引　言

对于编写 Windows 应用程序的人来说，Microsoft 公司的 Visual Basic for Windows 编程系统是一项令人兴奋的进步。Visual Basic 具有事件驱动的编程机制，新颖、易用的可视设计工具。它充分利用了 Windows 图形环境的优点，让开发人员快速地构造功能强大的应用程序。

但是，经验不足的程序员可能会猜疑 Visual Basic 是否适用。您也许已经掌握了诸如 Microsoft Excel 或 Microsoft PowerPoint 这样的应用程序，但却希望用计算机做更多的工作；或许您正在寻觅一种新的尝试，而对计算机编程发生兴趣；或许您已发现没有现成的商用程序能够完全符合自己的特定要求，所以正在考虑自己编写程序，以满足自己的特殊需要。

如果上述猜测之一成立，那么这本书正是您要找的书。本书是关于 Visual Basic 编程的初级读物。它虽然不能使您成为一个计算机专家，一个精明的玩家，或一个有希望成功的软件工程师，但它说明了怎样利用 Visual Basic，将您的想法转变为计算机的功能程序，以解决您的实际问题。

为什么会开发 Visual Basic 呢？为了了解 Visual Basic 是什么、以及为什么它在图形计算领域会产生这么大的震动，我们有必要先回顾一下历史。

历史回顾

最早的编程语言设计于 1950 年，其目的主要是为了解决复杂的数学问题。对于普通人来说，它是相当难懂的，但这在当时并不是一个严重的问题，因为当时只有在重要的研究室里才有计算机。当然，后来人们发现计算机技术的用途远不仅限于数学领域，它在企业和大学已得到了广泛的应用。当越来越多的人使用计算机时，深奥和复杂的编程语言就成了一种障碍。

因此，在六十年代初期，为适应当时的形势，Dartmouth 学院发明了一种称为“BASIC”的语言。原版本的 BASIC(Beginner's All-purpose Symbolic Instruction Code 的缩写)是一种非常简单的语言，设计它的特殊意图就是使学习编程简单化。一整代程序员在 BASIC 上付出了辛勤劳动，并用它编写了大量的程序。

BASIC 的简单性也使得它很小，当计算机也开始变小时语言的大小是至关重要的。麻省理工学院的 Altair 计算机诞生于 1975 年，它迎来了微机的革命。Microsoft 公司的共同缔造者 Bill Gates 和 Paul Allen 承担了为 Altair 开发一个 BASIC 版本的任务，它在计算机仅有的 4K 内存中运行，该 BASIC 版本最后发展成为个人计算机工业中最流行的产品。

若干年来，该编程语言得到了进一步的完善和发展。当早期的微机让位于 IBM PC 时，Microsoft 公司的 GW-BASIC 成为了标准。后来，对软件的快速、小型化和易用的要求导致了 Microsoft QuickBasic 语言的开发。QuickBasic 使 BASIC 语言加入到八十年代编程语言技术的行列，但不久，马上又出现了一种更大的变革——图形用户界面(GUI)。

Microsoft Windows 软件的出现，为 PC 用户提供了一个直观的、图形丰富的工作环境。图形用户界面使应用程序更易于学习和使用，用户只要简单地用鼠标器按钮点一下"菜单"中的选项就行了，而不必键入较长的命令。屏幕上的多窗口使用户可同时运行多个程序，当程序需要信息或需要用户决策时会出现对话框。

虽然 Windows 环境对于用户来说是相当出色的，但对于程序员来说，工作就变得比较困难了。程序员必须编写程序，去创建窗口、菜单、字体、对话框以及其它各种构件，甚至编写最简单的程序也不能例外。因此，在采用 Microsoft Windows 时，程序员是喜忧掺半——喜的是 Windows 给程序员提供了编写图形界面友好的应用程序平台，忧的是编程工作越来越复杂。

一个在屏幕上显示一条信息的简单程序在 MS-DOS 下编写时只需要 4 行语句，而在 Windows 下编写的类似程序需要 2～3 页代码；此外，程序员还要学习如何控制字体、菜单、窗口、内存和其它系统资源。但 Windows 对最终用户的好处是不容置疑的，数量众多的用户在购买 Windows 下的应用程序，专业程序员也只好痛下决心，开始编写冗长的代码。

许多人认为，Windows 预示着业余程序员的末日。在 MS-DOS 环境中，非计算机领域的专业人员一般都可学会足够的编程知识，编写简单的应用程序，以解决工作中的问题，如繁杂的计算及快速组织数据等。但在 Windows 中，最简单程序的编程工作都这么复杂，非计算机领域的专业人员还能进行编程工作吗？

用 Visual Basic 编写 Windows 程序

当 1991 年 Microsoft 公司开发出 Visual Basic 时，上述问题得到了肯定的回答。Visual Basic 编程系统用一种非常巧妙的方法将 Windows 的编程复杂性封装起来。Visual Basic 综合运用了 Basic 语言和新的可视设计工具，它既未牺牲 Windows 为之闻名的优良性能和图形工作环境，同时又提供了编程的简易性。菜单、字体、对话框、滚动正文域和所有其它构件的设计都相当容易，而且控制这些构件只需要为数不多的几行程序。

