

高等学校规划教材

工程应用型院校计算机系列教材

安徽省高等学校“十二五”规划教材
安徽省高等学校计算机教育研究会推荐用书

总主编 胡学钢



Visual C++面向对象程序设计

Visual C++ MIANXIANG DUIXIANG CHENGXU SHEJI

主 编 郭有强



北京师范大学出版集团
BEIJING NORMAL UNIVERSITY PUBLISHING GROUP
安徽大学出版社
北京师范大学出版社

Visual C++ 面向对象程序设计

总主编 胡学钢
主 编 郭有强
副主编 朱洪浩
参 编 马金金 李 妍



北京师范大学出版集团
BEIJING NORMAL UNIVERSITY PUBLISHING GROUP

安徽大学出版社
北京师范大学出版社

图书在版编目(CIP)数据

Visual C++面向对象程序设计/郭有强主编. 合肥:安徽大学出版社;北京:北京师范大学出版社,2015.8

工程应用型院校计算机系列教材

ISBN 978-7-5664-0956-0

I. ①V… II. ①郭… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 180745 号

Visual C++面向对象程序设计

胡学钢 总主编
郭有强 主 编

出版发行: 北京师范大学出版集团
安 徽 大 学 出 版 社
(安徽省合肥市肥西路3号 邮编 230039)
北京师范大学出版社
(北京新街口外大街19号 邮编 100875)
www.bnupg.com.cn
www.ahupress.com.cn
www.bnup.com.cn

印 刷: 合肥市裕同印刷包装有限公司
经 销: 全国新华书店
开 本: 184mm×260mm
印 张: 27.5
印 数: 670 千字
版 次: 2015 年 8 月第 1 版
印 次: 2015 年 8 月第 1 次印刷
定 价: 55.00 元

ISBN 978-7-5664-0956-0

策划编辑:李 梅 蒋 芳
责任编辑:蒋 芳
责任校对:程中业

装帧设计:李 军
美术编辑:李 军
责任印制:赵明炎

版权所有 侵权必究

反盗版、侵权举报电话:0551—65106311

外埠邮购电话:0551—65107716

本书如有印装质量问题,请与印制管理部联系调换。

印制管理部电话:0551—65106311

编写说明

计算机科学与技术的迅速发展,促进了许多相关学科领域以及应用分支的发展,同时也带动了各种技术和方法、系统与环境、产品以及思维方式等的发展,由此而进一步激发了对各种不同类型人才的需求。按照教育部计算机科学与技术专业教学指导委员会的研究报告来分,学校培养的人才类型可以分为科学型、工程型和应用型三类,其中科学型人才重在基础理论、技术和方法等的创新;工程型人才以开发实现预定功能要求的系统为主要目标;应用型人才以系统集成为主要途径实现特定功能的需求。

虽然这些不同类型人才的培养有许多共同之处,但是因不同类型人才的就业岗位所需要的责任意识、专业知识能力与素质、人文素养、治学态度、国际化程度等方面存在一定的差异,因而培养目标、培养模式等方面也存在不同。对大多数高校来说,很难兼顾各类人才的培养。因此,合理定位培养目标是确保教学目标和人才培养质量的关键。

由于当前社会领域从事工程开发和应用的岗位数量远远超过从事科学人才的数量,结合当前绝大多数高校的办学现状,安徽省高等学校计算机教育研究会在和多所高校专业负责人以及来自企业的专家反复研究和论证的基础上,确定了以培养工程应用型人才为主的安徽省高等学校计算机类专业的培养目标,并组织研讨组共同探索相关问题,共同建设相关教学资源,共享研究和建设成果,为全面推动安徽省高等学校计算机教育教学水平做出积极的贡献。北京师范大学出版集团安徽大学出版社积极支持安徽省高等学校计算机教育研究会的工作,成立了编委会,组织策划并出版了全套工程应用型计算机系列教材。

为了做好教材的出版工作,编委会在许多方面都采取了积极的措施:

