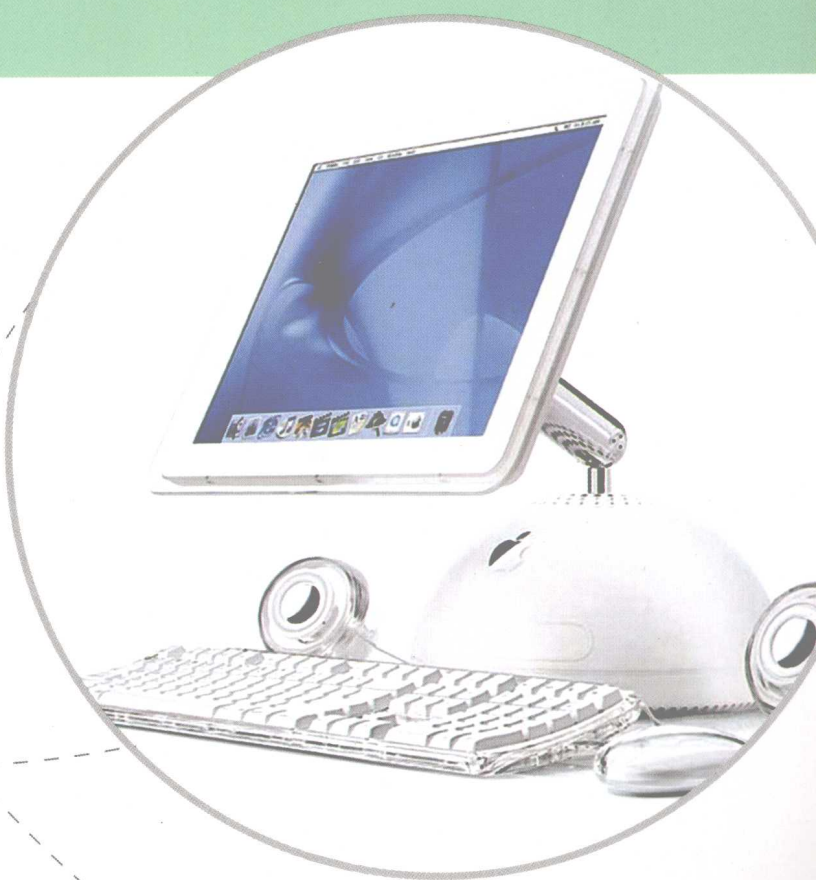




21世纪全国高职高专计算机教育“十一五”规划教材
丛书主编 全国高等学校计算机教育研究会课程与教材建设委员会主任 李大友

C++程序设计

主 编 余建军 张铁头
副主编 贺中立 苏宝程
谭营军 马文龙



中国计划出版社

21世纪全国高职高专计算机教育“十一五”规划教材

C++程序设计

本书编委会 编著

中国计划出版社

图书在版编目 (C I P) 数据

C++程序设计 / 《C++程序设计》编委会编著. —北京: 中国计划出版社, 2007. 8

21世纪全国高职高专计算机教育“十一五”规划教材
ISBN 978-7-80177-959-5

I. C… II. C… III. C语言—程序设计—高等学校: 技术学校—教材 IV. TP312

中国版本图书馆CIP数据核字 (2007) 第099282号

内 容 简 介

本书是全国高职高专计算机教育“十一五”规划教材之一, 根据高职高专院校的教学特点编写而成, 语言通俗易懂, 由浅入深地引领初学者顺利掌握C++语言。

本书共分11章, 内容包括Visual C++集成开发环境、数据类型与表达式、程序结构与流程控制语句、函数、数组与指针、类、继承与派生、多态性、重载以及输入/输出流。书中每一章都有大量的程序实例, 有利于初学者更好地掌握每章的理论知识。

本书既适合作为高职高专院校C++程序设计课程的教材, 也可以作为C++程序设计初学者的自学教材。

21世纪全国高职高专计算机教育“十一五”规划教材 C++程序设计

本书编委会 编著

☆

中国计划出版社出版

(地址: 北京市西城区木樨地北里甲11号国宏大厦C座4层)