Visual Basic 也是第一批采用事件驱动编程机制的计算机语言之一。事件驱动是一种非常适用于图形用户界面的编程方式。传统的编程是一种面向过程、按顺序进行的工作，很象烹饪菜肴的指令：打鸡蛋、加牛奶、加糖搅拌、烤 20 分钟。这种编程方式的缺点是写菜谱(程序)的人总是要关心什么时候要发生什么事情。这对于烹饪来说是可以接受的，但在现代的计算机应用中，目的是让用户来操纵程序的运行。

这实际上就是事件驱动程序所要解决的问题。程序员只要编写响应用户动作的程序，如选择命令、在窗口中按鼠标器按钮、移动鼠标等，而不必编写按精确次序执行的每个步骤。我们不必编写一个大型程序，而是创建一个由若干个微小程序组成的应用程序，这些微小程序都由用户启动的事件来激发。利用 Visual Basic，程序员可以以空前的速度，方便地编写此类应用程序。

Visual Basic 1.0 版已获得了巨大的成功，出售了数万份拷贝并获得了大多数重要计算机杂志的表扬。1992 年秋天，Microsoft 推出了 Visual Basic 2.0 版，它提供了许多重要的新功能和特性。

在 Visual Basic 3.0 版推出后，Visual Basic 已成为一套成熟的编程系统，它具备许多功能强大的编程工具。Visual Basic 3.0 版包括以下新的特性：

- 性能的提高
- 数据库创建工具
- 使用 Data 控件进行可视数据存取，不编写代码就可以创建数据浏览应用程序
- 新 OLE(对象链接与嵌入)控件允许实地编辑对象
- 一系列普通对话框简化并统一了用户界面的设计工作
- 在应用程序中创建弹出式菜单的能力

《运行 Visual Basic 3.0 for Windows》(本书)介绍了 Visual Basic 的 3.0 版，帮助程序员提高实际编程水平。在阅读本书时，您不必知道很多的计算机专业词汇，也不必有丰富的编程经验，但在计算机上必须装有 3.0 版以上的 Windows 程序，当然，还必须熟悉 Windows 操作系统及一个以上 Windows 应用程序。如果您原来为试算表或字处理程序创建过宏程序，这将会对学习编程有帮助作用。最后，您不必害怕代数知识，这本书不包含很多的数学。但请记住，计算机语言起源于数学计算，因此，现代的计算机语言中都保留了一些数学计算功能。

学习编程实际上是学习一种新语言，确实需要一定的努力和实践。正如学习弹钢琴一样，在能弹奏交响乐之前，必须练习音阶和琶音。当读完这本书时，您还不能指望会编写与交响乐难度相当的程序。因为，本书着重的是基础知识，不会包括编程的所有内容，甚至也未完全包括 Visual Basic 的全部。但在读完本书后，您就可以建造一些有用的、不复杂的应用程序，以满足自己的需要。更重要的是可以马上掌握其它一些应用程序的编程技术，例如，Microsoft Word for Windows 用 Basic 的一个版本作为其宏语言，Microsoft Access(一种新的数据库管理程序)的编程技术基础与 Visual Basic 相同。

Visual Basic 的优良工具、能力和特性使得更为广泛的用户可以掌握编程技术，甚至掌握复杂的图形计算领域的编程技术。好啦，现在是您将计算机投入使用的时候了。

第一章　Visual Basic 入门

为了说明 Visual Basic 的易学性和用其编写实用应用程序的速度，本章将立即投入编程，并介绍一个简单的程序。实际上，使用 Visual Basic 是感性认识 Visual Basic 的最佳方法，这也将展露 Visual Basic 令人振奋的潜力。在本章的实例介绍之后，第二章和第三章将提供关于 Visual Basic 内部机制的详细说明。

1.1　安装和启动 Visual Basic

和大多数 Windows 应用程序一样，Visual Basic 有一个名为 SETUP.EXE 的程序，它负责将 Visual Basic 安装在计算机的硬盘上。Visual Basic 的文档详细描述了整个安装过程，而实际上只要简单地执行以下四个步骤：

1. 在 MS-DOS 提示符下键入 Win 并按回车键，启动 Windows。Visual Basic 可在 Microsoft Windows 3.0 以上的版本中运行。

2. 在软盘驱动器中插入标有"Disk 1"(1 号盘)的软盘。

3. 从 Windows 程序管理员(Program Manager)的 File(文件)菜单中选择 Run(运行)命令。如果 Disk 1 插在 A 驱动器中，则键入 a:setup 和回车键；如果使用驱动器 B，则键入 b:setup 和回车键。

4. 遵循屏幕上的提示，并回答 Setup 提出的各种安装选项问题。

在回答完所有问题后，Setup 程序将把 Visual Basic 安装在计算机的硬盘上，并在程序管理员(Program Manager)窗口中加入一个 Visual Basic 组窗口和图标。

启动 Visual Basic 最简单的方法就是在程序管理员窗口中双击* Visual Basic 图标，也可以双击 Windows 文件管理员窗口中代表 Visual Basic 执行文件 VB.EXE 的图标。另外，还可以通过在 MS-DOS 提示下键入 win vb 命令同时启动 Windows 和 Visual Basic。图 1.1 说明了 Visual Basic 的三种启动方法。

1.2　Visual Basic 的屏幕布局

启动 Visual Basic 之后，屏幕上会出现五个窗口，如图 1.2 所示(在读者的 Visual Basic 屏幕中，有些窗口可能会重叠。为了清楚地观察各个窗口，图 1.2 中的窗口已进行缩放和重新定位)。屏幕的顶部是主窗口，它包含标准的文件(File)和编辑(Edit)菜单以及 Visual Basic 的其它菜单与工具条。屏幕中心是窗体(form)窗口，即标题为 Form1 的空窗口；在其左边出现的是一个称为工具箱的调色板式的窗口。窗体窗口的右边是属性(Properties)窗口，而属性窗口之下是项目(Project)窗口。

