HZ BOOKS
华章教育

计 算 机 基 础 课 程 系 列 教 材

Visual Basic 程序设计教程

邹 晓 主编
曹来成 廖成斌 徐志刚 参编

本书为教师
配有电子课件

机 械 工 业 出 版 社
China Machine Press

计 算 机 基 础 课 程 系 列 教 材

Visual Basic 程序设计教程

邹 晓 主编
曹来成 廖成斌 徐志刚 参编

机械工业出版社
China Machine Press

本书从初学者角度出发，通过大量实例，深入浅出地介绍了Visual Basic程序设计的相关知识。主要内容包括：Visual Basic程序设计概述、简单的Visual Basic程序设计、Visual Basic程序设计语言基础、程序的基本结构、数组、过程、常用控件、用户界面设计、图形程序与多媒体程序设计和数据库技术，并根据每一部分知识的重点和难点给出了相应的思考题。同时，根据多年教学经验，有针对性地编写了习题集与上机指导供读者课后上机练习以巩固所学知识。

本书可作为高等院校Visual Basic程序设计课程的教材，也可作为计算机培训班的教材以及全国计算机等级考试的应试教材，亦可供Visual Basic 爱好者自学使用。

图书在版编目（CIP）数据

Visual Basic程序设计教程/邹晓主编. —北京：机械工业出版社，2008.12
（计算机基础课程系列教材）

ISBN 978-7-111-25530-7

Ⅰ. V… Ⅱ. 邹… Ⅲ. BASIC语言—程序设计—教材 Ⅳ. TP312

中国版本图书馆CIP数据核字（2008）第193372号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）
责任编辑：杨庆燕
北京慧美印刷有限公司印刷
2009年1月第1版第1次印刷
184mm × 260mm · 18.5印张
标准书号：ISBN 978-7-111-25530-7
定价：32.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换
本社购书热线：（010）68326294

前　言

Visual Basic是Microsoft公司推出的可视化编程语言，自1991年问世以来，由于其语法简练、功能强大、采用结构化程序设计方法以及方便快捷的可视化编程手段，使得编写Windows环境下的应用程序变得非常容易，因而深受广大程序设计人员的青睐。目前，Visual Basic已经成为许多高校必选的教学用程序设计语言。

Visual Basic程序设计语言课程的教学主要包括两个方面，即程序设计语言和可视化界面设计。程序设计语言介绍Visual Basic的基本知识、基本语法、编程方法和常用算法，通过这部分学习，可以培养学生分析问题、解决问题的能力，这是Visual Basic程序设计语言课程的重点和难点；可视化界面设计是实际应用当中不可缺少的，由于用户界面可以直接在屏幕上呈现出来，因此Visual Basic的界面设计比较容易掌握和理解。

本书围绕以上两个方面，以Visual Basic 6.0 中文版为背景，从初学者角度出发，通过大量实例，深入浅出地介绍了Visual Basic程序设计的相关知识。主要内容包括：绪论、Visual Basic程序设计概述、简单的Visual Basic程序设计、Visual Basic程序设计语言基础、程序的基本结构、数组、过程、常用控件、用户界面设计、文件、图形程序与多媒体程序设计和数据库技术，并根据每一部分知识的重点和难点，给出了相应的思考题。同时，我们还编写了配套的《Visual Basic程序设计习题集与上机指导》一书，供读者在学习、练习和上机实践时使用。

本书由从事Visual Basic教学的一线老师编写，全书共分12章，其中第1章、第9章和第12章由邹晓编写；第2章、第5章和第11章由徐志刚编写；第3章、第7章和第10章由曹来成编写；第4章、第6章和第8章由廖成斌编写。全书由邹晓主编、统稿。本书得到了兰州理工大学计算机与通信学院、教务处等部门的领导和相关老师的大力支持和协助，在此表示衷心地感谢。

由于编者水平有限，书中难免会出现一些错误和不足之处，敬请专家和广大读者批评指正。

编　者

2008年10月

教师服务登记表

尊敬的老师：

您好！感谢您购买我们出版的 _____ 教材。

机械工业出版社华章公司本着为服务高等教育的出版原则，为进一步加强与高校教师的联系与沟通，更好地为高校教师服务，特制此表，请您填妥后发回给我们，我们将定期向您寄送华章公司最新的图书出版信息。为您的教材、论著或译著的出版提供可能的帮助。欢迎您对我们的教材和服务提出宝贵的意见，感谢您的大力支持与帮助！

