



工业和信息化普通高等教育“十二五”规划教材立项项目

21 世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

C++ 程序设计

C++ Programming Design

姚琳 主编

李小燕 汪红兵 副主编

- 概念阐述注意演绎与归纳
- 案例讲解侧重方法与应用
- 编程思想兼顾理论与实践



高校系列



人民邮电出版社
POSTS & TELECOM PRESS



工业和信息化部普通高等教育“十二五”规划教材立项项目
21 世纪高等学校计算机规划教材

C++ 程序设计

C++ Programming Design

姚琳 主编

李小燕 汪红兵 副主编

屈微 黄晓璐 段世红 徐惠民 齐悦 编著



高校系列

人民邮电出版社

北京

图书在版编目 (C I P) 数据

C++程序设计 / 姚琳主编. -- 北京 : 人民邮电出版社, 2011. 3
21世纪高等学校计算机规划教材
ISBN 978-7-115-24852-7

I. ①C… II. ①姚… 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(2011)第013670号

内 容 提 要

本书根据教育部非计算机专业计算机基础课程教学指导分委员会提出的《高等学校非计算机专业计算机基础课程教学基本要求》中的关于“程序设计”的课程教学要求编写。全书以面向对象的编程思想为主线, 主要讲解程序设计基础知识、类与对象的基本概念、继承与多态、输入/输出流以及泛型程序设计等内容。

本书可作为高等院校“程序设计”课程的教材使用, 也可作为学习程序设计的自学参考书和培训教材。

21 世纪高等学校计算机规划教材

C++程序设计

-
- ◆ 主 编 姚 琳
副 主 编 李小燕 汪红兵
编 著 屈 微 黄晓璐 段世红 徐惠民 齐 悦
责任编辑 武恩玉
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
大厂聚鑫印刷有限责任公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 21.5 2011 年 3 月第 1 版
字数: 564 千字 2011 年 3 月河北第 1 次印刷

ISBN 978-7-115-24852-7

定价: 38.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223
反盗版热线: (010)67171154

出版者的话

现今社会对人才的基本要求之一就是应用计算机的能力。在高等学校,培养学生应用计算机的能力,主要是通过计算机课程的体制改革,即计算机教学分层、分类规划与实施;密切联系实际,恰当体现与各专业其他课程配合;教学必须以市场需求为导向,目的是培养高素质创新型人才。

人民邮电出版社经过对教学改革新形势充分的调查研究,依据目前比较成熟的教学大纲,组织国内优秀的有丰富教学经验的教师编写一套体现教学改革最新形势的“高校系列计算机教材”。在本套教材的出版过程中,我社多次召开教材研讨会,广泛听取了一线教师的意见,也邀请众多专家对大纲和书稿做了认真的审读与研讨。本套教材具有以下特点。

1. 覆盖面广,突出教改特色

本套教材主要面向普通高等学校(包括计算机专业和非计算机专业),是在经过大量充分的调研基础上开发的计算机系列教材,涉及计算机教育领域中的所有课程(包括专业核心骨干课程与选修课程),适应了目前经济、社会对计算机教育的新要求、新动向,尤其适合于各专业计算机教学改革的特点特色。

2. 注重整体性、系统性

针对各专业的特点,同一门课程规划了组织结构与内容不同的几本教材,以适应不同教学需求,即分别满足不同层次计算机专业与非计算机专业(如工、理、管、文等)的课程安排。同时本套教材注重整体性的策划,在教材内容的选择上避免重叠与交叉,内容系统完善。学校可根据教学计划从中选择教材的各种组合,使其适合本校的教学特点。

3. 掌握基础知识,侧重培养应用能力

目前社会对人才的需要更侧重于其应用能力。培养应用能力,须具备计算机基础理论、良好的综合素质和实践能力。理论知识作为基础必须掌握,本套教材通过实践教学与实例教学培养解决实际问题的能力和知识综合运用能力。

4. 教学经验丰富的作者队伍

高等学校在计算机教学和教材改革上已经做了大量的工作,很多教师在计算机教育与科研方面积累了相当多的宝贵经验。本套教材均由有丰富教学经验的教师编写,并将这些宝贵经验渗透到教材中,使教材独具特色。

5. 配套资源完善

所有教材均配有 PPT 电子教案,部分教材配有实践教程、题库、教师手册、学习指南、习题解答、程序源代码、演示软件、素材、图书出版后要更新的内容等,以方便教与学。

我社致力于优秀教材的出版,恳请大家在使用的过程中,将发现的问题与提出的意见反馈给我们,以便再版时修改。

随着计算机科学技术的发展和计算应用技术的普及,越来越多的人需要操作和使用计算机。一般来说,操作和使用计算机分为两个层次。一是基于商业软件,进行简单的鼠标点击和键盘输入等基本操作。例如,使用 Microsoft Excel 进行业务数据处理,包括排序、分类汇总、绘制趋势线、分布图和饼图等。二是使用计算机语言,如 C++、C 和 Java 等进行定制开发。例如,使用 C++语言编制一个聊天应用程序,使用 Java 语言编制一个计算器应用程序等。现实生活中,大部分人对计算机的使用都属于第 1 个层次。而科研工作人员和技术开发人员等往往面临很多个性化的应用,这些应用不能简单地使用商业软件进行处理,必须自己开发应用软件来满足个性化需求,这属于第 2 个层次。作为高等学校的大学生,绝大多数人都越来越多地面临第 2 个层次的使用需求。因此,掌握一门计算机语言成为高等学校对大学生培养的基本要求。

