

<<C++程序设计>>

图书基本信息

书名：<<C++程序设计>>

13位ISBN编号：9787302198215

10位ISBN编号：7302198217

出版时间：2009-7

出版时间：清华大学出版社

作者：叶焕倬 编

页数：359

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<C++程序设计>>

前言

C++语言是一种使用非常广泛的计算机编程语言。它支持过程化程序设计、面向对象程序设计和泛型程序设计等多种程序设计风格。另外，C++语言虽然从C语言发展而来，但它并不依赖C语言，我们可以完全不学C语言，而直接学习C++语言，C++语言可以作为程序设计的入门语言。

C++语言的发展大概可以分为三个阶段：第一阶段从20世纪80年代到1995年。这一阶段C++语言基本上是传统类型上的面向对象语言，并且凭借着接近C语言的效率，在工业界使用的开发语言中占据了相当大的份额。

第二阶段从1995年到2000年。

这一阶段由于标准模板库（Standard Template Library, STL）和后来的Boost等程序库的出现，泛型程序设计的C++中占据了越来越多的比重。

当然，同时由于Java、C#等语言的出现和硬件价格的大规模下降，C++受到了一定的冲击。

第三阶段从2000年至今。

由于以Loki、MPL等程序库为代表的产生式编程和模板元编程的出现，C++出现了发展历史上的又一个新高峰，这些新技术的出现以及和原有技术的融合，使C++已经成为当今主流程序设计语言中最复杂的一员。

我们目前所用的相当多数软件都是用C++语言编写的。

主流的三种操作系统Windows、Linux和UNIX的内核都是用C语言和汇编语言编写的，上层高级特性是用C++语言编写的。

“魔兽世界”、CS等几乎所有的网络游戏以及百度搜索引擎都是用C++语言编写的。

C++语言的优势及魅力不言而喻。

本书编者综合了教学及软件设计经验，使得本书既具有较强的理论性，又具有较强的实用性。

本书编写的目的就是希望读者能够编写出规范的、稳定的程序。

它不仅仅是一本纯粹的C++语言的语法书，书中很多语法是通过实例引出的，使读者可以在有趣、高效的应用中领悟枯燥的语法。

为了巩固其中灵活、难解的语法知识，每章都有配套习题。

另外，为了让读者能够写出规范的程序，配套有编程基本规范的内容；为了让读者能够写出稳定的程序，配套有程序调试基本方法的内容。

看似内容庞大，但本书的可读性强，每个核心内容力求言简意赅，尽量用应用实例来进行解说。

总的来说，本书信息量大，综合面广，实用性强，可读性好，在同类教材中有很鲜明的特色。

全书共分为24章。

其中，前20章是本书的主体，第1~第9章以面向过程为基础介绍C++的基本语法和数据类型，第10~第20章介绍面向对象知识以及C++面向对象的程序设计特征和方法，这些章节都附有习题，部分章节还附有应用实例。

本书的最后4章介绍了使用C++程序设计必备的其他知识，第21章介绍了Microsoft公司推出的Visual C++集成开发环境。

Visual C++集成开发环境提供了建立控制台应用程序的方法，为学习和掌握C++语言提供了非常好的开发环境，同时，它又很好地利用了C++语言，并提供了大量的MFC（Microsoft Foundation Class）基础类库、应用程序向导和“可视化”的资源编辑器，这为程序员快速而高效地开发出Windows应用程序提供了方便。

考虑到将本书作为“Windows编程”课程的参考书，第22章介绍了怎样基于MFC开发Windows应用程序。

为了做到知识的全面性和实用性，第23章介绍了编程的基本规范，第24章介绍了调试程序的基本方法。

。

本书内容全面，不仅适合于教学，也适合于用Visual C++开发Windows应用程序的用户参考。如果作为教材，建议将第23章、第24章在恰当的时候穿插在前面章节讲解。

本书由叶焕倬主编，向华丽策划，向华丽、吴良霞、周晓华、程传慧和鲁敏等老师参与了初稿的编写。

全书由叶焕倬统稿，叶焕倬、屈振新、鲁敏、胡雨林审稿，高芳、何玉凝参与了校对，张露露和鲁敏参与了排版和统一格式的工作。

清华大学出版社的员工为本书的完成付出了大量的劳动，在此特致谢意。

由于编者水平有限，书中不当、错误之处在所难免，敬请读者批评指正。

<<C++程序设计>>

内容概要

本书以程序设计的初学者为主要对象，系统地介绍了C++程序设计语言的特性，既有面向过程的知识，又有面向对象的内容。

本书除备有习题供读者练习外，还专设章节讲解Visual C++集成开发环境、MFC编程、编程规范和调试技巧，对读者熟练掌握和正确使用C++有很好的帮助。

本书可以作为计算机相关专业程序设计语言课程的教材，也可以供其他理工科专业的程序设计课程使用，还可以满足C++自学者的需要。

<<C++程序设计>>

书籍目录

第1章 C++程序的基本组成第2章 C++的数据类型、运算符和表达式第3章 数据的输入与输出第4章 C++的控制语句第5章 数组及其使用第6章 函数第7章 指针第8章 引用第9章 结构体、公用体和枚举第10章 从面向过程的程序设计到面向对象的程序设计第11章 类与对象第12章 构造函数与析构函数第13章 继承与派生第14章 虚函数与多态性第15章 静态成员第16章 友元函数与友元类第17章 堆对象和对象数组第18章 运算符的重载第19章 模板与使用第20章 文件及其操作第21章 Visual C++集成开发环境第22章 基于MFC开发Windows应用程序第23章 编程基本规范第24章 程序调试的基本方法参考文献

章节摘录

在设置函数的默认参数值时要注意如下问题。

(1) 当函数既有声明又有定义时，不能再在函数定义中指定默认参数。

(2) 当一个函数中有多个默认参数时，则形参分布中，默认参数应从右到左逐渐定义。

在调用函数时，系统按从左到右的顺序将实参与形参结合，当实参的数目不足时，系统将按同样的顺序用声明或定义中的默认值来补齐所缺少的参数。

6.7 内联函数 在函数调用与返回的过程中，需要付出一定的时间与空间代价用于处理现场。当函数较小，又反复使用时，处理现场的开销会急剧增大。

若把函数嵌入函数调用处，便可以大大提高运行速度，节省开销。

内联函数就可以自动实现这一功能。

内联函数不是在调用时发生控制转移，而是在编译时将函数体嵌入在每一个调用语句处。

语法形式： `inline` 类型说明符 被调函数名 (含类型说明的形参表) {函数体} 注意：

(1) 内联函数体内一般不能有循环语句和 `switch` 语句。

(2) 内联函数的定义必须出现在第一次被调用之前，不能按先声明内联函数原型，再调用、再定义的次序使用。

版权说明

本站所提供下载的PDF图书仅提供预览和简介, 请支持正版图书。

更多资源请访问:<http://www.tushu007.com>