

江苏省高等学校计算机等级考试系列教材

21世纪高校教材

Visual Basic 程序设计教程

(2013年版)

江苏省教育厅组织编写

主编 牛又奇 孙建国

苏州大学出版社

江苏省高等学校计算机等级考试系列教材

Visual Basic 程序设计教程

(2013 年版)

主 编 牛又奇 孙建国

苏州大学出版社

图书在版编目(CIP)数据

Visual Basic 程序设计教程: 2013 年版 / 牛又奇, 孙建国主编. —苏州: 苏州大学出版社, 2014. 1
江苏省高等学校计算机等级考试系列教材
ISBN 978-7-5672-0529-1

I. ①V… II. ①牛… ②孙… III. ①
BASIC 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(2013)第 115606 号

Visual Basic 程序设计教程

(2013 年版)

牛又奇 孙建国 主编

责任编辑 周建兰

苏州大学出版社出版发行

(地址: 苏州市十梓街 1 号 邮编: 215006)

宜兴市盛世文化印刷有限公司印装

(地址: 宜兴市万石镇南漕河滨路 58 号 邮编: 214217)

开本 787 mm × 1 092 mm 1/16 印张 20.25 字数 506 千

2014 年 1 月第 1 版 2014 年 1 月第 1 次印刷

ISBN 978-7-5672-0529-1 定价: 34.00 元

苏州大学版图书若有印装错误, 本社负责调换

苏州大学出版社营销部 电话: 0512-65225020

苏州大学出版社网址 <http://www.sudapress.com>

江苏省高等学校计算机等级考试
系列教材编委会

顾 问 张福炎 孙志挥

主任委员 王 煌

副主任委员 叶晓风

委 员 (以姓氏笔画为序)

牛又奇 朱巧明 李 畅 严 明

吴乃陵 邵定宏 单启成 侯晓霞

殷新春 蔡 华 蔡正林 蔡绍稷

编者的话

经过将近三年的努力,“Visual Basic 程序设计教程”的最新修订工作终于完成了。在此对所有关心、支持和爱护我们的老师、朋友,特别是广大的读者,以及对我们的工作给予一贯全力支持的苏大出版社的各位编辑和领导,表示最衷心的感谢。

此次修订工作中,我们将江苏省高等学校计算机等级考试中心对计算机基础教学要“打好基础、引导教学”的意见作为指导方针,广泛地听取和采纳了众多高校第一线教师的意见和建议,参考了众多国内外相关教材,从中汲取了大量有益的营养。在此基础上,我们紧紧围绕普通高校非计算机专业“程序设计语言”课程的教学大纲,对原书逐字、逐句地进行了校核,特别是参照微软的Microsoft Visual Basic 6.0 中文版的手册,对原书中一些表述不够准确的地方一一做了订正。同时为了更切合当前教学的要求,对部分章节的内容也做了一些调整、补充与修改。我们还重新设计了每个章节的习题,使得习题的形式和内容更切合教学要求,也与等级考试的题型相吻合。

本次修订,保留了原书的第2章“程序设计与算法(问题求解)”。对于部分不再开设“大学计算机信息技术基础”课程的学校,建议在教授“程序设计语言”课程时适当向学生介绍本章节的内容,以便学生更好地学习掌握程序设计的方法。附录和带有“*”号的部分,属于阅读材料,主要供读者自学。

随着计算机系统的升级,在Windows 7系统的计算机上能否安装和使用VB 6.0,受到很多读者的关注。我们在本书的附录中,特别增加了如何在Windows 7系统的计算机上安装VB 6.0的方法和步骤。希望能对这部分的读者有所帮助。

南京农业大学的朱淑鑫老师在本次修订中做了大量具体的工作。

我们做出了努力,但深知学无止境,错误难免,敬请广大读者不吝指正,谢谢!