(邮政编码: 100038 电话: 63906433 63906381)

新华书店北京发行所发行

河北省高碑店市鑫宏源印刷厂印刷

787×1092毫米 1/16 20.25印张 492千字

2007年8月第一版 2007年8月第一次印刷

印数1—4000册

☆

ISBN 978-7-80177-959-5

定价: 29.00元

丛 书 序

编写背景和目的

高等职业教育是现代国民教育体系的重要组成部分，在实施科教兴国战略和人才强国战略中具有特殊的重要地位。现在，我国就业和经济发展正面临着两个大的变化，即：社会劳动力就业需要加强技能培训，产业结构优化升级需要培养更多的高级技术人才。温家宝总理在 2005 年 11 月 7 日的全国职业教育工作会议上指出，高等职业教育的发展仍然是薄弱环节，不适应经济社会发展的需要；大力发展高等职业教育，既是当务之急，又是长远大计。《国家教育事业发展规划“十一五”规划纲要》中提出，要以培养高素质劳动者和技能型人才为重点，提高学生创新精神和实践能力，大力发展职业教育；扩大高等职业教育招生规模，到 2010 年，使高等职业教育招生规模占高等教育招生规模的一半以上。在以上背景下，我国已进入了新一轮高等职业教育改革的高潮，目前高职院校的学校规模、专业设置、办学条件和招生数量，都超过了历史上任何一个时期。

随着信息社会的到来，灵活应用计算机知识、解决各自领域的实际问题成了当代人必须掌握的技能，为此，高职院校面向不同专业的学生开设了相关的计算机课程。然而，作为高职院校改革核心之一的教材建设大大滞后于高等职业教育发展和社会需求的步伐，尤其是多数计算机应用教材，或显得陈旧，或显得过于偏重理论而忽视应用。以致于一些通过 3 年学习的高职院校学生毕业后，所掌握的技能不能胜任用人单位的需求。

鉴于此，中国计划出版社与全国高等学校计算机教育研究会课程与教材建设委员会联合在全国 1105 所高职高专中做了广泛的市场调查，并成立了《21 世纪全国高职高专计算机教育“十一五”规划教材》编委会，由全国高等学校计算机教育研究会课程与教材建设委员会主任委员、北京工业大学李大友教授担任编委会主任。编委会进行了大量调查研究，通过借鉴国内外最新的、适用于高职高专教学的计算机技术经验成果，推出了切合当前高职教育改革需要、面向就业的系列职业技术型计算机教材。

系列教材

本计算机系列教材主要涵盖了当前较为热门的以下就业领域：

- 计算机基础及其应用
- 计算机网络技术
- 计算机图形图像处理和多媒体
- 计算机程序设计
- 计算机数据库

- 电子商务
- 计算机硬件技术
- 计算机辅助设计

教材特点

本套教材的目标是全面提高学生的计算机技术实践能力和职业技术素质,为此,中国计划出版社与全国高等学校计算机教育研究会课程与教材建设委员会合作,邀请了来自全国各类高等职业学校的骨干教师(其中很多为主管教学的院长或系主任)作为编委会成员外,还特聘了多位具有丰富实践经验的一线计算机各应用领域工程师参加教材的技术指导和编审工作,以期达到教学理论和实际应用紧密结合的效果。

同时,为配合各学校的精品课程建设工程,本套教材以国家级精品课程指标为指引方向,借鉴其他兄弟出版社的先进经验和成功案例,提出了建设“立体化教学资源平台”的概念,其内容包括教材、教学辅导资料、教学资源包、网络平台等内容,并将在后续培训、论文发表等多方面满足教师与精品课程建设的需求。

本系列教材的特点如下:

(1) 面向就业。本系列教材的编写完全从满足社会对技术人才需求和适应高等职业教育改革的角度出发,教材所涉及的内容是目前高职院校学生最迫切需要掌握的基本就业技能。

(2) 强调实践。高职高专自身教育的特点是强调实践能力,计算机技术本身也是实践性很强的学科,本系列教材紧扣提高学生实践能力这一目标,在讲解基本知识的同时配套了大量相关的上机指导、实训案例和习题。

