Program Design

省级精品课程教材
21世纪高等教育计算机规划教材

C语言程序设计
（第2版）

□ 徐新爱 胡佳 主编
□ 卢昕 吴瑜鹏 副主编

- 与实际应用紧密结合的应用型教材
- 提供课件、答案、视频案例等资源

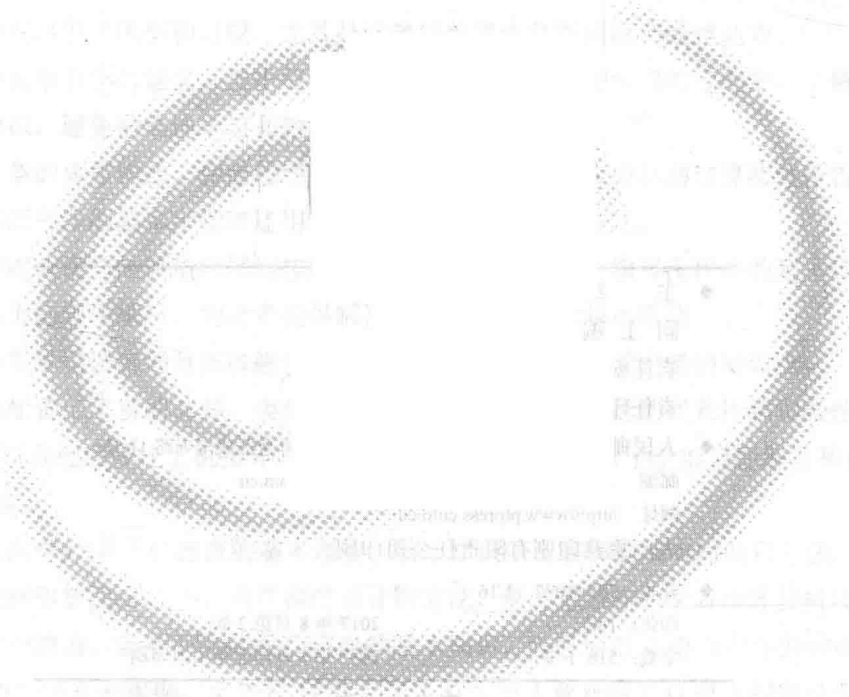
中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

Program Design

省级精品课程教材
21世纪高等教育计算机规划教材

C语言程序设计
（第2版）

□ 徐新爱 胡佳 主编
□ 卢昕 吴瑜鹏 副主编



人民邮电出版社
北京

图书在版编目（CIP）数据

C语言程序设计 / 徐新爱，胡佳主编. -- 2版. -- 北京 : 人民邮电出版社，2017.8
21世纪高等教育计算机规划教材
ISBN 978-7-115-46292-3

Ⅰ. ①C… Ⅱ. ①徐… ②胡… 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189006号

内 容 提 要

本书由浅入深地介绍了 C 语言的基本理论、基本知识以及编程的基本技能和方法。全书共 12 章，第 1 章～第 3 章介绍了 C 语言程序设计的基本概念、基本数据类型和运算符及表达式，第 4 章~第 6 章介绍了 C 语言程序设计的 3 种基本控制结构程序设计及应用，第 7 章~第 11 章介绍了 C 语言支持的构造数据类型及应用，第 12 章介绍了预处理命令。每章节都包含有丰富的贴近生活的例题和习题，帮助读者更深刻地掌握程序设计的基本思路和方法，巩固所学知识。

本书可以作为高等院校计算机类专业“C 语言程序设计”课程的教材，也可作为程序设计初学者的自学用书。

◆ 主　　编　徐新爱　胡　佳
副 主 编　卢　昕　吴瑜鹏
责任编辑　王　平
责任印制　彭志环

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路 11 号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
大厂聚鑫印刷有限责任公司印刷

◆ 开本：787×1092　1/16
印张：18.75　　2017 年 8 月第 2 版
字数：518 千字　　2017 年 8 月河北第 1 次印刷

定价：49.80 元

读者服务热线：(010)81055256　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字 20170147 号

第2版前言

PREFACE

C 语言是目前使用广泛的高级程序设计语言之一。它不仅具有丰富灵活的控制结构和数据结构，简洁而高效的表达式语句，清晰的程序结构和良好的可移植性等优点，而且具有直接操纵计算机硬件的强大能力。因此，C 语言已成为学习计算机程序设计的基础语言，且 C++、Java、C#、J#、Perl 等很多新型的语言均以 C 语言为基础。

