

轻松学
编程

轻松学

C++编程

胡超 闫玉宝 等编著



DVD-ROM

976分钟多媒体教学视频

- **结构独特:** 每个知识点按照“概念-
->运行结果->示例分析”的模式进行讲解
- **形式新颖:** 用准确的语言总结概念,用直观的图示演示过程,用详细的注释解释代码,用形象的比喻帮助记忆
- **内容丰富:** 重要知识点覆盖全面,实例丰富多彩
- **附赠光盘:** 在随书附赠的光盘中配备了完整的视频教学讲座和本书所使用的全部项目文件与代码
- **技术支持:** 读者可以直接登录www.rzchina.net获取更多的学习资料



化学工业出版社

轻松学
编程

轻松学

C++编程

胡超 闫玉宝 等编著



化学工业出版社

· 北京 ·

C++是目前最流行且被广泛应用的程序设计语言之一，它的高效率和面向对象技术备受推崇。本书由浅入深、循序渐进地讲解了 C++的各个知识点，并结合一些实用的知识讲解了 C++的主要应用。全书内容包括 C++的发展历史和特点，基本元素与结构，运算符、标点符号和表达式，控制语句与控制结构，数组，函数，指针与引用，自定义类型与字符串，面向对象设计思想和类，重载技术，继承与派生，虚函数与多态性，模板与命名空间，标准模板库，输入输出体系，异常处理，API 编程和 MFC 框架，多线程处理，链接库，基本算法，数据结构，数据库编程，网络编程等。为了使读者能够真正掌握 C++的应用，书中最后两章通过建立两个实用的程序来介绍 C++的具体应用。

本书配光盘一张，内容为本书教学视频、源代码、教学 PPT 及习题参考答案。

本书适合 C++初学者、具有一定 C 语言或者 C++语言基础的中级学习者、学习 C++的大中专院校的学生使用，同时也可以作为各高校 C++教师的教学参考书。

图书在版编目 (CIP) 数据

轻松学 C++编程 / 胡超, 闫玉宝等编著. —北京:
化学工业出版社, 2012. 8

(轻松学编程)

ISBN 978-7-122-14619-9

ISBN 978-7-89472-450-2 (光盘)

I. 轻… II. ①胡… ②闫… III. C 语言-程序设计
IV. TP312

中国版本图书馆 CIP 数据核字 (2012) 第 138337 号

责任编辑: 王思慧 李 萃

装帧设计: 王晓宇

出版发行: 化学工业出版社 (北京市东城区青年湖南街 13 号 邮政编码 100011)

印 装: 化学工业出版社印刷厂

787mm×1092mm 1/16 印张 32¹/₂ 字数 820 千字 2012 年 9 月北京第 1 版第 1 次印刷

购书咨询: 010-64518888 (传真: 010-64519686)

售后服务: 010-64518899

网 址: <http://www.cip.com.cn>

凡购买本书, 如有缺损质量问题, 本社销售中心负责调换。

定 价: 68.00 元 (含 1DVD-ROM)

版权所有 违者必究

21 世纪是科学技术迅速发展的时代，信息技术的飞速发展为社会带来了巨大的进步。一直以来，C++语言作为最流行、应用最广泛的语言已经被应用到各个行业的软件开发中，特别是那些对应用程序的效率要求比较高的行业，如通信、控制、嵌入式设计和图形软件设计等，C++技术的应用则更加广泛。

C++语言的前身是 C 语言。C 语言从诞生到现在已经有 30 多年，C++语言（1983 年第一次投入使用）也已有近 30 年的历史。这说明，C/C++语言是经得起时间考验的语言，越来越多的人在不断地学习和研究 C++语言。从笔者的实际教学经验和参加项目开发的过程来看，学习者普遍反映 C++语言十分难以掌握。经过调查发现，C++语言之所以“难学”，主要表现在以下几个方面。

(1) C++语言本身确实比较难以理解，特别是面向对象思想，并不是每个人学了就能理解的。要做到深入理解，必须坚持学习和运用 C++技术，这是大部分人做不到的。

(2) 目前与 C++技术相关的工具过多，而每个工具又具有自己的特点，导致学习者无法分清哪些技术是 C++语言本身的，哪些是和编译器相关的。这样一来，学习者就越来越糊涂，分不清方向。

