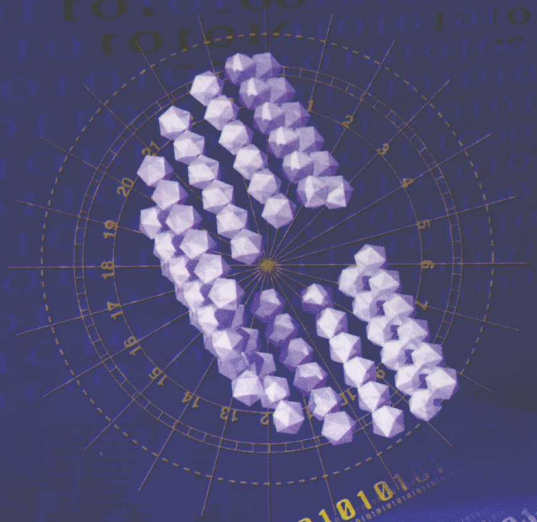
高职高专计算机系列教材

C语言程序设计与数据结构

周成义 汤德俊 钟 菊 主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

高职高专计算机系列教材

C 语言程序设计与数据结构

主　编　周成义　汤德俊　钟　菊
副主编　张英强　敖广武　孙红岩
参　编　温春友　王洪利　谢吉荣　康　伟

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书是在“C 语言程序设计”和“数据结构”这两门课的基础上，结合高职教学的特点，探索用数据结构的算法作为 C 语言程序设计的方法，使算法和程序相结合，将这两门课程整合为一门课程。本书以 C 语言为主线，在介绍 C 语言的过程中，将数据结构的基本知识融入其中。本书注重基础，注重实际应用。

本书共分为 10 章，第 1 章介绍了 C 语言的基本概念、结构化程序设计方法、数据结构的基本概念；第 2 章为 C 语言的数据结构、运算符和表达式；第 3 章为 C 语言的程序控制语句；第 4 章介绍了数组的基本知识；第 5 章为 C 语言函数的基本知识；第 6 章介绍了数据的顺序存储结构及其应用；第 7 章为指针的基本知识；第 8 章介绍了数据的链式存储结构及其应用；第 9 章介绍了树与图的基本知识；第 10 章介绍了文件的操作。

本教材可作为高等职业学院和高等专科院校计算机应用专业（包括网络方向）的教材，也可作为工科专业 C 语言教材。

图书在版编目（CIP）数据

C 语言程序设计与数据结构 / 周成义，汤德俊，钟菊主编. —北京：中国铁道出版社，2007.7
（高职高专计算机系列教材）
ISBN 978-7-113-07793-8

Ⅰ.C… Ⅱ.①周…②汤…③钟… Ⅲ.①C 语言－程序设计－高等学校：技术学校－教材②数据结构－高等学校：技术学校－教材 Ⅳ.TP312 TP311.12

中国版本图书馆 CIP 数据核字（2007）第 122713 号

书　　名：C 语言程序设计与数据结构
作　　者：周成义　汤德俊　钟　菊　等
出版发行：中国铁道出版社（100054，北京市宣武区右安门西街 8 号）
策划编辑：严晓舟　秦绪好
责任编辑：陈　宏
特邀编辑：李振华
封面设计：高　洋
封面制作：白　雪
责任较对：包　宁
印　　刷：三河市国英印务有限公司
开　　本：787×1092　1/16　　**印张**：12.75　　**字数**：295 千
版　　本：2007 年 8 月第 1 版　　2007 年 8 月第 1 次印刷
印　　数：1～5 000 册
书　　号：ISBN 978-7-113-07793-8/TP・2158
定　　价：20.00 元

前　言

高等职业教育是以培养应用技术型人才为目标的高等教育，为适应国家发展高职教育的要求，本着高职教育在理论上“必需、够用”的前提下提高学生的实际应用能力这一原则，我们根据高职计算机专业就业特点和用人单位要求，结合多年的教学实践，加强C语言与数据结构的衔接，将C语言与数据结构这两门课进行整合，形成一本新的教材。本教材探索出用数据结构方法实现C语言的程序设计，把算法和程序结合在一起，加强学生用C语言解决实际问题的能力。

