

清华大学计算机基础教育课程系列教材

# C++ 语言程序设计

郑莉 董渊 傅仕星 编著

清华大学出版社

(京)新登字 158 号

## 内 容 简 介

本书将 C++ 语言作为大学生学习程序设计的入门语言,不仅详细介绍了语言本身,而且介绍了常用的数据结构和算法。全书以面向对象的程序设计方法贯穿始终,每一章都首先阐述面向对象的程序设计思想和方法,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,力求使读者在掌握 C++ 语言的同时,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。针对初学者和自学读者的特点,书中以结合实例讲解基本概念和方法为主,力求将复杂的概念用简洁浅显的语言来描述,做到深入浅出。

本书是一本面向广大非计算机专业人员和初学者的教材,适合作为大中专院校非计算机专业程序设计课程的入门教材。与本书配套的《C++ 语言程序设计习题与实验指导》和《C++ 语言程序设计电子教案》由清华大学出版社出版。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名: C++ 语言程序设计

作 者: 郑莉 董渊 傅仕星

出 版 者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 石磊(E-mail: shl@tup.tsinghua.edu.cn)

印 刷 者: 北京市清华园胶印厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 21.75 字数: 513 千字

版 次: 1999 年 12 月第 1 版 2000 年 11 月第 4 次印刷

书 号: ISBN 7-302-03808-2/TP·2229

印 数: 22001~30000

定 价: 23.00 元

# 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,计算机教育在各国倍受重视,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,也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,也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。



1999 年 12 月

# 前言

## 一、本书的编写背景

C++语言是从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，一是全面兼容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。同时,通过继承与多态性,使程序具有很高的可重用性,使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++语言是面向对象的程序设计语言中应用最广泛的一种。

长期以来,C++语言被认为是较难使用的专业开发语言,非计算机专业人员常常不敢问津,在大学的非计算机专业中也很少被列为必修课讲授。那么C++语言与面向对象的程序设计方法是否真的是比较高深、难以掌握的技术呢?答案是否定的。

其实,C语言在它产生的初期,也只被少数专业开发人员使用。但随着计算机科学的发展,计算机技术已渗透到各学科的研究和应用之中,C语言已经被各专业的工程技术人员广泛应用于本专业的科研开发。C++全面兼容了C,同时提供了比C更严格更安全的语法,从这个意义上讲,C++首先是一个更好的C。

C++语言是一种面向对象的编程语言,而面向对象的程序设计一度被看作是一门比较高深的技术。这是因为在面向对象的系统分析(OOA)和面向对象的系统设计(OOD)理论出现之前,程序员要写一个好的面向对象的程序,首先要学会运用面向对象的方法来认识问题和描述问题。现在,OOP的工作比较简单了,认识问题域与设计系统成分的工作已经在系统分析和设计阶段完成,OOP的工作就是用一种面向对象的编程语言把OOD模型中的每个成分书写出来。

面向对象方法的出现实际上是程序设计方法发展的一个返朴归真的过程。从本质上讲,软件开发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。

那么,学习C++语言是否应该首先学习C语言呢?不是的。虽然C++语言是从C语言发展而来的,但是C++本身也是一个完整的程序设计语言,而且它与C语言的程序设计思想是完全不同的。学习的进程不必严格遵循科学技术的发展进程,只有尽快地掌握

最新的理论和技术,才能站到巨人的肩膀上。

因此,我们认为,C++语言是可以作为程序设计的入门语言来学习的。

## 二、本书的特色

本书是一本面向广大非计算机专业人员和初学者的教材,适合作为大学非计算机专业的程序设计课程的入门教材。本书在出版之前已经在清华大学非计算机专业的“计算机软件技术基础”课中试用,取得了良好的教学效果。

本书的作者多年来在清华大学从事非计算机专业的计算机基础教学。我们感到,非计算机专业的计算机基础课具有以下特点:

1. 程序设计课程的课时少,因此,如果先学C再学C++,则课时不够。

2. 计算机课程少,不可能分别开设程序设计语言、数据结构等课程。但从培养目标来看,学生不仅需要掌握编程语言,也需要掌握基本的数据结构和程序设计方法,具有分析问题和解决问题的能力。

3. 在有限的学时中,应着重讲解基本理论和基础知识,在此基础上对开发环境做最简单的介绍,使学生通过实验熟练掌握程序设计的技巧和方法以及开发环境的使用。

针对上述特点,本书将C++语言作为大学生学习计算机编程的入门语言,不仅详细介绍了语言本身,而且介绍了常用的数据结构和算法。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,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问题域的水平,还需要学习面向对象的软件工程课。

针对初学者和自学读者的特点,本书力求做到深入浅出,将复杂的概念用简洁浅显的语言来讲述,以结合实例讲解基本概念和方法为主。读者可以通过配套的《C++语言程序设计习题与实验指导》中的习题与实验,在实践中达到对本书中内容的深入理解和熟练掌握。这样,读者使用本套教材,可以轻松地入门,循序渐进地提高。

