

<<高级语言程序设计教程>>

图书基本信息

书名：<<高级语言程序设计教程>>

13位ISBN编号：9787811355406

10位ISBN编号：781135540X

出版时间：2010-9

出版时间：暨南大学出版社

作者：宋桂琴

页数：253

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>

<<高级语言程序设计教程>>

前言

在过去的30年中，C语言以具有丰富的运算符和数据类型、使用灵活、代码重用性高而广受人们的喜爱。

C++语言是一种优秀的程序设计语言，它在C语言的基础上发展而来，包含了C语言的所有内容，同时又支持面向对象的程序设计。

现在，大多数高校都把C / C++作为主要的教学语言。

本书是作者根据多年的教学经验，在听取同事意见，并对国内外同类著作和教材进行深入的比较研究后写成的。

本书先介绍了程序设计的基本概念，接着以Visualc++6.0作为实验开发平台，通过大量精选的案例，全面系统地介绍了C语言基本数据类型、运算符、表达式、数据的输入输出、分支和循环控制结构、数组、函数、指针、结构体、枚举、编译预处理等内容。

为使读者在熟悉了C语言的基础上，能轻松地过渡到C++语言的学习，本书在把C语言独立成一个部分进行介绍后，进一步介绍了C++语言，主要包括面向对象的程序设计方法、C++语言中类和对象的概念、继承性与派生类、运算符重载机制、多态性的用法、模板和输入输出流。

学习程序设计一定要有足够的耐心，绝不能浮躁。

在设计程序时，要让自己“钻”到程序里，在程序中浸泡自己。

同时一定要规范地书写代码，以提高程序的可读性。

除了自己要多动手编写代码上机调试外，阅读并评价别人的代码也是提高自己编程能力的一个有效途径。

另外，在学习计算机语言时，不能局限于一本参考书，要尽可能多看一些有关书籍，同时自己要多思考、多研究。

本书既可以作为计算机专业本科生程序设计课程的教材，也可以作为学习C / C++语言的入门参考书。

本书在编写过程中特别是在目录、内容的编排上得到同事的极大帮助，在此特向他们表示衷心的感谢。

同时，本书参考了大量相关文献，并引用了其中的一些案例和内容，在此，对这些文献的作者表示最真挚的谢意。

最后衷心感谢暨南大学出版社为了此书的出版而付出辛勤劳动的各位编辑。

因作者水平有限，书中难免会存在一些不当之处，真心地请求读者批评指正。

<<高级语言程序设计教程>>

内容概要

本书先介绍了程序设计的基本概念，接着以Visualc++6.0作为实验开发平台，通过大量精选的案例，全面系统地介绍了C语言基本数据类型、运算符、表达式、数据的输入输出、分支和循环控制结构、数组、函数、指针、结构体、枚举、编译预处理等内容。

<<高级语言程序设计教程>>

书籍目录

前言 解题和程序设计概要	1 计算机解决问题的初级概念	1.1 问题求解的一般概念	1.2
计算机解决的问题	2 程序设计的概念	2.1 算法和程序	2.2 程序设计方法 2.3 程序设计语言
2.4 计算机如何存储数据	程序语言基础(C语言)	3 数据类型与基本输入输出	3.1
概述	3.2 标识符	3.3 数据类型	3.4 变量和常量的定义
3.5 运算符和表达式	3.6 类型转换	3.7 简单的输入输出	3.8应用举例
4 程序控制结构	4.1 C语言语句概述	4.2 C程序的基本结构	4.3 分支语句(选择语句)
4.4 循环语句	4.5 应用举例	5 复合数据类型	5.1 数组
5.2 指针	5.3 数组与指针	5.4 动态内存分配和动态数组	5.5 字符数组与字符串
5.6 结构类型	5.7 枚举类型	6 函数	6.1 概述
6.2 函数的定义、调用、原型	6.3 函数的调用方式和返回值	6.4 函数的递归调用	6.5 内联函数和重载函数
6.6 作用域和存储类型	6.7 预处理命令	面向对象程序设计(c++语言)	7
面向对象程序设计方法和思想	7.1 结构化程序设计方法中存在的问题	7.2 面向对象方法	7.3 面向对象程序设计语言
7.4 面向对象的基本概念	8 类与对象	8.1 类的定义	8.2
对象	8.3 对象的初始化	8.4 静态成员	8.5 常成员与常对象
8.6 友元和友元函数	8.7 复合类	8.8 string类	9 继承与派生
9.1 继承的概念与形式	9.2 派生类	9.3 派生类的构造函数和析构函数	9.4 继承成员的调整
9.5 多重继承	9.6 重复继承	10 多态性与虚函数	10.1 静态联编和动态联编
10.2 类型兼容性	10.3 虚函数	10.4 纯虚函数和抽象类	10.5 应用案例
11 运算符重载	11.1 运算符重载的基本方法	11.2 运算符重载为类的成员函数	11.3 运算符重载为类的友元函数
11.4 应用案例	12 模板和命名空间	12.1 模板	12.2 标准模板库
12.3 命名空间	13 输入输出流	13.1 概述	13.2 C++的流类库
13.3 格式化输入输出	13.4 检测流操作的错误	13.5 磁盘文件的读写	13.6 字符串流string
14 异常处理	14.1 概述	14.2 C++语言中的异常处理	14.3 带有异常说明的函数原型
14.4 创建对象时的异常处理	附录	附录A C / C++语言关键字	附录B ASCII码表
附录C C / C++语言常用标准函数参考文献			

章节摘录

程序设计方法是指组织程序内部数据和逻辑所用的方法。

随着计算机应用的不断普及，程序规模随之越来越大，程序设计的目标也由原来追求程序的高效率转变为对程序的可读性、可靠性、移植性、重用性、可维护性的综合考虑，这促进了程序设计方法的发展。

2.2.1 结构化程序设计 早期的程序设计由于受硬件技术的限制，所以没有成型的设计方法，主要依赖个人技术和经验编程，因此程序的可读性、可维护性和重用性都很差。

随着计算机硬件技术和其他相关信息技术的发展，同时也为了能够编写和维护复杂庞大的程序，在20世纪60年代出现了结构化程序设计方法，又称面向过程的程序设计方法。

结构化程序设计强调程序设计风格和程序结构的规范化，提倡清晰的结构。

结构化程序设计方法的基本思路是：把一个复杂问题的求解过程分阶段进行。

具体方法包括：自顶向下；逐步细化；模块化设计；结构化编码。

自顶向下是一种问题分解技术，将复杂的问题分解为一系列复杂性相对较低的子问题，然后逐个解决这些子问题，从而使整个问题得到解决；逐步细化是指对问题按层次进行分解，每一层不断将子问题细化，到了最后一层所有问题都是简单易解决的小问题；模块化设计是指将大程序划分成若干个子程序，每个子程序称为一个模块；结构化编码是指用结构化的计算机语言编写程序代码。

结构化程序设计是对程序的功能进行分解，将数据和对数据的处理过程分开，以过程为中心设计程序。

因为用结构化程序设计方法设计的程序耦合度过高，所以这必然降低程序的安全性和重用性。

例如，过程1和过程2都要操作数据A（如图2-2所示），当数据A的结构改变时，过程1和过程2都要进行相应的修改；当过程1修改时，可能会引起过程2的修改。

而且每一种相对于老问题的新方法都要带来额外的开销。

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>