2011 年，我校“C 语言程序设计”课程被江西省教育厅评为江西省高等院校高职高专精品课程。经过多年的课程建设，2015 年，该课程又被评为本科共享资源建设课程。诸位编者将积累多年的教学经验和 C 语言应用的体会，加之新的教改思想倾注于此书中。

1．本书特色

本书在编写过程中，注重 C 语言在学科中的基础地位，以程序设计为中心，内容全面、概念清晰、层次分明，讲述力求循序渐进、深入浅出、通俗易懂，尽可能地贴近读者的接受能力，并注重培养良好的程序设计风格和习惯，尤其是注重培养读者分析问题和编程能力。

（1）注重分析章节学习意义，提出每章的学习目标。让读者在学习的过程中，了解每章学习的意义及有关知识，避免被动地学习和教条式学习。

（2）精选每章的典型例题，注重程序设计思想。每章的典型应用都通过算法分析告诉读者怎么做，通过编写源程序的基本步骤讲述用计算机解决问题的基本方法。

（3）教材中部分章节的典型应用均按照问题描述、算法分析、编写源程序的基本步骤和源代码展示及运行结果的顺序编写，为读者提供解决具体应用问题的基本思路。

（4）章末的常见错误集锦旨在提醒读者在编写程序的过程中容易出错的知识点。

（5）每章均配备了丰富的习题，类型多样，由易到难，具有广泛的代表性和实践性。同时，配套出版了《C 语言程序设计上机指导与习题解答（第 2 版）》，提供了全部习题解答和实验指导。

2．章节内容

全书由浅入深地介绍了 C 语言的基本理论、基本知识以及编程的基本技能和方法，使读者能全面、系统地理解和掌握用 C 语言进行程序设计的方法，更注重培养初学者用计算机程序设计语言解决实际问题的能力。全书不仅涵盖 C 语言的基本知识，而且注重 C 语言程序的详细讲解，所有源程序都在 VC++6.0 上实现。全书共 12 章，第 1 章～第 3 章介绍了 C 语言程序设计的基本概念、基本数据类型和运算符及表达式，第 4 章～第 6 章介绍了 C 语言程序设计的 3 种基本控制结构程序设计及应用，第 7 章～第 11 章介绍了 C 语言支持的构造数据类型及应用，第 12 章介绍了预处理命令。

本书由南昌师范学院数学与计算机科学系教师编写，由徐新爱组织并统稿。其中，第 1～6 章、第 12 章及附录 1～5 由徐新爱编写，第 7～9 章由胡佳编写，第 10 章由吴瑜鹏编写，第 11 章由卢昕编写。新版全部由徐新爱修订，并补充了微视频。在本书的编写过程中，还得到了计算机教研室的全体教师的大力支持，我们深表感谢。本书配套的电子教案及相关资料登录人邮教育社区（http://www.ryjiaoyu.com）下载。

编　者

2017 年 5 月

目录 CONTENTS

第1章 C语言程序设计概述

◇ 学习意义

要学好 C 语言编程，必须了解计算机语言的基础知识，明确编程过程中可能会遇到的问题，做好相应的心理准备。

本章先介绍编程的预备知识，然后逐步介绍计算机程序设计语言的基础知识，如程序设计语言的发展和特点及应用，最后介绍 C 语言的历史及发展、C 语言常用的几种编译平台和 C 语言程序编译的基本步骤。

◇ 学习目标

- 了解编程的预备知识
- 了解程序设计语言的发展
- 掌握程序设计的基本过程
- 了解 C 语言的发展及特点
- 了解 C 语言的开发环境
- 掌握 VC++6.0 编译 C 语言程序的基本步骤

1.1 编程的预备知识

1.1.1 学习编程的心理准备

随着信息技术及 Internet 的普及，人们对计算机功能的要求也越来越高，如办公自动化、购物网络化、理财网络化、生活智能化和学习 APP 普及化以及城市智能化等。有预言说，编程技能变得越来越重要，将会变成 21 世纪生存技能中的核心竞争力。同时，随着时间的推移，编程的工作岗位也将有大幅增加。据美国劳工统计局的数据，在 2010 年有 91.3 万个计算机程序员职位，到 2020 年这一岗位预计将增长 30%，其他所有的非农就业岗位平均增幅预计只有 14%。因此，希望有志于编程的读者能够不断努力学习，实现自己的终极理想。

