



通用程序设计案例丛书

Visual Basic

程序开发

案例解析

杨富国 主编

王健 王倩 陈立俊 编著



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



北京交通大学出版社

<http://press.bjtu.edu.cn>

通用程序设计案例丛书

Visual Basic 程序开发案例解析

杨富国 主 编
王 健 王 倩 陈立俊 编 著

清华大学出版社
北京交通大学出版社

• 北京 •

内 容 简 介

本书通过程序开发案例系统完整地介绍了 Visual Basic 编程基础和软件开发过程, 提供了分析应用系统中典型案例设计的思路、方法、技巧和经验。从开发平台的特点出发, 依照软件工程设计规范, 对每个案例进行需求分析、系统建模、界面设计、模块开发说明和系统实现全面加以介绍。

本书精选的案例范围涉及信息管理系统、网络应用系统、图形图像系统、多媒体系统、游戏开发等常用领域, 并提供了全部案例的完整代码供读者引用和借鉴。

本书内容翔实、案例丰富、实践性强, 可作为广大程序员和编程爱好者学习 Visual Basic 程序开发和应用系统设计的教材, 也可作为相关专业大学生及程序开发人员的参考书。

版权所有, 翻印必究。举报电话: 010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

本书防伪标签采用特殊防伪技术, 用户可通过在图案表面涂抹清水, 图案消失, 水干后图案复现; 或将表面膜揭下, 放在白纸上用彩笔涂抹, 图案在白纸上再现的方法识别真伪。

图书在版编目 (CIP) 数据

Visual Basic 程序开发案例解析/杨富国主编. —北京: 清华大学出版社; 北京交通大学出版社, 2006. 1

(通用程序设计案例丛书)

ISBN 7-81082-674-3

I. V… II. 杨… III. BASIC 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 147760 号

责任编辑: 招富刚

出版者: 清华大学出版社 邮编: 100084 电话: 010-62776969

北京交通大学出版社 邮编: 100044 电话: 010-51686414

印刷者: 北京瑞达方舟服务有限公司

发行者: 新华书店总店北京发行所

开 本: 185×260 印张: 22.25 字数: 556 千字

版 次: 2006 年 2 月第 1 版 2006 年 2 月第 1 次印刷

书 号: ISBN 7-81082-674-3/TP·253

印 数: 1~5000 册 定价: 29.00 元

本书如有质量问题, 请向北京交通大学出版社质监组反映。对您的意见和批评, 我们表示欢迎和感谢。

投诉电话: 010-51686043, 51686008; 传真: 010-62225406; E-mail: press@center.bjtu.edu.cn。

编委会成员名单

主 编：杨富国

编 委：（按姓氏笔画排序）

王 浩 王 倩 王 健

吕江英 邢 霞 陈立俊

杨茂龙 唐巧琪

前 言

Visual Basic 是 Microsoft 公司推出的系列开发软件之一，它以实用、方便、快捷、开发周期短、功能强大等特点越来越被广大编程人员所青睐。本书首先对 Visual Basic 程序开发的理论基础和主要技术作概略性介绍，紧接着阐述了软件工程设计规范，全书的重点放在其后的应用案例分析部分。在数据库信息管理、网络数据通信、图形图像处理、多媒体应用、游戏程序设计等常用领域中，各列举一两个较为复杂的案例，按照软件工程设计流程，全面地进行分析，力求从系统分析到程序开发的整个过程中，找出简便、清晰、快捷的实现方案，使程序开发人员熟悉和掌握系统分析与设计的思路和方法。

本书的特点是理论紧密联系实际，书中所列举的案例应用范围较广，涵盖了 Visual Basic 系统应用的大部分领域，对于正在进行课题设计或项目开发的读者来说，很多案例稍加修改就可直接应用。我们还特别注意了案例选择的适度，案例过小则不能体现软件 engineering 理论，不能为开发项目的工程人员和毕业设计的学生所借鉴；案例过大则覆盖面小，书稿容量受限。因此，书中案例难度和规模与计算机专业本科生毕业设计的要求相当。这些案例具有一定的系统性和可扩展性，覆盖面宽，重点突出，尽量避开了编程过程中一些旁枝末节的困扰，有助于读者系统开发能力的快速提高。