注意：在图 1.2 中，除 Visual Basic 之外的其它应用程序都缩成了图标。使应用程序缩小成图标可以解决屏幕混乱的问题。从程序管理员的选项(Options)菜单选择 Minimize On Use

---

* 双击(double click)——将鼠标指针置于某个对象之上，再连按两下鼠标左键。
——译者注

选项是有帮助作用的。当选中该选项后启动新程序时，程序管理员窗口自动缩小成图标。

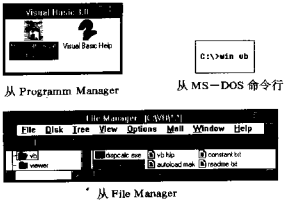


图 1.1 启动 Visual Basic 的三种方法

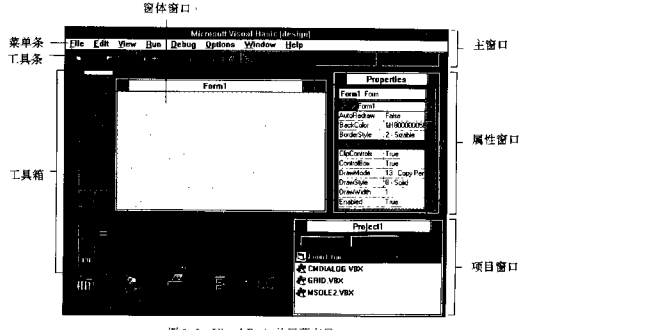


图 1.2 Visual Basic 的屏幕布局

也许您会抱怨这样太复杂了！它确实比较复杂，但降低复杂性的关键是组织，因此，Visual Basic 环境设计的目标就是保持屏幕的组织性。下面，让我们分别考察一下这五个窗口。

1.2.1 主窗口

主窗口包含菜单条，它有八个下拉式菜单。其中最重要的（尤其是学习 Visual Basic 的时候）是帮助(Help)菜单，从该菜单中可以查阅 Visual Basic 的使用指导，找到 Microsoft 产品支持服务机构的联系方法。现在，我们来考察一下复杂的 Visual Basic 联机帮助系统。

当选择 Help 菜单中的 Contents 时，我们将看到联机帮助系统中所含信息的分类表。用鼠标单击*其中的下划线词组，可以移动到需要查阅的特定段落和主题。当选择 Help 菜单中的 Search(搜索)命令时，会显示一个对话框，让您指定一个特定的主题，并马上显示该主题的帮助信息。

另外，Help 还提供了与上下文相关的帮助信息。若想了解关于按钮、对话框、窗口、错误信息或 Visual Basic 中的其它构件，只要在该构件醒目显示时按 F1 键，帮助系统将立即显示相关的帮助信息(第八章完整地描述了 Visual Basic 的帮助系统)。

主窗口还包含工具条。图 1.3 中工具条上的按钮是常用命令的捷径按钮。例如，不必打开 File 菜单再选择 Open Project 命令，只需单击 Open Project 按钮即可。

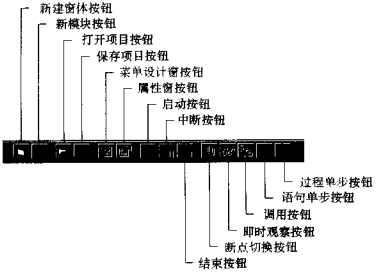


图 1.3 Visual Basic 工具条

最后，主窗口工具条右边有两个显示域，它们分别指示了窗体窗口中当前被选对象的位置和大小。

1.2.2 项目窗口

项目窗口包含了运行正在编写的 Visual Basic 程序所需的文件清单。虽然我们还没有开始设计程序，项目窗口中已出现了三个文件名，如图 1.2 所示。第一个文件名是 Form1.frm，右侧的标签(Form1)表明了该文件是与称为 Form1 的窗体窗口相关联的。如果不改变名字就在磁盘上存储窗体，Visual Basic 将使用缺省文件名 Form1.frm。

一个应用程序可由多个窗体组成，每一个窗体都可存储在一个文件中(因为文件是独立的，所以，其中的窗体也可以由其它应用程序共享)。

* 单击(click)——将鼠标指针置于某个对象之上，再按一下鼠标左键。请与双击比较。
——译者注

项目窗口中的其它两个项是文件名 GRID. VBX 和 OLECLIEN. VBX。文件扩展名 VBX 表示这些文件是 Visual Basic 的扩展文件。当装载一扩展文件时，则工具箱调色板中会增加一些辅助工具(在下节中讨论)。

项目窗口还包含两个按钮，标签名分别为 View Form(查看窗体)和 View Code(查看代码)。在项目窗口中选择一个文件时，Visual Basic 将显示相应的窗体，窗体可用于设计应用程序的用户界面，即应用程序中用户可以看见并与之交互的部分。如果按下项目窗口中的 View Code 按钮，选中文件的代码将出现在不同的窗口中。**代码**指的是编程语言中的语句，因此，编程的过程也称为**编码**。用 Visual Basic 创建程序时，工作可以分成两部分，即窗体设计(用户界面)和编码(控制程序的操作)。如果要从代码窗口切换到窗体窗口，只要在窗体上单击一下，也可按项目窗口中的 View Form 按钮。

