

全国普通高校适用教材

Visual Basic 程序设计

实验指导与习题

薛晓萍 吴欣明 郝晓军 主编

Visual Basic



科学普及出版社

全国普通高校适用教材

Visual Basic 程序设计

实验指导与习题

薛晓萍 吴欣明 郝晓军 主 编

科学普及出版社

· 北 京 ·

图书在版编目 (CIP) 数据

Visual Basic 程序设计实验指导与习题/薛晓萍, 吴欣明, 郝晓军, 主编. —北京: 科学普及出版社, 2009. 1

ISBN 978-7-110-07024-6

I. V... II. ①薛... ②吴... ③郝... III. BASIC 语言—程序设计—教学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 198321 号

自 2006 年 4 月起本社图书封面均贴有防伪标志, 未贴防伪标志的为盗版图书

策 划: 陆为民
责任编辑: 吕秀齐
责任校对: 赵丽英
封面设计: 严 忠
责任印制: 安利平

科学普及出版社出版

北京市海淀区中关村南大街 16 号 邮政编码: 100081

电话: 010-62103210 传真: 010-62183872

科学普及出版社发行部发行

北京玥实印刷有限公司印刷

*

开本: 787 毫米×1092 毫米 1/16 印张: 13 字数: 250 千字

2009 年 1 月第 1 版 2009 年 1 月第 1 次印刷

定价: 18.00 元

ISBN 978-7-110-07024-6/TP·201

《Visual Basic 程序设计》实验指导与习题

编委会

主 编 薛晓萍 吴欣明 郝晓军

副主编 米新江 王晓丰 李春伟

王怀军 许 维

编 委 (以下按姓氏笔画为序)

王怀军 王晓丰 米新江

许 维 吴欣明 张秀芳

李国峰 李春伟 李海健

闵 锐 阿建卓 郝晓军

薛晓萍

前言

“Visual Basic 程序设计”是一门实践性很强的课程，要真正掌握 Visual Basic 程序设计，就必须安排充足的上机实验。本书是《Visual Basic 程序设计》理论教材的配套实验教程，全书贯穿着实践性、实用性的编写原则。本书根据课程的教学内容和教学进度，针对各章节的重点和难点，结合教学大纲设计了相应的上机实验。全书分为实验基础、实验内容、练习题和附录 4 个部分。

实验基础部分主要介绍 Visual Basic 的运行环境、程序的调试方法和技巧，如何进行错误处理，如何及时发现并排除程序中的错误，以及如何发布应用程序。这一部分内容对开发 Visual Basic 应用程序非常有用，读者应充分重视。

15 个主题实验是本书的核心内容。每个实验都针对某个具体应用精心设计，并以该应用为主题展开。实验内容力求突出代表性、典型性和实用性，每个实验都作了详细的分析和提示，同时给出了详尽的解答和程序代码。本书采用了实验教学中的“实验目的——实验内容——实验步骤——实验思考”的组织结构。实验内容与理论教学同步，使学生通过实验灵活应用理论知识。

练习题部分主要给出全国计算机等级考试部分模拟习题及参考答案。

附录部分给出了全国计算机等级考试 Visual Basic 考试大纲。

全书注重培养和提高学生在集成环境下调试程序和处理错误的能力，力争使学生在本书的指导下，充分发挥主动性，独立完成各个实验。

本书可作为高等学校非计算机专业《Visual Basic 程序设计》课程的配套实验教材，也可作为各类计算机水平考试和编程爱好者的自学辅导教材。书中的所有实例全部在 Visual Basic 6.0 环境下调试通过。

由于时间仓促，加之编者水平有限，书中难免有欠妥之处，诚恳希望读者不吝指正。

编者
2009 年 1 月

目 录

第一部分 实验基础	1
第一章 VISUAL BASIC集成开发环境	1
1.1 Visual Basic 6.0 简介	1
1.2 Visual Basic 6.0 的启动及基本界面	2
1.3 Visual Basic应用程序的编程步骤	6
1.4 程序的运行	6
第二章 程序调试与错误处理	8
2.1 Visual Basic的工作模式	8
2.1.1 设计模式	8
2.1.2 执行模式	9
2.1.3 中断模式	9
2.2 错误类型	9
2.2.1 语法错误	9
2.2.2 编译错误	10
2.2.3 运行错误	10
2.2.4 逻辑错误	11
2.3 程序调试	11
2.3.1 “调试”工具栏	11
2.3.2 使用调试窗口	12
2.3.3 使用中断	13
2.4 错误处理	13
2.4.1 错误的捕捉及处理子程序	14
2.4.2 Error 函数与Error语句	15
2.4.3 设计错误处理程序	16
第三章 发布应用程序	18
3.1 制作可执行文件	18
3.2 使用打包和展开向导	19
第二部分 实验指导	25
实验一 VISUAL BASIC 6.0 集成开发环境与简单应用程序设计	25
实验二 数据类型、运算符、表达式和函数	30
实验三 顺序结构程序设计	35
实验四 选择结构程序设计	40