1．培养兴趣，对事物永远保持一颗好奇心

兴趣是对事物喜好或关切的情绪。心理学认为兴趣是人们力求认识某种事物和从事某项活动的意识倾向。它表现为人们对某件事物、某项活动的选择性态度和积极的情绪反应。兴趣在人的实践

活动中具有重要的意义，可以使人集中注意力，产生愉快紧张的心理状态。如果一直待在兴趣范围内，就会产生学习的欲望。同时，对事物保持一颗好奇心，也会触发学习的动机，积极地开展学习，享受学习的过程和学习带来的成就感，形成良性循环。

2．制订目标，通过实现目标获得成就感

学习过程中，制订计划是达到有效学习的一个重要方法。大学生活平淡无奇，有时觉得自己在不断地浪费时间，没有学到具体的知识，没有实现自己的目标。怎么办呢？最有效的方法就是制订自己的学习计划。计划可以包括短期计划，如日计划、周计划，也可以是长期计划，如月计划、学期计划或学年计划。通过制订的计划，按照计划去完成规定的目标。在完成了一定的学习目标后，可以适当对自己进行奖励或与同学、朋友一起分享，让自己的学习热情更持久，成就感也就会更加强烈，就有了挑战下一个目标的动力。同时，还可以积极参加一些计算机专业的学科竞赛，拓展自己的视野，结交更多志同道合的朋友，制订更高的目标。除此之外，也可以分学期去考一些计算机行业的证书如计算机软件资格证书及行业认证如微软认证、思科认证、Java 认证程序员（SCJP）等。

3．动手动脑，提高学习效果

很多学生在学习编程的过程中，经常会有这样的问题，“书都看了，也都看懂了，但还是写不出代码”。像这种情况的产生一般都是因为没有积极主动去思考和练习所致。在阅读程序的过程中，还要不断思考为什么要这样写，甚至可以质疑书上的内容，做到举一反三。在解决问题的过程中，如果有哪个环节存在疑问，一定要积极地去解决，绝不要把问题带到下一个学习环节。

1.1.2 认识编程

人与人之间交流的语言称为自然语言（Natural Language），计算机是一种机器，它能够识别的语言称为机器语言（Machine Language）。可见，人与计算机使用的是不同的语言，人不可能去学习计算机的机器语言，计算机也不可能学习人类的自然语言。因此，科学家们就设计了人与机器之间交流的工具——编程语言（Programming Language），又称为程序设计语言（Program Design Language，PDL）。而唯一能想到利用程序设计语言来解决问题的人是德国工程师康拉德·楚泽，他创造了编程语言，被称为编程语言之父，也称为数字计算机之父。

为了使计算机能够理解人的意图，人类就必须要将解决问题的思路、方法和手段通过计算机能够理解的形式告诉计算机，使得计算机能够根据人的指令一步一步去工作，完成某种特定的任务。这种人和计算机之间交流的过程就是编程。编程就是让计算机为解决某个问题而使用某种编程语言编写程序代码，并最终得到结果的过程。

有了编程语言，人类将问题用编程语言中的合法符号表达出来，然后通过转换程序转换成机器语言，最终交给计算机执行。在这个过程中，能使用编程语言写程序，并以此为职业的人，称为程序员（Programmer），或者程序设计师。程序员写出来的原始程序称为源代码（Source Code）、代码（Code）或源码。转换程序就是后面要介绍的编译程序。它们之间的关系如图 1-1 所示。

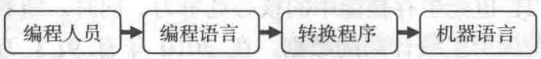


图 1-1 编程的基本过程

1.1.3 数据在计算机中的存储形式

计算机处理的信息可以是数字、文字、图像、声音等，这些信息数据分为数值型数据和非数值型数据。不管是哪种类型的信息在计算机中最终都是以二进制数据信息来表示和处理。计算机加工

