

程序设计24学时 学习教程

(美) Greg Perry 著
黄亚昌 何玉洁 等译
杜悦冬 耿胜华

机械工业出版社

本书深入地探讨了严格意义上的程序设计原则，论述了作为程序员应掌握的基础知识和良好习惯。介绍了程序设计的发展历史、当前流行的编程语言、编程公司的运作、编程任务的总体设计以及Internet编程。本书语言生动流畅，思路活跃清晰，附有练习和答案，是一本很好的编程入门教材。

Greg Perry: Sams Teach Yourself Beginning Programming in 24 Hours.

Authorized translation from the English language edition published by Sams Publishing.

Copyright © 1998 by Sams Publishing.

All rights reserved.

本书中文简体字版由机械工业出版社出版。未经出版者书面许可，本书的任何部分不得以任何方式复制或抄袭。

本书封底贴有Prentice Hall 防伪标签，无标签者不得销售。

版权所有，翻印必究。

本书版权登记号：图字：01-98-2505

图书在版编目(CIP)数据

程序设计24学时学习教程/(美)帕瑞(Perry,G.)著：黄亚昌等译，—北京：机械工业出版社，1999.3

书名原文：Sams Teach Yourself Beginning Programming in 24 Hours

ISBN 7-111-07016-x

I. 程… II. ①帕…②黄… III. 程序设计—教材 IV. TP311

中国版本图书馆CIP数据核字(1999)第00985号

出 版 人：马九荣（北京市百万庄大街22号 邮政编码100037）

责任编辑：吴 怡

北京市南方印刷厂印刷·新华书店北京发行所发行

1999年3月第1版第1次印刷

787mm×1092mm 1/16·20.5印张

印数：0 001-7 000册

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

译 者 序

现代社会已经进入了信息化时代，而信息化的实现很大部分是由计算机技术的飞速发展所推动的。社会的快速发展所造成的对计算机人才的巨大需求使得计算机成为高薪行业，也使得从事这个行业的人能够成为时代的弄潮儿。随着技术的发展和竞争的加剧，对从事计算机行业的人的素质的要求也越来越高了。

要想成为一名真正的程序员，必须对计算机的软件和硬件都有清晰的了解。尤其是现在计算机技术日新月异，各种新技术竞相登场的时候。从早先的QBasic语言到最新的Java语言，从单独的大型机到Internet上的全球联网的各型计算机。这一切都要求程序员既要和技术有全面的了解，又要对技术本质有深刻认识。

现在国内的很多计算机编程入门书籍老旧不堪、内容苦涩难懂、语言僵硬乏味，使得不少人对编程工作望而生畏，直至完全丧失了兴趣。所以我们及时地推出了这本编程基础教程，以飨那些有志于成为程序员，而又对计算机的编程技术缺乏全面和深刻认识的入门者。本书的作者用幽默诙谐的语言从各个方面讲解了编程所要具备的知识。从计算机的发展历史到计算机对人类生活产生的巨大影响，从计算机硬件到各种各具特色的编程语言，从常用的编程算法到程序的调试与发布，无所不包。读完此书，你将会发现编程并不象想象的那样困难和枯燥，它简直就是一种娱乐。你可以在编程中体会到个人聪明才智的发挥以及无穷无尽的乐趣。我们相信本书一定会不负重望的。

参与本书翻译的有：黄亚昌、何玉洁、杜悦冬、耿胜华、臧健、武欣、何青、宁靖、李云峰、贾增强、李迎、何玉书和彭志航。耿胜华和杜悦冬审校了全书。由于本书内容涉猎面极广，而译者水平有限，难免有疏漏或错误之处，欢迎广大读者批评指正。

1999年1月于北京

前言

编程简介

学习如何对计算机进行编程比你想象的要来得简单。如果你对计算机不太熟悉，如果你甚至于不知道如何拼写PC，如果你尽量避免把所有问题都搅到一块但还是没法奏效的话，那么你手头拿着的这本书里就说明了解决相关问题的办法，而这正是你在使用计算机遇上麻烦时需要的。

这24学时的教学课程不仅仅限于讲解编程技巧。本教程也不仅仅限于描述Visual Basic、C以及Java之间的区别，而且也不仅仅限于教你编程的全部含义是什么。本教程是一个培训工具，你可以用它来发展你自己的严格意义上的编程技巧。本书的目的就是要向你介绍在遵循专业上认同的原则的同时又能使事情简单化的编程方法。教给你一门编程语言并不是本书仅有的目的（尽管你在读完这本书之前就已经可以自己编写程序了）。本书的目标是为你日后成为你所能做到的最好的程序员打下基础。

这本24学时的课程深入地探讨了严格意义上的程序设计原则。你不仅将会学到如何编程，而且还会学到如何为编程作准备。本教程同时还教给你如何适应公司的编程，并从编程的角度向你讲述了应该如何做才能使你成为公司所需要的人力资源。你将学到所有有关编程工作的主要问题，以及如果你要为其他人编写程序的话应该做些什么。

谁应该使用这本书？

本书的书名说明了一切。如果你从来没有对计算机编过程序，如果你根本不喜欢编程，或者如果你的录像机的计时器老是让你中途停下的话，长叹三声松弛一下吧！这本书就是为你写的，以便让你在24学时课程的学习之后，就能理解计算机程序的特点，并且能够自己写程序。