实验五 循环结构程序设计	49
实验六 常用控件（一）	53
实验七 常用控件（二）	58
实验八 数组程序设计（一）	63
实验九 数组程序设计（二）	68
实验十 过程程序设计	73
实验十一 菜单和工具栏程序设计	79
实验十二 文件系统	85
实验十三 图形与绘图操作	91
实验十四 数据库应用技术	96
实验十五 综合实验	103

第三部分 练习题 118

第四部分 综合性设计实验 153

实验一 小学生四则运算练习系统	153
实验二 贪食蛇游戏	187

附 录 全国计算机等级考试二级VB考试大纲 197

第一部分 实验基础

第一章 Visual Basic 集成开发环境

Visual Basic (简称 VB) 是美国微软公司 (Microsoft) 推出的基于 Windows 操作系统环境下的软件开发工具, 是一种功能强大的高级程序设计语言。Visual Basic 是从 Basic 语言发展而来的, 20 世纪 70 年代后期, Microsoft 公司在 PC 机上开发出基于 DOS 操作系统的第一代 Basic 语言。Basic (Beginners All-Purpose Symbolic Instruction Code) 是最易于初学者学习使用, 在计算机程序设计发展历史上应用的最为广泛的语言。Visual 指的是开发 Windows 环境下图形用户界面 (GUI) 的方法, 即可视化的用户界面设计功能。Visual Basic 综合运用了 Basic 语言的结构化特点和 GUI 可视化程序设计的所见即所得功能, 因此 Visual Basic 既继承了 Basic 语言编程的简便性, 又具有 Windows 丰富的图形窗口工作环境。

1.1 Visual Basic 6.0 简介

Visual Basic 是一种可视化的、面向对象的 (Object Oriented Programming, OOP)、采用事件驱动对象对事件进行处理的程序设计语言。OOP 把编写程序解决具体问题的过程即一个应用程序的设计, 逐步划分成相互关联的多个对象, 并且建立起与这些对象相互关联的事件过程。Visual Basic 提供了可视化程序设计环境, 为用户准备了窗体对象及大量的在 Visual Basic 中被称为控件对象的界面元素, 如“菜单”、“命令按钮”、“复选框”、“文本框”等, 用户只需要利用鼠标、键盘把这些控件对象拖到窗体中适当的位置, 设置它们的大小、形状、颜色等特性, 就可以设计出应用程序的界面。

Windows 以其支持多任务、图形用户界面、对象链接与嵌入 (OLE)、动态数据交换 (DDE) 等强大功能, 成为目前微型计算机操作系统的主流。因此也产生了基于 Windows 环境下的多种软件开发工具, 如 Visual C++、Visual Basic、Delphi、Power Builder、Borland C++ 等。而 Visual Basic 是以上开发语言中最简单、最容易使用的, 其功能很强, 开发效率较高。Visual Basic 可以实现 Windows 应用程序设计的绝大部分高级功能, 主要包括以下几个方面。

(1) 支持多任务、多文档界面 (MDI) 设计。

(2) 支持对象链接与嵌入 (OLE), OLE 技术使得 Visual Basic 应用程序能够访问 Windows 环境中的其他应用程序, 将其他应用程序的文档链接或嵌入到 Visual Basic 应用程序中。如链接或嵌入 Word 文档, 当应用程序需要对 Word 文档进行操作时, 可直接启动 Microsoft Word 应用程序编辑该 Word 文档。有 OLE 技术的支持, 可以开发出集文字、声音、动画、图像、影像于一体的多媒体应用程序。

(3) Visual Basic 提供的动态数据交换 (DDE) 编程技术, 可以实现 Visual Basic 应用程序与其他 Windows 应用程序动态实时的交换数据。

(4) Visual Basic 支持的动态链接库 (DLL) 编程技术, 使得应用程序可以调用 Windows 操作系统提供的应用程序接口 (API) 函数资源, 或将用其他语言编写的程序加入到 Visual

Basic 应用程序中,提高了编程效率。

(5) Visual Basic 具有很强的数据库管理功能。数据库访问特性允许应用程序访问包括 Microsoft SQL Server 和大部分其他格式的企业数据库,建立各种数据库应用程序。

