

高 等 学 校 计 算 机 课 程 规 划 教 材

# C++程序设计

袁方 王亮 编著



清华大学出版社



013028369

TP312C-43  
795

高等学校计算机课程规划教材

# C++程序设计

袁方 王亮 编著



北航

C1635046

清华大学出版社

北京

TP312C-43

795

030880010

## 内 容 简 介

本书全面介绍了有关 C++ 程序设计的知识,包括 C++ 入门、基本数据类型与表达式、语句与结构化程序设计、数组与字符串、函数、指针、自定义数据类型、C++ 程序结构、类和对象、继承与派生、运算符重载、多态与虚函数、模板、输入输出流和文件、异常处理、综合实例、集成开发环境等内容。通过本书,读者可以学习程序设计知识、掌握程序设计方法、提高程序设计能力、形成程序设计风格、培养程序设计思维,并为进一步深入学习后续专业课程,提高综合素质和能力奠定良好的基础。

本书既可作为高等学校计算机及相关专业程序设计课程的教材,也可作为自学程序设计的参考书。

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/袁方,王亮编著. —北京:清华大学出版社,2013.4

高等学校计算机课程规划教材

ISBN 978-7-302-31216-1

I. ①C… II. ①袁… ②王… III. ①C 程序—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 001781 号

责任编辑:汪汉友

封面设计:傅瑞学

责任校对:时翠兰

责任印制:宋 林

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](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

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

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:23.75 字 数:581千字

版 次:2013年4月第1版 印 次:2013年4月第1次印刷

印 数:1~3000

定 价:39.00元

产品编号:041162-01

# 出版说明

信息时代早已显现其诱人魅力,当前几乎每个人随身都携有多个媒体、信息和通信设备,享受其带来的快乐和便捷。

我国高等教育早已进入大众化教育时代。而且计算机技术发展很快,知识更新速度也在快速增长,社会对计算机专业学生的专业能力要求也在不断翻新。这就使得我国目前的计算机教育面临严峻挑战。我们必须更新教育观念——弱化知识培养目的,强化对学生兴趣的培养,加强培养学生理论学习、快速学习的能力,强调培养学生的实践能力、动手能力、研究能力和创新能力。

教育观念的更新,必然导致教材的更新。一流的计算机人才需要一流的名师指导,而一流的名师需要精品教材的辅助,而精品教材也将有助于催生更多一流名师。名师们在长期的一线教学改革实践中,总结出了一整套面向学生的独特的教法、经验、教学内容等。本套丛书的目的就是推广他们的经验,并促使广大教育工作者进一步更新教育观念。

在教育部相关教学指导委员会专家的帮助和指导下,在各大学计算机院系领导的协助下,清华大学出版社规划并出版了本系列教材,以满足计算机课程群建设和课程教学的需要,并将各重点大学的优势专业学科的教育优势充分发挥出来。

本系列教材行文注重趣味性,立足课程改革和教材创新,广纳全国高校计算机专业一线优秀名师参与,从中精选出佳作予以出版。

本系列教材具有以下特点。

## 1. 有的放矢

针对计算机专业学生并站在计算机课程群建设、技术市场需求、创新人才培养的高度,规划相关课程群内各门课程的教学关系,以达到教学内容互相衔接、补充、相互贯穿和相互促进的目的。各门课程功能定位明确,并去掉课程中相互重复的部分,使学生既能够掌握这些课程的实质部分,又能节约一些课时,为开设社会需求的新技术课程准备条件。

## 2. 内容趣味性强

按照教学需求组织教学材料,注重教学内容的趣味性,在培养学习观念、学习兴趣的同时,注重创新教育,加强“创新思维”,“创新能力”的培养、训练;强调实践,案例选题注重实际和兴趣度,大部分课程各模块的内容分为基本、加深和拓宽内容3个层次。

