



普通高等教育“十一五”国家级规划教材

21世纪高等学校计算机**专业**实用规划教材

Visual C++面向对象 程序设计教程与实验 (第二版)

温秀梅 丁学钧 主编



清华大学出版社



普通高等教育“十一五”国家级规划教材

21世纪高等学校计算机**专业**实用规划教材

Visual C++面向对象 程序设计教程与实验 (第二版)

温秀梅 丁学钧 主编

清华大学出版社
北京

内 容 简 介

本书在结构上将 C++ 面向对象程序设计的思想和方法作为重点,结合例题进行了详细的分析解释,除在每章后附有习题外,还在附录中整合了实验设计。全书结构严谨、通俗易懂,兼有普及与提高的双重功能。

本书由三部分组成。第一部分第 1~8 章结合实例深入浅出地讲解了 C++ 面向对象程序设计的思想和方法;第二部分第 9~12 章是关于 Visual C++ 的 MFC 程序设计,该内容写得简明扼要,通俗易懂,以便读者理解;第三部分附录包括重要的实验内容设计及 Visual C++ 6.0 环境介绍,这是掌握编程语言的重要环节。

本书遵循少而精的原则,力求做到版面清晰、结构紧凑、信息含量高,因此特别适宜作为计算机专业本科教材。同时,还可以作为自学或函授学习的参考书。

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

Visual C++ 面向对象程序设计教程与实验/温秀梅,丁学钧主编. —2 版. —北京:清华大学出版社,2009.4

(21 世纪高等学校计算机专业实用规划教材)

ISBN 978-7-302-18621-2

I. V… II. ①温… ②丁… III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 147034 号

责任编辑:魏江江

责任校对:时翠兰

责任印制:孟凡玉

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:010-62776969, c-service@tup.tsinghua.edu.cn

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

印 刷 者:北京鑫海金澳胶印有限公司

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:23.25 字 数:574 千字

版 次:2005 年 9 月第 1 版 2009 年 4 月第 2 版

印 次:2009 年 4 月第 5 次印刷

印 数:11501~15500

定 价:33.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:028854-01

编审委员会成员

(按地区排序)

清华大学

周立柱 教授
章 征 教授
王建民 教授
刘 强 副教授
冯建华 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

孟庆昌 教授

北京科技大学

杨炳儒 教授

石油大学

陈 明 教授

天津大学

艾德才 教授

复旦大学

吴立德 教授

吴百锋 教授

同济大学

杨卫东 副教授

苗夺谦 教授

徐 安 教授

张惠娟 副教授

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
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	龚声蓉	教授
	陈建明	副教授
江苏大学	宋余庆	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
	王林平	副教授
	魏开平	副教授
国防科技大学	赵克佳	教授
	肖 侬	副教授
中南大学	陈松乔	教授
	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
云南大学	刘惟一	教授
电子科技大学	刘乃琦	教授
	罗 蕾	教授
重庆邮电学院	王国胤	教授
西南交通大学	曾华燊	教授
	杨 燕	副教授

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、教学内容和教学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。

本系列教材立足于计算机专业课程领域,以专业基础课为主、专业课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 反映计算机学科的最新发展,总结近年来计算机专业教学的最新成果。内容先进,充分吸收国外先进成果和理念。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,融合先进的教学思想、方法和手段,体现科学性、先进性和系统性,强调对学生实践能力的培养,为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。专业基础课和专业课教材配套,同一门课程可以有针对不同层次、面向不同应用的多本具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制定教材规划时依靠各课程专家在调查研究本课程教材

建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主编。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校计算机专业实用规划教材

联系人:魏江江 weijj@tup.tsinghua.edu.cn

第二版前言

《Visual C++ 面向对象程序设计教程与实验》教材 2005 年出版后,受到读者好评,被许多高校选做教材。根据师生反馈意见,我们及时修编原书,形成了本版教材,该教材被列为普通高等教育“十一五”国家级规划教材。

修编后的教材保持了与原书内容、风格一致,使采用原书的教师可以自然地过渡到新版教材。这次修编主要就以下几个方面进行了改进:

根据教学改革、实践教学的需要和教师多年的教学经验,适当修改、增补了第 3 章、第 5 章、第 6 章和第 7 章部分语言和问题描述的表达方式;增加了一套模拟试题,并增加两套模拟试题的答案;修改了部分例题的源程序,对部分例题进行了详细分析;更正了原书中的错误。