处理信息时，会将信息按照一定规则转换成二进制数。因此，有了计算机世界，作为一名编程者就必须了解数的存储和表示。

1．计算机中的存储单位

随着计算机技术的发展，计算机的存储容量在不断扩充，存储单位也在不断出现，尤其是大数据时代下，存储单位在原有的 Byte、KB、MB、GB 基础上出现了 TB、PB、EB、ZB、YB、DB、NB 等。它们之间的换算关系如下所示。

8bit = 1 Byte (一字节)　　1024 B = 1 KB (KiloByte) (千字节)

1024 KB = 1 MB (MegaByte) (兆字节)　　1024 MB = 1 GB (GigaByte) (吉字节)

1024 GB = 1 TB (TeraByte) (太字节)　　1024 TB = 1 PB (PetaByte) (拍字节)

1024 PB = 1 EB (ExaByte) (艾字节)　　1024 EB = 1 ZB (ZetaByte) (皆字节)

1024 ZB = 1 YB (YottaByte) (佑字节)　　1024 YB = 1BB(BrontoByte)

1024 BB = 1 NB (NonaByte) (诺字节)　　1024 NB = 1 DB (DoggaByte) (刀字节)

计算机处理数据是把数据存放在计算机硬件系统的内存中。计算机系统的内存分为若干个存储单元，每个存储单元的大小称为一个字节（Byte）。字节是计算机存储容量的基本单位，每个字节由 8 个二进制位（bit）组成。一个字节中的每个二进制位的取值是 0 或 1，最右端的二进制位称为最低位或第 0 位，最左端的二进制位称为最高位或第 7 位。一个字符占用一个字节，一个汉字占用两个字节。CPU 一次能处理的二进制数的位数称为计算机的字长。

2．进制

进位制简称为进制，是人们规定的一种进位方法。计算机表示数有十进制、八进制、十六进制和二进制的进制方法。十进制是人们日常生活中使用的进制。因此，给定一个数的值没有特别说明都是以十进制表示。数据存储在计算机中是二进制形式，这是因为二进制形式具有容易表示、运算规则简单和节省设备的优点。但为了方便，又引入了八进制和十六进制。

对于任何 R 进制，表示只用 R 个基本数字（0～R-1）表示数值，R 称为该数制的基数（Radix）。R 进制数进行运算时，逢 R 进一。例如，十进制组成的数只包括 0～9 这 10 个数字，它的基数是 10，运算规则是逢十进一。不同进制的共同特点如下所述。

（1）每一种数制都有固定的数字。例如，十进制数制的基本数字有 0，1，2，…，9；二进制数制的基本数字有 0 和 1。

（2）每一种数制都可以使用位置表示法，即处于不同位置的数符所代表的值不同，与它所在位的权值有关。例如，二进制整数 M 和小数 N 分别表示为：

二进制整数 M：$M=(a_n a_{n-1}\cdots a_1 a_0)_2$

$$=a_n\times 2^n+a_{n-1}\times 2^{n-1}+\cdots+a_1\times 2^1+a_0\times 2^0$$

二进制小数 N：$N=(a_{-1}a_{-2}\cdots a_{-m})_2$

$$=a_{-1}\times 2^{-1}+a_{-2}\times 2^{-2}+\cdots+a_{-m}\times 2^{-m}$$

从上面的表示可以看出，位置表示法中每个数符的权值正好是基数的某次幂，幂的次数就是该数符所在的位置。因此，对任何一种进位计数制表示的数可以写成按权展开的多项式。

计算机表示数的不同进制的数码、基数、权和运算规则如表 1-1 所示。

表 1-1　　不同进制的数码、基数、权和运算规则

数制	十进制数	二进制数	八进制数	十六进制数
数码	0～9	0，1	0～7	0～9，A～F（a～f）
基数	10	2	8	16

续表

数制	十进制数	二进制数	八进制数	十六进制数
权	从低位到高位依次为：10^0，10^1，10^2，…，10^n 从低位到高位依次为：10^{-1}，10^{-2}，10^{-3}，…，10^{-m}	从低位到高位依次为：2^0，2^1，2^2，…，2^n 从低位到高位依次为：2^{-1}，2^{-2}，2^{-3}，…，2^{-m}	从低位到高位依次为：8^0，8^1，8^2，…，8^n 从低位到高位依次为：8^{-1}，8^{-2}，8^{-3}，…，8^{-m}	从低位到高位依次为：16^0，16^1，16^2，…，16^n 从低位到高位依次为：16^{-1}，16^{-2}，16^{-3}，…，16^{-m}
运算规则	逢十进一	逢二进一	逢八进一	逢十六进一

