

<<面向对象程序设计>>

图书基本信息

书名：<<面向对象程序设计>>

13位ISBN编号：9787560924113

10位ISBN编号：7560924115

出版时间：2001年1月1日

出版时间：第1版 (2001年1月1日)

作者：刘正林

页数：548

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>

<<面向对象程序设计>>

内容概要

本书以最新ANSI C++标准为准则，以美国Microsoft公司开发的Visual C++ 6.0为语言蓝本，全面、系统、完整地讲授面向对象程序设计方法的核心概念、主要语言特性、思维方式和面向对象程序设计技术。

全书分成两大部分，第一部分是基础部分，可作为目前各大专院校已开设的“C语言程序设计”课程的教材，以取代那些内容老化的教材。

第二部分是面向对象程序设计部分，系统地介绍C++所支持的面向对象程序设计方法的核心内容，按照“循序渐进，突出重点，深入浅出，融会贯通”的教学原则，编写成自成体系的C++教科书。

每章都有小结，归纳出必须掌握的重点内容，并附有大量的习题，以加深读者对重点内容的理解。

在内容安排上有深有浅，且侧重应用，适用于各个层次的读者，既适合以前从未接触过C和C++的初学者，也适合具有一定编程基础的读者作为学习面向对象程序设计方法，提高编程能力的教材和参考书。

<<面向对象程序设计>>

书籍目录

第一部分 C++模块化语言基础	第1章 概论	1.1 C++语言的入门知识
1.1.1 从C到C++	1.1.2 计算机系统的层次结构	1.1.3 C和C++的特征
Visual C++ V6.0使用方法	1.2.1 源程序的编辑、存储和建立	1.2 编译、链接和运行源程序
1.2.3 关闭源程序	1.2.4 调试器Debugger的使用方法	1.2.5 查找信息
建立工程文件	习题1	1.2.6
小结	第2章 数据类型、运算符和表达式	
2.1 基本数据类型	2.2 变量和常量	2.2.1 变量
2.2 数据类型的自动转换和强制转换	2.3 运算符和表达式	2.2.2 常量
2.3.2 关系运算符和关系表达式	2.3.1 算术运算符和算术表达式	2.2.3 数
和增减运算符	2.3.3 逻辑运算符和位逻辑运算符	2.3.4 赋值运算符
2.3.5 条件语句 (if_else) 和条件运算符 (?	2.4 格式化输入/输出函数printf (
:) 及条件表达式	2.3.6 运算符的优先级和结合规则	2.4.1 格式化输入函数scanf ()
) 和scanf ()	2.4.1 格式化输出函数printf ()	2.4.2 格式化输入函数scanf ()
小结	习题2	3.1 语句
3.1.2 复合语句	第3章 语句和流程控制	3.1.1 表
3.2.1 while语句	3.1.3 流程控制语句	3.2 while语句和do-while语句
3.2.2 do-while语句	3.3 for语句	3.3.1 for语句的控制流程
3.3.2 嵌套的for语句	3.4 其它流程控制语句	3.4.1 switch语句
(Jump Statements)	小结	3.4.2 跳转语句
4.1 数组	习题3	第4章 数组与指针
4.1.1 数组的定义	4.1.2 字符数组	4.2 变量的地址和指针变量
4.2.1 变量的地址	4.2.2 指针变量	4.3 指针和数组
4.2.2 指针变量	4.2.3 指针的定义	4.3.1
4.3.2 指针与数组的关系	4.3 指针的运算	4.4.1 指针的赋值
4.4.2 指针的算术运算	4.4.3 指针的关系运算	4.4.4 应用举例
4.4.3 指针的关系运算	4.4.4 应用举例	小结
习题4	第5章 函数	5.1 函数的定义