本书为C语言和数据结构的入门与应用教材，全书共分10章，第1章介绍了C语言的基本概念、结构化程序设计方法、数据结构的基本概念；第2章为C语言的数据结构、运算符与表达式；第3章为C语言的程序控制语句；第4章介绍了数组的基本知识；第5章为C语言函数的基本知识；第6章介绍了数据的顺序存储结构及其应用；第7章为指针的基本知识；第8章介绍了数据的链式存储结构及其应用；第9章介绍了树与图的基本知识；第10章介绍了文件的操作。

本教材根据作者多年的教学经验编写而成，书中程序都在Turbo C 2.0环境下调试通过。

本教材主要为计算机应用专业的学生编写，其他要求参加全国计算机等级考试的学生也可以选用本教材。本教材的教学时数可以根据本校的具体情况而定，一般可在84～96学时之间。

本教材由周成义、汤德俊、钟菊主编，张英强、敖广武、孙红岩任副主编。参加本教材编写的还有温春友、王洪利、谢吉荣、康伟等几位老师。全书由周成义负责统稿。本书的出版得到了辽宁科技大学的支持，同时也得到了一些高职院校老师的建议和意见，在此谨向关心和支持本书出版的同志们表示衷心感谢。

本教材是高职计算机教学改革的一种尝试和探索，尽管作者已经非常认真和努力，但是书中不足之处在所难免，敬请各位专家和广大读者批评指正。任何意见和建议请发至：askdccy@126.com。

编　者

2007年6月

目 录

第 1 章　概　　论

C 语言是一种结构化程序设计语言。数据结构可以实现结构化编程思想。本章首先介绍有关 C 语言的基本知识，然后介绍结构化程序思想和方法，最后介绍数据结构的有关概念。

1.1　C 语言概述

C 语言是国内外流行的、应用比较广泛的计算机高级语言，它同时具有高级语言和低级语言的特点。它适合用来编写系统软件，也可以用来编写应用软件。

1.1.1　计算机语言的发展过程

1. 第一代语言——机器语言

机器语言（属于低级语言）是由 1、0 组成的机器指令的集合，是面向机器的计算机语言。要使计算机按人的指示工作，就必须使计算机懂得人的意图，接受人向它发出的指令和信息。人要和机器交换信息就要解决一个“语言”问题。计算机并不懂人的语言（无论是中文或者是英文），例如，写 $y=2x+3$，机器不接受。它只能识别 0 和 1 两种状态，如光电输入机中纸带有孔的地方代表 1，无孔的地方代表 0。由 0 和 1 组成各种排列组合，通过线路转变成电信号，让计算机执行各种不同的操作。

这种直接用 0 和 1 组成的机器指令编写的程序，就是机器语言源程序。对计算机来说，这是它唯一能直接“听”懂的语言。所以，常常称为面向机器的语言。但是，对使用计算机的人来说，这是十分难懂的语言，它难读、难记、难写，容易出错，不同机型又不通用。显然人和机器之间的通信存在巨大的障碍，只有消除这个障碍，才能让用户使用起来方便容易，机器又能理解，计算机才能发挥更大的作用。为此，人们发明了一种汇编语言。

2. 第二代语言——汇编语言

汇编语言（属于低级语言）也叫符号语言，它是把用二进制数表示的指令，用一些符号来表示，例如，用表示操作的英文缩写来代替汇编语言指令代码，用十六进制表示数字。下面是一段汇编语言的代码：

```
LDA     A       取出 A
ADD     B       A 和 B 相加
STA     C       存入 C
PRINT   C       打印 C
STOP    停止
```