本书将软件工程理论贯穿全书，着眼于提高读者系统分析与开发能力。全书避免泛泛地介绍程序设计语言的基础理论，也不是简单地罗列各式各样的零散实例，而是力求从开发平台的特点出发，依照软件工程设计规范，对每个案例进行需求分析、系统建模、界面设计、模块开发说明、系统实现与编译等，内容覆盖软件开发的整个过程，以帮助读者建立起系统分析的概念和规范性设计的思想，从而提高程序开发的水平。

全书共分七章，第 1 章是关于 Visual Basic 的一些基础知识，初学 Visual Basic 的用户可以通过该章了解到 Visual Basic 开发平台一些概念、基础和程序设计方法，有一定基础的读者可以通过该章对 Visual Basic 进行复习，以利于后面章节的学习；第 2 章主要介绍软件工程的基本知识；第 3 章到第 7 章是本书的重点，主要是通过五个不同应用方向的介绍和七个案例的开发向读者展示如何应用 Visual Basic 开发应用系统，其中第 3 章是关于数据库管理信息系统方面的案例，第 4 章是关于网络数据通信方面的案例，第 5 章是关于图形图像处理方面的案例，第 6 章是关于多媒体应用方面的案例，第 7 章是关于游戏开发方面的案例。读者可以从北京交通大学出版社网站上下载随书所附案例的完整代码和可执行程序。本书力争通过这些典型案例的开发过程介绍，向读者展示 Visual Basic 的强大功能和软件开发的基本过程。

本书结合编者多年程序开发的实践和经验编写而成。全书由杨富国教授总策划，王健、王倩、陈立俊执笔，最后由杨富国教授统稿、审定。此外，唐巧琪、王浩、吕江英在程序编

写、资料整理、内容审定方面做了大量工作，给予编者很大的支持和帮助，邢霞教授、杨茂龙博士对本书的编写进行了指导，并提出了不少的建议。

由于编者水平有限，加之时间仓促，难免有纰漏之处，敬请读者批评指正。

编 者

2006 年 2 月

目 录

第 1 章 Visual Basic 编程基础	(1)
1.1 Visual Basic 编程特点	(1)
1.1.1 Windows 工作方式	(1)
1.1.2 事件驱动程序	(2)
1.1.3 面向应用的开发	(3)
1.2 Visual Basic 开发环境	(4)
1.2.1 Visual Basic 开发环境组成	(4)
1.2.2 窗体对象的作用	(9)
1.2.3 添加控件与代码	(10)
1.2.4 使用 ActiveX 控件	(12)
1.3 常用控件介绍	(12)
1.3.1 命令按钮 (CommandButton) 控件	(12)
1.3.2 文本框 (TextBox) 控件	(13)
1.3.3 标签 (Label) 控件	(14)
1.3.4 选择按钮 (OptionButton) 控件	(14)
1.3.5 下拉列表框 (ListBox) 控件	(15)
1.3.6 时间 (Timer) 控件	(16)
1.4 编程举例	(17)
1.4.1 “Hello World!” 程序	(17)
1.4.2 图片显示系统	(19)
第 2 章 软件开发过程	(23)
2.1 概述	(23)
2.1.1 瀑布模型	(24)
2.1.2 原型模型	(24)
2.1.3 螺旋模型	(25)
2.2 软件定义及计划	(26)
2.2.1 问题定义	(26)
2.2.2 可行性分析	(26)
2.2.3 制定开发计划	(27)
2.3 需求分析	(27)
2.3.1 需求分析任务	(27)

2.3.2	需求分析过程	(27)
2.3.3	常用需求分析方法	(28)
2.3.4	需求文档	(29)
2.4	软件设计	(30)
2.4.1	设计目标	(30)
2.4.2	设计任务	(30)
2.4.3	软件设计过程	(31)
2.4.4	软件设计方法	(32)
2.5	编码	(33)
2.5.1	概述	(33)
2.5.2	程序设计风格	(33)
2.5.3	编码标准	(34)
2.6	软件测试	(34)
2.6.1	概述	(34)
2.6.2	测试目的和原则	(35)
2.6.3	测试方法	(35)
2.6.4	软件测试过程	(36)
2.7	软件维护	(37)
2.7.1	概述	(37)
2.7.2	影响维护工作量的因素	(38)
2.7.3	软件的可维护性	(39)
2.8	小结	(39)