(6) 支持基于因特网的应用程序的开发。

(7) 应用程序设计完成后,可被编译生成在 Windows 环境下独立运行的可执行 .EXE 程序文件。

总之,从开发简单的 Windows 界面应用程序到大企业的应用系统,以及基于因特网的分布式应用程序,都可以通过 Visual Basic 实现。

1.2 Visual Basic 6.0 的启动及基本界面

Visual Basic 是一种高级编程语言,为用户提供了集成开发环境。所谓集成开发环境是指将程序的设计、编辑修改、运行、调试查错等各种功能一体化的开发环境。

单击“开始”按钮,在弹出的菜单中单击“程序”,将鼠标指针移向新弹出菜单中的“Microsoft Visual Basic 6.0 中文版”,在弹出的菜单中,单击“Microsoft Visual Basic 6.0 中文

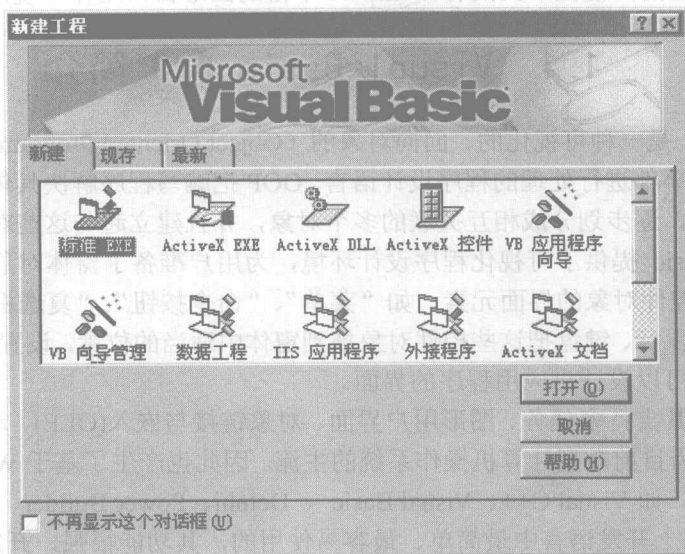


图 1-1-1 “新建工程”对话框

版”,启动 Visual Basic 6.0 并出现如图 1-1-1 所示的“新建工程”对话框,在该对话框中列出了 Visual Basic 6.0 能够创建的应用程序的类型。

使用 Visual Basic 开发的应用程序称为“工程”。选择“新建”标签选项,可以建立一个新的工程;单击“现存”选项,可以打开原来已有的工程文件,可以继续编辑、修改、调试;选择“最新”选项,可以在列表中选择打开最近编制的工程文件。如果不希望 Visual Basic 6.0 每次启动时都出现“新建工程”对话框,可以选择该对话框下方的“不再显示这个对话框”复选框,以后启动 Visual Basic 时就不再显示这个对话框,默认创建一个“标准.EXE”工程。

选择“新建”标签选项，选中“标准.EXE”，然后单击“打开”按钮，出现如图 1-1-2 所示的 Visual Basic 6.0 集成开发环境窗口。其中除包括标题栏、菜单栏、工具栏外，还有工具箱窗口、窗体设计器窗口、对象窗口、工程管理器窗口、属性窗口、窗体布局窗口等部分。另外，还有几个隐藏的窗口，如代码窗口、立即窗口、监视窗口等。