(3) 想要找到一本适合的 C++书籍来入门也绝非易事。

克服了这些问题，完全可以学好 C++。对于第 (1) 点，读者只要能做到坚持不懈，就可以克服；对于第 (2) 点，读者可以选定一个固定的 C++编译系统去学习，当熟悉了這個编译系统的应用后，即可触类旁通；对于第 (3) 点，只需要一本适合于初学者和中等水平者阅读和参考的书籍即可。

本书就是专为 C++初学者和中等水平者编写的学习和参考工具。本书选定经典而完善的开发工具 Visual C++ 6.0 为编译器，结合 C++基础语言和重要应用，逐一讲解 C++的知识点。相信读者在阅读完本书后，即可独立进行常规的应用开发，并为深入学习更高级的 C++技术作好铺垫。

本书特点

1. 体系完善，涵盖全面

为了使读者在学习完本书后就能进行常规的开发工作，书中不仅讲解了 C++的基础知识，还讲解了算法、数据结构、网络和数据库等方面的知识，最后将这些知识和 C++技术结合，深入分析了 C++在这些方面的应用，为读者进行实际开发提供了良好的知识体系。

2. 由浅入深，循序渐进

为了使读者能容易地理解本书所讲解的知识点，本书没有设置枯燥难懂的语言描述和难以理解的实例。所有的知识点都是先进行通俗的理论讲解后再用实例进行演示。让读者在彻底理



解此知识点的基础上能有更多的思考，而不是一开始就让读者陷入枯燥的语言和例子中。

3. 案例精讲，深入剖析

为了使读者对每个知识点都有较深的理解并熟练地掌握，大部分章节后面都有典型的实例分析。对于这些例子，一方面读者可以亲自动手，另一方面本书会对解决步骤和相关代码进行讲解。通过这两方面的措施，读者可以对实例中包含的知识点有非常直观的认识，进行练就实际动手的能力，从而真正掌握书中的知识。

本书内容

本书共分为四篇 25 章，本书由浅入深、循序渐进地讲解了 C++ 的各个知识点，并结合一些实用的知识讲解了 C++ 的主要应用。全书内容包括 C++ 的发展历史和特点，基本元素与结构，运算符、标点符号和表达式，控制语句与控制结构，数组，函数，指针与引用，自定义类型与字符串，面向对象设计思想和类，重载技术，继承与派生，虚函数与多态性，模板与命名空间，标准模板库，输入输出体系，异常处理，API 编程和 MFC 框架，多线程处理，链接库，基本算法，数据结构，数据库编程，网络编程等。为了使读者能够真正掌握 C++ 的应用，书中最后两章通过建立两个实用的程序来介绍 C++ 的具体应用。

本书配光盘一张，内容为本书教学视频、源代码、教学 PPT 及习题答案。

本书读者

- C++ 初学者。
- 具有一定 C 语言或者 C++ 语言基础的中级学习者。
- 学习 C++ 的大中专院校的学生。
- C++ 教学工作者。
- 社会培训班学员。

本书编者

本书主要由胡超、闫玉宝编写，其他参与编写和资料整理的人员有刘成、马臣云、潘娜、阮履学、陶则熙、王大强、王磊、徐琦、许少峰、颜盟盟、杨娟、杨瑞萍、于海波、俞菲、曾苗苗、赵莹、朱存等。

由于编者水平有限，书中难免出现不足与疏漏之处，敬请读者批评指正。

编者

2012 年 3 月

目 录

第一篇 基础篇

第 1 章 概述	2	2.4.2 预处理命令	23
1.1 计算机程序设计语言概述	2	2.4.3 基本输入输出	24
1.2 了解 C++	3	2.4.4 注释	28
1.2.1 发展历史和特点	3	2.5 本章实例	29
1.2.2 程序的构成	4	2.6 小结	30
1.2.3 C++的用途	4	2.7 习题	30
1.2.4 C++和 Visual C++	4	2.8 实训	31
1.3 开发环境	5	第 3 章 运算符、标点符号和表达式	33
1.3.1 开发系统组成和开发流程	5	3.1 运算符	33
1.3.2 典型 C++开发环境	7	3.1.1 算术运算符	33
1.3.3 安装 Visual C++ 6.0	8	3.1.2 关系运算符	36
1.3.4 第一个 C++程序: Hello World!	11	3.1.3 逻辑运算符	36
1.4 C++学习建议	13	3.1.4 运算符优先级和结合律	37
1.5 小结	14	3.2 标点符号	38
1.6 习题	14	3.3 表达式	39
1.7 实训	14	3.4 本章实例	39
第 2 章 基本元素与结构	16	3.5 小结	41
2.1 数据类型	16	3.6 习题	41
2.1.1 整型	16	3.7 实训	42
2.1.2 字符型	17	第 4 章 控制语句与控制结构	44
2.1.3 布尔型	18	4.1 流程图	44
2.1.4 浮点型	18	4.2 顺序结构	44
2.2 常量和变量	19	4.3 选择结构	45
2.2.1 常量	19	4.3.1 if 语句	45
2.2.2 变量	21	4.3.2 switch 语句	47
2.3 基本元素	21	4.4 循环结构	50
2.3.1 关键字	21	4.4.1 利用 goto 语句和 if 语句构成 循环	50
2.3.2 标识符	22	4.4.2 while 语句	51
2.4 基本结构	23		
2.4.1 main 函数	23		



