



中等职业教育规划教材
中等职业教育规划教材审定委员会审定

可视化编程应用 —— Visual Basic

主编 安小敏



国防科技大学出版社



中等职业教育规划教材

中等职业教育规划教材审定委员会审定

可视化编程应用 ——Visual Basic

主 编：安小敏

副主编：黄子华

编 委：张 越

仲 劲

吴文庆

张建

李秀荣

林 波

史东

李

江苏工业学院图书馆
藏书章

国防科技大学出版社

内 容 提 要

本书是根据“可视化编程应用教学基本要求”编写的教材,以程序设计为主线,由浅入深,循序渐进地组织教材内容,结构合理、例题丰富、通俗易懂。对学生可能遇到的难点作了清楚、详细的阐述,并且在每章开头指出了学习目标、每章结尾配有习题。内容分别为:可视化编程简介、Visual Basic 的开发环境、窗体与控件、菜单设计、Visual Basic 程序设计基础、Visual Basic 中的过程、文件系统、数据库应用程序开发、Visual Basic 的其他应用、管理信息和系统的开发。

本书是中等职业学校(三、四年制)计算机类专业通用教材,也可作为中专、技校、职业技术学院的计算机专业教材,还可供职业培训和计算机用户自学使用。

图书在版编目(CIP)数据

可视化编程应用: Visual Basic/安小敏主编, - 长沙:国防科技大学出版社,2007

中等职业教育规划教材

ISBN 978 - 7 - 81099 - 432 - 3

I. 可... II. 安... III. BASIC 语言 - 程序设计 - 专业学校 - 教材

IV. TP312

中国版本图书馆 CIP 数据核字(2007)第 139896 号

国防科技大学出版社出版发行

电话:(0731)4572640 邮政编码:410073

<http://www.gfkdcbs.com>

E-mail: faxing@gfkdcbs.com

责任编辑:陈靖

全国各新华书店经销

北京楠萍印刷有限责任公司印刷

*

开本:787×1092 1/16 印张:17 字数 400 千字

2007 年 11 月第 1 版第 2 次印刷 印数:1~3000 册

定价:24.00 元

章	节	学时
1	1.1	2
1	1.2	2
2	2.1	4
2	2.2	4
3	3.1	4
3	3.2	4
4	4.1	4
4	4.2	4
5	5.1	4
5	5.2	4
6	6.1	4
6	6.2	4
7	7.1	4
7	7.2	4
8	8.1	4
8	8.2	4
9	9.1	4
9	9.2	4
10	10.1	4
10	10.2	4

前 言

随着计算机的普及和计算机技术的飞速发展,计算机技术日臻完善。掌握计算机操作技能已经成为高素质劳动者的必备条件。

中等职业教育是我国职业教育的重要组成部分。为了贯彻落实国务院 2005 年 11 月发布的《关于大力发展职业教育的决定》的精神,推动中国中等职业教育又快又好地发展,我们特组织一些多年从事一线教育工作且具有丰富教学经验的优秀教师,以培养学生从事网络管理的基本知识和基本技能为准则编写,同时加入了新时代计算机使用频率高的功能及操作方法的介绍。在内容上做到了简单明了、通俗易懂、人机互动并与实际应用相结合。课程设置体现了较大的灵活性,为实际情况有别的各种职业学校开展程序设计课教学创造了良好的条件。

全书共分十章。内容分别为:可视化编程简介、Visual Basic 的开发环境、窗体与控件、菜单设计、Visual Basic 程序设计基础、Visual Basic 中的过程、文件系统、数据库应用程序开发、Visual Basic 的其他应用、管理信息系统的开发。编写时我们充分考虑到中职学生的认知特点与理解能力,在每章开始都有本章概述和学习目标,以便对学生学习本章知识进行引导;每章后附带有本章的综合习题,可供学生检查学习效果与自测使用。