编者

2013年3月

第1章 Visual Basic 导论

- 1.1 Visual Basic 与 Windows (1)
- 1.2 Windows 程序:界面与事件驱动..... (2)
- 1.3 对象、属性与方法 (3)
- 1.4 一个 Windows 程序示例 (4)
- 习题 (8)

第2章 程序设计与算法(问题求解)

- 2.1 程序的基本组成:输入、处理与输出..... (9)
 - 2.1.1 计算机解题示例 (9)
 - 2.1.2 程序设计的一般步骤 (10)
- 2.2 算法与编程工具 (11)
- 习题 (15)

第3章 Visual Basic 的界面设计

- 3.1 创建窗体 (16)
 - 3.1.1 定制窗体属性 (18)
 - 3.1.2 窗体的显示、隐藏、装载和卸载 (19)
 - 3.1.3 Print 方法 (21)
 - 3.1.4 Move 与 Cls 方法 (22)
- 3.2 Visual Basic 的常用控件 (22)
 - 3.2.1 概述 (22)
 - 3.2.2 常用基本控件 (24)
- 3.3 制作菜单 (34)
 - 3.3.1 菜单概述 (34)
 - 3.3.2 创建菜单 (35)
 - 3.3.3 创建弹出式菜单 (36)
- 3.4 多窗体界面程序设计 (37)
- 3.5 界面设计程序示例 (38)
- 习题 (40)

第4章 数据、表达式与简单程序设计

- 4.1 Visual Basic 程序代码的组织方式 (44)
 - 4.1.1 过程 (44)

4.1.2 模块	(45)
4.2 代码行的书写规则	(46)
4.3 Visual Basic 的数据	(46)
4.3.1 数据类型	(47)
4.3.2 常量	(48)
4.3.3 变量	(49)
4.4 运算符与表达式	(52)
4.4.1 算术运算符与算术表达式	(52)
4.4.2 关系运算符与关系表达式	(53)
4.4.3 逻辑运算符与逻辑表达式	(53)
4.4.4 运算规则	(54)
4.5 赋值语句	(54)
4.6 Visual Basic 公共函数	(56)
4.6.1 算术函数	(56)
4.6.2 字符函数	(57)
4.6.3 转换函数	(57)
4.6.4 日期与时间函数	(59)
4.6.5 格式化函数 Format	(59)
4.7 InputBox 函数与 MsgBox 函数	(60)
4.7.1 InputBox 函数	(60)
4.7.2 MsgBox 函数	(61)
习题	(63)
第5章 选择分支与循环	
5.1 分支结构与分支结构语句	(67)
5.1.1 If-Then-Else-End If 结构语句	(67)
5.1.2 IIf 函数	(70)
5.1.3 Select-Case-End Select 结构语句	(70)
5.2 循环结构与循环结构语句	(72)
5.2.1 Do-Loop 循环结构语句	(72)
5.2.2 For-Next 循环结构语句	(74)
5.2.3 循环嵌套	(77)
习题	(85)
第6章 数组	
6.1 数组的概念	(91)
6.1.1 数组命名与数组元素	(91)
6.1.2 数组定义	(92)
6.1.3 数组的结构	(94)
6.1.4 数组维界测试函数	(96)
6.2 数组的基本操作	(97)

6.2.1 数组元素的赋值	(97)
6.2.2 数组元素的输出	(102)
6.2.3 数组元素的引用	(105)
6.3 动态数组	(106)
6.4 控件数组	(110)
6.4.1 基本概念	(110)
6.4.2 建立控件数组	(111)
6.4.3 使用控件数组	(111)
6.5 程序示例	(114)
习题	(132)
第7章 过程	
7.1 过程的分类与引例	(138)
7.2 Sub 过程	(139)
7.2.1 事件过程	(140)
7.2.2 通用过程	(142)
7.3 Function 过程	(144)
7.4 过程调用	(146)
7.4.1 事件过程的调用	(146)
7.4.2 通用 Sub 过程调用	(147)
7.4.3 Function 过程调用	(148)
7.4.4 调用其他模块中的过程	(149)
7.5 参数的传递	(150)
7.5.1 形参与实参	(150)
7.5.2 按值传递参数	(151)
7.5.3 按地址传递参数	(152)
7.5.4 数组参数	(156)
7.5.5 对象参数	(157)
7.6 递归过程	(158)
7.7 变量的作用域	(160)
7.7.1 过程级变量	(160)
7.7.2 模块级变量	(160)
7.7.3 全局变量	(161)
7.7.4 关于同名变量	(163)
7.7.5 静态变量	(164)
7.8 程序示例	(165)
7.9 创建与设置启动过程	(179)
7.10 鼠标与键盘事件及事件过程	(184)
7.10.1 鼠标事件及鼠标事件过程	(184)
7.10.2 键盘事件及键盘事件过程	(187)

