

21世纪高等学校规划教材 | 计算机科学与技术



面向对象程序设计 (C++版)

董正言 张聪 主编



清华大学出版社

21世纪高等学校规划教材 | 计算机科学与技术

面向对象程序设计 (C++版)

董正言 张聪 主编

清华大学出版社
北京

内 容 简 介

本书以面向对象程序设计技术的本质属性为主线,以 C++ 语言为基础,全面地阐述了面向对象程序设计的基本原理。全书共分 13 章,前 5 章介绍 C++ 语言的基本编程技术,包括 C++ 的数据类型、常用的运算符、结构化控制语句和函数;第 6 章~第 9 章介绍面向对象程序设计方法的核心内容,包括类和对象、类的继承和多态;第 10 章介绍 C++ 的程序结构、编译预处理指令和命名空间等内容;第 11 章介绍 C++ 语言标准库中的输入/输出流类;第 12 章以 C++ 语言为基础,介绍面向对象的异常处理机制;第 13 章简要介绍使用 C++ 语言和面向对象的程序设计方法设计 Windows 应用程序的方法。

本书语言简洁流畅,通俗易懂,内容全面,重点突出,对核心和重点内容都佐以大量的例证。

本书既可以作为高等院校计算机科学与技术及相关专业“面向对象程序设计”课程的授课教材,也可以作为 C++ 程序开发人员的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

面向对象程序设计: C++ 版/董正言,张聪主编. —北京:清华大学出版社,2010.8

(21 世纪高等学校规划教材·计算机科学与技术)

ISBN 978-7-302-22781-6

I. ①面… II. ①董… ②张… III. ①C 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2010)第 090433 号

责任编辑:魏江江 王冰飞

责任校对:焦丽丽

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954, jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京四季青印刷厂

装 订 者:三河市溧源装订厂

经 销:全国新华书店

开 本:185×260 印 张:19.25 字 数:460 千字

版 次:2010 年 8 月第 1 版 印 次:2010 年 8 月第 1 次印刷

印 数:1~3000

定 价:29.50 元

产品编号:037097-01

编审委员会成员

(按地区排序)

清华大学

周立柱 教授
覃 征 教授
王建民 教授
冯建华 教授
刘 强 副教授

北京大学

杨冬青 教授
陈 钟 教授
陈立军 副教授

北京航空航天大学

马殿富 教授
吴超英 副教授
姚淑珍 教授

中国人民大学

王 珊 教授
孟小峰 教授
陈 红 教授

北京师范大学

周明全 教授

北京交通大学

阮秋琦 教授

北京信息工程学院

赵 宏 教授

北京科技大学

孟庆昌 教授

石油大学

杨炳儒 教授

天津大学

陈 明 教授

复旦大学

艾德才 教授

吴立德 教授

吴百锋 教授

杨卫东 副教授

同济大学

苗夺谦 教授

徐 安 教授

华东理工大学

邵志清 教授

华东师范大学

杨宗源 教授

应吉康 教授

上海大学

陆 铭 副教授

东华大学

乐嘉锦 教授

孙 莉 副教授

浙江大学	吴朝晖	教授
	李善平	教授
扬州大学	李 云	教授
南京大学	骆 斌	教授
	黄 强	副教授
南京航空航天大学	黄志球	教授
	秦小麟	教授
南京理工大学	张功萱	教授
南京邮电学院	朱秀昌	教授
苏州大学	王宜怀	教授
	陈建明	副教授
江苏大学	鲍可进	教授
武汉大学	何炎祥	教授
华中科技大学	刘乐善	教授
中南财经政法大学	刘腾红	教授
华中师范大学	叶俊民	教授
	郑世珏	教授
	陈 利	教授
江汉大学	颜 彬	教授
国防科技大学	赵克佳	教授
中南大学	刘卫国	教授
湖南大学	林亚平	教授
	邹北骥	教授
西安交通大学	沈钧毅	教授
	齐 勇	教授
长安大学	巨永峰	教授
哈尔滨工业大学	郭茂祖	教授
吉林大学	徐一平	教授
	毕 强	教授
山东大学	孟祥旭	教授
	郝兴伟	教授
中山大学	潘小轰	教授
厦门大学	冯少荣	教授
仰恩大学	张思民	教授
云南大学	刘惟一	教授
电子科技大学	刘乃琦	教授
	罗 蕾	教授
成都理工大学	蔡 淮	教授
	于 春	讲师
西南交通大学	曾华燊	教授

