

高等学校计算机程序设计课程系列教材

Visual Basic

程序设计基础教程

范荣强 主编

梁里宁 刘小丽 副主编



高等教育出版社
HIGHER EDUCATION PRESS

高等学校计算机程序设计课程系列教材

Visual Basic 程序设计基础教程

Visual Basic Chengxu Sheji Jichu Jiaocheng

范荣强 主编

梁里宁 刘小丽 副主编



高等教育出版社·北京
HIGHER EDUCATION PRESS BEIJING

内容提要

本书详细介绍了 Visual Basic 语言的程序结构和运行机制，Visual Basic 语言中的数据及其表示和处理，程序设计的基本方法和技术。

全书共分 9 章，内容主要包括各种数据类型的基本运算和程序的基本控制结构，程序设计在分类法、递推法和穷举法等日常解决实际问题的常用方法中的应用，程序设计中通过程的技术与意义，内外存数据的交换，程序界面设计与绘图操作等。基本涵盖了可视化程序设计基础训练所要求的内容。

本书安排了 11 个具有系统性、连贯性的实验，有助于教学活动的开展。在附录部分，包含了 Visual Basic 6.0 集成开发环境、窗体及常用控件介绍、程序的调试、标准数据类型、常用函数、运算符与表达式等内容，供读者随时查阅。

本书适合作为高等学校计算机程序设计类课程的教材，也可作为 Visual Basic 程序设计爱好者自学用书。

图书在版编目(CIP)数据

Visual Basic 程序设计基础教程/范荣强主编. —北京:
高等教育出版社,2011.2
ISBN 978 - 7 - 04 - 031511 - 0

I. ①V… II. ①范… III. ①BASIC 语言 - 程序设计 - 高等学校 - 教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2011) 第 007197 号

策划编辑	饶卉萍	责任编辑	许 可	封面设计	于文燕	责任绘图	杜晓丹
版式设计	马敬茹	责任校对	姜国萍	责任印制	毛斯璐		

出版发行	高等教育出版社	购书热线	010 - 58581118
社 址	北京市西城区德外大街 4 号	咨询电话	400 - 810 - 0598
邮政编码	100120	网 址	http://www.hep.edu.cn http://www.hep.com.cn
经 销	蓝色畅想图书发行有限公司	网上订购	http://www.landaco.com http://www.landaco.com.cn
印 刷	唐山市润丰印务有限公司	畅想教育	http://www.widedu.com
开 本	787 × 1092 1/16	版 次	2011 年 2 月第 1 版
印 张	16.25	印 次	2011 年 2 月第 1 次印刷
字 数	390 000	定 价	22.50 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。
版权所有 侵权必究
物料号 31511 - 00

前言

学习计算机程序设计要找一个好的切入点是毫无疑问的。这当然包括选用何种程序设计语言。作为非计算机专业学生, Visual Basic (以下简称 VB) 是一个较好的选择, 因为它相对简单、易学, 又引入了面向对象和事件驱动的程序设计思想。更重要的是, 它虽然简单, 但功能却很强大, 作为当今的主流操作系统, Windows 中的应用程序几乎都可以使用 VB 进行开发, 并且很多常用的应用软件都支持使用 VB 进行二次开发。

计算机程序设计应该如何学习? 这个问题是初学者首先要面对的。毋庸置疑, 掌握语言本身的语法规则是前提条件, 但如何去掌握确实是值得研究的。就像学习一门外语必须掌握一定的词汇。但有些词汇量很大的人也未必能很好地描述一件事情, 因为如何描述则是另外一回事。因此, 本书注重将语法和程序设计方法结合。同样, 可视化程序设计要求掌握界面设计的工具、对象属性的设置、事件的发生和方法的调用, 但掌握面向过程的程序设计方法依然是至关重要的。本书在这方面也做了精心安排。

根据多年的教学实践, 我们总结出初学者在设计程序时遇到的困难大多是不知如何描述所面对的问题, 然后才是进一步地应用分支和循环技术去组织程序。因此, 我们在数据的类型、表示以及运算一章中专门安排了基本运算一节, 用较大的篇幅和充实的内容来讲解各种数据类型的基本运算, 以期达到夯实整个课程基础的目的。