## 三、内容摘要

**第1章 概述:**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,接着讲述构成C++语句的基本部分:字符集、关键字、标识符、操作符等等。本章还介绍了C++的基本数据类型和自定义数据类型,以及算法的三种主要控制结构:顺序、选择和循环结构。

**第3章 函数:**讲述C++语言的函数。在面向对象的程序设计中,函数是模块划分的基本单位,是对处理问题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。本章主要从应用的角度讲述各种函数的定义和使用方法,

特别是系统函数的使用方法。

**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：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——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。

**第5章 C++程序的结构：**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；以及使用多文档结构来组织和编写程序，解决较为复杂的问题。

**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、指针的基本概念以及动态存储分配问题，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，本章介绍了两种方法：使用字符数组和使用 string 类。

**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。最后给出类的继承实例——“用全选主元高斯消去法求解线性方程组”和“一个小型公司的人员信息管理系统”。

**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++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，也是我们学习的重点。最后给出类的多态实例——“变步长梯形积分算法”和“一个小型公司的人员信息管理系统”。

**第9章 群体类：**群体是指由多个数据元素组成的集合体。群体可以分为两个大类：线性群体和非线性群体。本章介绍几种常用的群体类模板的定义、实现和应用举例。

**第10章 群体数据的组织：**讲述群体的组织问题，介绍对数组元素的排序与查找方法。排序（sorting）又称分类或整理，是将一个无序序列调整为有序的过程。查找（searching）是在一个序列中，按照某种方式找出需要的特定数据元素的过程。

**第11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。

**第12章 异常处理：**讲述异常处理问题。异常是一种程序定义的错误，C++中，异常处理是对所能预料的运行错误进行处理的一套实现机制。try、throw 和 catch 语句就是C++语言中用于实现异常处理的机制。有了C++异常处理，程序可以向更高的执行上下文传递意想不到的事件，这样程序能更好地从这些异常事件中恢复过来。

**第13章 MFC库与Windows程序开发概述：**讲述Windows程序设计的基本思想和MFC库的基本组成。

## 四、使用指南

本书全部内容需要的讲课学时数为 32 学时,实验学时数为 32 学时,课外上机学时数为 32 学时,课内外共 96 学时,每学时 45 分钟。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 2 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 4 学时,第 10 章 2 学时,第 11 章 2 学时,第 12 章 2 学时,第 13 章 2 学时。课程学时数较少的学校,可以用 28 个讲课学时只讲第 1~11 章,或用 20 学时讲第 1~8 章。实验学时数的分配请参考配套的《C++ 语言程序设计习题与实验指导》。

与本书配套的教材有:《C++ 语言程序设计习题与实验指导》、《C++ 语言程序设计电子教案》。《C++ 语言程序设计习题与实验指导》对本书中的习题进行了补充,并提供解答;同时还设计了与本书各章配套的实验,以帮助读者加深对本书内容的理解,并达到熟练掌握。《C++ 语言程序设计电子教案》是用 PowerPoint 制作的教案,可以在讲课时用联机大屏幕投影演示,也可以输出成投影胶片或幻灯片,部分或全部取代板书。这一电子教案已在清华大学本科生的“计算机软件技术基础”课和清华大学远程教育培训课“C++ 程序设计”中使用过,取得了良好的教学效果。教师不仅可以直接使用本教案,还可以方便地修改和重组其中的内容以适应自己的教学需要。使用本教案可以大大减少教师备课时编写教案的工作量以及书写板书所耗费的精力和时间,因而可以提高单位课时的知识含量,并以生动直观的形式达到板书所难以达到的效果。

本书第 1~3 章、第 9~13 章由郑莉编写,第 4~8 章由董渊、郑莉共同编写,傅仕星编写了本书的全部习题并对例题进行了验证,田荣牌、孟鸿俐参加了部分编写工作。特别感谢王行言教授审阅了全书并提出了许多宝贵建议。

对于选择本书作为教材的学校,作者可提供本书中例题、习题解答及实验的源程序代码。有需要的学校可以来信索取。来信请写明:学校名称、任课教师姓名、电子邮件地址、通信地址、学生人数和专业,并加盖学校或系教务部门章。我们收到来信后即刻以电子邮件形式发送上述资料。

感谢读者选择使用本书,欢迎您对本书内容提出批评和修改建议,我们将不胜感激,并在再版时予以考虑。作者的联系地址如下:

电子邮件地址:zhli@mailx.cic.tsinghua.edu.cn

通信地址:北京 清华大学计算机与信息管理中心(主楼 217) 郑莉 收

邮政编码:100084

作 者

1999 年 10 月于清华大学

# 目 录

第 1 章 概述 .....	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 .....	7
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 .....	11
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 程序的开发过程 .....	16
1.5.1 基本术语 .....	16
1.5.2 程序的开发过程 .....	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 .....	19