个人资料（请用正楷完整填写）

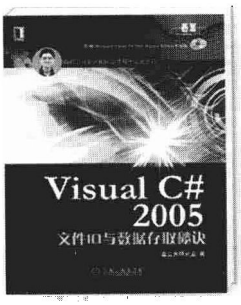
<table>
<tr><td>教师姓名</td><td></td><td>□先生
□女士</td><td>出生年月</td><td></td><td>职务</td><td></td><td colspan="2">职称：□教授 □副教授
□讲师 □助教 □其他</td></tr>
<tr><td>学校</td><td colspan="2"></td><td>学院</td><td colspan="2"></td><td>系别</td><td colspan="2"></td></tr>
<tr><td rowspan="2">联系电话</td><td colspan="3" rowspan="2">办公：
宅电：
移动：</td><td>联系地址及邮编</td><td colspan="3"></td></tr>
<tr><td>E-mail</td><td colspan="3"></td></tr>
<tr><td>学历</td><td></td><td>毕业院校</td><td colspan="2"></td><td>国外进修及讲学经历</td><td colspan="3"></td></tr>
<tr><td>研究领域</td><td colspan="8"></td></tr>
<tr><td colspan="2">主讲课程</td><td colspan="3">现用教材名</td><td>作者及出版社</td><td>共同授课教师</td><td colspan="2">教材满意度</td></tr>
<tr><td colspan="2">课程：
□专 □本 □研
人数： 学期：□春□秋</td><td colspan="3"></td><td></td><td></td><td colspan="2">□满意 □一般
□不满意 □希望更换</td></tr>
<tr><td colspan="2">课程：
□专 □本 □研
人数： 学期：□春□秋</td><td colspan="3"></td><td></td><td></td><td colspan="2">□满意 □一般
□不满意 □希望更换</td></tr>
<tr><td colspan="9">样书申请</td></tr>
<tr><td>已出版著作</td><td colspan="3"></td><td>已出版译作</td><td colspan="4"></td></tr>
<tr><td colspan="4">是否愿意从事翻译/著作工作 □是 □否</td><td>方向</td><td colspan="4"></td></tr>
<tr><td>意见和建议</td><td colspan="8"></td></tr>
</table>

填妥后请选择以下任何一种方式将此表返回：（如方便请赐名片）

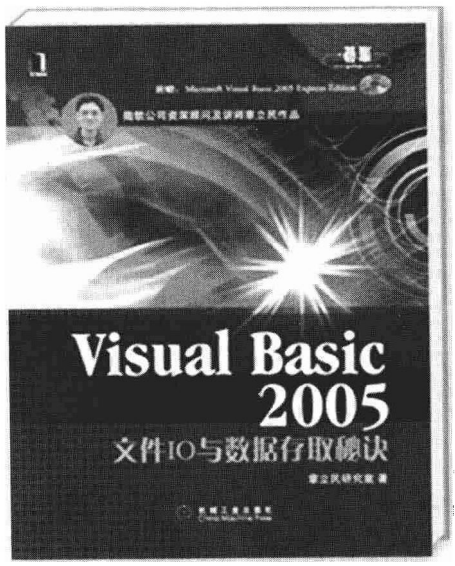
地 址：北京市西城区百万庄南街1号 华章公司营销中心 邮编：100037

电 话：(010) 68353079 88378995 传真：(010)68995260

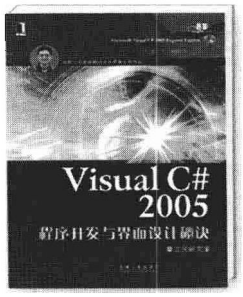
E-mail:hzedu@hzbook.com markerting@hzbook.com 图书详情可登录http://www.hzbook.com网站查询



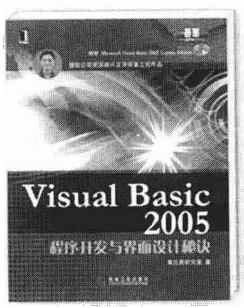
Visual C# 2005
文件IO与数据存取秘诀
书号：978-7-111-19972-4
定价：79.00元
作者：**章立民研究室**