本书是专门为三种不同的人准备的：

- 对编程一无所知、但希望了解编程是怎么回事的人。
- 希望把不懂计算机编程的用户培训成编程职员的公司。
- 学校——包括语言学习班和系统分析与设计班，学校希望对代码设计和语言类型进行升级改造，并向学生说明一个程序员的生活概况。

一些读者对现在书架上摆放着数量过剩的标题混乱的计算机速成图书感到精力交瘁，但在这本书中他们将会有一个受欢迎的缓解办法。你现在拿着的这本书也向初学者讲解编程的内容，但并不以居高临下的语气对他们说话。

本书将为你做些什么

在接下来的24学时，你将学习到编程所涉及的几乎每个方面的内容。下面的主题将在24学时的教程中逐步深入地进行讨论：

- 与编程有关的硬件和软件
- 编程的历史
- 编程语言
- 编程公司
- 编程任务
- 程序设计
- Internet编程
- 编程展望

这本书真的能在24学时内教会编程吗？

一句话，能。你可以用一个学时或者更短的时间掌握一学时的内容（顺便说一句，在本书后面的部分里，每学时都以“学时”来指代）。内容以一种合理的方法被均匀地分配开来，使你每个学时的学习更加有效，并且让你的编程技巧在每学时都得到提高。

本书中使用到的字体

这24学时教程中每个学时的结尾处都有问与答的对话，以及附加的测验，其答案在附录给出。

本书使用了多种字体来帮助说明编程中遇到的相关主题。

如果一项任务要求你从菜单中进行选择，本书用一个逗号将菜单命令隔开。所以，本书用File,Save来表示从File菜单中选择Save选项。

本书黑体标出的内容用来引出不同类型的信息，以便于辨认。

注意 是在每学时的阅读中专门加以强调的注解。这些注解用来标明概念和过程。

提示 提供了对于一些普通问题的解决方法 and 捷径。

警告 是对你发出关于某些缺陷的警告。阅读这些内容将会让你节约时间以避免麻烦。

新术语 解释重要的新词和新内容。

作为本书的读者，您是我们最重要的批评家和评论员，您的观点很有价值。如果对本书有何意见，或有何高见，请与我们联系，地址为：Fax:317-817-7070, E-mail: prog@mcp.com.

英文原书书名为：Sams Teach Yourself Beginning Programming in 24 Hours.

英文原书书号：ISBN: 0-672-31355

目 录

译者序
前言

第一部分 概 述

第1学时 把计算机当成一种工具	1
1.1 计算机能做什么	1
1.2 常见的错误理解	2
1.2.1 神话1: 只有数学专家才能进行 计算机编程	3
1.2.2 神话2: 计算机会做错事	3
1.2.3 神话3: 计算机难以使用	4
1.3 计算机编程带来的许多好处	5
1.3.1 家庭使用的计算机	5
1.3.2 商用计算机	6
1.3.3 计算机在就业市场上的作用	8
1.4 人与计算机	8
1.5 使用程序比买计算机开销更多	9
1.5.1 硬件	9
1.5.2 软件	10
1.5.3 人力	11
1.5.4 数据	11
1.5.5 规程	11
1.5.6 将以上内容进行推广	12
1.6 小结	12
第2学时 计算机硬件剖析	14
2.1 计算机的发展背景	14
2.2 今日的计算机	17
2.3 计算机的种类	17
2.3.1 超级计算机	17
2.3.2 大型机	18
2.3.3 小型机	18
2.3.4 微机	18
2.3.5 顾客/服务者计算模式	19
2.4 从程序员的角度看硬件	19

2.4.1 理解PC硬件	19
2.4.2 速度的计算	21
2.5 操作系统	21
2.6 硬件与程序	23
2.7 现在开始会不会太迟	24
2.8 小结	24
第3学时 程序是什么	26
3.1 理解对于程序的需要	26
3.2 程序无所不在	27
3.3 把程序作为方向指示	28
3.3.1 程序是保存好的指令	29
3.3.2 是艺术还是科学	31
3.3.3 语言的使用	31
3.3.4 计算机无法处理二义性	32
3.3.5 计算机能说多种语言	32
3.4 语言翻译器	33
3.4.1 解释器	34
3.4.2 编译器	35
3.5 精确就是一切	35
3.6 小结	37
第4学时 程序设计	38
4.1 理解设计的必要性	38
4.2 程序设计	38
4.2.1 第一步: 定义输出	39
4.2.2 第二步: 逻辑开发	45
4.2.3 第三步: 编写程序	52
4.3 小结	53