第3章	数据库管理信息案例	(40)
3.1	管理信息系统基础	(40)
3.1.1	管理信息系统的概述	(40)
3.1.2	管理信息系统的特点	(42)
3.1.3	管理信息系统的开发	(42)
3.2	SQL 与 ADO 概述	(43)
3.2.1	SQL 语言	(43)
3.2.2	ADO 数据对象	(45)
3.3	学生信息管理系统	(50)
3.3.1	需求分析	(50)
3.3.2	系统设计	(51)
3.3.3	数据库设计	(52)
3.3.4	界面设计	(54)
3.3.5	系统实现	(72)
3.3.6	系统演示	(92)
3.3.7	小结	(93)

第4章 网络数据通信案例	(94)
4.1 网络信息系统基础	(94)
4.1.1 网络信息系统的应用	(94)
4.1.2 网络信息系统的特点	(94)
4.1.3 网络信息系统的开发	(95)
4.2 网络数据通信编程	(98)
4.2.1 客户机/服务器模型	(98)
4.2.2 Windows Sockets 编程原理	(101)
4.3 企业网上寻呼系统	(104)
4.3.1 系统概述	(104)
4.3.2 需求分析	(105)
4.3.3 系统设计	(106)
4.3.4 数据库设计	(107)
4.3.5 通信模块设计	(109)
4.3.6 服务器端设计	(110)
4.3.7 客户端设计	(118)
4.3.8 系统运行	(149)
4.3.9 小结	(151)
第5章 图形图像处理案例	(152)
5.1 图形图像系统基础	(152)
5.1.1 图形图像系统的应用	(152)
5.1.2 图形图像系统的特点	(153)
5.1.3 图形图像系统的开发	(153)
5.2 Visual Basic 图形图像编程	(154)
5.2.1 坐标系统	(154)
5.2.2 色彩设置	(155)
5.2.3 绘图语句	(155)
5.2.4 绘图控件	(156)
5.3 图像绘制处理系统	(160)
5.3.1 系统概述	(160)
5.3.2 需求分析	(160)
5.3.3 系统设计	(160)
5.3.4 界面设计	(161)
5.3.5 文件相关模块	(166)
5.3.6 图像浏览模块	(170)
5.3.7 图像编辑模块	(173)
5.3.8 图像绘图模块	(177)

5.3.9 图像处理模块	(186)
5.3.10 系统运行	(194)
5.3.11 小结	(196)
5.4 图形绘制程序	(197)
5.4.1 程序简介	(197)
5.4.2 Visual Basic 类开发	(197)
5.4.3 DrawObject 类设计	(201)
5.4.4 DrawLine 类设计	(205)
5.4.5 DrawRec 类设计	(209)
5.4.6 DrawCircle 类设计	(213)
5.4.7 主窗体设计	(216)
5.4.8 系统运行	(228)
5.4.9 小结	(228)
第6章 多媒体应用案例	(229)
6.1 多媒体系统简介	(229)
6.1.1 多媒体技术的应用	(229)
6.1.2 多媒体系统的组成	(229)
6.1.3 多媒体系统的开发	(230)
6.2 多媒体程序设计基础	(231)
6.2.1 Windows 多媒体体系结构	(231)
6.2.2 Visual Basic 多媒体程序设计	(234)
6.3 MP3Player 播放器	(239)
6.3.1 系统简介	(239)
6.3.2 需求分析	(239)
6.3.3 系统设计	(240)
6.3.4 主模块设计	(240)
6.3.5 列表窗体设计	(255)
6.3.6 平衡器设计	(275)
6.3.7 系统运行	(277)
6.3.8 小结	(278)
第7章 游戏编程案例	(279)
7.1 游戏开发历史	(279)
7.1.1 早期历史	(279)
7.1.2 DOS 时代	(279)
7.1.3 Windows 的出现	(280)
7.1.4 DirectX 的出现	(280)
7.2 DirectX 简介	(281)

