21 世纪大学计算机专业教材

程序设计与 C 语言

(第 2 版)

梁力　原盛　韩菁　编著

西安交通大学出版社

内容简介

本教材以程序设计方法为主线，以C语言作为典型的程序设计语言，全面系统地介绍了程序设计的发展、结构化程序设计方法和面向对象程序设计方法。并用C语言具体描述了结构化程序设计。本教材详细地讲述了C语言的基本概念、语法规则和语义特点，通过三个层次的例子介绍了程序设计的基本方法和技巧。

本教材语言通俗易懂、内容深入浅出、重点突出，范例程序丰富，实用性、技巧性强不仅可以作为计算机专业本科生及相关专业的程序设计课程的教材，也可以供自学使用。

图书在版编目(CIP)数据

程序设计与C语言/梁力等编著. —西安：西安交通大学出版社，2005.8
21世纪大学计算机专业教材
ISBN 7-5605-2056-1

Ⅰ.程… Ⅱ.梁… Ⅲ.C语言-程序设计-高等学校-教材 Ⅳ.TP312

中国版本图书馆CIP数据核字(2005)第063818号

书　　名	程序设计与C语言(第2版)
编　　著	梁力　原盛　韩菁
责任编辑	屈晓燕　贺峰涛
出版发行	西安交通大学出版社
地　　址	西安市兴庆南路25号(邮编：710049)
电　　话	(029)82668315　82669096(总编办) (029)82668357　82667874(发行部)
电子邮件	eibooks@163.com
印　　刷	蓝田县立新印刷厂
版　　次	2005年8月第2版　2005年8月第1次印刷
开　　本	787mm×1092mm　1/16
印　　张	21
印　　数	0 001～4 000
字　　数	509千字
书　　号	ISBN 7-5605-2056-1/TP·412
定　　价	26.00元

前　言

本书是一本讲授程序设计和程序设计语言的教科书。

程序设计是学习计算机知识非常重要的基础课程之一。是学好计算机系列课程的基础。程序设计课程包括两方面的内容:程序设计方法和程序设计语言。自从第一台计算机诞生以来程序设计方法与程序设计语言就一直不断的发展,从主要用于计算的程序设计,到 20 世纪 70 年代的结构化程序设计,进而到 80 年代的面向对象的程序设计,其目的为了使计算机这一 20 世纪最伟大的科研成果,人类智慧的结晶能够更好地为人类服务。

程序设计语言从机器语言、汇编语言、高级语言,到目前更加便捷,更加拟人化的各种可视化语言使人们感到了计算机科学的飞速发展和它的勃勃生机。

编写本教材的目的是为学生打下一个扎扎实实的程序设计的基本理论与基本方法的基础,熟练掌握一门典型的程序设计语言,以适应计算机科学不断推出的新方法,新工具。书中以介绍程序设计方法为主,结合一种典型的程序设计语言 C 语言,通过列举大量的应用实例,系统地、较为全面地介绍结构化程序设计的思想和方法。

学习程序设计的目的是建立程序设计的基本思想,掌握程序设计的基本方法。不可能只通过一门程序设计课程的学习就要求学生立即编写大型软件。要想成为一名合格的程序设计者,他一定要掌握了许多计算机理论,还要不断地实践,不断地积累经验。任何人不可能只靠在学校学习的一门语言包打天下。只有掌握了程序设计的基本理论和基本方法,才能适应计算机不断发展的需要。

作者长期从事程序设计基础课程的教学工作,本教材是作者长期教学实践的总结。编写教材过程中,作者研究了学生学习程序设计认识规律,采用了通俗易懂的语言,循序渐进的方法。本教材的特点如下:

1. 以程序设计方法为纲,较为系统全面地介绍了程序设计的发展和结构化程序设计,其目的是让读者掌握程序设计的基本理论和基本方法。

2. 以 C 语言作为典型的程序设计语言,介绍了用程序设计语言描述结构化程序设计方法。按结构化程序设计方法的 3 种基本结构件介绍 C 语言的控制语句和程序设计。

3. 各章中列举了丰富的例子。例子分为 3 个层次,第 1 层次,为帮助读者理解基本语法而设;第 2 层次,以介绍基本的程序设计方法;第 3 层次,综合举一些具有较强编程技巧和实用性的例子。教师可以根据教学实际需要和不同程序的学生选取例子。