(3) 资源丰富。本系列教材注重教材的拓展配套,辅助教学资源丰富。除了由本书作为主干教材外,还配有电子课件、实训光盘、习题集和资源网站等辅助教学资源。

读者定位

本计算机应用系列教材完全针对职业教育,主要面向全国的高职高专院校。本系列教材还可作为同等学历的职业教育和继续教育的教学用书或自学参考书。

本系列教材的出版是高职教育在新形势下发展的产物。我们相信,通过精心的组织和编写,这套教材将不仅能得到广大高职院校师生的认可,还会成为一套具有时代鲜明特色、易教易学的高质量计算机系列教材。我们与时俱进,紧密配合高职院校的办学机制和运行体制改革,在后期的组织推广及未来的修订出版中不断汲取最新的教学改革经验和教师学生及用人单位的反馈意见,为国家高等职业教育奉献我们的力量。

丛书编委会

前 言

伴随着计算机技术的飞速发展,计算机应用的范围越来越广,计算机普及的程度越来越深入。传统的面向过程程序设计已不能满足软件系统复杂性的要求。巨大的需求引发了软件技术的革命,诞生了面向对象程序设计的思想和方法。C++是当今较为流行的一种面向对象程序设计语言,它完全兼容C语言,具有高效、灵活、功能强大以及可重用性好等特点。

本书的特点是入门起点较低,读者不需要具备C语言的基础,在学习C++语言前未学过其他语言的人也可以直接学习。本书便于教学计划的编排,并附有大量的典型程序实例,从基础知识到最新的高级特性,较全面地讲解了C++语言。

本书根据高职高专院校的教学特点,力求语言通俗易懂、概念明确,力争将复杂的概念实例化,将抽象的理论形象化,使读者逐步掌握C++语言的完整体系。本书的实用性和可操作性较强,提供了大量的编程实例,能够帮助读者透彻理解所学的概念,书中大量的练习题可以使读者进一步巩固所学的内容,确保读者能够真正掌握各章节的知识。为了使读者更好地了解本书,现将书中的章节安排介绍如下:

全书共分11章,第1章和第2章分别介绍了C++语言的基础知识和Visual C++ 6.0开发环境的使用;第3章介绍了数据类型与表达式;第4章介绍了C++语言的程序结构与流程控制语句;第5章介绍了C++语言函数的特点及应用;第6章介绍了数组与指针的概念及应用;第7章重点介绍了C++类的概念及应用;第8章和第9章详细介绍了类成员的继承、派生与多态性;第10章详细介绍了函数的重载与运算符的重载,重载的应用使C++具有更好的扩充性和适应性;第11章介绍了输入/输出流的概念及用法。

本书由余建军、张铁头主编,贺中立、苏宝程、谭营军、马文龙担任副主编,魏云华、方园、陈佳泉、洪新华、王伟、王彩玲、刘娟、江微参与编写。

此外,本书在编写过程中参考了国内外各层次优秀的C++程序设计方面的资料,在此谨向相关作者致谢。

由于时间仓促与编者水平有限,不足与欠妥之处在所难免,恳请广大读者不吝指正。

编者

2007年6月

目 录

第1章 绪论	1
1.1 程序语言的发展	1
1.1.1 机器语言	1
1.1.2 汇编语言	2
1.1.3 高级语言	2
1.1.4 C语言	3
1.1.5 C++语言	4
1.2 计算机信息的表示	4
1.2.1 计算机数字系统	4
1.2.2 进位计数制	5
1.2.3 二进制编码	9
1.2.4 计算机中字符的表示	13
1.3 面向过程程序设计方法概述	13
1.4 面向对象程序设计方法概述	15
1.4.1 面向对象程序设计的概念	15
1.4.2 面向对象程序设计的特点	16
1.4.3 类和对象的作用	16
1.4.4 面向对象的软件开发	19
1.5 本章小结	19
1.6 练习题	20
第2章 Visual C++集成开发环境	22
2.1 Visual C++界面组成	22
2.1.1 Visual C++用户界面	23
2.1.2 菜单栏	23
2.1.3 工具栏	34
2.2 编辑、编译和运行C++程序	35
2.2.1 简单的C++程序	36
2.2.2 项目包含头文件的C++程序	38
2.2.3 多项目在同一工作区	40
2.3 Visual C++下的程序调试	41
2.3.1 发现并处理错误	42
2.3.2 调试窗口	43