编委会组成的多元化:编委会不仅有来自高校的教育领域的资深教师和专家,而且还有从事工程开发、应用技术的资深专家,从而为教材内容的重组提供更为有力的支持。

教学资源建设的针对性:教材以及教学资源建设的目标就是要突出体现“学以致用”的原则,减少“学不好,用不上”的空泛内容,增加其应用案例,尤其是增设涵盖更多知识点和应用能力的系统性、综合性的案例,以培养学生系统解决问题的能力,进而激发其学习兴趣。

建设过程的规范性:编委会对整体的框架建设、对每本教材和资源的建设都采取汇报、交流和研讨的方式,以听取多方意见和建议;每本书的编写组也都进行反复的讨论和修订,努力提高教材和教学资源的质量。

如果我们的工作能对安徽省高等学校计算机类专业人才的培养做出贡献,那将是我们的荣幸。真诚欢迎有共同志向的高校、企业专家提出宝贵意见和建议,更期待你们参与我们的工作。

胡学钢

2013年8月10日于合肥

编委会名单

主 任 胡学钢(合肥工业大学)

委 员 (以姓氏笔画为序)

王 浩(合肥工业大学)

王一宾(安庆师范学院)

叶明全(皖南医学院)

孙 力(安徽农业大学)

刘仁金(皖西学院)

朱昌杰(淮北师范大学)

沈 杰(合肥炜煌电子有限公司)

李 鸿(宿州学院)

陈 磊(淮南师范学院)

陈桂林(滁州学院)

张先宜(合肥工业大学)

张润梅(安徽建筑大学)

张燕平(安徽大学)

金庆江(合肥文康科技有限公司)

周国祥(合肥工业大学)

周鸣争(安徽工程大学)

宗 瑜(皖西学院)

郑尚志(巢湖学院)

钟志水(铜陵学院)

姚志峰(蓝盾信息安全技术股份有限公司)

郭有强(蚌埠学院)

黄 勇(安徽科技学院)

黄海生(池州学院)

潘地林(安徽理工大学)

前 言

1. 关于本书

随着程序设计技术的不断发展,社会对软件人才的要求也越来越高。要想成为一个高水平的程序员,需要加强程序设计语言、算法设计与分析、程序设计方法学和程序设计环境与工具等方面的学习。因此,高校开设了相关课程,如很多高校的计算机及相关专业开设了面向对象程序设计类课程。面向对象程序设计是软件开发领域的主流技术,它集抽象性、封装性、继承性和多态性于一体,实现了代码重用和代码扩充,极大地减少了软件开发的繁杂性,提高了软件开发的效率。

C++为面向对象技术提供全面支持,是主流的面向对象程序设计语言。因此C++在当前程序设计领域中的地位是很重要的。高等院校开设该课程,其目的是为了让学生掌握面向对象程序设计的概念和方法,深刻理解面向对象程序设计的本质,并用面向对象技术来编写程序、开发软件。Visual C++是基于 Windows 的可视化开发环境,同时还提供了C++类库,其中最重要的是MFC。随着技术的发展,MFC已不是Visual C++中唯一的类库,还有ATL、STL等。Microsoft公司还推出了Visual C++.NET,但考虑作为专业基础课教材,重点是介绍程序设计思想和技术,而不是讲解具体的开发环境,因此,本书仍然以Visual C++ 6.0作为开发环境。

Visual C++程序设计内容广泛,本书除了包含C++的基础内容、基于MFC的Windows编程及数据库编程内容外,还增加了C++程序员不可或缺的基本技能STL内容,掌握它对提升C++编程能力大有裨益,还增加了图形绘制相关内容,因为图形在任何一个可视化工程项目中都是不可缺少的,掌握图形程序设计方法是非常重要的。

本书作者都是长期在高校从事计算机专业教学与科研的一线教师,有丰富的编程与教学经验。本书有助于帮助读者深刻理解面向对象程序设计的思想,掌握C++及Visual C++程序分析能力和设计技能。