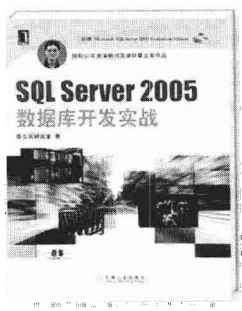
Visual Basic 2005
文件IO与数据存取秘诀
书号：978-7-111-19973-1
定价：75.00元
作者：**章立民研究室**



Visual C# 2005
程序开发与界面设计秘诀
书号：978-7-111-19947-2
定价：78.00元
作者：**章立民研究室**



Visual Basic 2005
程序开发与界面设计秘诀
书号：978-7-111-19918-9
定价：79.00元
作者：**章立民研究室**



SQL Server 2005
数据库开发实战
书号：978-7-111-19974-X
定价：79.00元
作者：**章立民研究室**

目　录

第1章 绪 论

当今社会是信息社会，信息社会的灵魂是作为“信息处理机”的计算机。自从1946年世界上第一台电子计算机ENIAC问世到今天，计算机的硬件得到了突飞猛进的发展，计算机的程序设计语言和程序设计方法也随之不断发展。

1.1 程序设计语言

人类自然语言（如汉语、英语等）是人们交流和表达思想的工具。那么，人与计算机如何“交流”呢？为此，产生了计算机语言。所谓计算机的程序设计语言（又称为计算机语言或编程语言）就是人为规定的、编程人员应遵守的、计算机可以识别的程序代码规则，是人指挥计算机进行工作、与计算机进行交流的工具。

计算机的程序设计语言是随着计算机技术的进步而不断发展的，纵观其历史，可以将其分为机器语言、汇编语言和高级语言3类。

1.1.1 机器语言

机器语言是由0和1二进制代码按一定规则组成的、能被计算机直接理解和执行的指令集合。机器语言中的每一条机器指令实际上是一条二进制形式的指令代码。在指令代码中一般包括操作码和操作数两部分，操作码告诉计算机做何种操作；操作数则指出参与操作的数本身或它在内存中的地址。例如，计算11+18的机器语言程序如表1-1所示。

表1-1 机器语言程序举例

指令序号	机 器 指 令	指 令 功 能
1	10110000 00001011	把加数11放入累加器AL中
2	00101100 00010010	把累加器AL中的值与18相加，结果仍放入AL中
3	11110100	结束，停机

从表1-1中可以看出，用机器语言编写的程序表现为一系列的二进制信息，编写程序的难度大，而且难学、难记、难修改、难检查，只适合专业人员使用；又由于不同的计算机其指令系统不同，因此机器语言随机而异，通用性差，是一种面向机器的语言。

由于机器语言程序是直接针对计算机硬件的，计算机硬件可以直接识别，因此它的优点是执行效率比较高，能充分发挥计算机的速度性能。

1.1.2 汇编语言

为了克服机器语言的缺点，出现了用符号来表示二进制指令代码的符号语言，称为汇编语言。汇编语言用容易记忆的英文单词缩写（称为指令助记符）代替约定的指令。例如，用MOV表示数据的传送指令，用ADD表示加法指令，用SUB表示减法指令等。上述计算11+18的汇编语言程序如表1-2所示。

从表1-2中可以看出，汇编语言与机器语言是一一对应的，因此，汇编语言是与具体使用的计算机有关的。由于汇编语言采用了助记符，因此，它比机器语言直观、容易理解和记忆，用汇

编语言编写的程序也比机器语言程序易读、易检查、易修改。但是，汇编语言程序不能在计算机上直接运行，必须经过翻译，转化为机器语言代码后才能在计算机上运行，这个过程是通过一个翻译程序自动完成的。将汇编语言源程序翻译成机器语言程序的翻译程序称为汇编程序，如图1-1所示。

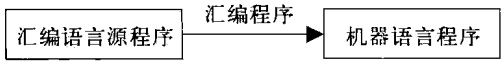


图1-1 汇编程序的作用

表1-2 汇编语言程序举例

指令序号	机器指令	指令功能
1	MOV AL，11	把加数11放入累加器AL中
2	ADD AL，18	把累加器AL中的值与18相加，结果仍放入AL中
3	HLT	结束，停机

