

21世纪高等学校规划教材 | 计算机科学与技术

C++高级 程序设计教程

张 娜 编著



清华大学出版社

张 娜 著

21世纪高等学校规划教材 | 计算机科学与技术



C++高级程序设计教程

张 娜 编著

清华大学出版社
北京

内 容 简 介

本书系统地讲解了C++的语法规则,重点介绍了面向对象程序设计方法。全书分为10章,包括C++编程简介、C++语言的基础知识、C++在非面向对象方面的常用新特性、类和对象、关于类和对象的进一步讨论、运算符的重载、类的继承与派生、多态性和虚函数,以及模板、字符串和异常,C++的输入与输出。

每章内容包括学习目标、综合实例、习题等环节,在体系结构上力求分散难点、突出重点,在语言叙述上逻辑清晰,引入了较多的示例并对示例中涉及的语法、编写思路和输出结果进行详尽的解释和分析,通过案例教学法让内容深入浅出,所有的示例都在Visual Studio C++环境下运行通过。本书配有相应的教学课件、习题答案以及所有示例、综合实例的源代码。

本书可以作为高等学校计算机专业以及软件学院、高职院校计算机专业“面向对象程序设计(C++程序设计)”课程的教材。

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++高级程序设计教程/张娜编著. —北京:清华大学出版社,2017

(21世纪高等学校规划教材·计算机科学与技术)

ISBN 978-7-302-47053-3

I. ①C… II. ①张… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆CIP数据核字(2017)第112309号

责任编辑:贾 斌 王冰

封面设计:傅瑞学

责任校对:徐俊伟

责任印制:沈 露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦A座 邮 编: 100084

社总机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者: 北京泽宇印刷有限公司

经 销: 全国新华书店

开 本: 185mm×260mm

印 张: 21.25

字 数: 606千字

版 次: 2017年10月第1版

印 次: 2017年10月第1次印刷

印 数: 1~2000

定 价: 59.00元

产品编号: 073607-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程”(简称“质量工程”),通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制定精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上。精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展,顺应并符合21世纪教学发展的规律,代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版

社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色精品教材包括:

- (1) 21 世纪高等学校规划教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。
- (2) 21 世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。
- (3) 21 世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。
- (4) 21 世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。
- (5) 21 世纪高等学校规划教材·信息管理与信息系统。
- (6) 21 世纪高等学校规划教材·财经管理与应用。
- (7) 21 世纪高等学校规划教材·电子商务。
- (8) 21 世纪高等学校规划教材·物联网。

清华大学出版社经过三十多年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

联系人:魏江江

E-mail: weijj@tup.tsinghua.edu.cn

C++语言是一种面向对象的编程语言,它是当今最主流的面向对象编程语言之一,它的功能很强大,很容易被人们理解和接受。目前,大多数高校计算机专业和IT培训学校都将C++作为基础的教学内容之一,这对于培养学生的基础编程能力具有非常重要的意义。

本书从教学的实际需求出发,结合初学者的认知规律,由浅入深、循序渐进地讲解了与C++程序设计相关的知识,重点讲解了C++面向对象的编程知识,并将C++语言知识和使用的示例有机结合起来,使知识和示例相辅相成,既有利于读者学习知识,又有利于指导读者实践,有效地强化了实践教学。具体来讲,本书具有以下特色:

(1)本书中的所有实例程序都是完整的,通过Visual Studio C++调试,能够在Windows XP、Windows 7系统下编译和运行。

(2)结合大量直观的示例来讲解各部分的内容,提供了程序设计实现的具体步骤,使初学者快速掌握C++的编程方法,体会C++的简单易学。

(3)本书的结构完整,根据循序渐进的认知规律设计各章节的内容。

(4)每章都配有学习目标、综合实例、本章小结和习题,可以帮助读者巩固所学的知识点,锻炼读者的实际动手能力。

全书分为10章,各章的基本内容如下。

第1章:介绍了面向对象的方法及其基本概念、C和C++的关系、C++程序的编写和实现,概括地说明了Visual Studio 2010集成开发环境,并以示例使读者熟悉开发C++程序的几种操作过程。

