



规划教材

普通高等学校应用型人才
培养系列规划教材
丛书主编 陈明

C++ 程序设计

任化敏 主编

王 丽 刘仁权 副主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

普通高等学校应用型人才培养系列规划教材
丛书主编 陈 明

C++程序设计

任化敏 主 编
王 丽 刘仁权 副主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

C++语言是目前最为流行的程序设计语言之一,本书共分三个部分,第一部分从C++的基础语法展开讲解,第二部分介绍面向对象的编程方法,第三部分以实际案例的开发过程为例,介绍应用程序的原型开发及优化过程。书中每个知识点的讲解都是通过实例进行的,在多章的最后一节安排了应用实例,以培养读者对本章中关键内容的总体把握和理解以及利用C++解决实际问题的能力。

本书的作者都是长期从事高校计算机专业教学与科研的一线教师,有丰富的教学与编程经验,在书中用大量的实例进行讲解,并对学习中容易出错的知识点进行了详尽的整理和细心的提示。

本书的知识点涵盖全面,代码丰富,示例详尽,适合高等院校信息类专业的学生使用,也可以作为理工科学生的公共课教材、C++培训教材,与本书配套的《C++程序设计实践教程》(中国铁道出版社出版)提供了Visual Studio 2005环境下C++的上机实验指导。

本书配有的电子教案和本书中所有使用的代码均可以通过登录中国铁道出版社(网址:<http://edu.tqbooks.net>)网站免费下载。

图书在版编目(CIP)数据

C++程序设计/任化敏主编. -- 北京:中国铁道出版社, 2010

(普通高等学校应用型人才培养系列规划教材)

ISBN 978-7-113-11164-9

I. ①C… II. ①任… 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(2010)第040302号

书 名: C++程序设计

作 者: 任化敏 主编

策划编辑: 秦绪好 周海燕

责任编辑: 周海燕

编辑助理: 巨 凤

版式设计: 于 洋

责任印制: 李 佳

编辑部电话: (010) 63560056

封面制作: 白 雪

出版发行: 中国铁道出版社(北京市宣武区右安门西街8号 邮政编码: 100054)

印 刷: 北京鑫正大印刷有限公司

版 次: 2010年6月第1版 2010年6月第1次印刷

开 本: 787mm×960mm 1/16 印张: 21.25 字数: 488千

印 数: 3 000册

书 号: ISBN 978-7-113-11164-9

定 价: 33.00元

版权所有 侵权必究

本书封面贴有中国铁道出版社激光防伪标签,无标签者不得销售

凡购买铁道版图书,如有印制质量问题,请与本社计算机图书批销部联系调换

普通高等学校应用型人才培养系列规划教材

主 任：陈 明

副主任：蒋宗礼 严晓舟 王锁柱

委 员：（按姓氏笔画排序）

王全民 刘宏志 刘贵龙 刘振华

李也白 张 钢 张晓明 陈志泊

郝 莹 秦绪好 袁 薇 解 凯

管建和 廖湖声

总策划：焦金生

编 辑：杨 勇 周海燕

经过几十年的研究与探索,现代计算机系统功能越来越强大、应用越来越广泛。计算机的广泛应用一方面对人类社会的发展做出了卓越的贡献,另一方面也在推动计算机学科的高速发展,因而一直受到社会的高度关注。

由于计算机学科呈现出的学科内涵宽泛化、分支相对独立化、社会需求多样化、专业规模巨大化和计算教育大众化等特点,使得计算机企业成为了朝阳企业,这些 IT 企业需要大量的具有专门计算机技能的人才,但传统的研究型计算机教育是以学术教育为基础,以培养计算机精英为目的的计算机教育,与社会和行业对计算机高等教育人才的需求产生了矛盾——大学中单一的计算机精英型教育培养的人才已不能满足实际需要,凸显职业特征的计算机应用型教育异军突起,迅速发展,备受关注。这种矛盾促使教学模式呈现了职业性,并在培养面向知识应用和全面能力方面,提出了多种职业性教学模式,如网络工程师、软件工程师、动画设计师、硬件工程师等。因此,研究和实施计算机应用型人才培养模式势在必行。

什么是计算机应用型教育?我们通过长时间的教育研究和对教学经验的总结,认为计算机应用型教育的培养目标可以用知识、能力和素质三个基本要素来描述。

知识是基础、载体和表现形式,从根本上影响着能力和素质。学习知识的目的是为了获得能力和不断地提升能力。知识可以转化为能力和素质,能力对知识具有反作用,促进知识的不断发展。