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制定精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上;精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展,顺应并符合21世纪教学发展的规律,代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版

社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色精品教材包括:

(1) 21 世纪高等学校规划教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。

(2) 21 世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 21 世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。

(4) 21 世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。

(5) 21 世纪高等学校规划教材·信息管理与信息系统。

(6) 21 世纪高等学校规划教材·财经管理与计算机应用。

(7) 21 世纪高等学校规划教材·电子商务。

清华大学出版社经过二十多年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

联系人:魏江江

E-mail: weijj@tup.tsinghua.edu.cn

前言

面向对象程序设计(OOP)技术是当前主流的程序设计技术。和传统的面向过程的程序设计技术相比,面向对象程序设计技术具有明显的优势。这种优势主要体现在以下几个方面:

(1) 传统的面向过程的程序设计方法忽略了数据和操作之间的内在联系,程序中的数据和操作它们的方法分离,而面向对象程序设计技术将程序要处理的数据和处理它们的方法封装在一起,构成一个统一体——对象。程序中用对象模型来建模现实世界中的事物。这样就使得空间模型的结构和问题空间模型的结构相一致。使用面向对象的方法解决问题的思路更加符合人类一贯的思维方法。

(2) 面向对象的封装技术为程序提供了更好的数据安全性。

(3) 面向对象的继承技术为程序提供了更好的可重用性。

(4) 面向对象的多态调用技术使程序具有更好的可扩展性。

(5) 和传统的面向过程的程序设计方法相比,面向对象的程序设计技术更适合开发大型的图形界面应用程序。

目前,常用的面向对象的编程语言有 C++、Java、C# 等。

C++语言是由 C 语言发展演变成的一种面向对象的程序设计语言。它既具备 C 语言面向过程设计方法的特点,同时又支持面向对象的程序设计方法。它是目前程序员使用最多的编程语言之一。

本书以 C++ 语言为基础,阐述面向对象的编程原理。本书的特色是内容全面、语言简洁易懂、重点突出,是一部面向大学本科计算机科学相关专业的人门教材。

为了使读者能够透彻理解面向对象编程的原理和方法,本书使用了大量自编的例程。全部例程的源代码均使用 Visual C++ 6.0 编写,并编译通过。

本书由董正言主编,张聪、刘文涛、阮灵等老师也参加了部分章节的编写工作。

由于学识水平和时间的限制,疏漏和不妥之处在所难免,敬请批评指正。

作 者

2010 年 5 月

目 录

第 1 章 绪论	1
1.1 编程语言的发展	1
1.2 C++ 语言简介	2
1.3 面向对象的程序设计方法	2
1.4 第一个 C++ 程序	4
1.5 小结	5
习题	6
第 2 章 基本数据类型和运算符	7
2.1 基本概念	7
2.1.1 程序实例	7
2.1.2 C++ 字符集	8
2.1.3 C++ 关键字	8
2.1.4 标识符	8
2.1.5 程序注释	9
2.2 基本数据类型	9
2.2.1 整数类型	9
2.2.2 浮点类型	9
2.2.3 字符类型	9
2.2.4 布尔类型	10
2.3 变量和常量	10
2.3.1 变量	10
2.3.2 常量	11
2.4 简单的输入和输出	13
2.5 C++ 运算符和表达式	14
2.5.1 赋值运算符和赋值表达式	14
2.5.2 算术运算符和算术表达式	15
2.5.3 关系运算符和关系表达式	16
2.5.4 逻辑运算符和逻辑表达式	16
2.5.5 位运算符	17
2.5.6 逗号运算符和逗号表达式	19

