不过是在c++编译单元中创建的对象身上进行操作。
最后，你可以改变你的新虚函数表中的成员，并将这个新的虚函数表挂到你要改变其行为的对象身上，这仍然是在C里面实现的，但目标对象可以是c++编译单元中创建的对象。

所有这3个步骤可以被包装在同一个函数中。

媒体关注与评论

“千万不要被书名所误导！
这是一本拥抱（而非诋毁）C++的著作。
它有着独特的定位：为现实世界中的程序员提供切合实际的解决方案，以解决C++语言自身的各种“不完美”。
世界上没有完美的编程语言。
在本书中，Matthew Wilson不但为我们指出C++中诸多不完美之处，还提供了经过实践检验的应对技术和技巧，便于我们利用“不完美的C++”编写出近乎完美的代码——强健、高效、灵活、可移植、优雅的代码，而这些代码在声称为“完美的语言”中往往更难实现。
本书对给出的每一个“不完美”都进行了细致的探讨：为什么说它是一个“不完美”？
对其修复的指导思想是什么？
有时候只是告诫你避免做些什么，给出一些约束和建议，更多的时候则为你提供现实的解决方案，这些方案往往离不开对现代模板编程技术的使用。
书中包含有许多你未曾听过或用过的技术，有些属于作者的创新，有些则是对现有技术的精化，二者均被提升到“范式”的高度。
例如：应用程序二进制接口（ABI）、垫片（Shim）、饰面（Veneer）、螺栓（Bolt-in）、区间（range）、属性（property）等。
不少主题难度大，此前为其他C++专家所忽略。
要探讨它们除了需要勇气外，第一手经验更是不可或缺。
作为STLSoft库的主创者，Matthew在举例时，对Windows API、MFC、ATL、COM以及UNIX等都是信手拈来。
除了丰富的实践、扎实的理论以及缜密的逻辑外，Matthew的文笔流畅，语言幽默，说理直接，字里行间流露出过人的自信，使得本书极具阅读趣味。
本书具有一定的阅读门槛，目标读者为中、高级职业C++程序员。
书中展示的代码示例、编程技术往往在几款甚至十几款编译器上进行验证，辅以表格对其各色特性加以比较，并针对不同编译器所表现出的差异性而给出高效、可移植的解决方案——就像很多现实世界中的C++程序员应该做（而没做到）的那样。
如果你正在寻找一本真材实料的“C++实战”参考书，本书不会让你失望。
本书中文版由我和刘未鹏先生合译。
未鹏思维敏捷，技术、文笔俱佳，我很高兴与他合作。
感谢陈冀康编辑给予的理解和支持。
感谢朱艳的照料和热爱。
荣珅则常常用他的小拳头乱砸书房的门，并大声地叫“爸爸”，这种干扰让我获得了必不可少的休息时间。
已有的经典名著使得C++新书问世难度加大，后来者若无过人之处就很难引起C++社群的注意，Imperfect C++、C++ Common Knowledge以及C++ Template Metaprogramming等佳作一经问世便得到广泛的关注。
作为译者（之一），我祝愿它们能够带给各位久违的快乐！
——荣耀　刀有很多种，有单刀，双刀，朴刀，戒刀，锯齿刀，砍山刀，鬼头刀，雁翎刀，五凤朝阳刀，鱼鳞紫金刀。
—— 古龙《飞刀，又见飞刀》　这里我们要说的刀，是瑞士军刀，瑞士军刀其实严格来说并不能算是一种刀，其功能的繁杂和精细已然超过了刀的范畴。
它包含的工具一般有主刀、小刀、剪刀、开瓶器、木锯、小改锥、拔木塞钻、牙签、小镊子等，而在一些工具上还设计了多种功用，如开瓶器上，就具有开瓶、平口改锥、电线剥皮槽3种功用。
随着时代的发展，一些新兴的电子技术也被引入瑞士军刀中，如内藏激光笔、电筒等。

瑞士军刀是军人在野外生存的必备工具，其小体积浓缩众多实用功能的精心设计能够将一把刀的容限发挥到最大，丝毫不逊于《第一滴血》中蓝博带在身上的那把锐利的寒光闪闪的钢刀。

那么现在你拿在手里的这本书就是一把瑞士军刀！

这是一本非常特别的C++图书，在市面上已经存在的大量经典C++书籍当中，这本书的着眼点和写作风格使它显得那么特立独行和标新立异，甚至有点另类。

书中几乎巨细靡遗地涵盖了C++中大大小小的不完美之处，并以一系列成功案例证明C++的确不完美同时也提供了迂回之道、解决之道，再加上其用本主义的立场，正如同一把实用的瑞士军刀，功能繁杂而面面俱到，实用之至。