习题	(188)
第8章 数据文件	
8.1 数据文件处理	(196)
8.1.1 数据文件概述	(196)
8.1.2 访问文件的语句和函数	(197)
8.2 顺序文件	(202)
8.2.1 顺序文件的写操作	(202)
8.2.2 顺序文件的读操作	(204)
8.2.3 使用外部程序处理顺序文件	(208)
8.3 随机文件	(210)
8.3.1 变量声明	(210)
8.3.2 随机文件的打开	(211)
8.3.3 随机文件的写操作	(211)
8.3.4 随机文件的读操作	(214)
8.3.5 增加、删除随机文件中的记录	(215)
8.4 二进制文件	(216)
习题	(218)
第9章 程序调试	
9.1 错误类型与程序调试工具	(220)
9.1.1 错误类型	(220)
9.1.2 VB 调试工具	(221)
9.2 程序调试	(222)
9.2.1 中断状态的进入与退出	(222)
9.2.2 使用调试窗口	(223)
9.2.3 断点设置与单步调试	(224)
习题	(226)
第10章 文件管理与公共对话框控件及其应用	
10.1 文件管理控件	(229)
10.1.1 目录(文件夹)列表框	(230)
10.1.2 文件列表框	(232)
10.1.3 组合使用文件管理控件	(234)
10.2 公共对话框	(237)
10.2.1 概述	(237)
10.2.2 公共对话框控件的应用	(237)
习题	(245)
第11章 图形处理及多媒体应用	
11.1 图形处理	(246)
11.1.1 坐标系统	(246)
11.1.2 自定义坐标系	(247)

11.1.3	色彩函数	(248)
11.1.4	使用绘图控件	(248)
11.1.5	使用绘图方法	(251)
11.1.6	使用图片框	(258)
11.1.7	应用鼠标事件	(260)
11.2	多媒体应用	(262)
11.2.1	动画控件	(262)
11.2.2	多媒体控件	(265)
习题		(268)
第 12 章 数据库操作与编程		
12.1	数据库基本知识	(269)
12.1.1	概述	(269)
12.1.2	数据库的基本概念	(269)
12.1.3	数据模型	(270)
12.2	数据库的建立	(271)
12.2.1	关系数据库的基本结构	(271)
12.2.2	数据库的建立	(272)
12.2.3	建立查询	(275)
12.3	数据控件	(275)
12.3.1	数据控件及其属性	(275)
12.3.2	应用数据控件	(276)
12.3.3	数据库操作	(277)
12.4	结构化查询语言 SQL	(282)
12.4.1	SQL 的基本组成	(282)
12.4.2	SQL 语句应用	(283)
12.5	数据处理	(285)
12.5.1	数据窗体向导	(285)
12.5.2	报表设计	(287)
12.6	ADO 数据访问对象	(289)
12.6.1	ADO 对象模型	(289)
12.6.2	ADO Data 控件	(290)
12.6.3	ActiveX 数据对象	(293)
12.6.4	应用示例	(299)
附录 1	Visual Basic 的集成开发环境	(304)
附录 2	Visual Basic 的帮助系统	(309)
附录 3	在 Windows 7 系统上安装 VB 6.0	(312)

第1章

Visual Basic 导论

Visual Basic 程序设计语言是用于开发 Windows 环境下应用程序的重要工具。对于学习者而言,了解和掌握面向对象的程序设计方法以及程序的事件驱动这两个 Visual Basic 语言的重要特性十分重要。

1.1 Visual Basic 与 Windows

Visual Basic(以下简称 VB)是用于开发和创建 Windows 操作平台下具有图形用户界面的应用程序的强有力工具之一。它以人们所熟知的 BASIC 语言(Beginners All-purpose Symbolic Instruction Code,初学者符号指令代码)为基础,不仅易于学习、掌握,它的可视化(Visual)特性,还为应用程序的界面设计提供了更迅速、便捷的途径。它不需编写大量代码去描述界面元素的外观和位置,而只要把预先建立的可视对象拖放到屏幕上的一点即可。VB 同时还是一个包括编辑、测试和程序调试等各种程序开发工具的集成开发环境(Integrated Development Environment, IDE),从应用程序的界面设计、程序编码、测试和调试、编译并建立可执行程序,直到应用程序的发行,种种功能 VB 无不包容。不论是 Microsoft Windows 应用程序的资深专业开发人员还是初学者,Visual Basic 都为他们提供了完整的开发工具。