众多的计算机语言中,如 C++、C、Java、C#和 VB 等,C++是一门最为复杂的计算机编程语言。造成这种复杂性的原因是,C++是一门兼有面向过程方法和面向对象方法的混合编程语言。一方面,C++这种混合编程语言的特点使得我们起步开始学习这门编程语言相比较学习其他编程语言要困难很多。另一方面,C++作为一门混合编程语言,涉及了面向过程方法的很多基本概念,同时也涉及了面向对象方法的很多基本概念,可以说,C++编程语言中涵盖了几乎所有编程语言的基本概念,使得我们在学习其他编程语言时,尤其是当下非常流行且得到大规模应用的 C#和 Java 语言时,能够进行快速的知识迁移。从这个角度来说,C++作为学习编程语言的入门是非常合适的。

C++编程语言具有以下特点。

(1) 接近汇编语言或者硬件。C++可以通过指针直接访问内存地址,并可以进行二进制运算,这有利于编写系统软件。

(2) 具有丰富的数据类型。C++可以使用系统定义的数据类型,如字符型、整型、浮点型,也可以自定义数据类型,如结构类型和类等。

(3) 良好的可移植性。使用 C++编写的程序可以运行在各种不同类型的计算机上。

(4) 面向对象的描述能力。C++可以基于面向对象方法,以类和对象为基本组织元素来描述系统。

本书根据教育部非计算机专业计算机基础课程教学指导分委员会提出的《高等学校非计算机专业计算机基础课程教学基本要求》中的关于“程序设计”课程教学要求编写,作为一门介绍面向对象编程语言的书籍,强调的是 C++编程语言的面向对象的特点。希望读者能够以如下的层次关系来理解本书内容。

(1) 基于面向对象的思想,使用计算机编程语言开发系统时以类和对象为基本组成元素,而一个完整的类包括数据部分和函数部分,这部分内容体现为第 1

章和第 8 章。

(2) 数据部分使用各种数据类型进行描述, 这部分内容体现为第 2 章的数据类型、第 4 章的数组、第 6 章的指针和引用以及第 7 章的自定义数据类型。

(3) 函数部分主要涉及多个操作如何构成一个完整的操作序列, 这部分内容体现为第 3 章的 C++ 控制语句和第 5 章的函数。

(4) 基于面向对象的思想开发系统时, 除了需要描述类的组成元素, 如数据部分和函数部分外, 还需要描述类与类之间的各种关系, 这部分内容体现为第 9 章继承和派生以及第 10 章多态性。

(5) 最后, 第 12 章、第 13 章和第 14 章是 C++ 编程语言中一些与实际应用开发相关的重要内容。

本书由姚琳主编, 共 14 章。其中的第 1、8、12 章由汪红兵编写; 第 2 章由姚琳编写; 第 3 章由姚琳、齐悦编写; 第 4、5 章由屈微编写; 第 6 章由黄晓璐编写; 第 7、9、10、11 章由李小燕编写; 第 13 章由段世红编写; 第 14 章由徐惠民、姚琳编写。全书由姚琳最后审阅统稿。

由于时间仓促, 书中错误再所难免, 恳请读者不吝赐教!

编者

2010 年 12 月

目 录

第 1 章 程序设计方法和 C++ 语言

概述1

- 1.1 程序编写过程1
- 1.2 面向过程程序设计方法2
- 1.3 面向对象程序设计方法3
- 1.4 C++ 语言的发展5
- 1.5 C++ 语言程序的开发过程7
- 1.6 一个简单的程序7
- 本章小结10
- 习题10

第 2 章 数据类型、运算符和

表达式11

- 2.1 数据类型11
- 2.2 常量和变量13
 - 2.2.1 C++ 的符号系统13
 - 2.2.2 C++ 的常量14
 - 2.2.3 C++ 的变量17
- 2.3 运算符和表达式18
 - 2.3.1 算术运算符和算术表达式18
 - 2.3.2 赋值运算符和赋值表达式20
 - 2.3.3 关系运算符和关系表达式21
 - 2.3.4 逻辑运算符和逻辑表达式22
 - 2.3.5 位运算符和位运算表达式24
 - 2.3.6 条件运算符和条件表达式27
 - 2.3.7 其他运算符27
- 2.4 数据类型转换29
 - 2.4.1 自动类型转换29
 - 2.4.2 强制类型转换30
 - 2.4.3 赋值表达式的类型转换31
- 2.5 基本输入/输出31
 - 2.5.1 标准输入流和标准输出流31
 - 2.5.2 I/O 流的格式控制32
- 2.6 C++ 的语句类型34
 - 2.6.1 说明性语句35
 - 2.6.2 可执行语句35

本章小结37

习题38

第 3 章 C++ 控制语句40

- 3.1 顺序结构程序设计40
- 3.2 分支结构程序设计42
 - 3.2.1 if 分支语句42
 - 3.2.2 if 语句的嵌套48
 - 3.2.3 switch 语句51
- 3.3 循环结构程序设计53
 - 3.3.1 while 语句53
 - 3.3.2 do...while 语句55
 - 3.3.3 for 语句58
 - 3.3.4 3 种循环语句的比较61
 - 3.3.5 循环嵌套62
 - 3.3.6 break 和 continue 语句63
- 3.4 程序举例66
- 本章小结68
- 习题69