4. 本书各章精选了典型习题供学生选作。程序设计是一门实践性很强的课程,要求学生大量阅读程序,动手编写程序,上机调试程序。从实践中不断总结积累经验。

5. 书中的例子都已上机检验。

6. 本书既可以作为计算机专业学生的教材,又可作为非计算机专业学生的教材。

本书由梁力主编、统稿。原盛编写了第 1、6 章,韩菁编写了第 2、3、4、7、9 章,梁力编写了

第5、8章。西安交通大学电信学院副院长董渭清教授仔细审阅了书稿，并提出了宝贵意见。在编写本教材过程中受到了西安交通大学软件学院副院长曾明教授、计算机系副主任陈建明副教授的大力支持。许多资深教授与作者深入讨论如何搞好程序设计基础教学并提出了许多建设性意见。徐宏喆老师在编书过程中给予了我们热情的帮助。在使用本教材的过程中，钱屹老师提出许多具体的意见。在这里一并表示感谢。本书经过几届西安交通大学电信学院本科生使用，编者在原书稿的基础上进行了修改。由于编者水平有限，书中肯定会有不少缺点和不足，恳请广大读者批评指正。

编　者

2004年月5月

目　录

第 1 章　程序设计基础

本章重点介绍程序设计的基本理论、基本知识和基本方法，为读者今后更好地从事程序设计和软件开发打下良好的基础。首先介绍计算机的基本组成，其次对程序设计和程序设计语言的基本概念及发展做以概述，着重介绍结构化程序设计和面向对象程序设计方法。

1.1　计算机基础

计算机被人类称为是 20 世纪最伟大的发明，问世几十年来，经历了电子管计算机、晶体管计算机、集成电路(IC)计算机、大规模集成电路(LSI)计算机、“智能”化计算机 5 个发展阶段。计算机的应用很广泛，涉及到国民经济、社会生活的各个领域和各个行业，应用于科学与工程计算、数据处理与信息管理、工业生产自动化、计算机辅助系统、人工智能、图像处理及个人服务等各个方面。

计算机技术与通信技术的结合，出现了计算机网络通信(internet/intranet)，尤其是 Internet 的快速发展，使得世界各地的人们可以互相交流，包括文字和视频的交流，缩短了彼此空间上的距离，使我们足不出户，就可以了解各国的国情。同时，随着 Internet 广泛应用，远程教学、远程医疗和电子商务的发展，使我们的生活方式和生活环境发生很大的变化，远程教学和远程医疗使得边远地区的人们也可以享受先进科学带来的好处，而电子商务的蓬勃发展，使我们可以在网上生存，从网上购得自己所需要的一切东西。当今社会已步入 21 世纪的信息时代，了解计算机，学会使用计算机是时代对我们的要求。

计算机系统是一个很复杂的系统，主要由两大部分组成：硬件系统和软件系统。硬件是计算机的设备装置，对计算机而言，更重要的是指令。告诉计算机怎么做的指令集合称为软件，或者称为程序。

1.1.1　计算机硬件系统

计算机硬件系统是由各种物理部件组成的，是指计算机的硬件实体。直观地看，计算机硬件系统就是一大堆物理设备。一台计算机从硬件系统看主要由 4 个部件组成：中央处理器、存储器、输入设备和输出设备，如图 1-1 所示(图中实线表示数据线，虚线表示控制线)。

