

第一章 Visual Basic 6.0 概述

很长一段时间以来，Basic 语言一直是大多数计算机初学者的入门语言，它以其简洁的语法、易用的编程方式赢得广泛的欢迎。但是，随着各种编程语言的日益丰富，Basic 语言由于其功能过于简单，已越来越不适应用户的要求。因此，人们对 Basic 语言进行了多种改进，从而产生了多种版本，如 Quick Basic、True Basic 等。但是，随着可视化编程技术的日益应用，广受欢迎的 Visual Basic 语言问世了。它提供了开发 Microsoft Windows 应用程序的最迅速、最简捷的方法，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 都为他们提供了整套工具，以方便开发应用程序。

1.1 Visual Basic 简介

何谓 Visual Basic？“Visual”指的是开发图形用户界面 (GUI) 的方法。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对象拖放到屏幕上的一点即可。如果已使用过诸如 Paint 之类的绘图程序，则实际上已掌握了创建用户界面的必要技巧。

“Basic”指的是 BASIC (Beginners All-Purpose Symbolic Instruction Code) 语言，一种在计算技术发展历史上应用得最为广泛的语言。Visual Basic 在原有 BASIC 语言的基础上进一步发展，至今包含了数百条语句、函数及关键词，其中很多和 Windows GUI 有直接关系。专业人员可以用 Visual Basic 实现其他任何 Windows 编程语言的功能，而初学者只要掌握几个关键词就可以建立实用的应用程序。

1.1.1 Visual Basic 的特点

Visual Basic 不仅是 Visual Basic 编程语言。Visual Basic 编程系统和 VBA 都使用这一语言。Visual Basic 编程系统和 VBS (VBScript) 是 Visual Basic 语言的子集。这样，在学习 Visual Basic 中得到的经验可应用到所有这些领域中。

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。

- 数据访问特性允许 Visual Basic 访问由 Microsoft SQL Server 和其他企业数据库在内的大部分数据库格式建立的数据库。
- 利用 ActiveX 技术就可在 Visual Basic 程序中使用其他应用程序提供的功能，例如 Microsoft Word 字处理器，Microsoft Excel 电子数据表及其他 Windows 应用程序。甚至可直接使用 VBP 或 VBE 创建的应用程序和对象。

- Internet 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。
- 已完成的应用程序是真正的 .exe 文件，或可供用户自由发布的动态链接库 (DLL)。

Visual Basic 有三种版本，各自满足不同的开发需要。

- Visual Basic 学习版使编程人员轻松开发 Windows 95 和 Windows NT(R) 的应用程序。该版本包括所有的内部控件连同 Grid、Tab 和 Data_Bound 控件。
- 专业版为专业编程人员提供了一整套进行开发的功能完备的工具。该版本包括学习版的全部功能连同 ActiveX 控件，还包括 Internet 控件和 Crystal Report Writer。
- 企业版使得专业编程人员能够开发功能强大的组内分布式应用程序。该版本包括专业版的全部功能连同自动化管理器、部件管理器、数据库管理工具、Microsoft Visual SourceSafe 面向工程版的控制系统等等。

1.1.2 Visual Basic 6.0 的新特点

和以前 VB 版本不同的是，Visual Basic 6.0、Visual FoxPro 6.0、Visual C++ 6.0 等各种编程语言的帮助均被集成到了 MSDN 库中。用户在使用时可单独安装该帮助库，并可定制所要安装的组件。

1. 数据访问的改进

- 增加了 ADO (ActiveX Data Objects) 数据对象 (在所有版本中)。该数据访问技术是一个简单的数据模型，可更好地与其他 Microsoft 和非 Microsoft 技术集成。它还是一个通用接口，可用于本地和远程数据、远程和离散记录集、用户可访问的数据绑定接口和分层记录集的访问。
- 提供了数据环境设计器 (在专业版和企业版中)。数据环境设计器提供了一种创建 ADO 对象的交互设计环境。数据环境可作为窗体或报表中所有要求数据对象的数据源，或者通过数据环境的方法和属性由程序访问。数据环境设计器具有用户连接设计器的全部功能和附加特性，如拖放、分层、分组和合计等。
- ADO 数据控件 (所有版本)。这是一个新的 OLEDB 数据源控件，其功能类似内部和远程数据控件。用户可以利用该控件，以最少的代码创建数据库应用程序。
- 增强的数据绑定 (专业版和企业版)。在 Visual Basic 的以前版本中，只能将控件绑定到窗体。在 VB 6.0 中，用户可将任何 ADO/OLE DB 数据源绑定到任何 ADO/OLE DB 数据桶 (Consumer) 上。用户可通过在运行时设置控件的 DataSource 属性动态绑定到数据源。可以创建数据源和数据桶类，并通过新的绑定集合对象绑定它们。用户还可创建类似 ADO 数据控件的数据源用户控件，类似 DataGrid 控件的组合绑定控件。
- OLE DB 支持 (所有版本)。OLE DB 是一个 COM 接口集合，它提供了访问存储在不同信息源 (无论是关联还是非关联) 中数据的统一方法。这些接口支持多个 DBMS 利用数据源，以便共享数据。ADO 还是编程访问 OLE DB 的一种方式。所