VB 包含了数百条语句、函数及关键词,其中很多和 Windows GUI(Graphic User Interface,图形用户界面)有直接关系。专业人员可以用 VB 实现其他任何 Windows 编程语言的功能,而初学者只要掌握几个简单语句就可以建立实用的应用程序。

VB 是 Microsoft Office 系列应用程序通用的程序设计语言。在 Windows 操作系统已经成为 PC 事实上的标准之时,Microsoft 公司的 Office 系列,包括 Word、Excel、PowerPoint 和 Access 数据库也已成为人们在使用 PC 处理办公室事务时的首选软件。所以,学习和掌握 VB 对于充分发挥 Office 系列软件的各项功能,具有不可替代的作用。

VB 全面支持 Windows 系统的 OLE(Object Linking and Embedding,对象的链接和嵌入)技术。因而可在不同的应用程序之间快速地传递数据,并自动地利用其他应用程序所支持的各种功能。

VB 为开发 Windows 应用程序不仅提供了全新的、相对简单的方式,而且引进了新的程序设计方法——面向对象的程序设计方法(Object-Oriented Programming, OOP)。从传统的

面向过程的程序设计,转移到采用更先进的面向对象的程序设计,无论是对老的程序员,还是初学者,都是一个挑战。而学习 VB,则是掌握这一新的程序设计方法的一条捷径。

现在,许多大型的商品化软件也都是采用 VB 平台开发的。学习和掌握 VB,已成为现代社会对信息技术人才的需求之一。

1.2 Windows 程序：界面与事件驱动

Windows 下的应用程序的用户界面都是由窗体、菜单和控件等对象构成的。图 1-1 是人们熟知的“写字板”程序的窗口界面。窗口中有命令菜单,工具栏中包含有多个命令按钮,列表框和作为工作区的文本框等。

与所有 Windows 下的应用程序相同,在使用该程序时,可以自由地不分先后地使用菜单命令或工具栏中的命令按钮,对工作区中输入的文字进行各种处理操作。而各个对象的动作以及各对象之间的关联,完全取决于操作者所做的操作。比如,用鼠标单击“文件”,就会打开“文件”菜单,选定某个菜单命令再单击,该命令就会执行。也就是说,程序的运行并没有固定的顺序。Windows 程序的这种工作模式,被称为事件或消息驱动方式。

所谓“事件”,就是使某个对象进入活动状态(又称激活)的一种操作或动作。比如,用鼠标单击窗体上菜单条的某个命令项,或双击窗体上的某个图标等,就会打开相应的下拉式命令菜单或打开该图标对应的窗口。鼠标的单击和双击都是“事件”。只要程序设计者为某对象在某个事件发生时计算机应当执行的各种操作进行了规定,计算机就会执行这些操作。

用一个“事件”激活某个对象,随着该对象的活动,会引发新的“事件”,这个事件又可能使另一个“对象”激活,对象之间就是以这种方式联系在一起。

VB 并不仅仅只是 Windows 环境下运行的一种语言,它与 Windows 有着非常紧密的联系。VB 中的窗体对应于 Windows 的窗口;VB 的文本框、标签、列表框等控件对应于 Windows 窗口中的相应组件。VB 对象的属性与事件,则对应于 Windows 窗口组件的属性与消息。VB 的系统对象如菜单、剪贴板、屏幕和打印机提供了访问 Windows 系统功能的途径。

使用 VB 不仅可以非常便捷地设计出 Windows 应用程序的窗口界面,设置界面中各种对象的属性,而且可以通过编写程序代码段为对象规定在被某个“事件”激活时应发生的各种动作以及所要进行的信息处理的具体内容,这样的代码段称为“事件过程”。为不同对象响应不同事件编写的事件过程是构成一个完整应用程序不可缺少的组成部分。这就是事件驱动方式的应用程序的设计原理。