1. 存储器

存储器，是计算机用来存放程序和数据以及运行时数据的记忆设备。根据存储器和中央处理器的关系，存储器可分为主存储器(简称主存，又称内存)和外存储器(简称外

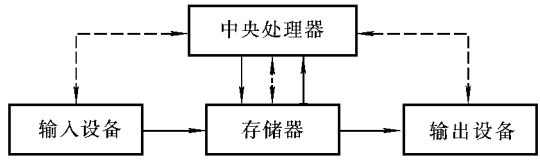

图 1-1　计算机的基本部件

存,又称辅存)。

主存储器紧靠 CPU,其内部储存的数据和指令控制着 CPU 的操作。主存储器由一系列存储位置(也称为地址)组成,每个位置保存 8 位(8 个二进制数位)或一个字节的信息。主存储器的一个重要特性是,任何位置或字节都能以固定的时间进行访问。因此,有时我们把主存也称为随机存取存储器(Random Access Memory,RAM)。这种访问特性与用磁带存储信息形成鲜明对比,要读磁带中部的信息必须转动带子到适当的位置才行。因此,访问信息的时间取决于信息所在的位置。这种类型的存储器称为顺序存取存储器(Sequential Access Memory)。主存储器有两个重要特征,分别是容量和速率。主存容量用它包含的字节数来测量。在个人计算机中,主存从 64MB 到 512MB,甚至更多。主存储器速率用它从特定位置读取信息所需时间来测量,个人计算机的典型速率是 10～60ns。

外存储器设置在主机外部,用来存放相对不经常使用的程序和数据。外存储器的容量一般较大,而且可以移动,如移动硬盘等,采用 USB 接口,更加方便计算机之间进行信息交换,常用的外存有磁盘、磁带和光盘。磁盘又分为软盘和硬盘。

2. 中央处理器

中央处理器简称 CPU(Central Processing Unit),它是整个计算机的核心,计算机发生的所有动作都是受 CPU 控制的。CPU 主要包括运算器、控制器和寄存器 3 个部分。

运算器也称为算术逻辑部件 ALU(Arithmetic Logic Unit),主要完成数据的算术运算(如加、减、乘、除)和逻辑运算(如与、或、非运算)。有趣的是计算机采用二进制而不是十进制来表示数。二进制只有两个数,0 和 1。采用二进制是因为计算机的基本部件都是开关元器件,类似我们使用的开关(开/关)。数字 0 或 1 用开关状态来表示。早期的计算机,采用机械继电器作开关,因此非常庞大,需要占据整个房间。而现在,这些开关采用超小型晶体管,所以整个计算机可以做在单个的硅芯片上。因此,将这些芯片称为微处理器。英特尔的奔腾Ⅱ处理器芯片,在 4.8 英寸高和 6.0 英寸宽的组件里有大约 7 500 000 个晶体管。而现在的奔腾 4XE 处理器芯片,在更小的空间里集成了 1 亿 8 千万个晶体管。

控制器负责从内存中读取各种指令,并对指令进行分析,根据指令的具体要求向计算机各部件发出相应的控制信息,协调它们的工作,从而完成对整个计算机系统的控制,因此控制器是计算机的指挥控制中心。CPU 内部的寄存器主要用来存放经常使用的数据。

3. 输入设备和输出设备

当 CPU 完成计算,把程序与结果存入主存后,我们还需要知道计算机处理信息的情况及其他结果的方式。我们还需要能储存信息,使计算机关机后不丢失已经处理好的结果。输入设备和输出设备就具有这两种功能。为了给计算机输入数据和指令,有上百种不同类型的输入设备。熟悉的常用设备有键盘、鼠标。不太熟悉的设备有扫描仪、麦克、游戏杆以及手写板。同样,也有上百种不同类型的输出设备,每种都指定输出不同类型的信息。常用输出设备包括各种打印机、显示器和绘图仪以及音响设备。有些设备既能处理输入,又能处理输出,它们是典型的输入/输出设备。大多数输入输出设备基于某种类型的磁记录技术。比如软盘、硬盘、U 盘以及可擦写的 CD 和 DVD。

1.1.2 计算机软件系统

就计算机自身而言,若没有程序指令或软件,只有硬件是没有用的,就像立体音响只有装

入 CD 才有音乐一样。我们可以通过任何硬件设备向计算机系统中输入指令，比如键盘或磁盘驱动器。可以说软件(也就是程序)是计算机的灵魂。一般情况下，我们可以将软件粗略地划分为应用软件和系统软件两大类。两者之间的区别有时可能是模糊的，但是一般来说，应用软件包括问题求解以及在特殊领域或应用范围之内提供服务。系统软件支持其他软件的开发与执行。在某种意义上，系统软件是应用软件与底层硬件之间的桥梁。这个目标使程序员与机器的底层细节隔开，并且提高了生产效率。计算机系统的组织可由图 1－2 说明。