有新的数据绑定控件、数据环境和数据报表设计器都属于 OLE DB 类型。

- **Visual 数据库工具集成（企业版）。**在 Visual Basic 6.0 中，用户可以可视地创建数据库和查询、SQL 服务程序、Oracle 数据库表、使用拖放方法创建视图，以及自动改变列数据类型。
- **设置向导已被增强（所有版本）。**新的 **Package and Deplyment** 向导（以前的设置向导）现在支持 ADO、OLE DB、RDOODBC 和 DAO。
- **数据报表（专业版和企业版）。**允许用户使用拖放方法利用任何记录集合（包括分层记录）快速创建报表。
- **数据源（专业版和企业版）。**创建可绑定至任何控件的用户定义控件和类。
- **数据视图窗口（专业版和企业版）。**用户可利用数据视图窗口浏览所有连接的数据库、表、视图和存储过程等。
- **SQL 编辑器（企业版）。**允许用户增加新的存储过程至 SQL 服务程序和 Oracle 数据库中。用户还可使用编辑器编写触发器。
- **Hierachical FlexGrid Control（所有版本）。**该控件为 FlexGrid 控件的升级版本。它除了支持 FlexGrid 的所有功能外，还可显示 ADO 记录集合的层次。每一个被返回的记录集都被显示在表格中的一个独立带区，并可单独设置格式。
- **数据复制器控件 DataRepeater（专业版和企业版）。**插入一个用户定义控件至 DataRepeater 可创建一个定制的数据库视图。用户定义控件可以包括文本框、检查框、数据表格等绑定至数据字段的其他控件。
- **传递 ADO 记录集（所有版本）。**用户现在可以通过进程，甚至使用 HTTP 或 DCOM 通过计算机传递 ADO 记录集。这提供了在多个应用程序之间移动数据的有效方式。
- **增强的数据窗体向导（专业版和企业版）。**现在用户不仅可以构造不绑定至数据控件的代码窗体，而且可以使用 ADO 代码。该向导已经集成到应用程序向导、Chart 向导和 FlexGrid 向导中。
- **数据对象向导（专业版和企业版）。**可以自动创建绑定至数据环境或用户控件的 Middle-tier 对象。
- **文件系统对象（所有版本）。**Visual Basic 6.0 提供了改进的在文件系统中移动的程序集，并可创建文本文件和目录。
- **格式对象（所有版本）。**提供了数据库和绑定控件之间数据的双向转换。当从数据库中读出数据时，格式对象将会为数据增加格式信息以适应绑定控件。
- **DataGrid 控件（所有版本）。**DBGrid 的 OLE DB 版本。该控件允许快速构造观察和编辑记录的程序，它也支持新的 ADO 数据控件。
- **DataList 和 DataCombo 控件（所有版本）。**分别为 DBList 和 DBCombo 控件的 OLE DB 版本，并支持新的 ADO 数据控件。

2. 新的 Internet 特性

- **IIS 程序（专业版和企业版）。**用户现在可以编写 Internet 服务器程序，以使 Visual Basic 代码能够响应来自浏览器的用户请求。

- DHTML 程序（专业版和企业版）。利用动态 HTML（DHTML）技术，Web 页面上的任何元素都可具有属性、方法和事件。在 Visual Basic 6.0 中，允许用户编写代码来响应 HTML 页上的动作，而不用去请求服务器。
- Web 出版向导（所有版本）。利用 Package and Deployment 向导（以前的 Setup 向导），用户可以轻松地传送已打包的应用程序包至 Web 站点。
- 增强的异步阅读（所有版本）。异步操作已被增强，以给出更多的有关操作进程和状态的信息，如已经阅读的字节数、有待阅读的字节数等。Visual Basic 6.0 新增了 AsyncReadProgress 事件、BytesMax 属性、BytesRead 属性、Status 属性（AsyncProperty 对象）、StatusCode 属性和 Target 属性，增强了 AsyncRead 方法和 AsyncReadCVomplete 事件。
- IE 4.x 下载支持 ActiveX 文档。现在，在 IE 4.x 中下载 ActiveX 文档和下载 ActiveX 控件完全相同，例如，支持相对 URLs。

3. 新的和改进的控件

在 Visual Basic 6.0 中，新增了如下一些控件：

- ADO Data 控件（所有版本）。该控件为新的 OLE DB 类型数据源控件，其功能类似 Intrinsic Data 和 Remote Data 控件。
- CoolBar 控件（专业版和企业版）。用于创建用户配置工具栏的容器控件。
- DataGrid 控件（所有版本）。DBGrid 的 OLE DB 版本，允许快速构造浏览和编辑记录集的应用程序。
- DataList 控件和 DataCombo 控件（所有版本）。为 DataList 和 DBComobo 控件的 OLE DB 版本。
- DataRepeater 控件（专业版和企业版）。可向该控件中插入 UserControl，以定制数据库视图。
- DateTimePicker 控件（专业版和企业版）。当需要提供下拉式日历时，可在窗体中放置该控件。
- Flat Scrollbar 控件（专业版和企业版）。可创建更为精致的滚动条。
- Hierarchical FlexGrid 控件（所有版本）。FlexGrid 控件的升级版本。
- ImageCombo 控件（所有版本）。类似标准组合框，但可将图像作为列表项目。
- MonthView 控件（专业版和企业版）。允许用户从图形日历上输入日期和指定日期范围。

在 Visual Basic 6.0 中，Intrinsic、ImageList、ListView、MSChart、ProgressBar、Toolbar 等控件被进行了改进或增强。

4. 组件创建增强

在 Visual Basic 6.0 中，用户可创建自己的 Data Consumer、数据源、项目开始选择、简化的 UserControl 等。

现在用户可以基于用户控件或类创建 Data Consumer，以支持复杂的绑定能力；基于用户控件或类创建数据源；按 F5 键允许用户决定从何处运行控件（用于程序调试），例

如，用户可以在浏览器中运行控件而不是在 Visual Basic 窗体中。用户现在还可以创建要求较少资源的无创刊和透明的用户控件。

由于现在过程可返回数组，这意味着在类中可将数组作为属性。如果被赋值的数组为动态数组，用户还可进行数组赋值，即将一个数组的值赋给另一个动态数组。

5. 语言增强

Visual Basic 6.0 还对某些语言特性进行了改进，例如，用户定义类型的参数化、返回公共属性和方法的类型、函数可返回数组、可变数组可出现在等式的左侧、通过指定包含属性或方法名的字符串访问属性和方法、CreateObject 函数现在允许用户指定机器名作为可选参数以便访问远程机器上的对象等。

此外，Visual Basic 6.0 还新增了一些字符串函数，如 Filter、FormatCurrency、FormatDateTime、FormatNumber、FormatPercent、InstrRev、Join、MonthName、Replace、Round、Split、StrReverse、WeekdayName 等。