## 3. 名师精品多

广罗名师参与,对于名师精品,予以重点扶持,教辅、教参、教案、PPT、实验大纲和实验指导等配套齐全,资源丰富。同一门课程,不同名师分出多个版本,方便选用。

## 4. 一线教师亲力

专家咨询指导,一线教师亲力;内容组织以教学需求为线索;注重理论知识学习,注重学

习能力培养,强调案例分析,注重工程技术能力锻炼。

经济要发展,国力要增强,教育必须先行。教育要靠教师和教材,因此建立一支高水平的教材编写队伍是社会发展的需要,特希望有志于教材建设的教师能够加入到本团队。通过本系列教材的辐射,培养一批热心为读者奉献的编写教师团队。

清华大学出版社

# 前 言

程序设计是计算机专业(包括计算机科学与技术、软工程、网络工程等专业)的一门重要核心课程,程序设计能力是计算机专业学生必备的基本能力,一本合适的教材有助于教师的讲授与学生的学习,有助于学习者程序设计能力的尽快提高。

本书是作者根据多年的教学经验与软件开发体会编写而成的,目的是帮助读者学习程序设计知识、掌握程序设计方法、提高程序设计能力、形成程序设计风格、培养程序设计思维。结构安排与内容选取符合教育部高等学校计算机科学与技术教学指导委员会编制的《高等学校计算机科学与技术专业发展战略研究报告暨专业规范》与《高等学校计算机科学与技术专业核心课程教学实施方案》的要求,适合作为计算机专业和相关专业“程序设计基础”、“高级语言程序设计”、“面向对象程序设计”等程序设计类课程的教材。

全书共分 17 章,各章主要内容如下:

第 1 章 C++ 入门。对 C++ 语言的产生、发展及特点作了简要介绍,并通过实例展示了 C++ 程序的基本结构,使读者在学习具体内容之前尽早对 C++ 语言及程序结构有一个总体了解,有助于对后续章节内容的学习与理解。

第 2~8 章 面向过程程序设计。介绍了基本数据类型与表达式、语句与结构化程序设计、数组与字符串、函数、指针、自定义数据类型、C++ 程序结构等内容。使学习者较好地掌握面向过程程序设计方法,并为学习面向对象程序设计打好基础。

第 9~12 章 面向对象程序设计。介绍了类和对象、构造函数、析构函数、继承与派生、派生类的构造函数和析构函数、基类与派生类的转换、类的继承与组合、运算符重载、多态与虚函数、纯虚函数与抽象类等内容。帮助学习者深入理解面向对象程序设计的基本思想、熟练掌握面向对象程序设计的基本方法,并仔细体会面向对象程序设计方法的优点。

第 13 章 模板。介绍了函数模板、类模板、类模板应用等内容。模板把函数或类要处理数据的数据类型参数化,进一步提高了代码的重用性,是开发大型软件和建立通用函数库与类库的有效工具。

第 14 章 输入输出流和文件。介绍了输入输出与流、C++ 的流类和流对象、标准输入输出流、设定输入输出格式、文件处理等内容。文件的使用,增强了 C++ 程序的数据存储和数据处理能力。

第 15 章 异常处理。介绍了简单异常处理、类类型异常处理等内容。C++ 的异常处理机制将异常的检测与处理分离,实际上是将功能代码与异常处理代码分开,提高了程序的可理解性和可维护性,适合于编写规模比较大的程序,能够有效保证程序的质量。

第 16 章 综合实例。以一个“五子棋游戏”为例,分别用面向过程和面向对象的程序设计方法实现了该游戏的功能。通过该综合实例,读者可实际体验 C++ 程序设计知识的综合应用,并可深入理解面向对象与面向过程程序设计方法的联系与区别。

第 17 章 集成开发环境。以目前较为常用的 Visual Studio 2010 为例,介绍了 C++ 集成开发环境的安装和使用方法,包括 Visual Studio 2010 的安装与启动、Visual Studio 2010



的操作界面、创建和打开项目、生成和运行项目、调试程序等内容。

需要说明的是,程序设计知识的学习,教师的讲解是必要的,有助于学习者较快且准确地理解所学内容,但要真正深入理解并切实掌握程序设计方法,需要在教师讲解的基础上,多读书、多思考、多编写程序、多上机调试程序。只有多读书、多思考,才能把教师的讲解转化为自己的理解,才能深入理解书中所讲内容的真正含义;只有多编写程序、多上机调试程序,才能准确掌握语法格式及常用的程序设计方法,才能逐渐积累程序调试经验。最终实现提高程序设计能力、培养程序设计思维的学习目的。