作为教材,使用者可以根据自己的需求,选取相应内容进行教学。如果前面开设的是《C++ 语言程序设计》课程,而不是《C 语言程序设计》,则可以略过第 2 章不讲,第 7 章的部分内容也可以不讲。如果课时不够,第 9 章到第 12 章总体通过一个实例讲解,其余的内容可以让学生自学,也可以在课程的开始先布置让学生自己上机通过实例学习。

本版教材由温秀梅、丁学钧任主编,刘建臣、高丽婷、赵巍任副主编。参加编写的有:丁学钧(第 1~2 章)、温秀梅(第 3~10 章、附录 F)、刘建臣(第 11 章)、高丽婷(第 12 章)、李建华(附录 A、B)、宋淑彩(附录 C)、赵巍(附录 D)、祁爱华(附录 E)、全书最后由温秀梅、丁学钧、刘建臣进行审校并统稿。

由于作者水平有限,书中难免有疏漏和错误之处,恳请广大专家和读者指正。

编 者

2009 年 1 月

第一版前言

作为一种计算机语言,C++有很多优点。它既可以进行过程化程序设计,也可以进行面向对象程序设计,很多复杂的算法和设计可以比较容易地用C++面向对象的思想来实现。

在编写本书之前,作者已在高校从事了多年的“C++ 语言程序设计”、“面向对象程序设计”教学及科研工作,对于该语言的概念、功能及应用有着较深入的理解和丰富的实践经验。在教学过程中,我们发现很多教材在讲解C++语言时既包括结构化程序设计又包括面向对象程序设计,而在面向对象程序设计部分讲得不透彻,不适合计算机专业的学生学习。故组织编写了这本教材,旨在通过本教材在内容安排、教学深度及实验要求等方面满足计算机专业本科生“面向对象程序设计”课程的教学要求。

作为一本教材,本书具有如下特点:

(1) 本书在结构上将C++面向对象程序设计的思想和方法作为重点,并结合例题进行了详细的分析解释,除在每章后附有习题外,还在附录中整合了实验设计。使全书结构严谨、通俗易懂,兼有普及与提高的双重功能。

(2) 本书没有涉及面向过程的程序设计内容,只在第2章中讲解了C++在结构化程序设计方面对C的扩充,因此学生应在学习了相关的基础知识之后再使用本教材。

(3) 本书以现代教育理念为指导,在讲授方式上注意结合应用开发实例,注重培养学生理解面向对象程序设计思想,以提高分析问题和解决实际问题的能力。

(4) 本书中的所有程序都是在VC 6.0环境下编译调试通过的。

本书由温秀梅、丁学钧主编并统稿,孟凡兴、刘建臣任副主编。参加编写的有:丁学钧(第1~2章),温秀梅(第3~11章、附录E),孟凡兴(第12章),李建华(附录A、B),宋淑彩(附录C),周丽莉(附录D),刘建臣担任本书的审校工作。参加本书部分内容编写工作的还有赵巍、徐晓君、岳杰、庞慧、董颖霞、王庆林、司亚超、刘海龙等。在本书的大纲讨论和分工编写过程中,我们始终互相帮助,彼此鼓励,是一次非常难忘的经历。

在此还要特别感谢我们的学生梁金龙,他为本书做了很多前期工作。

由于时间仓促,加之水平有限,书中难免有疏漏和错误之处,恳请广大读者和专家指正。

编者

目 录

第 1 章 绪论	1
1.1 面向对象方法的起源	1
1.2 面向对象是软件方法学的返璞归真	2
1.3 结构化程序设计与面向对象程序设计	3
1.4 面向对象的基本概念和面向对象系统的特性	5
1.4.1 面向对象的基本概念	5
1.4.2 面向对象系统的特性	6
1.5 面向对象程序设计语言的四大家族	7
1.6 面向对象的系统开发方法	8
1.6.1 面向对象分析 OOA	9
1.6.2 面向对象设计 OOD	10
1.6.3 OOA 和 OOD 的基本步骤	11
1.7 面向对象程序设计举例	12
习题	14
第 2 章 C++ 语言对 C 语言的扩充	15
2.1 C++ 语言的特点	15
2.2 C++ 语言的文件扩展名	15
2.3 注释符	16
2.4 名字空间	16
2.5 C++ 语言的输入输出	17
2.6 变量的定义	18
2.7 强制类型转换	18
2.8 动态内存的分配与释放	19
2.9 作用域运算符(::)	22
2.10 引用	22
2.11 const 修饰符	27
2.12 字符串	28
2.13 C++ 语言中函数的新特性	29
2.13.1 函数原型(function prototype)	29

