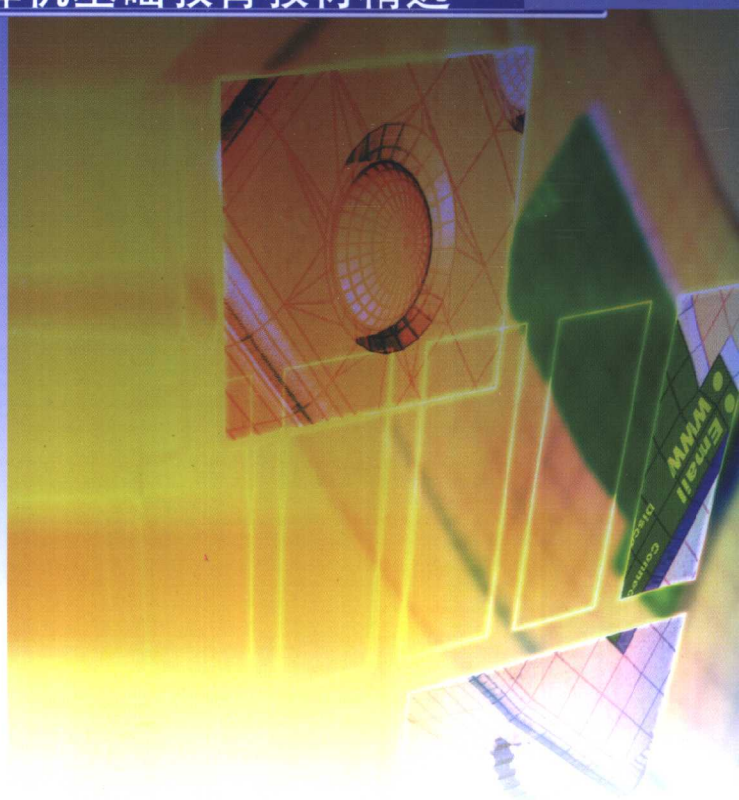
高等学校计算机基础教育教材精选

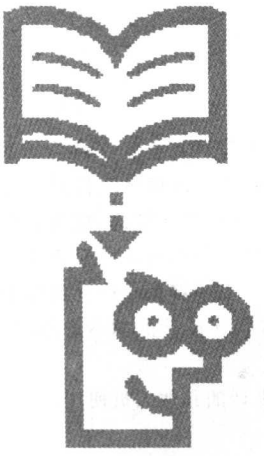


姚普选 编著

程序设计教程(Delphi)

清华大学出版社

高等学校计算机基础教育教材精选



程序设计教程(Delphi)

姚普选 编著

清华大学出版社
北京

MJ599/05

内 容 简 介

本书以 Delphi 7 为工具，结合大量具有应用价值的实例，循序渐进地介绍了计算机程序设计的基本思想和常用的程序设计方法，并简明扼要地介绍了编程时经常涉及的算法概念和计算机软件系统的工作机理等方面的知识，主要内容包括程序设计的基础知识、算法设计、数据类型、程序结构、复杂数据类型、面向对象程序设计、用户界面设计、图处理与多媒体播放、数据库应用程序。书中各章均配有习题。

本书讲求文字的准确性、思想的连贯性、方法的实用性和内容的先进性。书中将程序设计过程中所涉及的多方面的知识有机地融合在一起，力求使读者在有限的时间内，理解程序设计的基本思想，掌握程序设计的基本方法。

本书适合作为高等学校计算机程序设计课程的教材，也可供学习程序设计的其他人员使用。

版权所有，翻印必究。

本书封面贴有清华大学出版社激光防伪标签，无标签者不得销售。

图书在版编目(CIP)数据

程序设计教程(Delphi)/姚普选编著. —北京：清华大学出版社，2004

(高等学校计算机基础教育教材精选)

ISBN 7-302-08028-3

Ⅰ. 程… Ⅱ. 姚… Ⅲ. 软件工具—程序设计—高等学校—教材 Ⅳ. TP311.56

中国版本图书馆 CIP 数据核字(2004)第 006328 号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦

http://www.tup.com.cn　　**邮　　编**：100084

社 总 机：010-62770175　　**客户服务**：010-62776969

组稿编辑：焦　虹

文稿编辑：张为民

印 装 者：北京鑫霸印务有限公司

发 行 者：新华书店总店北京发行所

开　　本：185×260　**印张**：24　**字数**：548千字

版　　次：2004 年 3 月第 1 版　2004 年 3 月第 1 次印刷

书　　号：ISBN 7-302-08028-3/TP・5812

印　　数：1～5000

定　　价：30.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：(010)62770175-3103 或(010)62795704