第2章:对C++语言的基础知识进行了介绍,包括标识符与关键字、常量与变量、运算符与表达式、语句及其流程控制、函数、构造数据类型、指针、数组与指针的关系、结构体与共用体,旨在使读者对C++语言有一个初步认识,为全书的学习打下基础,也为之前学过C语言的同学对C++语言中的C语言知识做一个简单的复习总结。

第3章:讲解了C++在非面向对象方面的常用新特性,主要讲解与C语言的不同,包括简单的C++输入与输出、用const定义常量、函数的重载、有默认参数的函数、引用、动态分配内存、布尔类型、函数原型、作用域运算符、内置函数以及C++的注释,目的是在第2章的基础上引入C++在非面向对象方面的新特性。

第4章:介绍了面向对象的概念、类、对象的创建、类的成员函数、对象成员的引用、构造函数、析构函数、对象数组、对象指针、对象成员、对象创建时内存的动态分配。

第5章:关于类和对象的进一步讨论,主要讲解了类的封装性、作用域和可见性、类的静态成员、友元、类模板、结构体和类、联合体和类、共享数据的保护。

第6章:讲解了运算符的重载,主要包括运算符重载的一般概念、重载运算符的实现,以及单目运算符、双目运算符和特殊运算符的重载。

第7章:类的继承与派生,主要包括继承与派生的概念、单继承、继承中的构造函数与

析构函数、多继承与虚基类等。

第8章：多态性和虚函数，主要讲解 C++ 多态的概念、虚函数、虚析构函数、纯虚函数和抽象类。

第9章：模板、字符串和异常，详细介绍 C++ 函数模板、模板类、C++ 模板中的函数式参数；详细描述 C++ 中的 string 类和字符串，string 字符串的访问和拼接，string 字符串的增、删、改、查等操作；介绍了 C++ 异常处理、用 throw 抛出异常和 C++ 中的 exception 类。

第10章：详细介绍了 C++ 的输入与输出，不仅介绍了 C++ 输入输出的概念，而且详细讲解了与 C++ 输入输出有关的类和对象、标准的输出流、C++ 格式化输出，并在本章的最后几节介绍了文件的概念、文件流类与文件流对象、文件的打开与关闭、对 ASCII 文件和二进制文件的读写操作以及对字符串流的读写操作。

本书由张娜编著，作为编者，虽有多年的程序设计语言教学经验和工程实践经验，但也深知在这一领域仍有许多知识尚未融会贯通并正确使用，尽管在编写过程中参考了多部相关教材和参考书，也和学生进行了深入交流，但由于编者水平有限，书中难免存在一些疏漏和不足之处，敬请广大读者批评指正。

编 者

2017 年 5 月

目 录

第 1 章 C++ 编程简介	1
1.1 面向对象的方法	1
1.1.1 面向对象编程	1
1.1.2 面向对象方法的由来	1
1.1.3 面向对象的语言	2
1.2 面向对象的基本概念	3
1.2.1 类和对象	3
1.2.2 面向对象的特点	3
1.3 C 和 C++ 的关系	4
1.4 C++ 程序的编写和实现	7
1.4.1 用 C++ 语言编写程序	7
1.4.2 对源程序进行编译	7
1.4.3 对目标文件进行链接	8
1.4.4 运行程序	8
1.4.5 分析运行结果	8
1.5 如何用 Visual Studio 2010 编写 C++ 程序	8
1.5.1 Visual Studio 2010 编程环境	8
1.5.2 Visual Studio 2010 的启动	9
1.5.3 新建一个 C++ 项目	9
1.5.4 Visual Studio 2010 界面介绍	9
综合实例	13
本章小结	14
习题	14
第 2 章 C++ 语言的基础知识	16
2.1 标识符与关键字	16
2.1.1 标识符	16
2.1.2 关键字	16
2.2 常量与变量	17
2.2.1 常量	17
2.2.2 变量	18
2.3 运算符与表达式	18