2.13.2	内联(inline)函数	29
2.13.3	带默认参数的函数	30
2.13.4	函数重载(overload)	31
2.13.5	函数模板(function template)	33
习题		36
第3章	类和对象	38
3.1	类	38
3.1.1	类的定义	38
3.1.2	类中成员函数的定义	40
3.2	对象	43
3.3	构造函数和析构函数	49
3.3.1	构造函数	50
3.3.2	析构函数	62
3.4	类的聚集——对象成员	64
3.5	静态成员	66
3.6	指向类成员的指针	73
3.7	综合举例	76
习题		81
第4章	友元	83
4.1	友元的概念和定义	83
4.2	友元函数	85
4.3	友元成员	91
4.4	友元类	92
4.5	友元综合举例	95
习题		97
第5章	继承与派生	98
5.1	单一继承	98
5.1.1	继承与派生	98
5.1.2	派生类的定义	99
5.1.3	类的继承方式	101
5.1.4	派生类的构造函数和析构函数	105
5.1.5	派生类对基类成员的继承	111
5.2	多重继承	112
5.2.1	多重继承的概念和定义	112
5.2.2	二义性和支配规则	113
5.2.3	赋值兼容规则	114

5.3	虚基类	115
5.3.1	虚基类的概念	115
5.3.2	多重继承的构造函数和析构函数	116
5.4	类模板	119
5.5	应用举例	123
	习题	132
第6章	多态性和虚函数	134
6.1	运算符重载	134
6.1.1	运算符重载概述	134
6.1.2	用成员函数重载运算符	135
6.1.3	用友元函数重载运算符	140
6.1.4	几个常用运算符的重载	145
6.2	虚函数	155
6.2.1	为什么要引入虚函数	155
6.2.2	虚函数的定义与使用	156
6.3	纯虚函数和抽象类	167
6.3.1	纯虚函数的概念	167
6.3.2	抽象类的概念	167
6.4	虚析构函数	169
	习题	171
第7章	C++ 语言的输入输出流库	174
7.1	C++ 语言标准输入输出	174
7.1.1	C++ 语言输入输出流库简介	174
7.1.2	C++ 语言格式化输入输出	176
7.2	用户自定义数据类型的 I/O 流	184
7.3	文件输入输出流	187
7.3.1	文件 I/O 流	187
7.3.2	文件的打开与关闭	187
7.3.3	文件的读写操作	190
	习题	196
第8章	异常处理	197
8.1	异常处理概述	197
8.2	C++ 语言异常处理的实现	198
8.3	重新抛出异常和异常规范	204
8.4	C++ 标准库中的异常类	205
	习题	206

第 9 章 Windows 编程基础和 MFC 编程基础	207
9.1 Windows 编程基础	207
9.2 MFC 编程基础	213
9.2.1 MFC 编程概述	213
9.2.2 MFC 的类层次	214
9.2.3 常用的 MFC 类	220
9.2.4 MFC 应用程序的消息映射	235
9.2.5 一个最简单的 MFC 应用程序	237
9.2.6 典型的 Windows 应用程序	239
习题	241
第 10 章 对话框和控件	242
10.1 对话框和控件的基本概念	242
10.1.1 对话框的基本概念	242
10.1.2 控件的基本概念	243
10.2 使用 AppWizard 开发 MFC 应用程序	244
10.2.1 生成基于对话框的 MFC 应用程序框架	245
10.2.2 AppWizard 向导自动生成的文件	250
10.3 基本控件	252
10.3.1 按钮控件	252
10.3.2 编辑框控件(文本框控件)	253
10.3.3 静态控件	255
10.3.4 列表框控件	255
10.3.5 滚动条控件	256
10.3.6 组合框控件	257
10.3.7 基本控件应用举例	258
10.4 通用对话框	274
10.4.1 CColorDialog 类	275
10.4.2 CFileDialog 类	275
10.4.3 CFindReplaceDialog 类	277
10.4.4 CFontDialog 类	277
10.4.5 CPrintDialog 类	278
10.4.6 通用对话框应用举例	279
习题	281
第 11 章 菜单和文档/视图结构	282
11.1 文档/视图的概念	282
11.2 文档类	283

11.3	视图类	284
11.4	菜单	285
11.5	菜单和文档/视图结构程序设计举例	286
	习题	294
第 12 章	图形设备接口	295
12.1	设备环境	295
12.2	映射模式	296
12.3	绘制基本图形	297
12.4	画笔和画刷	299
12.4.1	画笔	299
12.4.2	画刷	301
12.4.3	画笔和画刷的应用程序举例	302
12.5	字体	303
	习题	305
附录 A	程序的调试与运行	306
附录 B	标准字符 ASCII 表	327
附录 C	实验	329
附录 D	模拟考试题一	335
附录 E	模拟考试题二	345
附录 F	参考课时安排	353
	主要参考文献	354