2.3.3 调试程序	44
2.4 本章小结	47
2.5 练习题	47
第3章 数据类型与表达式	49
3.1 C++的基本数据类型	49
3.1.1 数据类型概述	49
3.1.2 字符集	50
3.1.3 基本数据类型	50
3.2 常量	51
3.2.1 数值常量	51
3.2.2 符号常量	54
3.2.3 字符串常量	54
3.3 变量	55
3.3.1 标识符与关键字	55
3.3.2 变量的定义与初始化	56
3.4 运算符的使用	57
3.4.1 运算符概述	57
3.4.2 运算符的优先级	58
3.5 表达式的使用	59
3.5.1 算术运算符与算术表达式	59
3.5.2 赋值运算符与赋值表达式	60
3.5.3 逗号运算符与逗号表达式	60
3.5.4 关系运算符与关系表达式	61
3.5.5 逻辑运算符与逻辑表达式	62
3.5.6 位运算	63
3.6 类型转换	64
3.6.1 自动转换	64
3.6.2 强制类型转换	65
3.7 自定义数据类型	66
3.7.1 结构体	66
3.7.2 共用体	70
3.7.3 枚举类型	73
3.7.4 类型声明	75
3.8 本章小结	75
3.9 练习题	76
第4章 程序结构与流程控制语句	79
4.1 C++程序的结构与语句	79
4.1.1 C++程序的结构与语句概述	79

4.1.2 C++的输入与输出	82
4.2 判断与循环	85
4.3 if...else...if判断式	87
4.3.1 基本的if语句	87
4.3.2 完整的if语句	89
4.3.3 if语句的嵌套	91
4.4 switch...case判断式	92
4.5 for循环	94
4.5.1 for语句	94
4.5.2 for语句嵌套	97
4.6 while循环	98
4.6.1 while语句	98
4.6.2 do...while语句	100
4.7 break语句、continue语句、goto语句	103
4.7.1 break语句	103
4.7.2 continue语句	103
4.7.3 goto语句	104
4.8 本章小结	106
4.9 练习题	106
第5章 函数	110
5.1 函数的定义与调用	110
5.1.1 函数概述	110
5.1.2 函数的定义	111
5.1.3 函数的调用	114
5.1.4 实参与形参的数据传送	116
5.2 函数的嵌套调用与递归调用	123
5.2.1 函数的嵌套调用	123
5.2.2 函数的递归调用	124
5.3 数组作函数参数	129
5.3.1 数组元素作函数实参	129
5.3.2 数组名作函数参数	129
5.4 变量的作用域与存储类型	130
5.4.1 变量的作用域	130
5.4.2 变量的存储类型	134
5.5 内联函数	138
5.6 具有默认参数值的函数	139
5.7 预处理命令	140
5.8 本章小结	143

5.9 练习题	144
第6章 数组与指针	147
6.1 数组	147
6.1.1 一维数组	148
6.1.2 多维数组	150
6.1.3 数组与函数	153
6.1.4 字符数组与字符串	155
6.2 指针	165
6.2.1 指针的概念	165
6.2.2 指针与变量、字符串	166
6.2.3 指针与数组	167
6.2.4 多重指针	174
6.2.5 指针与函数	176
6.2.6 指针与动态内存分配	177
6.3 本章小结	179
6.4 练习题	179
第7章 类	183
7.1 类与对象	183
7.1.1 类的声明	184
7.1.2 对象的定义及使用	187
7.2 构造函数	190
7.2.1 构造函数的定义	190
7.2.2 用构造函数初始化对象的过程	190
7.2.3 默认的构造函数	196
7.3 析构函数	198
7.3.1 析构函数的定义	198
7.3.2 析构函数的调用	199
7.3.3 默认的析构函数	201
7.4 类的组合	201
7.5 静态成员	204
7.5.1 静态数据成员与静态成员函数	204
7.5.2 静态数据成员的初始化	207
7.6 友元	209
7.6.1 友元的说明与使用	209
7.6.2 使用友元的条件	216
7.7 共享数据的保护	217
7.8 本章小结	218
7.9 练习题	219