3．进制间的相互转换

（1）八进制、十六进制、二进制转换为十进制。这种转换规则是将八进制、十六进制、二进制的数按权展开，然后计算该多项式的各项和，此“和”就是十进制表示的数。

【例 1-1】 二进制数 1011.1 转换为十进制数。

$$1\times2^3+0\times2^2+1\times2^1+1\times2^0+1\times2^{-1}=11.5$$

【例 1-2】 八进制数 13.1 转换为十进制数。

$$1\times8^1+3\times8^0+1\times8^{-1}=11.125$$

【例 1-3】 十六进制数 13.1 转换为十进制数。

$$1\times16^1+3\times16^0+1\times16^{-1}=19.0625$$

（2）十进制转换为八进制、十六进制、二进制。这种转换规则是采用短除法，其中整数部分除以基数（R）取余，小数部分乘以基数（R）取整。

【例 1-4】 十进制数 10.25 转换为二进制数。

具体方法为：整数部分 10 除以 2，每一步取余，直至商为 0 结束，然后将余数从下往上写出 1010；小数部分 0.25 乘以 2，每步取结果的整数部分，直至结果为 0 或需要的位数结束，然后将整数部分从上到下写出 01，最后将这两部分合成，中间用小数点分开即得结果 1010.01。具体步骤如图 1-2 所示。

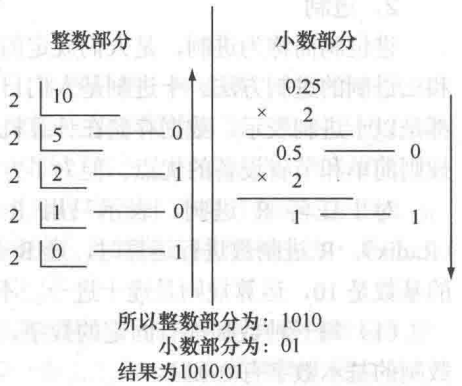


图 1-2　转换过程

（3）二进制与八进制或十六进制相互转换。二进制转换成八进制的转换规则是：以小数点为分界点，把二进制数的整数部分从右往左每 3 位分成一组，不够 3 位的前面补 0；把二进制数的小数部分从左往右每 3 位分成一组，不够 3 位的后面补 0，然后把每一组转换成对应的十进制数即可。

【例 1-5】 二进制数 10101111.10111 转换为八进制数。

具体步骤为：

第 1 步（划分组）：　　（010　101　111.101　110）

第 2 步（转换成十进制数）：　2　5　7. 5　6

第 3 步（八进制）：　　257.56

二进制数转换成十六进制数的规则是：以小数点为分界点，把二进制数的整数部分从右往左每 4 位分成一组，不够 4 位的前面补 0；把二进制数的小数部分从左往右每 4 位分成一组，不够 4 位的后面补 0，然后把每一组转换成对应的十六进制数即可。

【例 1-6】 二进制数 10101111.10111 转换为十六进制数。

具体步骤为：

第 1 步（划分组）：　　（1010　1111.1011 1000）

第 2 步（转换成十六进制数）：　　A　　F.　B　　8

第 3 步（十六进制）：　　AF.B8

反过来，将一个八进制数或十六进制数转换为二进制数的规则是将该八进制数或十六进制数的每一位，按照十进制转换二进制的方法变成用 3 位或 4 位二进制表示，然后按照顺序排列，就转换成二进制数了。如果整数部分最高位或小数部分最低位是 0，可以省略不写。

【例 1-7】 八进制数 257.56 转换为二进制数。

2　　5　　7.　5　　6

=（010　101　111.101　110）

=10101111.10111

同样，将一个十六进制数转换为二进制数的规则是将该十六进制数的每一位，按照十进制转换二进制的方法变成用 4 位二进制表示的序列，然后按照顺序排列，就转换为二进制数了。

【例 1-8】 十六进制数 AF.B8 转换为二进制数。

A　　F.　　B　　8

=（1010　1111.1011　1000）

=10101111.10111

4．数据在内存中的存储形式

所有的数据在计算机中都是以二进制形式存放，其机内表示形式称为机器数。机器数的表示方法通常有原码、反码和补码。运算规则是补码运算最简单，可连同符号位一起参与运算。因此，计算机系统规定：数据都是以补码的形式存放在内存中。