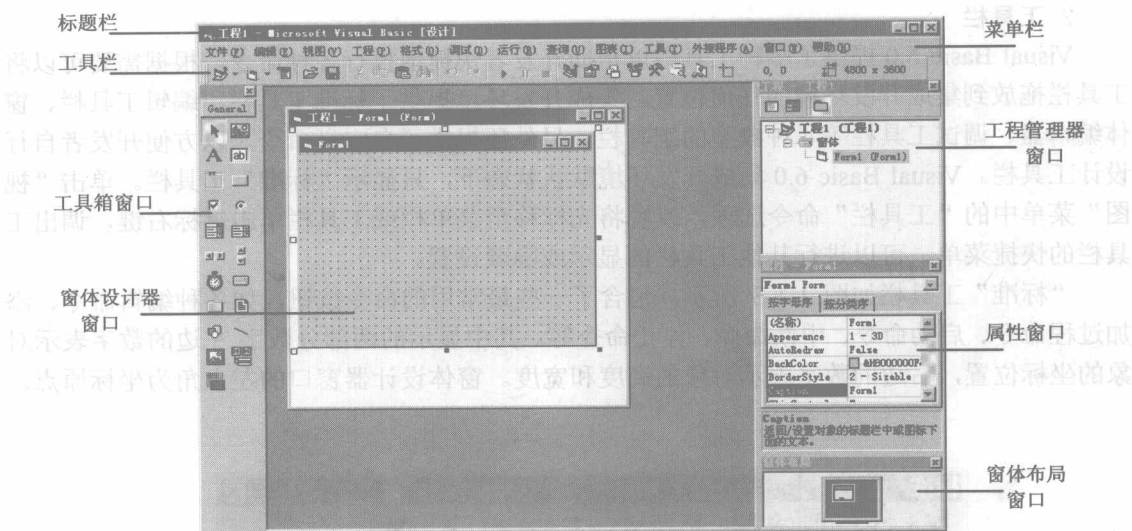


图 1-1-2 Visual Basic 6.0 集成开发环境界面

1. 菜单栏

显示 Visual Basic 命令，有多个菜单选项，每一个选项都有一个下拉式菜单。

(1) 文件菜单：包含了文件操作的有关命令，主要有新建、打开、添加、移除、保存工程，打印窗体及程序代码，将工程编译成在 Windows 环境下直接运行的可执行文件，以及退出 Visual Basic 集成开发环境等功能。

(2) 编辑菜单：包含了各种编辑命令，完全符合 Windows 的编辑操作风格，如复制、粘贴、剪切、删除等。该下拉菜单较长，把光标移到最下端的箭头处，菜单自动向上滚动，显示出后面的编辑菜单项。

(3) 视图菜单：主要控制 Visual Basic 集成开发环境中的各种窗口的显示，如对象窗口、代码窗口、立即窗口、工程资源管理器窗口、属性窗口、工具箱窗口等。可以显示或隐藏工具栏，激活调色板选项，以及改变窗体或控件的颜色。

(4) 工程菜单：主要包括 Visual Basic 工程管理的一些命令，如添加窗体、控件、模块等。

(5) 格式菜单：主要设置界面的格式，调整控件的布局。包括对齐方式、统一尺寸，改变对象之间的水平、垂直间隔，锁定对象的位置等。

(6) 调试菜单：为开发人员提供调试程序代码的各种方法，如添加监视、切换断点，以逐语句、逐过程方式运行程序等。

(7) 运行菜单：主要包括启动运行程序，中断和中止应用程序运行等命令。

(8) 工具菜单：通过其中的“选项”命令，对 Visual Basic 集成开发环境进行设置，它

还包括向程序添加过程等命令。

(9) 外接程序菜单：管理外接程序，增强 Visual Basic 的开发功能。

(10) 帮助菜单：Visual Basic 为开发人员提供了关于集成开发环境使用方面的详细帮助和参考资料。通过帮助菜单，选择相应的菜单项进入帮助，然后能以多种方式浏览帮助。除使用帮助菜单获得帮助外，也可以在需要帮助时按 F1 键，显示与当前状态有关的帮助内容。

2. 工具栏

Visual Basic 6.0 提供了多种工具栏以帮助开发者快捷地执行各种命令，根据需要可以将工具栏拖放到集成开发环境的任何位置。集成开发环境提供了标准工具栏、编辑工具栏、窗体编辑器、调试工具栏等 4 种典型的工具栏。另外还提供了自定义命令，以方便开发者自行设计工具栏。Visual Basic 6.0 集成开发环境默认状态下，只显示“标准”工具栏。单击“视图”菜单中的“工具栏”命令选项，或者将光标移至菜单栏或工具栏单击鼠标右键，调出工具栏的快捷菜单，可以进行其他工具栏的显示或隐藏设置。

“标准”工具栏如图 1-1-3 所示，包含了一些最常用的命令按钮，如各种编辑命令、添加过程命令、启动命令、中断命令、停止命令等。其中显示的两部分数字，左边的数字表示对象的坐标位置，右边的数字表示对象的高度和宽度。窗体设计器窗口的左上角为坐标原点。

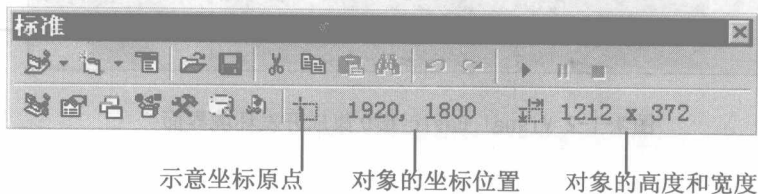


图 1-1-3 “标准”工具栏

3. 工具箱

工具箱窗口位于集成开发环境的左边，如图 1-1-4 所示，里面包含的是用来构造应用程序界面的各种控件。控件是组成程序与用户交互界面的基本元素，如命令按钮、文本框、单选按钮、复选框等。工具箱上不同的图标代表不同的控件类型，把鼠标指针移至一个控件图标上，将自动显示出相应的控件类型名。工具箱左上角的箭头不代表控件，向窗体中放置控件过程中，单击它可以将鼠标指针恢复到正常状态。