2.1.3	C++ 程序实例 .....	20
2.1.4	字符集 .....	21
2.1.5	词法记号 .....	21
2.2	基本数据类型和表达式 .....	23
2.2.1	基本数据类型 .....	23
2.2.2	常量 .....	24
2.2.3	变量 .....	26
2.2.4	符号常量 .....	27
2.2.5	运算符与表达式 .....	27
2.3	数据的输入与输出 .....	35
2.3.1	I/O 流 .....	35
2.3.2	预定义的插入符和提取符 .....	35
2.3.3	简单的 I/O 格式控制 .....	36
2.4	算法的基本控制结构 .....	36
2.4.1	用 if 语句实现选择结构 .....	37
2.4.2	多重选择结构 .....	38
2.4.3	循环结构 .....	42
2.4.4	循环结构与选择结构的嵌套 .....	47
2.4.5	其他控制语句 .....	48
2.5	自定义数据类型 .....	48
2.5.1	类型定义语句——typedef .....	48
2.5.2	枚举类型——enum .....	49
2.5.3	结构体 .....	53
2.5.4	联合体 .....	56
2.6	小结 .....	58
	习题 .....	59
<b>第 3 章</b>	<b>函数 .....</b>	<b>63</b>
3.1	函数的定义与使用 .....	63
3.1.1	函数的定义 .....	63
3.1.2	函数的调用 .....	64
3.1.3	函数的参数传递 .....	70
3.2	内联函数 .....	74
3.3	带缺省形参值的函数 .....	75
3.4	函数重载 .....	77
3.5	函数模板 .....	78
3.6	使用 C++ 系统函数 .....	79
3.7	小结 .....	81

习题 .....	81
<b>第 4 章 类与对象 .....</b>	<b>84</b>
4.1 面向对象的思想 .....	84
4.1.1 结构化程序设计 .....	84
4.1.2 面向对象程序设计 .....	85
4.2 面向对象程序设计的基本特点 .....	86
4.2.1 抽象 .....	86
4.2.2 封装 .....	87
4.2.3 继承 .....	87
4.2.4 多态 .....	88
4.3 类和对象 .....	88
4.3.1 类的定义 .....	89
4.3.2 类成员的访问控制 .....	90
4.3.3 类的成员函数 .....	91
4.3.4 对象 .....	92
4.3.5 程序实例 .....	93
4.4 构造函数和析构函数 .....	94
4.4.1 构造函数 .....	94
4.4.2 拷贝构造函数 .....	95
4.4.3 析构函数 .....	98
4.4.4 程序实例 .....	99
4.5 类的聚集 .....	101
4.5.1 类的聚集 .....	101
4.5.2 前向引用声明 .....	104
4.6 类模板 .....	104
4.7 面向对象标记 .....	107
4.8 小结 .....	109
习题 .....	110
<b>第 5 章 C++ 程序的结构 .....</b>	<b>111</b>
5.1 作用域与可见性 .....	111
5.1.1 作用域 .....	111
5.1.2 可见性 .....	113
5.2 生存期 .....	114
5.2.1 静态生存期 .....	114
5.2.2 动态生存期 .....	114
5.3 数据与函数 .....	116

5.3.1	局部变量	116
5.3.2	全局变量	116
5.3.3	数据成员	117
5.4	静态成员	117
5.4.1	静态数据成员	118
5.4.2	静态函数成员	119
5.5	友元	121
5.5.1	友元函数	121
5.5.2	友元类	122
5.6	共享数据的保护	123
5.6.1	常引用	123
5.6.2	常对象	124
5.6.3	用 const 修饰的对象成员	124
5.7	多文件结构和编译预处理命令	126
5.7.1	C++ 程序的一般组织结构	126
5.7.2	编译预处理程序	128
5.8	小结	132
	习题	133
<b>第 6 章</b>	<b>数组、指针与字符串</b>	<b>135</b>
6.1	数组	135
6.1.1	数组的说明与使用	136
6.1.2	数组的存储与初始化	137
6.1.3	数组作为函数参数	139
6.1.4	对象数组	141
6.1.5	程序实例	141
6.2	指针	144
6.2.1	指针的说明与使用	145
6.2.2	指针的赋值	146
6.2.3	指针运算	148
6.2.4	对象的指针	149
6.3	动态内存分配	152
6.3.1	new 运算	152
6.3.2	delete 运算	153
6.3.3	动态内存分配与释放函数	153
6.3.4	程序实例	153
6.4	指针与数组	160
6.4.1	指向数组元素的指针	160