7.2.1	DirectX 发展历史	(281)
7.2.2	DirectX 组件	(283)
7.2.3	DirectX 游戏编程	(284)
7.3	TankWar 游戏	(288)
7.3.1	系统简介	(288)
7.3.2	界面设计	(289)
7.3.3	CSprite 精灵类设计	(289)
7.3.4	CTank 坦克类设计	(297)
7.3.5	通用模块设计	(302)
7.3.6	主模块设计	(304)
7.3.7	系统运行	(314)
7.3.8	小结	(315)
7.4	俄罗斯方块	(315)
7.4.1	系统简介	(315)
7.4.2	资源文件	(316)
7.4.3	通用模块设计	(317)
7.4.4	界面设计	(318)
7.4.5	系统设计	(319)
7.4.6	代码实现	(321)
7.4.7	系统运行	(339)
7.4.8	小结	(340)
参考文献		(341)

第 1 章

Visual Basic 编程基础

1.1 Visual Basic 编程特点

1.1.1 Windows 工作方式

Windows 系统是一个消息驱动操作系统，消息的发送、接收、处理过程贯穿于 Windows 系统及其应用程序生存的整个过程，消息驱动方式是 Windows 系统的工作方式（如图 1-1 所示）。当用户进行输入或是窗口的状态发生改变时系统都会发送消息到某一个窗口。例如当菜单被选择之后，消息就会发送到操作系统或应用程序，操作系统或应用程序对这个消息进行处理，完成对用户选择菜单这一动作的响应。大多数消息是系统定义的，用户也可以定义自己的消息，可以利用自定义消息来发送、通知和传送数据。当然一个消息必须被窗口接收、处理后才能作出响应。窗口定义的过程可以对消息进行分析，对自己感兴趣

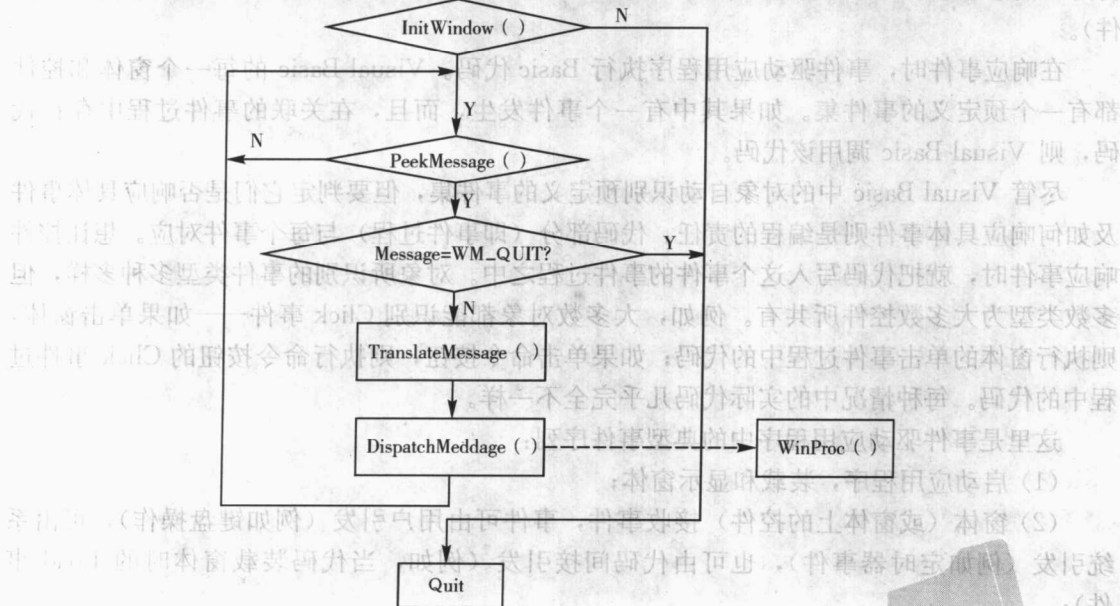


图 1-1 Windows 消息处理机制



的消息进行处理。例如希望对菜单选择进行处理,就可以在菜单选择事件的过程中加入处理代码。Windows 为窗口编写了一些默认的窗口过程,这个窗口过程负责处理那些用户不感兴趣的消息,这些消息直接由 Windows 系统自己进行处理。有了这个默认窗口过程,用户就可以把精力放在感兴趣的应用上,而不必过多关注窗口各种消息的处理。例如窗口在被拖动时会有很多消息发送,用户可以编写代码进行处理,也可以什么都不做,由系统自己处理这些消息。