4.4.3 do-while 语句	52	6.4.2 main()函数的参数	82
4.4.4 for 语句	52	6.4.3 省略符形参	83
4.4.5 break 语句	56	6.5 函数作用域规则	83
4.4.6 continue 语句	57	6.5.1 作用域	83
4.5 本章实例	57	6.5.2 局部变量	84
4.6 小结	60	6.5.3 全局变量	85
4.7 习题	60	6.6 函数的嵌套与递归调用	85
4.8 实训	61	6.7 内联函数	87
第 5 章 数组	63	6.8 函数的模板	88
5.1 一维数组的概念和存储	63	6.9 本章实例	89
5.1.1 一维数组的定义和初始化	63	6.10 小结	91
5.1.2 一维数组的输入和输出	65	6.11 习题	91
5.1.3 一维数组的引用	65	6.12 实训	91
5.1.4 一维数组内存结构和寻址	66	第 7 章 指针与引用	93
5.2 二维数组	67	7.1 指针的概念和基本用法	93
5.2.1 二维数组的定义和初始化	67	7.1.1 指针的概念	93
5.2.2 二维数组的引用	68	7.1.2 指针变量的声明	94
5.2.3 二维数组内存结构和寻址	68	7.1.3 地址运算符	94
5.3 多维数组	69	7.1.4 指针的赋值	95
5.3.1 多维数组的定义和初始化	69	7.1.5 指针运算	96
5.3.2 多维数组的引用	70	7.1.6 const 指针	98
5.3.3 多维数组内存结构和寻址	70	7.1.7 void 指针	99
5.4 字符数组	70	7.1.8 指针的指针	100
5.4.1 字符数组的定义和初始化	70	7.2 指针与数组	101
5.4.2 字符数组的引用	71	7.2.1 指针数组	101
5.4.3 利用字符数组操作字符串	71	7.2.2 数组名以及下标和指针的关系	102
5.5 本章实例	73	7.3 指针与函数	104
5.6 小结	74	7.3.1 指针作为函数参数	104
5.7 习题	74	7.3.2 指针函数	106
5.8 实训	75	7.3.3 指向函数的指针	107
第 6 章 函数	77	7.4 指针和动态内存的分配	109
6.1 函数的概念和定义	77	7.4.1 C++内存分配机制	110
6.2 函数原型	78	7.4.2 C++风格的动态内存分配方法	111
6.3 函数调用	80	7.4.3 C 风格的动态内存分配方法	115
6.4 函数参数	80	7.4.4 动态内存分配陷阱	117
6.4.1 函数参数传递方式	81	7.4.5 动态内存的传递	119
		7.5 引用	121

7.5.1 引用的概念和基本用法	121	8.2.2 共用体的使用	138
7.5.2 引用作为函数参数	123	8.3 枚举类型	140
7.5.3 引用作为返回值	124	8.3.1 枚举类型与枚举变量	140
7.6 本章实例	125	8.3.2 枚举类型的使用	141
7.7 小结	126	8.4 类型定义	142
7.8 习题	127	8.5 字符串	143
7.9 实训	127	8.5.1 C 风格字符串处理	143
第 8 章 自定义类型与字符串	129	8.5.2 用指针操作字符串	146
8.1 结构体	129	8.5.3 C++字符串处理	149
8.1.1 结构体的概念和声明	129	8.5.4 常用字符串的操作函数	151
8.1.2 结构体变量的定义	130	8.6 本章实例	153
8.1.3 结构体变量的初始化	132	8.7 小结	157
8.1.4 结构体的使用	133	8.8 习题	157
8.2 共用体	137	8.9 实训	158
8.2.1 共用体类型及其变量	137		