出版说明

高等学校计算机基础教育教材精选

在教育部关于高等学校计算机基础教育三层次方案的指导下，我国高等学校的计算机基础教育事业蓬勃发展。经过多年的教学改革与实践，全国很多学校在计算机基础教育这一领域中积累了大量宝贵的经验，取得了许多可喜的成果。

随着科教兴国战略的实施以及社会信息化进程的加快，目前我国的高等教育事业正面临着新的发展机遇，但同时也必须面对新的挑战。这些都对高等学校的计算机基础教育提出了更高的要求。为了适应教学改革的需要，进一步推动我国高等学校计算机基础教育事业的发展，我们在全国各高等学校精心挖掘和遴选了一批经过教学实践检验的优秀的教学成果，编辑出版了这套教材。教材的选题范围涵盖了计算机基础教育的三个层次，面向各高校开设的计算机必修课、选修课以及与各类专业相结合的计算机课程。

为了保证出版质量，同时更好地适应教学需求，本套教材将采取开放的体系和滚动出版的方式（即成熟一本，出版一本，并保持不断更新），坚持宁缺勿滥的原则，力求反映我国高等学校计算机基础教育的最新成果，使本套丛书无论在技术质量上，还是出版质量上均成为真正的“精选”。

清华大学出版社一直致力于计算机教育用书的出版工作，在计算机基础教育领域出版了许多优秀的教材。本套教材的出版将进一步丰富和扩大我社在这一领域的选题范围、层次和深度，以适应高校计算机基础教育课程层次化、多样化的发展趋势，从而更好地满足各学校由于条件、师资和生源水平、专业领域等的差异而产生的不同需求。我们热切期望全国广大教师能够积极参与到本套丛书的编写工作中来，把自己的教学成果与全国的同行们分享；同时也欢迎广大读者对本套教材提出宝贵意见，以便我们改进工作，为读者提供更好的服务。

我们的电子邮件地址：jiaoh@tup.tsinghua.edu.cn。联系人：焦虹。

清华大学出版社

2001年8月

前言

计算机程序设计是计算机基础教育的重点和基础。作为高等学校的学生，尤其是将来要成为工程技术人员的理工科学生，不能只满足于使用别人设计好的软件，而应该具有一定的程序设计能力。

实际上，程序设计并不是一件容易的事情。要编程序解决一个实际问题，首先要理解问题本身的逻辑结构和工作方式，然后再考虑选用哪种程序设计语言和工具。而且在程序设计的过程中，还要考虑问题在计算机中如何表示，应该采用哪种算法，按照什么方法和步骤来编制和调试程序等。这就涉及了程序设计语言的语法规则、程序设计工具的使用、算法设计策略、数据结构常识、程序设计的技巧等多方面的知识和技能。要使学生在有限的时间内掌握程序设计技术，首先就要有能够将这些知识和技能有机地融合在一起的教材。

多年来，围绕着计算机程序设计课程，出现了一大批各具特色的教材和参考书，其中不乏立意深刻、内容丰富的好教材。但遗憾的是，当程序设计发展到了面向对象设计方法和可视化设计环境之后，需要解决的实际问题的范围越来越广，程序设计的工具越来越复杂，程序设计手段更加丰富多彩，而现有的教材和参考书大都有所侧重，或者详细地介绍程序设计语言和工具的使用方法，或者系统地论述算法设计的知识和技能，或者深入地探讨各种程序设计方法的优劣，程序设计课程所需要的语言的知识、工具的使用技能、数据结构的知识以及程序设计方法等，散见于各种不同类型的书籍中，这就给教材的选用和学生的学习带来了一定的困难。有鉴于此，作者结合多年的教学实践，编写了本书。

本书选择了核心的程序设计技术、常用的算法设计策略以及 Delphi 7 开发环境的常用功能，由浅入深地进行了详细的讲解，力图使读者在有限的时间内，对课程的相关知识有一个清晰和完整的理解。本书中对重要的概念和方法一般都先在例题或讲解中以极易理解的简单方式加以运用，然后在其后的某个章节集中讲解。这样既起到了分散难点的作用，便于初学者理解和掌握，又避免了因刻意分散难点而将相关内容割裂开来，造成不便查阅，不易形成完整印象的弊病。同时，本书注重知识和技能的合理调配，力图避免因强调某些方面，忽略其他方面所造成的程序设计能力整体上的缺失。

