

国家精品课程配套教材
高等教育国家级教学成果二等奖

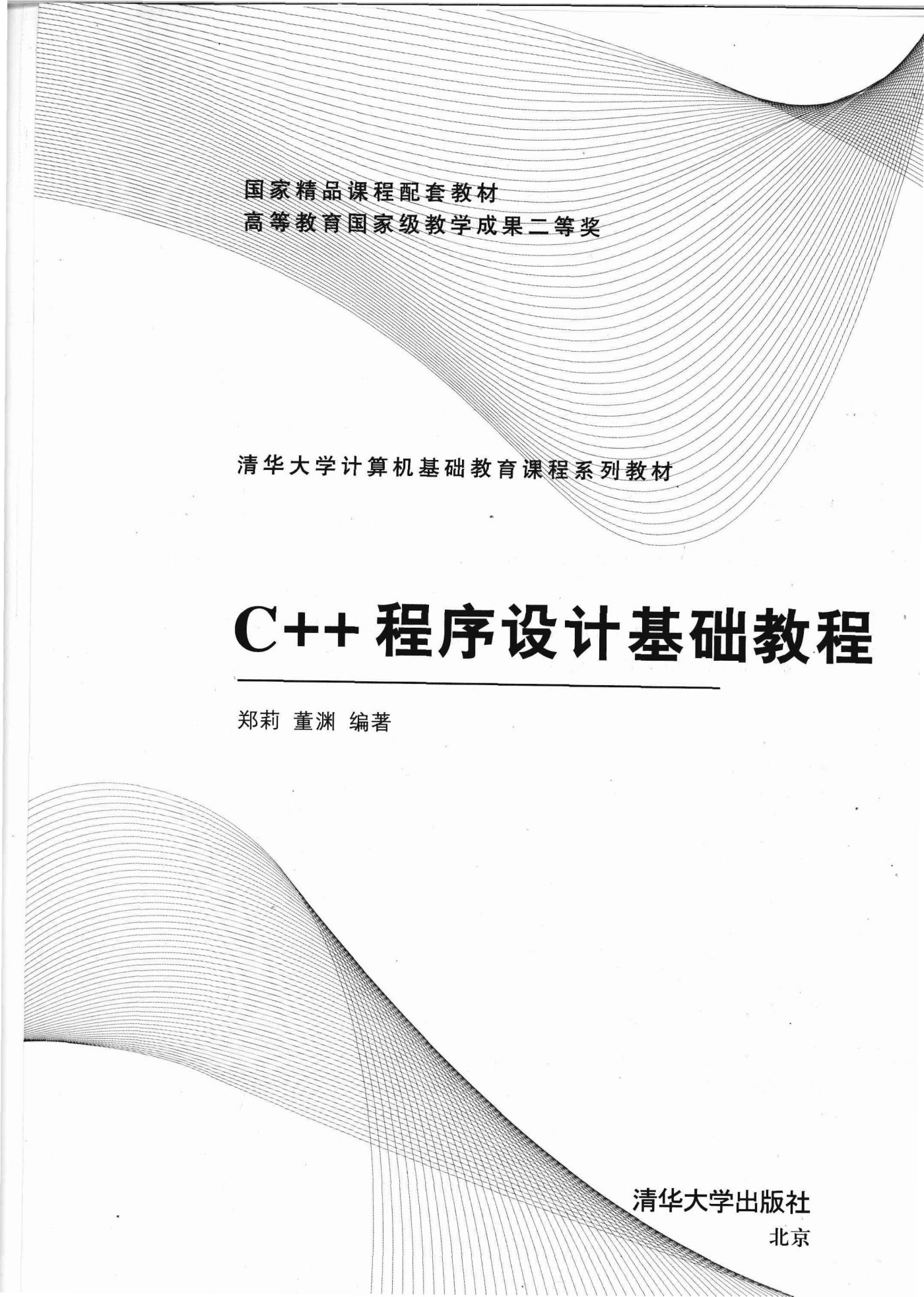
清华大学计算机基础教育课程系列教材

C++ 程序设计基础教程

郑莉 董渊 编著



清华大学出版社



国家精品课程配套教材
高等教育国家级教学成果二等奖

清华大学计算机基础教育课程系列教材

C++ 程序设计基础教程

郑莉 董渊 编著

清华大学出版社
北京

内 容 简 介

本书针对程序设计的初学者,以面向对象的程序设计思想为主线,以通俗易懂的方法介绍 C++ 语言,引导读者以最自然的方式,将人类习惯的面向对象的思维方法运用到程序设计中。主要内容包括程序设计基础知识、类与对象的基本概念、继承与多态以及输入输出流。整套教材语言生动、流畅,深入浅出。适用于各类学校的 C++ 语言程序设计课程。