第 4 章 数组72

- 4.1 概述71
- 4.2 一维数组72
 - 4.2.1 一维数组定义和初始化72
 - 4.2.2 一维数组元素的引用74
- 4.3 二维数组75
 - 4.3.1 二维数组定义和初始化75
 - 4.3.2 二维数组元素的引用76
- 4.4 字符数组与字符串77
 - 4.4.1 字符数组的定义和初始化77
 - 4.4.2 字符数组的引用78
- 4.5 数组应用举例80
 - 4.5.1 一维数组应用举例80
 - 4.5.2 二维数组应用举例81
 - 4.5.3 字符数组应用举例83
 - 4.5.4 综合应用举例84

本章小结	85	6.2.1 指针与一维数组	135
习题	85	6.2.2 指针与二维数组	138
第 5 章 函数与预处理	91	6.2.3 指向字符串的指针变量	140
5.1 概述	91	6.2.4 指针数组	143
5.1.1 函数简介	91	6.2.5 多级指针	146
5.1.2 函数的种类	92	6.3 指针和函数	147
5.2 函数定义及调用	93	6.3.1 指针型函数	147
5.2.1 函数的定义	93	6.3.2 用函数指针调用函数	148
5.2.2 函数的调用	95	6.3.3 用指向函数的指针作函数 参数	150
5.2.3 函数参数传递与返回值	96	6.3.4 带参数的 main()函数	151
5.2.4 函数的嵌套调用	100	6.4 动态存储分配	152
5.2.5 函数原型声明	102	6.4.1 内存的动态分配	152
5.3 C++中的特殊函数	103	6.4.2 动态内存分配操作符	153
5.3.1 重载函数	103	6.5 引用	154
5.3.2 内联函数	105	6.5.1 引用的概念	154
5.3.3 具有默认参数值的函数	106	6.5.2 引用的操作	155
5.4 函数模板	107	6.5.3 不能被定义引用的情况	156
5.4.1 函数模板的定义	107	6.5.4 函数参数中引用的传递	157
5.4.2 重载函数模板	108	6.5.5 用引用返回多个值	158
5.5 局部变量和全局变量	109	6.5.6 用函数返回引用	159
5.5.1 局部作用域和局部变量	109	6.5.7 const 引用	159
5.5.2 全局作用域和全局变量	111	本章小结	161
5.6 变量的生存期和存储类别	113	习题	162
5.6.1 变量的生存期	113	第 7 章 其他自定义数据类型	168
5.6.2 变量的存储类别	113	7.1 结构体类型	168
5.7 编译预处理	117	7.1.1 结构体类型的定义	168
5.7.1 宏定义	118	7.1.2 结构体类型变量的定义及其 初始化	169
5.7.2 文件包含	120	7.1.3 结构体类型的使用	171
5.7.3 条件编译	121	7.2 枚举类型	175
本章小结	121	7.3 共用体类型	178
习题	121	本章小结	180
第 6 章 指针和引用	127	习题	181
6.1 指针的概念	127	第 8 章 类与对象	185
6.1.1 指针和指针变量	127	8.1 类的概念	185
6.1.2 指针变量的定义	127	8.2 类的定义	186
6.1.3 指针的基本运算	128	8.3 对象的定义	187
6.1.4 指针作为函数参数	132		
6.2 指针与数组	135		

8.4 类的成员函数	188	9.4.2 虚基类	235
8.5 类的访问属性	191	本章小结	236
8.6 对象的使用	192	习题	237
8.6.1 对象指针	194	第 10 章 多态性与虚函数	242
8.6.2 对象引用	194	10.1 多态性	242
8.6.3 this 指针	194	10.1.1 多态性的概念	242
8.6.4 对象数组	195	10.1.2 多态的实现——联编	243
8.6.5 普通对象做函数参数	196	10.2 继承中的静态联编	243
8.6.6 对象指针做函数参数	197	10.2.1 派生类对象调用同名函数	243
8.6.7 对象引用做函数参数	199	10.2.2 通过基类指针调用同名函数	245
8.7 构造函数	201	10.3 虚函数和运行时的多态	247
8.8 析构函数	202	10.3.1 虚函数	247
8.9 拷贝构造函数	204	10.3.2 虚函数的使用	248
8.10 浅拷贝和深拷贝	205	10.3.3 虚析构函数	249
8.11 静态成员	206	10.4 纯虚函数和抽象类	251
8.11.1 静态成员数据	206	10.4.1 纯虚函数	251
8.11.2 静态成员函数	208	10.4.2 抽象类	252
8.12 友元	210	10.4.3 应用实例	252
8.12.1 友元函数	210	本章小结	255
8.12.2 友元类	211	习题	255
8.13 常对象	212	第 11 章 运算符重载	260
8.14 常成员	212	11.1 运算符重载的概念	260
8.14.1 常成员数据	213	11.2 运算符重载的规则和语法	261
8.14.2 常成员函数	213	11.3 “++”、“—” 运算符的重载	265
8.15 组合关系	214	11.4 赋值运算符“=”的重载	268
8.16 类模板	216	11.5 插入提取运算符“<<”“>>”的 重载	271
本章小结	218	11.6 类型转换运算符的重载	272
习题	218	本章小结	274
第 9 章 继承与派生	223	习题	274
9.1 继承和派生的概念	223	第 12 章 标准模板库	275
9.2 继承的实现	224	12.1 标准模板库概述	275
9.2.1 派生类的定义	224	12.2 容器	276
9.2.2 派生类的构成	226	12.2.1 向量	276
9.2.3 继承的访问控制	228	12.2.2 列表	279
9.3 派生类的构造函数和析构函数	230	12.2.3 栈	280
9.3.1 派生类的构造函数	230	12.2.4 集合	282
9.3.2 派生类的析构函数	232		
9.4 多继承中的二义性与虚函数	233		
9.4.1 多继承中的二义性	233		