面向对象程序设计是软件系统设计与实现的新方法,这种方法是通过增加软件的可扩充性和可重用性来提高程序员的生产能力,控制软件的复杂性,降低软件维护的开销。因此,它的应用使软件开发的难度和费用大幅度降低,已为世界软件产业带来了革命性的突破。

1.1 面向对象方法的起源

“对象”一词在现实生活中经常会遇到,它表示现实世界中的某个具体的事物。社会的进步和计算机科学的发展是相互促进的,随着计算机的普及和应用,人们越来越希望能更直接地与计算机进行交互,而不需要经过专门学习和长时间训练后才能使用它。这使得软件设计人员的负担越来越重,软件的实现越来越复杂,并且对计算机领域自身的发展也提出了新的要求。利用传统的程序设计思想无法满足这一要求时,人们就开始寻求一种能帮助人类解决问题的自然方法,这就是“面向对象”技术。

20 世纪 50 年代的程序都是用指令代码或汇编语言编写的,这种程序的设计相当复杂,编制和调试一个稍大一点的程序常常要花费很长时间,培养一个熟练的程序员更需经过长期训练和实践,这种局面严重影响了计算机的普及和应用。

20 世纪 60 年代高级语言的出现大大简化了程序设计,缩短了软件开发周期,显示出了强大的生命力。此后,编制程序已不再是专业软件人员才能做的事了,一般工程技术人员花较短的时间学习后,也可以使用计算机解题。这个时期,随着计算机日益广泛地渗透到各个学科和技术领域,一系列不同风格、为不同目标服务的程序设计语言发展起来了,其中较为著名的有 FORTRAN、COBOL、ALGOL、LISP、PASCAL 等十几种语言。高级语言的蓬勃兴起,使得编译原理和形式语言理论日趋完善,这是该时期的主要特征。但是就整个程序设计方法而言,并无实质性的改进。

自 20 世纪 60 年代末到 20 世纪 70 年代初,出现了大型软件系统,如操作系统、数据库,这给程序设计带来了新的问题。大型系统的研制需要花费大量的资金和人力,可是研制出来的产品却可靠性差、错误多、不易维护和修改。一个大型操作系统有时需要每年几千人的工作量,而所获得的系统又常常会隐藏着几百甚至几千个错误。当时,人们称这种现象为“软件危机”。

为了克服 20 世纪 60 年代出现的软件危机,1968 年北约组织提出“软件工程”的概念。对程序设计语言的认识从强调表达能力为重点转向以结构化和简明性为重点,将程序从语句序列转向相互作用的模块集合。1969 年, E. W. Dijkstra 首先提出了结构化程序设计的

概念,他强调从程序结构和风格上来研究程序设计。在软件工程迫切需要改进的背景下,20 世纪70 年代结构化语言获得蓬勃发展并得到广泛应用。使用结构化程序设计方法可显著地减少软件的复杂性,提高软件的可靠性、可测试性和可维护性。经过几年的探索和实践,结构化程序设计的应用确实取得了成效,用结构化程序设计的方法编写出来的程序不仅结构良好、易写易读,而且易于验证其正确性。

进入 20 世纪 80 年代,由于一系列高技术的研究,如第五代计算机、计算机辅助制造(CAM)和知识工程等领域的研究,都迫切要求大型的软件系统作支撑。它们所用的数据类型也超出了常规的结构化数据类型的范畴,并且提出了对图像、声音、规则等非结构化信息的管理。为了满足这些应用领域的需要,就迫切要求软件模块具有更强的独立自治性,以便于大型软件的管理、维护和重用。由于结构化语言的数据类型较为简单,采用过程调用机制也不够灵活,独立性较差,所以不能胜任对非结构化数据的定义与管理。

为了适应高技术发展的需要,消除结构化程序设计语言的局限,自 20 世纪 80 年代以来,出现了面向对象程序设计流派,研制出了多种面向对象程序设计语言(Object Oriented Programming Language, OOPL),如 Ada、Smalltalk、C++ 语言和当前使用在 Internet 上与平台无关的 Java 语言等。

由于 OOPL 的对象、类具有高度的抽象性,所以能很好地表达复杂的数据类型,并且 OOPL 也允许程序员灵活地定义自己所需要的数据类型。类本身具有完整的封装性,可以使用它作为编程中的模块单元,满足模块独立自治的要求。另外,类的继承性和多态性功能更有助于简化大型软件和大量重复定义的模块,从而增强了模块的可重用性,提高了软件的可靠性,缩短了软件的开发周期。