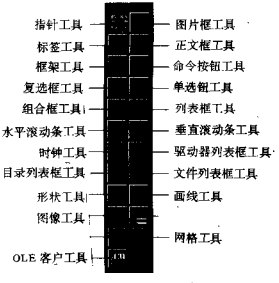


图 1.4 Visual Basic 工具箱

1.2.3 窗体窗口和工具箱

窗体是一个对应于应用程序运行时所见的窗口的显示区域。当启动一个新项目时，Visual Basic 创建一个空的窗体，并命名为 Form1。在设计应用程序时，窗体就象一块画布，在画布上可以画出组成应用程序的各个构件，窗体上的应用程序构件称为**对象**(object)或**控件**(control)— 如图片框、单选钮和滚动条(另外，Visual Basic 将窗体本身也视为对象)。

控件创建于工具箱调色板，如图 1.4 所示。每个控件由一个工具图标(简称为工具)表示。图 1.4 中的大多数工具都是 Visual Basic 固有的(第五章将详细描述这些标准工具)。当然，工具箱可以扩充，以包含新的工具。项目窗口中扩展名为 VBX 的文件均为工具箱提供了一种或多种新工具。例如，工具 Grid 和 OLE Client 分别由文件 GRID. VBX 和 OLECLIEN. VBX 提供，它们是项目文件指定的缺省工具。

Microsoft Visual Basic Professional Edition(专业版)提供了更多的 VBX 文件，并允许程序员设计新的工具(对于 C 或 Pascal 程序员)。同时，第三方软件开发商还提供了大量其它工具。

通过从工具箱中选择控件并把它们放到窗体中，即可设计程序的外貌。在设计应用程序时，Visual Basic 的操作和运行该应用程序是不一样的。在设计阶段，Visual Basic 提供了用于创建显示对象和编写程序的工具。在窗体上改变对象的大小、位置和其它属性，可以控制对象的外观和行为，但这些对象不是活动的，代码也不执行。例如，在窗体上放置一个滚动条(scroll bar)对象后，我们可以改变该对象的大小和位置，但不能用它滚动任何东西。

当然，在运行编写完成后的应用程序时，Visual Basic 会移去设计工具。窗口布局是设计阶段的窗体，屏幕上的对象可以被激活，用户可以按我们设计的方式与显示区进行交互。应用程序一旦启动，Visual Basic 就开始执行程序语句(在主窗口的标题条中，Visual Basic 会提示是要设计应用程序还是运行应用程序)。

1.2.4 属性窗口

Visual Basic 的**属性**(Properties)是描述对象性质的形式化(formal)机制。在现实世界中，我们可以问"这头牛是什么颜色"？而在 Visual Basic 中，问题则变为"这头牛的颜色属性值是多少"？这两个问题的答案当然都一样，即是"紫色"。品种是牛的另一种典型属性，品种属性的期望值(或设置值)可能包括水牛、黄牛和牦牛等。

每个 Visual Basic 对象都有特殊的属性，其设置值控制了该对象的外观和行为。有些属性仅限于一定的值，例如，对象的可见性只能设置为 True 或 False(即可见或不可见)。其它的属性(如某窗体窗口的标题)几乎可设为任意正文。注意，不必设置每个对象的所有属性，许多属性可采用其缺省值。

虽然在设计阶段和应用程序运行时可改变许多属性，但属性窗口(如图 1.5 所示)只有在设计阶段才是活动的。激活属性窗口有多种方法，包括简单地单击属性窗口，在 Windows 菜单中选择 Properties 命令，按 F4，或选择工具条上的 Properties Window 按钮。

图 1.5 属性窗口

属性窗口顶部的下拉式列表框称为对象框，它显示了应用程序中每个对象的名字及对象的类型。初始时，对象框只包含该窗体的信息，但将控件加入到窗体中时，Visual Basic 会把这些对象相应地加入到对象框的下拉列表中。

位于对象框之下的是设置框和属性列表，属性列表允许滚动对象框中显示的对象的所有属性，以便观察每项属性的当前设置值。当从该列表中选择一项属性时，其当前设置值会出现在属性列表上的设置框中。如果希望改变设置值，可以根据特定的属性，在设置框中键入一个新的条目，或从下拉列表中选择一个新的、预先定义的设置值。

为了说明怎样用各种 Visual Basic 窗口来创建应用程序，让我们开始设计第一个项目。

1.3 创建用户界面

作为第一个项目，让我们创建一个简单的程序，它只记录走过的时间，就象一个秒表一样。秒表有一个计时启动按钮，还有一个计时停止按钮，在表的正面可以观察走过的时间。这种模型可以作为程序的基础，在 Visual Basic 中，窗体相当于秒表表面，在窗体上面也有启动按钮和停止按钮。在开始设计应用程序之前，首先需要检查和修改一些窗体的属性值。

1.3.1 设置窗体的属性

在运行应用程序时，窗体将作为一个标准应用程序窗口出现。如果希望该窗口类似于其它 Windows 应用程序显示的窗口，则应该使它具有类似的属性，例如，Windows 应用程序一个共同的属性就是应用程序的名字作为其标题出现在其标题条中。在 Visual Basic 中，许多这样的属性均由对象的属性来控制。