2.5.7	条件运算符和条件表达式	19
2.5.8	sizeof 运算符	19
2.5.9	其他运算符	19
2.6	数据类型转换	20
2.7	小结	21
习题		22
第3章	控制语句	24
3.1	选择结构	24
3.1.1	if-else 选择语句	24
3.1.2	嵌套的 if-else 语句	25
3.1.3	if-else if 语句	25
3.1.4	switch 语句	26
3.2	循环结构	28
3.2.1	while 循环语句	28
3.2.2	do-while 循环语句	29
3.2.3	for 循环语句	30
3.2.4	嵌套的循环语句	31
3.3	其他流控制语句	32
3.3.1	break 语句和 continue 语句	32
3.3.2	goto 语句	33
3.4	小结	33
习题		33
第4章	复合数据类型	35
4.1	数组	35
4.1.1	数组的定义和初始化	35
4.1.2	使用数组	36
4.1.3	字符数组和字符串	38
4.2	指针和引用	39
4.2.1	定义和使用指针	39
4.2.2	指针和数组	41
4.2.3	数组指针和指针数组	43
4.2.4	使用操作符 new 和 delete 进行动态存储分配	45
4.2.5	引用的定义和使用	46
4.3	枚举和结构	48
4.3.1	枚举	48

4.3.2 结构	50
4.4 小结	51
习题	52
第 5 章 函数	54
5.1 定义和调用函数	54
5.1.1 函数的定义	54
5.1.2 函数的调用	55
5.1.3 函数原型	56
5.2 传递参数	59
5.2.1 传值传递	59
5.2.2 引用传递	60
5.3 局部变量和全局变量	61
5.4 函数调用的实现	63
5.5 内联函数	64
5.6 递归函数	65
5.7 参数的默认值	69
5.8 指针函数和函数指针	70
5.8.1 指针函数	70
5.8.2 函数指针	72
5.9 函数重载	73
5.10 函数模板	75
5.11 小结	77
习题	78
第 6 章 类和对象(上)	80
6.1 面向对象程序设计概述	80
6.2 创建类	81
6.2.1 定义类	81
6.2.2 类的实现	82
6.3 创建和使用对象	84
6.4 类成员的访问控制	87
6.4.1 类的公有成员	88
6.4.2 类的私有成员	88
6.4.3 类的保护成员	89
6.5 内联的成员函数	93
6.6 构造函数	93

6.6.1	定义类的构造函数	94
6.6.2	默认的构造函数	95
6.6.3	带默认参数值的构造函数	96
6.7	拷贝构造函数	97
6.7.1	定义类的拷贝构造函数	98
6.7.2	默认的拷贝构造函数与对象的浅拷贝问题	100
6.8	析构函数	103
6.9	小结	106
习题		107
第7章	类和对象(下)	108
7.1	类的静态成员	108
7.1.1	静态数据成员	108
7.1.2	静态函数成员	110
7.2	对象指针	112
7.3	动态创建	114
7.3.1	动态创建对象	114
7.3.2	动态创建对象数组	115
7.4	类作用域	117
7.4.1	类成员具有类作用域	117
7.4.2	具有类作用域的数据成员被局部变量屏蔽	118
7.5	this 指针	118
7.6	类的组合	122
7.7	常对象和类的常成员	128
7.7.1	常对象	128
7.7.2	常数据成员	129
7.7.3	const 成员函数	129
7.8	类模板	130
7.9	友元	136
7.9.1	友元函数	136
7.9.2	友元类	138
7.10	string 类	141
7.10.1	构造字符串	141
7.10.2	常用的字符串操作	142
7.11	小结	144
习题		145

第 8 章 类的继承	147
8.1 基类和派生类	147
8.2 定义派生类	148
8.3 继承方式与访问权限	149
8.4 构造派生类对象	151
8.4.1 派生类对象的结构	152
8.4.2 派生类的构造函数	152
8.5 成员覆盖	157
8.6 实例学习——图形类家族	159
8.7 多重继承	163
8.7.1 多重继承简介	163
8.7.2 多重继承引发的二义性问题	167
8.8 虚基类	172
8.9 对象类型转换	176
8.10 小结	178
习题	178
第 9 章 多态	183
9.1 什么是多态	183
9.2 虚函数和动态绑定	184
9.3 纯虚函数和抽象类	192
9.4 编译期多态——运算符重载	193
9.4.1 什么是运算符重载	193
9.4.2 用类的成员函数实现运算符重载	194
9.4.3 用类的友元函数实现运算符重载	199
9.4.4 重载赋值运算符=	202
9.5 运行时类型识别(RTTI)	204
9.5.1 dynamic_cast 操作符	205
9.5.2 typeid 操作符和 type_info 类	206
9.6 小结	210
习题	210
第 10 章 程序结构、预处理和命名空间	212
10.1 多文件结构的源程序	212
10.2 文件间的信息共享	217
10.2.1 头文件	217