1.2 面向对象是软件方法学的返璞归真

客观世界是由许多具体事物、抽象概念、规则等组成的,人们将任何感兴趣或要加以研究的事、物、概念统称为对象(object)。每个对象都有各自的内部状态和运动规律,不同对象之间通过消息传递进行相互作用和联系就构成了各种不同的系统。面向对象的方法正是以对象作为基本元素的一种分析问题和解决问题的方法。

传统的结构化方法强调的是功能抽象和模块化,每个模块都是一个过程,结构化方法处理问题是以过程为中心的。对象包含数据和对数据的操作,是对数据和功能的抽象和统一。而面向对象强调的是功能抽象和数据抽象,用对象来描述事物和过程,面向对象方法处理问题的过程是对一系列相关对象的操纵,即发送消息到目标对象,由目标对象执行相应的操作。因此面向对象方法是以对象为中心的,这种以对象为中心的方法更自然、更直接地反映现实世界的问题空间,从而具有独特的抽象性、封装性、继承性和多态性的特点,更好地适应了复杂大系统不断发展与变化的要求。

采用对象的观点看待所要解决的问题,并将其抽象为应用系统是极其自然与简单的,因为它符合人类的思维习惯,使得应用系统更容易理解。同时,由于应用系统是由相互独立的对象构成的,系统的修改可以局部化,因此系统维护更加容易。

软件开发从本质上讲就是对软件所要处理的问题域进行正确的认识,并把这种认识正确地描述出来。既然如此,那就应该直接面对问题域中客观存在的事物来进行软件开发,这

就是面向对象。另一方面,人类在认识世界的过程中形成的普遍有效的思维方法,在软件开发中也是适用的。在软件开发中尽量采用人们日常生活中习惯的思维方式和表达方式,这就是面向对象方法所强调的基本原则。软件开发从过分专业化的方法、规则和技巧中脱离出来,并重新回到了客观世界,回到了人们的日常思维当中,所以说面向对象方法是软件方法学的返璞归真。

1.3 结构化程序设计与面向对象程序设计

要想真正了解面向对象程序设计,首先需要回顾一下结构化程序设计的含义。

1. 结构化程序设计

结构化程序设计是20世纪60年代诞生的,在70年代到80年代已遍及全球,成为软件开发设计所有领域及每个程序员都采用的程序设计方法,它的产生和发展形成了现代软件工程的基础。

结构化程序设计的设计思想是:自顶向下、逐步求精;其程序结构按功能划分为若干个基本模块,这些模块形成一个树状结构;各模块之间的关系尽可能简单,在功能上相对独立;每一模块内部均由顺序、选择和循环三种基本结构组成;其模块化实现的具体方法是使用子程序、过程或函数。

结构化程序设计由于采用了模块分解和功能抽象、自顶向下、分而治之的手段,从而有效地将一个复杂的软件系统的设计任务分成许多容易控制和处理的子任务,这些子任务都是可独立编程实验的子程序模块。每一个子程序都有一个清晰的界面,使用起来非常方便。

结构化程序设计方法虽然具有许多的优点,但它仍是一种面向过程的设计方法,它把数据和过程分离为相互独立的实体,程序员在编程时必须时刻考虑所要处理的数据格式。对于不同的数据格式即使做同样的处理或对相同的数据格式做不同的处理都需要编写不同的程序。因此结构化程序的可重用性不好;另一方面,当数据和过程相互独立时,总存在着用错误的数据调用正确的程序模块或用正确的数据调用错误的程序模块的可能性。因此,要使数据和程序始终保持相容,已成为程序员的一个沉重负担,并且随着软件系统的规模越来越大,程序的复杂性越来越难以控制。上述这些问题,结构化程序设计方法本身是解决不了的,需要借助于下面要讨论的面向对象程序设计方法给予解决。

程序设计的任务是描述问题并解决问题,在结构化程序设计中可以用下面的式子表示程序:

$$\text{程序} = \text{数据结构} + \text{算法} + \text{程序设计语言} + \text{语言环境}$$

图1.1所示为结构化程序设计中程序的结构。

2. 面向对象程序设计——程序设计的新思维

面向对象程序设计既吸取了结构化程序设计的一切优点,又考虑了现实世界与面向对象空间的映射关系,它所追求的目标是将现实世界问题的求解尽可能简单化。

面向对象程序设计将数据及对数据的操作放在一起,作为一个相互依存、不可分割的整体来处理,它采用了数据抽象和信息隐藏技术。它将对象及对对象的操作抽象成一种新的数据类型——类,并且考虑不同对象之间的联系和对象所在类的可重用性。