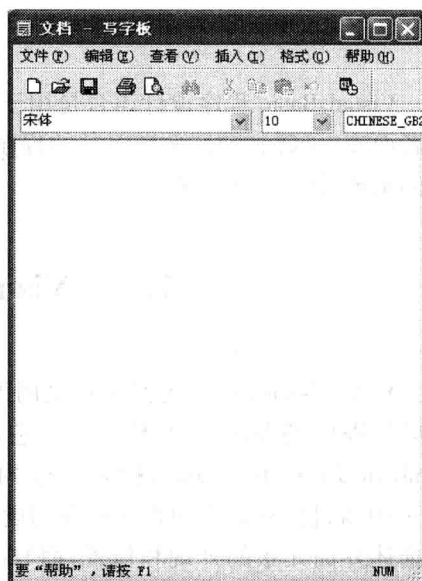


图 1-1

1.3 对象、属性与方法

VB 是一种采用面向对象的程序设计方法的语言。因此了解面向对象的程序设计方法,对于学习和掌握 VB 十分重要。那么,面向对象的程序设计方法究竟是一种什么样的方法呢?

面向对象的程序设计(OOP)是近年来发展起来的一种新的程序设计思想。计算机程序本是对现实世界的模型化,而现实世界则是由一个一个动作主体构成的,一个复杂的动作主体又由若干简单的动作体组成。比如,一辆汽车是一个动作主体,汽车又是由诸如发动机、传动系统、转向系统、刹车系统、车轮等动作体组合而成。在使用计算机程序描述一辆汽车的动作的时候,如果着眼点是汽车从一地到另一地的运动过程,这就是传统的“面向过程的程序设计思想”;如果着眼点是组成汽车的一个个部件,即动作体的特性、工作规律和动作方式,通过对这些动作体的描述,进而确定整个汽车的工作特性和规律,这种程序设计思想就是所谓的“面向对象的程序设计思想”。显然,面向对象的程序设计思想是对现实世界的更精确的反映。

1. 对象及对象类

动作体的逻辑模型称为“对象”。在 VB 中,对象就是人们可控制的某种东西。

对象类是对象的正式定义。比如,我们在说“汽车”时,并不是专指某个特定品牌的汽车,而是指一切装有内燃式发动机,有传动装置、转向装置、车轮等的可运载人或物的交通工具。而一辆具体的汽车,则是“汽车”的一个实例。

Windows 下的应用程序界面都是以窗口的形式出现的,窗口就是代表屏幕上某个矩形区域的对象,一个窗口可能包容其他窗口,这些被包容的窗口称为子窗口。在 VB 中,把这种窗口的界面称为“窗体”。在窗体上,可以设置用于和用户交互的各种部件,如文本框(TextBox)、标签(Label)、命令按钮(CommandButton)、选项按钮(OptionButton)和列表框(ListBox)等,这些部件统称为“控件”。应用程序的每个窗体和窗体上的种种控件都是 VB 的对象。

2. 属性

“属性”用来描述对象的特性。对象类定义了类的一般属性,比如汽车轮胎的一般属性包括由橡胶制成、中空充气等。就具体的对象而言,除要继承对象类规定的各种属性(称为继承性)之外,还具有它的特殊属性,如轮胎直径的大小、厚度、胎面的花纹等。规定了对象的特殊属性,也就真正将这个对象“实例化”了。

VB 为每一类对象都规定了若干属性。比如,窗体的属性就有显示方式、背景颜色、边框线型、窗体名称、窗体标题、前景颜色、大小位置和可见性等。通过为窗体设置具体的属性值,就可获得所需要的窗体外观及相关特性,如窗口的行为以及如何对按键及鼠标事件进行响应等。

3. 方法

“方法”指对象可以进行的动作或行为。人们可以通过“方法”使对象以指定的方式去做某种动作或改变行为。比如,通过“转向”方法使方向盘对象旋转,从而使车轮转往规定

的方向。

VB 程序中每个窗体或控件对象,都具有若干可改变其行为或实现某个特定动作(操作)的方法。例如,窗体可被“显示”或被“隐藏”等。显示(Show)和隐藏(Hide)都是控制窗体对象的方法。