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

C++ 程序设计基础教程 / 郑莉,董渊编著. —北京:清华大学出版社,2010.8

(清华大学计算机基础教育课程系列教材)

ISBN 978-7-302-23361-9

I. ①C… II. ①郑… ②董… 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 145980 号

责任编辑:谢琛 薛阳

责任校对:白蕾

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:北京密云胶印厂

装 订 者:三河市金元印装有限公司

经 销:全国新华书店

开 本:185×260 印 张:22

字 数:515 千字

版 次:2010 年 8 月第 1 版

印 次:2010 年 8 月第 1 次印刷

印 数:1~5000

定 价:33.00 元

产品编号:037925-01

序

计算机科学技术的发展不仅极大地促进了整个科学技术的发展,而且明显地加快了经济信息化和社会信息化的进程。因此,计算机教育在各国备受重视,计算机知识与能力已成为 21 世纪人才素质的基本要素之一。

清华大学自 1990 年开始将计算机教学纳入基础课的范畴,作为校重点课程进行建设和管理,并按照“计算机文化基础”、“计算机技术基础”和“计算机应用基础”三个层次的课程体系组织教学:

第一层次“计算机文化基础”的教学目的是培养学生掌握在未来信息化社会里更好地学习、工作和生活所必须具备的计算机基础知识和基本操作技能,并进行计算机文化道德规范教育。

第二层次“计算机技术基础”是讲授计算机软硬件的基础知识、基本技术与方法,从而为学生进一步学习计算机的后续课程,并利用计算机解决本专业及相关领域中的问题打下必要的基础。

第三层次“计算机应用基础”则是讲解计算机应用中带有基础性、普遍性的知识,讲解计算机应用与开发中的基本技术、工具与环境。

以上述课程体系为依据,设计了计算机基础教育系列课程。随着计算机技术的飞速发展,计算机教学的内容与方法也在不断更新。近几年来,清华大学不断丰富和完善教学内容,在有关课程中先后引入了面向对象技术、多媒体技术、Internet 与互联网技术等。与此同时,在教材与 CAI 课件建设、网络化的教学环境建设等方面也正在大力开展工作,并积极探索适应 21 世纪人才培养的教学模式。

为进一步加强计算机基础教学工作,适应高校正在开展的课程体系与教学内容的改革,及时反映清华大学计算机基础教学的成果,加强与兄弟院校的交流,清华大学在原有工作的基础上,重新规划了“清华大学计算机基础教育课程系列教材”。

该系列教材有如下几个特色:

1. 自成体系:该系列教材覆盖了计算机基础教学三个层次的教学内容。其中既包括所有大学生都必须掌握的计算机文化基础,又包括适用于各专业的软、硬件基础知识;既包括基本概念、方法与规范,又包括计算机应用开发的工具与环境。

2. 内容先进:该系列教材注重将计算机技术的最新发展适当地引入教学中来,保持了教学内容的先进性。例如,系列教材中包括了面向对象与可视化编程、多媒体技术与应用、Internet 与互联网技术、大型数据库技术等。

3. 适应面广：该系列教材照顾了理、工、文等各种类型专业的教学要求。

4. 立体配套：为适应教学模式、教学方法和手段的改革，该系列教材中多数都配有习题集和实验指导、多媒体电子教案，有的还配有 CAI 课件以及相应的网络教学资源。

本系列教材源于清华大学计算机基础教育的教学实践，凝聚了工作在第一线的任课教师的教学经验与科研成果。我希望本系列教材不断完善，不断更新，为我国高校计算机基础教育做出新的贡献。