第二部分 编程基础知识

第5学时 编程过程和结构化技术	55
5.1 使用编辑器	55
5.1.1 行编辑器	56
5.1.2 全屏编辑器	56
5.2 结构化编程	58

5.2.1 顺序执行	60	8.1.1 安装QBasic	95
5.2.2 判断	60	8.1.2 启动QBasic	96
5.2.3 循环	61	8.2 使用QBasic编辑器	97
5.3 测试程序	62	8.2.1 用QBasic编辑器进行工作	97
5.4 学习编程语言	63	8.2.2 运行你的第一个QBasic程序	98
5.5 小结	63	8.3 深入考察QBasic	100
第6学时 早年的编程语言	65	8.3.1 对代码进行注释	100
6.1 储存程序和数据	65	8.3.2 存储数据	102
6.1.1 理解ASCII码表	66	8.4 赋值	103
6.1.2 ASCII码的本质	66	8.5 查看值	104
6.2 理解二进制算术	67	8.6 清除屏幕	105
6.3 记住第一个程序	69	8.7 用QBasic进行计算	106
6.4 机器语言的简化	70	8.8 小结	108
6.4.1 低级编程	70	第9学时 输入与输出	110
6.4.2 安全性也是一个问题	71	9.1 高级输出语句	110
6.5 更接近英语的表达	71	9.1.1 使用分号	110
6.5.1 使用FORTRAN语言	71	9.1.2 使用逗号	111
6.5.2 使用商务语言COBOL	73	9.1.3 输出空行	112
6.6 过去年代中的其他语言	76	9.1.4 控制空格距离	112
6.6.1 PL/I	76	9.1.5 打印到打印机	113
6.6.2 RPG	77	9.2 用INPUT得到键盘数据	114
6.6.3 APL和ADA	78	9.2.1 输入字符串和多个变量	115
6.7 小结	79	9.2.2 合并PRINT和INPUT	116
第7学时 现代的编程语言	81	9.2.3 使用USING语句	118
7.1 使用Pascal进行结构化编程	81	9.3 小结	120
7.2 回顾C语言的背景	83	第10学时 用QBasic进行数据处理	121
7.2.1 C的成功	84	10.1 用IF比较数据	121
7.2.2 理解C语言	84	10.2 编写关系测试	123
7.2.3 C++语言: 一种更好的C语言	86	10.3 使用SELECT CASE语句	124
7.3 学习BASIC	87	10.4 循环语句	124
7.3.1 用Visual Basic进行图形化编程	90	10.4.1 使用FOR...NEXT循环	125
7.3.2 图形化对象	92	10.4.2 控制FOR 循环	127
7.4 游戏是个例外	92	10.4.3 使用DO...WHILE循环	130
7.5 哪种语言最好	92	10.4.4 使用DO...UNTIL循环	130
7.6 小结	93	10.5 小结	132
第三部分 基础编程语言			
第8学时 你学到的第一种语言: QBasic	95	第11学时 管理数据和磁盘文件	134
8.1 准备开始	95	11.1 理解磁盘存储和数组的重要性	134
		11.2 介绍数组	134
		11.2.1 为数组分配空间	136

11.2.2 并行数组	137	14.2.1 应用程序向导	177
11.2.3 清除数组	138	14.2.2 向导完成后	178
11.3 介绍磁盘文件	138	14.3 从基础开始创建一个简单的应用程序	179
11.3.1 记录和字段	139	14.3.1 添加细节	180
11.3.2 访问类型	140	14.3.2 完成代码	182
11.3.3 关闭打开的文件	141	14.4 其他Visual Basic编程考虑	183
11.4 创建输出文件	141	14.4.1 理解过程	184
11.5 读取磁盘文件	143	14.4.2 理解Visual Basic所蕴含的语言	184
11.6 对数据文件进行添加	145	14.5 你的下一步	185
11.7 打开两个或更多的文件	146	14.6 小结	185
11.8 高级数据文件	147	第15学时 用C语言编程	187
11.9 小结	147	15.1 C语言介绍	187
第12学时 QBasic趣味编程	149	15.2 你需要什么	188
12.1 让喇叭发音	149	15.3 C概览	189
12.2 使用SOUND命令	150	15.3.1 使用main()函数的格式	190
12.3 使用PLAY命令	151	15.3.2 使用#include语句	190
12.4 图形介绍	152	15.4 C的数据	190
12.5 使用SCREEN 命令	153	15.5 C的注释	191
12.6 控制像素的显示	153	15.6 声明变量	191
12.7 绘制直线和长方形	156	15.7 C函数	192
12.8 画圆	159	15.7.1 使用内定义的函数	192
12.9 文本图形	159	15.8 C运算符	197
12.10 改变PRINT的定位	160	15.9 C控制语句	198
12.11 小结	161	15.10 小结	199
第四部分 各种环境下的编程		第16学时 用C++编程	201
第13学时 Windows编程	163	16.1 了解C++	201
13.1 可视化编程的问题	163	16.2 对象术语	202
13.2 控件给用户交互能力	165	16.3 C和C++的基本差别	202
13.2.1 控件减轻了你的工作负担	166	16.3.1 注释	202
13.2.2 放置控件	167	16.3.2 名字差异	202
13.2.3 属性区分了控件	167	16.3.3 I/O差别	203
13.2.4 捕获焦点	169	16.4 引入对象	203
13.3 事件驱动的程序	171	16.4.1 定义类	204
13.4 事件处理方法	173	16.4.2 声明对象变量	205
13.5 小结	173	16.4.3 访问成员	206
第14学时 用Visual Basic编程	175	16.4.4 给对象添加行为	206
14.1 回顾Visual Basic屏幕	175	16.4.5 在类的作用域内工作	208
14.2 Visual Basic编程向导	176	16.5 将来的事情	209
		16.6 小结	210