12.2.5 映射	284	第 14 章 异常处理	308
本章小结	285	14.1 程序的出错处理	308
习题	285	14.2 异常及异常处理	310
第 13 章 输入/输出流	287	14.2.1 异常及其特点	310
13.1 流	287	14.2.2 异常处理方法	310
13.2 文件流	288	14.3 C++异常处理机制	311
13.2.1 数据的层次	288	14.3.1 C++异常处理的基本过程	311
13.2.2 文件和流	290	14.3.2 C++异常处理的其他形式	315
13.2.3 文件操作	290	14.4 用 exception 类处理异常	319
13.3 顺序文件操作	294	14.4.1 C++的 exception 类	319
13.3.1 建立顺序文件	294	14.4.2 用户自定义类的对象传递异常	321
13.3.2 读取顺序访问文件中的数据	297	本章小结	324
13.3.3 更新顺序访问文件	298	习题	324
13.4 随机访问文件	299	附录 A C++语言中运算符的 优先级和结合性	327
13.4.1 建立随机访问文件	299	附录 B ASCII 码表	329
13.4.2 向随机访问文件中随机地写入 数据	301	附录 C C++常用函数	330
13.4.3 从随机访问文件中顺序地读取 数据	302	参考文献	334
本章小结	304		
习题	305		

第 1 章

程序设计方法和 C++ 语言概述

【本章内容提要】

C++语言是一种同时支持面向过程程序设计方法和面向对象程序设计方法的优秀编程语言，已经在计算机相关的工程领域得到了广泛应用。本章从程序编写过程入手，介绍两种程序设计方法，进而明确 C++语言能够同时支持这两种程序设计方法。

【本章学习重点】

程序编写过程；两种不同的程序设计方法：面向过程程序设计方法和面向对象程序设计方法；一种同时支持面向过程程序设计方法和面向对象程序设计方法的 C++语言及其发展历程；使用 C++语言编写程序的过程；一个简单的入门程序 Hello World。

1.1 程序编写过程

程序编写过程就是使用某种计算机语言对要解决的问题进行描述，然后形成能在计算机上运行的程序代码。计算机语言是人类和计算机之间进行交流的媒介工具，它是计算机能够识别的语言，人类为了要与之进行交互，必须学习计算机语言。这如同我们学习英语是为了与外国人进行交流一样。

人类在使用计算机语言描述现实世界问题之前，一般需要在头脑中形成关于该问题的框架结构。这种框架结构首先在大脑中形成，然后使用某种工具（例如，Microsoft Visio 和 Microsoft Word）加以描述。一个完整的程序编写过程，如图 1-1 所示。其中，现实世界是计算机要解决问题的真实世界，例如，开发一个学生选课信息系统，学生、教师和课程等组成了真实世界；概念世界是头脑世界，是人类在使用计算机语言解决现实世界问题之前，在大脑中形成的关于现实世界问题的框架结构，这种结构是人类认识现实世界的结果，而这种结构的具体形式与我们采用的程序设计方法密切相关；计算机世界由具体的计算机语言组成，它是对概念世界问题结构的描述。

概念世界的问题结构与采用的程序设计方法有关。因此，程序设计方法实际是人类从计算机解决实际问题的角度认识现实世界问题所采用的思考方法。当前，典型的程序设计方法包括面向过程程序设计和面向对象程序设计。

计算机语言只是一种工具，用来实现程序设计方法所观察的概念世界的框架结构。本书要特别注意区分计算机语言和程序设计方法的不同。

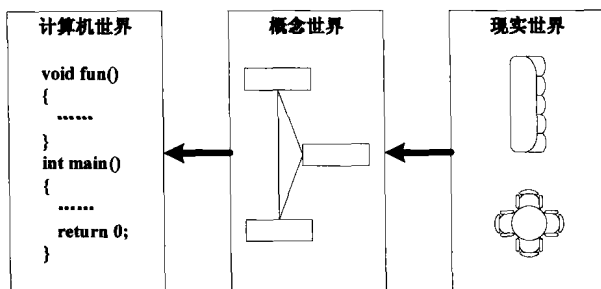


图 1-1 程序编写过程

1.2 面向过程程序设计方法

面向过程程序设计方法又称为结构化程序设计方法。其基本观点是：使用 3 种基本结构（即顺序结构、选择结构和循环结构）强调模块的单人 and 单出。

采用面向过程程序设计方法来认识现实世界问题，其基本的思维粒度是过程。所谓过程，指的是一个功能的完整描述。在没有特别说明的情况下，本书中的过程和功能是等价的说法。例如，开发一个学生选课管理信息系统时，经过需求调研后，发现该系统的基本功能如下。

