



普通高等教育“十一五”国家级规划教材

国家精品课程教材

清华大学计算机系列教材

C++语言程序设计 (第4版)

郑莉 董渊 何江舟 编著



清华大学出版社



普通高等教育“十一五”国家级规划教材

国家精品课程教材

清华大学计算机系列教材

C++语言程序设计 (第4版)

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、输入输出流,以及泛型程序设计。此外,本教材还介绍了一些常用数据结构基础知识,使得读者学习本书后,能够解决一些简单的实际问题。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++ 语言程序设计 / 郑莉,董渊,何江舟编著. —4 版. —北京:清华大学出版社,2010.7
(清华大学计算机系列教材)

ISBN 978-7-302-22798-4

I. ①C… II. ①郑… ②董… ③何… 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 095875 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:清华大学印刷厂

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:34.25

字 数:784 千字

版 次:2010 年 7 月第 4 版

印 次:2010 年 7 月第 1 次印刷

印 数:1~8000

定 价:48.00 元

产品编号:027971-01

前 言

一、版本说明

本书第 1 版于 1999 年出版,第 2 版于 2001 年出版,第 3 版于 2003 年出版。第 4 版是在前三版的基础上,广泛听取了读者和同行的建议,参考了最新的资料,并根据作者在授课过程中的经验而形成的。第 4 版主要修改内容如下。

- (1) 删除了原来的第 13 章。
- (2) 在第 2~12 章中,每章增加了一节“深度探索”,作为选学内容。
- (3) 编写了贯穿第 4~10 章的新的综合实例“个人银行账户管理程序”。
- (4) 重新编写了第 10 章。

二、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信。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,很少被作为入门语言来讲授。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、难以掌握的技术吗?不是的!

其实 C 语言产生的初期,也只在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,也被很多学校作为第一门程序设计语言来讲授。C++ 全面兼容了 C 语言,同时提供了比 C 语言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 语言。

C++ 语言是一个面向对象的编程语言,而面向对象的编程方法(OOP)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 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璞归真过程。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。

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。我们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。

三、本书的特色

本书的特色是内容全面、深入浅出、灵活剪裁、立体配套。

本书是一本面向广大初学者的入门教材,自 1999 年第一版出版以来,已经在清华大学等 100 多所学校的不同专业中使用,取得了良好的教学效果。

本书将 C++ 语言作为计算机编程入门语言,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、面向对象的设计思想和编程方法、UML 建模语言。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。不同学校可以针对自身的教学特点,选择不同的章节组合进行教学。如果学时较少,可以只选择第 1~8 章、第 11 章。每一章的深度探索也是选学内容,可以根据不同专业的教学需求进行选择。

作者使用本书授课时的讲课学时数为 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 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 4 学时,第 10 章 2 学时,第 11 章 2 学时,第 12 章 2 学时。

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》。

读者还可以通过配套的《C++ 语言程序设计学生用书》,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：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：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。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：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。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、指针的基本概念和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++ 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 群体类和群体数据的组织：群体是指由多个数据元素组成的集合体。群体

可以分为两大类：线性群体和非线性群体。本章介绍几种常用的群体类模板。

本章讨论的群体的组织问题,指的是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。最后“深度探索”介绍模板的实例化机制、为模板定义特殊的实现。

第 10 章 泛型程序设计与 C++ 标准模板库: 泛型程序设计就是将程序写得尽可能通用,同时并不损失效率。本章简单介绍 C++ 标准模板库(STL)中涉及的一些概念、术语,以及它的结构、主要的组件的使用方法。重点介绍容器、迭代器、算法和函数对象的基本应用。目的是使读者对 STL 与泛型程序设计方法有一个概要性的了解。“深度探索”深入介绍深复制与浅复制的问题,另外还介绍了 STL 组件的类型特征与 STL 的扩展问题,以及 Boost 库。

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。“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 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。“深度探索”介绍异常安全性问题和避免异常发生时的资源泄漏。

五、作者分工

本书第 1~3 章和第 9,11,12 章由郑莉编写,第 4~8 章由董渊、郑莉共同编写,第 10 章由何江舟编写,各章的综合实例和深度探索由何江舟编写,全书例题由何江舟调试。傅仕星、张瑞丰、田荣牌、孟鸿利、刘才良参加了部分编写工作。另外特别感谢谭浩强教授和王行言教授审阅了本书第 2 版,并提出了许多宝贵建议。

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 3 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	深度探索	55
2.6.1	变量的实现机制	55
2.6.2	C++ 表达式的执行原理	58
2.7	小结	60
	习题	60
第3章	函数	64
3.1	函数的定义与使用	64
3.1.1	函数的定义	64
3.1.2	函数的调用	65
3.1.3	函数的参数传递	78
3.2	内联函数	81
3.3	带默认形参值的函数	82
3.4	函数重载	84
3.5	C++ 系统函数	86