6.4.2	指针数组 .....	162
6.5	指针与函数 .....	164
6.5.1	指针作为函数参数 .....	164
6.5.2	指针型函数 .....	165
6.5.3	函数指针 .....	166
6.6	字符串 .....	168
6.6.1	用字符数组存储和处理字符串 .....	168
6.6.2	string 类 .....	170
6.7	小结 .....	173
	习题 .....	174
<b>第 7 章</b>	<b>继承与派生 .....</b>	<b>176</b>
7.1	继承与派生 .....	176
7.1.1	派生与继承的实例 .....	176
7.1.2	派生类的定义 .....	177
7.1.3	派生类生成过程 .....	178
7.2	访问控制 .....	179
7.2.1	公有继承 .....	180
7.2.2	私有继承 .....	182
7.2.3	保护继承 .....	184
7.3	派生类的构造函数和析构函数 .....	186
7.3.1	构造函数 .....	186
7.3.2	析构函数 .....	189
7.4	派生类成员的标识与访问 .....	191
7.4.1	作用域分辨 .....	191
7.4.2	虚基类 .....	196
7.5	赋值兼容规则 .....	198
7.6	程序实例——用高斯消去法解线性方程组 .....	201
7.6.1	算法基本原理 .....	201
7.6.2	程序设计分析 .....	202
7.6.3	源程序及说明 .....	203
7.6.4	运行结果与分析 .....	208
7.7	程序实例——人员信息管理 .....	209
7.7.1	问题的提出 .....	209
7.7.2	类设计 .....	209
7.7.3	源程序及说明 .....	209
7.7.4	运行结果与分析 .....	215
7.8	小结 .....	216

习题	217
<b>第 8 章 多态性</b>	<b>218</b>
8.1 多态性概述	218
8.1.1 多态的类型	218
8.1.2 多态的实现	219
8.2 运算符重载	219
8.2.1 运算符重载的规则	219
8.2.2 运算符重载为成员函数	220
8.2.3 运算符重载为友元函数	224
8.3 虚函数	226
8.3.1 一般虚函数成员	226
8.3.2 虚析构造函数	228
8.4 抽象类	228
8.4.1 纯虚函数	229
8.4.2 抽象类	229
8.5 程序实例——变步长梯形积分算法	231
8.5.1 算法基本原理	231
8.5.2 程序设计分析	232
8.5.3 源程序及说明	233
8.5.4 运行结果与分析	236
8.6 程序实例——人员信息管理程序	236
8.7 小结	243
习题	244
<b>第 9 章 群体类</b>	<b>245</b>
9.1 线性群体	245
9.1.1 线性群体的概念	245
9.1.2 直接访问群体——数组类	248
9.1.3 顺序访问群体——链表类	254
9.1.4 栈类	267
9.1.5 队列类	275
9.2 非线性群体	279
9.2.1 树	279
9.2.2 二叉树	280
9.2.3 二叉树结点类	280
9.2.4 基本二叉树类	282
9.3 小结	284

习题 .....	284
<b>第 10 章 群体数据的组织 .....</b>	<b>287</b>
10.1 插入排序 .....	287
10.2 选择排序 .....	288
10.3 交换排序 .....	290
10.4 顺序查找 .....	291
10.5 折半查找 .....	292
10.6 小结 .....	293
习题 .....	294
<b>第 11 章 流类库与输入/输出 .....</b>	<b>295</b>
11.1 I/O 流的概念 .....	295
11.2 输出流 .....	297
11.2.1 构造输出流对象 .....	297
11.2.2 使用插入运算符和控制格式 .....	297
11.2.3 输出文件流成员函数 .....	302
11.2.4 二进制输出文件 .....	304
11.3 输入流 .....	305
11.3.1 构造输入流对象 .....	305
11.3.2 使用提取运算符 .....	306
11.3.3 输入流操纵符 .....	306
11.3.4 输入流成员函数 .....	306
11.4 输入/输出流 .....	310
11.5 小结 .....	310
习题 .....	310
<b>第 12 章 异常处理 .....</b>	<b>312</b>
12.1 异常处理的基本思想 .....	312
12.2 C++ 异常处理的实现 .....	312
12.2.1 异常处理的语法 .....	313
12.2.2 异常接口声明 .....	315
12.3 异常处理中的构造与析构 .....	316
12.4 小结 .....	318
习题 .....	319
<b>第 13 章 MFC 库与 Windows 程序开发概述 .....</b>	<b>320</b>
13.1 Windows 的编程模式 .....	320

---

13.2	MFC 库简介 .....	321
13.2.1	应用程序框架 .....	322
13.2.2	MFC 类层次 .....	322
13.2.3	MFC 类功能简介 .....	322
13.3	使用 Visual C++ 开发 Windows 程序的步骤 .....	326
13.4	小结 .....	329
	习题 .....	329
	参考文献 .....	330

# 第 1 章

## 概述

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,在这暂短的 50 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。