

高等学校计算机基础教育规划教材

C/C++程序设计 进阶教程

张玉春 主编

清华大学出版社

高等学校计算机基础教育规划教材

C/C++程序设计 进阶教程

张玉春 主编

黄玥 李晓峰 刘通 副主编

孙元 赵永华 王洋 曹婧华 张春飞 段云娜 杨卉 胡瑞华 参编

清华大学出版社
北京

内 容 简 介

本书强调实用性,注重教材的理论与实践相结合,介绍程序设计的基本方法和基本技能。全书分三部分:第一部分 C 程序设计,主要介绍 C 语言基本类型数据及其运算、选择结构程序设计、循环结构程序设计、数组、函数、指针、结构体与共用体等;第二部分 C++ 基础,主要介绍类和对象、继承和派生等;第三部分 MFC 编程入门,主要介绍基于 MFC 创建单文档应用程序和基于对话框创建应用程序的方法,以及菜单和工具栏的编辑。

本书以程序设计为中心,语法介绍精练,内容叙述深入浅出、循序渐进,程序案例生动易懂,具有很好的启发性,每章均配有教学课件和精心设计的习题。

本书既可作为高等学校本科及专科程序设计课程的教材,又可作为自学者的参考用书,还可供各类考试人员复习参考。

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/C++ 程序设计进阶教程/张玉春主编. —北京:清华大学出版社,2019
(高等学校计算机基础教育规划教材)
ISBN 978-7-302-51773-3

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(2018)第 280598 号

责任编辑:袁勤勇
封面设计:常雪影
责任校对:时翠兰
责任印制:丛怀宇

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:三河市铭诚印务有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:20

字 数:452 千字

版 次:2019 年 3 月第 1 版

印 次:2019 年 3 月第 1 次印刷

印 数:1~1500

定 价:49.80 元

产品编号:081361-01

前言

本书共分三部分,从 C 语言到 C++,再到 Visual C++ (简称 VC++),进阶式地从面向过程语言程序设计介绍到面向对象语言程序设计,再到可视化的面向对象语言程序设计,既适合用 C 开发软件的需求,又适合用 VC++ 开发界面软件的需求。

第一部分 C 程序设计:是全书的基础,介绍 C 语言基本概念和编程的基本思想与方法。C 语言是一种结构化程序设计语言,兼有高级语言和低级语言的功能,不仅可用于编写系统软件,也可用于编写各类应用程序以及工业控制程序。目前流行的面向对象程序设计语言,如 C++、Java、C# 等都是在 C 语言的基础上发展派生而来的。通过学习 C 语言,学生不仅能够掌握程序设计的基本思想,也可为今后学习 Java、C++、VB 等语言打下良好的基础。

第二部分 C++ 基础:在 C 语言的基础上,介绍类和对象两大核心概念,以封装、继承为主线展开讲解。

第三部分 MFC 编程入门:介绍基于 Windows 编程的两种途径,并分别对两种编程的方法和操作步骤进行说明;讲述基于 MFC 编程的特点、MFC 程序的运行机制,重点介绍基于 MFC 创建单文档应用程序和基于对话框创建应用程序的方法,以及菜单和工具栏的编辑。

本书的特点是强调实用性,注重理论与实践相结合,目标是让学生掌握程序设计的基本方法和基本技能。本书内容组织注重基础,突出应用,兼顾提高,强化主干知识,弱化细枝末节。

目前,国内外教材一般是介绍纯 C 语言的,或者纯 C++,或者 Visual C++ 的,而从 C 介绍到 C++,再到 Visual C++ 的教材很少。

参加编写的教师及其编写内容如下:

作者姓名	编写内容	作者姓名	编写内容
王洋	第 1、8 章	张春飞	4.1.4 节、4.2.3 节、6.2 节、6.3 节
赵永华	第 2 章(除 2.6.6 节、2.6.7 节、2.8.3 节外)、3.1 节、3.2 节	刘通	第 7 章
李晓峰	2.6.6 节、2.6.7 节、第 3 章(除 3.1 节、3.2 节、3.5 节外)	段云娜	2.8.3 节、3.5 节、4.3.3 节的一部分、7.1.6 节

续表

作者姓名	编写内容	作者姓名	编写内容
黄玥	第4章(除4.1.4节、4.2.3节、4.3.3节的第一部分外)	杨卉	第9~11章
孙元	第5章	张玉春	第12、13章
曹婧华	6.1节、6.4~6.7节	胡瑞华	第14章、附录A~附录C