为方便教师的讲授与学生的学习,本书配有电子教案,完整程序实例都上机调试通过。

本书由袁方提出编写计划和结构安排,袁方编写第1~15章(第8、10、11、15章为部分内容),王亮编写第16~17章,王煜(第8章)、刘宇(第10章)、魏勇刚(第11章)、郝亚辉(第15章)、王苗(习题)分别参与了部分章节内容和习题的编写工作,最后由袁方统编定稿。梁森调试了部分程序,边茜、孟京朝绘制了书中的插图。

本书的编写参考了大量的同类书籍。为此,我们向有关的作者、编者和译者表示衷心的感谢。

清华大学出版社相关人员为本书的出版做了大量细致的工作,在此一并表示衷心的感谢。

由于C++程序设计涉及的内容非常丰富,限于编者水平有限,书中定有不妥之处,敬请读者批评指正,如能提出修改建议和意见,我们非常感谢。联系方式 [cpp1209@163.com](mailto:cpp1209@163.com)。

编 者

2012年9月

# 目 录

<b>第 1 章 C++ 入门</b> .....	1
1.1 C++ 简介 .....	1
1.1.1 C++ 的产生与发展 .....	1
1.1.2 C++ 的特点 .....	2
1.2 简单的 C++ 程序 .....	3
1.2.1 C++ 程序结构 .....	3
1.2.2 输入与输出 .....	5
1.2.3 注释 .....	6
1.3 C++ 程序的上机执行过程 .....	7
1.4 程序设计语言介绍 .....	8
1.4.1 机器语言 .....	8
1.4.2 汇编语言 .....	9
1.4.3 高级语言 .....	10
习题 1 .....	13
<b>第 2 章 基本数据类型与表达式</b> .....	14
2.1 字符集与标识符 .....	14
2.1.1 字符集 .....	14
2.1.2 标识符 .....	14
2.1.3 关键字 .....	15
2.2 基本数据类型 .....	16
2.2.1 整型 .....	16
2.2.2 字符型 .....	16
2.2.3 布尔型 .....	17
2.2.4 浮点型 .....	17
2.3 常量 .....	18
2.3.1 数值型常量 .....	18
2.3.2 字符型常量 .....	19
2.4 变量 .....	20
2.4.1 变量的定义 .....	20
2.4.2 常变量 .....	21
2.5 运算符与表达式 .....	23
2.5.1 算术运算符与算术表达式 .....	23
2.5.2 赋值运算符与赋值表达式 .....	24



2.5.3	逗号运算符与逗号表达式 .....	24
2.5.4	自增自减运算符 .....	25
2.5.5	类型转换 .....	25
2.5.6	位运算符与位运算表达式 .....	27
2.6	计算机中的数据表示 .....	29
2.6.1	计算机中的数制 .....	29
2.6.2	数值型数据的表示 .....	31
2.6.3	字符型数据的编码表示 .....	33
习题 2	.....	33
<b>第 3 章</b>	<b>语句与结构化程序设计 .....</b>	<b>34</b>
3.1	功能语句与顺序结构程序设计 .....	34
3.1.1	赋值语句 .....	34
3.1.2	空语句 .....	34
3.1.3	复合语句 .....	34
3.1.4	顺序结构程序设计 .....	34
3.2	分支语句与分支结构程序设计 .....	35
3.2.1	关系表达式和逻辑表达式 .....	35
3.2.2	if 语句 .....	38
3.2.3	if...else 语句 .....	38
3.2.4	条件运算符与条件表达式 .....	40
3.2.5	switch 语句 .....	41
3.3	循环语句与循环结构程序设计 .....	42
3.3.1	for 循环语句 .....	43
3.3.2	do...while 循环语句 .....	45
3.3.3	while 循环语句 .....	47
3.3.4	循环语句的嵌套 .....	48
3.4	转移语句 .....	49
3.4.1	break 语句 .....	49
3.4.2	continue 语句 .....	49
3.4.3	goto 语句 .....	50
3.5	程序举例 .....	51
3.6	算法与程序设计 .....	58
3.6.1	算法设计与分析 .....	58
3.6.2	程序设计风格 .....	61
习题 3	.....	62
<b>第 4 章</b>	<b>数组与字符串 .....</b>	<b>63</b>
4.1	数组的定义与使用 .....	63
4.1.1	一维数组的定义与使用 .....	63