1.4 一个 Windows 程序示例

VB 是一个功能强大而又易于操作的开发环境,它为 VB 应用程序的开发提供了极大的便利。

按照 VB 用户指南的说明,可非常容易地将 VB 系统安装到用户计算机的硬盘上。在 Windows 9X/XP 下,启动 VB,在显示版权页之后,稍待片刻,屏幕就会出现 VB 集成开发环境(IDE)的主画面(图 1-2)。不同版本的 VB 的主画面略有差别,图 1-2 是 VB 6.0 的画面。

VB 集成开发环境的主画面是一个典型的 Windows 界面,它由标题条、菜单条、弹出式菜单(又称上下文菜单)、工具栏、控件工具箱、初始窗体和工程资源管理器子窗口、属性子窗口、窗体布局子窗口等组成。VB 系统还有几个在必要时才会显示出来的子窗口,即“代码编辑器”窗口和用于程序调试的“立即”、“本地”和“监视”窗口等(有关 VB 集成开发环境详情,请参看本书附录 1)。

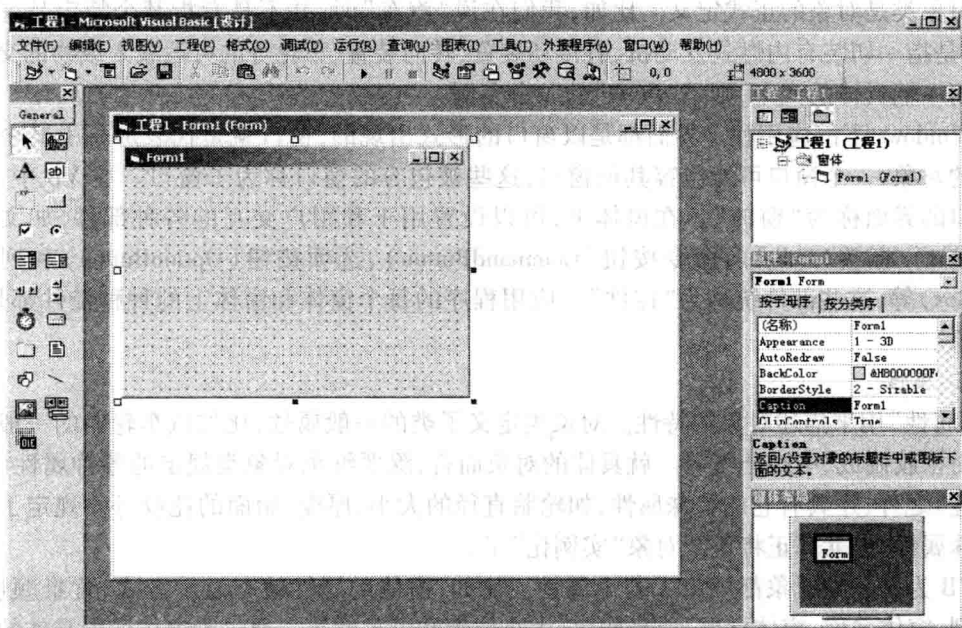


图 1-2

在 VB 中,创建一个应用程序,被称为建立一个工程。一个 VB 应用程序是由若干个不同类型的文件组成的。工程就是这些文件的集合。启动 VB 时,系统总是开始一个称为“工程 1”(Project1)的新工程。

【例 1-1】图 1-3 是本例的程序界面。在窗口中有一行文字“你好!”和一个命令按钮。

用鼠标单击命令按钮,窗口中的文字就会自动变成“欢迎学习 VB!”。

1. 启动 VB,开始新工程

在 Windows 环境下,启动 VB,如果主画面上没有出现窗体编辑器窗口,可用鼠标单击工具栏上的“新窗体”按钮。

2. 创建用户界面

本程序只需要一个窗口来与用户交互。现在就使用显示在 VB 主画面的窗体编辑器来创建这个窗口。

(1) 设置窗体属性:在属性窗口中先选定窗体对象,再将属性列表框的“Caption”(标题)属性改为“程序示例”。

(2) 为窗体增加控件和设置属性。

示例中的文字是由“标签”控件提供的。用鼠标单击控件箱的“标签”按钮,然后移动鼠标光标到窗体的适当位置(此时鼠标光标为十字形),再按住鼠标左键拖动,得到一个大小合适的矩形框(图 1-4)。