在本书的编写过程中,得到吉林大学公共计算机教学与研究中心领导的大力支持,在此表示感谢。在本书的出版过程中,得到清华大学出版社的大力支持,在此表示感谢。本书是所有参编教师辛勤努力的结果,在此向他们表示感谢。

由于编者水平有限,书中难免存在疏漏与不足之处,敬请读者指正。为方便教师的教学工作和读者的学习,本书有配套的源程序代码、习题答案和电子教案,需要者可通过出版社与编者联系获取。

张玉春

2018年8月

目录

第一部分 C 程序设计

第 1 章 C 语言与程序设计	3
1.1 程序设计语言及其发展	3
1.1.1 程序设计语言的发展历程	3
1.1.2 程序处理方式	4
1.2 程序的基本结构及其表示	5
1.3 C 语言概述	6
1.3.1 C 语言发展简史	6
1.3.2 C 语言的特点	6
1.3.3 简单的 C 程序介绍	7
1.3.4 C 程序的上机步骤	10
习题 1	14
第 2 章 基本类型数据及其运算	15
2.1 C 语言的数据类型	15
2.2 常量与变量	16
2.2.1 标识符	16
2.2.2 常量	17
2.2.3 变量	18
2.3 整型数据	19
2.3.1 整型常量	19
2.3.2 整型变量	20
2.4 实型数据	22
2.4.1 实型常量	22
2.4.2 实型变量	22
2.5 字符型数据	23
2.5.1 字符常量	23

2.5.2	字符变量	24
2.5.3	字符串常量	26
2.6	运算符与表达式	26
2.6.1	运算符概述	26
2.6.2	表达式概述	27
2.6.3	算术运算符与算术表达式	27
2.6.4	赋值运算符与赋值表达式	29
2.6.5	自增、自减运算符	32
2.6.6	关系运算符与关系表达式	33
2.6.7	逻辑运算符与逻辑表达式	34
2.6.8	逗号运算符与逗号表达式	35
2.7	数据的类型转换	36
2.8	数据的输入与输出	36
2.8.1	格式化输入输出函数	37
2.8.2	字符输入输出函数	44
2.8.3	文件格式化读写函数	45
习题 2	53
第 3 章	C 程序控制结构	55
3.1	C 语句	55
3.2	顺序结构程序举例	56
3.3	选择结构	57
3.3.1	if 语句	57
3.3.2	switch 语句	63
3.3.3	条件运算符与条件表达式	66
3.3.4	选择结构程序举例	66
3.4	循环结构	68
3.4.1	while 语句	68
3.4.2	do-while 语句	70
3.4.3	for 语句	71
3.4.4	循环嵌套	73
3.4.5	break 语句和 continue 语句	74
3.4.6	循环结构程序举例	75
3.5	文件字符读/写函数	78
3.5.1	文件读字符函数 fgetc	78
3.5.2	文件写字符函数 fputc	78
3.5.3	文件结束判断函数 feof	80
习题 3	81

第 4 章 数组	83
4.1 一维数组	83
4.1.1 一维数组的定义	83
4.1.2 一维数组元素的引用	84
4.1.3 一维数组的初始化	86
4.1.4 一维数组的指针	86
4.1.5 一维数组程序举例	87
4.2 二维数组	91
4.2.1 二维数组的定义	91
4.2.2 二维数组元素的引用	92
4.2.3 二维数组的指针	94
4.2.4 二维数组的初始化	95
4.2.5 二维数组程序举例	96
4.3 字符数组	99
4.3.1 字符数组的定义	99
4.3.2 字符数组的初始化	100
4.3.3 字符数组的输入与输出	101
4.3.4 字符串处理函数	107
4.3.5 字符数组应用举例	110
习题 4	114
第 5 章 函数	117
5.1 C 语言函数概述	117
5.2 函数的定义	118
5.2.1 函数定义的一般形式	118
5.2.2 函数参数与函数返回值	119
5.3 函数的调用	123
5.3.1 函数调用的一般形式	123
5.3.2 被调用函数的声明	124
5.4 函数的嵌套调用与递归调用	126
5.4.1 函数的嵌套调用	126
5.4.2 函数的递归调用	128
5.5 用数组做函数参数	130
5.5.1 用数组元素作函数参数	130
5.5.2 用一维数组名作函数参数	131
5.5.3 用二维数组名作函数参数	135
5.6 局部变量和全局变量	136