3.6	深度探索	89
3.6.1	运行栈与函数调用的执行	89
3.6.2	函数声明与类型安全	94
3.7	小结	95
	习题	96
第4章	类与对象	98
4.1	面向对象程序设计的基本特点	98
4.1.1	抽象	98
4.1.2	封装	99
4.1.3	继承	100
4.1.4	多态	100
4.2	类和对象	100
4.2.1	类的定义	101
4.2.2	类成员的访问控制	102
4.2.3	对象	103
4.2.4	类的成员函数	104
4.2.5	程序实例	106
4.3	构造函数和析构函数	107
4.3.1	构造函数	107
4.3.2	复制构造函数	109
4.3.3	析构函数	113
4.3.4	程序实例	114
4.4	类的组合	116
4.4.1	组合	116
4.4.2	前向引用声明	120
4.5	UML 图形标识	122
4.5.1	UML 简介	122
4.5.2	UML 类图	123
4.6	结构体和联合体	129
4.6.1	结构体	129
4.6.2	联合体	130
4.7	综合实例——个人银行账户管理程序	133
4.7.1	类的设计	133
4.7.2	源程序及说明	134
4.8	深度探索	137
4.8.1	位域	137
4.8.2	用构造函数定义类型转换	139

4.8.3 对象作为函数参数和返回值的传递方式·····	140
4.9 小结·····	143
习题·····	144
第5章 数据的共享与保护 ·····	146
5.1 标识符的作用域与可见性·····	146
5.1.1 作用域·····	146
5.1.2 可见性·····	150
5.2 对象的生存期·····	150
5.2.1 静态生存期·····	150
5.2.2 动态生存期·····	151
5.3 类的静态成员·····	153
5.3.1 静态数据成员·····	154
5.3.2 静态函数成员·····	156
5.4 类的友元·····	158
5.4.1 友元函数·····	160
5.4.2 友元类·····	161
5.5 共享数据的保护·····	163
5.5.1 常对象·····	163
5.5.2 用 const 修饰的类成员·····	164
5.5.3 常引用·····	166
5.6 多文件结构和编译预处理命令·····	168
5.6.1 C++ 程序的一般组织结构·····	168
5.6.2 外部变量与外部函数·····	170
5.6.3 标准 C++ 库·····	172
5.6.4 编译预处理·····	173
5.7 综合实例——个人银行账户管理程序·····	176
5.8 深度探索·····	180
5.8.1 常成员函数的声明原则·····	180
5.8.2 代码的编译连接与执行过程·····	182
5.9 小结·····	185
习题·····	185
第6章 数组、指针与字符串 ·····	188
6.1 数组·····	188
6.1.1 数组的声明与使用·····	188
6.1.2 数组的存储与初始化·····	190
6.1.3 数组作为函数参数·····	193

6.1.4	对象数组	194
6.1.5	程序实例	196
6.2	指针	199
6.2.1	内存空间的访问方式	199
6.2.2	指针变量的声明	200
6.2.3	与地址相关的运算“*”和“&.”	200
6.2.4	指针的赋值	201
6.2.5	指针运算	203
6.2.6	用指针处理数组元素	205
6.2.7	指针数组	206
6.2.8	用指针作为函数参数	209
6.2.9	指针型函数	210
6.2.10	指向函数的指针	211
6.2.11	对象指针	213
6.3	动态内存分配	218
6.4	用 vector 创建数组对象	224
6.5	深复制与浅复制	226
6.6	字符串	229
6.6.1	用字符数组存储和处理字符串	230
6.6.2	string 类	230
6.7	综合实例——个人银行账户管理程序	234
6.8	深度探索	241
6.8.1	指针与引用	241
6.8.2	指针的安全性隐患及其应对方案	243
6.8.3	const_cast 的应用	246
6.9	小结	248
	习题	248
第 7 章	继承与派生	251
7.1	类的继承与派生	251
7.1.1	继承关系举例	251
7.1.2	派生类的定义	252
7.1.3	派生类生成过程	254
7.2	访问控制	256
7.2.1	公有继承	256
7.2.2	私有继承	258
7.2.3	保护继承	260
7.3	类型兼容规则	262