能力是核心,是应用型人才特征的突出表现。从计算机学科而言,培养的人才应具备计算思维能力、算法设计与分析能力、程序设计与实现能力、系统能力(系统的认知、设计、开发、应用能力)。而计算机应用型人才的能力有着其独特的属性,主要包括应用能力(专业能力)和通用能力。应用能力主要是指用所学知识解决专业实际问题的能力。通用能力是指跨职业能力,并不是具体的专业能力和职业技能,而是对不同职业的适应能力,也就是当职业发生变更时,这些能力依然在从业者身上起作用。计算机应用型本科教育所要培养的三种通用能力是:学习能力、工作能力和创新能力。

素质要素主要是指基本素质,即具有良好的公民道德和职业道德,具有合格的政治思想素养,遵守计算机法律和法规,具有人文、科学素养和良好的职业素质等。计算机应用型人才素质主要是指工作的基本素质,且要求在从业中必须具备责任意识,能够对自己职责范围内的工作认真负责地完成。

与此同时，我们认为要实现计算机应用型教育的培养目标关键在于课程体系的构建，即课程内容和课程性质的确定。因此，我们将计算机应用型教育课程分为通用课程、基础课程、专业核心课程、专业选修课程、应用课程、实验课程、实践课程等，并建立了相应的教育课程体系，如公共基础课程平台、专业基础课程平台、专业选修课程平台、基本素质平台等，力图通过相应课程的开展来实现培养目标。

目前，应用型人才培养的研究方兴未艾，我们也将会在较长一段时间的教学实践工作中继续探讨和总结经验。本套“普通高等学校应用型人才培养系列规划教材”系列丛书是应社会需求而组织编写的，经过系统规划与设计，定位于高等院校计算机应用型人才的培养，整套丛书从架构到具体内容都力主突出能力培养的理念。本系列教材现正在陆续出版中，希望各位老师和读者指正。

《普通高等学校应用性人才培养系列规划教材》丛书主编

陳明

目前,计算机技术的发展日新月异,多种计算机语言都被广泛应用。其中标准 C++ 语言无论在科学理论研究还是在软件开发应用中都一直处于较高的地位。其可靠性、可移植、高效率等众多特性,都使它焕发出无穷魅力,成为众多程序开发人员热衷使用的一种语言。程序设计是理工科大学生的必修课程,选择何种程序设计语言作为第一门程序设计课程的内容也是目前多所高校面临的问题之一。C++ 作为一种高级程序设计语言,既具有面向对象的主要特征,又保留了传统的结构化的 C 语言的高效特征。因此,选择 C++ 语言作为程序设计课程的学习内容无论在提高学生的能力还是满足社会实际需要方面都是一种不错的选择。

全书分为三大部分,共 13 章。其中前 8 章是第一部分,介绍 C++ 程序设计语言基础知识,其中第 1 章为 C++ 简介;第 2 章介绍数据类型和表达式;第 3 章介绍程序控制语句;第 4 章介绍标准库类型;第 5 章介绍数组、指针与结构体;第 6 章介绍函数;第 7 章介绍输入/输出流;第 8 章介绍异常处理和预编译。对于没有程序设计基础的读者可以从这部分入手,了解面向过程的结构化程序设计方法。从第 9~12 章是第二部分,介绍面向对象编程知识,其中第 9 章是对类的分析;第 10 章是类的进一步学习;第 11 章介绍类的派生与继承;第 12 章介绍多态性和虚函数。对于已经有了程序设计基础的读者可以选择从这部分进入面向对象的程序设计的學習。第 13 章作为本书的第三部分介绍了一个项目实例,希望读者在掌握了基本的内容之后能够通过实践更深刻地理解和运用 C++ 语言。

参与本书编写的作者都是长期工作在教学和科研第一线的教师,有着多年的 C++ 程序设计课程的教学经验和编程经验,编写本书时,总结并参考了在日常教学工作中的经验和体会,有针对性地对教学过程中学生常见的一些问题进行了总结和重点解释,力争使本书做到通俗易懂,贴近教学实际。同时,更强调以培养具有实际编程能力的程序员为目标。

全书概念清晰、结构合理、内容安排由浅入深,在编写中既考虑到读者对于知识点的理解又兼顾了读者对于知识点的运用能力。本书通过大量的实例来讲解基础概念,通过应用实例培养读者解决实际问题的能力。既考虑到初学者的特点,同时又兼顾软件设计人员的要求;既可以作为教师讲授 C++ 语言的教材,又可以作为大专院校学生以及计算机培训班学员学习 C++ 语言的有力工具,同样也适用于广大软件开发人员、自学人员和等级考试人员的参考用书。