4.1.2	二维数组的定义与使用 .....	64
4.2	数组的初始化 .....	66
4.3	数组应用举例 .....	67
4.3.1	一维数组应用 .....	68
4.3.2	二维数组应用 .....	73
4.4	字符串 .....	76
4.4.1	字符串变量的定义与使用 .....	76
4.4.2	字符串变量的输入输出 .....	77
4.4.3	字符串运算 .....	78
4.4.4	字符串数组 .....	79
4.5	字符数组 .....	81
4.6	动态数组 .....	82
4.7	静态数组访问的下标越界问题 .....	84
习题 4	.....	85
<b>第 5 章</b>	<b>函数</b> .....	<b>86</b>
5.1	函数定义 .....	86
5.2	函数调用与函数声明 .....	87
5.2.1	函数调用 .....	87
5.2.2	函数声明 .....	88
5.3	函数的参数传递 .....	90
5.3.1	值传递 .....	90
5.3.2	引用传递 .....	91
5.3.3	函数的默认形参值 .....	92
5.4	函数的嵌套与递归 .....	93
5.4.1	函数嵌套 .....	94
5.4.2	函数递归 .....	94
5.5	数组作函数参数 .....	97
5.5.1	数组元素作函数参数 .....	97
5.5.2	一维数组名作函数参数 .....	98
5.5.3	二维数组名作函数参数 .....	99
5.6	内联函数 .....	100
5.7	重载函数 .....	101
5.8	系统函数 .....	103
习题 5	.....	104
<b>第 6 章</b>	<b>指针</b> .....	<b>106</b>
6.1	指针变量 .....	106
6.1.1	内存单元地址 .....	106

6.1.2	指针变量的定义	107
6.2	指针的赋值与运算	107
6.2.1	指针的赋值	107
6.2.2	指针的运算	108
6.3	指针作为函数参数	110
6.4	指针与数组	111
6.4.1	数组的地址	111
6.4.2	指针与一维数组	111
6.4.3	指针与二维数组	112
6.4.4	指针数组	114
6.5	动态内存分配	116
6.6	函数指针	119
	习题 6	122
<b>第 7 章</b>	<b>自定义数据类型</b>	123
7.1	结构体类型	123
7.1.1	结构体类型的定义	123
7.1.2	结构体变量的定义与初始化	124
7.1.3	结构体变量的引用	125
7.2	结构体应用	127
7.2.1	结构体数组	127
7.2.2	结构体和指针	129
7.2.3	结构体和函数	131
7.3	共用体	134
7.3.1	共用体的定义	135
7.3.2	共用体的应用	136
7.4	链表	137
7.4.1	线性表	137
7.4.2	线性链表	138
7.4.3	线性链表的操作	139
7.5	枚举类型	144
7.5.1	枚举类型和枚举变量的定义	144
7.5.2	枚举变量的赋值和引用	145
7.5.3	枚举变量值的输入与输出	145
	习题 7	147
<b>第 8 章</b>	<b>C++ 程序结构</b>	149
8.1	C++ 程序构成	149
8.2	预处理命令	150