注：周远清，曾任教育部副部长，原清华大学副校长、计算机专业教授。

前言

一、版本说明

本书是《C++ 语言程序设计(第 4 版)》的姊妹篇。《C++ 语言程序设计》第 1 版于 1999 年出版,第 2 版于 2001 年出版,第 3 版于 2003 年出版。在前三版基础上,广泛听取了读者和同行的建议,参考了最新的资料,并根据作者本人在授课过程中的经验,针对不同的教学要求,将这本书分成了两个不同的版本:《C++ 程序设计基础教程》和《C++ 语言程序设计(第四版)》。本书适合用于程序设计课程学时较少的学校和专业。

与《C++ 语言程序设计(第 3 版)》相比,本书主要在以下方面作了修订:

1. 去掉了原来的第 9、10、12、13 章。
2. 从第 4 章开始,每章编写了新的综合实例“个人银行账户管理程序”。
3. 部分例题有更新。

与《C++ 语言程序设计(第 4 版)》相比,去年了第 9、10、12 章,去掉了每章的“深度探索”一节。

二、本书的编写背景

C++ 是从 C 语言发展演变而来的一种面向对象的程序设计语言。C++ 语言的主要特点表现在两个方面:一是兼容 C,二是支持面向对象的方法。

面向对象的程序设计(OOP)方法将数据及对数据的操作方法封装在一起,作为一个相互依存、不可分离的整体——对象。对同类型对象抽象出其共性,形成类。类中的大多数数据,只能用本类的方法进行处理。类通过一个简单的外部接口,与外界发生关系,对象与对象之间通过消息进行通讯。这样,程序模块间的关系简单,程序模块的独立性、数据的安全性具有良好的保障,通过继承与多态性,使程序具有很高的可重用性,使得软件的开发和维护都更为方便。

由于面向对象方法的突出优点,目前它已经成为开发大型软件时所采用的主要方法。而 C++ 语言是应用最广泛的面向对象的程序设计语言之一。

长期以来,C++ 语言被认为是较难使用的专业开发语言,特别是对于程序设计类课程学时较少的学校和专业,很难开设 C++ 程序设计课程。

实际上面向对象方法的出现,是程序设计方法发展的一个返朴归真过程。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。面向对象方法所强调的基本原则,就是直接面对客观存在的事物来进行软件开发,将人们在日常生活中习惯的思维方式和表达方式应用在软件开发中,使软件开发从过分专业化的方法、规则和技巧中回到客观世界,回到人们通常的思维。

虽然 C++ 语言是从 C 语言发展而来的,但是学习 C++ 可以不必从 C 语言开始,C++

本身是一个完整的程序设计语言,是可以作为入门的程序设计语言来学习的。

本书就是针对初学者的入门教程,适合用于开设一门学时不多的C++课程。

三、本书的特色

本书的特色是涵盖了C++语言的主要语法、面向对象的程序设计方法,内容深入浅出、立体配套。

全书以面向对象的程序设计方法贯穿始终,每一章都是首先阐述面向对象的程序设计思想和方法,由实际问题入手,然后引出必要的语法知识,在讲解语法时着重从程序设计方法学的角度讲述其意义和用途。本书的宗旨是,不仅要使读者掌握C++语言本身,而且要能够对现实世界中较简单的问题及其解决方法用计算机语言进行描述。当然,要达到能够描述较复杂的问题域还需要学习面向对象的软件工程课等其他课程。

针对初学者和自学者的特点,本书力求做到深入浅出,将复杂的概念用简洁浅显的语言娓娓道来。

本书建议的讲课学时数为24学时。建议讲课学时数分配如下:

第1章2学时,第2章4学时,第3章2学时,第4章4学时,第5章2学时,第6章4学时,第7章2学时,第8章2学时,第9章2学时。

实验学时数的分配请参考配套的《C++程序设计基础教程学生用书》。

学生可以通过配套的《C++程序设计基础教程学生用书》,在实践中达到对内容的深入理解和熟练掌握。为了方便教师备课,本书还配有电子教案(PPT文件)。

四、内容摘要

第1章 绪论:从发展的角度概要介绍了面向对象程序设计语言的产生和特点,面向对象方法的由来和主要的基本概念,并简单介绍了什么是面向对象的软件工程。最后介绍了信息在计算机中的表示和存储以及程序的开发过程。

第2章 C++ 简单程序设计:讲述C++程序设计的基础知识。首先简要介绍C++语言的发展历史及其特点;接着学习构成C++语句的基本部分,包括字符集、关键字、标识符、操作符等;还有C++的基本数据类型和自定义数据类型以及算法的控制结构,包括顺序、选择和循环结构。“深度探索”介绍变量的实现机制和C++表达式的执行原理。