2.3.1	算术运算符	19
2.3.2	关系运算符	20
2.3.3	逻辑运算符	20
2.3.4	条件运算符	20
2.3.5	位运算符	21
2.4	语句及其流程控制	21
2.4.1	语句	21
2.4.2	if...else 选择结构	21
2.4.3	switch 选择结构	23
2.4.4	while 语句	24
2.4.5	do...while 语句	26
2.4.6	for 循环结构	27
2.4.7	程序中止函数 exit()	28
2.5	函数	29
2.5.1	函数的结构和函数的定义	29
2.5.2	函数名重载	30
2.5.3	递归函数	31
2.5.4	C++库函数	33
2.6	构造数据类型	34
2.6.1	数组	34
2.6.2	一维数组	34
2.6.3	多维数组	38
2.7	指针	39
2.7.1	指针类型与指针的声明	39
2.7.2	指针变量的引用	40
2.7.3	指向指针的指针	40
2.7.4	指针的运算	42
2.8	数组与指针的关系	43
2.8.1	一维数组指针的表示法	43
2.8.2	二维数组指针的表示法	43
2.8.3	指针与字符串	44
2.8.4	指针数组	46
2.8.5	指针与函数	47
2.8.6	数组指针作为函数参数	48
2.8.7	指向函数的指针	49
2.8.8	返回指针的函数	51
2.8.9	main 函数中的参数	52
2.9	结构体、共用体和枚举数据类型	52
2.9.1	结构体类型概述	52

2.9.2 结构体类型变量的定义和引用	53
2.9.3 结构体数组	55
2.9.4 结构体函数参数以及返回结构体类型值的函数	58
2.9.5 结构体变量和指针	60
2.9.6 枚举类型数据	65
2.9.7 用 typedef 定义类型	67
综合实例	67
本章小结	70
习题	71
 第 3 章 C++ 在非面向对象方面的常用新特性	 73
3.1 C++ 的输入与输出简介	74
3.1.1 用 cout 输出数据流	74
3.1.2 用 cin 输入数据流	74
3.2 用 const 定义常量	76
3.3 函数的重载	78
3.4 有默认参数的函数	80
3.5 引用	81
3.5.1 引用的概念	82
3.5.2 将引用作为函数的参数	83
3.5.3 引用和指针的区别	85
3.5.4 常指针与常引用	85
3.5.5 引用小结	88
3.6 动态分配内存	88
3.6.1 new 关键字	88
3.6.2 delete 运算符	89
3.7 布尔类型	91
3.8 函数原型	92
3.9 作用域运算符	94
3.10 内置函数	95
3.11 C++ 的注释	97
综合实例	98
本章小结	101
习题	102
 第 4 章 类和对象	 104
4.1 面向对象的概念	104
4.1.1 概述	104
4.1.2 面向对象程序设计	105

4.1.3	类和对象简介	106
4.1.4	封装性	107
4.1.5	抽象	107
4.1.6	继承性	108
4.1.7	多态性	108
4.1.8	面向对象程序设计的特点	109
4.1.9	面向对象的软件工程	109
4.2	类	110
4.2.1	类和对象的关系	110
4.2.2	类的定义	111
4.2.3	类和结构体的区别	114
4.3	对象的创建	115
4.4	类的成员函数	121
4.4.1	成员函数的特性	121
4.4.2	内部函数	122
4.4.3	在类外定义成员函数	124
4.4.4	成员函数的存储方式	125
4.5	对象成员的引用	126
4.5.1	通过对象名和成员运算访问对象中的成员	126
4.5.2	通过对象的引用变量访问对象中的成员	127
4.6	构造函数	127
4.6.1	对象的初始化	127
4.6.2	构造函数的作用	129
4.6.3	构造函数的种类	132
4.6.4	用参数初始化表对数据成员初始化	136
4.6.5	构造函数的重载	136
4.6.6	使用默认参数的构造函数	137
4.7	析构函数	139
4.8	对象数组	143
4.8.1	对象数组的定义	143
4.8.2	对象数组的初始化	145
4.9	对象指针	146
4.9.1	指向对象的指针	146
4.9.2	指向对象成员的指针	148
4.9.3	this 指针	152
4.10	对象成员	153
4.11	对象创建时内存的动态分配	156
	综合实例	159
	本章小结	162