第二篇 面向对象篇

第 9 章 面向对象设计思想和类	162	9.4.2 组合类的构造函数和析构函数	196
9.1 程序设计思想	162	9.4.3 组合类的初始化	198
9.1.1 结构化程序设计思想	162	9.5 友元函数和友元类	199
9.1.2 面向对象程序设计思想	163	9.5.1 友元的需求性和定义	200
9.2 类	164	9.5.2 友元函数	201
9.2.1 类的定义和组成	165	9.5.3 友元类	204
9.2.2 类成员的访问控制	167	9.6 本章实例	206
9.2.3 类实例和类成员访问	170	9.7 小结	209
9.2.4 类的作用域和对象的生存周期	172	9.8 习题	209
9.2.5 this 指针	173	9.9 实训	210
9.2.6 静态成员	174	第 10 章 重载技术	213
9.2.7 常成员	179	10.1 重载函数	213
9.3 构造函数和析构函数	181	10.1.1 重载函数的概念和定义	213
9.3.1 构造函数	181	10.1.2 重载函数的使用	214
9.3.2 拷贝构造函数	186	10.2 运算符重载	215
9.3.3 默认拷贝构造函数	188	10.2.1 运算符重载的需求	215
9.3.4 浅拷贝和深拷贝	190	10.2.2 运算符重载的基本方法和规则	216
9.3.5 析构函数	193	10.2.3 增减量运算符的重载	220
9.4 类的组合	195	10.2.4 转换运算符的重载	223
9.4.1 类的组合的概念	195		



10.2.5 赋值运算符的重载	225	12.3.2 抽象类和抽象基类	274
10.3 本章实例	226	12.4 本章实例	276
10.4 小结	227	12.5 小结	283
10.5 习题	228	12.6 习题	283
10.6 实训	228	12.7 实训	283
第 11 章 继承与派生	230	第 13 章 模板与命名空间	286
11.1 继承与派生	230	13.1 模板的概念与作用	286
11.1.1 继承与派生的概念	230	13.1.1 模板的概念	286
11.1.2 派生类的声明	231	13.1.2 模板的作用	286
11.1.3 生成派生类的步骤	232	13.2 函数模板	288
11.2 继承中的访问控制	235	13.2.1 函数模板的定义和使用	288
11.2.1 公用继承的访问控制	235	13.2.2 重载模板函数	290
11.2.2 私有继承的访问控制	237	13.3 类模板	291
11.2.3 保护继承的访问控制	238	13.4 命名空间	297
11.3 派生类的构造函数和析构函数	241	13.4.1 程序中的命名冲突分析	297
11.3.1 派生类的构造函数	241	13.4.2 命名空间的概念	299
11.3.2 派生类的析构函数	244	13.4.3 命名空间的使用	300
11.4 基类与派生类的相互作用	246	13.4.4 标准命名空间 std	303
11.4.1 派生类成员的标识和访问	246	13.4.5 C++头文件的使用	304
11.4.2 基类和派生类赋值规则	249	13.5 本章实例	305
11.5 多重继承特性	250	13.6 小结	307
11.5.1 多重继承引起的二义性问题	250	13.7 习题	308
11.5.2 虚拟继承和虚基类	251	13.8 实训	308
11.6 本章实例	254	第 14 章 标准模板库	310
11.7 小结	260	14.1 泛型化编程与 STL 介绍	310
11.8 习题	261	14.1.1 泛型化编程的概念	310
11.9 实训	262	14.1.2 C++标准库与 STL 的内容	310
第 12 章 虚函数与多态性	266	14.2 STL 的使用	312
12.1 多态性	266	14.2.1 容器	312
12.1.1 多态的概念和类型	266	14.2.2 迭代器	314
12.1.2 多态的实现方式	267	14.2.3 算法	316
12.2 虚函数	267	14.3 本章实例	318
12.2.1 虚函数的概念和定义	267	14.4 小结	322
12.2.2 虚函数的使用规则	273	14.5 习题	323
12.3 纯虚函数与抽象类	273	14.6 实训	323
12.3.1 纯虚函数	274		