第 1 条 LDA，即 Load Accumulate 的缩写，是“取数”的操作，表示取出 A。

第 2 条 ADD，是“加”的意思，表示将 A 和 B 中存放的数相加。

第 3 条 STA，即 Store Accumulate 的缩写，表示把结果放在 C 中。

第 4 条 PRINT，表示输出 C 的值。

第 5 条 STOP，表示程序停止。

这种用符号代替二进制代码的指令称为汇编语言。像 LDA、ADD 等这些符号称为指令符号或助记符。用汇编语言编写的程序称为汇编语言源程序，常常简称为汇编语言程序。

这种语言，相比机器语言容易读、容易记，但是机器却不能识别。因此，计算机是无法直接执行的。正如一个不懂汉语的外国人到中国来无法和中国人直接交谈，只好求助翻译。在计算机中，也同样采取这种方法，人们编写程序用汇编语言，然后请一位“翻译”，把汇编语言程序翻译成机器能懂得的机器语言程序，这个翻译过程称为“汇编”。汇编后产生的机器代码称为目标程序。翻译可以手工完成，但做起来既烦琐又单调，且容易出错。实际中，人们让计算机来进行。因而就发明了担任翻译的程序，称为汇编程序。

汇编语言使程序设计工作前进了一大步，但是仍然存在很多的缺点：第一，不方便求解问题过程的描述，例如，一个数学公式，汇编语言的表达形式与人们的习惯表达式差距很大；第二，它仍是面向机器的语言，不同机型，汇编语言也不一样，因此用它编制的程序，没有通用性。为了克服这些不足之处，人们进一步发明了高级语言。

3. 第三代语言——算法语言

算法语言（属于高级语言）是接近人类的自然语言和数学语言，且计算机能接受的语言。由表达不同意义的“关键字”和“表达式”按照一定的语法语义规则组成、完全不依赖机器的指令系统。这样的高级语言为人们提供了很大的方便，编制出来的程序容易读、容易记，也方便修改、调试，大大提高了编制程序的效率和程序的通用性，方便推广和交流，从而极大地推动了计算机的普及和应用。

例如，使用高级语言 BASIC，要想得到的结果，只需要写出 print 3*8*Sin(3.14159/2)，计算机就能计算出输出结果。

高级语言离人们的理解越接近，离计算机的理解就越远。计算机是不能理解那些单词、数学表达式的。所以，为了消除人机之间的障碍，还是得求助“翻译”。这种“翻译”通常有两种方法，即编译方式和解释方式。

编译方式是：事先编好一个称为编译程序的机器指令程序，并放在计算机中，把用高级语言编写的源程序输入计算机，编译程序就把源程序整个翻译成用机器指令表示的目标程序。然后执行该目标程序，得到计算结果。

解释方式是：事先编好一个称为解释程序的机器指令程序，并放在计算机中，把用高级语言编写的源程序输入计算机，它并不像编译方式那样把源程序整个翻译成用机器指令表示的目标程序，而是逐句的翻译，翻译出一句立即执行，即边解释边执行。这种方式比编译方式费时间。

FORTRAN、ALGOL、COBOL、C 等高级语言采用编译执行方式，而 BASIC 等高级语言可采用解释执行方式。

由于编译（解释）程序代替了人工将高级语言源程序翻译为机器指令程序，因而大大降低了编程人员的工作量。自从有了高级语言后，科技人员、大学生、中学生等都能很快地学会使用计算机，可以完全不考虑机器指令，也可以不必深入理解计算机的内部结构和工作原理，就能方便地使用计算机进行各种科学计算或事务处理等工作。因此有人说，高级语言的出现是计算机发展史中“最惊人的成就”。

目前，世界上已有一百多种高级语言，比较流行的有几十种，下面列举几种。

- FORTRAN（Formula Translator）是世界上最早出现的高级语言，从1954年问世以来，经过几次大的发展，功能有很大的增强。它特别适用于科学、工程计算。
- COBOL（Common Business Oriented Language）适用于非数值计算的商业、管理领域。
- PASCAL语言是最早出现的结构化语言，适用于计算机教学。
- Ada语言是一种工程化的语言，适用于大型软件工程。
- C语言是近年来广泛推广的结构化语言，适用于编写系统软件。
- BASIC语言是一种简单会话式语言，应用较广泛。