第17学时 编程算法	212
17.1 计数器和累加器	212
17.2 对换数据值	215
17.3 排序	216
17.3.1 使用起泡排序	216
17.3.2 分析起泡排序	218
17.4 对数组进行查找	219
17.4.1 执行顺序查找	220
17.4.2 改进顺序查找	221
17.4.3 执行折半查找	223
17.5 子程序	226
17.5.1 理解对子程序的需要	226
17.5.2 组织例程	228
17.6 嵌套循环	230
17.7 小结	231
第18学时 批处理、宏和脚本编制语言	232
18.1 批处理: 启动每个程序的语言	232
18.1.1 批处理文件的用法	233
18.1.2 评审批处理语言	233
18.2 宏语言	234
18.2.1 对宏的需要	235
18.2.2 使用键盘宏	235
18.2.3 非键盘宏	237
18.3 Visual Basic for Applications	237
18.4 在Windows中编制脚本	239
18.5 小结	241
第19学时 Internet 编程概念	243
19.1 Internet编程	243
19.1.1 Internet连接	244
19.1.2 Internet通信	244
19.2 对简单导航的需要	245
19.3 HTML编程	246
19.3.1 理解一个HTML实例	246
19.3.2 理解HTML	247
19.4 方兴未艾的ActiveX控件	249
19.5 在Internet应用程序中编制脚本	250
19.6 小结	252
第20学时 Java编程	253
20.1 Java介绍	253

20.2 Java提供了可执行的内容	254
20.3 可在多平台执行的内容	255
20.4 Java使用概要	256
20.5 安全问题	257
20.6 Java的其他内容	258
20.7 Visual J++: 一个Java系统的例子	258
20.8 Java语言特点	260
20.8.1 Java的例子	260
20.8.2 代码分析	261
20.9 小结	263

第五部分 编程业务

第21学时 公司怎样编程	265
21.1 数据处理和其他部门	265
21.2 支付数据处理部门的费用	267
21.2.1 了解摊派方法	267
21.2.2 理解收费服务方法	267
21.3 计算机工作	268
21.4 工作名称	269
21.4.1 学位和证书	269
21.4.2 技术证书	269
21.4.3 数据录入	269
21.4.4 编程	270
21.4.5 分析和设计职员	271
21.4.6 Internet和与网络有关的工作	272
21.4.7 进入管理层的可能性	272
21.5 结构化的预排	273
21.6 使程序成为产品	274
21.7 咨询	275
21.8 小结	275
第22学时 调试工具	277
22.1 用QBasic练习调试	277
22.1.1 输入程序	277
22.1.2 单步执行	280
22.1.3 跳过过程	281
22.1.4 打印变量	281
22.1.5 设置断点	283
22.2 其他调试工具	284
22.3 小结	285

第23学时 发布应用程序	287
23.1 软件发布中的问题	287
23.2 QBasic太容易了	287
23.3 Windows 应用程序发布	288
23.4 第一步: 编译	289
23.5 布署应用程序	290
23.5.1 启动打包和布署向导	291
23.5.2 向导的选项	291
23.5.3 安装位置	292
23.5.4 依赖文件	292
23.5.5 选择位置	292
23.5.6 完成向导	293
23.6 生成安装之后	294
23.7 卸载应用程序	294
23.8 小结	294
第24学时 编程展望	296

24.1 一些有用的工具	296
24.1.1 剖视器	296
24.1.2 版本控制器	297
24.1.3 资源编辑器	298
24.2 编程会消失吗	298
24.2.1 CASE工具	299
24.2.2 UML: 数据建模	299
24.3 Y2K启示	300
24.3.1 Y2K的原因	300
24.3.2 解决Y2K	301
24.4 培训的需要	302
24.5 小结	303

第六部分 附 录

每学时后面问题的答案	305
------------------	-----

第一部分 概 述

第1学时 把计算机当成一种工具

如果你对各种计算机技术手册和出错信息有点无所适从的话，那么本书将告诉你，计算机不仅容易理解，而且对其进行编程也同样简单。尽管“友好的用户界面”的承诺满天飞，但直到现在人们依然对计算机的一些奥妙感到难以理解。在每个书店，人们都在寻找计算机方面的书籍，看看你们当地的社会大学的课程表，各个班级都开设了五花八门的计算机课程，如“计算机入门”、“计算机概念”、“从零开始学PC”、以及“初学者PC编程”等等。

本书充分尊重你的思考和学习能力，它教给你的内容既不低于也不高于你的领会水平，你不必非得成为精通计算机的老手（或甚至是专家）。本学时以一个程序员的角度向你介绍计算机是如何用于商务和家庭的。离开了程序，计算机就比大个儿的计算器强不到哪儿去。

本学时的主要内容包括：

- 如何区别数据和信息之间的真正差异。
- 什么是程序。
- 对计算机三个最常见的错误理解。
- 如何在家庭中使用计算机。
- 如何在商务上使用计算机。
- 设计成功的计算机信息系统。