同其他C++著作不一样，本书虽然尊重标准，但同时又超越标准，当标准不能满足需求或成为拦路石的时候，需求才是第一位的，于是有了作者所谓的“不完美主义的实践者”以及“不完美工具箱”之说。

此外，作者的所谓“苦行僧式编程”哲学在我看来也是极其实用的一种编码方式！

我们以前看到的绝大部分C++书籍可说是统统走的“阳关大道”，然而Matthew这本书却偏要走他的“独木小桥”，蹊径虽小，然则别有一番风味和景观。

我们意识到原来C++中也存在着如此多大大小小的不完美之处，就像宫崎俊电影中的那些打满补丁的海盗飞机一样。

Bjarne本就说过，C++是为“用本”而设计的，诚然！

而本书最大的趣味就在于它并不去一味抱怨这些缺点，而是积极地采取其他替代方案来达到同样的目的，并借此展现出C++自由强大的一面！

作者Matthew常用“survive”一词来描述在编码的现实世界中的境况，作为STLSoft库的主要编写者，他十几年来积累的经验在书中充盈四溢，很多我们平常看不到的方面都会被他挑出来，甚至连我这个译者都觉得有点“啰嗦”。

不过，对于喜欢他这种“唠叨”讲法的人，他那种辨证的严密论证法倒是能令你获益颇多。

另外，书中随处可见具有作者个人特色的幽默，在大量平淡无奇的技术书籍当中可算是一个亮点。

本书的一个小缺憾就是它不适合初学者，某些地方甚至对于中级读者来说都有一定的难度，作者自己经验非常丰富，因此有些地方就不加解释地一带而过，为此译者适当添加了一些译注，以便读者理解和阅读。

最后，感谢荣耀先生在本书初译的过程中一直给予的支持和信任，并容忍我总是延期交付各章译稿。

荣耀先生对技术的精益求精和一丝不苟也令我在翻译的过程中获益良多。

最大的感激要归于我的父母和我的爷爷，感谢他们一直以来对我的追求的支持和鼓励，没有他们我无法想象能够完成这项工作。

希望这本令我在翻译过程中获益匪浅的书也能够给你带来美妙而独一无二的阅读享受，Let’s dig in！

——刘未鹏

编辑推荐

《Imperfect C++中文版》是市面上唯一一本讨论C++的不足之处，并给出解决方案的C++经典图书，荣获Amazon五星级评价。
由知名译者荣耀与刘未鹏合译，在技术开发者中拥有极大影响力。

名人推荐

千万不要被书名所误导！
这是一本拥抱（而非诋毁）C++的著作。
它有着独特的定位：为现实世界中的程序员提供切合实际的解决方案，以解决C++语言自身的各种“不完美”。
世界上没有完美的编程语言。
在本书中，Matthew Wilson不但为我们指出C++中诸多不完美之处，还提供了经过实践检验的应对技术和技巧，便于我们利用“不完美的C++”编写出近乎完美的代码——强健、高效、灵活、可移植、优雅的代码，而这些代码在声称为“完美的语言”中往往更难实现。
本书对给出的每一个“不完美”都进行了细致的探讨：为什么说它是一个“不完美”？
对其修复的指导思想是什么？
有时候只是告诫你避免做些什么，给出一些约束和建议，更多的时候则为你提供现实的解决方案，这些方案往往离不开对现代模板编程技术的使用。
书中包含有许多你未曾听过或用过的技术，有些属于作者的创新，有些则是对现有技术的精化，二者均被提升到“范式”的高度。
例如：应用程序二进制接口（ABI）、垫片（Shim）、饰面（Veneer）、螺栓（Bolt-in）、区间（range）、属性（property）等。
不少主题难度大，此前为其他C++专家所忽略。
要探讨它们除了需要勇气外，第一手经验更是不可或缺。
作为STLSoft库的主创者，Matthew在举例时，对Windows API、MFC、ATL、COM以及UNIX等都是信手拈来。
除了丰富的实践、扎实的理论以及缜密的逻辑外，Matthew的文笔流畅，语言幽默，说理直接，字里行间流露出过人的自信，使得本书极具阅读趣味。
本书具有一定的阅读门槛，目标读者为中、高级职业C++程序员。
书中展示的代码示例、编程技术往往在几款甚至十几款编译器上进行验证，辅以表格对其各色特性加以比较，并针对不同编译器所表现出的差异性而给出高效、可移植的解决方案——就像很多现实世界中的C++程序员应该做（而没做到）的那样。
如果你正在寻找一本真材实料的“C++实战”参考书，本书不会让你失望。
本书中文版由我和刘未鹏先生合译。
未鹏思维敏捷，技术、文笔俱佳，我很高兴与他合作。
感谢陈冀康编辑给予的理解和支持。
感谢朱艳的照料和热爱。
荣珅则常常用他的小拳头乱砸书房的门，并大声地叫“爸爸”，这种干扰让我获得了必不可少的休息时间。
已有的经典名著使得C++新书问世难度加大，后来者若无过人之处就很难引起C++社群的注意，Imperfect C++、C++ Common Knowledge以及C++ Template Metaprogramming等佳作一经问世便得到广泛的关注。
作为译者（之一），我祝愿它们能够带给各位久违的快乐！
——荣耀刀有很多种，有单刀，双刀，朴刀，戒刀，锯齿刀，砍山刀，鬼头刀，雁翎刀，五凤朝阳刀，鱼鳞紫金刀。
——古龙《飞刀，又见飞刀》这里我们要说的刀，是瑞士军刀，瑞士军刀其实严格来说并不能算是一种刀，其功能的繁杂和精细已然超过了刀的范畴。
它包含的工具一般有主刀、小刀、剪刀、开瓶器、木锯、小改锥、拔木塞钻、牙签、小镊子等，而在一些工具上还设计了多种功用，如开瓶器上，就具有开瓶、平口改锥、电线剥皮槽3种功用。
随着时代的发展，一些新兴的电子技术也被引入瑞士军刀中，如内藏激光笔、电筒等。
瑞士军刀是军人在野外生存的必备工具，其小体积浓缩众多实用功能的精心设计能够将一把刀的容限