4. 第四代语言——非过程化语言

非过程化语言（属于高生产率语言）就是目前比较流行的面向对象的语言，这种语言只要求编程人员对问题进行描述，比如C++、Java语言等。

目前也出现了智能化语言，主要用于人工智能等领域。比较有代表性的有LISP语言和PROLOG语言。

1.1.2 C语言出现的历史背景

以前操作系统等系统软件主要是用汇编语言编写的，由于汇编语言依赖于计算机硬件，程序的可读性和可移植性差，因此，为了提高程序的可读性和可移植性，人们设想能否找到一种既具有一般高级语言特性又具有低级语言特性的语言。于是，C语言就应运而生了。

1963年英国剑桥大学推出了CPL（Combined Programming Language）语言。CPL语言在ALGOL60的基础上更接近硬件，但规模比较大，难以实现。1967年英国剑桥大学的Matin Richards对CPL语言作了简化，推出了BCPL（Basic Combined Programming Language）语言。1970年美国贝尔实验室的Ken Thompson以BCPL语言为基础，又作了进一步简化，设计出了很简单而且很接近硬件的B语言（取BCPL的第一个字母），并用B语言写了第一个UNIX操作系统。但B语言过于简单，功能有限。1972年至1973年间，贝尔实验室的D.M.Ritchie在B语言的基础上设计出了C语言（取BCPL的第二个字母）。C语言既保持了BCPL和B语言的优点（精练、接近硬件），又克服了它们的缺点（过于简单、数据无类型等）。最初的C语言只是为描述和实现UNIX操作系统提供一种工作语言而设计的。1973年，K.Thompson和D.M.Ritchie两个人合作把UNIX的90%以上用C改写（即UNIX第5版，原来的UNIX操作系统是1969年由美国贝尔实验室的K.Thompson和D.M.Ritchie开发成功的，是用汇编语言写的）。

后来，C语言多次进行了改进，到了1975年UNIX第6版公布后，C语言的突出优点才引起人们普遍注意。1977年出现了不依赖于具体机器的C语言编译文本——《可移植C语言编译程序》，使C移植到其他机器时所需要的工作大大简化了，这也推动了UNIX操作系统迅速地在各种机器上实现。C语言和UNIX可以说是一对孪生兄弟，在发展过程中相辅相成。1978年以后，C语言已先后移植到大、中、小、微机上，独立于UNIX了。现在C语言已经风靡全世界，成为世界上应用最广泛的几种计算机语言之一。

在C语言的基础上，1983年又由贝尔实验室的Bjarne Stroustrup推出了C++。C++进一步扩充和完善了C语言，成为一种面向对象的程序设计语言。C++目前流行的最新版本

是 Borland C++、Symantec C++和 Microsoft Visual C++。C++提出了一些更为深入的概念，它所支持的那些面向对象的概念很容易将问题空间直接地映射到程序空间，为程序员提供了一种与传统结构程序设计不同的思维方式和编程方法。因而也增加了整个语言的复杂性，掌握起来有一定难度。C 语言是 C++语言的基础，C++语言和 C 语言在很多方面是兼容的。因此，掌握了 C 语言，再进一步学习 C++语言就能以一种熟悉的语法来学习面向对象的语言，从而达到事半功倍的目的。

目前流行的 C 语言版本有标准 C、ANSI C、Turbo C、Quick C。各种版本的 C 语言编译系统虽然基本部分是相同的，但也有一些不同。在微型机上使用有 Microsoft C、Turbo C、Quick C 等，它们的不同版本又略有差异，因此读者应了解所有计算机系统的 C 编译的特点和规定。