书中对例题的讲解都先给出解题思路, 引导读者分析问题、分解任务, 最后完成设计和编码, 使读者体会问题分析和解题方法设计的重要性, 并受益于日常问题的处理。

我们在每章的开始都给出了学习导读。它的意义是引导读者在学习本章时应着重理解的目标, 也许在未学之前不太好理解, 但在学完之后回顾时将会有一定的帮助。

随着视窗界面程序的普遍使用, 人们对窗体界面和常见控件的认知和熟悉程度已相当高。所以在本书中不再专门安排常用控件的章节, 而是将它们融于许多实例与练习中, 但是为了便于读者查阅, 仍将常用控件的主要属性、方法和事件在附录中列出。

本书在编排上注重内容的系统性, 但更重视如何让学生学到系统的知识与技能。例如, 为了让学生能够尽快地在不同的过程中使用同一变量, 我们在尚未系统介绍变量作用域之前就引入了窗体变量的概念; 为了使学生尽早地使用分支结构处理列表框控件, 在没有介绍数组之前就引入了列表框控件, 而将其深入一步的功能留在数组之后介绍。虽然没有一次性地将它介绍完整, 但这更符合循序渐进的要求。另外, 本书主要围绕程序设计的基础问题展开, 至于 VB 的编程环境、程序调试、内部函数和运

算等则在附录中说明。

程序设计是综合性的、高强度的脑力劳动，该课程的学习除了需要课堂讲授以外，还需要通过大量的实践练习才能掌握。因此，读者一定要重视实践环节，包括编程与上机。既要能编写程序，又要会调试和运行程序。