5.6.1	局部变量	136
5.6.2	全局变量	137
5.7	变量的存储类别	140
5.7.1	动态存储方式与静态存储方式	140
5.7.2	局部变量的存储类别	140
5.7.3	全局变量的存储类别	143
5.8	C语言预处理	145
5.8.1	宏定义	146
5.8.2	文件包含	148
习题5		149
第6章	指针	151
6.1	地址和指针	151
6.1.1	变量的地址和变量的值	151
6.1.2	间接寻址	152
6.2	指针变量的定义与引用	153
6.2.1	指针变量的类型	153
6.2.2	指向变量的指针变量的定义	153
6.2.3	指针变量的引用	154
6.2.4	指针运算符	154
6.2.5	指针运算	157
6.2.6	用指针变量作函数参数	158
6.3	指针与数组	160
6.3.1	指向一维数组的指针	160
6.3.2	指向二维数组的指针	163
6.3.3	用指向数组的指针变量作函数参数	168
6.4	指针与字符串	175
6.4.1	字符指针与字符数组	175
6.4.2	用指向字符的指针作函数参数	178
6.5	指针与函数	181
6.5.1	指向函数的指针	181
6.5.2	返回指针的函数	185
6.6	指向指针的指针	186
6.6.1	间接访问	186
6.6.2	指针数组	187
6.7	main函数的参数	189
习题6		190

第 7 章 结构体与共用体	192
7.1 结构体	192
7.1.1 结构体类型的定义.....	192
7.1.2 结构体类型变量定义.....	193
7.1.3 结构体变量的引用.....	195
7.1.4 结构体变量的赋值.....	196
7.1.5 结构体数组.....	197
7.1.6 文件结构体.....	200
7.1.7 文件数据块读写函数.....	201
7.1.8 结构体指针变量.....	204
7.1.9 用结构体数据作函数参数.....	207
7.2 共用体	209
7.2.1 共用体类型的定义.....	209
7.2.2 共用体变量的定义.....	209
7.2.3 共用体变量的引用和赋值.....	210
7.3 用 typedef 定义类型	212
7.4 动态链表	214
7.4.1 动态存储分配.....	214
7.4.2 动态链表概述.....	215
7.4.3 单链表的基本操作.....	215
习题 7	226
第 8 章 位运算	228
8.1 位运算符及其运算	228
8.1.1 位运算符.....	228
8.1.2 位运算应用举例.....	231
8.2 位段及其应用	234
8.2.1 位段.....	234
8.2.2 位段应用举例.....	236
习题 8	237

第二部分 C++ 基础

第 9 章 C++ 初步知识	241
9.1 从 C 到 C++	241
9.2 简单的 C++ 程序	241
9.3 C++ 的命名空间	242

9.4 C++ 头文件	243
9.5 C++ 基本输入输出	243
习题 9	244
第 10 章 类和对象	245
10.1 类的定义	245
10.2 对象的定义	246
10.3 成员访问权限	248
10.4 成员函数重载	249
10.5 构造函数和析构函数	250
习题 10	254
第 11 章 继承	256
11.1 类继承和派生的概念	256
11.2 类继承方式	258
习题 11	261

第三部分 MFC 编程入门

第 12 章 Windows 编程	265
12.1 基于 API 的 Windows 编程	266
12.2 基于 MFC 的 Windows 编程	270
12.2.1 MFC 应用程序框架类型	271
12.2.2 单文档应用程序创建	271
12.2.3 项目文件和项目配置	272
12.2.4 框架窗口、文档和视图	275
习题 12	275
第 13 章 对话框和控件	276
13.1 对话框	276
13.1.1 基于对话框应用程序创建	276
13.1.2 设置对话框属性	277
13.2 控件	278
13.2.1 控件的添加和布局	278
13.2.2 添加控件变量	279
13.3 消息和消息映射	280
13.4 添加对话框代码	280
13.5 对话框调用	281

13.5.1	创建单文档应用程序.....	281
13.5.2	添加对话框.....	282
13.5.3	创建对话框类.....	282
13.5.4	在程序中调用对话框.....	283
习题 13	284
第 14 章	菜单和工具栏	285
14.1	菜单.....	285
14.1.1	编辑菜单.....	285
14.1.2	菜单命令的消息映射.....	286
14.2	工具栏.....	287
14.2.1	编辑工具栏.....	288
14.2.2	工具栏按钮和菜单项相结合.....	288
14.3	综合应用.....	289
习题 14	295
附录 A	C 语言关键字	296
附录 B	运算符和结合性	297
附录 C	C 库函数	299
参考文献	305