8.2.1	宏定义	150
8.2.2	文件包含	151
8.2.3	条件编译	153
8.3	变量的作用域	156
8.3.1	局部变量	156
8.3.2	全局变量	156
8.4	变量的存储类别	157
8.4.1	局部自动变量	158
8.4.2	局部寄存器变量	159
8.4.3	局部静态变量	159
8.4.4	外部变量声明	160
8.5	内部函数与外部函数	162
8.5.1	内部函数	162
8.5.2	外部函数	163
习题 8		164
<b>第 9 章</b>	<b>类和对象</b>	165
9.1	面向对象程序设计概述	165
9.1.1	面向对象的概念	165
9.1.2	面向对象程序设计的特点	165
9.1.3	面向对象与面向过程的区别	166
9.2	类和对象	167
9.2.1	类的定义	167
9.2.2	对象的定义	170
9.2.3	对象成员的访问	171
9.3	构造函数	173
9.3.1	不带参数的构造函数	173
9.3.2	带参数的构造函数	174
9.3.3	带默认值的构造函数	175
9.3.4	带初始化表的构造函数	176
9.3.5	构造函数的重载	178
9.4	析构函数	180
9.5	复制构造函数	183
9.5.1	对象的赋值	183
9.5.2	对象的复制	184
9.6	对象数组	188
9.7	const 对象和 const 成员	190
9.8	成员函数存储与 this 指针	192
9.9	类静态成员	194

9.9.1	静态数据成员	194
9.9.2	静态成员函数	195
9.10	友元函数与友元类	197
9.10.1	友元函数	199
9.10.2	友元类	201
习题 9		202
<b>第 10 章</b>	<b>继承与派生</b>	<b>203</b>
10.1	继承与派生的概念	203
10.2	派生类的声明	204
10.2.1	派生类的声明	204
10.2.2	派生类的构成	205
10.3	派生类成员的访问特性	207
10.3.1	公有继承	207
10.3.2	私有继承	208
10.3.3	保护继承	208
10.3.4	多级派生时的访问特性	209
10.4	派生类的构造函数和析构函数	212
10.4.1	简单派生类的构造函数	213
10.4.2	有子对象的派生类的构造函数	215
10.4.3	派生类的析构函数	216
10.5	基类与派生类的转换	218
10.5.1	派生类对象的值可以赋给基类对象	218
10.5.2	派生类对象可以作为基类对象使用	219
10.6	类的继承与组合	220
10.7	继承在软件开发中的作用	221
10.8	多基类继承	224
10.8.1	定义多基类继承	224
10.8.2	多继承派生类的构造函数	226
10.8.3	基类成员名冲突	227
10.8.4	虚基类	228
习题 10		232
<b>第 11 章</b>	<b>运算符重载</b>	<b>233</b>
11.1	运算符重载的实现	233
11.2	运算符重载规则	237
11.3	运算符重载函数	238
11.4	重载流插入和流提取运算符	241
11.5	数据的类型转换	244

11.5.1	用转换构造函数实现类型转换	245
11.5.2	用类型转换函数实现类型转换	247
习题 11		250
<b>第 12 章</b>	<b>多态与虚函数</b>	251
12.1	多态性和绑定	251
12.2	虚函数	254
12.2.1	虚函数的定义	254
12.2.2	虚析构造函数	255
12.3	纯虚函数与抽象类	257
12.3.1	纯虚函数	257
12.3.2	抽象类	260
12.4	动态绑定与虚函数表	263
12.4.1	多态类与非多态类	263
12.4.2	动态绑定的实现	265
习题 12		267
<b>第 13 章</b>	<b>模板</b>	268
13.1	模板的引入	268
13.2	函数模板	269
13.2.1	函数模板的定义	269
13.2.2	函数模板的调用	269
13.2.3	函数模板的重载	271
13.3	类模板	274
13.3.1	类模板的定义和使用	274
13.3.2	类模板的继承	275
13.4	类模板应用	275
13.4.1	数组类模板	275
13.4.2	线性链表类模板	278
习题 13		281
<b>第 14 章</b>	<b>输入输出流和文件</b>	282
14.1	输入输出与流	282
14.2	C++ 的流类和流对象	283
14.3	标准输入输出流	283
14.3.1	标准输出流	284
14.3.2	标准输入流	285
14.3.3	流成员函数	287
14.4	设定输入输出格式	290