本书可作为高等学校程序设计课程的教材，也可作为程序设计工作者的参考书。采用本书作为教材的程序设计课程以 64～72(包括上机时数)学时为宜。带 * 的章节为选

此为试读，需要完整PDF请访问：www.ertongbook.com

学内容，如果学时数较少，对这部分内容可以不学。本书每章都配备了内容丰富的习题，不同类型的读者可根据自己的实际情况选做部分习题。

程序设计技术博大精深，其内容绝非一本书所能包括。由于本书的编写不可避免地要受到作者水平、写作时间、篇幅等种种限制，因此，作者要传达的信息是否到位或者是否得体，还要经过读者的检验，望广大读者批评指正。

姚普选

2004 年 1 月于西安交通大学

目录

此为试读,需要完整PDF请访问: www.ertongbook.com

第1章 程序设计基础知识

计算机的基本工作原理，就是用人们预先编制好并存放在计算机内部的程序来控制计算机自动运行，完成程序所规定的任务。因而，使用计算机的基本方式就是编写程序和运行程序。这种将预定的任务用程序表达出来的过程叫做程序设计。

程序设计工作要使用程序设计语言，或使用将程序设计语言、编辑工具、调试工具等集成在一起的软件开发工具来完成。因而，理解程序的一般形式和基本设计方法，掌握程序设计语言或软件开发工具的使用方法，都是程序设计人员的首要任务。

1.1 程序与程序设计

为了使计算机能按照人的意图来完成特定的任务，就必须先让计算机接受和执行人给出的命令和数据。用于向计算机发出操作命令，在计算机和人之间互相传递数据的指令序列或语句序列称为程序。

“程序”和“软件”常被当作同义词来使用，“程序设计”和“软件开发”有时也被不加区分地使用。一般来说，编写较小的程序常称为程序设计，而大型程序设计则更多地称为软件开发。

1.1.1 程序的性能与结构

计算机是增强人类信息处理能力的工具，相当于人的头脑功能的延伸，自然具备人类处理问题的特点。因此，计算机的基本工作方式与人处理问题的基本方式十分相似。

1. 人处理问题的基本方式

人们在日常生产生活中要解决各种各样的问题，尽管不同种类问题的内容和解决方法千差万别，但解决问题的基本过程大致可归结为3步。

(1) 接受原始信息：解决一个问题时，首先要通过眼耳等感官感知该问题的原始信息，并将其记忆在大脑的相应功能区；或者通过手口等感官转而记录在纸、录音机和录像机这样的外部设备上。

(2) 分析处理信息：解决问题最关键的步骤是通过大脑并借助其他工具和手段对已获取的信息进行综合分析处理，主要包括以下几个方面。

① 综合分析问题的相关信息，包括原始信息和其他辅助信息，建立起必要的信息之间的总体联系，如数量关系、逻辑关系等，并将其记忆在大脑或纸、录音机、录像机这样的外部设备上。

② 运用某些基本信息，这些基本信息主要是在解决该问题之前已记忆在大脑中的经验、方法、技巧和知识等，将上述信息之间的总体联系进行必要的数学推演或逻辑推理，从而得出所需的结果信息，即中间结果和最终结果，并将其记忆在大脑或纸、录音机和录像机这样的外部设备上。

③ 根据原始信息、相关信息和基本信息，核验所得结果特别是最终结果的信息的可靠性、合理性和正确性。必要时，还可借助外部设备来进行。

(3) 表达出最终答案：经过核验确认是正确的最终结果将通过感官输出。例如，通过报表的形式向有关部门汇报，通过讲演的形式进行宣传，或通过书籍、电子文档等形式提供给公众。

上述人处理问题的基本方式可以用示意图表现出来，如图 1-1 所示。

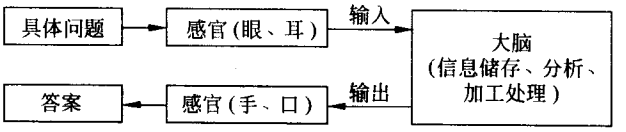


图 1-1　人处理问题的基本方式

2. 计算机的基本工作方式

计算机种类繁多，在规模、处理能力、价格和复杂程度等方面都有很大差别，但各种计算机的基本原理是相同的。数学家冯·诺依曼(Von Neumann)曾经提出过数字计算机设计的一些基本思想，其中最重要的一点是：内存程序控制原理是计算机的基本工作原理。