1.1.3 C 语言的特点

C 语言的存在和发展在于它有不同于其他计算机语言的特点。C 语言是一种结构化、模块化、编译式通过的程序设计语言。它具有良好的移植性，适合于编制各种系统软件和应用软件，各种各样的程序设计任务几乎都可以用 C 语言完成。其主要特点有如下几个方面。

（1）语言简洁紧凑、使用方便灵活。C 语言一共有 32 个关键字，9 种控制语句，程序书写形式自由，主要用小写字母，压缩了一切不必要的成分。C 语言比其他高级语言和程序设计更简练，代码行少。

（2）语言运算能力丰富。C 的运算符包含的范围很广泛，共有 34 种运算符。C 把括号、赋值、强制类型转换等都作为运算符处理，从而使 C 的运算类型极其丰富，表达式类型多样化。灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

（3）C 语言有丰富的数据类型，具有现代化语言的各种数据结构。C 的数据类型有：整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等。能用来实现各种复杂的数据结构的运算。尤其是指针类型数据，使用起来比 PASCAL 更灵活、多样。

（4）具有结构化的控制语句（如 if...else 语句、while 语句、do...while 语句、switch 语句、for 语句）。C 语言用函数作为程序模块以实现模块化，是结构化的理想语言，符合现代编程风格要求。

（5）语法限制不太严格，程序设计自由度大。例如，对数组下标越界不作检查，由程序员自己保证程序的正确。对变量的类型使用比较灵活，例如，整型数据与字符型数据可以通用。一般的高级语言语法检查比较严，能检查出几乎所有的语法错误。而 C 语言允许程序员有较大的自由度，因此放宽了语法检查。程序员应当仔细检查程序，保证其正确，而不要过分依赖 C 编译程序去查错。“限制”与“灵活”是一对矛盾。限制严格，就失去灵活性；而强调灵活，就必然放松限制。一个对程序设计不熟练的人员，编一个正确的 C 程序可能会比编一个其他高级语言程序难一些。也就是说，对用 C 语言的人，要求对程序设计更熟练一些。

（6）C 语言允许直接访问物理地址，能进行位（bit）操作，能实现汇编语言的大部分功能，可以直接对硬件进行操作。因此 C 既具有高级语言的功能，又具有低级语言的许多功能，可用来编写系统软件。C 语言的这种双重性，使它既是成功的系统描述语言，又是成功的程序设计语言。有人把 C 称为“高级语言中的低级语言”，也有人称它为“中级语言”，意为兼有高级和低级语言的特点。

（7）生成目标代码质量高，程序执行效率高。一般只比汇编程序生成的目标代码效率低10%～20%。

（8）用C语言写的程序可移植性好（与汇编语言比）。基本上不做修改就能用于各种型号的计算机和各种操作系统。

上面只介绍了C语言的最容易理解的一般特点，至于C语言内部的其他特点将结合以后各章的内容作介绍。由于C语言的这些优点，使C语言应用面很广。许多大的软件都用C编写，这主要是由于C的可移植性好和硬件控制能力高，表达和运算能力强。许多以前只能用汇编语言处理的问题现在可以改用C语言来处理了。

1.1.4 C语言的程序介绍

下面介绍几个简单的C程序，从而分析C程序的特性。

【例1-1】输出字符串。

```
main()
{
   printf("Welcome to Liaoning University of Science and Technology!\n");
}
```

本程序的作用是打印如下信息：

```
Welcome to Liaoning University of Science and Technology!
```

其中，main表示“主函数”，每一个C语言程序都必须有一个main函数，函数体由大括号“{}”括起来，本例中主函数的函数体中只有一个输出语句，双引号中的字符串原样输出，“\n”为输出换行符。

【例1-2】求两个整数的和。

```
main()
{
    int a,b,sum;
    a=12;b=34;
    sum=a+b;
    printf("sum=%d\n",sum);
}
```