由于作者水平所限，加之时间仓促，不足之处在所难免，希望读者批评指正。

编 者

2010年10月

目 录

第 1 章 VB 程序与编程环境..... 1

1.1 观察一个程序..... 1

1.2 动手编写程序..... 2

1.3 认识 VB 程序..... 5

1.3.1 可视化程序设计的基本概念..... 5

1.3.2 VB 程序的结构..... 6

1.3.3 VB 程序的工作方式..... 7

1.3.4 需要进一步掌握的内容..... 9

习题..... 9

第 2 章 数据的类型、表示以

及运算..... 11

2.1 标准数据类型..... 11

2.2 标识符的命名规则..... 12

2.3 数据的表示..... 12

2.3.1 常量与变量..... 13

2.3.2 内部函数..... 14

2.3.3 表达式..... 15

2.3.4 变量赋值..... 16

2.3.5 命名常量..... 16

2.3.6 静态变量..... 17

2.3.7 窗体变量和局部变量..... 17

2.4 基本计算..... 19

2.4.1 算术运算..... 20

2.4.2 字符串运算..... 23

2.4.3 Choose 函数——选取给定的 第 n 个值..... 25

2.4.4 条件运算..... 25

2.4.5 If 函数——要“此”还是要 “彼”..... 28

2.4.6 日期与时间运算..... 29

2.5 数据类型转换..... 32

2.5.1 隐式转换..... 32

2.5.2 显式转换..... 32

2.6 应用举例..... 33

习题..... 36

第 3 章 基本输入输出..... 39

3.1 输出计算结果..... 39

3.1.1 使用标签或文本框 输出数据..... 39

3.1.2 使用 Print 方法输出 数据..... 39

3.1.3 使用消息对话框函数 MsgBox 输出数据..... 41

3.2 接收用户输入数据..... 42

3.2.1 使用文本框输入数据..... 42

3.2.2 使用输入对话框函数 InputBox 输入数据..... 43

3.2.3 使用滚动条输入数值..... 43

3.2.4 交互性控件的焦点与 Tab 键序..... 45

3.3 图形文件的调用与显示..... 45

3.3.1 图形框..... 46

3.3.2 图像框..... 46

3.3.3 图形框与图像框控件 的区别..... 46

习题..... 48

第 4 章 控制结构..... 50

4.1 顺序结构..... 50

4.2 选择结构..... 51

4.2.1 单分支语句..... 51

4.2.2 双分支语句	52	5.4 查找	98
4.2.3 多分支语句	53	5.4.1 顺序查找法	99
4.3 选择类控件及常用编程方法	55	5.4.2 折半查找法	99
4.3.1 单选按钮	55	5.5 与数组有关的控件	100
4.3.2 复选框	57	5.5.1 列表框的高级属性	100
4.3.3 列表框	57	5.5.2 组合框	101
4.4 选择结构的程序设计	59	5.5.3 列表框和组合框的方法	102
4.4.1 修正值问题	59	5.6 用户自定义的数据类型	103
4.4.2 分类问题	60	习题	105
4.4.3 多功能按钮问题	62		
4.5 循环结构	64	第 6 章 过程	111
4.5.1 计数型循环语句	64	6.1 自定义函数	111
4.5.2 条件型循环语句	66	6.1.1 函数的定义	112
4.6 多重循环	69	6.1.2 函数的调用	113
4.7 循环结构的程序设计	71	6.2 子过程	114
4.7.1 字符阵图与二维表	71	6.2.1 子过程的定义	114
4.7.2 组合问题	73	6.2.2 子过程的调用	114
4.7.3 递推法	73	6.3 参数传递	115
4.7.4 穷举法	80	6.3.1 形参和实参	115
4.7.5 求最值	81	6.3.2 形参和实参的传递方式	116
4.7.6 整数因子与素数判别问题	82	6.3.3 数组参数的传递	118
习题	83	6.4 将通用过程写在标准模块中	119
第 5 章 数组和自定义类型	87	6.5 变量的作用域	121
5.1 数组的概念	87	6.6 过程调用时程序执行的流程	123
5.1.1 引例	87	6.7 递归	124
5.1.2 数组的声明	88	6.8 对通用过程作用的深入认识	127
5.1.3 静态数组和动态数组	90	习题	129
5.2 数组的基本操作	91	第 7 章 外存数据的访问	134
5.2.1 数组的赋值	91	7.1 文件的概念	134
5.2.2 数组的输出	93	7.1.1 引例	134
5.2.3 求数组元素之和、最大值、最小值 和平均值	95	7.1.2 文件的结构	135
5.2.4 分类统计	96	7.2 基本的文件操作	136
5.3 排序	96	7.2.1 文件的打开与关闭	136
5.3.1 选择排序法	97	7.2.2 文件的管理操作语句	137
5.3.2 冒泡排序法	98	7.3 文件的应用	137
		7.3.1 新建文件	137

7.3.2 读取文件	140	9.2 高效的绘图操作	184
7.3.3 替换文件	143	9.2.1 Line 方法	184
7.3.4 追加文件	143	9.2.2 Circle 方法	188
7.4 应用举例	144	9.3 自定义坐标系统	190
习题	149	习题	196
第 8 章 对话框与菜单的设计	151	实验 A 编程环境及控件	198
8.1 对话框	151	实验 B 表达式与常用内部函数	202
8.1.1 通用对话框	151	实验 C 输入输出方法	204
8.1.2 用户自定义对话框	159	实验 D 分支结构	207
8.2 菜单	166	实验 E 循环结构	209
8.2.1 菜单编辑器	167	实验 F 穷举与递推	212
8.2.2 输入菜单命令的 Click 事件的 过程代码	169	实验 G 数组	214
8.2.3 窗口菜单	170	实验 H 子过程与函数	217
8.2.4 弹出菜单	171	实验 I 文件	219
习题	174	实验 J 对话框与菜单设计	221
第 9 章 绘图	176	实验 K 绘图	224
9.1 最基本的绘图操作	176	附录 1 VB 6.0 集成开发环境	226
9.1.1 VB 坐标系统	176	附录 2 窗体及常用控件介绍	234
9.1.2 PSet 方法	176	附录 3 程序的调试	241
9.1.3 Point 方法	180	附录 4 标准数据类型	244
9.1.4 Cls 方法	180	附录 5 常用函数	246
		附录 6 运算符与表达式	249



第 1 章

VB 程序与编程环境

学习导读:

■ 本章从观察一个程序入手, 开始动手编写程序, 从而对 VB 程序设计有一个基本认识。

- 了解可视化程序设计的基本概念。
- 掌握 VB 程序的结构和程序的工作方式。
- 熟悉 VB 6.0 编程环境。

一个在计算机中运行的程序可以用任何一种程序设计语言来编写。但无论使用何种语言, 在设计和编辑程序的源代码以及调试程序时都需要一个特定的环境平台。

本书通过一个实例先介绍 VB 6.0 编程环境。读者需要了解它的使用方法, 掌握建立工程、设计窗体、创建控件、编写代码、调试运行、保存文件以及最终生成可执行程序等一系列操作。