程序就是为了解决一个信息处理任务而预先编制的工作执行方案，是由一串 CPU 能够执行的基本指令组成的序列，每一条指令规定了计算机应该执行什么操作(如加、减、乘、判断等)以及执行时所需要的数据。例如，从存储器读一个数送到运算器就是一条指令，从存储器读出一个数并和运算器中原有的数相加也是一条指令。

当要求计算机执行某项任务时，就设法把完成任务的方法分解成一个一个的步骤，使每一步骤完成的工作可由若干条指令来完成，并将这些指令所构成的程序送入计算机，以二进制代码的形式存放在存储器中(习惯上把这一过程叫做程序设计)。一旦程序被“启动”，计算机就严格地一条一条地分析执行程序中的指令，从而可以逐步地自动完成这项任务。

程序存储功能使计算机变成了一种自动运行的机器，将程序存入计算机并启动它，计算机就可以独立工作，一条一条地执行指令。虽然每一条指令能够完成的工作很简单，但通过成千上万条指令的执行，计算机就能够完成非常复杂、意义重大的工作。

3. 程序的一般结构

在计算机上解算一个题目之前，先要编制相应的程序，然后存放在计算机中，程序便会控制计算机自动运行，算完这个题目。如果不考虑程序的细节，则可以将程序看成一个函数 F(X)，它接收原始数据集合 X，经过运算处理，最后产生出结果数据集合 Y。即

$$Y=F(X)$$

例如，下面给出一个 Delphi 控制台工程(Delphi 语言[①]源程序)的内容，它由一系列 Delphi 语言的语句组成，其功能是：由用户输入的 x 值，按照如下给定的分段函数，计算得到相应的 y 值。

$$y=\begin{cases}2x+1 & (x\geqslant 0)\\ -x/2 & (x<0)\end{cases}$$

程序代码如下：

```
program Project1;
...
var
  x,y: real;
begin
  read(x);
  if x>=0 then
    y := 2 * x+1
else
    y := -x/2;
  writeln(y);
end.
```

其中，program 引出的 Project1 是工程(程序)的名字。var 引出的“x,y:real”是程序中要用到的变量的声明。begin 到 end 之间的语句为程序的执行部分，可以按人处理问题的方式将其划分为 3 个部分：

1) 输入数据部分

语句 read(x)；用于输入程序中要用到的原始数据，语句的功能是：等待用户从标准输入设备(键盘)上输入一个数据，并将它赋给语句中指定的变量 x。用户键入的数据要与变量 x 的数据类型相容。例如，如果指定的变量是数字，则键入的也应该是数字；如果指定的变量是字符串，则键入的也应该是字符串。

2) 运算部分

以 if 开头的语句，即

```
if x>=0 then
  y:= 2 * x+1
else
  y:=-x/2;
```

① Delphi 7 的支持语言已正式命名为 Delphi 语言，以前称为 Object Pascal 语言。

是程序中的运算部分，该语句按 x 的值所属的范围计算 y 的值。如果 x≥0，则按 y=2x+1 计算 y 值；如果 x<0，则按 y=－x/2 计算 y 值。

3）输出部分

语句 writeln(y)；用于输出运算结果，即按 x 值所属范围计算得到的 y 值。该语句的功能是：将 y 值按默认的格式输出到标准输出设备（显示器）上。

该程序在执行时，屏幕显示输入提示符（一个闪烁的短线），等待用户输入。例如，当用户输入一个数 8.9 并按回车键后，屏幕显示运算结果：

1.88000000000000E+0001

表示 1.88×10^{1}。

1.1.2 程序设计语言

人用计算机解决问题时，必须用某种“语言”来和计算机进行交流。具体地说，就是利用某种计算机程序设计语言提供的命令来编制程序，并把程序存储在计算机的存储器中。计算机在这个程序的控制下运行，达到解决问题的目的。