本书在每章中都配备了实例讲解和精选习题,旨在帮助读者更好地掌握本章的内容。并且所有示例均在 Microsoft Visual C++ 2005 环境下调试通过。本书配有的电子教案和本书中所有使用的代

码均可以通过登录中国铁道出版社网站（<http://edu.tqbooks.net>）免费下载。

本书第一部分的第1~4章、6~8章由王丽编写，第一部分的第5章由刘仁权编写，第二部分和第三部分由任化敏编写，全书由任化敏统稿。

特别感谢陈明教授对于本书在编写内容和风格方面给予的无私帮助和指导；中国铁道出版社的编辑对全书进行了认真仔细的审核与校正，在此一并表示感谢。同时，作者在编写本书的过程中，翻阅并参考了大量的文献资料，在此谨向这些资料的作者表示衷心的感谢。

由于时间仓促，作者水平有限，书中难免有遗漏和不足之处，敬请读者批评指正，激励我们为大家编写出更加优秀的教材。如您有任何宝贵的意见和建议，请通过 E-mail: renhuamin_bucm@sina.com 与我们联系。

编 者

2010年4月

第一部分 C++程序设计语言基础

第 1 章 C++简介	1
1.1 C++语言概述	1
1.1.1 计算机语言概述	1
1.1.2 C++语言的发展	2
1.1.3 C++的特点	2
1.2 简单的 C++程序体验	3
1.2.1 C++程序的组成	3
1.2.2 C++程序的书写格式	5
1.3 C++程序的编辑、编译、连接和运行	6
1.4 练习	10
第 2 章 数据类型和表达式	11
2.1 数据类型概述	11
2.2 C++基本数据类型	12
2.3 常量和变量	13
2.3.1 常量	13
2.3.2 变量	16
2.4 运算符	18
2.4.1 算术运算符	18
2.4.2 赋值运算符	19
2.4.3 关系运算符	20
2.4.4 逻辑运算符	21
2.4.5 位运算符	22
2.4.6 自增和自减运算符	24
2.4.7 条件运算符	25
2.4.8 逗号运算符	25
2.5 表达式中运算符的优先级	26

2.6	表达式中数据类型的转换.....	27
2.6.1	何时发生数据类型转换.....	27
2.6.2	隐式转换.....	27
2.6.3	显式转换.....	28
2.7	练习.....	29
第3章	程序控制语句.....	30
3.1	条件语句.....	30
3.1.1	条件语句的标准形式.....	30
3.1.2	嵌套条件语句及其优化.....	33
3.1.3	switch 语句.....	36
3.2	循环语句.....	41
3.2.1	while 循环.....	42
3.2.2	do-while 循环.....	42
3.2.3	for 循环.....	43
3.2.4	嵌套循环.....	46
3.3	转移语句.....	46
3.3.1	break 语句.....	47
3.3.2	continue 语句.....	48
3.3.3	goto 语句.....	49
3.4	应用实例.....	50
3.5	练习.....	52
第4章	标准库类型.....	55
4.1	标准库的组织结构.....	55
4.2	命名空间.....	56
4.3	标准库的 string 类型.....	59
4.3.1	string 对象的定义和初始化.....	59
4.3.2	string 对象的操作.....	60
4.4	标准库的 vector 类型.....	62
4.4.1	vector 对象的定义和初始化.....	63
4.4.2	vector 的操作.....	63
4.5	应用实例.....	64
4.6	练习.....	69
第5章	数组、指针与结构体.....	70
5.1	数组.....	70

5.1.1	数组的声明和初始化.....	71
5.1.2	引用数组元素	74
5.1.3	字符数组	77
5.2	结构体	78
5.2.1	结构体类型的定义.....	78
5.2.2	结构体变量的声明和引用	79
5.2.3	结构体数组	81
5.3	指针	84
5.3.1	什么是指针	84
5.3.2	指针变量的声明和初始化	85
5.3.3	指针操作	87
5.3.4	指针和数组	88
5.3.5	sizeof 运算符	90
5.3.6	结构体指针变量.....	90
5.3.7	动态分配/释放内存.....	91
5.3.8	const 指针	93
5.4	引用	94
5.5	应用实例	94
5.6	练习	96
第 6 章	函数	98
6.1	函数概述	98
6.1.1	函数定义	99
6.1.2	函数的声明	100
6.1.3	函数的调用	101
6.1.4	函数的默认参数值.....	103
6.2	参数传递和返回值	104
6.2.1	指针作函数的参数和返回值	104
6.2.2	引用作函数参数和返回值	107
6.2.3	数组形参	108
6.2.4	vector 形参.....	111
6.3	变量的作用域和存储类型.....	112
6.3.1	变量的作用域	112
6.3.2	变量的存储类型.....	115
6.4	内联函数	118