2. 本书结构

本书共分12章。第1章主要介绍面向过程与面向对象的程序设计方法的区别;第2章介绍在面向过程的程序设计方面C++对C语言的改进及扩充;第3~6章以面向对象程序设计的抽象性、封装性、继承性和多态性4大主要特征为主线组织内容,系统而全面地介绍面向对象程序设计的基本概念和方法;第7章以容器、迭代器和算法为主介绍类模板与STL编程;第8章介绍了流类库、格式化I/O流及文件操作;第9章通过循序渐进的方式讲述利

用 MFC 开发 Windows 应用程序的基本编程模式和程序设计方法,配以具体应用的详细实训步骤,使读者能够把握 Visual C++ 的脉搏;第 10 章以 ODBC 和 ADO 对象为主介绍了 Visual C++ 数据库编程技术;第 11 章介绍了用 Visual C++ 语言进行绘图程序设计;第 12 章给出了基于两种编程模式的综合实训及图形绘制实训,以提高读者的综合编程能力。

本书配套《Visual C++面向对象程序设计实验指导与习题集》,提供了上机实验指导、课程设计、习题解答、模拟试题等内容。

3. 本书特色

本书通过比较的方法介绍了面向过程和面向对象程序设计的概念和方法,读者可以从中理解到思维方式的变化;在讲授面向对象编程时,紧紧围绕面向对象程序设计的特征展开;在讲述每部分内容时,采用先说明开设这部分内容的理由的方法,使读者对所学概念有充分的理解;每部分在具体阐述的过程中既突出重点,又注重细节用法;为了便于读者自学,全书使用了大量的注释;每部分都精选了大量的典型实例。本书追求用通俗易懂的语言、生活中的现象来阐述基本理论,突出重点,解释到位,讲深讲透“难点”部分;所有内容均采用“提出问题,说明问题,解决问题”的模式编写,更多地考虑到学习者的接受能力和接受方式;采用先例题后理论的模式,用程序例子来说明难懂的抽象概念;对许多问题描述多个编程方案,以使读者在比较中学习,进而深刻理解相关设计思想和分析相关算法。本书除了正面阐述,说明什么是正确的,还注意从反面来说明什么是不正确的,对编程中常用的以及在学习过程中容易出错的知识点进行了详尽的讲解;在讲解 MFC 编程和图形绘制的过程中,注重操作步骤及细节,以使读者具有很强的模仿性和可操作性。

本书注重可读性和可用性。内容丰富,结构紧凑,概念阐述清楚,例题典型丰富,注重编程能力培养。在每章开头对本章内容进行总体描述;在每章结尾有本章小结,对本章的主要内容作归纳总结;每章最后还配有一定数量的习题帮助读者巩固所学知识。每章通过大量典型的实例解析新的知识点。书中还通过大量图、表对知识点作总结或从本质上进行分析,便于读者记忆和理解。本书还为授课教师提供配套的电子课件、全部例题源代码以及习题源代码。本书中所有的程序都在 Visual C++ 6.0 下调试通过。

4. 适用对象

本书是在已经出版的两套 C++ 教材和省级本科精品课程建设的基础上,作者总结多年教学经验、参考大量的国内外有关资料并结合自身的实际工程项目经验编写而成,融会贯通了 C++ 面向对象、STL 编程、Windows 程序设计、数据库编程及图形绘制等内容,是一本内容全面、实用性很强的应用型教材。本书既可作为高等院校程序设计课程教材,也可作为工程技术人员参考用书。

本书由郭有强担任主编并统稿,编写第5、7、9~12章;朱洪浩担任副主编,编写第6、8章,并负责本书全部例题源代码的测试。参加本书编写工作的还有:李妍编写第1、2章,马金金编写第3、4章并负责电子讲稿的制作。本书电子讲稿及全部例题的源代码可在网站<http://cjpke.bbxy.edu.cn/>下载。