习题	163
第 5 章 关于类和对象的进一步讨论	167
5.1 类的封装性	167
5.1.1 公用接口与私有实现的分离	167
5.1.2 类声明和成员函数定义的分离	168
5.2 作用域和可见性	170
5.2.1 标识符的作用域	170
5.2.2 可见性	174
5.2.3 程序实体的生存期	175
5.2.4 C++的存储属性	175
5.3 类的静态成员	177
5.3.1 类静态成员	177
5.3.2 类静态成员函数	179
5.4 友元	181
5.4.1 友元函数	181
5.4.2 友元类	184
5.5 类模板的基本概念	186
5.6 结构体和类	187
5.7 联合体和类	188
5.8 共享数据的保护	190
5.8.1 常对象	190
5.8.2 用 const 修饰的类成员	191
5.8.3 常引用	193
5.8.4 指向对象的常指针	195
5.8.5 指向常对象的指针变量	195
5.9 类模板的进一步讨论	196
综合实例	200
本章小结	201
习题	201
第 6 章 运算符的重载	205
6.1 运算符重载的一般概念	205
6.2 重载运算符的实现	210
6.2.1 重载为类的成员函数	210
6.2.2 重载为类的友元函数	213
6.3 单目运算符重载	214
6.4 双目运算符重载	217
6.5 特殊运算符重载	220

6.5.1 赋值运算符重载	220
6.5.2 下标运算符重载	223
6.5.3 比较运算符重载	223
6.5.4 new 和 delete 运算符重载	225
6.5.5 逗号运算符重载	227
6.5.6 类型转换运算符重载	228
6.5.7 ->运算符重载	234
6.5.8 函数调用运算符重载	235
6.5.9 I/O 运算符重载	236
综合实例	237
本章小结	240
习题	240
第7章 类的继承与派生	241
7.1 继承与派生的概念	241
7.2 单继承	243
7.2.1 单继承和派生	243
7.2.2 重名成员	249
7.3 继承中的构造函数与析构函数	254
7.3.1 继承中的构造函数	254
7.3.2 析构函数	262
7.4 多继承与虚基类	263
7.4.1 多继承	264
7.4.2 多继承中的二义性和支配原则	267
7.4.3 虚基类	269
综合实例	272
本章小结	274
习题	274
第8章 多态性和虚函数	276
8.1 C++多态的概念	276
8.1.1 多态概念的引入	277
8.1.2 构成多态的条件	279
8.2 虚函数	281
8.3 虚析构函数	284
8.4 纯虚函数和抽象类	286
综合实例	289
本章小结	296
习题	296

第 9 章 模板、字符串和异常	298
9.1 C++ 函数模板	298
9.2 模板类	301
9.2.1 模板类的定义	301
9.2.2 模板类的实例化	302
9.3 C++ 模板中的函数式参数	304
9.4 C++ 中的 string 类和字符串	306
9.4.1 string 类的定义	306
9.4.2 转换为 char 数组字符串	307
9.4.3 string 字符串的输入与输出	307
9.5 C++ 中 string 字符串的访问和拼接	308
9.5.1 访问字符串中的字符	308
9.5.2 字符串的拼接	308
9.6 C++ 中 string 字符串的增、删、改、查	309
9.6.1 插入字符串	309
9.6.2 删除字符串	310
9.6.3 提取子字符串	311
9.6.4 字符串的查找	311
9.7 C++ 异常处理	313
9.7.1 C++ 的异常引入	313
9.7.2 捕获异常	314
9.7.3 异常的类型	315
9.8 用 throw 抛出异常	317
9.8.1 throw 关键字	317
9.8.2 不被建议的用法	319
9.9 C++ 中的 exception 类	320
综合实例	322
本章小结	324
习题	324
第 10 章 C++ 的输入与输出	327
10.1 C++ 输入与输出的概念	327
10.1.1 输入与输出	327
10.1.2 C++ 的 I/O 相对于 C 的发展	328
10.2 与 C++ 输入和输出有关的类和对象	329
10.2.1 输入输出流类的概念	329
10.2.2 与 iostream 类库有关的头文件	330
10.2.3 在 iostream 头文件中定义的流对象	331