发挥到最大，丝毫不逊于《第一滴血》中蓝博带在身上的那把锐利的寒光闪闪的钢刀。
那么现在你拿在手里的这本书就是一把瑞士军刀！
这是一本非常特别的C++图书，在市面上已经存在的大量经典C++书籍当中，这本书的着眼点和写作风格使它显得那么特立独行和标新立异，甚至有点另类。
书中几乎巨细靡遗地涵盖了C++中大大小小的不完美之处，并以一系列成功案例证明C++的确不完美同时也提供了迂回之道、解决之道，再加上其用本主义的立场，正如同一把实用的瑞士军刀，功能繁杂而面面俱到，实用之至。
同其他C++著作不一样，本书虽然尊重标准，但同时又超越标准，当标准不能满足需求或成为拦路石的时候，需求才是第一位的，于是有了作者所谓的“不完美主义的实践者”以及“不完美工具箱”之说。
此外，作者的所谓“苦行僧式编程”哲学在我看来也是极其实用的一种编码方式！
我们以前看到的绝大部分C++书籍可说是统统走的“阳关大道”，然而Matthew这本书却偏要走他的“独木小桥”，蹊径虽小，然则别有一番风味和景观。
我们意识到原来C++中也存在着如此多大大小小的不完美之处，就像宫崎俊电影中的那些打满补丁的海盗飞机一样。
Bjarne本就说过，C++是为“用本”而设计的，诚然！
而本书最大的趣味就在于它并不去一味抱怨这些缺点，而是积极地采取其他替代方案来达到同样的目的，并借此展现出C++自由强大的一面！
作者Matthew常用“survive”一词来描述在编码的现实世界中的境况，作为STLSoft库的主要编写者，他十几年来积累的经验在书中充盈四溢，很多我们平常看不到的方面都会被他挑出来，甚至连我这个译者都觉得有点“啰嗦”。
不过，对于喜欢他这种“唠叨”讲法的人，他那种辨证的严密论证法倒是能令你获益颇多。
另外，书中随处可见具有作者个人特色的幽默，在大量平淡无奇的技术书籍当中可算是一个亮点。
本书的一个小缺憾就是它不适合初学者，某些地方甚至对于中级读者来说都有一定的难度，作者自己经验非常丰富，因此有些地方就不加解释地一带而过，为此译者适当添加了一些译注，以便读者理解和阅读。
最后，感谢荣耀先生在本书初译的过程中一直给予的支持和信任，并容忍我总是延期交付各章译稿。
荣耀先生对技术的精益求精和一丝不苟也令我在翻译的过程中获益良多。
最大的感激要归于我的父母和我的爷爷，感谢他们一直以来对我的追求的支持和鼓励，没有他们我无法想象能够完成这项工作。
希望这本令我在翻译过程中获益匪浅的书也能够给你带来美妙而独一无二的阅读享受，Let’s dig in！
——刘未鹏

<<Imperfect C++中文版>>

版权说明