由于 Visual Basic 应用程序是基于对象的,所以应用程序的代码结构就是该程序在屏幕上物理表示的模型。根据定义,对象包含数据和代码。在屏幕上看到的窗体代表属性,这些属性定义了窗体的外观和内在特性。应用程序中的每个窗体都有一个包含其代码的相关的窗体模块(文件扩展名为 .frm)。每个窗体模块都包含事件过程,即代码部分,其中有为响应特定事件而执行的指令。窗体可包含控件。在窗体模块中,对窗体上的每个控件都有一个对应的事件过程集。除了事件过程,窗体模块还可包含通用过程,它对来自任何事件过程的调用都作出响应。可将那些与特定窗体或控件无关的代码放入另一类型的模块——标准模块(文件扩展名为 .BAS)中。如果一个过程可能用来响应几个不同对象中的事件,应该将这个过程放在标准模块中,而不应在每一个对象的事件过程中重复相同的代码。用类模块(文件扩展名为 .CLS)创建对象,这些对象可被应用程序内的过程调用。标准模块只包含代码,而类模块既包含代码又包含数据,可视为没有物理表示的控件。

1.1.2 事件驱动程序

事件是窗体或控件识别的动作。每发生一次事件,就会引发一条消息发送至操作系统。操作系统处理该消息并将该消息发送到其他窗口的消息队列中。然后,每一个窗口才能根据自身的消息处理程序采取适当的操作(例如,当窗口被缩小或放大时,重画窗口中的各种控件)。

在响应事件时,事件驱动应用程序执行 Basic 代码。Visual Basic 的每一个窗体和控件都有一个预定义的事件集。如果其中有一个事件发生,而且,在关联的事件过程中存在代码,则 Visual Basic 调用该代码。

尽管 Visual Basic 中的对象自动识别预定义的事件集,但要判定它们是否响应具体事件及如何响应具体事件则是编程的责任。代码部分(即事件过程)与每个事件对应。想让控件响应事件时,就把代码写入这个事件的事件过程之中。对象所识别的事件类型多种多样,但多数类型为大多数控件所共有。例如,大多数对象都能识别 Click 事件——如果单击窗体,则执行窗体的单击事件过程中的代码;如果单击命令按钮,则执行命令按钮的 Click 事件过程中的代码。每种情况中的实际代码几乎完全不一样。

这里是事件驱动应用程序中的典型事件序列:

- (1) 启动应用程序,装载和显示窗体;
- (2) 窗体(或窗体上的控件)接收事件,事件可由用户引发(例如键盘操作),可由系统引发(例如定时器事件),也可由代码间接引发(例如,当代码装载窗体时的 Load 事件);
- (3) 如果在相应的事件过程中存在代码,就执行代码;

(4) 应用程序等待下一次事件。

从事件驱动应用程序这个典型事件序列中, 可以看到事件驱动模型与传统的“过程化”应用程序有明显区别, 传统应用程序自身控制执行哪部分代码和按何种顺序执行代码, 并且在执行程序过程中, 一般从第一行执行起, 按照预定的路径执行下去, 直到程序结束。而在事件驱动的应用程序中, 程序无法给出一个预定的执行序列, 程序代码也不会按照预定的路径执行, 因为程序在响应不同的事件时会执行不同的代码片段。事件可以用操作触发, 也可以由来自操作系统或其他应用程序的消息触发, 甚至由应用程序本身的消息触发。如单击鼠标左键会产生一个“鼠标左键被单击”事件, 时间控件 (Timer) 设置的时间间隔到达后, 会产生一个 Timer 事件。这些事件发生的顺序决定了代码执行的顺序, 因此应用程序每次运行时所执行的代码的路径都是不同的。所以, 在用 Visual Basic 设计应用程序时, 理解并灵活运用事件驱动模型是非常重要的。

1.1.3 面向应用的开发

Visual Basic 是一个强大的 Windows 平台上的开发工具, 从开发个人或小组使用的小工具, 到大型企业应用系统, 甚至通过 Internet 的遍及全球分布式应用程序, 都可在 Visual Basic 提供的工具中各取所需。Visual Basic 是一种面向对象的程序设计语言 (OOP), 采用控件为主体的事件驱动模型程序设计方法, 大大减轻了程序员开发 Windows 应用程序的效率, 使程序员把精力投入到系统应用中, 而不是陷入 Windows 烦琐的处理机制和函数名称中。