10.2.2	关键字 extern	217
10.2.3	使用关键字 static 避免同名冲突	218
10.2.4	函数的声明	219
10.2.5	类的声明	219
10.3	预处理	220
10.3.1	#include 指令	220
10.3.2	#define 指令	221
10.3.3	条件预处理指令	221
10.3.4	使用条件预处理指令避免重复包含	222
10.4	命名空间	223
10.5	小结	224
	习题	225

第 11 章 输入和输出 226

11.1	什么是输入/输出流	226
11.2	输入/输出流类	227
11.2.1	预定义的流对象	227
11.2.2	插入运算符和提取运算符	228
11.2.3	格式化标志和设置格式化标志的函数	230
11.2.4	I/O 格式操作符	231
11.2.5	控制输入/输出格式的函数	232
11.2.6	常用的 I/O 函数	234
11.3	磁盘文件的输入/输出	236
11.3.1	打开文件	236
11.3.2	数据的存储格式和文件的打开模式	237
11.3.3	文件输入/输出	238
11.3.4	文件指针	242
11.4	小结	243
	习题	244

第 12 章 异常处理 245

12.1	抛出异常	245
12.2	捕获和处理异常	245
12.3	异常的传递途径	247
12.4	异常类	248
12.5	小结	251
	习题	251

第 13 章 Windows 编程基础	252
13.1 事件驱动机制和 Windows SDK 编程	252
13.2 MFC	256
13.2.1 一个简单的 MFC 应用程序	256
13.2.2 MFC 简介	258
13.2.3 MFC 程序结构分析	263
13.3 消息映射机制	265
13.3.1 Windows 消息	265
13.3.2 MFC 消息映射机制	267
13.4 Windows 程序实例	273
13.5 小结	281
习题	281
附录 A	282
参考文献	288

第 1 章

绪论

1.1 编程语言的发展

计算机由硬件和软件组成。硬件是指组成计算机的各种零部件,如中央处理器(CPU)、主存储器(内存)、硬盘等;软件是指指挥硬件运行的程序,如操作系统、各种硬件的驱动程序和其他应用程序等。程序设计语言是用来设计计算机软件的,自 1946 年第一台电子数字计算机诞生至今,计算机科学发展迅猛,从一开始每秒钟只能做几千次加法运算的电子管计算机,到目前每秒几亿次运算的大规模集成电路计算机;从一开始只是作为一个应用于科学计算的工具,到目前走进了千家万户,遍及社会应用的各个领域;从一开始的单机运行,各自为战,到目前的并行计算,网络共享,……在这半个多世纪以来,计算机科学取得了长足的发展和进步,同时程序设计语言也经历了漫长的发展过程。

第一代程序设计语言是机器语言。机器语言是计算机硬件可以识别的机器指令,由二进制编码组成。不难想象,采用机器语言编写程序的难度很大,编程人员既要掌握数目繁多的机器指令,又要熟知计算机的硬件结构,只有极少数专业人员才能胜任。

不久后,出现了汇编语言,汇编语言就是用一些容易记忆的助记符代替机器语言中的二进制编码,使机器指令看上去更容易理解。常用的助记符有 MOV、ADD、SUB 等。计算机在运行由汇编语言编写的程序时,首先需要把程序翻译成计算机能够识别的机器语言,完成这个翻译功能的程序叫做汇编程序。因为不同类型的计算机具有不同的硬件结构和指令系统,所以,由汇编语言编写的程序不是平台独立的,也就是说,为一种计算机编写的汇编语言程序必须经过修改才能在另一种类型的计算机上运行。