- ① 教师信息的登记，例如，当学校新增教师时，需要该功能。
 - ② 教师信息的修改，例如，当教师转移工作岗位时，需要该功能。
 - ③ 教师信息的删除，例如，当学校开除教师时，需要该功能。
 - ④ 教师信息的打印，例如，当学校存档教师档案时，需要该功能。
 - ⑤ 教师查询学生名单，例如，当教师需要查看授课学生名单时，需要根据课程查询学生名单的功能。
 - ⑥ 学生信息的登记，例如，当新生入学时，需要该功能。
 - ⑦ 学生信息的修改，例如，当学生转系时，需要该功能。
 - ⑧ 学生信息的删除，例如，当学生毕业时，需要该功能。
- 注意，这种删除只是一种临时性的删除，经过删除后的学生信息可以实现归档处理。
- ⑨ 学生信息的打印，例如，当学校存档学生档案时，需要该功能。
 - ⑩ 学生查询教师名单，例如，当学生选课需要查看某个课程的授课教师时，需要该功能。
 - ⑪ 课程信息的增加，例如，当学校开设一门新课时，需要该功能。
 - ⑫ 课程信息的修改，例如，当需要变更课程名称时，需要该功能。
 - ⑬ 课程信息的删除，例如，当学校删除一门旧课时，需要该功能。
 - ⑭ 课程信息的打印，例如，当学生需要某个课程详细情况时，需要该功能。
 - ⑮ 查询教师和学生名单，例如，当管理员需要根据课程获知该课程的教师和学生的详细信息时，需要该功能。
 - ⑯ 设定必修课，系统根据学生的专业和年级等信息设定其必修课。
 - ⑰ 选择选修课，学生根据自己的爱好和发展等信息选择其选修课。
 - ⑱ 选择教师，学生根据自己的喜好选择必修课和选修课的授课教师。

学生选课管理信息系统的功能结构，如图 1-2 所示。面向过程程序设计方法强调描述一个系统时以该系统包含的基本功能为基础，一个过程对应一个功能。

面向过程程序设计方法有着明显的不足：这种方法以过程或功能作为系统构建的基本单元，而忽视了过程或功能涉及的数据。面向过程程序设计方法没有将功能和数据作为整体考虑，造成了功能和数据的人为割裂。此外，我们在研究现实世界的问题时，经常以实体或物体作为基本单元，这与计算机世界将功能作为基本单元是不一致的，造成了计算机世界与现实世界映射粒度的不一致。

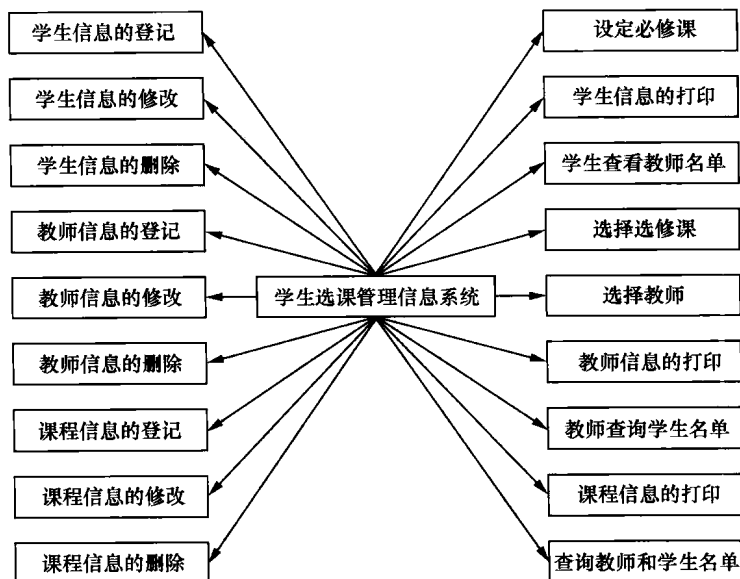


图 1-2 基于面向过程程序设计方法的学生选课管理信息系统的功能结构

一种最为自然的解决方式是，在概念世界观察或设计现实世界问题时，也以实体或物体作为基本单元。这样，现实世界和计算机世界的基本单元就一一对应了，这就是面向对象程序设计方法的初衷。

1.3 面向对象程序设计方法

对象指现实世界的实体或物体。因此，面向对象程序设计方法可以说是面向实体或物体的程序设计方法。其基本观点是：以对象或类作为系统构建的基本单元；系统开发的主要任务是使用计算机语言描述系统的对象以及对象与对象之间的关系；强调封装、继承和多态等概念。

基于面向对象程序设计方法研究学生选课管理信息系统时，功能不再是系统的基本单元，取而代之的是对象。基于面向对象程序设计方法的学生选课管理信息系统的结构，如图 1-3 所示。系统构建的基本单元是管理员、学生、教师和课程 4 个基本对象。开发学生选课管理信息系统的工作就是维护管理员、学生、教师和课程 4 个对象以及这些对象之间的关系。