第三篇 应用篇

第 15 章 输入输出体系	326	16.7 实训	378
15.1 流的概念	326	第 17 章 API 编程和 MFC 框架	380
15.1.1 C 语言中 I/O 的弊端	327	17.1 API 编程介绍	380
15.1.2 流类	328	17.1.1 认识 API 的编程	380
15.2 流对象和格式化输出	329	17.1.2 API 函数的概念和作用	384
15.2.1 预定义的流对象	329	17.1.3 Windows API 分类	385
15.2.2 流格式化输出	330	17.1.4 Windows API 基本术语	385
15.3 重载流运算符	334	17.1.5 Windows API 基本使用	387
15.3.1 流运算符重载概述	334	17.2 MFC 框架简介	389
15.3.2 插入运算符重载	334	17.2.1 认识 MFC 的程序	390
15.3.3 提取运算符重载	335	17.2.2 MFC 的编程框架	390
15.4 文件操作	337	17.2.3 MFC 的模板	392
15.4.1 文件概述	337	17.2.4 MDI 应用程序的构成	393
15.4.2 文件操作基础	338	17.3 小结	394
15.4.3 打开和关闭文件	338	17.4 习题	394
15.4.4 文本文件输入输出	340	17.5 实训	395
15.4.5 二进制文件输入输出	345	第 18 章 多线程处理	397
15.4.6 文件定位	347	18.1 进程和线程	397
15.5 本章实例	348	18.1.1 进程和线程的概念	397
15.6 小结	359	18.1.2 线程的优先级	398
15.7 习题	359	18.1.3 线程运行状态	398
15.8 实训	359	18.2 线程的操作	399
第 16 章 异常处理	362	18.2.1 线程的建立	399
16.1 异常概述	362	18.2.2 线程的优先级设定	403
16.1.1 异常的概念和分类	362	18.2.3 线程的挂起、恢复	404
16.1.2 异常现象的举例	363	18.2.4 线程的结束	405
16.2 异常处理机制	364	18.3 本章实例	407
16.2.1 异常处理的任务	364	18.4 小结	409
16.2.2 异常处理基本思想	364	18.5 习题	409
16.2.3 异常处理的组成	365	18.6 实训	409
16.2.4 异常处理的执行过程	367	第 19 章 链接库	411
16.3 析构函数与异常处理	370	19.1 链接库概述	411
16.4 本章实例	374	19.2 静态链接库	412
16.5 小结	376	19.2.1 静态链接库的编写	412
16.6 习题	377		



19.2.2 静态链接库的使用.....	414	19.4.1 隐式链接.....	420
19.3 动态链接库.....	415	19.4.2 显式链接.....	421
19.3.1 DLL 文件的编写.....	415	19.5 小结.....	423
19.3.2 导出 DLL 资源.....	417	19.6 习题.....	423
19.4 动态链接库的使用.....	420	19.7 实训.....	423

第四篇 开发篇

第 20 章 基本算法.....	426	21.11 实训.....	448
20.1 算法的概念和特征.....	426	第 22 章 数据库编程.....	451
20.1.1 算法的概念.....	426	22.1 概述.....	451
20.1.2 算法的特征.....	426	22.1.1 数据库和数据库系统.....	451
20.1.3 算法的设计要求和描述方法.....	427	22.1.2 常见数据库访问技术.....	452
20.1.4 算法的衡量.....	428	22.2 数据访问接口 ODBC.....	453
20.2 常用算法.....	429	22.2.1 ODBC 概述.....	453
20.2.1 递推法.....	430	22.2.2 ODBC API 编程.....	454
20.2.2 递归法.....	430	22.3 MFC 的 ODBC 编程.....	458
20.2.3 回溯法.....	431	22.3.1 CDatabase 和 CRecordset 类	
20.2.4 贪婪法.....	432	介绍.....	458
20.2.5 分治法.....	433	22.3.2 数据查询.....	461
20.2.6 动态规划法.....	434	22.3.3 参数化数据查询.....	462
20.2.7 迭代法.....	435	22.3.4 新建数据.....	462
20.2.8 穷举法.....	435	22.3.5 数据删除.....	462
20.3 小结.....	436	22.3.6 数据更新.....	463
20.4 习题.....	436	22.3.7 直接执行 SQL 语句.....	463
20.5 实训.....	437	22.4 小结.....	464
第 21 章 数据结构.....	439	22.5 习题.....	464
21.1 线性表及其应用.....	439	第 23 章 网络编程.....	465
21.2 链表及其应用.....	440	23.1 网络通信.....	465
21.3 堆栈及其应用.....	442	23.1.1 网络类型.....	465
21.4 队列及其应用.....	443	23.1.2 网络拓扑结构.....	466
21.5 特殊矩阵、广义表及其应用.....	444	23.2 网络通信协议.....	467
21.6 二叉树及其应用.....	445	23.2.1 TCP/IP 结构.....	467
21.7 散列结构及其应用.....	446	23.2.2 UDP 协议.....	468
21.8 图及其应用.....	447	23.2.3 相关术语.....	468
21.9 小结.....	448	23.3 SOCKET 网络编程接口.....	469
21.10 习题.....	448	23.3.1 Socket 模型.....	469