由于机器语言和汇编语言都能直接操作计算机硬件,所以被称为低级语言。

20 世纪 60 年代末,出现了结构化高级编程语言。高级语言采用数量有限的、容易理解的执行语句来编写程序,并允许程序员用具有一定含义的名字为程序中使用的数据命名。高级语言致力于解决问题,而不针对特定的硬件,屏蔽了计算机的硬件细节,采用高级语言编写的程序不需要修改就可以在不同类型的计算机上运行,程序的可移植性高。计算机在运行高级语言编写的程序时,首先要用一个翻译程序把高级语言翻译成硬件可识别的机器语言,这个翻译程序叫做编译程序。常见的结构化高级语言有 C、FORTRAN、PASCAL、BASIC 等。采用高级语言进行结构化程序设计的原则是:自顶向下,逐步求精。

随着科学技术的飞速发展,计算机逐步走进了各行各业,计算机应用软件的规模也越来越庞大,尤其是进入 20 世纪 90 年代,图形界面操作系统(例如微软的 Windows 操作系统)

以其优秀的人机交互性和可操作性逐步取代了命令流操作系统(例如 MS-DOS),并得到了普及,采用结构化程序设计方法设计大型的图形界面应用程序显得力不从心;这时,一种新型的程序设计方法——面向对象的程序设计方法应运而生并取代了结构化程序设计方法,成为大型程序设计的主流技术。同时,面向对象的程序设计语言也取代了结构化高级语言,成为主流的编程语言。在面向对象的程序设计方法中,把数据和操作它们的方法封装在一起,抽象出解域空间对象模型,用于描述实际的问题域空间对象,更直接地反映了客观世界中的事物,以及它们之间的关系;同时,利用继承和多态技术,极大地提高了程序代码的重用性和程序设计效率。当前常见的面向对象程序设计语言有 C++、Java 等。

1.2 C++ 语言简介

C++语言是由C语言进化、发展而来,C语言于20世纪70年代诞生于美国的贝尔实验室。C语言的特点是使用简洁、灵活,数据类型和运算符丰富,具有结构化控制语句,既能像汇编语言一样直接访问存储器的物理地址、管理通信端口和磁盘驱动器,又具有结构化高级语言的特点——程序具备良好的可读性和可移植性;总之,C语言是一种功能强大的结构化程序设计语言。由于它同时具有低级语言和结构化高级语言的特点,也被一些人称为中级语言。C语言从诞生时起,就以其使用简洁、灵活、功能强大等特点赢得了多数程序员的青睐,成为20世纪80年代占据统治地位的编程语言。但是C语言不支持面向对象的程序设计。

C++语言诞生于20世纪80年代,和C语言相同,也是由美国的贝尔实验室开发的。C++由C语言发展演变而来,一方面C++继承了C语言使用简洁、灵活、功能强大等特性,同时C++也支持面向对象的程序设计(OOP);因此,很多人把C++称为“带类的C”;其实,和C语言相比,C++不光是增加了对面向对象程序设计的支持,在其他很多方面,C++都对C语言进行了改进和优化。

为了提高C++程序的可移植性,美国国家标准局(ANSI)和国际标准化组织(ISO)于1990年建立了联合组织ANSI/ISO,负责制定C++标准。1998年发布了C++标准第一版(ISO/IEC14882:1998),2003年发布了C++标准第二版(ISO/IEC14882:2003)。

C++数据类型和运算符丰富,具有结构化的控制语句,能直接访问和控制硬件,同时支持面向对象程序设计(OOP)。是一种使用简单、灵活,功能强大的面向对象的程序设计语言。

1.3 面向对象的程序设计方法

早期的计算机主要用于科学计算,随着计算机技术的发展,计算机的应用领域也不断扩大,要解决问题的复杂程度不断提高,计算机程序的规模日益增大,大型程序的修改和维护工作越来越困难,程序的维护成本日益增高,甚至远远高于程序的开发成本。程序结构的混乱是造成这种现象的根本因素之一。

20世纪60年代产生的面向过程的结构化程序设计思想,为使用计算机开发和维护复杂程序提供了有利的手段。那么,什么是面向过程的程序设计方法呢?面向过程的结构化程序设计方法简单地说就是:采用自顶向下、逐步求精的思路。即对复杂的命题进行层层