在本书的编写过程中,查阅了一些文献,“参考文献”中列出了部分文献的作者,在此表示感谢;感谢北京师范大学出版集团各位编辑,他们为本教材的出版倾注了大量的心血和热情;感谢读者选择使用本书。由于作者水平有限,书中难免有错误、疏漏、不妥之处,恳请读者提出批评和修改意见,我们将不胜感激。在使用本书时如遇到什么问题需要与作者商榷,或想索取其他相关资料,请与作者联系。联系方式:bbxyguo@163.com。

郭有强

2015年8月

内容简介

本书全面介绍面向对象程序设计的编程思想、C++及 Visual C++程序设计技术。主要包括面向对象程序设计概述、C++在面向过程方面对 C 语言的扩充、C++面向对象程序设计基础内容(包括类与对象、继承与派生、多态性与虚函数、运算符重载、流等)、类模板与 STL 编程、利用 MFC 开发 Windows 应用程序、数据库编程技术、图形绘制等。本书注重知识的系统性和连贯性,在内容上循序渐进,注意与后续课程的衔接,内容取舍合理,设计了易于读者学习的教材体系,并以通俗易懂的语言深入浅出地讲解复杂概念,通过精选大量典型例题,帮助读者理解编程思想及相关概念;通过综合性实训内容,注重读者实际编程能力的提高。

本书中所有的程序都在 Visual C++ 6.0 下调试通过。为了配合教学,本书配有《Visual C++面向对象程序设计实验指导与习题集》,并提供全方位的教学资源。本书既可作为高等院校程序设计课程的教材,也可作为工程技术人员的参考用书。

目 录

第 1 章 C++概述	1
1.1 C++语言简介	1
1.1.1 程序	1
1.1.2 程序设计语言	1
1.1.3 程序设计	2
1.1.4 C++语言的发展及举例	3
1.2 面向过程和面向对象编程概述	4
1.2.1 面向过程程序设计	4
1.2.2 面向对象程序设计	6
1.2.3 泛型程序设计	7
1.3 面向过程和面向对象的程序设计方法比较	8
1.4 C++程序开发过程及调试	10
1.4.1 C++程序开发过程	10
1.4.2 Visual C++ 6.0 调试C++程序的操作过程	11
本章小结	14
习题 1	15
第 2 章 C++对 C 基本语法的扩充	16
2.1 C++的输入输出	16
2.1.1 输出流对象	16
2.1.2 输入流对象	18
2.1.3 字符数组的输入和输出	21
2.2 重载函数	24
2.2.1 为什么要进行函数重载	24
2.2.2 使用函数重载的条件	25
2.2.3 重载函数的使用方法	25
2.2.4 函数的默认参数	25
2.3 函数模板与模板函数	27
2.4 内联函数	28
2.5 内存的动态分配与释放	29
2.5.1 new 运算符的用法	30
2.5.2 delete 运算符的用法	30
2.6 引用	31
2.6.1 引用的概念	31

2.6.2	使用引用的注意事项	32
2.6.3	引用作为函数参数	33
2.6.4	函数返回引用	36
2.7	const 修饰符	37
2.7.1	用 const 定义常量	37
2.7.2	用 const 来限制指针	38
2.7.3	const 函数	39
2.8	名字空间	40
2.8.1	名字空间的作用	40
2.8.2	名字空间的使用	41
2.8.3	名字空间综合举例	43
2.9	sizeof 运算	47
2.10	程序设计举例	48
	本章小结	56
	习题 2	57
第 3 章	类和对象	62
3.1	类的定义和使用	62
3.1.1	类定义格式	62
3.1.2	成员函数的定义位置	63
3.1.3	内联成员函数	65
3.1.4	常量成员函数	66
3.2	对象的创建和使用	67
3.2.1	对象的种类和创建	67
3.2.2	对象作为函数的参数和返回值	68
3.2.3	this 指针	69
3.3	构造函数的概念和使用	71
3.3.1	为什么引入构造函数	71
3.3.2	重载构造函数	73
3.3.3	默认参数的构造函数	75
3.3.4	拷贝构造函数	76
3.3.5	成员初始化参数表	78
3.4	析构函数的概念和使用	81
3.4.1	为什么引入析构函数	81
3.4.2	析构函数的使用	81
3.5	堆对象的概念和使用	82
3.5.1	创建和删除单个堆对象的方法	82
3.5.2	创建和删除堆对象数组的方法	84
3.6	静态数据成员和静态成员函数	85
3.6.1	为什么需要静态数据成员	85
3.6.2	静态数据成员的访问和初始化	86