23.3.2	Socket 的 C/S 编程方式	469	24.6	小结	486
23.3.3	C++下的 Socket	470	24.7	习题	487
23.3.4	利用 Socket 建立服务器端程序	472	第 25 章	火车信息查询系统	488
23.3.5	利用 Socket 建立客户端程序	474	25.1	程序需求	488
23.4	小结	477	25.2	程序界面	488
23.5	习题	477	25.3	服务端程序	489
第 24 章	利用 ADO 实现简单的学生信息管理系统	478	25.3.1	服务器通信类	489
24.1	ADO 简介	478	25.3.2	建立接收线程	490
24.1.1	ADO 概述	478	25.3.3	火车信息处理	493
24.1.2	MFC 的 ADO 编程	479	25.3.4	发送和接收	494
24.1.3	ADO 操作数据库的基本步骤	480	25.4	客户端程序	495
24.2	程序的需求	480	25.4.1	连接服务器并查询	496
24.3	程序的界面设计	481	25.4.2	发送车次信息	497
24.4	程序的数据库	481	25.4.3	接收服务器返回信息	498
24.5	核心程序实现分析	482	25.5	小结	498
			附录	C++面试题目录	500

第一篇 基础篇

第 1 章 概述

第 2 章 基本元素与结构

第 3 章 运算符、标点符号和表达式

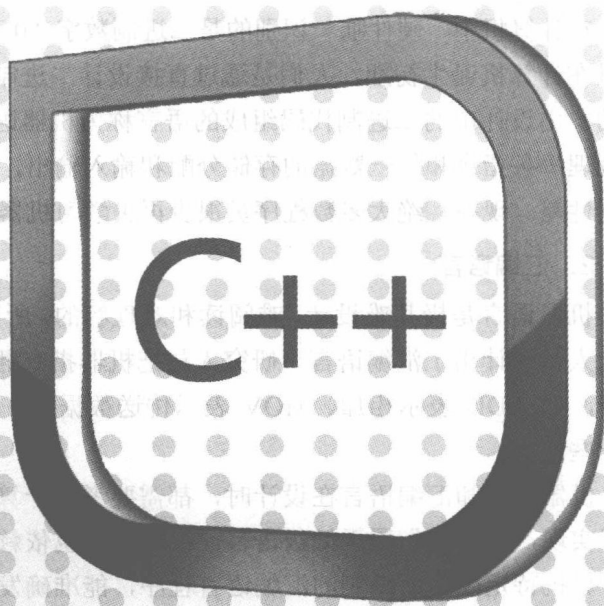
第 4 章 控制语句与控制结构

第 5 章 数组

第 6 章 函数

第 7 章 指针与引用

第 8 章 自定义类型与字符串



第 1 章 概 述

C++是一种优秀的面向对象程序设计语言。它从C语言的基础上发展而来，但又比C语言更容易学习和掌握。C++以其独特的语言机制在计算机科学的各个领域中得到了广泛的应用。C++完美地体现了面向对象的各种特性。通过本章的学习，读者可以了解C++语言的产生及特点，能够基本掌握安装和配置 Visual C++6.0 开发环境的方法，并理解C++程序的开发过程。

1.1 计算机程序设计语言概述

计算机是20世纪40年代诞生的，目前已得到了迅猛的发展，并深入到了各个行业中。随着计算机的产生和发展，程序设计也诞生并发展。软件的作用是控制和指导如何利用硬件系统来处理信息，而程序设计的目的就是开发出这些软件。计算机程序设计语言首先通过特定的描述方法来描述世界，再将这些描述传递给计算机，以达到计算机识别世界的目的。计算机程序设计语言的发展，经历了从机器语言、汇编语言到高级语言的历程，下面分别进行介绍。