10.2.4	在 iostream 头文件中重载运算符	331
10.3	C++ 标准输出流详解	332
10.3.1	cout 流对象	332
10.3.2	cerr 流对象	333
10.3.3	clog 流对象	334
10.4	C++ 格式化输出	334
10.4.1	用控制符控制输出格式	335
10.4.2	用流对象的成员函数控制输出格式	336
10.5	用 C++ 流成员函数 put 输出单个字符	339
10.6	cin 输入流详解	341
10.7	用 get() 函数读入一个字符	342
10.7.1	不带参数的 get 函数	342
10.7.2	有一个参数的 get 函数	343
10.7.3	有三个参数的 get 函数	344
10.8	用 getline() 函数读入一行字符	345
10.9	一些与输入有关的 istream 类成员函数	347
10.9.1	eof 函数	347
10.9.2	peek 函数	347
10.9.3	putback 函数	347
10.9.4	ignore 函数	348
10.10	C++ 中文件的概念	350
10.11	C++ 中的文件流类与文件流对象	351
10.12	文件的打开与关闭	352
10.12.1	打开文件	352
10.12.2	关闭磁盘文件	354
10.13	对 ASCII 文件的读写操作	354
10.14	对二进制文件的读写操作	359
10.14.1	用成员函数 read 和 write 读写二进制文件	360
10.14.2	与文件指针有关的流成员函数	362
10.14.3	随机访问二进制数据文件	363
10.15	对字符串流的读写操作	365
10.15.1	建立输出字符串流对象	366
10.15.2	建立输入字符串流对象	366
10.15.3	建立输入输出字符串流对象	367
综合实例		370
本章小结		372
习题		372

第1章

C++编程简介

本章学习目标：

- 掌握面向对象程序设计的概念；
- 掌握类和对象的概念；
- 掌握面向对象的特性；
- 会编写一个简单的 C++ 程序。

1.1 面向对象的方法

1.1.1 面向对象编程

面向对象编程(Object Oriented Programming, OOP, 面向对象程序设计)是一种计算机编程架构,它的基本原则是计算机程序是由单个能够起到子程序作用的单元或对象组合而成。面向对象编程是为了达到软件工程的3个主要目标,即重用性、灵活性和扩展性。

在程序设计发展过程中有两种重要的编程方法,即面向过程程序设计方法和面向对象程序设计方法。面向过程的程序设计是以具体解题过程为研究和实现的主体,而面向对象的程序设计是以解决问题中的各种对象为主体。随着软件开发规模的不断扩大、升级加快、维护量增加以及在软件开发过程中分工日趋精细,面向对象技术能够很好地解决面向过程程序设计难于解决的种种难题。

面向对象技术的核心是以更接近于人类思维的方法建立计算机逻辑模型,即尽可能运用人类的思维方式,以现实世界中的事物为中心思考问题、认识问题,使得软件开发的方法和过程尽可能接近人类认识世界、解决问题的方法与过程。利用类和对象机制将数据及附着在数据之上的操作封装在一起,并通过统一的接口对外交互,使反映现实世界实体的各个类能够在程序中独立、自治和继承。面向对象方法可大大提高程序的可维护性和可重用性,也大大提高了程序开发的效率和程序的可管理性。