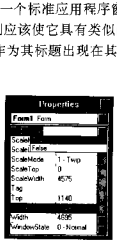


图 1.6 设置窗体的可见属性

在开始设置窗体的属性时，首先要在标名为 Form1 窗口中的任何地方按一下鼠标键，选择该窗体，使其成为当前对象。现在，请看 Visual Basic 的属性窗口，Form1 被显示在对象框中。如果要改变窗体的标题，应单击属性列表中的标题(Caption)属性。当前(缺省)的标题 Form1 将出现在设置框中，键入 Stop watch，将它作为新的标题，在键入之后，该新标题会显示在窗体窗口的设置框和标题条中。

注意设置框中左边的两个按钮，它们分别带一个“×”和“√”标签。“√”按钮用于确认设置框中所输入的设置值，按下该按钮等效于按回车(Enter)键。按“×”按钮则取消当前输入，并在设置框中恢复以前的设置值。

当选中的属性只能接受一组有限的预定义值时，设置框的作用就象一个下拉列表一样，而不是作为一个正文输入域。例如，滚动属性列表，并选择可见(Visible)属性。现在，设置框边上的箭头按钮已激活，说明可见属性的设置值仅限于下拉列表中的值。如果单击向下箭头按钮，该列表将拉下来，以显示设置值 True 和 False，如图 1.6 所示。注意，可见时，属性应设为 True。

和窗体外貌有关的另一属性是 BorderStyle(边框类型)，它控制着运行应用程序时用户是否可以改变显示窗口的大小。因为秒表应用程序将有一个固定的外貌，其窗口的大小不允许改变，所以，应从属性列表中选择 BorderStyle 属性，并从设置框的下拉列表中选择 3-Fixed Double，将该属性设为 3-Fixed Double 值(Fixed Double 表示窗口的边框大小不能改

变，并且不带最小和最大按钮）。

Name（名称）属性

每个 Visual Basic 对象都有一个称为 Name 的属性。当设置 Name（名称）属性时，即给对象取一个标识名，在程序中可用它表示对象。在窗体窗口中，单击一个对象就可以访问该对象；但在程序的代码中，访问对象必须使用名称属性中所指定的名字（注意，Name 属性和 Caption 属性是不同的。对象名称是对象在程序代码中的标识符，而对象标题是用户在屏幕上所见到的——即在应用程序窗口中显示的标识正文）。在属性列表中，选择窗体的名称属性，其缺省值为 Form1，它显示在设置框中。请您在设置框中键入正文 MyForm 以改变窗体的 Name 属性。

注意：当改变窗体的名称时，项目窗口中显示的文件名列表中会反映这一变化。请记住，每个窗体的定义存储在一个独立的文件中。当保存应用程序时，您可以根据个人的喜好来选择文件名，但 Visual Basic 会根据窗体名称，为您推荐一个缺省文件名。

1.3.2 增加显示对象

现在让我们返回窗体的设计，创建用于启动和停止计时器（秒表）的两个按钮，步骤如下：

1. 单击工具箱中的命令按钮（Command Button）工具，再将鼠标移到空的窗体窗口。这时，光标会变成十字光标，表示正处于绘图模式；

2. 将光标放在窗体的左上角；

3. 按住鼠标器按钮不放，将光标向右下角拖动；

4. 放开鼠标器按钮。

在放开鼠标器按钮之后，Visual Basic 在拖动操作所定义的矩形区域内创建一个命令按钮对象，如图 1.7 所示。

在创建命令按钮后，工具箱中的箭头工具又变成了活动的工具。当箭头工具激活时，我们可以移动对象和改变对象的大小。若要移动对象，只需把它拖动到新位置；如果要改变对象的大小，可拖动它的句柄（如图 1.7 所示）以扩大或缩小对象；如果句柄不可见，可单击该对象，这时它会出现句柄。

现在，开始创建第二个命令按钮，这次使用一种更简单的方法（捷径方法）。请双击工具箱中的命令按钮工具，Visual Basic 会创建一个缺省大小的按钮，并把它放在窗体的中心位置。现在，再移动并缩放第二个按钮，使两个按钮都在屏幕的左边，并且大小相同。

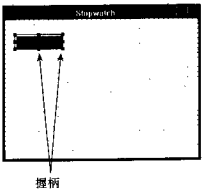


图 1.7 一个新创建的命令按钮

这时，两个按钮的缺省标签为 Command1 和 Command2，它们分别作为秒表应用程序的启动按钮和停止按钮。为了清楚起见，应赋予它们相应的标题。在设置这两个按钮的属性时，应遵循设置窗体属性同样的过程，即：

1. 单击上方的命令按钮，选中该按钮；
2. 从属性窗口的属性列表中选择 Caption 属性；
3. 在设置框中，将设置值改为 Start；
4. 单击下方的命令按钮并将其标题改为 Stop。

尽管刚才已改变了标题，还应改变按钮的名字，使得在程序代码中可以方便地引用它们（请记住，Caption 属性和 Name 属性是两个不同的属性，改变其中一个并不影响另一个）。单击上方的命令按钮并选择其 Name 属性。在设置框中，将缺省名 Command1 改为 btnStart，然后，单击下方的按钮并将其名称改为 btnStop。