1.2 安装 Visual Basic

使用安装程序安装 Visual Basic。安装程序将 Visual Basic 本身、帮助系统、应用程序示例，以及其他产品部件从 CD-ROM 安装到硬盘上。

注意：不能直接将 CD-ROM 上的文件复制到硬盘，然后从硬盘运行 Visual Basic。必须使用安装程序将文件解压缩并安装到合适的目录中。

1.1.1 运行安装之前

在安装 Visual Basic 之前，必须确认计算机满足最低安装要求，并阅读安装盘根目录下的文件 Readme。

为运行 Visual Basic，必须在计算机上安装相应的硬件和软件系统。这些系统要求包括：

- Microsoft Windows NT 3.51 或更新版本，或 Microsoft Windows 95。
- 80486DX/66 或更高微处理器（推荐使用 Pentium 或更高的处理器）。
- 一个 CD-ROM 驱动器。
- Microsoft Windows 支持的 VGA 或分辨率更高的监视器。
- 在 Windows 95 下 16 MB RAM，在 Windows NT 下 32MB RAM。
- 鼠标或其他定点设备。

1.1.2 安装 Visual Basic

运行安装程序时将会为 Visual Basic 创建目录，从而可选择要安装的 Visual Basic 部件。除了 \OS 目录下的操作系统文件，CD 上的其他文件是不压缩的，所以可从盘上直

接使用。例如，即使不在硬盘上安装“帮助”，仍可从 Visual Basic 中运行它们。按 F1 键就能得到需要的帮助信息，也就是启用上下文有关的帮助。

1. 从 CD 盘上安装

(1) 在 CD-ROM 驱动器中插入 CD 盘。

(2) 安装程序在 CD 盘的根目录下，可用操作系统中的适当命令来运行。如果你的计算机能够在系统中运行 AutoPlay，则在插入 CD 盘时，安装程序将被自动加载。

(3) 选取“Install Visual Basic 6.0”。

(4) 依照屏幕上的安装指令行事。

2. 启动 Visual Basic

一旦完成安装过程，就可单击 Windows 95 任务栏中的“开始”按钮启动 Visual Basic。

思考题

1. Visual Basic 有哪三种版本？其区别何在？
2. 和 Visual Basic 以前版本相比，Visual Basic 6.0 有哪些新特点？

第二章 使用 Visual Basic 开发应用程序

创建第一个 Visual Basic 应用程序只需要几分钟的时间。用户可以首先通过在窗体上“绘制”诸如文本框和命令按钮等控件来创建用户界面，然后为窗体和控件设置属性以规定诸如标题、颜色和大小等的值，最后编写代码将操作赋予应用程序。创建第一个应用程序所采取的基本步骤将是开发其他的应用程序所遵循的原则。

本章描述了使用 Visual Basic 所需要的术语和技巧，并且一步一步地指导创建几个简单的应用程序。

2.1 Visual Basic 程序设计基本概念

为了理解应用程序开发过程，先要理解 Visual Basic 赖以创建程序的一些关键概念。因为 Visual Basic 是 Windows 开发语言，所以有必要与 Windows 环境保持一定的相似性。如果不熟悉 Windows 编程，就需要明白在 Windows 环境下编程和在其他环境下编程的一些根本性的差别。

2.1.1 Windows 的工作方式：窗口、事件和消息

简单地说，Windows 的工作机制就是三个关键的概念：窗口、事件和消息。

不妨简单地将窗口看做带有边界的矩形区域。也许用户已经了解几种不同类型的窗口：如，Windows 95 的“资源管理器”窗口、文字处理程序中的文档窗口或者弹出提示有约会信息的对话框。除了这些最普通的窗口外，实际上还有许多其他类型的窗口。命令按钮是一个窗口，图标、文本框、选项按钮和菜单条也都是窗口。

Microsoft Windows 操作系统通过给每一个窗口指定一个唯一的标识号（称为窗口句柄或 `hWnd`）来管理所有的窗口。操作系统连续地监视每一个窗口的活动或事件的信号。事件可以通过诸如单击鼠标或按下按键的操作而产生，也可以通过程序的控制而产生，甚至可以由另一个窗口的操作而产生。

每发生一次事件，将引发一条消息发送至操作系统。操作系统处理该消息并广播给其他窗口。然后，每一个窗口才能根据自身处理该条消息的指令而采取适当的操作（例如，当窗口解除了其他窗口的覆盖时，重显自身窗口）。

可以想象，处理各种窗口、事件和消息的所有可能的组合将有惊人的工作量。幸运的是，Visual Basic 使你摆脱了所有的低层消息处理。许多消息由 Visual Basic 自动处理了，其他的作为事件过程由编程者自行处理。这样可以快速创建强大的应用程序而不必涉及不必要的细节。

2.1.2 事件驱动模型

在传统的或“过程化”的应用程序中，应用程序自身控制了执行哪一部分代码和按何种顺序执行代码。从第一行代码执行程序并按应用程序中预定的路径执行，必要时调用过程。

在事件驱动的应用程序中，代码不是按照预定的路径执行，而是在响应不同的事件时执行不同的代码片段。事件可以由用户操作触发，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。这些事件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经过的代码的路径都是不同的。

因为事件的顺序是无法预测的，所以在代码中必须对执行时的“各种状态”作一定的假设。当作出某些假设时（例如，假设在运行处理某一输入字段的过程之前，该输入字段必须包含确定的值），应该组织好应用程序的结构，以确保该假设始终有效（例如，在输入字段之前禁止使用启动该处理过程的命令按钮）。

在执行中代码也可以触发事件。例如，在程序中改变文本框中的文本将引发文本框的 **Change** 事件。如果 **Change** 事件中包含有代码，则将导致该代码的执行。如果原来假设该事件仅能由用户的交互操作所触发，则可能会产生意料之外的结果。正因为这一原因，所以在设计应用程序时理解事件驱动模型并牢记在心是非常重要的。

2.1.3 工程、窗体和控件

总体来讲，在 Visual Basic 中开发应用程序主要包括两部分工作：一是界面设计，二是编码设计。其中界面设计包括窗体布局和属性设置，此后用户还可根据需向窗体中放置控件。例如，要创建图 2-1 所示的记事本程序，可首先向程序中添加一个窗体（即程序运行的窗口），然后向窗体中放置一些控件，并设置该窗体和控件的尺寸、标题和图符等属性。最后，还要使用菜单编辑器为窗体加上一个菜单。图 2-2 显示了该程序的设计界面。