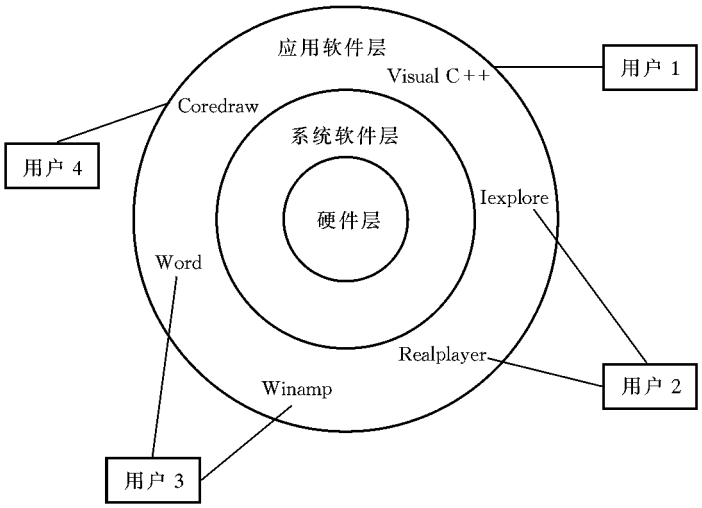

图 1－2　计算机系统组织图

1. 系统软件

系统软件很重要的一块是操作系统。操作系统(operating system)是控制和管理计算机资源的软件。这些资源包括内存、输入/输出设备和 CPU。操作系统提供许多服务，比如为程序分配内存和处理输入/输出设备，对显示器、键盘、鼠标以及磁盘驱动器的控制。操作系统提供的一个重要服务是文件系统，它控制着信息在磁盘中的组织，以便能快速地检索并找到所需要的文件。常用的操作系统有 windows98，windows2000，windowsXP，Linux，Unix，Macos 等。

另一种系统软件称为翻译系统(translation system)。翻译系统是用来开发软件的程序系列。翻译系统的一个关键部件是翻译器，这个程序能读入使用某种编程语言编写的程序，并输出使用不同编程语言编写的新程序。对翻译器而言，输入的是源程序(source program)，输出的是目标程序(target program)。源程序使用的语言称为源语言(source language)，与此相应，目标程序使用的语言称为目标语言(target language)。翻译器是按源语言和目标语言的类型分类的。

编译器(complier)是一种类型的翻译器。编译器处理用高级编程语言所写的程序并生成对象文件。使用编译器翻译高级语言程序的过程就叫做编译(compilation)。不同的高级语言(C，C＋＋，Java，C＃，Basic 等)所对应的编译器是不同的。

2. 应用软件

应用软件是用户利用计算机及其提供的系统软件，为解决实际问题而编制的计算机程序，由于计算机的日益普及，各行各业、各个领域的应用软件越来越多，常用的应用软件有以下几种：

(1) 信息管理软件，如 MIS(管理信息系统)；

(2) 办公自动化系统,如 OFFICE,WPS;
(3) 辅助设计软件以及辅助教学软件,如 AUTOCAD;
(4) 图像处理软件,如 PHOTOSHOP,COREDRAW;
(5) 数值处理软件,如 MATHLAB;
(6) 网络应用及影音处理软件,如 IE,RealPlayer 等。
用户可以根据需要解决的各种实际问题,选择不同的应用软件。

1.2 程序设计发展史

1.2.1 程序设计语言的进化

上节中我们告诉大家计算机是靠无数的逻辑电路开开合合来运行的,这些物理元器件根本就没有智慧,它们识别的只是高、低电位,用二进制的方法来表示,刚好就是 1 和 0。要想让计算机能够做我们想让它们做的事情,那只有发指令,编写这些指令的过程就是编写程序。指令的样式也不是一成不变,它就像计算机的硬件一样,随着时代而发展。

早期的计算机系统比较偏重于硬件设计,能够涉及程序方面的非常少。同时由于当时计算机硬件的技术水平不高,比如计算机的内存规模小、运算速度低等,造成了即使对于相同要解决的问题,人们也会针对不同的硬件和应用环境重新定制程序。这个时期的程序设计几乎没有统一的风格,编程就像是在制作一个艺术品。因此高德纳把这一时期的程序称之为艺术品。这个时期的编程语言主要表现为机器语言和汇编语言。