Visual Basic 的主要特点可以归结如下。

① 真正的面向对象编程, 使开发人员在维护系统运行时只需修改很小的代码, 同时也加快了系统开发的速度。

② 可视化的编程方法, 以及向导的功能, 使开发人员几乎不用加入太多代码就可以开发出标准的 Windows 程序。

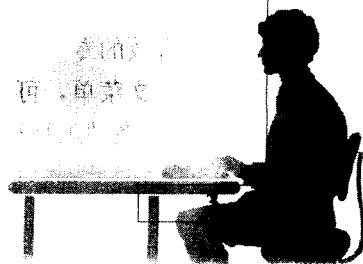
③ 数据访问特性允许对包括 Microsoft SQL Server 和其他企业数据库在内的大部分数据库格式建构数据库和前端应用程序。

④ 通过 ActiveX 技术可使用其他应用程序提供的功能器, Microsoft Excel 电子数据表及其他 Windows 应用程序, 创建的应用程序和对象。例如 Microsoft Word 文字处理, 甚至可直接使用 Visual Basic。

⑤ Visual Basic 能力强大, 使得在应用程序内很容易通过 Internet 访问文档和应用程序。

⑥ 已完成的应用程序是真正的 .exe 文件, 或者是可自由发布的动态链接库 (DLL)。

传统的应用程序开发过程可以分为三个明显的步骤: 编码、编译和测试代码。但是 Visual Basic 与传统的程序开发语言不同, 它直接面向应用, 采用交互式方法开发, 使传统应用程序开发的三个步骤不再有明显界限。



1.2 Visual Basic 开发环境

1.2.1 Visual Basic 开发环境组成

Visual Basic 集成开发环境在一个公共环境里集成了许多不同的功能，包括设计、编辑、编译和调试。而在大多数传统开发工具中，每个功能都是一个独立的程序运行，拥有自己的界面。如图 1-2 所示集成开发环境。

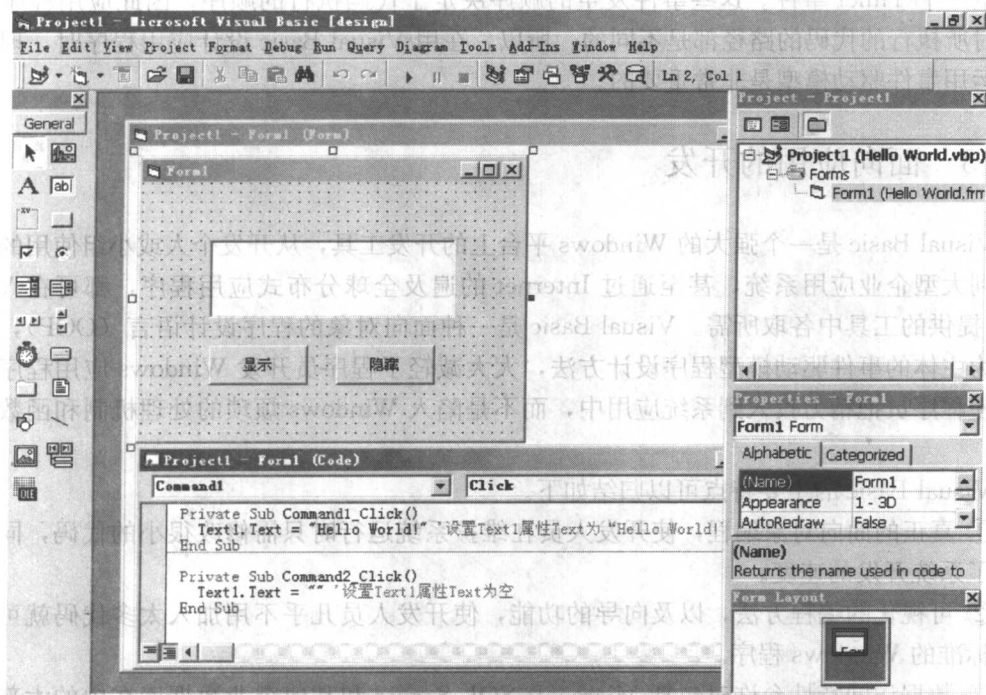


图 1-2 集成开发环境

(1) 菜单条

显示所使用的 Visual Basic 命令。除了提供标准“文件”、“编辑”、“视图”、“窗口”和“帮助”菜单之外，还提供编程专用的功能菜单，例如“工程”、“格式”或“调试”。如图 1-3 所示。