5.1.2 函数的说明	5.2 变量的存储类型	5.1.1 函数的定义
5.2.2 外部变量 (extern)	5.2.1 自动变量 (auto)	
5.3 函数的调用	5.2.2 静态变量 (static)	5.2.3 寄存器变量 (register)
5.3.1 函数的调用格式	5.3.2 函数调用时参数间的传递方式	5.3.3
5.3.2 函数调用时参数间的传递方式	5.4 外部函数和静态函数	5.5 函数的递归调用
5.4 外部函数和静态函数	5.5 函数的递归调用	5.6
5.6.1 条件编译命令的格式	5.6.2 内部链接 (external	
5.6.2 内部链接 (external	linkage) 和外部链接 (internal linkage)	小结
linkage) 和外部链接 (internal linkage)	习题5	第6章 复杂的数
数据结构和函数调用	6.1 main () 函数	6.2 指针数组和多级指针
6.1 main () 函数	6.2 指针数组和多级指针	6.2.1
6.2.2 多级指针	6.3 多维数组	6.3.1 多维数组的定义
6.3 多维数组	6.4 数组指针	6.3.2 二维
6.4 数组指针	6.4.1 数组指针的定义	6.4.2 数组指针作为函数的参数传递二
6.5 指针函数	6.5.1 指针函数的定义	6.5.2 动态存储技术
6.5.1 指针函数的定义	6.6.1 函数的人口地址和函数指针的定义	6.6 函
6.6.1 函数的人口地址和函数指针的定义	6.6.2 函数指针作为函数的参数	
6.6.2 函数指针作为函数的参数	小结	习题6
6.6.3 函数指针数组和二级函数指针	习题6	第7章 结构体
7.1 结构体 (structure) 的定义和结构变量的说明	7.1.1 结构体的定义	7.1.2
7.1.1 结构体的定义	7.2 结构数组和结构指针	7.2.1 结构数组
7.2 结构数组和结构指针	7.2.1 结构数组	7.2.2 结构指针
7.3 结构体的运算与传递	7.3.1 结构体的运算	7.3.2 结构体在函数间的传递
7.3.1 结构体的运算	7.4 类型定义语句typedef	7.4.1 用typedef语句定义新类型
7.3.2 结构体在函数间的传递	7.4.1 用typedef语句定义新类型	7.5.1 结构型函数
7.3.3 位字段 (Bit Field) 结构体	7.5 结构型函数和结构指针型函数	7.5.2 结构指针型函数
7.4.2 新类型名的应用	7.5.1 结构型函数	7.5.3 用结构体处理链表 (Linked List)
7.5.2 结构指针型函数	7.5.3 用结构体处理链表 (Linked List)	7.6 枚举类型
7.6.1 枚举类型的定义和枚举变量的说明	7.6.2 枚举类型的应用	小结
7.6.2 枚举类型的应用	小结	习题7
8.1 文件的存取 (File Access) 操作	8.1.1 文件	
8.1.1 文件	8.1.2 打开流文件	8.1.3 流文件的读/写
8.1.2 打开流文件	8.1.3 流文件的读/写	8.1.4 关闭流文件函
8.1.3 流文件的读/写	8.1.4 关闭流文件函	数fclose ()
8.1.4 关闭流文件函	8.1.5 文件指针	8.2 标准函数库
8.1.5 文件指针	8.1.6 应用举例	8.2.1 文件
8.1.6 应用举例	8.2 标准函数库	8.2.2 文件的格式化I/O操作函数
8.2 标准函数库	8.2.1 文件	8.2.3 其它标准函数
8.2.1 文件	8.2.2 文件的格式化I/O操作函数	
8.2.2 文件的格式化I/O操作函数	8.2.3 其它标准函数	
8.2.3 其它标准函数	第二部分 面向对象程序设计	第9章 面向对象
小结	习题8	