（1）原码。原码是指将数值的最高位用作符号位，其他各位表示除符号外的数据值的二进制形式。其中 0 表示正号，1 表示负号。

【例 1-9】 若采用 8 个二进制位存储以下数据（此位数称为机器字长），则

[+1]原=0000 0001　　[−1]原=1000 0001

[+45]原=0010 1101　　[−45]原=1010 1101

（2）反码。对于一个带符号的数来说，正数的反码与其原码相同，负数的反码为其原码除符号位以外的各位取反。

【例 1-10】 若机器字长 n 等于 8，则

[+1]反=0000 0001　　[−1]反=1111 1110

[+45]反=0010 1101　　[−45]反=1101 0010

（3）补码。数值的补码的最高位是符号位，其他位取决于该数是正数还是负数。其中，正数的补码与其原码相同，负数的补码为其反码加 1。

【例 1-11】 若机器字长 n 等于 8，则

[+1]补=0000 0001　　[−1]补=1111 1111

[+45]补=0010 1101　　[−45]补=1101 0011

1.2 程序设计语言的基础

1.2.1 程序设计语言的发展

编程语言又称为程序设计语言，是一组用来定义计算机程序的语法规则，是进行程序设计的必

备工具。它是一种被标准化的交流技巧，用来向计算机发出指令。自 20 世纪 60 年代以来，世界上公布的程序设计语言已有上千种之多，但是只有很小一部分得到了广泛应用。程序设计语言的发展历程如图 1-3 所示。

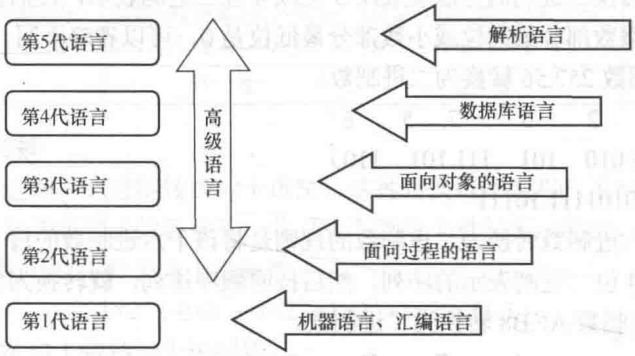


图 1-3 程序设计语言的发展历程

从程序设计语言的发展历程来看，程序设计语言分为 5 代，其中除第 1 代之外，其余 4 代都属于高级语言，其中第 2 代语言和第 3 代语言作为高级语言的典型代表，第 4 代语言和第 5 代语言更具有特定的应用。下面将计算机程序设计语言的发展分为 4 个阶段进行介绍。

1. 第 1 阶段

第 1 代语言属于程序设计语言发展的第 1 阶段，简称为 1GL，又称为低级语言。这代语言包括机器语言和汇编语言，是在计算机诞生和发展初期使用的语言。

（1）机器语言。机器语言是计算机能够唯一识别的语言，也称为面向机器的语言。机器语言程序是由二进制数字“0”和“1”组成的串，依赖具体的计算机系统，因而程序的通用性、移植性都很差。使用机器语言编写的程序，由于每条指令都对应计算机一个特定的基本动作，所以程序占用内存少、执行效率高。但是，使用机器语言编程对程序员来说是一件非常痛苦和困难的事情，一不小心，“0”写成了“1”，就可能会造成灾难性的后果。

（2）汇编语言。为了解决使用机器语言编写程序的困难，人们想到了使用助记符号来代替机器指令，如用“ADD”代表加法。像这种使用助记符号来表示计算机指令的语言称为符号语言，也称汇编语言。在汇编语言中，每一条用符号来表示的汇编指令与计算机机器指令一一对应；记忆难度减少了，而且易于检查和修改程序错误，指令、数据的存放位置也可以由计算机自动分配。