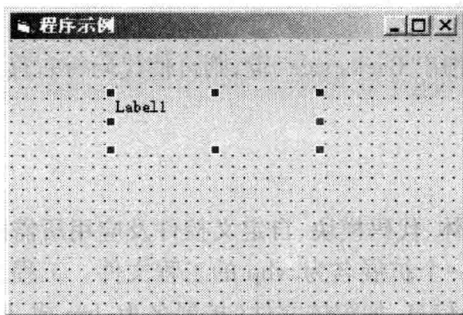


图 1-4

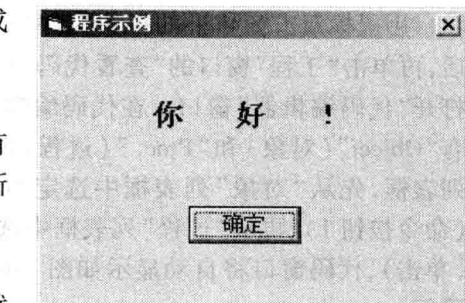


图 1-3



图 1-5

在属性窗口为“标签”设置属性:滚动属性窗口,将 Caption(标题)属性设为“你好!”,将 AutoSize 属性设为“True”。方法是:双击该属性项,再从给出的两个值中选定(图 1-5)或单击该属性表行最右侧的列表按钮,再从选项列表中选定。设定本属性可使标签的大小能自动调节,以容纳相应的信息内容。另外,用户还可通过设置 ForeColor 和 Font 等属性,以改变标签文字的颜色、字体及字号。本例设置为二号楷体字。

设置完毕,在标签外部单击鼠标或按回车键。

用同样的方法,为窗体再增加一个命令按钮。将命令按钮的 Caption 属性设为“确定”。

在为窗体增加控件时,控件的位置和大小可从工具栏右侧显示的坐标信息中获知。该坐标的单位是“Twip”(特维,1Twip = 1/1440 英寸)。

3. 加入程序代码

本程序所要响应的事件是用鼠标单击命令按钮。

用鼠标双击窗体上的按钮控件(或单击按钮后,再单击“工程”窗口的“查看代码”按钮),系统打开“代码编辑器”窗口。在代码编辑器窗口中,有“Object”(对象)和“Proc.”(过程)两个下拉式列表框,先从“对象”列表框中选定“Command1”(命令按钮1),再从“过程”列表框中选定“Click”(单击),代码窗口将自动显示如图 1-6 所示的代码行。



图 1-6

第一行代码表示这是命令按钮 Command1 响应单击事件的过程,下面一行是过程的结束行。两行之间可添加具体的用以响应单击事件的程序代码。将鼠标在两行中间的空白行处单击,并输入以下代码:

```
Label1.Caption = "欢迎学习 VB!"
```

代码编辑器是一个典型的文本编辑系统,其使用方法与其他文本编辑程序基本类似。

代码输入的基本规则:

- 按行输入,一行输入完,按回车,光标指向下一行,可接着输入下一行代码;
- 输入英文字母可不分大小写(用双引号括起来的文字除外);
- 代码行中所有有意义的符号均为西文符号;
- 使用“Tab”键,可使代码行向左缩进;使用“Backspace”键,则可使代码行右移。

4. 保存工程文件

代码输入完后,就可以保存工程了。

一个 VB 程序也称为一个工程。它是由窗体、代码模块、自定义控件及应用所需的环境设置组成的。在设计一个应用时,系统会建立一个扩展名为 .vbp 的工程文件。工程文件列出了在创建该工程时所建立的所有文件的相关信息,如窗体文件(扩展名为 .frm 或 .frx),它包括有窗体、窗体上的对象及窗体上的事件响应代码;标准模块文件(扩展名为 .bas),它包含有可被任何窗体或对象调用的过程程序代码,标准模块文件在一个工程中是可选的。除此之外,一个工程还可包括自定义控件文件(扩展名为 .ocx)、Visual Basic 类模块(扩展名为 .cls)、资源文件(扩展名为 .res)和用户文档(扩展名为 .dob 或 .dox)等。