1.1 计算机能做些什么

现在大多数人对于计算机的用途已经有了一些了解，计算机可以帮人们存款或记录发明成果。如果你想开始编程的话，你使用计算机大概已经有一段时间了。无论如何，作为一个未来的程序员，你或许应该在掌握一门新语言之前复习以下基本概念。

简单地说，计算机是用来处理数据的。很多公司将它们的计算机编程部门称作数据处理部，因为计算机将数据处理后转变成有实际意义的信息。在此之前你可能并未意识到数据和信息的区别，但对于一个计算机专业人员而言，这两者之间的差别是巨大的。

新术语 数据是由原始事实和数字组成。信息则是经过处理的数据。信息具有实际含义；而数据作为其原始形式，并无实际意义。图1-1显示了数据处理的基本模型，注意数据进入计算机后计算机处理数据从而产生有意义的信息。计算机及其程序进行输入处理，然后程序将数据转变为有用的信息。

数据通常用某种输入设备进行输入，如键盘。信息则往往由输出设备进行输出，如显示器、打印机、或磁盘文件等。在这24学时的课程内容中你将会看到，计算机可以用多种不同类型的设备来进行输入和输出。

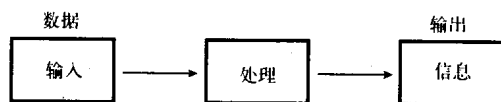


图1-1 最基本的数据处理模型

信息是有用的

数据自身对于需要据其作出决定的人们而言是无用的。人们需要信息(即处理过的数据)来作决定。一家大公司的总裁关心的是公司1500名员工的工资单。当公司生意不错的时候,总裁会给雇员们相应的奖励,于是,总裁可能要求看看最近三个月的工资单,并确定分析给哪个部门更多的奖励。

总裁会要一张包括每个员工(甚至是拿最低报酬的兼职雇员)的单子吗?他会要最近三个月里1500人的每月工资表吗?或许吧,但是这类数据与每个部门的总工资表、去年同期工资增长幅度、或同行业其他公司雇员的平均工资相比不是那么有用。对于总裁来说,由这些数据加工而成的信息才更为有用。带有合适程序的计算机可以迅速地提供总裁需要的这些资料。有用信息的来源就是这些详细的工资数据。

新术语 程序是计算机做任何工作的驱动力。程序是计算机所执行的详细指令的程序清单列表。离开了程序,计算机什么也干不了。程序员的工作就是设计并写出相应的程序,使得计算机能够取得原始数据并将其转化为有意义的信息,送到最终用户手中。计算机的最终用户(或简称用户)往往需要计算机提供的结果(信息),但他自己很可能不是技术人员或程序员。

注意 对于一个人来说是数据的,对于别人可能是信息。对于职员来说,他的薪水数目是有意义的信息;而对于总裁来说,这可能只是大量无意义数据中的一个部分。

程序员是最有责任来引导计算机进行工作的人。学习计算机编程要花上一点时间,但这将带来回报。计算机编程可以给你直接的回报,这一点不像其他工作,你可以训练计算机。在你学完接下来的24学时内容之前,你就已经能够自己写程序并看到运行结果了。许多编程指导书都假设你已经知道了计算机的一些背景知识,并且是以程序员的眼光来看待计算机是如何工作的,但这本书绕开了这些麻烦,直接带你像一个程序员一样进行工作。

1.2 常见的错误理解

似乎喜欢计算机的人往往会爱上计算机,而不喜欢计算机的人则对其深恶痛绝。大多数时候,一个人对于计算机的厌恶直接反映了此人计算机知识的贫乏。尽管计算机已经得到广泛应用,但许多人仍对其几乎一无所知。对这些人来说,计算机以及驱动它们的程序与魔术盒毫无二致,根本无须去理解。

这段文章直接对准问题的要害:计算机易用易编程。计算机就是一台哑巴机器,它什么也不知道。计算机像机器人一样,等待你向它发命令,并且执行你让它执行的指令。当然,有时你的指令是错误的,这时计算机依然会不顾一切地执行指令,一往无前。

警告 不要害怕计算机编程。计算机是帮助你完成任务的工具,你有能力学会编程。

你可能听说过由于程序错误而导致某人的银行存款被删除的可怕故事。你或许相信某一天计算机将统治世界。对计算机一无所知的人却总是害怕最糟糕的事情发生。这些人害怕的

是计算机在自身功能范围内所做的工作。计算机仅仅是一种机器而已，而不是一个人头脑中想象的复制品。正如一辆小汽车、一把电钻或一台洗碗机不能接管这个世界一样，计算机也不行（如果它真干出点什么……的话，你总可以把插头拔下来吧，但它并不会干什么）。

对于计算机的大多数误解来自于对计算机如何工作以及计算机实际上能干什么缺乏了解。这本书想要做的正是打破神话并增加你对这种机器的了解，不久你就可以进行计算机编程了。计算机只不过是一件工具，帮你干某类工作。计算机本身并无好坏之分，锤子可以用来做好事的工具（用来修房子），也可以是用来做坏事的工具（用来搞破坏），计算机落在坏人手里可能会被利用来干坏事，但那并非是计算机的错，正如坏人用锤子搞破坏并非锤子的错一样。