6.5	重载函数	119
6.6	应用实例	121
6.7	练习	127
第7章	输入/输出流	129
7.1	IO类型的标准库	129
7.2	标准输入/输出	130
7.2.1	标准输出	130
7.2.2	标准输入	131
7.3	文件的输入与输出	133
7.3.1	文件的打开与关闭	133
7.3.2	文件的读/写	135
7.3.3	二进制文件的读/写	136
7.4	字符串流	137
7.5	应用实例	139
7.6	练习	139
第8章	异常处理和预编译	140
8.1	异常处理	140
8.1.1	异常概述	140
8.1.2	异常处理机制	141
8.2	编译预处理	143
8.2.1	宏定义	143
8.2.2	文件包含	144
8.2.3	条件编译	145
8.3	应用实例	147
8.4	练习	149

第二部分 面向对象编程

第9章	类	150
9.1	类的概述	150
9.1.1	类定义	151
9.1.2	数据抽象和封装	154
9.1.3	访问标号	155
9.1.4	类对象	158

9.1.5	this 指针	159
9.2	构造函数和析构函数	160
9.2.1	构造函数	160
9.2.2	带有成员初始化表的构造函数	161
9.2.3	析构函数	163
9.3	友元	165
9.3.1	为什么要引入友元	165
9.3.2	友元函数	166
9.3.3	友元类	167
9.4	静态成员	168
9.4.1	静态数据成员	169
9.4.2	静态成员函数	170
9.5	类作用域	172
9.6	应用实例	172
9.7	练习	177
第 10 章	类的进一步学习	179
10.1	inline 成员函数	179
10.2	类对象数组	180
10.3	const 型数据	182
10.3.1	const 对象 (常对象)	182
10.3.2	const 成员函数 (常成员函数)	182
10.3.3	指向对象的常指针	184
10.4	复制构造函数	184
10.4.1	定义形式	185
10.4.2	合成的复制构造函数	185
10.4.3	为什么需要复制构造函数	186
10.4.4	自定义复制构造函数	187
10.5	运算符重载	188
10.5.1	重载运算符的定义	188
10.5.2	重载运算符的注意事项	190
10.6	异常类	193
10.6.1	标准库异常类	193
10.6.2	用户自定义异常类	193

10.7	应用实例	195
10.8	练习	200
第 11 章	类的派生与继承	202
11.1	概述	202
11.2	类的继承与派生	203
11.2.1	定义基类	204
11.2.2	定义派生类	204
11.3	继承方式与访问控制	208
11.3.1	公有派生	208
11.3.2	私有派生	211
11.3.3	保护派生	212
11.4	基类与派生类的构造函数和析构函数	215
11.4.1	派生类的构造函数	215
11.4.2	派生类的析构函数	219
11.5	基类与派生类的关系	219
11.5.1	基类与派生类定义的成员的优先关系	219
11.5.2	类型兼容	220
11.6	多继承	223
11.6.1	多继承的定义	223
11.6.2	多继承的构造与析构函数	226
11.6.3	多继承的歧义性与解决方法	227
11.7	虚基类	230
11.8	应用实例	235
11.9	练习	238
第 12 章	多态性和虚函数	240
12.1	多态	240
12.1.1	静态多态性	241
12.2.2	动态多态性	243
12.2	虚函数	243
12.2.1	虚函数的定义和使用规则	243
12.2.2	虚函数的特征	245
12.2.3	虚析构函数	247
12.3	纯虚函数和抽象类	248
12.3.1	纯虚函数	248

12.3.2 抽象类	249
12.4 应用实例	250
12.5 练习	257

第三部分 案例开发

第 13 章 项目实例	259
13.1 Visual Studio 开发环境简介	259
13.2 核心功能与原型设计	260
13.2.1 核心功能分析	261
13.2.2 原型设计	261
13.2.3 原型分析	272
13.3 MFC	272
13.3.1 MFC 简介	272
13.3.2 文件结构	274
13.4 建立数据库连接	276
13.4.1 ODBC 简介	277
13.4.2 建立数据库	278
13.4.3 MFC ODBC 数据库类	279
13.4.4 优化程序——建立数据库连接	283
13.5 设计友好的用户界面	289
13.5.1 界面	290
13.5.2 优化程序——设计界面	291
13.6 程序功能分析	295
13.7 小结	295
附录 参考答案	296
参考文献	323