保存工程时,系统将把该工程的所有相关文件一起保存;在打开一个工程文件时,系统也将把该工程文件中列出的所有文件同时装载。

为了使用和管理方便起见,建议把一个工程存储在一个独立的文件夹内。

注意,工程文件并不包含相关的文件和模块本身。因此可以在多个工程中使用同一个窗体文件。

使用“文件”菜单中的“保存工程”命令,在打开的“文件另存为”对话框中(图 1-7),先给窗体取个名字保存,再在“工程另存为”对话框中给工程取个名字,最后单击“保存”按钮即可。

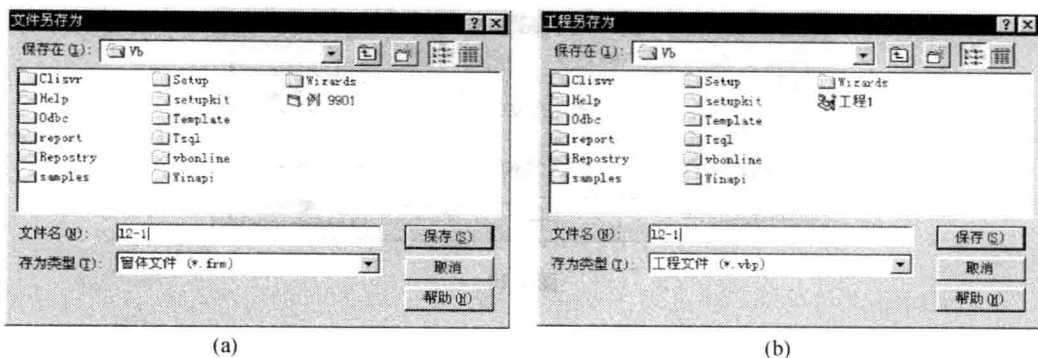


图 1-7

5. 运行及调试程序

这是一个十分简单的程序。用鼠标单击工具栏上的“运行”按钮，程序即显示出如图 1-3 所示的窗口。注意，此时 VB 环境的标题已从设计态变成了运行态。用鼠标单击窗口中的命令按钮，即可看到窗口中文字的变化(图 1-8)。

用鼠标单击工具栏的“停止”按钮，程序运行结束，系统又回到设计态。

如果程序在运行时发生错误，或实现不了预定的功能，或界面外观不够理想，用户可进行修改。修改完毕，可再次保存。

有关程序调试的方法，后面将详细介绍。

6. 生成可执行程序

使用“文件”菜单中的“生成[工程名].exe...”(建立可执行文件)命令，即可把设计完成并经过调试的工程编译成可脱离 VB 环境独立运行的可执行程序。VB 6.0 的创建可执行文件命令可把当前打开的工程名自动填入该命令。

7. 打印窗体和代码

如果用户希望打印当前窗体、窗体中的代码，甚至整个工程的所有窗体和程序代码，应首先使用“文件”菜单中的“打印设置”命令，选择打印用的打印机及相关参数(图 1-9)；然后再使用“文件”菜单中的“打印”命令，在“打印”对话框(图 1-10)中，设定打印的范围、打印对象及打印质量等后，单击“确定”按钮即可。

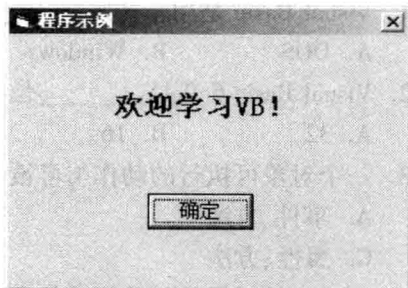


图 1-8

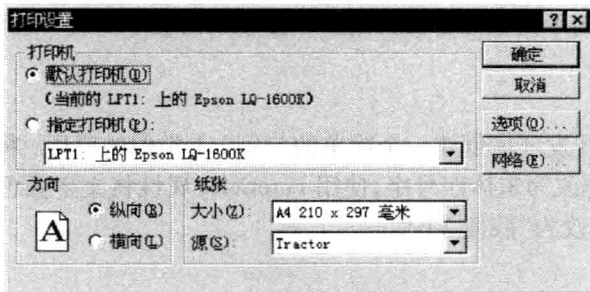


图 1-9