第3章 函数:讲述C++语言的函数。在面向对象的程序设计中,函数对处理过程的基本抽象单元,是对功能的抽象。同时,使用函数也为代码的重用提供了技术上的支持。我们主要从应用的角度讲述各种函数的定义和使用方法。“深度探索”介绍运行栈与函数调用的执行、函数声明与类型安全。

第4章 类与对象:首先介绍面向对象程序设计的基本思想及其主要特点,包括抽象、封装、继承和多态;接着围绕数据封装这一特点,着重讲解面向对象设计方法的核心概念一类,其中包括类的定义、实现以及如何利用类来解决具体问题;最后,简单介绍了如何用UML语言描述类的特性。“深度探索”介绍位域、用构造函数定义类型转换,以及对象作为函数参数和返回值的传递方式。

第5章 数据的共享与保护:讲述标识符的作用域和可见性及变量、对象的生存期;

使用局部变量、全局变量、类的数据成员、类的静态成员和友元来实现数据共享,共享数据的保护,以及使用多文件结构来组织和编写程序解决较为复杂的问题。“深度探索”介绍常成员函数的声明原则、代码的编译连接与执行过程。

第6章 数组、指针与字符串:讨论数组、指针与字符串。数组和指针是C++语言中最常用的复合(构造)类型数据,是数据和对象组织、表示的最主要手段,也是组织运算的有力工具。本章首先介绍数组、指针的基本概念动态存储分配以及动态数组对象。接着围绕数据和对象组织这一问题,着重讲解如何通过使用数组和指针解决数据、函数以及对象之间的联系和协调。对于字符串及其处理,本章重点介绍string类。“深度探索”介绍指针与引用的联系、指针的安全性隐患及其应对方案,以及const_cast的应用。

第7章 继承与派生:讲述类的继承特性。围绕派生过程,着重讨论不同继承方式下的基类成员的访问控制问题、添加构造函数和析构函数。接着讨论在较为复杂的继承关系中,类成员的唯一标识和访问问题。“深度探索”介绍组合与继承的区别与联系、派生类对象的内存布局,以及基类向派生类的转换及其安全性问题。

第8章 多态性:讲述类的另一个重要特性—多态性。多态是指同样的消息被不同类型的对象接收时导致完全不同的行为,是对类的特定成员函数的再抽象。C++语言支持的多态有多种类型,重载(包括函数重载和运算符重载)和虚函数是其中主要的方式。“深度探索”介绍多态类型与非多态类型的区别、运行时类型识别机制,以及虚函数动态绑定的实现原理。

第9章 流类库与输入输出:讲述流的概念,然后介绍流类库的结构和使用。就像C语言一样,C++语言中也没有输入输出语句。但C++编译系统带有一个面向对象的I/O软件包,它就是I/O流类库。“深度探索”介绍宽字符、宽字符串与宽流,以及对象的串行化问题。

第10章 异常处理。

感谢读者选择使用本书,欢迎您对本书内容提出意见和建议,我们将不胜感激。作者的电子邮件地址:zhengli@mail. tsinghua. edu. cn,来信标题请包含“C++ book”。

作者

2010年5月于清华大学

目 录

第 1 章 绪论	1
1.1 计算机程序设计语言的发展	1
1.1.1 机器语言与汇编语言	1
1.1.2 高级语言	2
1.1.3 面向对象的语言	2
1.2 面向对象的方法	2
1.2.1 面向对象方法的由来	3
1.2.2 面向对象的基本概念	4
1.3 面向对象的软件开发	5
1.3.1 分析	5
1.3.2 设计	5
1.3.3 编程	6
1.3.4 测试	6
1.3.5 维护	6
1.4 信息的表示与存储	6
1.4.1 计算机的数字系统	6
1.4.2 几种进位记数制之间的转换	8
1.4.3 信息的存储单位	10
1.4.4 二进制数的编码表示	11
1.4.5 定点数和浮点数	15
1.4.6 数的表示范围	15
1.4.7 非数值信息的表示	16
1.5 程序开发的基本概念	16
1.5.1 基本术语	16
1.5.2 完整的程序过程	17
1.6 小结	18
习题	18
第 2 章 C++ 简单程序设计	19
2.1 C++ 语言概述	19
2.1.1 C++ 的产生	19
2.1.2 C++ 的特点	20