名称的选取

在该秒表（Stopwatch）应用程序中，改变按钮的名称不是绝对必要的，但很显然，在程序代码中引用 btnStart 会比 Command1 更清楚。T. S. Eliot 曾说过："猫的命名是很困难的"；同样，程序中对象的命名也是相当困难的。简短的名字易于键入，系统提供的缺省名也很方便，但在编程六个月后再修改该程序时，如果名称没有任何意义，要了解程序做什么无疑是非常困难的。

同时，还应使用一致性的命名法则，它同时能提供上下文和描述。例如，名称 btnStart 和 btnStop 带三个字母的前缀，它标明了程序代码中对象的类型。在英文中，经常用描述性的词组建立上下文。例如，不是简单地称"Cardinal Fang"而是称"我的猫 Cardinal Fang"，以避免我的猫和一些基督教会的官员之间产生混淆。当然，在 Visual Basic 中名称不能写为"the button named Start"（名称为 Start 的按钮），所以，有必要压缩上下文（btn）和名字（Start），使它们变为单个标识符（btnSart）。

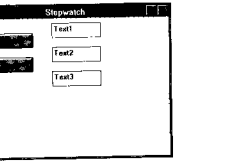
1.3.3 完成窗体设计

现在，我们已在屏幕上为秒表应用程序定义了两个用户按钮，但还需要程序运行时用于显示结果(时间)的显示区。下面将给窗体增加几个正文框。

1. 双击工具箱中的正文框(Text Box)，创建一个正文框。该正文框将出现在窗体的中央；

2. 将正文框移动到窗体窗口的上部，并放在命令按钮的右边；

3. 使用同样的方法创建另外二个正文框，参照下图，排放这三个正文框；



4. 用鼠标"抓住"窗体窗口的右下角，改变窗体的大小，使它刚好包住命令按钮和正文框。

现在，我们来给这三个新对象命名。单击每个正文框并改变其名称属性的设置值。对于上方的正文框，给它起名为 txtStart，中间的正文框起名为 txtStop，下方的正文框起名为 txtElapsed。

此外，还需要删除正文框中出现的缺省正文。对于每个正文框，从属性列表中选择正文(Text)属性，并删除设置框中的所有字符，亦即，将正文属性值都设为空(长度为零的正文串)。

现在，应用程序用户界面中的所有必要构件都已准备就绪。事实上，我们现在就可以"运行"该应用程序，从 Run(运行)菜单中选择 Start(启动)命令，单击工具条中的启动按钮，或按 F5 键。工具箱会消失并出现 Stopwatch(秒表)窗口，如图 1.8 所示。该窗口可以在屏幕上移动，也可以按下 Start 或 Stop 按钮(因为我们还未编写计算走过时间的任何 Visual Basic 语句，所以什么也不会发生)。从 Visual Basic 的 Run 菜单中选择 End(结束)命令或单击工具条中的结束按钮即可退出该应用程序，并返回 Visual Basic 的设计环境。

图 1.8 运行中的秒表应用程序窗口

1.4 编写代码

编码，即编写控制程序操作的语句，是编程的核心。前面，我们已设计了应用程序的界面，现在需要的是使程序运转起来，也就是说，我们要用 Visual Basic 语言编写命令，让这些命令间接地控制计算机。为了了解这一概念，下面以喷气飞机为例说明这一问题。

当飞行员希望喷气式飞机飞高一点时，飞行员必须将飞机座舱中的控制杆向后拉。这一动作的效果是改变飞机升降舵的位置，飞机开始向上飞行。在现代的喷气式飞机中，操纵杆和升降舵之间没有物理上的连接关系。当飞行员向后拉操纵杆时，传感器检测操纵杆运动的距离，并将一个信号发送给液压设备，该液压设备会调整升降舵的位置。当然，从飞行员的角度来看，操纵杆控制着飞机，操纵杆 1 英寸转化为升降舵转 5 度还是转化为 10 度，对于飞行员来说并不是关键问题，飞行员只要知道飞机响应操纵杆的变化就行了。

和喷气式飞机的飞行员一样，程序员控制着计算机，Visual Basic 就是仪器操纵板。以下部分集中说明系统的"驾驶舱布局"——即 Visual Basic 程序员所看到的东西。当然，在这个外表之下蕴涵了许多丰富的内容，但在这里，我们只集中讨论总体问题，而将系统相关细节的讨论留在第二章和第三章中进行。

1.4.1 事件过程

前面，我们已经学习了对象的属性及怎样修改这些属性，以影响对象的外貌和行为。同样，每个对象都可以和一系列在特定时间执行的**过程**(procedure)相关连。所谓过程，是指一组 Visual Basic 语句。当过程运行时，就执行过程中的语句，所有可执行代码将封装在这种或那种过程中。

和对象相关联的每个过程都对应着一个特定的**事件**(event)或动作，因此称为事件过程(event procedure)。事件包括动作，如按鼠标器按钮(当用户按鼠标器按钮时触发)和改变大小(当用户改变窗体的大小时发生)。事件只发生在运行时刻，而不会发生在设计阶段。对于不同的对象，可以触发许多不同的事件，第五章将详细描述这一问题。

现在，让我们回到秒表应用程序。在窗体窗口中双击 Start 按钮，Visual Basic 将打开一个如图 1.9 所示的代码窗口(code window)。代码窗口的标题为 MyForm.frm，它表示代码和窗体之间的关系。在本窗口中输入的代码将存储在包含窗体对象的同一个文件中。