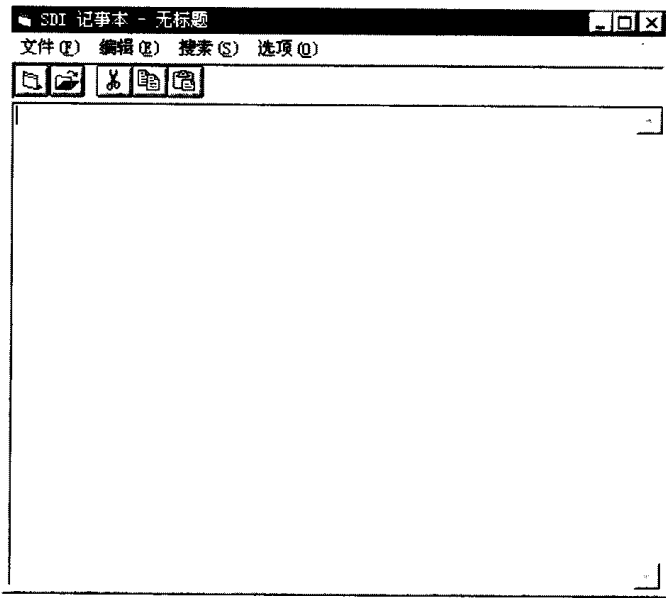


图 2-1 记事本程序运行画面

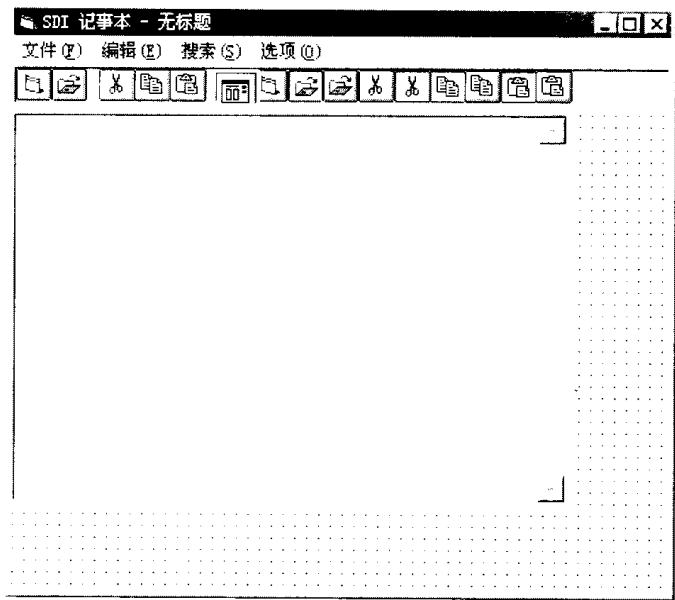


图 2-2 记事本程序设计界面

比较图 2-1 和图 2-2 不难发现，在 Visual Basic 中，程序的设计界面和运行画面基本相同，而这正是目前所谓可视化程序设计较以前的过程程序设计的优点所在。现在，对于 Visual Basic、Visual FoxPro、Visual C++ 等可视化程序设计语言，用户均可摆脱以往大量繁琐的界面设计任务。

注意：设计界面中的某些按钮只是为了程序设计方便，故未出现在程序运行画面中。

当然，完成程序的界面设计只是完成了程序的“脸面”部分，至于选择某项菜单或单击某个工具按钮干什么？还依赖于程序编码，即分别对窗体和控件的事件、方法等进行编程。

当然，一个程序除了可以包含窗体（可以不止一个）、控件之外，还可包含多种其他部件，如程序模块、类模块、用户控件等，以分别适应不同的用途。为此，系统提出了工程文件这一概念。利用工程文件，系统可帮助用户管理所有这些部件，其具体形式是通过工程资源管理器窗口来实现的。例如，本例的工程资源管理器窗口如图 2-3 所示。

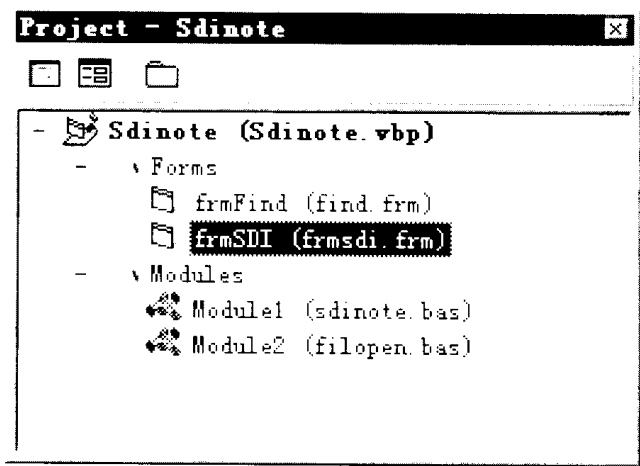


图 2-3 工程资源管理器窗口

由图 2-3 可以看出，本程序包含了两个窗体和两个模块。

2.1.4 属性、方法和事件

Visual Basic 的窗体和控件是具有自己的属性、方法和事件的对象。可以把属性看作一个对象的性质，把方法看作对象的动作，把事件看作对象的响应。

日常生活中的对象，如小孩玩的气球同样具有属性、方法和事件。气球的属性包括可以看到的一些性质，如它的直径和颜色。其他一些属性描述气球的状态（充气的或未充气的）或不可见的性质，如它的寿命。通过定义，所有气球都具有这些属性；这些属性也会因气球的不同而不同。

气球还具有本身所固有的方法和动作。如：充气方法（用氦气充满气球的动作），放气方法（排出气球中的气体）和上升方法（放手让气球飞走）。

气球还有预定义的对某些外部事件的响应。例如，气球对刺破它的事件响应是放气，对放手事件的响应是升空。

如果能对气球编程的话，Visual Basic 代码将如下所示。其作用是设置气球的属性：

```
Balloon.Color = Red  
Balloon.Diameter = 10  
Balloon.Inflated = True
```