与面向过程程序设计方法相比，对象是比功能具有更大粒度的单元，每个功能都应该属于一个对象。但是，对象不仅仅是功能的简单组合，它还包括功能涉及的数据。带有数据信息的基于面向对象程序设计方法的学生选课管理信息系统的结构，如图 1-4 所示。因此，完整的管理员对象应具有的功能为学生信息的登记、学生信息的修改、学生信息的删除、教师信息的登记、教师信息的修改、教师信息的删除、课程信息的登记、课程信息的修改和课程信息的删除，并由数据信息如姓名、工号和年龄组成；完整的学生对象应具有的功能为设定必修课、学生信息的打印、学生查看教师名单、选择选修课和选择教师，并数据信息如姓名、学号、年龄、专

业、学院、选修课列表和必修课列表组成；完整的教师对象应具有的功能为教师信息的打印和教师查询学生名单，并由数据信息如姓名、工号、年龄、学院和开设课程列表组成；完整的课程对象应具有的功能为课程信息的打印、查询教师和学生名单，并由数据信息如课程名、课程号和授课教师列表组成。

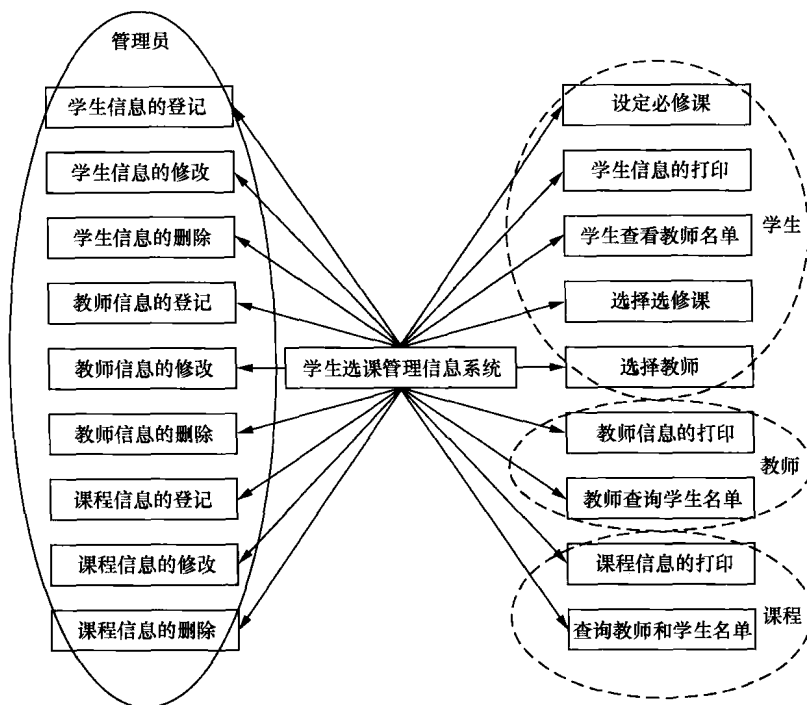


图 1-3 基于面向对象程序设计方法的学生选课管理信息系统的结构（无数据信息）

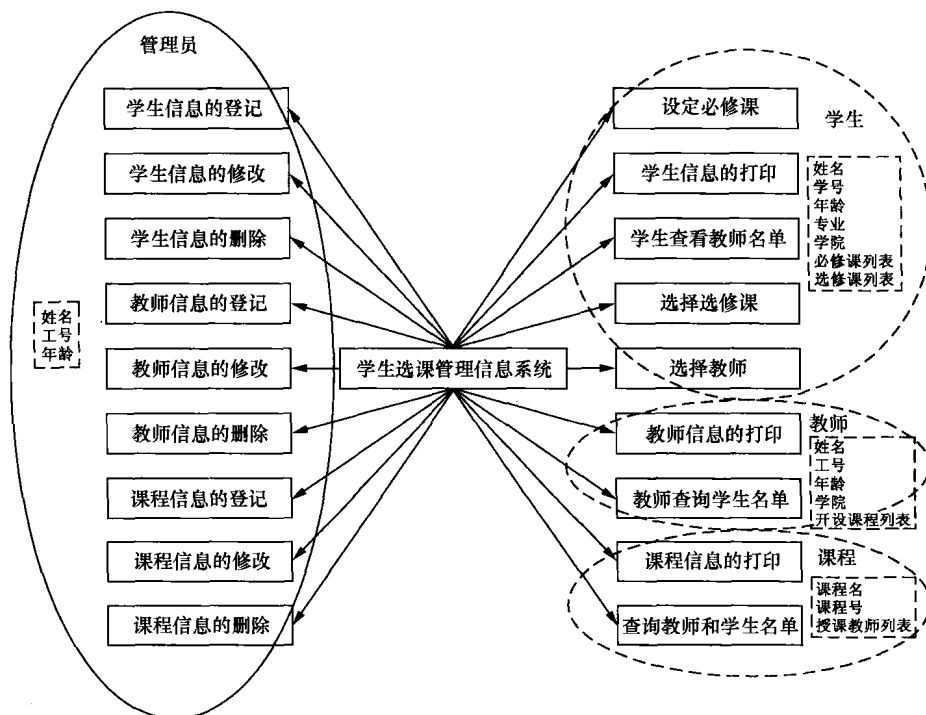


图 1-4 基于面向对象程序设计方法的学生选课管理信息系统的结构（有数据信息）

实际上,人类在认识现实世界的过程中普遍运用了以下 3 个构造法则,这与面向对象程序设计方法的思想不谋而合。

法则 1: 区分对象及其属性。例如,区分一棵树和树的大小或空间位置关系。

法则 2: 区分整体对象及其组成部分。例如,区分一棵树和树枝。

法则 3: 形成并区分不同对象的类。例如,形成所有树的类和所有石头的类,并区分它们。

法则 1 给出了对象的构成;法则 2 给出了对象与对象之间的整体—部分关系;法则 3 给出了对象与对象之间的一般—特殊关系。法则 2 和法则 3 给出了对象与对象之间的两种常见关系,我们可以将现实世界中除了这两种关系之外的所有关系统称为关联关系。而这些关系都是静态层次上描述的关系,故称为静态关系。实际上,现实世界总是不断发展运动着的,因此对象之间就不可避免地发生着各种交互关系,这种关系称为动态关系。对象之间的关系如图 1-5 所示。

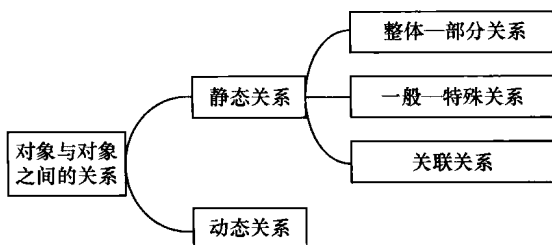


图 1-5 对象与对象之间的关系

基于面向对象程序设计方法来开发一个系统就是在计算机上实现对现实世界对象（现实世界中一般称为实体）以及对象之间关系（包括静态关系和动态关系）的描述。C++语言能够支持这一功能的实现,所以称 C++语言是面向对象的编程语言。具体来说,C++语言能够实现的功能如下。

① 支持对象数据部分的描述,见本书“第 2 章 数据类型与表达式”的数据类型部分、“第 4 章 数组”、“第 6 章 指针和引用”以及“第 7 章 自定义数据类型”。其中,第 2 章的数据类型部分介绍了简单的数据类型;第 4 章、第 6 章和第 7 章介绍了复杂的数据类型。