本程序的作用是求两个整数 a 和 b 之和sum，本程序输出信息为：

```
    sum=46
```

【例1-3】输入两个整数，输出较大者。

```
main()                              /*主函数*/
{
    int a,b,c;                      /*定义变量*/
    scanf("%d,%d",&a,&b);           /*输入变量a和b的值*/
    c=max(a,b);                     /*调用max函数，将得到的值赋给c*/
    printf("max=%d",c);             /*输出c的值*/
}
int max(x,y)                        /*定义max函数，函数值为整型，x，y为形式参数*/
int x,y;                            /*对形参x、y作类型定义*/
{
int z;                              /*对max函数中用到的就是z进行类型定义*/
if(x>y)z=x
```

```
    else z=y;
    return(z);                          /*将 z 的值作为函数值返回调用处*/
    }
```

本程序包括两个函数：主函数 main 和被调用函数 max。max 函数的作用是将 x 和 y 中较大者的值赋值给变量 z。return 语句将 z 的值返回给主调函数 main。

通过以上几个例子，可以总结出以下几点。

（1）一个 C 语言源程序可以由一个或多个源文件组成。

（2）每个源文件可由一个或多个函数组成。

（3）一个源程序不论由多少个文件组成，都有一个且只能有一个 main 函数，即主函数。

（4）源程序中可以有预处理命令（include 命令仅为其中的一种），预处理命令通常应放在源文件或源程序的最前面。

（5）每一个说明，每一个语句都必须以分号结尾。但预处理命令、函数头和花括号“}”之后不能加分号。

（6）标识符与关键字之间必须至少加一个空格以示间隔。若已有明显的间隔符，也可不再加空格来间隔。

从书写清晰，便于阅读、理解、维护的角度出发，在书写程序时应遵循以下规范。

（1）一个说明或一个语句占一行。

（2）用“{}”括起来的部分，通常表示了程序的某一层次结构。“{}”一般与该结构语句的第一个字母对齐，并单独占一行。

（3）低一层次的语句或说明可比高一层次的语句或说明缩进若干格后书写。以便看起来更加清晰，增加程序的可读性。

在编程时应力求遵循这些规则，以养成良好的编程风格。

1.1.5 C 语言程序设计的上机步骤

C 语言采用编译方式将源程序转换成二进制目标代码。从编写一个 C 程序到完成运行得到结果一般需要经过以下几个步骤。

1. 编辑

所谓编辑，包括以下内容：①在 C 语言的编辑环境中输入程序源代码并逐个字符输入到计算机内存中。②修改源程序。③将修改好的源程序，保存到磁盘文件中，其文件扩展名为“.c”（如 file1.c，此文件叫 C 语言的源程序文件）。

2. 编译

编译就是将已经编辑好的源程序翻译成二进制的目标程序。在编译时，系统还要对源程序进行语法检查，如发现错误，则显示出错信息，此时应重新进入编辑状态，对源程序进行修改后再重新编译，直到通过编译为止，生成扩展名为“.obj”的同名文件。

3. 连接

将各个模块的二进制代码与系统标准模块连接处理后，得到可执行文件，生成扩展名为“.exe”的同名文件。

4. 执行

经过编译和连接的可执行文件，只有在操作系统的支持和管理下才能运行。

1.2 结构化程序设计的方法

结构化程序设计的概念是以模块化设计为中心，将待开发的软件系统划分为若干个相互独立的模块，这样使完成每一个模块的工作变单纯而明确，为设计一些较大的软件打下了良好的基础。

由于模块相互独立，因此在设计其中一个模块时，不会受到其他模块的影响，因而可将原来较为复杂的问题简化为一系列简单模块的设计。模块的独立性还为扩充已有的系统、建立新系统带来了不少的方便，因为可以充分利用现有的模块作积木式的扩展。

按照结构化程序设计的观点，任何算法功能都可以通过由程序模块组成的3种基本程序结构（顺序结构、选择结构和循环结构）的组合来实现。

1.2.1 结构化程序设计的思想

结构化程序设计的基本思想是采用“自顶向下、逐步求精”的程序设计方法和“单入口、单出口”的控制结构。自顶向下、逐步求精的程序设计方法从问题本身开始，经过逐步细化，将解决问题的步骤分解为由基本程序结构模块组成的结构化程序框图；“单入口、单出口”的思想认为一个复杂的程序，如果它仅是由顺序、选择和循环3种基本程序结构通过组合、嵌套构成，那么这个新构造的程序一定是一个单入口、单出口的程序。据此就很容易编写出结构良好、易于调试的程序来。