请注意代码的语法，首先是对象 (Balloon)，接着是属性 (Color)，然后是赋值 (Red)。重复这条语句，但是，换一个不同的值，就可以改变气球的颜色。属性也可以在程序设计阶段在“属性”窗口中设置。

调用气球的方法是这样的：

```
Balloon.Inflate  
Balloon.Deflate  
Balloon.Rise 5
```

它的语法与属性的语法相似，对象（一个名词）后面紧跟着方法（一个动词）。在第三个例子中有一个附加项，称为一个参数，表示气球上升的高度。一些方法有一个或多个参数，它们对执行的动作做进一步的描述。

气球对一个事件的响应如下所示：

```
Sub Balloon_Puncture()  
    Balloon.Deflate  
    Balloon.MakeNoise "Bang"  
    Balloon.Inflated = False  
    Balloon.Diameter = 1  
End Sub
```

本例中代码描述了当刺穿事件发生时，气球的行为：调用 Deflate 方法，然后以 "Bang"（发出的声响）为参数调用 MakeNoise 方法。因为气球已不再是充气状态，所以把 Inflated 属性置为 False，并给 Diameter 属性设置一个新值。

尽管实际上对气球不能编程，但是，用户可以对 Visual Basic 的窗体或控件进行编程。作为程序员，用户决定应更改哪些属性、调用哪些方法、对哪些事件作出响应，从而得到希望的外观和行为。

2.1.5 交互式开发

图 书 馆

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 与传统的语言不同，它使用交互式方法开发应用程序，使三个步骤之间不再有明显的界限。

在大多数语言里，如果编写代码时发生了错误，则在开始编译应用程序时该错误就会被编译器捕获。此时必须查找并改正该错误，然后再次进行编译，对每一个发现的错误都要重复这样的过程。Visual Basic 在编程者输入代码时便进行解释，即时捕获并突出显示大多数语法或拼写错误。看起来就像一位专家在监视代码的输入。

除即时捕获错误以外，Visual Basic 也在输入代码时部分地编译该代码。当准备运行和测试应用程序时，只需极短时间即可完成编译。如果编译器发现了错误，则将错误突出显示于代码中。这时可以更正错误并继续编译，而不需从头开始。

由于 Visual Basic 的交互特性，因此可以在开发应用程序时运行应用程序。通过这种方式，代码运行的效果可以在开发时进行测试，而不必等到编译完成以后。

2.2 了解 Visual Basic 集成开发环境

Visual Basic 的工作环境常常是指集成开发环境或 IDE，这是因为它在一个公共环境里集成了许多不同的功能，例如，设计、编辑、编译和调试。

2.2.1 启动 Visual Basic 集成开发环境

第一次启动 Visual Basic 时，用户将首先看到图 2-4 所示画面。其中 New Project 窗口中列出了用户可以增加的各种 VB 工程类型。

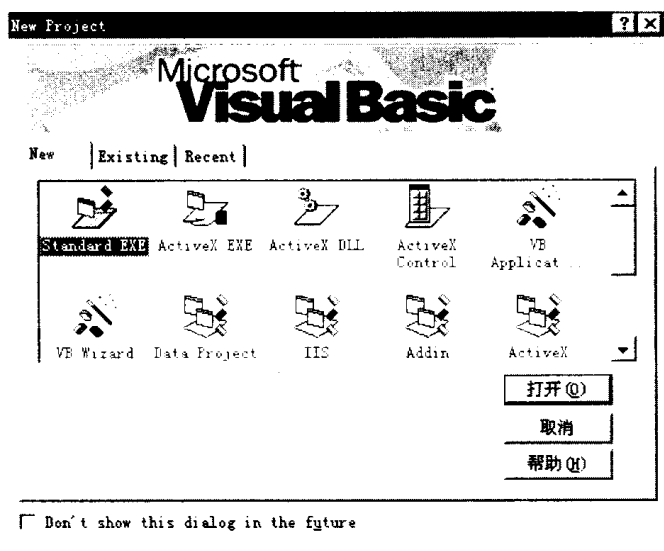


图 2-4 Visual Basic 集成开发环境

若直接单击“打开”按钮（创建标准 EXE 文件），则可见到如图 2-5 所示的集成开发环境。

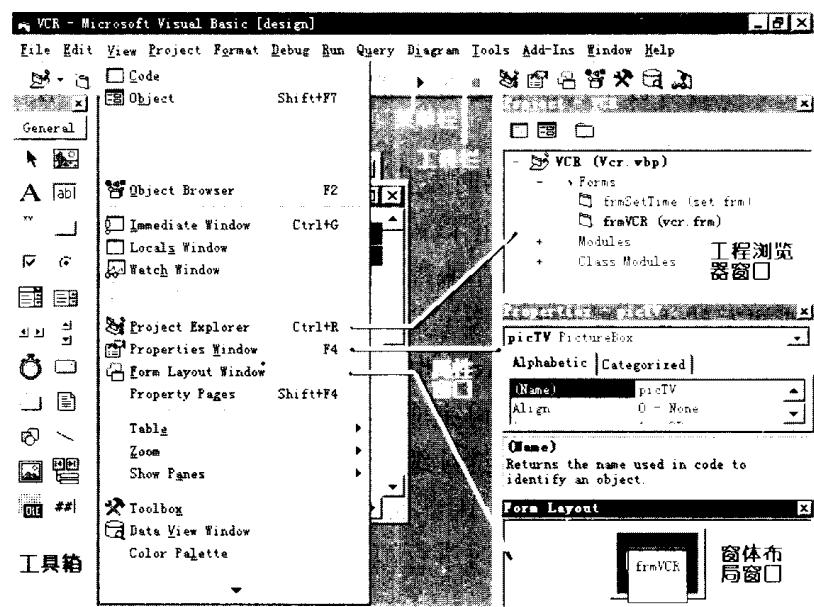


图 2-5 Visual Basic 集成开发环境

2.2.2 集成开发环境的元素

由图 2-5 不难看出，Visual Basic 集成开发环境 (IDE) 主要由以下元素组成：

1. 菜单条

显示所使用的 Visual Basic 命令，除了提供标准“File”、“Edit”、“View”、“Window”和“Help”菜单之外，还提供了编程专用的功能菜单，例如“Project”、“Format”或“Debug”。