7.4	派生类的构造和析构函数	264
7.4.1	构造函数	265
7.4.2	复制构造函数	267
7.4.3	析构函数	268
7.5	派生类成员的标识与访问	270
7.5.1	作用域分辨符	270
7.5.2	虚基类	276
7.5.3	虚基类及其派生类构造函数	278
7.6	程序实例——用高斯消去法解线性方程组	279
7.6.1	算法基本原理	279
7.6.2	程序设计分析	280
7.6.3	源程序及说明	281
7.6.4	运行结果与分析	286
7.7	综合实例——个人银行账户管理程序	287
7.7.1	问题的提出	287
7.7.2	类设计	288
7.7.3	源程序及说明	289
7.7.4	运行结果与分析	294
7.8	深度探索	295
7.8.1	组合与继承	295
7.8.2	派生类对象的内存布局	298
7.8.3	基类向派生类的转换及其安全性问题	301
7.9	小结	303
	习题	304
第8章	多态性	306
8.1	多态性概述	306
8.1.1	多态的类型	306
8.1.2	多态的实现	306
8.2	运算符重载	307
8.2.1	运算符重载的规则	308
8.2.2	运算符重载为成员函数	309
8.2.3	运算符重载为非成员函数	313
8.3	虚函数	315
8.3.1	一般虚函数成员	316
8.3.2	虚析构函数	319
8.4	纯虚函数与抽象类	321
8.4.1	纯虚函数	321

8.4.2 抽象类	322
8.5 程序实例——变步长梯形积分算法求解函数的定积分	324
8.5.1 算法基本原理	324
8.5.2 程序设计分析	325
8.5.3 源程序及说明	326
8.5.4 运行结果与分析	329
8.6 综合实例——对个人银行账户管理程序的改进	329
8.7 深度探索	335
8.7.1 多态类型与非多态类型	335
8.7.2 运行时类型识别	337
8.7.3 虚函数动态绑定的实现原理	340
8.8 小结	343
习题	344
第9章 群体类和群体数据的组织	346
9.1 函数模板与类模板	347
9.1.1 函数模板	347
9.1.2 类模板	350
9.2 线性群体	353
9.2.1 线性群体的概念	353
9.2.2 直接访问群体——数组类	355
9.2.3 顺序访问群体——链表类	363
9.2.4 栈类	368
9.2.5 队列类	374
9.3 群体数据的组织	377
9.3.1 插入排序	377
9.3.2 选择排序	379
9.3.3 交换排序	380
9.3.4 顺序查找	381
9.3.5 折半查找	381
9.4 综合实例——对个人银行账户管理程序的改进	382
9.5 深度探索	385
9.5.1 模板的实例化机制	385
9.5.2 为模板定义特殊的实现	388
9.5.3 模板元编程简介	392
9.6 小结	395
习题	395

第 10 章 泛型程序设计与 C++ 标准模板库	398
10.1 泛型程序设计及 STL 的结构	398
10.1.1 泛型程序设计的基本概念	398
10.1.2 STL 简介	399
10.2 迭代器	402
10.2.1 输入流迭代器和输出流迭代器	403
10.2.2 迭代器的分类	405
10.2.3 迭代器的区间	407
10.2.4 迭代器的辅助函数	409
10.3 容器	410
10.3.1 容器的基本功能与分类	410
10.3.2 顺序容器	412
10.3.3 关联容器	427
10.4 函数对象	435
10.4.1 函数对象基本概念及分类	436
10.4.2 函数适配器	441
10.5 算法	447
10.5.1 STL 算法基础	448
10.5.2 不可变序列算法	448
10.5.3 可变序列算法	451
10.5.4 排序和搜索算法	455
10.5.5 数值算法	459
10.6 综合实例——对个人银行账户管理程序的改进	461
10.7 深度探索	467
10.7.1 swap	467
10.7.2 STL 组件的类型特征与 STL 的扩展	469
10.7.3 Boost 简介	476
10.8 小结	478
习题	478
第 11 章 流类库与输入输出	481
11.1 I/O 流的概念及流类库结构	481
11.2 输出流	483
11.2.1 构造输出流对象	483
11.2.2 使用插入运算符和操纵符	484
11.2.3 文件输出流成员函数	488
11.2.4 二进制输出文件	490
11.2.5 字符串输出流	491

11.3	输入流	492
11.3.1	构造输入流对象	492
11.3.2	使用提取运算符	493
11.3.3	输入流操纵符	493
11.3.4	输入流相关函数	493
11.3.5	字符串输入流	497
11.4	输入输出流	498
11.5	综合实例——对个人银行账户管理程序的改进	498
11.6	深度探索	503
11.6.1	宽字符、宽字符串与宽流	503
11.6.2	对象的串行化	507
11.7	小结	510
	习题	510
第 12 章	异常处理	512
12.1	异常处理的基本思想	512
12.2	C++ 异常处理的实现	512
12.2.1	异常处理的语法	513
12.2.2	异常接口声明	515
12.3	异常处理中的构造与析构	516
12.4	标准程序库异常处理	517
12.5	综合实例——对个人银行账户管理程序的改进	520
12.6	深度探索	523
12.6.1	异常安全性问题	523
12.6.2	避免异常发生时的资源泄漏	525
12.7	小结	528
	习题	529

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。虽然在人类社会中,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