编制复杂程序时，需要其他未显示在工具箱中的控件。要在工具箱中增添新控件，可以通过下面的方法实现。将鼠标指针移至工具箱上右击鼠标，在弹出的快捷菜单中，单击“部件”选项，显示如图 1-1-5 所示的对话框，在列表中显示出系统提供的控件集合列表，选中要增添的控件集合左侧的复选框，单击“确

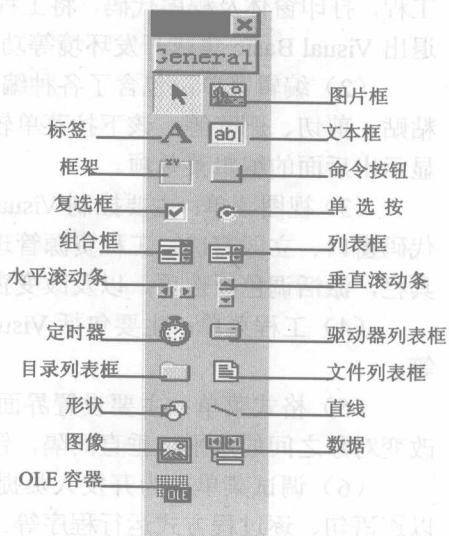


图 1-1-4 工具箱中的控件

定”按钮，新的控件将出现在工具箱窗口中。

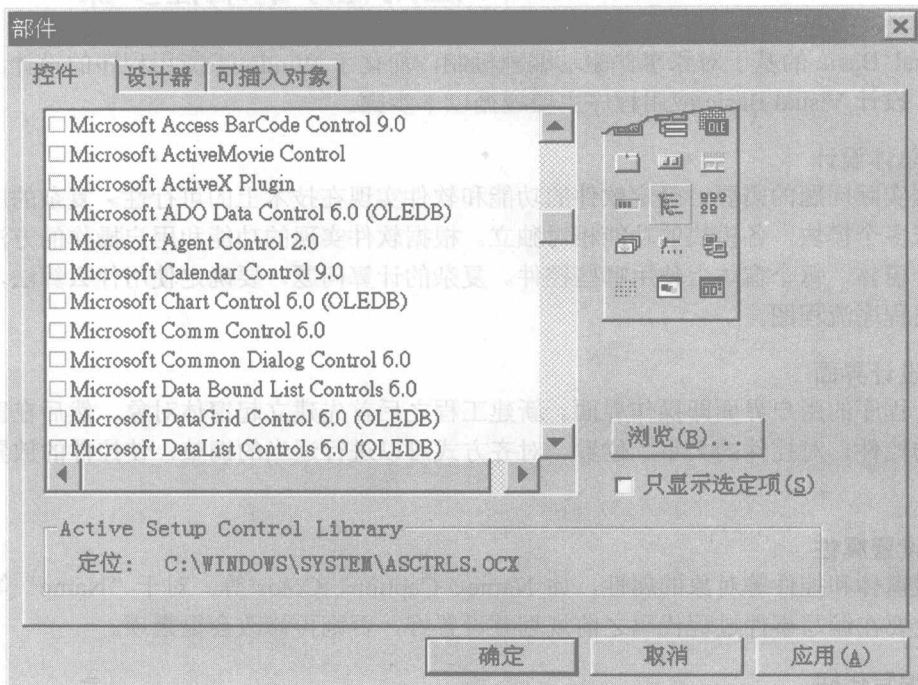


图 1-1-5 “部件”对话框

4. 工程资源管理器窗口

工程资源管理器窗口简称为工程管理器窗口，它以树型结构方式列出了工程中的窗体、模块和相应的模块文件名。用鼠标双击工程中的列表项，可以转换到相应对象中去。

在工程管理器窗口的上方有“查看代码”、“查看对象”和“切换文件夹”按钮。单击“查看代码”按钮，可显示代码窗口，在代码窗口中可以显示和编写程序代码；选定工程管理器窗口中的列表项，单击“查看对象”按钮，相当于双击工程窗口中的列表项，用来查看选中的对象；单击“切换文件夹”决定工程中的列表项是否以树型结构的形式显示，即是否将文件夹略去，仅显示包含在文件夹中的各个文件。