用于编写计算机可执行程序的语言称为程序设计语言，程序设计语言大体上可按其发展的先后顺序分为 3 类：机器语言、汇编语言和高级语言。

1. 机器语言

能被计算机直接理解和执行的指令称为机器指令，它在形式上是由“0”和“1”构成的一串二进制代码，每种计算机都有自己的一套机器指令。机器指令的集合就是机器语言。

机器语言是计算机诞生和发展初期使用的语言，它和人们习惯使用的语言（如自然语言和数学语言等）差别很大。直接使用机器语言来编写程序是一种十分复杂的手工劳动。例如，下面给出的一条机器指令的功能是：计算机中央处理器从指定的内存单元取出数据，装入指定的寄存器。

10001011 00000101 00000000 01111001 10001111 10101101

可以看出，这种机器指令使人难以理解和使用，也必然使程序不易修改，无法由一种计算机环境移植到其他计算机环境中去，因此很难用机器语言开发实用的计算机程序。

2. 汇编语言

为了克服机器语言的缺点，人们采用一些特定符号（称为助记符）来取代原机器指令中的二进制指令代码，如用 ADD 表示加法（Addition），用 SUB 表示减法（Subtraction）等。同时又用变量取代各类地址，如用 A 取代地址码等。这样构成的计算机符号语言，称之为汇编语言。用汇编语言编写的程序称为汇编语言源程序。例如，下面给出的汇编语言语句与前面的机器指令功能相仿，其功能为：从 DATA1 命名的内存单元中取出数据，装入 AX 寄存器中。

MOV AX, DATA1

汇编语言在一定程度上克服了机器语言难于辨认和记忆的缺点，但汇编语言程序的

大部分语句还是和机器指令一一对应的，更接近于机器语言而不是人的自然语言，而且汇编语言都是针对特定的计算机而设计的，对机器的依赖性仍然很强，因此，对大多数用户来说，仍然不便于理解和使用。

汇编语言程序必须翻译（称为汇编）成机器语言程序才能被计算机识别和执行。

汇编语言和机器语言都是只适用于特定类型计算机的程序设计语言，常被称为“低级”语言。

3. 高级语言

为了克服低级语言的缺点，高级程序设计语言（即高级语言）应运而生，这是一种类似于“数学表达式”（如 Y=5 * cos(A)+1）、接近自然语言（如英文）、能为机器所接受的程序设计语言。高级语言具有学习容易、使用方便、通用性强和移植性好的特点，便于学习、掌握和应用。例如，使用 Pascal 语言，如果想得到 $5\sqrt{x+1}$ 的计算结果，编写和执行下面的语句即可。

```
Write(5 * Sqrt(x+1));
```

高级语言种类繁多，以下是几种曾经或正在产生重要影响的高级语言。

（1）FORTRAN 语言：是最早产生的高级语言，适合于处理公式和进行数值计算。它的产生和发展，曾经极大地推动了计算机的普及和应用。

（2）BASIC 语言：是一种易于学习和使用的高级语言，可用于一般的科技计算以及小型数据处理、计算机辅助教育、音乐演奏和电子游戏等许多方面。目前较为流行的 Visual Basic 就是由 Basic 语言发展而来的。

（3）Pascal 语言：是系统地体现结构化程序设计[①]思想的高级语言。它在支持结构化程序设计和表达各种算法，尤其是非数值型算法上比其他语言都要规整和方便。它是书写算法和进行教学的首选语言。Delphi 语言就是由 Pascal 语言发展而来的。

（4）C 语言：是一种功能较强、使用灵活的语言。用 C 语言编写的程序简洁、易读、易修改，而且运行效率较高。C 语言既具有高级语言的优点，又包含了汇编语言的许多特点，很适合于高水平的程序员开发系统软件或者应用软件。很多著名的软件，如 UNIX 等，都是用 C 语言编写的。C 语言本身也在不断改进，C++，Visual C++等可以认为是 C 语言的提高版。

使用高级语言编写的程序（称为源程序）必须经过相应的翻译程序翻译成由机器指令表示的程序（称为目标程序），然后才能由计算机来执行。这种翻译通常有两种作法：编译方式和解释方式。

编译方式是将高级语言源程序送入计算机后，调用编译程序（事先设计的专用于翻译的程序）将其全部翻译成机器指令表示的目标程序，然后再执行目标程序，得到计算结果，如图 1-2 所示。

解释方式是在高级语言源程序输入计算机后，启动解释程序，翻译一句，执行一句，直到程序执行完为止，如图 1-3 所示。

---

① 为提高程序质量（可靠性）而采用的较为规范的程序设计方法。

此为试读，需要完整PDF请访问：www.ertongbook.com