1.1.3 高级语言

机器语言和汇编语言都是面向机器的语言，一般称为低级语言。使用这类语言，可以编出效率极高的程序，但它们对机器的依赖性太大，用它们开发出的程序通用性差，对程序设计人员的要求很高，普通的计算机用户很难胜任这一工作。高级语言是接近于自然语言或数学语言的计算机语言，它与具体的计算机硬件无关，其表达方式接近于被描述的问题，易被人们接受和掌握。例如，上述计算11+18的Visual Basic语言程序如表1-3所示。

表1-3 高级语言程序举例

Visual Basic语言程序	注　释
Private Sub Form_Load()	窗体的Load事件
A=11+18	11与18相加的结果放入变量A中
Print A	输出变量A的值
End Sub	结束Load事件过程

用高级语言编写的程序计算机不能直接执行，必须经过语言处理程序翻译后才能被计算机执行。翻译过程分成两步，即编译和连接，如图1-2所示。

高级语言的特点是独立于具体的计算机硬件，通用性强和可移植性好。利用高级语言编写程序，编程者不需要掌握过多的计算机专业知识，特别适合于非计算机专业的技术人员利用计算机技术解决本专业的问题。高级语言的产生，大大扩展了计算机的应用范围，推动了各行各业的发展。

高级语言可分为两种类型：面向过程的高级语言和面向对象的高级语言。

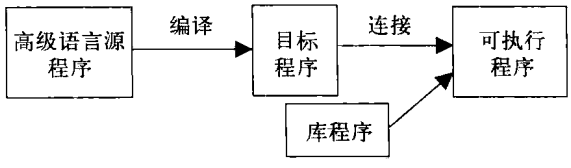


图1-2 编译、连接过程

1. 面向过程的高级语言

面向过程的高级语言是接近于数学语言的计算机语言。利用面向过程的高级语言设计程序，完成一定的任务，无论任务简单还是复杂，都必须将具体的步骤描述清楚。例如，利用高级语言编写程序完成两个整数相加的工作，必须描述以下步骤：

1）定义3个变量x、y和z分别用来存放被加数、加数与和；

2）将被加数、加数分别输入到变量x、y中；

3）计算$x+y$的值，并将结果存入变量z中；

4）输出变量z的值；

5）程序结束。

完成某项任务的具体步骤通常称为算法，所以面向过程的高级语言也称为算法语言。

常用的面向过程的高级语言有FORTRAN、BASIC、PASCAL、C等。

2. 面向对象的高级语言

面向过程的高级语言过分强调求解过程的细节，程序不易重复使用。为此，在20世纪80年代推出了面向对象的高级语言。面向对象的高级语言与面向过程的高级语言的根本不同点在于，面向对象的高级语言设计的出发点就是为了能更直接地描述客观世界中存在的事物（即对象）以及它们之间的关系。

面向对象的高级语言将客观事物看作具有属性和行为的对象，通过抽象找出同一类对象的共同属性和行为，形成类。通过类的继承与多态可以很方便地实现代码重用，这大大提高了程序的复用能力和程序的开发效率。

常用的面向对象的高级语言有Visual Basic、Java、C++等。

1.2 算法

为了有效地进行程序设计，不仅要掌握一门程序设计语言，还应该学会针对各类问题拟定有效的解题方法和步骤，即进行算法设计。有了正确的算法才能够编制程序。算法的好坏决定了程序的优劣，因此，程序设计的核心任务之一就是设计算法。

1.2.1 算法的概念

算法是对解决某一特定问题的求解步骤的详细描述。广义地说，算法就是解决某个问题的方法和步骤。从计算机应用的角度来说，算法是用于求解某个问题的一些指令的集合。具体地说，我们用计算机所能实现的操作或指令来描述问题的求解过程，就得到了这一问题的计算机算法。

计算机算法可以分为两大类：数值计算算法和非数值计算算法。数值计算算法的目的是求数值解，其特点是输入、输出少，运算复杂，如求方程的根、求函数的定积分等；非数值计算算法的目的是处理数据，其特点是输入、输出多，运算简单，例如，对数据的排序、查找等算法。