3.6.3 静态成员函数的概念和使用	88
3.7 友元函数和友元类	89
3.7.1 友元的概念和使用	89
3.7.2 使用友元的注意事项	93
3.8 程序设计举例	93
本章小结	97
习题 3	97
第 4 章 继承与派生	100
4.1 为什么要引入继承的概念	100
4.1.1 继承与派生问题举例	101
4.1.2 继承与派生的概念	101
4.2 基类和派生类	101
4.2.1 基类与派生类的概念	101
4.2.2 派生类的定义	101
4.3 派生方式	103
4.3.1 public 派生	103
4.3.2 private 派生	104
4.3.3 protected 派生	106
4.4 不同派生方式之间的区别	106
4.5 派生类的构造函数和析构函数	108
4.5.1 派生类的构造函数	108
4.5.2 基类构造函数的调用方式	108
4.5.3 派生类的析构函数	109
4.6 多继承和虚基类	110
4.6.1 多继承的定义	110
4.6.2 多继承中的构造函数和析构函数	112
4.6.3 二义性与虚基类	113
4.7 程序设计举例	118
本章小结	122
习题 4	123
第 5 章 多态性与虚函数	125
5.1 为什么需要多态性	125
5.1.1 多态性的实现方法	125
5.1.2 静态多态性和动态多态性	126
5.2 对虚函数的限制	131
5.2.1 声明虚函数的限制	131
5.2.2 虚函数的使用限制	132
5.3 在成员函数中调用虚函数	135
5.4 在构造函数中调用虚函数	136

5.5 纯虚函数和抽象类	137
5.6 程序设计举例	139
本章小结	140
习题 5	141
第 6 章 运算符重载	144
6.1 为什么要进行运算符重载	144
6.1.1 运算符重载的例子	144
6.1.2 运算符重载的注意事项	147
6.2 赋值运算符和四则运算符的重载	148
6.3 自增 1 和自减 1 运算符重载	150
6.4 关系运算符的重载	152
6.5 算术赋值运算符的重载	153
6.6 下标运算符的重载	155
6.7 插入与抽取运算符的重载	157
6.7.1 插入运算符的重载	157
6.7.2 抽取运算符的重载	158
6.8 类型转换	160
6.8.1 基本类型转换和自定义类型的相互转换	161
6.8.2 自定义类型之间的转换	161
6.9 程序设计举例	165
本章小结	169
习题 6	169
第 7 章 类模板与 STL 编程	171
7.1 类模板	171
7.1.1 为什么要引进类模板	171
7.1.2 类模板定义格式	171
7.1.3 使用类模板创建对象	173
7.2 STL 编程	178
7.2.1 标准模板库 STL	178
7.2.2 容器	180
7.2.3 迭代器	207
7.2.4 算法	229
7.2.5 函数对象	238
本章小结	240
习题 7	240
第 8 章 流	244
8.1 C 的标准 I/O 函数的缺陷	244
8.2 I/O 流的概念	245