2. 上下文菜单

上下文菜单又称快捷菜单或弹出菜单，它包括了用户经常要执行的操作的快捷键。在要使用的对象上单击鼠标键右键即可打开上下文菜单。

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。按照缺省规定，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通过选择“View”菜单上的“Toolbars”来打开或关闭。

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。

4. 工具箱

Visual Basic 提供了一组工具，用于设计时在窗体中放置控件。除了缺省的工具箱布局之外（称为 General），还可以通过在工具箱区右击打开一个快捷菜单，然后从该菜单中

选定“Add Tab”并在随后打开的对话框中输入工具箱名来创建自定义布局。

5. 工程资源管理器窗口

工程资源管理器窗口列出了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。

6. 属性窗口

属性窗口列出了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、标题或颜色等。

7. 对象浏览器

对象浏览器中列出了工程中有效的对象，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。

要打开对象浏览器窗口，可选择“View”菜单中的“Object Browser”。

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。

9. 代码编辑器窗口

代码编辑器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。要打开代码编辑器窗口，可在选定控件或窗体的快捷菜单中选择“View Code”(参见图 2-6)。

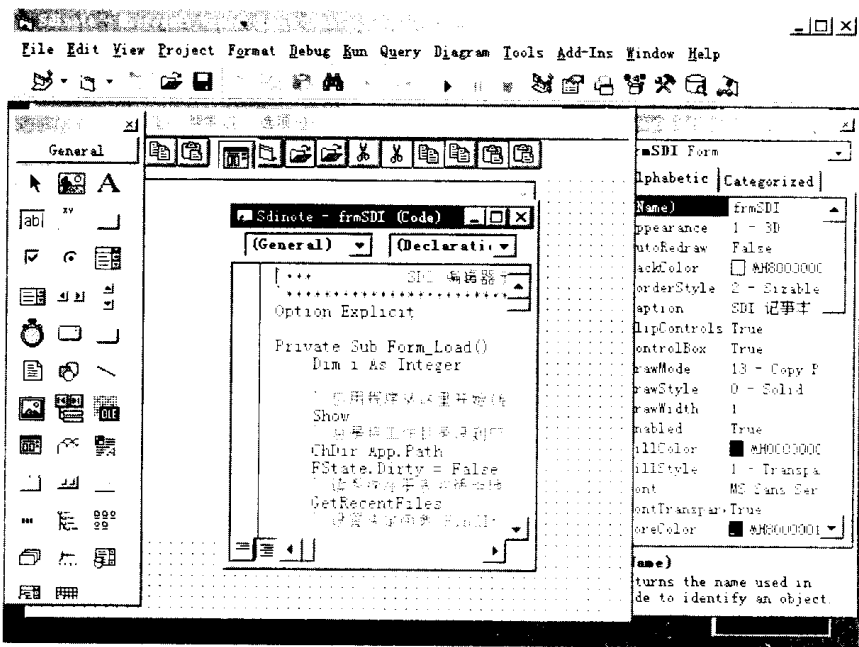


图 2-6 代码窗口

10. 窗体布局窗口

窗体布局窗口属性窗口的下方，允许使用表示屏幕的小图像来布置应用程序中各窗体的位置。

11. 立即、本地和监视窗口

这些附加窗口是为调试应用程序提供的，它们只在 IDE 之中运行应用程序时才有效。

2.2.3 集成开发环境设置

Visual Basic 具有很大的灵活性，可以通过配置工作环境满足个人风格的最佳需要。可以在单个或多文档界面中间进行选择，并能调节各种集成开发环境 (IDE) 元素的尺寸和位置。

1. SDI 或 MDI 界面

Visual Basic IDE 有两种不同的类型：多文档界面 (MDI) 和单文档界面 (SDI)。对于 MDI 界面而言，所有子窗口包含在一个大小可调的父窗口内（参见图 2-7）；对 SDI 界面，所有窗口可在屏幕上任何地方自由移动（参见图 2-8）；缺省情况下，用户所使用的界面为 MDI 界面。

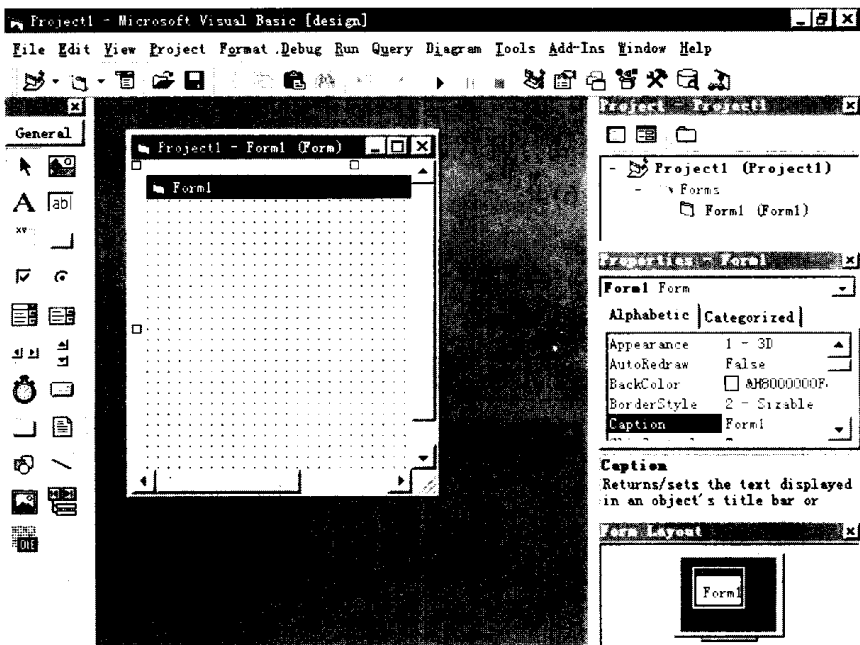


图 2-7 MDI 开发界面

要在 SDI 和 MDI 模式间进行切换，可以按照如下步骤进行：

- (1) 从“Tools”菜单中选定“Options”，打开显示“Options”对话框。
- (2) 选定“Advanced”选项卡。
- (3) 选择或不选择“SDI Development Environment”复选框，下次启动 Visual Basic 时，