1.1 观察一个程序

让我们先来运行一个成绩计算器程序, 它对考试成绩与平时成绩按 7:3 的比例计算并输出总评成绩。该程序的运行界面如图 1.1 所示, 具体步骤介绍如下所示:

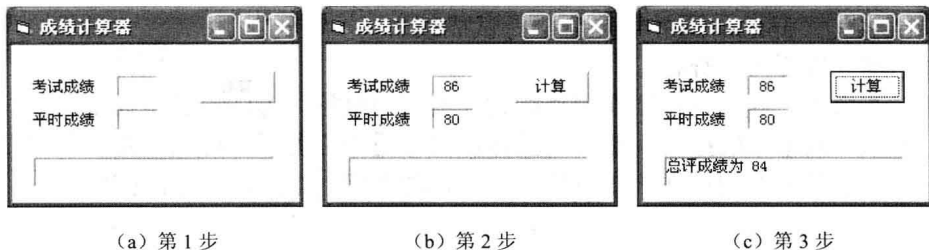


图 1.1 运行界面

- ① 运行程序, 此时只能输入考试成绩和平时成绩。
- ② 输入了考试成绩和平时成绩 (都是 0~100 的数) 后, “计算”按钮变为可用状态。

③ 单击“计算”按钮，显示出总评成绩。

可见这个程序先是要求输入数据，然后对输入的数据进行处理，最后将计算结果输出。

实际上，任何一个程序都要包括 3 方面的要素：输入原始数据、对得到的数据进行处理加工和输出计算结果，见图 1.2。

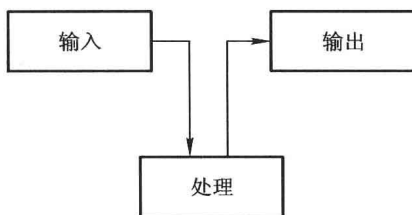


图 1.2 程序的三要素

1.2 动手编写程序

1. 界面设计

图 1.3 所示的程序界面的构成如下：①为窗体（Form），②、③为标签（Label），④、⑤为文本框（TextBox），⑥为图形框（PictureBox），⑦为命令按钮（CommandButton），⑧为定时器（Timer），定时器在程序运行时是看不到的。

下面我们就来建立程序界面。操作中所涉及的 VB 集成开发环境使用方法的介绍详见附录 1。

① 打开 VB 6.0，新建“标准 exe”工程，程序会自动生成一个窗体 Form1，在属性窗口中将窗体的 Caption 属性设置为“成绩计算器”。

② 在窗体上添加一个标签控件 Label1，在属性窗口中将其 Caption 属性设置为“考试成绩”，AutoSize 属性设置为“True”。

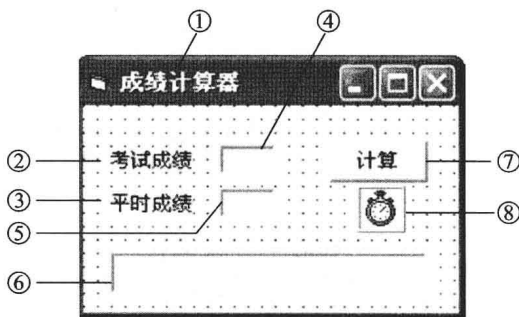


图 1.3 设计界面

注：在窗体上建立控件要在窗体的设计窗口中进行，方法一般有如下两种：

- 单击工具箱中的控件，然后用鼠标在窗体适当位置手工拖曳创建一个控件。
- 双击工具箱上的控件，在窗体中央会自动创建一个默认尺寸的控件。

③ 模仿②，建立其余的控件并摆放到恰当的位置。

- 标签 Label2: Caption 属性为“平时成绩”，AutoSize 属性为“True”。
- 文本框 Text1: Text 属性为空（在属性窗口中将 Text 属性清空），大小适当（用于输入考试成绩）。
- 文本框 Text2: Text 属性为空，大小适当（用于输入平时成绩）。
- 图形框 Picture1: 大小适当（用于输出总评成绩）。
- 命令按钮 Command1: Caption 属性为“计算”，Enabled 属性为“False”。