<<面向对象程序设计>>

程序设计概论	9.1 面向对象程序设计OOP (Object-Oriented Programming) 的基本思想	9.1.1 面向对象程序设计的有关概念	9.1.2 面向对象程序设计的基本思想
9.2 面向对象程序设计的要点	9.2.1 抽象数据类型ADT (Abstract Data Type)	9.2.2 消息传递机制	9.2.3 继承
9.3 C++程序结构的特点	9.3.1 标识符	9.3.2 预处理语句	9.3.3 输入 (Input) /输出 (Output) 流操作语句
9.3.4 函数和语句	9.3.5 其它	小结	习题9
第10章 从C快速过渡到C++			
10.1 数据类型	10.1.1 基本数据类型	10.1.2 复杂数据类型	10.2 C++的常量和变量
10.2.1 常量	10.2.2 变量	10.3 C++的指针	10.4 引用变量 (reference)
10.4.1 “ 引用 ” 的概念	10.4.2 引用的初始化	10.4.3 引用的使用	10.5 C++的运算符
10.5.1 表达式中的类型转换	10.5.2 new和delete运算符	10.5.3 C++的运算符集	10.6 C++的函数
10.6.1 引用的应用	10.6.2 设置函数参数的默认值 (缺省值Default)	10.6.3 内联函数 (Inline Function)	10.6.4 函数重载
10.6.4 函数重载	小结	习题10	第11章 类和对象
第11章 类和对象			
11.1 类的定义	11.1.1 类的定义格式	11.1.2 访问限制符public.Protected	11.1.3 数据成员 (Data Members)
11.1.4 成员函数 (Member Function)	11.2 对象的定义	11.2.1 对象的定义格式	11.2.2 对象指针和对象引用的定义格式
11.2.3 访问类对象成员的方法	11.3 对象的初始化	11.3.1 构造函数和析构函数 (Constructor and Destructor)	11.3.2 构造函数的种类
11.3.3 C++的结构体	11.4 对象指针和对象引用的应用	11.4.1 对象和对象指针作为函数的参数	11.4.2 对象引用作函数参数
11.4.3 this指针	11.4.4 速归类	11.5 静态成员 (Static Members)	11.5.1 静态数据成员 (Static Data Members)
11.5.2 静态成员函数 (State Member Functions)	11.6 友元 (Friend)	11.6.1 友元函数	11.6.2 友元类
11.7 标识符的作用域.可见性和名空间	11.7.1 标识符的作用域规则	11.7.2 作用域的种类	11.7.3 头文件
11.7.4 标识符的名空间 (Namespace of Identifiers)	11.8 对象数组和成员对象	11.8.1 对象数组	11.8.2 成员对象和容器类
11.9 对象的存储类	11.9.1 对象的生存期和存储区域	11.9.2 各种存储类的对象	小结
习题11	第12章 派生类.基类和继承性		
第12章 派生类.基类和继承性			
12.1 继承的概念	12.1.1 什么是继承	12.1.2 两种继承类型	12.2 单继承的派生类
12.2.1 派生类的概念和定义	12.2.2 公有继承和私有继承	12.2.3 基类对象和派生类对象	12.2.4 基类和派生类的成员函数
12.2.5 C++结构体的继承	12.2.6 继承的传递性	12.3 派生类的构造函数和析构函数	12.3.1 派生类构造函数定义格式
12.3.2 派生类构造函数和析构函数的执行次序	12.4 基类和派生类的赋值规则	12.4.1 赋值兼容性规则	12.4.2 基类和派生类的对象指针
12.4.3 子类型和类型适应	12.4.4 不能继承的部分	12.5 多继承	12.5.1 多继承派生类
12.5.2 多继承派生类的构造函数	12.5.3 虚基类	小结	习题12
第13章 多态性和虚函数			
第13章 多态性和虚函数			
13.1 C++的多态性	13.2 运算符重载	13.2.1 运算符函数的定义	13.2.2 运算符重载规则
13.2.3 重载赋值运算符 (operator= ())	13.3 其它运算符的重载	13.3.1 重载增量.减量运算符	13.3.2 函数调用运算符 () 的重载
13.3.3 下标运算符[]的重载	13.4 同名成员函数	13.4.1 重载成员函数	13.4.2 基类和派生类的同名成员函数
13.4.3 基类指针和派生类对象	13.5 虚函数	13.5.1 静态联编	13.5.2 虚函数机制和动态联编技术
13.5.3 典型例程	13.5.4 虚函数表VFT (Virtual Function Table)	13.6 纯虚函数和抽象类	13.6.1 纯虚函数
13.6.2 抽象类	小结	习题13	第14章 C++的流库
第14章 C++的流库			
第14章 C++的流库			
14.1 流库的类层次结构	14.1.1 什么是流 (Stream)	14.1.2 流库的类层次结构	14.1.3 4个标准流对象
14.2 输出流	14.2.1 内部数据类型的输出	14.2.2 ostream类中的成员函数put () 和write ()	14.2.3 用户定义的类型输出
14.3 输入流	14.3.1 istream类中的主要成员函数	14.3.2 用户定义类型的输入	14.3.3 用户定义类型的输入
14.4 格式控制	14.4.1 设置状态标志的成员函数setf ()	14.4.2 清除状态标志位成员函数unsetf ()	14.4.3 取状态标志的成员函数flags ()
14.4.4 设置域			

<<面向对象程序设计>>

宽.填充字符和浮点数精度	14.5 文件I/O流	14.5.1 文件的打开和关闭	14.5.2
ifstream, ofstream和fstream类的构造函数和析构函数		14.5.3 文件的读/写	14.5.4 带命令行
参数的文件	14.6 内存格式化I/O流	小结	第15章 模板
	15.1 函数模板	15.1.1 引入函数模板	15.1.2 函数模板的定义
15.2 类模板	15.2.1 类模板的定义	15.2.2 模板类的实例化和对象的定义	小结
附录	参考文献		

<<面向对象程序设计>>

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>