IDE 将以选定模式的模式启动（参见图 2-9）。

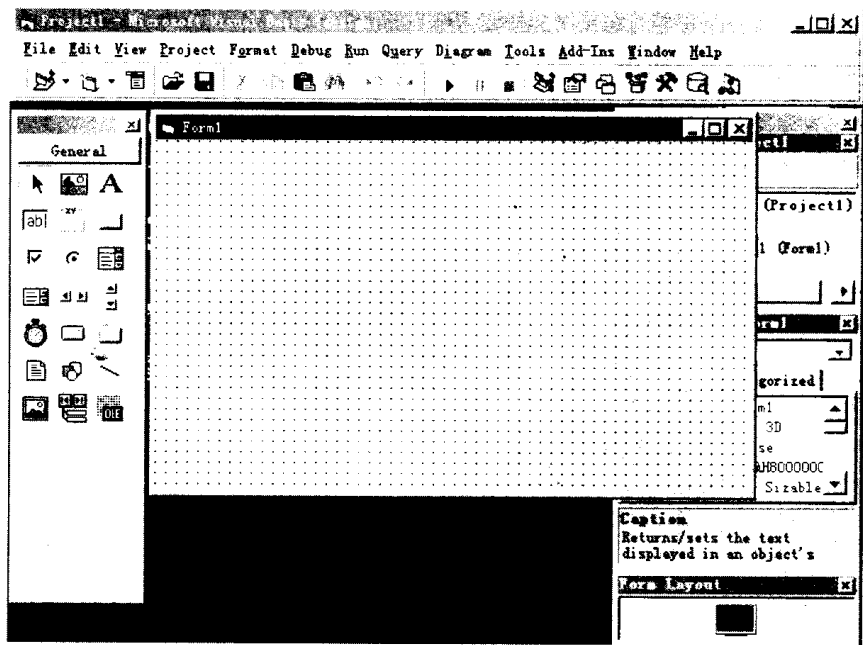


图 2-8 SDI 开发界面

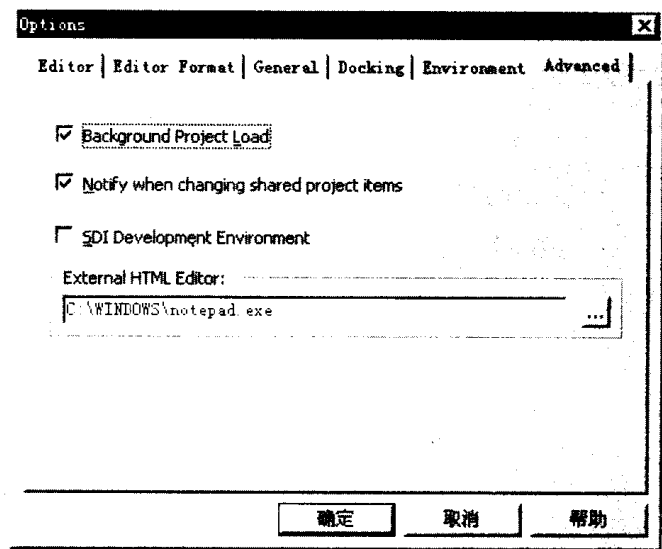


图 2-9 设置 SDI 或 MDI 开发环境

此外，用户也可用带 /sdi 或 /mdi 参数命令行运行 Visual Basic。

2. 可连接窗口

集成开发环境中的许多窗口能相互连接，或停放在屏幕边缘，正如用户在图 2-7 所看到的那样，工程资源管理器窗口、属性窗口、窗体布局窗口被连接放置在了屏幕的右侧。可连接放置的窗口包括：工具箱、窗体布局窗口、工程资源管理器、属性窗口、调色板、本地窗口和监视窗口等。

对 MDI 选项，窗口可停放在父窗口的任意侧，而对于 SDI，窗口只能停放在菜单条下面。一给定窗口的“可连接的”功能，可以通过在“Options”对话框的“Docking”

选项卡上选定合适的复选框来打开或关闭（参见图 2-10）。

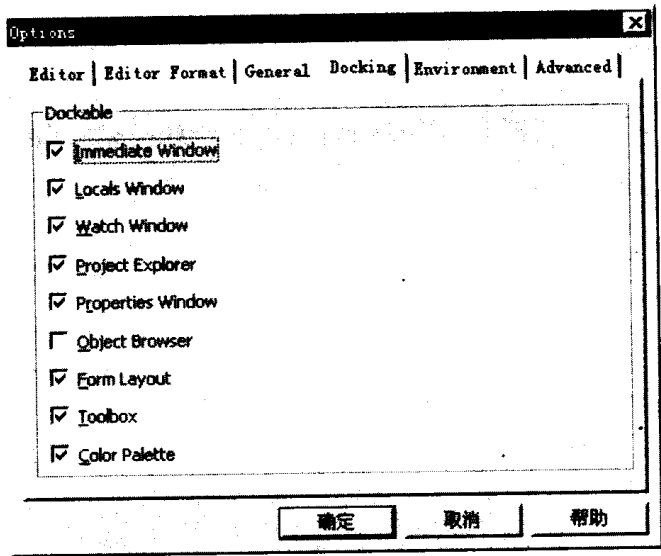


图 2-10 设置可连接窗口

要停放或移动窗口，可首先选定要停放或移动的窗口，然后按住鼠标左键拖动窗口到希望到达的位置（拖动时会显示窗口轮廓，参见图 2-11），最后释放鼠标按钮，结果将如图 2-12 所示。