- 定时器 Timer1: Interval 属性为“500”（每 500 ms 发出一个 Timer 事件）。

2. 编写事件过程代码

设计好的一个程序界面就可以作为一个程序来运行，不过它此时不能实现任何功能，要让程序完成规定的功能，就要编写事件过程。例如要想在单击一个命令按钮时完成某种操作，就必须为它的 Click 事件编写过程。

事件过程代码的编写要在窗体的代码窗口中进行。事件过程代码结构的首行和末行是固定的，例如单击命令按钮 Command1 的事件过程的结构可以是：

```
Private Sub Command1_Click()  
<语句块>  
End Sub
```

事件

系统关键字，表明这是一个过程。

对象名

在窗体的设计窗口中双击一个对象，系统会自动切换到窗体的代码窗口并生成一个该对象最常用的事件过程的首行和末行，用户只需直接输入其中的“<语句块>”，其他方法详见附录 1。例如，双击 Timer1 就可以方便地完成下面代码的输入，其中灰色底纹部分是用户完成的，其他由系统自动生成。

```
Private Sub Timer1_Timer()  
Dim d1 As Boolean, d2 As Boolean  
d1=IsNumeric(Text1.Text) And Val(Text1.Text)>=0 And Val(Text1.Text)<=100  
d2=IsNumeric(Text2.Text) And Val(Text2.Text)>=0 And Val(Text2.Text)<=100  
Command1.Enabled=d1 And d2  
End Sub
```

成绩计算器程序的完整 VB 代码见图 1.4。关于代码的几点说明如下：



图 1.4 成绩计算器程序的代码

① 在程序中，单击“计算”按钮就可以完成计算和输出。计算依赖于输入的数据，但文本框中的内容都是字符型的，所以要用 Val 函数将它们转换成数值型。为使显示的结果为整数，使用了四舍五入函数（Round），它们的功能将在后面进行介绍。

② 在程序运行时我们看到，当文本框中没有输入数据时，“计算”按钮是禁用的，这是为了更好地保证输入的有效性。但是如何检查两个文本框中都输入了 0~100 内的数？如果输入完成后，如何启用“计算”按钮？在输入过程中如何判断有效数据变无效（例如输入 80 符合要求，输入 800 却不符合要求）怎么办？这一系列问题都需要由系统来处理，因此可以设置一个定时器，让它定时地去检查，这里我们设的是每秒 2 次。其中 `IsNumeric` 函数用来测试一个字符串是否是一个数值型数据。

3. 运行

单击工具栏中的“启动”按钮或执行菜单中的“运行”→“启动”命令，即可运行程序。程序运行后，显示程序窗体，输入考试和平时成绩，如果输入的都是 0~100 之间的数时，“计算”按钮就变为可用，单击它就会显示出所计算的总评成绩；否则，“计算”按钮就为禁用状态。

需要指出的是，这时的源程序是靠 VB 集成开发环境的解释实现运行的。如果代码输入有误，就不能正确运行，此时需要对程序进行跟踪调试，具体方法详见附录 3。

4. 保存工程

本程序涉及两个文件，一个是窗体文件（.frm），另一个是工程文件（.vbp）。首先要确定它们的文件名和保存路径，例如，分别以 `jscj.frm` 和 `jscj.vbp` 为文件名保存到文件夹 `E:\VBEX` 中。

执行“文件”菜单中的“保存”命令，或单击常用工具栏上的“保存”按钮。集成环境会先后弹出“文件另存为...”对话框和“工程另存为”对话框，提示用户输入工程中窗体文件名及工程文件名等，系统会以不同类型的扩展名将它们保存到磁盘指定位置，如图 1.5 所示。



(a) 保存窗体对话框

(b) 保存工程对话框

图 1.5 保存窗体及工程对话框

如果用户保存的是已存过盘的文件，那么单击工具栏上的“保存”按钮时，系统直接用原名保存文件，而不再弹出“另存为”对话框。

5. 生成可执行文件

当源程序经过调试运行无误之后便可以通过编译方式生成一个可执行文件。方法为：执行“文件”菜单中的“生成 exe”命令，输入要生成的可执行文件名后单击“确定”按钮即可。这样生成的可执行文件就可以直接运行而不必依赖于 VB 编程环境了，它与本章一开始所运行的程序就没有任何区别了。