用汇编语言编写的程序称为源程序，计算机又不认识这些符号。因此，就出现了汇编程序。汇编程序就是专门负责将这些符号翻译成机器语言。它们之间的关系如图 1-4 所示。

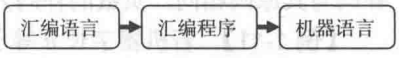


图 1-4 汇编语言的汇编过程

使用汇编语言编写计算机程序，需要程序员十分熟悉计算机系统的硬件结构，因此，从程序设计本身上来看仍然是低效率和烦琐。但是，与计算机硬件系统关系密切的特定场合，如对时空效率要求很高的系统核心程序以及实时控制程序等，汇编语言仍然是十分强有力的程序设计工具。

2. 第 2 阶段

第 2 代语言和第 3 代语言属于程序设计语言发展的第 2 阶段，简称为 2GL，起始于 20 世纪 50 年代中期。汇编语言虽然在一定程度上克服了机器语言的不足，但是对程序员又提出了更高的要求。因此，人们意识到，应该设计一些接近于数学语言或人的自然语言的编程语言，同时又不依赖于计算机硬件，编写出来的程序能在所有机器上通用，高级语言也就由此产生。高级语言是按照一

定的语法规则，由表达各种意义的运算对象和运算方法构成。1954 年，第一个完全脱离机器硬件的高级语言——Fortran 问世了，继 Fortran 之后先后出现过几百种高级语言。当前使用较普遍的计算机编程语言主要有：C 语言、C++语言、Java 语言、C#语言和 Delphi 等。

（1）从应用角度分类。高级语言可以分为基础语言、结构化语言和专用语言。

① 基础语言。基础语言也称通用语言。它历史悠久，流传很广，有大量的已开发的软件库。像 Fortran、Cobol、Basic、Algol 等都是基础语言。Fortran 语言是目前国际上广为流行，也是使用最早的一种高级语言，在工程与科学计算中占有重要地位，备受科技人员的欢迎。Basic 语言是在 20 世纪 60 年代初为适应分时系统而研制的一种交互式语言，可用于一般的数值计算与事务处理。Basic 语言结构简单，易学易用，并且具有交互能力，成为许多初学者学习程序设计的入门语言。

② 结构化语言。20 世纪 70 年代以来，结构化程序设计和软件工程的思想日益为人们所接受和欣赏，先后出现了一些很有影响力的结构化语言，它们直接支持结构化的控制结构，具有很强的过程结构和数据结构能力。Pascal、C、ADA 语言就是结构化语言的突出代表。

③ 专用语言。专用语言是为某种特殊应用而专门设计的语言，通常具有特殊的语法形式。一般来说，这种语言的应用范围狭窄，移植性和可维护性不如结构化程序设计语言。目前使用的专业语言已有数百种，应用比较广泛的有 APL 语言、Forth 语言、LISP 语言。

（2）从客观系统的描述分类。程序设计语言分为面向过程语言和面向对象语言。

① 面向过程语言。以“数据结构+算法”程序设计范式构成的程序设计语言，称为面向过程语言。面向过程的语言具有以下特点。

采用模块分解与功能抽象的方法，自顶向下，逐步求精；按功能划分为若干个基本的功能模块，形成一个树状结构。各模块间的关系尽可能简单，功能上相对独立。每一个功能模块内部都由顺序结构、选择结构和循环结构 3 种基本结构组成。

② 面向对象语言。以“对象+消息”程序设计范式构成的程序设计语言，称为面向对象语言。面向对象语言的目标是实现软件的集成化，把相互联系的数据以及对数据的操作封装成通用的功能模块，各功能模块可以相互组合，完成具体的应用，还可以重复使用，而用户不必关心其功能是如何实现的。目前比较流行的面向对象语言有 Java、C++等。

（3）从需要的转换方式分类。高级语言分为解释型语言和编译型语言。

① 解释型语言。解释型转换是指将编写的程序一边翻译一边执行，每翻译一句就执行一 句，称这个解释型转换为解释器（interpreter）。需要解释器的语言称为解释型语言。采用解释型语言写出来的代码常称为脚本（script）。常见的解释型语言有 Basic 语言和 Perl 语言。用解释型语言写出来的程序，每次执行都要再次翻译，所以不足之处是效率比较低，但优点是可以跨平台。

② 编译型语言。编译型转换是将编写的程序通过编译器（compiler）转换成计算机能执行的机器码。需要编译器的语言称为编译型语言。C 语言就是编译型语言。用编译型语言写出来的程序，每次执行都是直接执行其机器码，所以执行效率高。