计算机不仅是有用的工具，而且是如今公司里必需的工具。如果没有了计算机，大多数公司、学校和银行都得关门，原因很简单，大量的数据要用手工来处理实在是不太可能。想想看，如果没有计算机的协助，要控制机场上空成群结队的飞机该是多么困难和危险。商务分析、邮政业务、天气预报、航空管理，甚至教孩子们学习数学、自然科学和阅读技巧，都需要计算机。

接下来的几个段落将打破三个最大众化的计算机神话。你曾听说过这些说法吗？你相信这些话吗？

1.2.1 神话1：只有数学专家才能进行计算机编程

幸好这只是一个神话而非现实，否则成千上万的人要失去工作了（包括大多数计算机书籍的作者！）。计算机如果是高人一等的机器，只能由最好的工程师和科学家来使用，普通人可用不了，那么，计算机在某些领域会带来好处，但却不会让这么多人都从中分享利益。

你不仅可以数学学得差，而且还无须喜欢数学，甚至无须为了成为一名优秀的计算机程序员而不得不学好数学。计算机为你完成全部数学任务，那是他的工作之一。世界上有大量的专业计算机程序员说不出圆的面积或64的平方根，所以如果你对这个神话信以为真的话，现在你仅管放心。

编程能带来额外的好处。当你成为一个优秀的程序员时，你会发现你的数学技巧得到了提高。编程技巧的提高将促进你的左脑开发（医生认为左脑是控制数学技能的部分）。所以，数学学得好是编程的结果，而不是先决条件。

提示 喜欢逻辑猜谜、纵横拼字以及单词搜索等游戏的人似乎能更快适应编程，不过类似这些游戏技巧并非编程的先决条件。你将会发现，你也能学会计算机编程，而且还能干得漂亮，而这并不需要你一定得喜欢数学，精通数学、或具有各种推理和拼字游戏的能力。

1.2.2 神话2：计算机会做错事

你可能听说过这句话：“人的想法错了，但确实是计算机做错了。”这话也许对，但这仅仅是因为计算机实在太快了，以致于它将人的错误翻了一番。

计算机并不做错事，人才会做错事。如果你听到一个银行职员这样告诉你，你的存款数目由于“计算机程序出错”而被错误地删减了24美元，那么这人肯定不知道到底发生了什么情况。人对计算机进行编程，人运行计算机，人输入数据，而计算机只是处理数据。

计算机偶然弄错储户银行存款的事是不会有。除非程序编写错误，计算机是不会犯随机错误的。计算机是能力有限的机器；当输入相同时，它产生的输出结果总是相同的。也就是说，计算机在同样的条件下总是做同样的事。你学习编程，就是为了减少计算机出错的机会。

当计算机工作不正常时，就不是简单的错误了；相反，它的确会把事情弄糟。当计算机自身出错时，它往往会完全瘫痪，或是存贮设备中断，或是电源切断。无论发生何种错误，计算机都会出现异常，而且计算机出错时，往往很容易看出来。幸运的是，计算机自身的问题极为少见。

注意 你和其他每个人都应当知道什么时候计算机出错了。

在人们发明计算机之前，银行用帐户卡进行记录。当出纳发现一个错误（也许是出纳自己造成的）时，你会同意他说“帐户卡犯了一个错误”吗？当然不会。计算机会有机械上的问题，像有会儿不稳定诸如此类的小错误实在是微不足道的。

1.2.3 神话3：计算机难以使用

如果计算机使用困难，那么对其进行编程也很困难。计算机的使用一天比一天变得容易，编程也是一样。如果你使用过微波炉或驾驶过最近出产的小汽车，那么你就已经有过使用计算机的时候了。你自己知道当时正在用计算机吗？也许不知道。计算机厂商已经将计算机引入到你的日常生活中，帮助你解决生活中的许多问题。

当然，如果你正在看这本书，你就会希望对计算机有足够的了解，以便编写自己的程序。编写计算机程序当然比使用微波炉的定时功能要学的东西多，这主要看你希望计算机做些什么及计算机的水平。

现在不仅计算机的使用越来越方便，而且你比以前有更多的机会来学习计算机。有线电视频道开设了计算机使用和编程方法的学习教育节目，相关的书籍和音像资料不计其数。

想想计算机产业的目的，计算机厂商需要赚钱。这些商家想赚许许多多的钱，而人们买的计算机越多，它们赚的也越多。它们赚得越多，它们提供的工作机会和为国家积累的财富也越多。你认为计算机产业会限制学习使用计算机的人数吗？根本不会。当计算机厂商将计算机制造得更容易使用之后，就有更多的人会使用计算机，更多的人购买计算机，而计算机也会帮助更多的人更为有效地工作。

本书的部分内容介绍了一些计算机厂商使计算机更易于使用和编程的方法。例如，象Windows这样的操作环境现在集成了字处理器，使得打印和检查你的文本变得跟按按钮一样简单，而字处理器只是市场上成千上万的例子之一。计算机制造商都试图使自己的计算机比他的竞争对手的更易使用，这样，你就会买他的产品。

友好的用户界面…不!