1.2.2 算法的特征

算法应该具备以下5个特征：

- 有穷性。算法必须在执行有穷步之后结束。
- 确定性。算法的每一步骤都必须准确定义，而不能是含混的或模棱两可的。
- 可执行性。首先，每个算法所实现的动作应该是可付诸实施的。例如，进行除法运算B/A时，A不能为零，否则无法执行。其次，算法执行后应该能达到预期的目的。
- 输入。算法总是要施加到运算对象上的，输入说明了运算对象的初始情况。输入是算法的起点，所以一个算法要有零个或多个输入，零个输入是指在算法中直接给定初始条件。例如，要比较a和b的大小，首先要输入数据a和b的值。如果仅比较10和9的大小，则数据已经确定，不再需要输入数据。
- 输出。一个算法执行完毕后要有一个或多个输出。没有输出的算法是毫无意义的，由输出可以判断算法的正确性，并且可以得到人们想要的结果。

1.2.3 算法的表示

为了描述算法，可以使用多种方法。常用的方法有自然语言、传统流程图、N-S流程图，伪代码和计算机语言等。

1. 自然语言

可以用人们使用的语言，即自然语言描述算法。

例如：交换两个变量A和B的值。

分析：交换有直接交换和间接交换两种方式。比如，两个人交换座位，只要各自去坐对方的位置即可，这种交换就是直接交换。再比如，一杯咖啡和一杯茶互换，就不能直接从一个杯子倒入另一个杯子，必须先借助于一个空杯子，先把咖啡倒入空杯，再将茶倒入已空的咖啡杯，最后将咖啡倒入已空的茶杯，这样才能实现咖啡和茶的交换，这种交换就是间接交换。

由于计算内存有取之不尽、一冲就走的特点，因此计算机交换两个变量的值只能采用间接交换的方法。故此问题中需要引入一个中间变量C。

解决此问题的算法表示如下：

步骤1：将A的值存入中间变量C 中，A→C；

步骤2：将B的值存入变量A中，B→A；

步骤3：将中间变量C的值存入B中，C→B。

用自然语言表示算法通俗易懂，但文字冗长，容易产生歧义，因为同一段文字，不同的人会有不同的理解。因此，除了简单的问题外，一般不用自然语言描述算法。

2. 传统流程图

传统流程图用一些图框、线条以及文字说明来形象、直观地描述算法。美国国家标准化协会（American National Standard Institute，ANSI）规定了一些常用的流程图符号，如表1-4所示。图1-3为交换两个变量值的传统流程图。

表1-4 流程图符号

符 号	功 能
	表示程序的开始或结束，即端点符
	表示一般的处理功能，例如赋值等
	表示条件选择
	表示输入或输出
	流程线，表示流程的路径与方向
	连接点
	表示以文件方式进行输入输出
	表示调用或引用
	表示准备

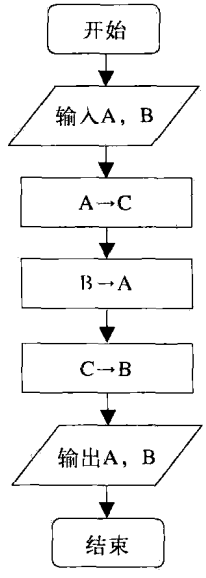


图1-3 交换两个变量值的传统流程图

传统流程图形象、直观，便于交流，是目前广泛使用的描述方法。

3. N-S流程图

传统流程图虽然形象、直观，但对流程线的使用没做限制。使用者可以不受限制地随意转移流程，使流程变得毫无规律，难以阅读和维护。为此，人们对流程图进行了改进。1973年，美国学者I.Nassit和B.Shneideman提出了一种新的流程图形式，并以他们姓名的第一个字母命名，这种流程图称为N-S流程图。在N-S流程图中，完全去掉了带箭头的流程线，全部算法写在一个大矩形框中，在该大矩形框中还可以包含一些从属于它的小矩形框，这种流程图特别适合于结构化程序设计。例如，用N-S流程图表示的交换两个变量值的算法如

图1-4所示。

4. 伪代码

由于绘制流程图较费时，而自然语言容易产生歧义且难以清楚地表达算法的逻辑流程，因此可采用伪代码。伪代码产生于20世纪70年代，它用介于自然语言和计算机语言之间的文字和符号来描述算法。“伪”意味着假，因此，伪代码是一种假的代码——不能被计算机所理解，但接近于某种语言编写的程序，便于转换成编程语言。

例如，交换两个变量的值的Visual Basic伪代码表示如下：

```
Proc Exchange
   Input  A, B
   A→C
   B→A
   C→B
   Print A,B
End
```