最早的程序就是用纯的二进制写的。这样的程序看上去就是一大串毫无生气的 0、1 字符。不要小看这些 0、1 码,它们的排列顺序代表着指令内容和所需要的数据。每一个程序员不仅要熟知这些指令格式,而且还要在执行每条指令前仔细地计算这条指令该被存放的内存地址。繁杂的机器代码程序很难被记忆,未受过专门训练的人不易掌握,这严重阻碍了计算机的应用和发展。

汇编语言的产生将程序员在使用机器语言中的这种负担解脱出来,再也不用记忆数百上千个用 0、1 码组成的指令,再也不用自己去计算内存地址了。复杂的指令代码在汇编语言中用诸如 load,store,add 和 sub 等等这样的符号替代。请大家注意,计算机真正认识的还只是 0,1 码,对于 load,store 这样的符号它是不知道的。所以在产生了汇编语言之后,编译器就随之产生。编译器的作用就是将汇编语言中的那些容易让人们记忆的符号翻译成计算机真正能看得懂得 0、1 码。从这个角度看,汇编语言实质上是机器语言的符号化形式。依旧属于面向机器的一种低级语言,其程序的通用性和可读性较差,仍需改进。

随着技术的进步,计算机的内存规模越来越大,运算速度也越来越快,所能处理的问题规模和范围变大,程序可读性、可重用性、可维护性等问题不断被提出,科学和工程计算对计算机的大量需求导致第一个高级程序设计语言 FORTRAN 的出现。

高级语言的出现,使得程序员在计算 x^2+y^2 这样的数学公式的时候,再也不用关心该使用哪个寄存器等等这些与硬件紧密相关的知识。在高级语言中,程序员只需要将上面数学公式中出现的 x,y 当作变量就可以了。例如,在 FORTRAN 语言中,我们把 x^2+y^2 这样的数学公式写成 x**2+y**2 就可以了。计算机系统会借助编译器把在高级语言中出现的表达

式翻译成计算机能够识别的二进制形式。

随着 FORTRAN 和其他高级语言的发展，到 20 世纪 60 年代，大的软件项目遭遇了不能按期完成、高成本开发和质量差等不好的名声。紧随其后的让人震惊的消息就是开发软件的成本远远高于运行这个软件的硬件产品的价格。在 IBM 管理 OS/360 软件程序开发的 Frederick P. Brooks，Jr. 就把大型系统的编程比喻成一个“焦油坑”，并声称“每个人都会惊奇地发现他们已被这个问题缠身，最终将会陷入到坑底”。而且程序员们确实对软件开发失去了必要的信心。大家开始寻找解决方案和查询问题根源，就在这个关键时刻，结构化编程的概念被提出，荷兰教授 Edsger W. Dijkstar 提出了“GOTO 语句是有害的”观点，指出程序的质量与程序中所包含的 GOTO 语句的数量成反比，认为应该在一切高级语言中取消 GOTO 语句。这一观点在计算机学术界激起了强烈的反响，引发了一场长达数年的广泛的论战，其直接结果是结构化程序设计方法的产生。同时 Corrado Bohm 和 Giuseppe Jacopini 也证明了只用 3 种基本的控制结构即顺序（A 执行完之后，执行 B，B 执行完之后，执行 C）、选择（要么执行 A，要么执行 B）、循环（反复执行 A 直到满足某个条件）就可以实现所有基本程序。由瑞士计算机科学家 Niklaus Wirth 开发的 Pascal 就是基于结构化程序设计方法的高级语言。该语言的简洁明了以及丰富的数据结构和 3 种基本控制结构，为程序员们提供了极大的方便性与灵活性，同时它特别适合微计算机系统，因此大受欢迎。

结构化程序设计发展还伴随着很多与之相配套的设计原理，比如自顶向下逐步求精，采用这种方式设计可以有效地将一个较复杂的系统设计任务分解成许多易于控制和处理的子程序，便于开发和维护。因此，结构化方法迅速走红，并在整个 20 世纪 70 年代的软件开发中占绝对统治地位。