1.2.2 结构化程序设计的步骤

学习计算机语言的目的是利用这种语言工具设计出可供计算机运行的程序。在拿到一个需要求解的实际问题之后，怎样才能编写出程序呢？

一般来说，从实际问题抽象出数学模型是有关领域专业工作者的任务。程序设计人员的工作，最关键的一步就是设计算法。一般用流程图来表示算法。如果算法是正确的，将它转换为任何一种高级语言程序都是不困难的，这一步常称为“编码”。

结构化程序设计的步骤如下：

1. 分析问题

准确而完整地描述和分析问题是解决问题的第一步。例如，该问题要求你设计的程序的整体功能是什么？涉及哪些数据对象？

2. 数据对象的描述

采用何种数据结构？用高级语言实现时，采用何种数据类型？

3. 算法的设计

在设计结构化的算法时一般采用下述方法。

（1）自顶向下

这种方法是先有全局，先进行整体设计，然后再进行下层的设计，逐步地实现精细化。采用这种方法能够做到胸有全局，能全盘考虑，不至于顾此失彼、头重脚轻。

（2）逐步细化

一步步地将上层的任务分解成较小的、易于实现的任务，直到可以很简单地实现为止。

（3）模块化

将一个大任务分解成若干个较小的部分，每一部分承担一定的功能，称为“功能模块”。各个模块都可以被独立地编写和调试，便于组织人力完成较复杂的任务。

4. 编写源程序

这一步也称为“编码”，就是将已设计好的算法用高级语言源程序表达出来。

5. 程序的编译与运行

编译的一项重要功能就是检查源程序的语法错误，编译成功之后得到与源程序对应的目标程序。运行就是指运行可执行的目标文件。

1.3 数据结构与算法

1.3.1 数据结构的基本概念

在学习数据结构知识之前，先掌握一些基本概念和术语的定义。

（1）数据

数据（Data）是信息的载体，它能够被计算机识别、存储和加工、处理，它是计算机程序加工的原料。应用程序可以处理各种各样的数据，它可以是数值数据，也可以是非数值数据。数值数据是一些整数、实数或复数；非数值数据包括符号、文字、图形、图像、语言等。

（2）数据元素

数据（Data Element）元素是数据的基本单位，在计算机程序中通常作为一个整体进行考虑和处理。一个数据元素可由若干个数据项组成。在不同的条件下，数据元素又可称为元素、结点、顶点、记录等。例如，学生信息检索系统中学生信息表（见表 1-1）中的一个记录、教学计划编排问题中的一个顶点等，都被称为一个数据元素。

表 1-1 学生信息表

学　号	姓　名	性　别	年　龄	职　务
206001	王力宏	男	18	学委
206002	张三丰	男	19	班长
……	……	……	……	……

（3）数据项

数据项（Data Item）是指不可分割的、含有独立意义的最小数据单位，数据项有时也称字段或域。例如，学籍管理系统中学生信息表的每一个数据元素就是一个学生记录。它包括学生的学号、姓名、性别、籍贯、出生年月、成绩等数据项。这些数据项分为两种：一种叫做初等项，如学生的性别、籍贯等，这些数据项是在数据处理时不能再分割的最小单位；另一种叫做组合项，如学生的成绩，它可以划分为数学、物理、化学等更小的项。通常，在解决实际应用问题的时候是把每个学生当作一个基本单位进行访问和处理的。

（4）数据结构

数据结构（Data Structure）是指相互之间存在着一种或多种关系的数据元素的集合。在任何问题中，数据元素都不会是孤立的。在它们之间都存在着这样或那样的关系，这种