2.1.3	C++ 程序实例	20
2.1.4	字符集	21
2.1.5	词法记号	22
2.2	基本数据类型和表达式	23
2.2.1	基本数据类型	24
2.2.2	常量	25
2.2.3	变量	27
2.2.4	符号常量	28
2.2.5	运算符与表达式	28
2.2.6	语句	37
2.3	数据的输入与输出	37
2.3.1	I/O 流	37
2.3.2	预定义的插入符和提取符	37
2.3.3	简单的 I/O 格式控制	38
2.4	算法的基本控制结构	39
2.4.1	用 if 语句实现选择结构	39
2.4.2	多重选择结构	41
2.4.3	循环结构	44
2.4.4	循环结构与选择结构的嵌套	49
2.4.5	其他控制语句	52
2.5	自定义数据类型	52
2.5.1	typedef 声明	53
2.5.2	枚举类型 enum	53
2.6	小结	55
习题		56
第 3 章	函数	60
3.1	函数的定义与使用	60
3.1.1	函数的定义	60
3.1.2	函数的调用	61
3.1.3	函数的参数传递	74
3.2	内联函数	77
3.3	带默认形参值的函数	78
3.4	函数重载	80
3.5	C++ 系统函数	82
3.6	小结	85
习题		85

第 4 章 类与对象	88
4.1 面向对象程序设计的基本特点	88
4.1.1 抽象	88
4.1.2 封装	89
4.1.3 继承	90
4.1.4 多态	90
4.2 类和对象	90
4.2.1 类的定义	91
4.2.2 类成员的访问控制	92
4.2.3 对象	93
4.2.4 类的成员函数	94
4.2.5 程序实例	96
4.3 构造函数和析构函数	97
4.3.1 构造函数	97
4.3.2 复制构造函数	99
4.3.3 析构函数	103
4.3.4 程序实例	104
4.4 类的组合	106
4.4.1 组合	106
4.4.2 前向引用声明	110
4.5 UML 图形标识	112
4.5.1 UML 简介	112
4.5.2 UML 类图	113
4.6 结构体和联合体	119
4.6.1 结构体	119
4.6.2 联合体	120
4.7 综合实例——个人银行账户管理程序	123
4.7.1 类的设计	123
4.7.2 源程序及说明	124
4.8 小结	127
习题	127
第 5 章 数据的共享与保护	129
5.1 标识符的作用域与可见性	129
5.1.1 作用域	129
5.1.2 可见性	133
5.2 对象的生存期	133
5.2.1 静态生存期	133

5.2.2 动态生存期.....	134
5.3 类的静态成员	136
5.3.1 静态数据成员.....	137
5.3.2 静态函数成员.....	139
5.4 类的友元	141
5.4.1 友元函数.....	143
5.4.2 友元类.....	144
5.5 共享数据的保护	146
5.5.1 常对象.....	146
5.5.2 用 const 修饰的类成员	147
5.5.3 常引用.....	149
5.6 多文件结构和编译预处理命令	151
5.6.1 C++ 程序的一般组织结构	151
5.6.2 外部变量与外部函数.....	153
5.6.3 标准 C++ 库	155
5.6.4 编译预处理.....	156
5.7 综合实例——个人银行账户管理程序	159
5.8 小结	163
习题.....	163
第 6 章 数组、指针与字符串	165
6.1 数组	165
6.1.1 数组的声明与使用.....	165
6.1.2 数组的存储与初始化.....	167
6.1.3 数组作为函数参数.....	170
6.1.4 对象数组.....	171
6.1.5 程序实例.....	173
6.2 指针	176
6.2.1 内存空间的访问方式.....	176
6.2.2 指针变量的声明.....	177
6.2.3 与地址相关的运算“*”和“&”.....	177
6.2.4 指针的赋值.....	178
6.2.5 指针运算.....	180
6.2.6 用指针处理数组元素.....	182
6.2.7 指针数组.....	183
6.2.8 用指针作为函数参数.....	186
6.2.9 指针型函数.....	187
6.2.10 指向函数的指针	188