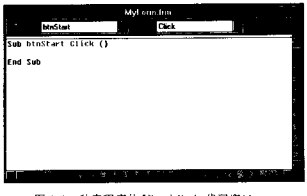


图 1.9 秒表程序的 Visual Basic 代码窗口

代码窗口左上部的对象框显示了选中对象的名称：btnStart，右上角的过程框(缩略为Proc)显示了正在编辑的事件过程。因为还没有选择过程，所以 Visual Basic 选 tnStart，右上角的过程框(缩略为 Proc)显示了正在编辑的事件过程。因为还没有选择过程，所以 Visual Basic 选择一个可能的缺省值，即单击鼠标器按钮(Click)事件过程。代码窗口中显示的过程是在运行应用程序时用户单击 Start 按钮时执行的过程。虽然还未编写过任何代码，但在代码窗口的正文部分已出现了下列两条语句：

```
Sub btnStart_Click ()

End Sub
```

在创建此命令按钮时，Visual Basic 已为该对象创建一个缺省的事件过程。也许您已猜想到，缺省过程不做任何事情，它仅由过程声明(第一行)和表示过程结束的语句(End Sub)组成。

过程声明

在过程声明中，关键字 Sub 表示过程开始，后面跟过程名(在此为 btnStart_Click)。接下来是一对小括号，它结束过程定义(虽然必须包括小括号，但本过程并未利用小括号，其功能将在以后的章节中讨论)。

过程名用于表明它是一个事件过程，并不是所有过程都是事件过程。我们可以用 Visual Basic 创建一个过程，并起一个喜欢的名字——只要喜欢，甚至可起名为 Spot(污斑)或 Puff(喘息)都可以。但事件过程名必须遵循下述命名规则：

- 名称的第一部分必须和窗体上创建的对象名相匹配(当相应的对象是窗体本身时，名称必须为窗体名)；
- 名称的最后一部分必须为事件名；
- 两部分必须用下划线(_)分开。

对于 btnStart_Click 过程，相关联的对象为 Start 命令按钮(btnStart)，事件为单击鼠标器按钮(Click)事件。当程序运行时，剧情如下：应用程序在 CRT 中挂起，等待事件的发生；此时，用户珍妮带来一个鼠标器，她将鼠标移到屏幕的图像上并按一下鼠标按钮，Windows 在应用程序的裤子上踢了一下，说："喂，起来，干活啦！"。当 Visual Basic 发现鼠标动作发生在名为 btnStart 的按钮之上时，它则寻找 btnStart_Click 过程，并执行该过程中的语句。作为程序员，要做的工作就是决定这些语句是什么。

为了感性认识这一工作，在 Sub 和 End 语句之间的空行上键入语句 Debug. Print "Hello，Sailor"。这时代码窗口如下所示：

```
Sub btnStart_Click ()
    Debug. Print "Hello，Sailor"
End Sub
```

Debug. Print 语句告诉 Visual Basic 在一个名为 Debug 的特殊窗口中显示正文。请按 F5 键退出设计模式并开始运行应用程序。当按下 Start 按钮时，Debug 窗口会显示 Hello，Sailor 语句。

现在从 Visual Basic 的 Run 菜单中选择 End 命令，Visual Basic 将停止应用程序的执行并返回设计模式，在设计模式中可继续设计应用程序（在进一步设计应用程序之前，请记住删去 btnStart_Click 过程中的 Debug.Print 一行）。

提示：关键字（如 Sub、Debug、Print 和 End）在代码窗口屏幕上以不同于其它代码的颜色出现。Visual Basic 用不同的颜色标识不同的代码部分，使您可以方便地找到各个部分。选项（Options）菜单中的环境（Environment）命令用于控制这些颜色。当出现环境选项（Environment Options）对话框时，您就可以设置关键字、标识符、注释和其它代码的正文颜色和背景颜色。

变量声明

当用户按下秒表应用程序 Stopwatch 中的 Start 按钮时，程序必须给出秒表启动的时刻；而当用户按下停止（Stop）按钮时，程序必须用结束时刻减去开始启动时刻，计算出秒表走过的总时间。当然，完成这些动作必须编写程序代码。

Visual Basic 使用一个名为 Now 的函数提供当前时刻。**函数**是一种特殊的过程，它返回一个值。

Visual Basic 函数和数学函数的概念相当接近。例如，Visual Basic 函数 Sin、Cos 和 Tan 分别返回某一角度的正弦、余弦和正切值。当用户按下 Start 按钮时，程序必须调用函数 Now，它返回一个时间值——当前（启动）时刻，程序必须记住这一时间，以便在用户按下停止（Stop）按钮时，可以用结束时刻减去它。为了在计算机程序中存储一个值，必须在计算机内存中为该值保留一个空间。变量声明就是完成这项工作的。

在代码窗口中，按下对象（Object）框中的向下箭头按钮，会拉下一个窗体中所有对象的列表，从该列表中选择条目 general，它包含与特定对象无关联的所有代码。Visual Basic 会自动地将 Procedure 框中的设置值改为（declarations）（在本书中我们称它为代码的通用声明段）。现在，请在代码窗口中输入以下几行：

```
Dim StartTime As Variant
Dim EndTime As Variant
Dim ElapsedTime As Variant
```

这些语句告诉 Visual Basic 希望在称为 StartTime、EndTime 和 ElapsedTime 三个变量中存储启动、结束和走过的时间值。变量是在计算机内存中保留空间的最基本的名称（第二章和第三章将进一步描述变量）。Dim 语句告诉 Visual Basic 保留内存空间的类型及名称。因为 Dim 告诉 Visual Basic 应该怎样建立程序，所以 Dim 是一种声明。