随着计算机技术的迅猛发展，硬件成本不断降低，而软件成本却不断增加，因此，如何缩短软件生产周期和提高维护效率，研制出高质量的软件产品成为一个重要课题。同时到了 20 世纪 70 年代末期，计算机科学的应用领域也在不断扩大，各方面对计算机技术的要求越来越高。结构化程序设计语言和结构化分析与设计已无法满足用户需求的变化，于是面向对象技术开始浮出水面。

面向对象程序设计方法起源于 Simula 67 语言。它本身虽因为比较难学、难用而未能广泛流行，但在它的影响下所产生的面向对象技术却迅速传播开来，并在全世界掀起了一股 OO（Object Oriented，面向对象）热潮，至今盛行不衰。面向对象程序设计在软件开发领域引起了大的变革，极大地提高了软件开发的效率，为解决软件危机带来了一线光明。

面向对象程序设计建立在结构化程序设计基础上，最重要的改变是程序围绕被操作的数据来设计，而不是围绕操作本身。面向对象程序设计以类作为构造程序的基本单位，具有封装、数据抽象、继承、多态性等特点。

C++语言则是目前应用最广泛的面向对象程序设计语言之一。C++语言是对 C 语言的扩充（或称为 C 语言的超集），它继承了 C 语言高效、灵活的特点，完善了 C 语言的类型检查、代码重用、数据抽象机制，扩充了面向对象程序设计的支持。

下面我们就目前对程序影响最大的两种设计方法做一简单的阐述。

1.2.2　结构化程序设计

结构化程序设计的思想是在 20 世纪 60 年代末、70 年代初为解决“软件危机”的需要而形

成的，这一思想已被广泛接受。多年来的实践证明，结构化程序设计策略确实使程序执行效率提高，并且由于减少了程序的出错率，而大大减少维护费用。正因为如此，结构化程序设计这一名词的应用比比皆是，结构化 FORTRAN 、结构化 COBLE 等屡见不鲜。

结构化程序设计是按照一定的原则与原理，组织和编写正确且易读的程序的软件技术。结构化程序设计的目标在于使程序具有一个合理结构，以保证和验证程序的正确性，从而开发出正确、合理的程序。

结构化程序设计的提出对计算机科学产生了巨大影响。这里列举几点：

1. 对程序设计语言的影响

PASCAL 语言是基于结构化程序设计思想设计，并系统地体现了这一思想的第一个程序设计语言。这是程序设计语言发展的一个里程碑。结构化程序设计思想也深深地影响到其后的各种语言之设计。至于早先存在的语言则纷纷研制它们的结构化版本，形成所谓的结构化 FORTRAN 等。C 语言也是一个比较典型的结构化的程序设计语言。

2. 推动了程序设计方法学的发展

Software-Practice and Experience 杂志 1975 年第 1 期社论把结构化程序设计看作是在程序实践所依据的概念中的一场革命。结构化程序设计的提出让人们看到，与其他很多科学一样，程序设计也有自身的规律与原理，从而产生一种新的程序设计原理与方法，并促使人们进一步去探索与研究，特别是探索研究大规模程序设计的特点与规律，创造研制正确程序并证明其正确性的有效方法，进而实现程序自动构造的方法与工具。

3. 对计算机程序设计教学的影响

结构化程序设计的提出向传统的程序设计教学方法提出一个挑战，这就是说，讲授程序设计，不应是仅仅讲授一种程序设计语言，告诉学生可以用它写出怎样的程序，而是应该讲授怎样用它来编写易读易理解的程序。就是说不仅仅用仔细挑选的一些例子去解释阐明程序成分直到整个程序，教师应该传授一种系统的有纪律性的思考方式，从而能对程序进行组织和编码，使程序容易理解与修改，能保证其正确性。

结构化程序设计方法的主要技术是自顶向下、逐步求精。具体地说，就是在接受一个任务之后，纵观全局，先设想好整个任务分为几个子任务，每一个子任务又可以进行细分，直到不需要细分为止。这种方法就叫做“自顶向下、逐步求精”。比如设计房屋就是采用了自顶向下、逐步求精的方法。先进行整体规划，然后确定建筑物方案，再进行各部分结构的设计，最后进行细节的设计(如门窗、楼道、给排水等)。