5. 属性窗口

属性窗口列出了选定窗体和控件的属性和属性值。属性是指对象的特征，如大小、标题、名称、颜色等。

6. 窗体布局窗口

通过窗体布局窗口可以布置应用程序中各个窗体在屏幕上的位置。窗口“显示器”的“屏幕”上显示有缩小的窗体，可以通过拖动鼠标移动窗体，来设置窗体在“屏幕”中的位置，这种设置将影响程序运行时窗体在屏幕上的初始位置。窗体布局窗口增强了 Visual Basic 的可视化功能。

7. 代码窗口

它是用来编写程序代码的窗口，是一个代码编辑器。应用程序中的每一个窗体或代码模块都有一个单独的代码编辑窗口。

1.3 Visual Basic 应用程序的编程步骤

Visual Basic 的基于对象事件驱动编程机制, 简化了 Windows 应用程序的设计。在这种机制下, 设计 Visual Basic 应用程序主要遵循以下步骤。

1. 总体设计

根据实际问题的需要, 确定软件的功能和软件实现技术上的可行性。复杂的应用程序应分解成多个模块, 各模块的功能相对独立。根据软件实现的功能和用户操作的方法, 确定使用几个窗体, 每个窗体上使用哪些控件。复杂的计算问题, 要确定使用什么算法, 必要时首先画出程序流程图。

2. 设计界面

设计程序的用户界面即操作界面。新建工程之后首先建立起窗体对象, 然后在窗体上放置需要的控件。对控件的大小、位置、对齐方式可以进行适当的调整, 使控件的放置尽量美观。

3. 设置属性

设置窗体和控件等对象的属性, 如 Name、Caption、Color 等。对于“Name”等重要属性, 一定要在编写事件过程代码之前规划并设置好, 否则再修改会很繁琐。

4. 编写代码

确定用到的事件, 并给事件编写相应的事件过程代码, 主要靠编写代码实现程序的功能。在编写的过程中, 要不断地运行、调试和修改错误。

设计界面、设置属性使用 Visual Basic 集成开发环境中的可视化编程工具即可完成。以上步骤中也可以在创建窗体和控件对象的同时, 设置对象的属性并编写该对象的事件过程代码。

1.4 程序的运行

Visual Basic 集成开发环境为应用程序开发提供了三种工作模式: 设计、运行、中断。通常将向窗体上添加控件, 设置窗体和控件的属性值, 以及编写程序代码看成是程序的设计阶段, 该阶段集成开发环境窗口标题栏上标题内容中显示有“[设计]”字样。应用程序设计完成或设计到一定程度, 通过运行验证程序是否正确, 通常要经过反复的调试和修改, 才能使程序最终实现其功能。

程序的运行可以采用两种方式: 一是解释方式, 二是编译方式。一般调试程序过程中采用解释方式, 简单方便, 但运行程序的时间较长, 占用的内存容量大, 效率低。程序调试通过后, 正式投入使用, 采用编译方式执行程序。

1. 解释方式

在解释方式下运行程序时, 不生成应用程序的目标程序。解释系统对源程序翻译一句, 执行一句, 即边解释边执行。具体讲 Visual Basic 应用程序的解释方式运行, 是指不脱离 Visual Basic 集成开发环境, 由集成开发环境负责对应用程序的解释执行, 并提供调试程序的方法。程序运行时, 当遇到设计错误, 执行到相关的语句时程序会暂停, 进入“中断”状态。

也可以单击工具栏中的“中断”按钮  或单击“运行”菜单中的“中断”命令，或者同时按下 **Ctrl+Break** 键进入“中断”状态，这时集成开发环境窗口标题栏上的标题内容显示有“[break]”字样。中断状态下便于对程序进行检查调试。

2. 编译方式

编译方式运行程序是指使用编译程序首先把应用程序全部翻译成机器指令表示的目标程序，并链接成可执行的程序文件，该程序在 Windows 环境下直接运行的方式。

Visual Basic 集成开发环境具有把应用程序编译成可执行文件(.EXE)的功能，这样程序就可以在 Windows 操作系统环境下直接运行，脱离 Visual Basic 集成开发环境生成可执行文件的方法是：选择“文件”菜单中“生成工程 1.EXE”命令，出现如图 1-1-6 所示的“生成工程”对话框，确定程序保存的位置和文件名，单击“确定”按钮则可生成 .EXE 程序文件。