6.2.11 对象指针	190
6.3 动态内存分配	195
6.4 用 vector 创建数组对象	201
6.5 深复制与浅复制	203
6.6 字符串	206
6.6.1 用字符数组存储和处理字符串	207
6.6.2 string 类	207
6.7 综合实例——个人银行账户管理程序	211
6.8 小结	218
习题	218
第 7 章 继承与派生	221
7.1 类的继承与派生	221
7.1.1 继承关系举例	221
7.1.2 派生类的定义	222
7.1.3 派生类生成过程	224
7.2 访问控制	226
7.2.1 公有继承	226
7.2.2 私有继承	228
7.2.3 保护继承	230
7.3 类型兼容规则	232
7.4 派生类的构造和析构造函数	234
7.4.1 构造函数	235
7.4.2 复制构造函数	237
7.4.3 析构造函数	238
7.5 派生类成员的标识与访问	240
7.5.1 作用域分辨符	240
7.5.2 虚基类	246
7.5.3 虚基类及其派生类构造函数	248
7.6 程序实例——用高斯消去法解线性方程组	249
7.6.1 算法基本原理	249
7.6.2 程序设计分析	250
7.6.3 源程序及说明	251
7.6.4 运行结果与分析	256
7.7 综合实例——个人银行账户管理程序	257
7.7.1 问题的提出	257
7.7.2 类设计	258
7.7.3 源程序及说明	259

7.7.4 运行结果与分析	264
7.8 小结	265
习题	266
第8章 多态性	268
8.1 多态性概述	268
8.1.1 多态的类型	268
8.1.2 多态的实现	268
8.2 运算符重载	269
8.2.1 运算符重载的规则	270
8.2.2 运算符重载为成员函数	271
8.2.3 运算符重载为非成员函数	275
8.3 虚函数	277
8.3.1 一般虚函数成员	278
8.3.2 虚析构函数	281
8.4 纯虚函数与抽象类	283
8.4.1 纯虚函数	283
8.4.2 抽象类	284
8.5 程序实例——变步长梯形积分算法求解函数的定积分	286
8.5.1 算法基本原理	286
8.5.2 程序设计分析	287
8.5.3 源程序及说明	288
8.5.4 运行结果与分析	291
8.6 综合实例——对个人银行账户管理程序的改进	291
8.7 小结	297
习题	298
第9章 流类库与输入输出	300
9.1 I/O 流的概念及流类库结构	300
9.2 输出流	302
9.2.1 构造输出流对象	302
9.2.2 使用插入运算符和操纵符	303
9.2.3 文件输出流成员函数	307
9.2.4 二进制输出文件	309
9.2.5 字符串输出流	310
9.3 输入流	311
9.3.1 构造输入流对象	311
9.3.2 使用提取运算符	312

9.3.3 输入流操纵符.....	312
9.3.4 输入流相关函数.....	312
9.3.5 字符串输入流.....	316
9.4 输入输出流	317
9.5 综合实例——对个人银行账户管理程序的改进	317
9.6 小结	323
习题.....	323
 第 10 章 异常处理	325
10.1 异常处理的基本思想.....	325
10.2 C++ 异常处理的实现	325
10.2.1 异常处理的语法.....	326
10.2.2 异常接口声明.....	328
10.3 异常处理中的构造与析构.....	329
10.4 标准程序库异常处理.....	330
10.5 综合实例——对个人银行账户管理程序的改进.....	333
10.6 小结.....	336
习题.....	336

第 1 章

绪 论

本章首先从发展的角度概要介绍面向对象程序设计语言的产生和特点、面向对象方法的由来及其基本概念,以及什么是面向对象的软件工程,然后介绍信息在计算机中的表示与存储以及程序的开发过程。

1.1 计算机程序设计语言的发展

语言是一套具有语法、词法规则的系统。语言是思维的工具,思维是通过语言来表述的。计算机程序设计语言是计算机可以识别的语言,用于描述解决问题的方法,供计算机阅读和执行。

1.1.1 机器语言与汇编语言

自从1946年2月世界上第一台数字电子计算机ENIAC诞生以来,在这短暂的60多年间,计算机科学得到了迅猛发展,计算机及其应用已渗透到社会的各个领域,有力地推动了整个信息化社会的发展,计算机已成为信息化社会中必不可少的工具。