第8章 继承与派生	221
8.1 继承与派生的概念	221
8.1.1 派生类的构成	222
8.1.2 派生类的声明方式	223
8.2 派生类成员的访问属性	226
8.2.1 公用继承与私有继承	227
8.2.2 保护成员与保护继承	232
8.2.3 多级派生时的访问属性	234
8.3 派生类的构造函数与析构函数	235
8.4 多重继承	239
8.4.1 声明多重继承的方法	239
8.4.2 多重继承派生类的构造函数与析构函数	239
8.4.3 虚基类	242
8.5 本章小结	245
8.6 练习题	246
第9章 多态性	247
9.1 多态性概述	247
9.2 虚函数	249
9.2.1 虚函数的定义与使用	251
9.2.2 虚析构函数	258
9.3 抽象类	260
9.3.1 纯虚函数	260
9.3.2 抽象类与具体类	263
9.4 本章小结	268
9.5 练习题	268
第10章 重载	270
10.1 函数重载	270
10.1.1 构造函数重载	272
10.1.2 类成员函数重载	273
10.2 运算符重载的概念	276
10.3 成员函数重载运算符	279
10.4 友元函数重载运算符	282
10.5 常用的运算符重载	285
10.5.1 赋值运算符	285
10.5.2 下标运算符	288
10.5.3 函数调用运算符	290
10.6 本章小结	291
10.7 练习题	291

第11章 输入/输出流	292
11.1 C++输入/输出流的概念	292
11.2 标准输入与输出	294
11.2.1 输入流	294
11.2.2 输出流	295
11.3 流的格式化控制与流操作符	296
11.4 文件	301
11.4.1 文件的概念	301
11.4.2 文件流类与文件流对象	302
11.4.3 文件的打开与关闭	302
11.4.4 对ASCII文件的操作	304
11.4.5 对二进制文件的操作	305
11.5 本章小结	308
11.6 练习题	308
主要参考文献	310

第 1 章

绪论

C++是一门优秀的程序设计语言。C++比C语言更容易学习和掌握，并且以其独特的语言机制在计算机科学领域中得到广泛的应用。本章首先从发展的角度概要介绍了程序语言的发展过程，然后介绍信息在计算机中的表示方法，重点介绍了结构化程序设计方法和面向对象的程序设计方法的基本概念、基本特征和特点，最后概要介绍了面向对象的软件开发步骤。



本章主要内容

- ❑ 计算机语言及其发展，高级语言的优点
- ❑ C++语言的特点
- ❑ 计算机信息表示方法
- ❑ 面向过程程序设计的基本概念和设计方法
- ❑ 面向对象程序设计思想、基本概念和3个主要特征
- ❑ 面向对象软件开发

1.1 程序语言的发展

计算机是无知觉、无生命的机器，要使它能够按照人们的意图工作，就必须使计算机懂得人的意图，接受人向它发出的命令和信息，这就要解决一个“语言”的问题。因为计算机不懂人类的语言，无论是汉语还是法语、英语、日语等，计算机都不能接受，因此要求人们用特定的、计算机可以理解的语言与计算机交流，这就是计算机语言。这种语言同英语、汉语等人类长期使用的自然语言有很大的不同，它有严格的定义，并且一般没有二义性。人们在使用自然语言交谈时，很容易根据语境确定具有二义性的对话的含义，但是计算机却做不到这一点，至少现在它还没有这么聪明。