语言与程序设计

第一部分

C程序设计

其发展

语言的发展与程序的发展历史

计算机的发展史，就是程序语言的发展史。

在计算机发展的初期，人们使用机器语言（即二进制代码）来编写程序。这种语言虽然直接对应于计算机的硬件结构，但编写起来非常繁琐且容易出错。随着计算机科学的进步，人们开始寻求更高级、更抽象的编程方式。于是，第一代编程语言（如Fortran、ALGOL）应运而生。这些语言引入了变量、运算符和基本的控制结构，使得编程变得更加直观和易于理解。第二代编程语言（如C、Pascal）进一步丰富了语言的表达能力，引入了数据类型、函数和模块化编程的概念。第三代编程语言（如Java、Python）则更加注重开发效率和易用性，引入了面向对象编程、垃圾回收等高级特性。如今，随着人工智能和大数据时代的到来，又出现了许多专门为特定领域设计的编程语言，如R用于统计分析、MATLAB用于工程计算等。可以说，程序语言的发展始终与计算机硬件的发展紧密相连，不断推动着计算机科学的进步。

C 语言与程序设计

计算机通过执行程序完成工作。所谓程序,就是一组计算机能够识别和执行的指令。每个指令都具有不同的含义,计算机能够有效地区分并执行。程序设计就是给出解决特定问题的过程,也就是编写程序。程序设计往往以某种程序设计语言为工具,设计出这种语言环境下的程序。

1.1 程序设计语言及其发展

程序设计语言也称为计算机语言,它是人和计算机进行交流的语言,是用于书写计算机程序的语言。程序,其实就是把需要做的事情用程序设计语言描述出来,然后在计算机上执行,以此来解决实际问题。

自从世界上第一台计算机诞生以来,程序设计语言就不断发展。特别是近十几年来,随着计算机软硬件的飞速发展,程序设计语言更加完善,种类也越来越多。目前,有多种不同类型的程序设计语言,主要分为低级语言和高级语言两大类。目前流行的程序中以面向对象的程序设计方法为主。

1.1.1 程序设计语言的发展历程

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。

1. 机器语言

机器语言是直接由二进制代码指令表达的计算机语言,可以用 0 和 1 组成的一串代码表示指令,该串代码有一定的位数且分成若干段,每段代码表示不同的含义。目前,计算机全称仍然是“电子计算机”。由于电子元器件有“开”和“关”两种稳定的工作状态,所以从物理上决定了目前电子计算机采用二进制进行运算,因此计算机不能识别人所识别的文字,而只能识别机器语言,即由 0 和 1 构成的代码。例如,某台计算机的字长为 16 位,即一条指令或某个信息由 16 个二进制数组成。16 个 0 和 1 可组成多种排列组合,从而形成计算机可以识别的不同操作。对于人来说,机器语言难以记忆和识别,出错的时候也难以修改。

2. 汇编语言

在汇编语言中,用助记符代替操作码,用地址标号或符号代替地址。用符号代替二进制编码,可以把二进制编码的机器语言变成汇编语言,即汇编语言实际上就是机器语言的符号化。例如,指令 ADD 代表加,指令 MOV 代表数据传送等。汇编程序的每一个指令都对应一个实际操作,类似的符号对于编程的人来说比机器语言更易懂,同时维护更方便。尽管如此,一般的汇编源程序还是相对冗长、复杂、易出错。由于汇编语言可以直接对硬件进行操作,所以其源程序经汇编生成的可执行文件不仅小,而且执行速度很快,缺点是由于与硬件紧密相关,所以可移植性差。

机器语言和汇编语言都是面向机器的语言,由于它们“贴近”计算机,所以被称为低级语言。低级语言与特定的机器有关,功效高,但使用复杂、烦琐、费时、易出错。

3. 高级语言

为了克服低级语言的缺点,20 世纪 50 年代中期出现了高级语言。高级语言主要是相对于低级语言而言的,这种语言接近于数学语言或人的自然语言,同时又不依赖于计算机硬件,其特点是在一定程度上与具体机器无关,易学、易用、易维护。

1954 年,第一个高级语言——FORTRAN 问世了,到目前为止,共有几百种高级语言出现,其中,影响较大的是 FORTRAN、Algol、Cobol、BASIC、Lisp、Pascal、C、Prolog、C++、VC、VB、Delphi、Java、Python 等。