上述每条语句都声明了一个变量，三条语句的一般形式如下：

```
Dim name As Variant
```

Dim 和 As 是**关键字**，即 Visual Basic 中有特殊意义的保留字。因为三个变量的名称各不相同，所以才需三条语句。关键字 Variant 告诉 Visual Basic 为该类型的值保留足够的内存空间（可用其它关键字使声明更特殊化，如将 Variant 改为整数 Integer 时，它将告诉 Visual Basic 只保留存储一个整数的空间）。

Visual Basic 的自动语法检查

如果在代码窗口键入代码时发生了错误，Visual Basic 将醒目显示该错误，并显示一个消息对话框来通知程序员。这一监测手段称为自动的法检查，它在按回车键后执行。当然 Visual Basic 不能检查出所有键入错误，但它可以检查出句法错误——即关于怎样使用 Visual Basic 语言的错误，如关键字的拼写错误、语句中字的次序颠倒、不正确的标点等。例如，如果键入了 txtStop.. Text，而不是键入了正确的 txtStop. Text，Visual Basic 将提示出现了错误。如果愿意的话，您也可以关闭自动句法检查。方法是：从 Options 菜单中选择 Environment 命令，并在 Environment Options(环境选项)对话框中将 Syntax Checking(句法检查)选项设为 No。

术语

如果认为 Dim 是个相当蠢的字，而认为 Declare 比较合适，这种看法是正确的。但是，Dim 是 Dimension 的缩写，它是 Basic 语言最早版本的产物，其来源可以追溯到本世纪六十年代。当计算机语言发展时，为了保持其兼容性，一般要保持原版本的一些特性。

可执行语句

现在，我们已为存储时间的值申请了内存空间，故可以编写程序的可执行代码。与设置程序的声明相反，可执行语句(executable statement)在程序执行时做某些事情，例如，以前使用过的 Debug. Print 语句就是一条可执行语句。

在代码窗口的对象框中，选择 btnStart 对象，在 Sub btnStart_Click 和 End Sub 语句之间输入以下语句来编辑 btnStart_Click 过程：

```
StartTime = Now
txtStart. Text = Format(StartTime,"hh:mm:ss")
txtStop. Text = ""
txtElapsed = ""
```

在运行程序时，若用户按 Start 按钮，这些命令就会执行。代码的第一行用 Now 函数取出当前时间值，然后将它存储在名为 StartTime 的变量的存储位置。第二行设置该对象的 Text 属性，在最上面一个正文框中显示时间，它使用 Visual Basic 的 Format 函数将时间格式变为小时、分和秒。最后两行将正文属性设为空串，将其它两个正文框(txtStop 和 txtElapsed)中的原有显示内容清除掉。

最后，我们再选择代码窗口中对象框中的 btnStop 对象来完成该程序，按以下方式编辑 Click 事件过程：

```
Sub btnStop_Click ()
    EndTime = Now
```

```
    ElapsedTime = EndTime - StartTime
    txtStop.Text = Format(StopTime,"hh:mm:ss")
    txtElapsed.Text = Format(ElapsedTime,"hh:mm:ss")
End Sub
```

当用户按 Stop 按钮时执行以上过程。首先，Now 函数返回当前时间值，并将它存储在 EndTime 变量中；第二行用 EndTime 值减去已存储的 StartTime 值，计算走过的时间。然后再设置 txtStop 和 txtElapsed 的正文(Text)属性，在屏幕上显示出结束时间和走过的时间。和 btnStart_Click 过程一样，Format 函数将时间值转化为小时、分、秒的格式。

1.5 运行程序

只要按 F5 就可以启动应用程序 Stopwatch。按 Start 按钮启动"秒表"，等待几秒钟后，再按 Stop 按钮，这时会看到类似于图 1.10 所示的结果。

可再次按 Start 按钮重新启动秒表。如果要终止应用程序 Stopwatch，可以单击工具条上的结束按钮，也可按 Alt-F4 或执行控制菜单中的 Close 命令(单击窗口左上角的框时会出现控制菜单，有时也叫做系统菜单)。

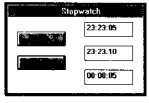


图 1.10 运行中的秒表应用程序

1.6 设计复审

程序员很容易在编码的细节上徘徊，而忘记了程序的用户。让我们来审查一下秒表用户界面的设计：两个按钮、三个正文框。这种设计相当简洁，但要考虑到不知道该程序用途的人，Start 和 Stop 两词都有自我解释的特性，但从三个空正文框中又能看到什么信息呢？一个明显的改进方案就是给这三个正文框加上"标签"，使它们所显示信息的含义一目了然。

工具箱中的标签(Label)工具图标看起来象一个大写字母 A。借助标签工具，我们可以在窗体窗口最上面的正文框边上画一个标签对象，这时需要改变窗体的大小并将正文框向右移动，以便给标签留出空间。也可以一次同时选中三个正文框，并作为一组同时移动它们，方法是：简单地单击最上面的正文框，按住 Ctrl 键不放，再单击其它两个正文框。

然后，单击新的标签对象，在属性列表中选择标签的 Caption 属性，并在设置框中键入 Starting Time。同样，在其它两个正文框的边上创建两个相似的标号，分别将其 Caption 属性设置为 Ending Time 和 Elapsed Time，如图 1.11 所示。