1.1.2 面向对象方法的由来

在面向对象的方法出现之前都是采用面向过程的程序设计方法。早期的计算机主要是用于科学计算,程序设计的本质是设计出一个计算方法求解问题。

随着计算机的发展,计算机的性能越来越强,用途也越来越广,不再仅仅限于数学计算,

随着计算任务越来越复杂,程序的规模也越来越复杂。20 世纪 60 年代产生的结构化程序设计思想为面向过程的方法解决复杂问题提供了有力的方法,因而在 20 世纪 70 年代到 80 年代,结构化程序设计方法成为所有软件开发设计领域及每个程序员都采用的方法。结构化程序设计方法的思想是自顶向下、逐步求精;其程序设计结构按照功能划分为若干个基本模块,这些模块形成了一个树状结构;各模块之间的关系尽可能简单,在功能上相对独立,即模块的内聚性强,耦合度低。每一个模块内部均是由顺序、选择和循环 3 种基本结构组成。结构化程序设计由于采用模块分解和功能抽象以及自顶向下、分而治之的方法,从而有效地将一个较复杂的程序系统设计任务分解成许多易于控制和处理的子任务,便于开发和维护。

虽然结构化程序设计方法具有很多的优点,但是它仍是一种面向过程的程序设计方法,它把数据和处理数据的过程分离为相互独立的实体,当数据结构改变时所有相关的处理过程都要进行相应的修改,每一种相对于老问题的新方法都要带来额外的开销,程序的可重用性差。另外,面向过程的程序设计方法的可维护性差,原因是一个好的应用软件不会让用户按照固定的操作顺序,相应地这种软件的功能很难用过程来描述和实现。

而面向对象的程序设计可以屏蔽这些缺点,什么是面向对象的方法呢?首先,它是将数据及数据的操作方法放在一起,作为一个相互依存、不可分割的整体——对象。对于相同类型的对象抽象出共性,形成类。类通过一个简单的外部接口与外界发生联系,而对象与对象之间通过消息进行通信。通过后续章节中类的继承和多态性还可以提高软件的可重用性,使得软件的开发和维护都更为方便。

1.1.3 面向对象的语言

面向对象方法起源于面向对象的编程语言(Object Oriented Programming Language, OOPL)。20 世纪 50 年代后期,在用 FORTRAN 语言编写大型程序时常出现变量名在程序的不同部分发生冲突的问题。鉴于此,ALGOL 语言的设计者在 ALGOL60 中采用了以 Begin...End 为标识的程序块,使程序块内的变量名是局部的,以避免它们与程序块外的同名变量相冲突。这是编程语言中首次提供封装(保护)的尝试。此后程序块结构广泛用于高级语言(例如 Pascal、Ada、C)之中。

20 世纪 60 年代中后期,Simula 语言在 ALGOL 的基础上研制开发,它将 ALGOL 的块结构概念向前发展一步,提出了对象的概念,并使用了类,也支持类继承。20 世纪 70 年代,Smalltalk 语言诞生,它取 Simula 的类为核心概念,它的很多内容借鉴于 Lisp 语言。Xerox 公司对 Smalltalk72/76 持续不断地研究和改进,在系统设计中强调对象概念的统一,引入对象、对象类、方法、实例等概念和术语,采用动态联编和单继承机制。

从 20 世纪 80 年代起,人们基于以往已提出的有关信息隐蔽和抽象数据类型等概念,以及由 Modula2、Ada 和 Smalltalk 等语言所奠定的基础,再加上客观需求的推动,进行了大量的理论研究和实践探索,不同类型的面向对象语言(例如 Object-C、Eiffel、C++、Java、Object-Pascal 等)逐步发展和建立起来,并建立起完整的 OO 方法的概念理论体系和实用的软件系统。

面向对象源自于 Simula,真正的 OOP 由 Smalltalk 奠基。Smalltalk 现在被认为是最纯的 OOPL。正是通过 Smalltalk80 的研制与推广应用,使人们注意到 OO 方法所具有模块