1. 机器语言

在计算机中，硬件唯一识别的是二进制数字（0和1）。计算机的指令都是一串二进制代码，所以在计算机诞生初期，人们是通过直接设计二进制代码来设计程序的。这种设计方法是最原始的，由设计出的二进制代码组成的语言称为机器语言。用机器语言编写程序，程序员需要自己处理每条指令和每一数据的存储分配和输入输出，而且编出的程序全是0和1的指令代码，直观性差。现在，绝大多数程序员很少再去学习机器语言了。

2. 汇编语言

机器语言是极其难设计、难阅读和难理解的，所以学习和记忆起来就更加困难。随后不久，设计人员设计出了汇编语言。研究人员把机器指令翻译成为了一些容易被人们阅读和记忆的助记符，如ADD表示相加、MOV表示传送数据等，这使得程序设计比使用机器语言简单、易懂一些。

机器语言和汇编语言在设计时，都需要考虑计算机底层的问题，在实现上更接近于计算机底层实现，所以它们属于低级语言。汇编语言也依赖于机器硬件，移植性不好，但效率高。针对计算机特定硬件而编制的汇编语言程序，能准确发挥计算机硬件的功能和特长，所以至今仍是一种常用而强有力的软件开发工具。

3. 高级语言

汇编语言虽然较机器语言来说把程序设计变得容易了一些,但是它与自然语言(人们日常表达思维的语言)之间的差距还是很大。设计人员在研究了人类的思维、表达方式后,通过的不断发展和创新,发明了高级语言。高级语言的出现是程序设计语言发展中的一个巨大进步。它采用了具有一定含义的数据命名方法和容易理解的执行语句,提高了语言的抽象层次,使得程序设计语言可以更接近人们的自然语言,所以开发者在设计程序时可以更多地联系到所描述的事物。

高级语言的种类有很多,比较流行的有 BASIC、PASCAL、FORTRAN、C 等。C++也属于高级语言,但它相对于前几种语言来说有着更为高级的特性。

1.2 了解 C++

C++经常被人们称为“带类的C”。它是在C语言的基础上引入了面向对象的机制而形成的一种程序设计语言,而C语言是面向过程的程序设计语言。C++几乎继承了C语言的所有特点,同时添加了面向对象的特征。C++既支持面向过程的程序设计,又支持面向对象的程序设计。

1.2.1 发展历史和特点

编程语言的发展是一个循序渐进的过程。C++是从C语言发展而来的。C语言是1972年由美国贝尔实验室的 Denis Ritchie 改造了B语言而来的。C语言是一门优秀的编程语言,但是它也存在一些缺陷。例如,类型检查机制相对较弱、缺少支持代码重用的语言结构等。用C语言开发大程序比较困难,从而限制了C语言的发展。

针对C语言的缺点,贝尔实验室的 Bjarne Stroustrup 博士及其同事开始对C语言进行改进和扩充。为C语言引入了“类”的概念,并在1983年命名为C++,这就是最早的C++语言。后来,C++又被引入了运算符重载、引用、虚函数等许多特性。1998年,美国国家标准化协会 ANSI (American National Standard Institute) 和国际标准化组织 ISO (International Standards Organization) 共同完成了对C++的标准化工作,并正式发布了C++语言的国际标准 ISO/IEC:98-14882。

C++语言是一种规范,它规定了C++语言所要遵循的规则和需要实现的基本功能。其具体的功能由软件开发商去实现,所以各软件开发商推出的C++编译器都要支持该标准。当然,不同的编译器对其都有不同程度的拓展。

在C++发展的后期,对范型程序设计的支持又被作为一个目标,并很快得到了实现和发展,这使得C++在代码重用性方面又获得了质的飞跃。

C++支持面向对象的程序设计方法,所以它特别适合于大中型项目的软件开发。不管是从开发时间、费用还是从软件的重用性、可扩充性、可维护性和可靠性等方面,C++相对于其他语言



均具有很大的优越性。同时，C++又保留了对 C 的支持，这就使得许多 C 代码不用修改或者稍加修改就可以在 C++下编译执行。

1.2.2 程序的构成

C++程序由类 (class) 和函数 (function) 组成。开发人员可以用多个小的软件模块构成 C++ 程序。C++编译系统除了支持语言本身外，还提供 C++标准中规定的标准库函数的实现，所以 C++程序员可以利用 C++标准库中已有的类和函数进行编程。