1.3 认识 VB 程序

1.3.1 可视化程序设计的基本概念

所谓可视化程序设计的主要特征，是程序界面本身的代码不需要编程者自行编写，而只需要用系统提供的工具在设计窗口中“画”出一个个对象并设置它的属性，从而显示特定的界面效果。由于在设计过程中，对象不是以代码形式而是以图形方式显示在界面上，所以称为可视的。

可视化程序设计的优越性在于界面是“画”出来而不是“写”出来的，所以可极大地提高程序设计的效率。但界面终归是界面，而实现程序功能的那部分代码还是需要程序设计人员自己来编写，因此这方面能力的训练正是本书更为重视的。

1. 窗体

窗体 (Form) 是 VB 中最重要的一个对象，它是建立应用程序界面的基础。窗体有自己的属性、事件和方法，可以在其上输出文本和图形。窗体还可以作为其他控件的容器，通过在窗体上添加各种控件，用户可以设计丰富多彩的程序界面。

2. 控件

控件是构成可视化界面的基本元素，常用的控件有文本框、标签、命令按钮、单选按钮等。每个控件都有各自的属性、事件和方法，其中最基本的属性为 Name (名称)，它是控件在程序中的标识。控件也称为对象。窗体也是一个控件，不过它是一个容器类控件，在其中可以放置其他控件。

3. 控件的属性、事件和方法

(1) 属性

各个控件的名称、大小、位置和其他性状特征都由其属性确定。控件属性名的意义一般都比较明确，例如 Caption 属性指的是标题，BackColor 属性为背景色，Style 属性为风格等。新建的控件都有默认属性值，如果需要对属性进行调整，可以重新设置属性值。实际上程序界面上的任何变化都是对一些控件的属性值重新设置的结果。

根据属性值设置的不同方式，控件属性可以分为以下 4 类。

① 设计属性：只能在设计阶段通过属性窗口设置，例如控件的 Name 属性，窗体的 MinButton (最小化按钮) 等。

② 运行属性：只能在代码中使用赋值语句进行设置，例如窗体的 CurrentX 属性。

③ 常规属性：可用两种方法进行设置，例如文本框的 Text 属性。

④ 只读属性：不能用任何方法对其进行设置，例如列表框的 ListCount 属性。

(2) 事件

控件的事件是指由系统事先设定的、能被控件识别和响应的操作。例如，鼠标单击 (Click)、双击 (DoubleClick)、改变 (Change)、获得焦点 (GotFocus)、键盘按下 (KeyPress) 等。控件的事件可以分为以下几类。

① 系统发生事件：由系统自身产生的事件，程序使用者无法控制，例如窗体加载时产生 `Form_Load` 事件。

② 用户操纵事件：指程序使用者的某个动作触发的事件，例如单击一个命令按钮触发该按钮的 `Click` 事件，改变一个窗体的大小触发该窗体的 `Resize` 事件等。

③ 定时器自动事件：定时器按规定的时间间隔自动触发自身的 `Timer` 事件。

④ 运行触发事件：只能由代码运行来触发的事件，例如标签的 `Change` 事件。

不同的控件能够识别的事件是不同的。例如文本框可以识别 `KeyPress` 事件，而标签则不能。

有时一个动作可以同时触发多个事件，例如在一个文本框中输入数据时就会同时触发该文本框的 `KeyPress` 和 `Change` 等多个事件。有的事件可以有多个触发渠道，例如窗体的 `Resize` 事件可以在用户改变窗体大小时触发，也可被窗体的 `Load` 事件触发。

（3）方法

方法是系统事先设计好的、控件可以直接完成的操作。例如，一个控件的 `Move` 方法可以改变其位置与大小，一个文本框的 `SetFocus` 方法可将光标移到自身。

过程的运行也是完成规定的操作，但方法与过程不同。过程需要编程人员编写，而方法是系统已经定义好并封装在控件内的，是控件的一部分，需要时在过程内直接调用即可。

最后简单地介绍面向对象的程序设计中类与对象的概念。我们习惯将研究处理的事物进行分类，比如说汽车、飞机、潜艇等。汽车作为一个类的特征是从一辆辆具体的汽车上抽象出来的，比如都有发动机型号、车轮个数、载重量规定等属性，有添加燃油、装车、卸车等事件，具有实施转向、加速与减速方法。作为一个对象的一辆汽车就是上述特征的具体化，如几个车轮、载重量几何等。一个类中不同对象的区别就在于具体属性上的差别。而不同类的对象则是在是否具有某个方面的属性、事件或方法上不同，例如飞机有升空的方法，汽车与潜艇则没有，潜艇有排水量属性，汽车则没有，汽车有爆胎事件，潜艇则没有。