14.5	文件处理	294
14.5.1	文件的打开和关闭	294
14.5.2	对 ASCII 码文件的读写	296
14.5.3	对二进制文件的读写	298
习题 14		302
<b>第 15 章</b>	<b>异常处理</b>	<b>303</b>
15.1	异常处理的基本思路	303
15.2	C++ 异常处理的实现	306
15.2.1	简单异常处理	306
15.2.2	类类型异常处理	309
习题 15		311
<b>第 16 章</b>	<b>综合实例</b>	<b>312</b>
16.1	用面向过程的方法设计人人对弈程序	313
16.2	用面向对象的方法设计人人对弈程序	329
16.3	扩展游戏程序	343
习题 16		344
<b>第 17 章</b>	<b>集成开发环境</b>	<b>345</b>
17.1	Visual Studio 2010 的安装与启动	345
17.2	Visual Studio 2010 的操作界面	346
17.3	创建和打开项目	349
17.3.1	创建项目	349
17.3.2	在解决方案中新建项目	350
17.3.3	在项目中添加项	351
17.3.4	打开项目	355
17.3.5	编辑源程序	356
17.4	生成和运行项目	356
17.5	调试程序	357
习题 17		364
<b>附录 A</b>	<b>ASCII 码表</b>	<b>365</b>
<b>参考文献</b>		<b>366</b>

# 第 1 章 C++ 入门

学习程序设计,当然要学语法知识,但更重要的是学习程序设计方法,提高程序设计能力,培养程序设计思维。实际动手编写程序与上机调试程序是提高程序设计能力的最主要途径,为尽早对 C++ 程序有一个直观的认识并能尽快动手编写 C++ 程序,本章通过简单的程序示例来说明 C++ 程序的基本结构和基本程序设计方法。后面各章再逐一介绍语法知识,逐步学会编写比较复杂的 C++ 程序。

## 1.1 C++ 简介

### 1.1.1 C++ 的产生与发展

计算机是一种通过程序控制其运行的电子设备,要想让计算机完成某项工作,需要编写相应的程序。在计算机发展的早期,是用机器语言和汇编语言编写程序的,这些低级语言的优点是编写出的程序的执行速度比较快、占用内存空间比较小,比较适合于在早期的内存较小、运算速度比较慢的计算机上编写小程序。其缺点是难以学习和掌握,编写出的程序容易出错,而且难以发现和改正程序中的错误。随着计算机性能的提高和程序规模的不断变大,机器语言和汇编语言越来越不适应解决实际问题的需要。

1957 年,出现了第一个方便用户编写程序的高级语言——FORTRAN 语言,之后 ALGOL、COBOL、BASIC、Pascal 等高级语言相继诞生并得到广泛的应用,这些高级语言比较容易学习和掌握,为编写规模比较大一些的实际问题的应用程序带来了方便。但这些高级语言也有其不足之处,那就是不能充分利用计算机硬件的特性,对于编写数值计算、数据处理等应用程序还可以,不大适合编写利用计算机硬件资源较多的操作系统等系统软件。

1963 年,英国剑桥大学在 ALGOL 60 的基础上增加了硬件处理功能,推出了 CPL (combined programming language)。但 CPL 规模比较大,其编译程序难以有效实现。1967 年,剑桥大学对 CPL 进行了简化,推出了 BCPL (basic CPL)。1970 年,美国贝尔实验室以 BCPL 为基础,又做了进一步简化,设计出更简单且更接近硬件的 B (取 BCPL 的第一个字母)语言,并用 B 语言编写了第一个高级语言版的 UNIX 操作系统,以前的 UNIX 版本都是用汇编语言编写的。也许是精简的太多了,B 语言过于简单,功能有限。1972 年~1973 年间,贝尔实验室在 B 语言的基础上设计出了 C (取 BCPL 的第二个字母)语言。C 语言既保持了 BCPL 和 B 语言精练、接近硬件的优点,又克服了它们过于简单、无数据类型的缺点。使 C 语言既具有汇编语言能够充分利用硬件特性的优点,又有高级语言简单、易学易用的优点。