友好的用户界面这个术语在计算机学科当中太常见了。在60和70年代，当计算机的销售刚步入黄金时代、但其使用仍较为困难时，市场部门就有人提出了友好的用户界面这个提法。友好的用户界面意味着对于计算机用户来说，程序更为友善，而不要求编程非得经过大量训练而且难以掌握。

友好的用户界面随之带来的问题是，自从这一术语提出之后，似乎每个程序都被描述成具有友好的用户界面。没有人能确切地衡量一个程序是否友好，因为每个人的喜好不同，而

且人们学习的技巧也各不相同。

1.3 计算机编程带来的许多好处

计算机是一种美妙的工具，它使世界变得更为绚丽多彩。我们的社会正从工业社会步入信息社会，信息被认为是决定性的本钱。多亏有了计算机，现在人们可以从指尖得到比以前更多的信息。

接下来的部分将向你简要介绍现在的商用和家用计算机。在了解以下内容时，务必记住：计算机之所以带来好处的原因是有人花了时间去学习它的工作原理以及如何用它来编写完成各种任务的程序。没有程序的计算机就象一个空的珠宝盒；离开了程序，计算机就变成了这种机器，它要你把工作过程的每一步都告诉它。

1.3.1 家庭使用的计算机

走进任意一家计算机商店，销售人员都会对你说它能够用它来管理支票本和食谱。虽然计算机能干这些事情，但它能干的事远不止这些。实际上，用计算机管理食谱可能会在筛面粉的时候把机器弄脏，而且你大概也不会在每次签支票时都打开计算机。不管怎样，计算机还是能毫不费力地迅速贮存记录。

家用计算机的最大好处之一是用于教育。成人和孩子一样，可以把计算机当作教师，并在学习时得到直接的回报。完全计算机化的百科全书如今已十分普遍。使用现在的多媒体计算机（能提供声音和图像的计算机），你可以在计算机化百科全书屏幕上在阅读时听到和看到其中的人物与故事（如图1-2所示）。

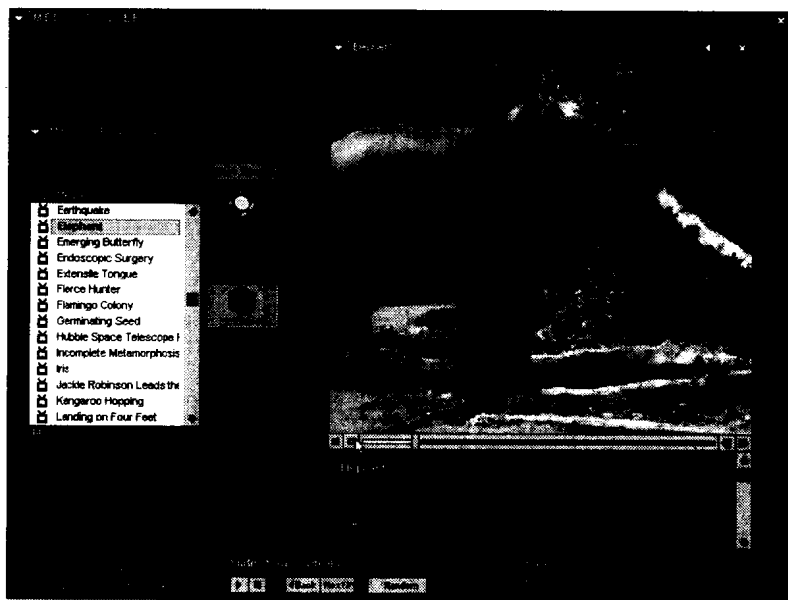


图1-2 在计算机化百科全书上观看和阅读相关主题

提示 如果你有孩子，那么一旦你学会如何编程之后，你可能想为他们编一些学习程序。

你可以设计一些与商店出售的不同的程序，视你孩子的专门需要而定。

计算机无法离开程序来运行多媒体百科全书，而没有程序员就没有程序的存在。

因特网(Internet)已经改变了人们使用计算机的方式。因特网是一个计算机互联系统，你可

以通过电话线访问它。它可以让你在家里办理银行业务或购物，可以让你读到成千上万的杂志和书籍，可以查找广告信息，并给网上的其他任何人发电子邮件（不管他们在世界上的哪个地方）。因特网不是一个程序，而是为这个松散的计算机互联系统起的通用名字，但如果没有程序，你就不可能从你的PC上访问因特网。

如今家用计算机最普遍的应用也许是字处理。你可能已经知道，字处理器是一个计算机化的打字机，但它比打字机强多了。你可以对文档轻松地进行打字、编辑、移动、拷贝、存储、恢复以及打印，而这些在以前的打字机上无法进行。现在的字处理器程序是如此强大，你甚至在一张桌子上就可以将正文和图片集成起来并出版自己的著作。图1-3展示了一幅用当今最流行的字处理器Word创建的时事通讯稿。