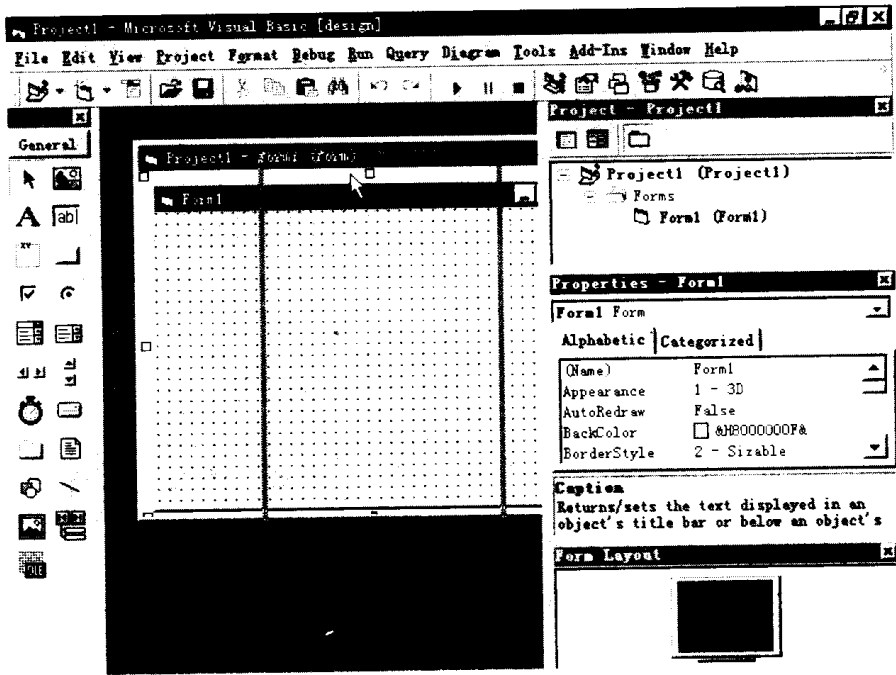


图 2-11 移动窗口

要恢复被移动的窗口与原窗口的连接，可双击标题条；要关闭窗口，可选择窗口右上角的关闭按钮“X”；要打开窗口，可选择“View”菜单中适当选项。例如，如果用户已经关闭了窗体布局窗口，而现在又要打开它，则可选择“View”菜单中的“Form Layout Window”选项。

要调整连接窗口中各部分窗口的尺寸，可将光标移至窗口的分界线上（此时光标变

为“”形状)，然后拖动光标即可（参见图 2-13）。

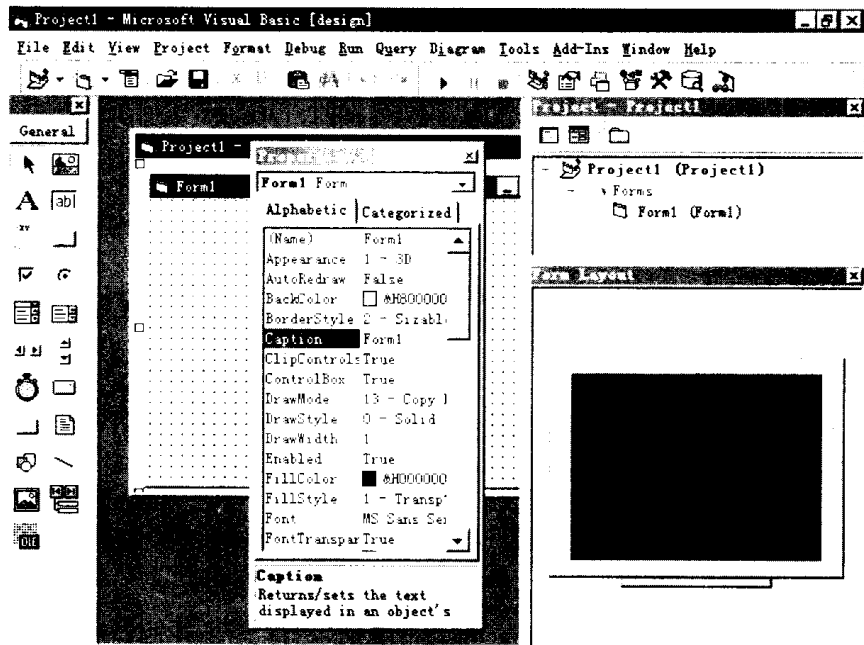


图 2-12 移动窗口后的结果

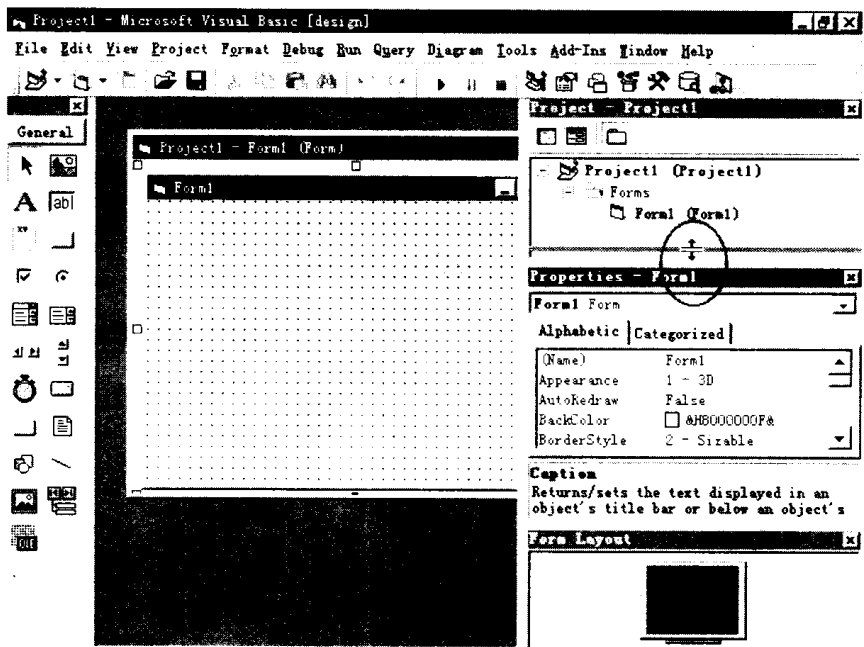


图 2-13 调整连接窗口中各子窗口的尺寸

2.3 Visual Basic 应用程序设计示例

创建 Visual Basic 应用程序是很简单的事。有多简单？为回答这个问题，试着创建下面的 Hello 和 Firstapp 应用程序。