1.1.1 机器语言

机器语言是机器指令的集合。每种计算机都有自己的指令集合，计算机能直接执行用机器语言所编写的程序。机器语言包括指令系统、数的形式、通道指令、中断字、屏蔽字、

控制寄存器的信息等。机器码是计算机能理解和执行的唯一语言。机器种类不同，其机器码的组合方式也不一样。同一个题目到不同的计算机上计算时，必须编写不同机器语言的程序。机器语言是最低级的语言。由于机器码指令是用许多二进制数表示的，用机器语言编程必然很烦琐，非常消耗精力和时间，难记忆，易弄错，并且难以检查程序和调试程序，工作效率低。但由于使用的是针对特定型号计算机的语言，故而运算效率是所有语言中最高的。机器语言是第一代计算机语言。

1.1.2 汇编语言

早期的计算机由于速度慢、内存小，衡量程序质量高低最重要的指标是机器执行的效率。但是，随着计算机技术的发展，机器硬件性能的大幅度提高，程序的复杂度也在增加，人们越来越要求把简单、重复性的工作交给机器去做，而人们更多地从事创造性的工作。程序的可读性和可维护性渐渐成为衡量程序质量高低的最重要的指标。为了减轻使用机器语言编程的痛苦，人们进行了一种有益的改进：用一些简洁的英文字母、符号串来替代一个特定指令的二进制串，例如，用“ADD”代表加法，“MOV”代表数据传递等。这样，人们就能很容易读懂并理解程序在干什么，纠错及维护都变得方便了，这种程序设计语言就称为汇编语言，即第二代计算机语言。汇编语言同机器语言相比，并没有本质的区别，只不过是把机器指令用助记符号代替。但这已是很大的进步，它提高了编程的效率，改进了程序的可读性和可维护性。直到今天，仍然有人在用汇编语言编程。

1.1.3 高级语言

虽然汇编语言较机器语言已有很大的改进，但仍是低级语言，它有如下两个主要缺点：

- 涉及过多细节。
- 与具体的计算机相关。

所以，汇编语言也被称为面向机器的语言。为了进一步提高编程效率，改进程序的可读性、可维护性，又出现了许多高级语言（也称为第三代语言），例如 FORTRAN、BASIC、PASCAL、Java、C 和 C++ 等，其中 C/C++ 是当今最流行的高级程序设计语言。

高级语言比低级语言更加抽象、简洁，主要表现为：

- 一条高级语言的指令相当于几条机器语言的指令。
- 用高级语言编写的程序同自然英语语言非常接近，易于学习。

同汇编语言类似，高级语言也需要专门的翻译程序，称为 compiler（编译器）或 interpreter（解释器），将它翻译成机器语言后，才能运行。

高级语言的发展也经历了从早期语言到结构化程序设计语言、从面向过程到非过程化程序语言的过程。相应地，软件的开发也由最初的个体手工作坊式的封闭式生产，发展为产业化、流水线式的工业化生产。

高级语言的下一个发展目标是面向应用，也就是说，只需要告诉程序你要干什么，程序就能自动生成算法，自动进行处理，这就是非过程化的程序语言。

1.1.4 C语言

C语言是贝尔实验室的 Dennis Ritchie 在 BCPL 语言的基础上开发出来的,1972 年在一台 DEC PDP-11 计算机上实现了最初的 C 语言。C 是作为 UNIX 操作系统的开发语言而开始广为人们所认识的。实际上,当今许多新的重要的操作系统都是用 C 或 C++编写的。在过去 20 年内,C 语言已经能够用在绝大多数的计算机上了。

C 语言是块结构(block-structured)语言,但是它还是常被称为结构化语言。这是因为它在结构上类似于 ALGOL、PASCAL 和 Modula-2(从技术上讲,块结构语言允许在过程和函数中定义过程或函数。用这种方法,全局和局部的概念可以通过“作用域”规则加以扩展,“作用域”管理变量和过程的“可见性”。因为 C 语言不允许在函数中定义函数,所以不能称之为通常意义上的块结构语言)。