计算机系统包括硬件和软件。计算机之所以有如此强大的功能,不仅因为它具有强大的硬件系统,而且依赖于软件系统。软件包括了使计算机运行所需的各种程序及其有关的文档资料。计算机的工作是用程序来控制的,离开了程序,计算机将一事无成。程序是指令的集合。软件工程师将解决问题的方法、步骤编写为由一条条指令组成的程序,输入到计算机的存储设备中。计算机执行这一指令序列,便可完成预定的任务。

所谓指令,就是计算机可以识别的命令。虽然在人类社会,各民族都有丰富的语言用来表达思想、交流感情、记录信息,但计算机却不能识别它们。计算机所能识别的指令形式,只能是简单的“0”和“1”的组合。一台计算机硬件系统能够识别的所有指令的集合,称为它的指令系统。

由计算机硬件系统可以识别的二进制指令组成的语言称为机器语言。毫无疑问,虽然机器语言便于计算机识别,但对于人类来说却是晦涩难懂,更难以记忆。可是在计算机发展的初期,软件工程师们只能用机器语言来编写程序。这一阶段,在人类的自然语言和计算机编程语言之间存在着巨大的鸿沟,软件开发的难度大、周期长,开发出的软件功能却很简单,界面也不友好。

不久,出现了汇编语言,它将机器指令映射为一些可以被人们读懂的助记符,如ADD、SUB等。此时编程语言与人类自然语言间的鸿沟略有缩小,但仍与人类的思维相差甚

远。因为它的抽象层次太低,程序员需要考虑大量的机器细节。

尽管如此,从机器语言到汇编语言,仍是一大进步。这意味着人与计算机的硬件系统不必非得使用同一种语言。程序员可以使用较适合人类思维习惯的语言,而计算机硬件系统仍只识别机器指令。那么两种语言间的沟通如何实现呢?这就需要一种翻译工具(软件)。汇编语言的翻译软件称为汇编程序,它可以将程序员写的助记符直接转换为机器指令,然后再由计算机去识别和执行。

1.1.2 高级语言

高级语言的出现是计算机编程语言的一大进步。它屏蔽了机器的细节,提高了语言的抽象层次,程序中可以采用具有一定含义的数据命名和容易理解的执行语句。这使得在书写程序时可以联系到程序所描述的具体事物。

20 世纪 60 年代末开始出现的结构化编程语言进一步提高了语言的层次。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,这使得编程语言与人类的自然语言更接近。但是二者之间仍有不少差距。主要问题是程序中的数据 and 操作分离,不能够有效地组成与自然界中的具体事物紧密对应的程序成分。

目前应用比较广泛的几种高级语言有 FORTRAN, BASIC, Pascal, C 等。当然本书介绍的 C++ 语言也是高级语言,但它与其他面向过程的高级语言有着根本的不同。

1.1.3 面向对象的语言

面向对象的编程语言与以往各种编程语言的根本不同点在于,它设计的出发点就是为了能更直接地描述客观世界中存在的事物(即对象)以及它们之间的关系。

开发一个软件是为了解决某些问题,这些问题所涉及的业务范围称为该软件的问题域。面向对象的编程语言将客观事物看作具有属性和行为(或称服务)的对象,通过抽象找出同一类对象的共同属性(静态特征)和行为(动态特征),形成类。通过类的继承与多态可以很方便地实现代码重用,大大缩短了软件开发周期,并使得软件风格统一。因此,面向对象的编程语言使程序能够比较直接地反映问题域的本来面目,软件开发人员能够利用人类认识事物所采用的一般思维方法来进行软件开发。

面向对象的程序设计语言经历了一个很长的发展阶段。例如, LISP 家族的面向对象语言、Simula67 语言、Smalltalk 语言以及 CLU, Ada, Modula-2 等语言,或多或少地都引入了面向对象的概念,其中 Smalltalk 是第一个真正的面向对象的程序语言。

然而,应用最广的面向对象程序语言是在 C 语言基础上扩充出来的 C++ 语言。由于 C++ 对 C 兼容,而 C 语言又早已被广大程序员所熟知,所以, C++ 语言也就理所当然地成为应用最广的面向对象程序语言。

1.2 面向对象的方法

程序设计语言是编写程序的工具,因此程序设计语言的发展恰好反映了程序设计方法的演变过程。下面首先初步介绍一下面向对象方法的基本概念和基本思想,学习完本