高级语言的发展经历了从早期语言到结构化程序设计语言,从面向过程到面向对象程序语言的过程。20 世纪 60 年代中后期,软件的需求越来越多,软件的规模越来越大,但由于缺乏科学规范的系统规划、测试与评估标准,往往耗费巨资编写出来的软件系统,由于含有错误而无法使用,甚至带来巨大的损失。为了解决和避免这种情况,1969 年提出了结构化程序设计方法,1970 年,第一个结构化程序设计语言——Pascal 出现,标志着结构化程序设计时期的开始。在面向过程的程序设计方法中,数据与数据的处理过程是分开的,把处理过程按功能组成一个个独立的模块,在对数据进行处理的过程中,再分别调用各个独立的模块来实现功能。但是随着模块的程度越来越复杂,面向过程程序设计的缺点暴露出来,例如生产率低下、代码重用程度低、软件仍然很难维护等。针对这些缺点,面向对象的程序设计方法应运而生。面向对象的程序设计方法把数据和处理数据的过程当作一个整体,即对象。在分析过程中,把系统分解成一个个对象,同时把数据和相应数据的处理过程封装在对象中。在此之前的高级语言,几乎都是面向过程的,而 C++、VC、VB、Delphi、Java 等是典型的面向对象程序设计语言。

1.1.2 程序处理方式

计算机不能直接识别和执行用汇编语言或高级语言编写的程序,必须通过“翻译程序”将程序翻译成机器语言形式的目标程序,计算机才能识别和执行。这种“翻译”通常有两种方式,即解释方式和编译方式。

1. 解释方式

解释方式是将程序的每条语句一边翻译一边执行,即程序一边由相应语言的解释器

“翻译”成目标代码(即计算机可以识别的机器语言),一边执行。此类方式的典型程序语言是 BASIC。这种翻译方式较灵活,可以动态地调整、修改程序。但该方式没有对整个程序优化的过程,所以效率较低;而且不能生成独立的可执行文件,即程序不能脱离其解释器。

2. 编译方式

编译方式是将程序源代码“翻译”成目标代码(二进制),再经过连接程序连接,形成可执行文件。可执行文件可以脱离其语言环境独立执行,因此和解释类语言相比,其使用比较方便,可移植性好,效率较高。但如果程序需要修改,那么必须先修改源代码,再经过编译、连接生成新的可执行文件才能执行。大多数高级语言程序都是编译型的,例如 C、FORTRAN、Pascal 等。这类高级语言是面向过程的,即程序设计语言都属于过程化语言,在编写程序时需要具体指定每一个过程的细节。在编写规模较小的程序时,使用这些编程语言比较方便,但在处理规模较大的程序时,就显得很复杂。

1.2 程序的基本结构及其表示

程序的基本结构主要有顺序结构、选择(也可称为分支)结构和循环结构 3 种。数学证明,这 3 种结构可以组成所有程序结构。

顺序结构就是按照程序语句出现的先后顺序一步一步进行。如图 1-1 所示,当执行完成 A 所指定的操作之后,顺序执行 B 指定的操作。

选择结构则需要条件判断,当条件成立才会执行相应条件下的语句,如果条件都不成立,则执行其他的语句或什么也不执行。如图 1-2 所示,当条件 P 成立时,执行 A 操作;如果条件 P 不成立,则执行 B 操作。如图 1-3 所示,当条件 P 成立时,执行 A 操作;如果条件 P 不成立,则什么操作也不执行。

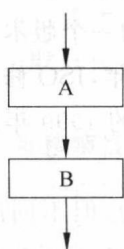


图 1-1 顺序结构

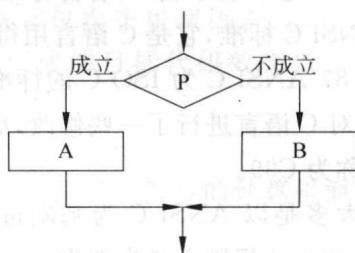


图 1-2 选择结构(1)

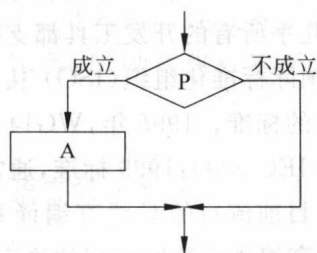


图 1-3 选择结构(2)

循环结构是在条件成立的前提下不断重复执行相同的语句,直至条件不成立为止,其循环分为直到型和当型两种类型。直到型循环如图 1-4 所示,首先执行 A 操作,然后判断条件 P 是否成立,如果成立则结束循环;如果不成立则继续执行 A 操作,循环往复,直到条件 P 成立则结束循环。当型循环如图 1-5 所示,首先判断条件 P 是否成立,如果不成立,则不执行任何操作;如果条件 P 成立,则执行 A 操作,循环往复,直至条件 P 不成立则结束循环。