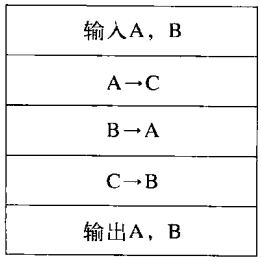


图1-4　交换两个变量值的N-S流程图

5. 计算机语言

自然语言、流程图、伪代码只是为了帮助人们描述、理解算法，而无法被计算机识别。要用计算机解题，就必须用计算机语言来描述算法。只有用计算机机器语言编写的程序才能被计算机执行。因此，最终还是要将它转换成计算机语言程序。

用计算机语言描述算法必须严格遵循所选择的编程语言的语法规则。

1.3　程序设计方法

使用某种计算机语言，按照某种算法编写程序的活动，称为程序设计。程序设计需要一定的方法来指导，以提高程序的扩充性、重用性、可读性、稳定性以及编程效率。目前有两种重要的程序设计方法，即面向过程的结构化程序设计方法和面向对象的程序设计方法。

1.3.1　结构化程序设计方法

结构化程序设计（structured programming）的概念最早是由荷兰科学家E.W.Dijkstra提出的。结构化程序设计方法要求程序设计者按照一定的结构形式来设计和编写程序，而不能随心所欲地把流程从程序的一处转到另一处。结构化程序设计的定义方法较多，曾经有一段时间不少人误认为没有GOTO语句的程序就是结构化程序，但在众多的定义中，有一个共同的特征，那就是结构化程序有顺序、选择和循环3种基本控制结构和程序块只具有“一个入口和一个出口”的原则。

1. 结构化程序设计的3种基本结构

- 顺序结构。顺序结构是最简单、最基本的一种结构，计算机在执行顺序结构程序时，按语句出现的先后顺序依次执行，图1-5a和图1-5b分别是用传统流程图和N-S流程图表示的顺序结构。其中，A和B表示操作步骤，计算机先执行A操作，再执行B操作。
- 选择结构。当程序在执行过程中需要根据某种条件的成立与否有选择地执行一些操作时，就需要使用选择结构。图1-6a和图1-6b分别是用传统流程图和N-S流程图表示的选择结构。当条件成立（称为“真”）时执行A，否则执行B。从图1-6a中可以看出，无论执行A或B，都将汇合在一起，然后出口。

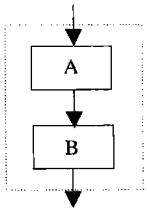


图1-5a) 顺序结构的传统流程图

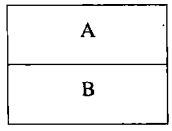


图1-5b) 顺序结构的N-S流程图

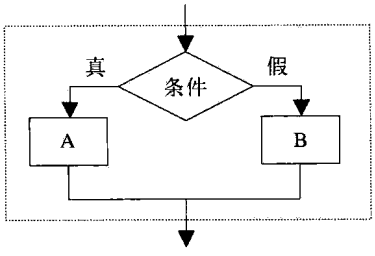


图1-6a) 选择结构的传统流程图

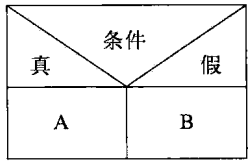


图1-6b) 选择结构的N-S流程图

- 循环结构。程序的循环就是反复执行一些步骤，直到符合某种条件时才停止。循环有两种结构：当型循环和直到型循环。
  - 当型循环结构如图1-7a和图1-7b所示。这种循环当条件成立（真）时，反复执行A操作，直到条件为“假”时才停止循环。