8.3	I/O 流类库结构	246
8.4	标准 I/O 流	246
8.4.1	标准 I/O 流的类层次	246
8.4.2	预定义流对象	247
8.4.3	预定义的插入类型	247
8.4.4	预定义的抽取类型	248
8.5	用于无格式 I/O 的 ios 类成员函数	249
8.6	格式化 I/O 流	253
8.6.1	使用格式状态标志或调用格式化成员函数	253
8.6.2	使用操纵算子	257
8.7	文件操作	259
8.7.1	文件与文件流概述	259
8.7.2	文件流的类层次	260
8.7.3	文件的打开和关闭	260
8.7.4	文件读写操作举例	262
8.8	随机访问数据文件	266
8.9	程序设计举例	268
	本章小结	270
	习题 8	270
第 9 章	利用 MFC 开发 Windows 应用程序	273
9.1	Windows 应用程序的特点与消息驱动机制	273
9.1.1	基于 Windows 操作系统的应用程序的特点	273
9.1.2	典型的 Windows 应用程序结构	274
9.1.3	学习 MFC 的方法	274
9.2	利用 MFC AppWizard 创建 Windows 应用程序	275
9.3	MFC 应用程序的类和文件	276
9.3.1	类说明	276
9.3.2	文件说明	277
9.4	Windows 消息处理	279
9.4.1	利用 ClassWizard 编制消息处理函数	279
9.4.2	Windows 消息	282
9.4.3	消息的发送与接收的基本过程和机制	284
9.5	文档/视图结构	285
9.5.1	视图类	285
9.5.2	文档类	286
9.6	菜单与工具栏	290
9.7	对话框与控件	292
9.7.1	对话框	292
9.7.2	控件	299
9.8	程序设计举例	301

本章小结	311
习题 9	311
第 10 章 数据库编程	313
10.1 数据库访问技术概述	313
10.2 ODBC 编程技术	315
10.3 ADO 编程技术	321
10.3.1 ADO 技术简介	321
10.3.2 ADO 对象模型	322
10.3.3 使用 ADO 对象开发数据库应用程序	323
本章小结	336
习题 10	336
第 11 章 图形绘制	339
11.1 设备环境和设备环境类	339
11.1.1 设备环境	339
11.1.2 图形设备接口	340
11.1.3 设备环境类	341
11.1.4 获取设备环境的方法	343
11.2 坐标映射	343
11.3 常用的图形数据结构和类	345
11.4 绘图工具类	348
11.4.1 使用 GDI 对象	349
11.4.2 CPen 类和 CBrush 类	350
11.4.3 简单图形的绘制	358
11.4.4 CFont 类	367
11.4.5 CBitmap	380
11.4.6 显示位图	382
11.4.7 动画图形的制作	385
本章小结	390
习题 11	390
第 12 章 综合实训	394
12.1 控制台应用程序:商品信息管理系统	394
12.2 MFC 应用程序:班级信息管理系统	401
12.3 图形绘制:图像局部放大实现	416
本章小结	421
习题 12	421
附录 标准 ASCII 码表	424
参考文献	425

第 1 章 C++ 概述

【学习目标】

- 了解程序设计语言及程序设计的一般步骤
- 掌握面向过程和面向对象编程的特点和不同点
- 掌握利用 Visual C++ 6.0 集成开发环境调试 C++ 控制台应用程序

1.1 C++ 语言简介

程序设计是软件开发中一个重要的部分,但不是全部。即使是专业的程序员或 IT 工程师,也不必将全部精力都放在编写代码上(通常,分析用户需求要占用更多的时间)。程序设计与数学类似,非常挑战开发者的智力。与数学不同的是,计算机能够做出反馈,人们可以从经验中学习,并根据学到的东西改变行为。另外,程序设计不像数学那样抽象,因此更容易被接受。学习程序设计,必须借助于一门计算机语言。C++ 语言不仅应用广泛,还支持现实世界中软件所使用的关键概念和技术。在任何一种计算机平台上,都能找到高质量和免费的 C++ 语言例子。更重要的是,在 C++ 语言中学到的程序设计思想绝大多数都可以直接用于其他程序设计语言中,如 C#、Java 等。

1.1.1 程序

一个程序应包括如下信息:

①对数据的描述。在程序中要指定数据的类型和数据的组织形式,即数据结构(data structure)。

②对操作的描述,即操作的方法和步骤,也就是算法(algorithm)。

数据是操作的对象,操作的目的是对数据进行加工处理,以便得到结果。

因此,著名计算机科学家沃思(Niklaus Wirth)提出一个公式:

数据结构 + 算法 = 程序

后来人们将其扩充为:

程序 = 算法 + 数据结构 + 程序设计方法 + 语言工具和环境

由此可见,编写程序是让计算机解决实际问题的关键。一般编写计算机程序必须具备两个基本条件:一是掌握一门计算机高级语言的规则,二是要掌握解题的方法和步骤。

1.1.2 程序设计语言

编写程序所使用的语言称为“程序设计语言”,程序设计语言的发展从低级到高级经历了机器语言、汇编语言和高级语言几个阶段,目前的程序大部分使用高级语言编写。

高级语言采用接近自然语言的命令或语句进行编程。它具有学习容易、使用方便、通用性强、移植性好等特点,便于人们学习和应用。早期应用比较广泛的几种高级语言有 BASIC、FORTRAN、PASCAL 和 C 等,在此之后,又诞生了上百种高级程序设计语言,并根据应用领域的不同和语言本身侧重点的差异,分成了许多种。

高级语言的运行方式有三种:解释执行、编译执行、先编译后解释。第一种是通过语言解释程序将源程序解释翻译一句,执行一句,如 BASIC、JavaScript、VBScript、PHP、ASP、Perl 和 PYTHON 等语言,这些语言的共同特点是运行速度慢,但简单。第二种是通过编译程序将源程序编译成二进制目标代码程序,最后执行的是目标代码程序,如 C 语言、C++ 等。显然,后者程序执行效率高、速度快,并且能够脱离语言环境独立运行。第三种是一种新型的语言,是先编译后解释执行的,如 Java、C# 语言等。

20 世纪 60 年代末开始出现的结构化程序设计语言提高了语言的层次。结构化数据、结构化语句、数据抽象、过程抽象等概念使程序更便于体现客观事物的结构和逻辑含义。结构化程序设计是当时程序设计的主流语言,几乎为所有的程序员所接受和使用。它的产生和发展形成了现代软件工程的基础。

面向对象技术从 1986 年以来逐步走出了实验室和研究部门,开始进入实际应用。面向对象语言比面向过程语言更接近于自然语言。程序员设计程序时,基本避开“怎么做”的解题过程,更多地考虑计算机“做什么”,计算机就会完成具体的解题过程。面向对象语言有很多种,可进行如下分类:

①全新的面向对象语言。其代表是 Object-C、Smalltalk 和 Eiffel。这类语言的缺点是,程序员需要从头学习一门全新的语言。

②对传统语言进行面向对象的扩充。这类语言又称为“混合型语言”,其主要代表就是 C++ 语言。

程序设计技术是随着编程语言的发展而发展的,编程语言的发展趋于相互同化,往往彼此相似,所以熟练掌握一种语言后,再学其他语言就容易了。

1.1.3 程序设计

在计算机科学中,把解决问题的方法、步骤描述出来,这个描述的过程就是程序设计。

程序设计与软件设计不是一个概念,它们的侧重点不同。如果说软件设计是工程项目设计,则程序设计是工程中核心部分的实施过程。从另一个角度看,程序设计与软件设计之间又难以找出明显的分界线,交叉地带有很多共同之处。整个程序设计过程为软件开发的一部分。程序设计的一般步骤如下:

(1) 确定数据结构

依据所需要处理的任务要求,规划输入的数据和输出的结果,确定存放数据的数据结构。

(2) 确定算法

算法是指为解决某一特定问题而采取的确定的有限步骤。算法有优有劣,本书实例中介绍的均为一些常用的算法。

(3) 编写程序

在充分论证数据结构和算法以后才能考虑编写程序,编写程序需要结合程序设计方法(面向过程的或是面向对象的)和程序设计语言(C 语言、C++ 语言和 Delphi 等)。