1973 年,贝尔实验室将原来用汇编语言编写的 UNIX 操作系统用 C 语言改写成 UNIX 第 5 版,C 语言代码占 90% 以上。1975 年,UNIX 第 6 版公布后,C 语言的优点引起人们的广泛关注,随着 UNIX 的日益广泛使用,C 语言的强大功能和优点逐渐为人们认识,C 语言

得以迅速传播,成为应用最为广泛的程序设计语言。作为发明 C 语言和开发 UNIX 操作系统的关键人物,肯尼思·汤普森(K. L. Thompson)和丹尼斯·里奇(D. M. Ritchie)荣获 1983 年度的图灵奖(Turing Award)。图灵奖是美国计算机学会(Association for Computing Machine, ACM)在 1966 年设立的,专门奖励那些在计算机科学领域的学术研究中作出创造性贡献,对推动计算机科学技术发展具有持久作用的杰出科学家,是目前计算机界最崇高的荣誉,有“计算领域的诺贝尔奖”之称。

C 语言的确是一种优秀的程序设计语言,用 C 语言编写的程序解决了大量的实际问题。但随着计算机要解决的问题越来越复杂,需要编写的程序的规模越来越大,C 语言在可靠性、可维护性及代码重用等方面逐渐暴露出了其不足之处。设计开发更适合于编写大规模程序的程序设计语言提上了日程,C++ 就是这样一种优秀的语言。

C++ 最先是由贝尔实验室计算机科学研究中心的 Bjarne Stroustrup(本贾尼·斯特劳斯特卢普)博士在 20 世纪 80 年代初设计并实现的,它是一种以 C 语言为基础的支持数据抽象和面向对象特性的通用程序设计语言。C++ 是 C 语言的扩充,并保持了 C 语言紧凑、灵活、高效和移植性好的优点。借助于 C 语言的广泛普及基础,C++ 是目前应用最广泛的一种面向对象程序设计语言。1995 年美国著名的科技杂志《Byte》将 Bjarne Stroustrup 博士列入“计算机工业 20 位最具影响力的人”。

### 1.1.2 C++ 的特点

相对于之前的程序设计语言,C++ 语言主要有两个方面的特点,一是在兼容 C 语言的基础上增强了 C 语言的功能;二是增加了面向对象机制。

#### 1. 增强了 C 语言的功能

C++ 由 C 语言发展而来,在兼容 C 的基础上对 C 进行了有效的功能扩充,用 C 语言编写的程序可以直接在 C++ 编译环境中使用,原来熟悉 C 语言的编程人员很容易接受 C++, 使得 C++ 借助于 C 的普及基础而得到广泛应用。但 C++ 在输入输出、函数原型声明、函数的重载、函数模板、变量的引用、异常处理等方面进行了扩充,提高了 C++ 程序的安全性、可靠性和可读性,更好地保证了程序的正确性,这对于编写高质量的规模比较大的程序来说是至关重要的。

#### 2. 增加了面向对象机制

C 语言是一种优秀的面向过程的程序设计语言,由于其功能强大、使用方便而得到广泛的应用。但随着程序规模越来越大,C 语言显得有些力不从心,程序易出错、错误不易发现和修改、代码不能重用、编程效率低等不足逐渐显露出来。C++ 克服了 C 的不足,增加了面向对象机制,代码重用性好,适合于开发规模比较大的程序。由于 C++ 对 C 的改进主要是引入了适用于面向对象程序设计的“类”(class),最初的 C++ 称为“带类的 C”(C with classes),1983 年正式定名为 C++ (C plus plus),表示 C++ 是 C 的增强版。

面向过程是一种“就事论事”的解决问题方法,程序中需要什么功能,就编写实现该具体功能的程序代码,这种方法的代码重用性比较差,对于编写解决比较简单问题的小规模程序比较合适,结构化是保证面向过程程序质量的一种规范化的程序设计方法;而面向对象是一种针对某一类问题的“通用”解决方法,有了通用的程序代码,再通过继承、多态等特性实现具体功能,由于具有代码重用(通用的程序代码部分具有很好的重用性)等特性,比较适合大