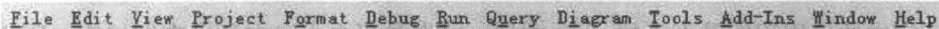


图 1-3 菜单条

(2) 上下文菜单

包括经常使用操作的快捷键。在要使用的对象上单击即可打开上下文菜单。在上下文菜单中有效的专用快捷键清单取决于单击鼠标键所在环境。例如，在“工具箱”上右击时显示的上下文菜单，可以在上面选择显示“部件”对话框，隐含“工具箱”，停放或挂断“工具箱”，或在“工具箱”中添加自定义选项卡。如图 1-4 所示。

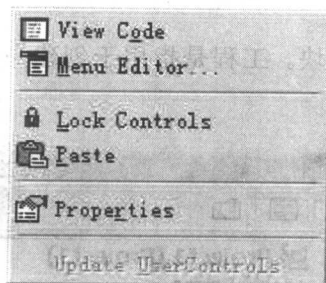


图 1-4 上下文菜单

(3) 工具栏

在编程环境下提供对于常用命令的快速访问。单击工具栏上的按钮，则执行该按钮所代表的操作。按照默认规定，启动 Visual Basic 之后显示“标准”工具栏。附加的编辑、窗体设计和调试的工具栏可以从“视图”菜单上的“工具栏”命令中移进或移出。工具栏能紧贴在菜单条之下，或以垂直条状紧贴在左边框上。如果将它从菜单下面拖开，则它能“悬”在窗口中。如图 1-5 所示。



图 1-5 工具栏

(4) 工具箱

提供一组工具，用于设计时在窗体中放置控件。除了默认的工具箱布局之外，还可以通过从上下文菜单中选定“添加选项卡”并在结果选项卡中添加控件来创建自定义布局。如图 1-6 所示。

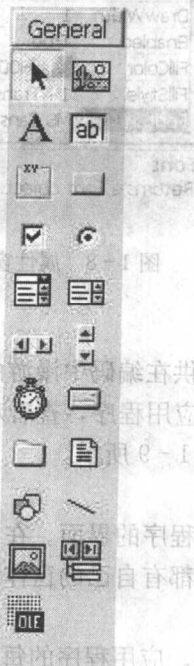


图 1-6 工具箱

(5) 工程管理窗口

列出当前工程中的窗体和模块。工程是指用于创建一个应用程序的文件的集合。如图 1-7 所示。

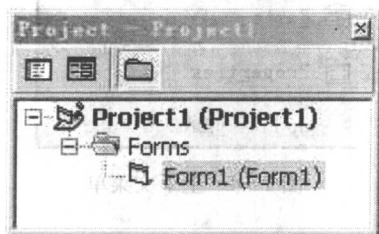


图 1-7 工程管理窗口

(6) 属性窗口

列出对选定窗体和控件的属性设置值。属性是指对象的特征，如大小、标题或颜色。如图 1-8 所示。

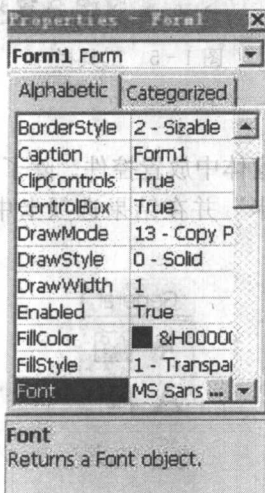


图 1-8 属性窗口

(7) 对象浏览器

列出工程中有效的对象，并提供在编码中漫游的快速方法。可以使用“对象浏览器”浏览在 Visual Basic 中的对象和其他应用程序，查看对那些对象有效的方法和属性，并将代码过程粘贴进自己的应用程序。如图 1-9 所示。

(8) 窗体设计器

作为自定义窗口用来设计应用程序的界面。在窗体中添加控件、图形和图片来创建所希望的外观。应用程序中每一个窗体都有自己的窗体设计器窗口。如图 1-10 所示。

(9) 代码编辑窗口

是输入应用程序代码的编辑器。应用程序的每个窗体或代码模块都有一个单独的代码编辑器窗口。如图 1-11 所示。