结构化语言的显著特征是代码和数据的分离。这种语言能够把执行某个特殊任务的指令和数据从程序的其余部分分离出去、隐藏起来。获得隔离的一个方法是调用局部(临时)变量的子程序。通过使用局部变量,我们能够写出对程序其他部分没有副作用的子程序。这使得编写共享代码段的程序变得十分简单。如果开发了一些分离得很好的函数,在引用时仅需要知道函数做什么,不必知道它如何做。

结构化语言比非结构化语言更易于程序设计,用结构化语言编写的程序的清晰性使得它们更易于维护,这已是人们普遍接受的观点了。C 语言的主要结构成分是函数。在 C 语言中,函数是一种构件(程序块),是完成程序功能的基本构件。函数允许一个程序的几个任务被分别定义和编码,使程序模块化。一个好的函数不仅能正常工作,并且不会对程序的其他部分产生副作用。

用 C 语言编程,程序员可以获得高效机器代码,其效率几乎接近汇编语言代码。C 语言被程序员广泛使用的另一个原因是可以用它代替汇编语言。汇编语言使用的汇编指令,是能够在计算机上直接执行的二进制机器码的符号表示。汇编语言的每个操作都对应计算机执行的一个指令。虽然汇编语言可以达到最大的灵活性和最高的效率,但开发和调试汇编语言程序的困难是难以忍受的。非结构性使得汇编语言程序难于阅读、改进和维护。更重要的是,汇编语言程序的可移植性不好。

随着 C 语言的普及,又加上它的可移植性和高效率,许多程序员用它来设计各类程序。几乎所有的计算机上都有 C 语言编译程序,可移植性节省了时间和财力。

C 语言不仅在速度和结构上有它的优势,而且每个 C 语言系统都提供了专门的函数库,程序员可以根据不同需要对其进行剪裁,以适应各种程序的设计。由于它允许(更准确地说 是鼓励)分别编译,所以 C 语言可使程序员方便地管理大型项目,最大限度地减少重复劳动。

C 语言的主要特点如下:

- (1) 语言简洁,使用灵活、方便。C 语言只有 32 个关键字,程序书写形式自由。
- (2) 具有丰富的运算符和数据类型。
- (3) C 语言可以直接访问内存地址,能进行位操作,使其能够胜任开发操作系统的工作。
- (4) 生成的目标代码质量高,程序运行效率高。
- (5) 可移植性好。

1.1.5 C++语言

C语言在广泛应用的同时,其局限性也被暴露出来:

- (1) C 的类型检查机制相对较弱,这使得程序中的一些错误不能在编译时被发现。
- (2) C 本身几乎没有支持代码重用的语言结构,因此一个程序员编写的程序,很难应用到其他程序。
- (3) 当程序规模很大时,程序员很难控制程序,程序的可读性就会变得很差。

为了更好地克服 C 语言上述的局限,美国 Bell 实验室的 Bjarne Stroustrup 博士及其同事在 C 语言的基础上,从 Simula 中引进面向对象的特征,于 1980 年开发出来一种过程性与对象性相结合的程序设计语言。最初称之为“带类的 C”,1983 年 Rick Mascitti 将其更名为“C++”。

C++从 C 语言发展而来,比 C 更好,其优点主要包括:

- (1) 与 C 语言兼容,既支持面向对象的程序设计,也支持结构化的程序设计。同时,熟悉 C 语言的程序员,能够迅速掌握 C++语言。
- (2) 修补了 C 语言中的一些漏洞,提供了更好的类型检查和编译时的分析。即使程序员在 C++环境下继续写 C 代码,也能得到直接的好处。
- (3) 生成目标程序质量高,程序执行效率高。一般来说,用面向对象的 C++编写的程序,执行速度与 C 语言程序不相上下。
- (4) 提供了异常处理机制,简化了程序的出错处理。利用 throw、try 和 catch 等关键字,出错处理程序不必与正常的代码紧密结合,提高了程序的可靠性和可读性。
- (5) 函数可以重载及可以使用默认参数。重载允许相同的函数名具有不同的参数表,系统根据参数的个数和类型匹配相应的函数。默认参数可以使程序员能够以不同的方法调用同一个函数,并自动对某些默认参数提供默认值。
- (6) 提供了模板机制。模板包括类模板和函数模板两种,它们将数据类型作为参数。对于具体的数据类型,编译器自动地生成模板类或模板函数,它提供了源代码复用的一种手段。