标准 C++库中的主要组件包括标准 C 库、I/O 流技术、String、容器、算法、对国际化的支持、对数字处理的支持和诊断支持。

在学习 C++的过程中，实际要学习两方面的知识：第一是学习 C++语言本身的语法和规则；第二是学习如何利用 C++标准库中现有的类和函数。学习标准类和函数的使用，不但可以帮助程序员深入理解 ANSI C++语言库函数的使用和实现，还可以帮助程序员了解如何用库函数编写可移植代码。标准库函数通常由编译器厂家提供。许多独立软件供应商也提供各种专用类库。

1.2.3 C++的用途

C++有什么作用？Bjarne Stroustrup 清晰地回答了这个问题：在低级系统程序设计、高级系统程序设计、嵌入式程序设计、数值科学计算、通用程序设计以及混合系统设计等领域，C++有着根本性的优势。让我们略微展开描述一下。

(1) 低级系统程序设计：C++是迄今为止最好的低级程序设计语言。

(2) 高级系统程序设计：包括操作系统核心、网络管理系统、编译系统、电子邮件系统、文字排版系统、图像和声音的编排系统、通信系统、用户界面、数据库系统等。

(3) 嵌入式系统：包括照相机、汽车、火箭、电话交换机、汽车等。

(4) 数值科学计算：包括仿真、实时数据获取和数据库访问等。

下面给出几个 C++的具体应用。

- Microsoft: Windows 操作系统、Microsoft Office (Word、Excel、Access、PowerPoint、Outlook)、Internet Explore (包括 Outlook Express)、Visual Studio (Visual C++、Visual Basic、Visual FoxPro)、SQL Server、FrontPage。
- Amazon.com: 大型电子商务软件。
- Google、百度: 搜索引擎。
- 几乎所有的网络游戏。

1.2.4 C++ 和 Visual C++

C++以及 Visual C++是很容易混淆的一对概念。C++是一种语言，是一种程序设计标准规范；而 Visual C++是微软公司提供的开发环境，它使用 C++语言的规范，并给程序员提供了各种各样的在 Windows 环境下进行程序设计的类库和大量的库函数。Visual C++可以看作是使用

C++语言的一个开发工具，同样我们可以使用别的 C++开发工具（如 C++ Builder）来完成相同的工作。

Windows API (Application Programming Interface, 应用程序编程接口) 是微软提供的 Windows 下应用程序的编程语言接口，是一套用来控制 Windows 各个部件的外观和行为的一套预先定义的 Windows 函数。用户的每个动作都会引发一个或几个函数的运行以告诉 Windows 发生了什么。

MFC (Microsoft Foundation Classes) 是微软公司提供的类库 (class libraries)，以 C++类的形式封装了 Windows 的 API，并且包含一个应用程序框架，以减少应用程序开发人员的工作量。MFC 可以看作是挂在 Visual C++ 之上的一个辅助软件开发包。

1.3 开发环境

进行软件开发，需要一些特定的开发环境。一般 C++开发系统由程序开发环境、语言和 C++标准库这几个部分组成。在本书中，将使用 Visual C++ 6.0 作为开发环境来讲解 C++语言。

1.3.1 开发系统组成和开发流程

C++程序的开发通常可以分为编辑 (Edit)、预处理 (Preprocess) 和编译 (Compile)、连接 (Link)、加载 (Load) 和执行 (Excute)。C++程序开发流程如图 1-1 所示。

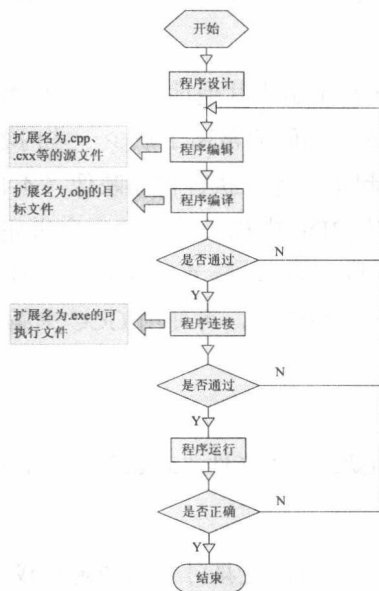


图 1-1 C++程序开发流程

1. 第一阶段——编辑

第一个阶段是编辑，这是用编辑器程序 (Editor Program) 完成的。开发者用编辑器输入 C++程序，并进行必要的修改，然后将程序存放在磁盘之类的辅助存储设备中。C++程序文件名通常以 .cpp、.cxx 或 .C 的扩展名结尾 (注意 C 为大写，C++常见文件扩展名解释如表 1-1 所示)。在