对于类与对象的关系，如果简单地将“类”比喻为一枚图章，“对象”就是图章印出来的印记，它可以印在不同的位置，有颜色的变化等，但更重要的是，印记出自图章而不是图章。在面向对象的程序设计中，类是封装了一组规定的属性、方法和事件的一个模具，用它所产生的对象是程序中的一个实体，此对象具有该类规定的各种属性、方法和事件，用户还可以修改一些属性的默认值。

在 VB 集成开发环境中，工具箱中的图标就是系统已经设计好的标准控件类的符号，用户还可以加入某些外部的类，也可以自行创建新类。

1.3.2 VB 程序的结构

1. 工程

使用 VB 创建应用程序时，系统会根据应用程序的功能建立一系列的文件，比如常见的窗体文件（`.frm`）和二进制窗体文件（`.frx`，存储诸如图片之类的信息）。这些文件的关联信息（注：不是文件本身）就登记保存在一个称为“工程”的内部，工程文件的扩展名为 `.vbp`。我们通常说的一个 VB 程序就是指对应的工程，但其内部登记的各个文件都是独立保存的，如果在操作系统层面上对它们改动（移动、删除或重命名）将破坏工程的完整性。

VB 程序中至少要包含一个窗体，否则就不可能有任何对象了。

2. 窗体模块

一个窗体模块由用户界面和代码两部分组成，其中界面部分上面已介绍过了，现在来看代码部分。

代码分为不同的层级，每个层级都有固定的结构（框架）。例如，过程代码层级的结构为：

```
...Sub ...
    [声明部分]
    [操作部分]
End Sub
```

事件过程是可执行的代码的最小层级，即过程内部不能再包含过程。一个窗体中的全部代码也构成一个层级，它包括通用声明部分和一些过程层级。其中，通用声明部分必须放在代码部分的最上端，其他过程层级的顺序不分先后。

（1）通用声明部分

该部分用来声明窗体中的一些规则和公共数据等。在这部分声明的数据在整个窗体中都可以使用，而在单个过程中声明的只能在该过程中有效。

（2）事件过程

如 1.2 节例子中所介绍的，事件过程有固定的结构，其过程名的形式为：

<对象名>_<事件名>

当对象的一个事件发生后，应用程序就要在整个窗体层级内查找是否有相应的事件过程。如果有，就执行这个过程，没有就什么也不做。

书写代码时要遵循代码的语法规则，主要有：

① 代码不区分英文字母的大、小写。

② 一般在一行上写一条语句。但允许一行写多句，两条语句之间用冒号“:”分开，也允许一句分多行，但要在每个非结束行尾加上续行符，续行符由一个空格和一个下划线组成。

③ 需要时可以为一条语句加标号。语句标号必须放在一行之首，它可以是一个数字或一个标识符加冒号“:”，标识符的意义见第 2 章。

④ 允许使用注释语句。在代码中，注释语句是一种特殊的语句，其特殊性在于它是给人看的，不会被编译、执行。注释语句以 ASCII 字符的单引号（'）开头，它可以单独占一行或写在一条语句行的后面，对代码进行说明、标注。适当地使用注释语句可以增强程序代码的可读性，给自己或他人在阅读程序时提供帮助。由于注释是不会被编译执行的，所以不会对程序的运行产生任何影响。

一个功能简单的 VB 程序只用一个窗体模块就可以实现，功能要求复杂时，则可能需要多个窗体模块或其他模块，如标准模块（.bas）或类模块（.cls）。这里不再展开论述，需要时可以参考有关资料。

1.3.3 VB 程序的工作方式

VB 通过事件驱动的方式来实现对象的一个操作。为对象的某个事件编写过程后，它就具有“潜在”的功能，一旦被“召唤”就“一显身手”，这可以用一个升降电梯来类比。电梯通电

后即启动，但它并不是自主升降而是处于等待状态。一旦有人请求，它便做出响应。超载时它报警并拒绝运行。用户按关门按钮就关门，门打开的时间达到规定的时长也关门等。一次操作过程完毕便又处于等待状态。