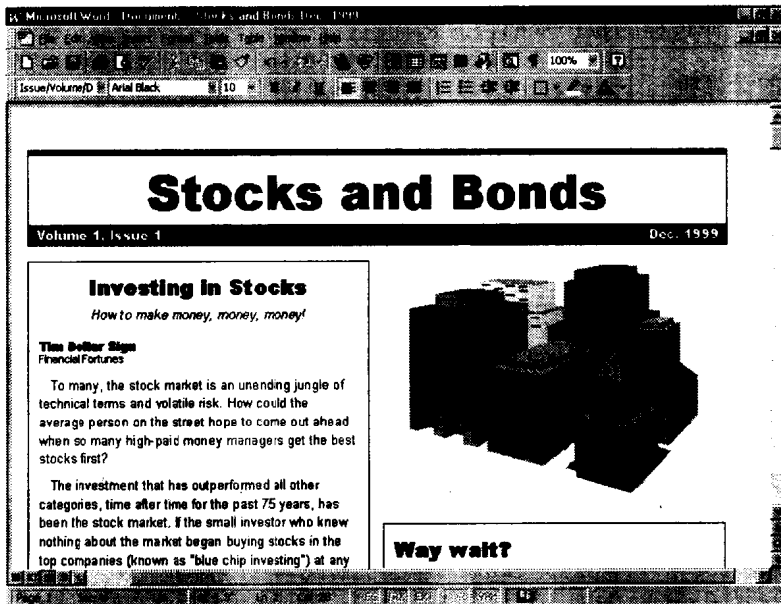


图1-3 在家里创建出版物

这24学时的教学指导并不是对整个计算机产业的描述性文章。这本书的目的是让你作好计算机编程的准备。如果你正在看这本书，你或许已经在家或在工作中使用过计算机，而且也可能已经用过了字处理器。无论如何，在这段文章里，我们一直试图让你记住：是计算机编程才使得计算机有了今天的所有应用。

这一学时告诉你计算机无非是件工具，尽管它是一类特殊的工具。它是你一生中最常用的工具。一会儿你的孩子在百科全书上读有关第一次月球行走的内容，一会儿又是在为你的生意作财务分析。你可以用计算机拨通银行在线服务，要求在星期天早上4:15将钱从一个帐户转到另一个上面去。你也可以把你的家族谱系和假期邮件程序清单存到同一张磁盘上。

尽管有许多程序你都可以直接使用，但有时你会需要一个特殊的程序用于特殊的用途，而你又找不到这样一个现成的程序。当你学完这本书以后，你就会清楚地知道需要在编写自己的程序时设计什么内容。

1.3.2 商用计算机

新术语 尽管字处理器在商务中的使用也和家用的情况同样流行，但电子表格（也称电

子制表软件)也十分常用。电子表格是类似于字处理器的一种程序,用于数字处理。会计师、财务分析员以及其他商务人员用它进行有关分析,得出情况改变后的可能结果。

电子表格被广泛使用的原因有两个。首先,它是一个通用程序,可用于完成多种不同任务。你可以创建一个电子表格用于计算资金流动,另一个电子表格用于计划雇员的午餐费。实际上,一个电子表格就是一个程序,非程序员可以用它来直接做某项专门任务。

新术语 图1-4显示了一个用于计算工资总数的电子表格。在电子表格流行之前(80年代早期),商务人员不得不雇用程序员来编写用于专门计算的程序。现在,商务人员自己就可以开发电子表格应用程序了。电子表格省去了专门编程的必要;也就是说,电子表格的使用无须有经验的程序员介入了。电子表格使得非程序员在一个更高的层次上编程,这一层次不要求掌握某一门计算机编程语言。通过移动鼠标、定义行和列、进行计算以及使用电子表格现成的宏语句(一种简单的高级电子表格编程命令集,在第18学时详细解释),非程序员可以愉快地自主使用计算机。

1996	
1996 Fed tax Refund	534.35
1996 State Tax Refund	232.11
Local City Sales Tax rebate	21.45
Alimony Received	0.00
Fully taxable pensions, annuities, IRAs	409.85
Other pensions, annuities, etc.	1,205.00
Unemployment compensation received	0.00
Social Security benefits received	0.00
Tax free interest received	16.00
Other Tax-Free income	32.75
Other income: Writing, Speaking, and Oil/Gas income on next pages	
Adjustments to income	
Payments to 1996 IRA	
Sam (paid 9-Apr-96)	2,000.00
Sam (paid 9-Apr-96)	250.00

图1-4 电子表格可编程以替代其他一些程序

尽管电子表格功能强大并且可以让非程序员来进行编程,但它仍然是能力有限的工具。对于各种数据分析的工作它是胜任的,但还是有大量它无法完成的工作。如果计算机处理的是各种各样的任务,如打印工资单,或是与其他用户就某个焦点问题进行合作而非数字和文字处理(例如在教学程序中就有这种情况),那么一个电子表格就不够了。

新术语 另一个常用的商务应用是数据库管理。计算机做得最好的一项工作就是对大量数据的处理。当商家把它们的信息组织成一个数据库程序后,就很容易进行管理了。大多数数据库程序都提供了自己的编程语言,这些语言往往比程序员使用的语言更为高级(如电子表格编程语言)。使用数据库编程语言编写的程序,最终用户(非计算机程序员)就可以按自己喜欢的方式开发简单的应用来管理数据。

这三大功能(字处理、电子表格和数据库管理程序)只是现在方便有效的商务应用的一部分。商家、非盈利性组织、教会、科学实验室、体育团体、音乐家、制图员、工程师以及