② 支持对象功能部分的描述,见本书第 2 章的表达式部分、“第 3 章 C++控制语句”、“第 5 章 函数及预处理”以及“第 11 章 运算符重载”。其中,第 2 章的表达式部分介绍了基本运算的实现;第 3 章介绍了如何组织基本运算实现一个较为复杂的功能;第 5 章介绍了对象的功能如何通过函数的方式实现封装;第 11 章介绍了不同操作数类型的相同操作的实现,例如,整数和实数都有加法运算。

③ 支持对象完整的描述,见本书“第 8 章 类和对象”。第 8 章涉及对象之间整体—部分关系的描述,即组合关系的描述。

④ 支持对象之间一般—特殊关系的描述,见本书“第 9 章 继承与派生”和“第 10 章 多态与虚函数”,其中第 9 章介绍了一般—特殊关系的代码实现;第 10 章介绍了由于这种一般—特殊关系的引入而特别需要解决的问题,即多态的实现。

1.4 C++语言的发展

C++语言是从 C 语言发展而来的,学习 C++语言有必要了解 C 语言的发展历程。1970 年,

AT&T 的 Bell 实验室的 D.Ritchie 和 K.Thompson 共同发明了 C 语言。研制 C 语言的初衷是使用它来编写 UNIX 操作系统。实践证明：C 语言是一种高效而灵活，且容易移植的优秀计算机编程语言。它一经推出便获得了大多数程序开发人员的喜爱，迅速成为计算机产业界的首选编程语言。直到现在，很多系统软件，如操作系统、通信程序和编译系统等都是使用 C 语言开发的。

C 语言是一种面向过程的编程语言，它的优点如下。

- ① 语法简洁紧凑。
- ② 具有丰富的运算符，可以进行算术运算、逻辑运算、条件运算、位运算和逗号运算等。
- ③ 具有丰富的数据类型，可以定义整数数据类型、浮点数据类型、字符数据类型、指针类型和数组类型等。
- ④ 能直接访问内存和进行位运算，这对于很多底层程序的开发是十分必要的。
- ⑤ 程序执行效率高，这主要是与其他编程语言（如 BASIC 语言、Pascal 语言）相比较而言的。
- ⑥ 可移植性好，这主要得益于 C 语言的标准化工作。

但是，随着软件规模的不断扩大，使用 C 语言编写程序遇到了前所未有的挑战。C 语言是面向过程、结构化和模块化的语言。当开发较小规模的程序时，程序员使用 C 语言还比较顺利。但是，当问题变得复杂，其规模日益扩大的时候，这种只支持面向过程程序设计方法的语言就逐渐显示出了其不足。究其原因，主要是由 C 语言的以下缺点造成的：类型检查机制弱；几乎不支持代码重用；程序的复杂性难以控制。

为了解决 C 语言的局限性，更是为了适应当今大规模程序开发的复杂性要求，同时也为了保持和发展 C 语言的很多优点，1980 年，贝尔实验室的 Bjarne Stroustrup 开始对 C 语言进行改进和扩充，将“类”的概念引入到 C 语言，形成了最初的“带类的 C”。1983 年，正式命名为 C++。1985 年，Bjarne Stroustrup 出版了《The C++ Programming Language》一书，这是最早介绍 C++ 语言的经典著作。