采用这种方法考虑问题比较周全，结构清晰，层次分明。用这种方法也便于验证算法的正确性，在向下一层细分之前应检查本层设计是否正确，只有上一层是正确的才可以继续细分。如果每一层设计都没有问题，则整个算法就是正确的。由于每一层向下细分时都不太复杂，因此容易保证整个算法的正确性。检查时也是由上而下逐层检查，这样做思路清晰，有条不紊地一步一步进行，既严谨又方便。

我们应当掌握自顶向下、逐步求精的设计方法。这种设计方案的过程是将问题求解由抽象逐步具体化的过程。自顶向下是一种分解问题的技术，与控制结构无关；逐步求精是结构化程序的连续分解，最终使其成为顺序、选择和循环这三种基本控制结构的组合。结构化程序设计的结果是使一个结构化程序最终由若干个过程组成，每一过程完成一个确定的功能，并且每

一过程都是由顺序、选择和循环这 3 种基本控制结构组成。

结构化程序设计的主要特征与风格如下：

(1) 一个程序按结构化程序设计方式构造时，一般总是一个结构化程序，即由三种基本控制结构：顺序结构、选择结构和循环结构构成。这 3 种结构都是单入口/单出口的程序结构。已经证明，一个任意大且复杂的程序总能转换成这 3 种标准形式的组合。

(2) 有限制的使用 goto 语句。鉴于 goto 语句的存在使程序的静态书写、顺序与动态执行顺序十分不一致，导致程序难读难理解，容易存在潜在的错误，难于证明正确性，有人主张程序中禁止使用 goto 语句。但有人则认为 goto 语句是一种有效设施，不应全盘否定而完全禁止使用。结构程序设计并不在于是否使用 goto 语句，因此作为一种折衷，允许在程序中有限制地使用 goto 语句。

(3) 往往借助于体现结构化程序设计思想的所谓结构化程序设计语言来书写结构化程序，并采用一定的书写格式以提高程序结构的清晰性，增进程序的易读性。对于一些早期非结构化程序设计语言，为了有助于结构化程序设计，往往扩充以结构化方案。

(4) 它强调了程序设计过程中人的思维方式与规律，是一种自顶向下的程序设计策略，它通过一组规则、纪律与特有的风格对程序设计细分和组织。对于小规模程序设计，它与逐步精化的设计策略相联系，即采用自顶向下、逐步求精的方法对其进行分析和设计；对于大规模程序设计，它则与模块化程序设计策略相结合，即将一个大规模的问题划分为几个模块，每一个模块完成一定的功能。

1.2.3　面向对象程序设计

结构化程序设计方法，可以使得程序的结构很好，如 FORTRAN，C 和 PASCAL，也称为非面向对象的过程语言，其数据结构是解决问题的中心。一个软件系统的结构是围绕一个或几个关键数据结构为核心而组成的。在这种情况下，软件开发一直被两大问题所困扰：一是如何超越程序的复杂性障碍；二是计算机系统中如何自然地表示客观世界。诚然，Niklans Wirth 提出的“算法＋数据结构＝程序设计”，在软件开发进程中产生了深远的影响，但软件系统的规模越来越大，复杂性不断增长，以致不得不对“关键数据结构”重新评价。在这种系统中，许多重要的过程和函数(子程序)的实现严格依赖于关键数据结构，如果这些关键数据结构的一个或几个数据有所改变，则涉及到整个软件系统，许多过程和函数必须重写，甚至因为几个关键数据结构改变，导致软件系统的彻底崩溃。因此，往往是用效率的降低来换取程序的可读性。

上世纪 80 年代，计算机科学家为提高软件生产率所作的种种探讨和努力，或多或少与面向对象的程序设计这一思想有关联。作为克服复杂性的手段，在面向对象程序设计中，把密切相关的数据与过程定义为一个整体(即对象)，而且一旦作为一个整体定义了之后，就可以使用它，而无需了解其内部的实现细节。面向对象系统的构造不依赖于面向对象的内部结构，而依赖于定义在对象内部数据的方法。采用面向对象的方法设计出来的软件能够更加直接地反映现实中的问题。

面向对象程序设计的基本思想是，将要构造的软件系统描述为对象集，而其中的每个对象都是将其数据和操作封装在一起，对象间通过消息传递来联系。围绕几个主要概念：类，类层次(子类)，继承性和多态性。类和继承是符合人们一般思维方式的描述模式。