1.2 计算机信息的表示

计算机最重要的功能是处理信息,如数值、文字、符号、语音、图形和图像等。在计算机内部,各种信息都必须采用数字化的形式被保存、加工与传送,掌握信息编码的概念和技术是至关重要的。

1.2.1 计算机数字系统

人们通常使用最熟悉的十进制数系进行计算,但是,几乎所有的计算机采用的都是二进制数系,所有的外界信息在被转化为不同的二进制数后,计算机才能对其进行传送、存储、加工和处理。在进行程序设计时,十进制、二进制、八进制和十六进制数系经常进行相互转换。无论哪种数系,其共同之处都是进位计数制。

采用二进制码表示信息，有如下几个优点：

(1) 易于物理实现。

因为具有两种稳定状态的物理器件是很多的，如门电路导通与截止，电压的高与低，而它们恰好对应表示 1 和 0 两个符号。假如采用十进制，要制造具有 10 种稳定状态的物理电路，那是非常困难的。

(2) 二进制运算简单。

数学推导证明，对 R 进制的算术求和、求积规则各有 $R(R+1)/2$ 种。如采用十进制，就有 55 种求和的运算规则，而二进制仅各有 3 种，因而简化了运算器等物理器件的设计。

(3) 机器可靠性高。

由于电压的高低、电流的有无等都是一种质的变化，两种状态分明。所以基 2 码传递的抗干扰能力强，鉴别信息的可靠性高。

(4) 通用性强。

基 2 码不仅成功地运用于数值信息编码（二进制），而且适用于各种非数值信息的数字化编码。特别是仅有的两个符号 0 和 1，正好与逻辑命题的两个值“真”与“假”相对应，从而为计算机实现逻辑运算和逻辑判断提供了方便。

1.2.2 进位计数制

进位制涉及到两个基本概念：基数和各数位的权。它们是构成某种进位制的两个基本要素。一般来说，如果数制只采用 R 个基本符号，则称为基 R 数制， R 称为数制的“基数”，而数制中每一固定位置对应的单位值称为“权”。

基数是指进位制中会产生进位的数值，它等于每个数位所允许的最大数码值加 1，也就是这种进位计数制中每个数位允许使用的数码个数。例如十进制，每个数位允许使用 0~9 这 10 个数码中的一个，基数为 10。或者说，各数位中允许使用的最大数码值为 9，再加 1 就会逢 10 进位，基数为 10。

一个数码处在不同的数位上，它所代表的数值不同，例如十进制中，个位上的 1 表示 10^0 ，而百位上的 1 表示 10^2 。因此，在进位制中每个数码所表示的数值等于该数码本身的价值乘以一个与它所在数位有关的常数，这个常数称为该位的位权，简称为权。显然，各数位的权值是不同的。例如十进制数，从小数点往左，整数部分的位权依次是： 10^0 ， 10^1 ， 10^2 ，…；从小数点往右，小数部分的位权依次是： 10^{-1} ， 10^{-2} ，…。

进位计数制具有统一的位置编号，小数点左边的第一位数码的位置编号是 0，向左依次增加，小数点右边的第一位数码的位置编号是 -1，向右依次减少。根据基数和位置编号可以总结出位权公式：

位权 = 基数^{位置编号}

例如，一个十进制数 125.47 可按权展开为：

$$125.47 = 1 \times 10^2 + 2 \times 10^1 + 5 \times 10^0 + 4 \times 10^{-1} + 7 \times 10^{-2}$$

1. 计算机常用的几种进位计数制

(1) 二进制。

在二进制中，每个数位仅能选择 0、1 这两个数码中的一个，逢 2 进位或借 1 当 2，基