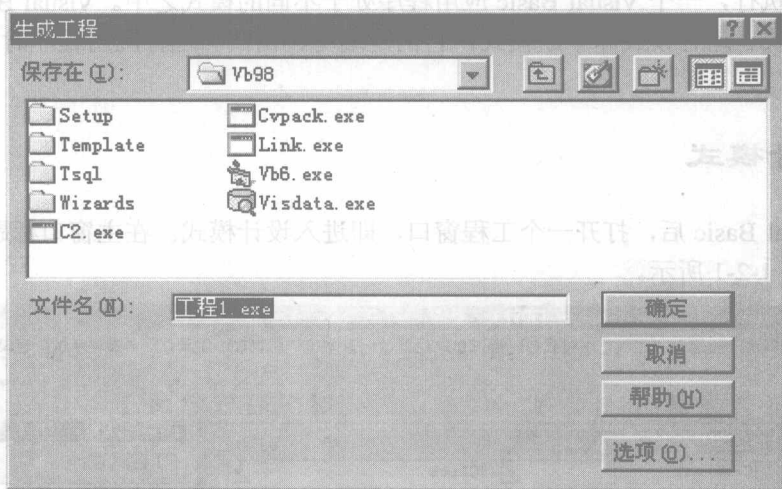


图 1-1-6 “生成工程”对话框

窗口标题栏 1-2-1 图

第二章 程序调试与错误处理

无论用户如何仔细地编写代码，都可能会出现这样那样的错误，而且随着应用程序代码量的增加，出现错误的概率也将成倍地增长。因此，用户在编写较大一些程序时，应该了解如何查找程序的出错。本章介绍的就是如何处理应用程序中可能出现的各种错误，同时介绍如何使用 VB 中的调试工具查找逻辑错误，使程序可以适应各种复杂的情况。

2.1 Visual Basic 的工作模式

从设计到执行，一个 Visual Basic 应用程序处于不同的模式之中。Visual Basic 有三种模式，即设计模式、运行模式和中断模式，它们是编写、调试和运行应用程序时应用程序所处的不同状态。