VB 程序一般包含若干个功能独立的事件过程。程序启动后系统等待某个事件的发生，然后去执行处理此事件的事件过程，待事件过程执行完后，系统又处于等待事件发生的状态，这就是事件驱动程序的设计方式。例如 1.1 节中的应用程序，只有当用户单击“计算”按钮 (Command1) 时，才会触发 Command1_Click 事件，Command1_Click 过程才会执行计算操作。

对于只有一个窗体的程序，当关闭该窗体时，程序就自行结束运行了。另外，在 VB 中有一个专门结束程序运行的语句 End。在任何过程运行时，只要遇到 End 语句，整个程序的运行即告结束。

在使用程序时，许多事件是由用户操控的，而这些事件发生的顺序又决定了代码执行的顺序，因此应用程序每次运行时所经过的代码路径不一定是完全相同的。

了解了事件驱动的工作方式之后，在设计程序时就可按下面的思路来考虑问题：

- ① 将一个复杂的功能分解为若干个相对独立的子功能块时应以过程为基本单位。
- ② 一个过程应该分配给哪些对象的哪些事件。

仍以 1.1 节中的应用程序为例，计算与输出由“计算”按钮的单击事件过程来实现，决定禁用还是启用“计算”按钮则由一个定时器的 Timer 事件过程来实现。

最后要专门介绍定时器对象，因为它只是一个事件发生器。具体地说，它的作用就是自动按指定的时间间隔 (Interval) 周期性地触发定时器事件 (Timer)。定时器控件在设计时可见，运行时并不可见。一个定时器的禁用还是启用通过设置它的 Enabled 属性来完成。但启用一个定时器并不意味着它就自动开始，还要取决于是否“定了时间”。指定时间间隔就是为它的 Interval 属性设置一个大于 0 的毫秒 (ms) 数，等于 0 时就取消了定时。

例 1.1 设计一个消逝的进度条的应用程序，程序运行效果见图 1.6。

【解题思路】

用一个图形框 Picture1 作为进度条的初始形态，它的高度和宽度是固定的。在 Picture1 中再用一个长方形的控件 Shape1 模拟进度条的变化形态，它的高度是固定的，宽度从 Picture1 的宽度开始，每次缩减一个固定的正整数，使其宽度逐渐变小，直至 0 时即可。

【程序界面】

- ① 窗体 Form1: Caption: “模拟进度条”, Height: 2200, Width: 4700。
- ② 图形框 Picture1: Left: 300, Top: 300, Height: 375, Width: 4000。
- ③ 形状 Shape1: 置于容器控件 Picture1 内，属性为 Left: 0, Top: 0, Height: 320, Width: 4 000, Shape: 0, FillStyle: 0, FillColor: &H00FF0000&, BorderStyle: 0。
- ④ 命令按钮 Command1: Caption: “开始”。
- ⑤ 定时器 Timer1: Interval: 50, Enabled: False。



图 1.6 例 1.1 程序界面

【程序代码】

```
'=====
```

```
'EX-01: 消逝的进度条
```

```
'=====
```

```
Private Sub Command1_Click()
    Shape1.Width = Picture1.Width
    Timer1.Enabled = True
End Sub

Private Sub Timer1_Timer()
    Dim a As Integer
    a = Shape1.Width - Picture1.Width / 100
    Shape1.Width = IIf(a > 0, a, 0)
    Timer1.Enabled = (a > 0)
End Sub
```

1.3.4 需要进一步掌握的内容

一个 VB 程序制作的基本步骤分为建立工程、设计窗体界面、编写事件代码、调试运行、保存文件和生成可执行程序等多个环节。完成这些操作需要熟悉 VB 6.0 编程环境，因此，请在需要时反复阅读下面的内容。

附录 1 VB 6.0 集成开发环境

附录 2 窗体及常用控件介绍

附录 3 程序的调试

习 题

一、填空题

1. 启动 VB 6.0 后，就意味着要建立一个新_____，一个工程中至少含有一个窗体。当运行程序时，系统自动启动窗体的_____事件。
2. VB 采用_____的编程机制，任何事件过程代码的执行都是基于事件的触发。
3. _____是应用程序的对外接口，是其他控件的载体和容器。每个控件都具有的属性是_____。
4. 确定一个控件在窗体上位置的属性是_____和_____。