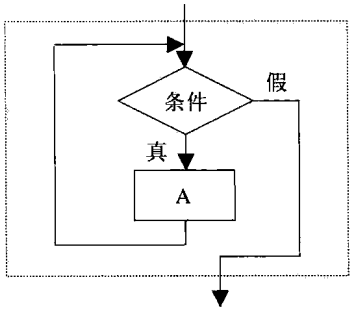


图1-7a) 当型循环结构的传统流程图

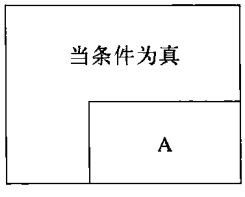


图1-7b) 当型循环结构的N-S流程图

  - 直到型循环结构如图1-8a和图1-8b所示。这种循环先执行A操作，再判断条件是否为“假”，若条件为“假”，再执行A，如此反复，直到条件为“真”为止。

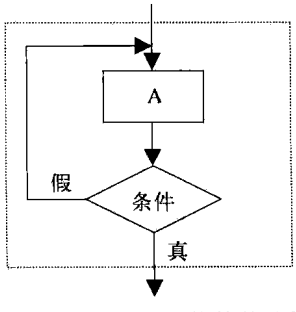


图1-8a) 直到型循环结构的传统流程图

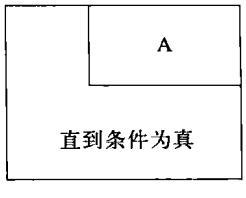


图1-8b) 直到型循环结构的N-S流程图

当型循环结构和直到型循环结构的区别是：当型循环是先判断（条件）后执行，而直到型循环是先执行再判断。

2. 结构化程序设计方法

结构化程序设计方法的基本思路是，把一个复杂问题的求解过程分阶段进行，每个阶段处理的问题都控制在人们容易理解和处理的范围内。

结构化程序设计方法是面向过程的，程序由模块（module）构成，一个模块就是一个过程。每一模块均是由顺序、选择和循环这3种基本结构组成。模块之间的信息传递主要通过模块的接口（interface）来实现，这一机制隐藏了模块的内部细节，增强了模块的独立性。

结构化程序设计方法强调程序结构的规范性，强调程序设计的自顶向下、逐步求精的演化过程。在这种方法中，待解决的问题和程序设计语言中的子过程紧密相联。例如，要开发一个学生成绩管理系统，由于问题较复杂，可以将待解决的问题分解成若干子问题：

- 输入成绩。
- 处理成绩。
- 打印成绩。

每个子问题对应程序设计语言中的一个子过程。如果用C语言来解决上述问题，则待解决的问题将对应main( )函数，每个子问题对应main( )的调用函数。当然，每个子问题还可以继续分解，直到每个子问题都足够简单为止，相应的子过程也很容易处理。

可见，这种方法着眼于系统要实现的功能，从系统的输入和输出出发，分析系统要做哪些事情，以及如何做这些事情，自顶向下地对系统的功能进行分解，建立系统的功能结构和相应的程序模块结构，有效地将一个较复杂的程序设计任务分解成许多易于控制和处理的子任务，便于开发和维护。

到今天，结构化程序设计已无处不在，几乎每种程序设计语言都具备支持结构化程序设计的机制。然而，随着程序规模的扩大与复杂性的增加，面向过程的结构化程序设计方法的不足日益暴露出来。首先是数据安全性问题。例如，在上述成绩管理系统中，成绩数据被每个模块共用，因此是不安全的，一旦出错，很难查明原因；其次，可维护性及可重用性差，它把数据结构和算法分离为相互独立的实体。一旦数据结构需要改变时，常常涉及整个程序，修改工作量大并容易产生新的错误。每一种针对老问题的新方法都要带来额外的开销。另外，图形用户界面的应用程序很难用过程来描述和实现，而且开发和维护也很困难，于是面向对象的程序设计方法便应运而生了。

1.3.2　面向对象程序设计方法

面向对象程序设计方法（Object Oriented Programming，OOP）源于20世纪70年代中后期，起源于Smalltalk语言。用面向对象的方法解决问题，不是将问题分解为过程，而是将问题分解为对象。对象是现实世界中可以独立存在、可以区分的实体，也可以是一些概念上的实体，世界是由众多对象组成的。对象有自己的数据（属性），也有作用于数据的操作（方法），将对象的属性和方法封装成一个整体，供程序设计者调用。对象之间的相互作用通过消息传递来实现。目前，这种“对象+消息”的面向对象程序设计模式有取代“数据结构+算法”的面向过程的程序设计模式的趋势。当然，面向对象的程序设计并不是要抛弃结构化的程序设计方法，而是站在比结构化程序设计更高、更抽象的层次上解决问题。当所要解决的问题被分解为底层代码模块时，仍需要结构化的编程方法和技巧。结构化的分解突出过程，即如何做（How to do），它强调代码的功能是如何得以实现的。面向对象的分解突出真实世界和抽象的对象，即做什么（What to do），它将大量的工作分配给相应的对象完成，程序员在程序设计时只需说明要求对象完成的任务。

面向对象的程序设计方法符合人们习惯的思维方式，便于分析复杂而多变的问题；易于软件的维护和功能的增减；可重用性好，能用继承的方式缩短程序开发所花的时间；与可视