2.1.1 设计模式

启动 Visual Basic 后，打开一个工程窗口，即进入设计模式。在主窗口标题栏上显示[设计]字样，如图 1-2-1 所示。

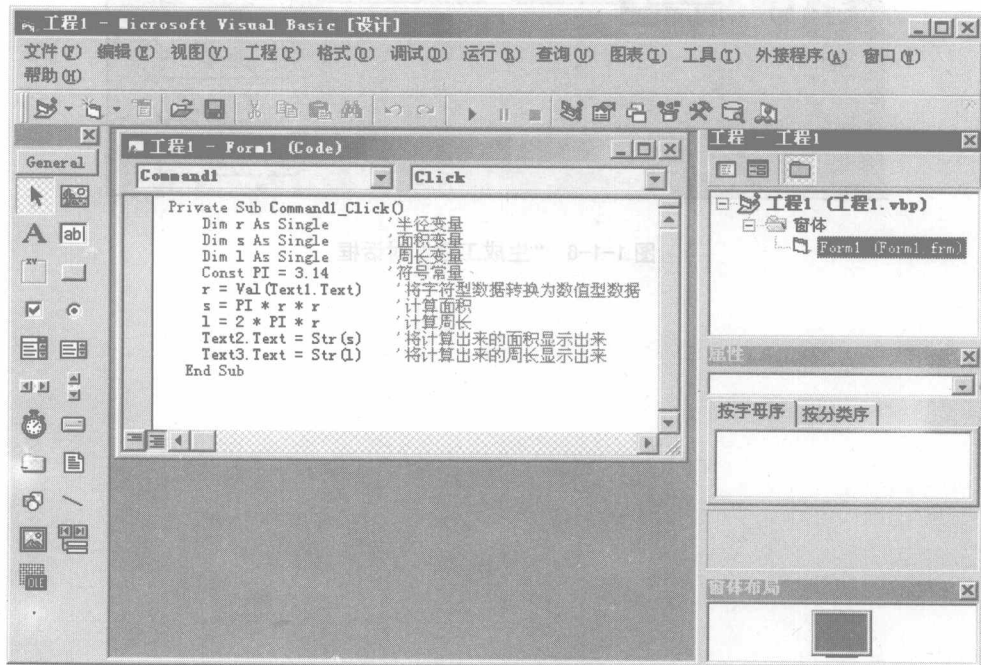


图 1-2-1 设计模式窗口

创建一个应用程序的所有工作都是在设计模式下完成。在设计时，可以设计窗体，绘制控件，编写程序代码，并使用“属性”窗口设置属性值或查看当前属性值等。当程序处于设

计模式时,不能执行操作,也不能使用调试工具对之进行调试,但可以指定断点和创建监视表达式。

2.1.2 执行模式

选择“运行”菜单中的“启动”命令(或者按【F5】键或单击工具栏中的“运行”按钮)。应用模式即可进入执行模式,此时主窗口标题栏上原来显示的“设计”被“运行”取代。

在执行模式中,Visual Basic 将全部控制权交给应用程序,用户可以对应用程序进行测试,可以查看程序代码,检验程序运行的结果,但不能修改代码。

选择“运行”菜单中的“结束”命令(或者单击工具栏中的“结束”按钮),可返回设计模式。

选择“运行”菜单中的“中断”命令(或者单击工具栏中的“中断”按钮或按【Ctrl+Break】组合键),可进入中断模式。

2.1.3 中断模式

当程序处于中断模式时,暂停程序的执行。进入中断模式后,主窗口的标题栏中原来显示的“设计”或“运行”由“Break”取代。在中断模式下,因为变量和属性设置值被保留下来,所以,可以分析应用程序的当前状态并输入修改内容,这些修改将影响程序的运行。此时,可以在应用程序中修改代码,观察应用程序界面的情况,确定哪个过程已被调用,监视变量值、属性和语句,改变变量值和属性设置值,查看或控制应用程序下一步运行的语句,立即运行 Visual Basic 语句,手工控制应用程序的操作。

可以用以下 4 种方式进入中断模式:

- (1) 在执行模式下,选择“运行”|“中断”命令。
- (2) 在程序中设置断电,程序执行到该断点时自动进入中断模式。
- (3) 执行程序时遇到了 Stop 语句。
- (4) 在程序执行过程中,如果出现错误,则将自动进入中断模式。

2.2 错误类型

用户在编写程序时,难免会出现一些语法和逻辑等错误,这些错误称为 Bug。找出并纠正这些错误的过程称为 Debug(调试)。

Visual Basic 应用程序的错误一般有语法错误、编译错误、运行错误和逻辑错误 4 类。

2.2.1 语法错误

语法错误是由于不正确地创建代码,即在语法不正确时出现这种错误。如错误地输入了关键字、丢失或写错了符号、遗漏了必须的语句成分、括号不匹配等。

Visual Basic 具有自动语法查错功能,在设计阶段输入程序代码时就能检查出语法错误。如以下代码:

```
Private Sub Command2_Click()
```

```
a = 10: n = 20
```

```
c = a + b
```

```
Print c
```

```
End Sub
```

在输入代码时，若将第三行输入为：

```
c = a { b
```

按【Enter】键后就会显示出错提示框，并且刚输入的一行变为红色，出错的部分高亮显示，如图 1-2-2 所示。

只要改正该语法错误后，红色和高亮显示才会消失。

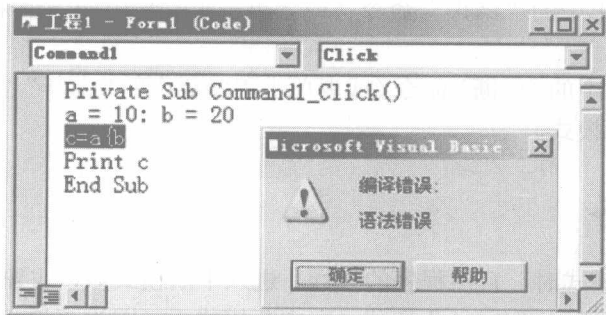


图 1-2-2 语法错误提示

2.2.2 编译错误

Visual Basic 在运行程序前，先编译执行程序。如果用户未定义变量或遗漏关键字等，则 Visual Basic 将出现出错提示，并使有错误的程序行高亮显示，如图 1-2-3 所示，这种错误称为编译错误。出现这类错误后，Visual Basic 将停止编译，并返回有错误程序代码的窗口。

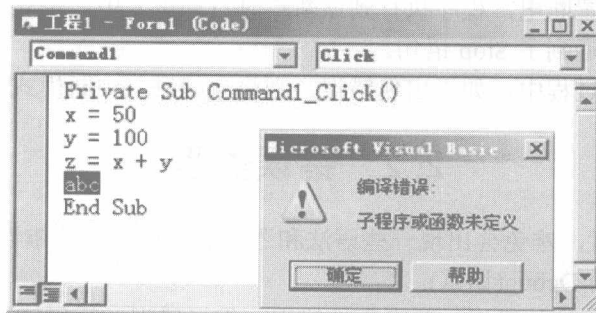


图 1-2-3 编译错误提示

2.2.3 运行错误

若语法正确，并且程序运行期间，一个语句试图执行一个不能执行的操作，就会产生运行错误。有运行错误的代码在一般情况下运行正常，但遇到非法数据或是系统条件禁止代码运行时就会产生错误。一个典型的运行错误是以 0 作为除数，运行时将弹出错误对话框，如图 1-2-4 所示。