使用高级语言编写程序的优点是：编程相对简单、直观、易理解、不容易出错；同时，高级语言是独立于计算机、程序通用性好、具有较好的移植性。

3. 第 3 阶段

第 4 代语言属于程序设计语言发展的第 3 阶段，简称 3GL。3GL 是非过程化语言，编码时只需说明“做什么”，无需描述算法细节。

3GL 是以数据库管理系统所提供的功能为核心，进一步构造了开发高层软件系统的开发环境，如报表生成、菜单生成系统、图形图像处理系统和决策支持系统等，从而为用户提供了一个良好的应用开发环境。它提供了功能强大的非过程化问题定义手段，用户只需告知系统做什么，而无

需说明怎么做，因此可大大提高软件生产率。进入 20 世纪 90 年代，随着计算机软硬件技术的发展和应用水平的提高，大量基于数据库管理系统的 3GL 商品化软件已在计算机应用开发领域中获得广泛应用，成为了面向数据库应用开发的主流工具，如 Oracle 应用开发环境、Power Builder 等。它们缩短了软件开发周期，提高软件质量发挥了巨大的作用，为软件开发注入了新的活力。

4. 第 4 阶段

第 5 代语言属于程序设计语言发展的第 4 阶段，简称 4GL。4GL 是指解析语言，是博科资讯 MAP 发明的一种面向应用的程序开发语言。从计算机技术角度看，该语言是面向管理业务的 DSL（Domain-Specific Language，领域特定语言），使用该语言的目的是基于标准化的管理业务描述定义，用于开发具有丰富业务模型的企业管理应用，例如供应链管理系统（SCM）、企业资源计划系统（ERP）、客户关系管理系统（CRM）、供应商关系管理系统（SRM）等。

1.2.2 程序设计语言的特点及发展趋势

1. 程序设计语言的特点

每一种程序设计语言都可以被看作是一套包含语法、词汇和含义的正式规范。这些规范通常包括数据和数据结构、指令及流程控制、引用机制和重用。

（1）数据和数据结构。现代计算机内部的数据都只以二进制存储，即开-关模式（on-off）。现实世界中代表信息的各种数据，例如名字、银行账号、度量以及同样低端的二元数据，都经由程序设计语言整理。

（2）指令及流程控制。一旦数据被确定，机器必须被告知如何对这些数据进行处理。较简单的指令可以使用关键字或定义好的语法结构来完成。不同的语言利用序列系统来取得或组合这些语句。除此之外，一个语言中的其他指令也可以用来控制处理过程（例如分支、循环等）。

（3）引用机制和重用。引用的中心思想是必须有一种间接设计存储空间的方法，最常见的方法是命名变量。根据不同的语言，进一步地引用可以包括指向其他存储空间的指针。还有一种类似的方法就是命名一组指令。大多数程序设计语言使用宏调用、过程调用或函数调用。使用这些代替的名字能让程序更灵活，并更具重用性。

2. 程序设计语言的发展趋势

程序设计语言是开发软件的重要平台，它的发展趋势是模块化、简明性和形式化。

（1）模块化。不仅语言具有模块成分，程序也是由模块组成的，而且语言本身的结构也是模块化。

（2）简明性。涉及的基本概念不多，成分简单、结构清晰、易学易用。

（3）形式化。发展合适的形式体系，以描述语言的语法、语义、语用。

1.2.3 程序设计的基本过程

程序是软件开发人员根据用户需求开发的、用程序设计语言描述的适合计算机执行的指令（语句）序列，这个序列指示计算机如何完成一个具体的任务。也有人把程序定义为人们为解决某种问题用计算机可以识别的代码编排的一系列加工步骤。任何一种计算机语言程序的主体都是由 3 种基本结构组成的，即顺序结构、选择结构和循环结构。

一个程序至少包括以下两个部分。

（1）对数据的描述，在程序中要指定数据的类型和数据的组织形式，即数据结构（Data Structure）。

（2）对操作的描述，即操作步骤，也就是算法（Algorithm）。1976 年，著名的计算机科学家、Pascal 程序设计语言之父、结构化程序设计首创者、1984 年图灵奖获得者沃斯(Niklaus Wirth)就提出了“Algorithms + Data Structures = Programs”，即“算法＋数据结构＝程序”。在这个著名的公式