1998 年，ISO/ANSI C++ 标准正式发布。标准化给人们带来了很多好处。首先，C++ 标准使得语言的设计者（如 Bjarne Stroustrup）、程序编写人员和系统用户三者有了一致的语言来进行交互；其次，C++ 标准是人们共同遵循的关于 C++ 语言的准则，程序编写人员在各种论坛可以非常畅通地进行交流，开发的程序更为有效、通用；最后，C++ 标准大大提升了其开发的软件系统的可移植性。当然，C++ 标准更进一步推进了 C++ 语言的应用范围。

C++ 语言全面支持并兼容 C 语言，保持了 C 语言简洁、高效等很多优点，而且比 C 语言更安全（C++ 语言引入了强类型检查机制），更为重要的是全面支持面向对象的程序设计方法。C 和 C++ 语言的关系可以用公式“C++=C+面向对象”来进行简单概括。C++ 语言对 C 语言的增强表现在以下两个方面：

- ① 增加了面向对象的机制。
- ② 在原来面向过程的基础上，对 C 语言的功能做了很多扩充，如强类型检查机制和引用数据类型等。

最后，需要强调的是，不论 C 语言还是 C++ 语言，它们只是一种编写程序的工具，而程序设计方法是开发程序的过程中所采用的方法论。C 语言只支持面向过程程序设计方法，而 C++ 语言不仅支持面向过程程序设计方法，而且支持面向对象程序设计方法。本书强调的是，C++ 语言支持面向对象的程序设计方法。

1.5 C++语言程序的开发过程

学习 C++ 语言,首先要搞清楚使用 C++ 语言进行程序开发的基本过程。

一般来说,使用编程语言编写完程序之后,接着就是翻译为机器代码,以便让计算机运行获得结果。翻译的方式分为以下两种:

① 解释型。边读程序边翻译,翻译成机器代码后立刻执行。

② 编译型。先翻译整个程序为机器代码,并保存到可执行的程序文件中,然后运行获得结果。

C++ 语言是一种编译型编程语言。但是,程序设计语言发展到现在,无论是编译型还是解释型,一般都必须有集成开发环境 (Integrated Development Environment, IDE) 的支持软件。程序员在该软件中,编辑程序代码,编译源文件,链接相关资源文件,直到调试运行,其完整过程如图 1-6 所示。这个过程涉及以下几个概念:

① 源程序。使用某种编程语言 (如 C++ 语言) 编写的程序称为源程序。C++ 语言的源程序文件以扩展名 .cpp 作为标识。这里, cpp 是 c plus plus 的缩写,意思为 C++。

② 目标程序。源程序经过翻译加工后生成的程序称为目标程序。C++ 语言的目标程序文件以扩展名 .obj 作为标识。这里, obj 是 object 的缩写,意思为目标。

③ 可执行程序。将目标程序和其相关的资源文件进行链接,生成的程序称为可执行程序。C++ 语言的可执行程序文件以扩展名 .exe 作为标识。这里, exe 是 execution 的缩写,意思为可执行。

此外, C++ 语言的开发过程中还需要编辑程序、编译程序和链接程序 3 种工具的支持。编辑程序是撰写 C++ 程序代码的工具;编译程序将 C++ 源程序翻译为目标程序;而链接程序将 C++ 目标程序和相关资源文件组装成可执行程序。

对于 C++ 语言的集成开发环境 Microsoft Visual C++ 6.0,其安装目录下的文件 cl.exe 即为编译程序,而安装目录下的文件 link.exe 即为链接程序。

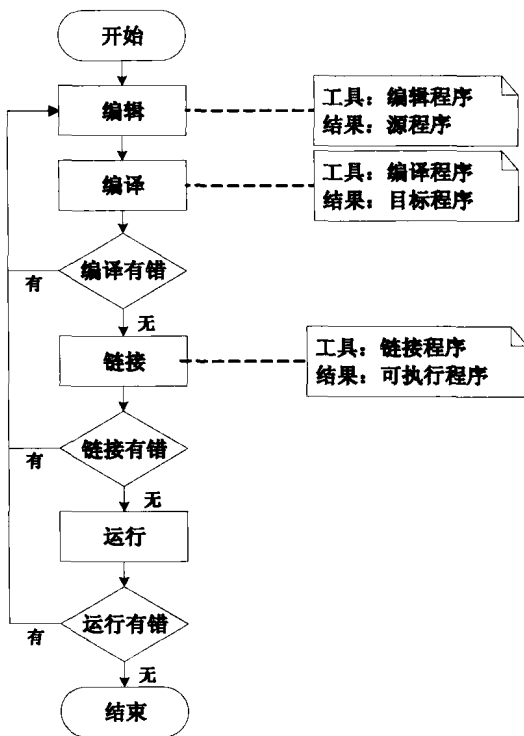


图 1-6 C++语言程序的开发过程

1.6 一个简单的程序

下面从一个简单的 Hello World 程序开始,了解 C++ 程序的基本构成。

【例 1-1】输出一行字符串: "Hello World."。

程序如下。