第1章

C++简介

教学目标:

- 了解计算机程序设计语言发展的过程
- 了解 C++语言发展的过程
- 掌握 C++语言的特点
- 掌握 C++程序的构成以及实现过程

1.1 C++语言概述

1.1.1 计算机语言概述

随着科技的发展和进步,计算机已成为人们处理信息的重要工具。为了使计算机能够遵照人的意志并在人的控制下工作,就需要建立一种人和计算机之间的通信机制,这种机制是通过计算机语言来实现的。计算机语言是用于实现人与计算机之间交流的语言,它将人类用自然语言描述的问题以及解决问题的方法用计算机能够识别的语言描述出来。自从计算机出现至今,伴随着计算机硬件和程序设计方法的发展,计算机语言也出现了一个由低级到高级发展的过程,大致分为机器语言、汇编语言和高级语言三个阶段。

机器语言是唯一能被计算机直接识别和执行的语言,这是因为从本质上讲计算机只能识别由二进制数“0”和“1”组成的二进制信息,而机器语言就是使用“0”和“1”这样的代码组成指令的集合。机器语言与计算机硬件紧密结合,每个指令集合只能被特定的计算机类型所使用。所以,使用这样的机器语言编写的程序非常复杂,人们阅读程序和调试程序都很困难。

汇编语言是使用一些简单的英文字母和符号表示指令的计算机语言;并且能够让使用这些语言编写的指令与机器语言的指令一一对应。例如在汇编语言中,用 ADD 表示加法,用 SUB 表示减法

等，这样大大改善了程序的可读性。但是，由于计算机不能直接识别汇编程序，这就需要在运行程序时利用某个翻译程序将汇编语言源程序翻译成机器语言程序后再运行。

与机器语言一样，汇编语言也是面向计算机硬件的语言，二者统称为低级语言。它们与计算机的 CPU 紧密相关。不同的 CPU 的指令集各不相同，所对应的低级语言也不相同。但是同机器语言相比较，汇编语言的可读性和易用性都有了很大的提高。由于其具有占有存储空间小、执行效率较高等特点，目前在计算机底层硬件开发方面，汇编语言仍然被广泛使用。

由于低级语言都是面向计算机硬件的，具有不便于移植的缺点，并且它们与人类的思维和表达方式都有很大的差距，为此人们又相继开发了一系列的高级语言。高级语言更接近于人类的自然语言，它不依赖具体的计算机硬件，更容易被人们理解和接受。例如，BASIC、C、C++、C#、Java 等都属于高级语言。无论用哪一种高级语言编写的程序都需要经过编译系统翻译才能被计算机理解和执行。

1.1.2 C++语言的发展

C++语言由贝尔实验室的 Bjarne Stroustrup 设计，是目前广泛应用的一种面向对象程序设计语言。它的开发过程是被人们所津津乐道的。1979 年，Bjarne Stroustrup 在研究 UNIX 系统时完成了一个可以运行的预处理程序，它为 C 加上了类似 Simula 的类机制，当时人们把这个预处理器所接受的语言称为“带类的 C”。1980 年，“带类的 C”第一次投入使用，当时它只支持系统程序设计和数据抽象技术。1983 年，C++ 的名字正式诞生，并且将支持面向对象程序设计的语言加入其中。伴随着 C++ 产品的出现和广泛应用，1990 年 ANSI (American National Standard Institute) C++ 标准委员会正式启动了 C++ 标准化工程，后来 ISO (International Standards Organization) 也加入其中，并于 1998 年正式发布了 C++ 语言的国际标准。2003 年，这个标准又被重新修订，成为目前的 C++，并且一直发展至今。

1.1.3 C++的特点

C++语言是在 C 语言的基础上发展而来的一种面向对象程序设计语言。它既继承了 C 语言的诸多特性，又对 C 语言进行了扩充和改进。下面分别介绍 C++ 的这几方面的特性。

(1) C++语言是面向对象的程序设计语言

之所以说 C++ 是一种面向对象程序设计语言，是因为 C++ 语言支持面向对象方法中的封装、继承和多态三种主要特性。

- 封装是指将一个数据和与这个数据有关的操作集合放在一起形成的对象。在 C++ 语言中允许使用类和对象，类是支持数据封装的工具，而对象可以将数据封装实现。
- 继承是指一个对象从另一个对象中获得属性的过程，体现特殊类与一般类之间的关系。在 C++ 语言中不仅支持单重继承，而且支持多重继承。继承性增强了程序的扩展性和可重用性，