本书在内容的选择和编排上,充分考虑了当前计算机发展的现状以及中等职业学校学生的实际需求,遵循了由浅入深、循序渐进的原则,以应用为目的,力求从实际出发,尽量减少枯燥死板的理论概念,坚持以理论为基础,重视实践操作与社会需求相结合是本书最大的特点。全书内容丰富、图文并茂、结构严谨、重点突出,既可以作为中等职业学校计算机类专业通用教材,也适用于计算机培训班的教学,对计算机专业和非计算机专业的读者均适用。

本书由安小敏担任主编,黄子华、张建忠担任副主编,参加编写的还有张越、李秀荣、史东蕾、仲劲、林波、李腾和吴文庆。

全书建议不低于 72 学时,学时安排建议如下(仅供参考):

(1) 基础模块学时安排建议

章节	课程内容	讲课	上机实习	合计
第一章	可视化编程简介	2	0	2
第二章	Visual Basic 开发环境	2	2	4
第三章	窗体与控件	4	4	8
第四章	菜单设计	2	6	8
第五章	Visual Basic 程序设计基础	4	4	8
第六章	Visual Basic 中的过程	2	4	6
第七章	文件系统	4	4	8
第八章	数据库应用程序开发	4	6	10
第九章	Visual Basic 的其他应用	2	4	6
第十章	管理信息系统的开发	2	6	8
	机动		4	4
合计		28	44	72

由于作者水平有限,加上时间比较仓促,书中难免有不妥之处,我们衷心地希望得到广大读者的批评指正,以使本书在教学实践中得到不断完善。

编者

2007年11月

目 录

501	0.3.2
501	0.3
501	0.3.1
501	0.3.2
501	0.3.3
501	0.3.4

60	1.2.3
60	1.2.4
70	1.2.5
75	1.2.6
75	1.2.7
75	1.2.8

第一章 可视化编程简介	1
1.1 可视化编程的概念.....	1
1.2 面向对象的程序设计.....	2
1.3 类、对象、属性及方法.....	3
1.4 事件及事件的过程.....	4
1.5 对象、属性、事件和方法之间的关系.....	5
本章习题.....	6
第二章 Visual Basic 的开发环境	8
2.1 Visual Basic 简介.....	8
2.1.1 Visual Basic 的发展过程.....	8
2.1.2 Visual Basic 版本简介.....	9
2.1.3 Visual Basic 的功能特点.....	9
2.2 Visual Basic 的安装及开发环境简介.....	10
2.2.1 Visual Basic 的运行.....	10
2.2.2 Visual Basic 的安装步骤.....	10
2.3 Visual Basic 的快速入门.....	19
2.3.1 Visual Basic 的工程.....	19
2.3.2 Visual Basic 程序设计的一般过程.....	21
2.3.3 环境设置.....	27
2.3.4 打包与发布程序.....	31
本章习题.....	36

第三章 窗体与控件	40
3.1 类、对象及面向对象的编程.....	40
3.2 窗体和控件的共性.....	41
3.2.1 对象的三要素.....	41
3.2.2 窗体及控件的常见属性.....	42
3.2.3 窗体入控件的常用事件.....	43
3.2.4 窗体、控件的基本操作.....	43
3.3 窗体.....	45
3.3.1 窗体(Form)的属性.....	46
3.3.2 窗体的事件.....	50
3.3.3 窗体的方法.....	52
3.3.4 多窗体操作.....	53
3.3.5 实例解析.....	54
3.4 一般类控件.....	56
3.4.1 标签.....	56
3.4.2 文本框.....	57
3.4.3 命令按钮.....	58
3.4.4 图片框与图像框.....	59
3.4.5 直线与形状.....	61
3.4.6 定时器与滚动条.....	63
3.4.7 实例解析.....	64
3.5 选择类控件.....	69

3.5.1 框架	69	5.2.6 逻辑运算符与逻辑表达式	102
3.5.2 单选按钮和复选框	69	5.3 标准函数	102
3.5.3 列表框	70	5.3.1 标准函数调用格式	102
3.5.4 组合框	72	5.3.2 数学函数	102
5.5.5 实例解析	73	5.3.3 转换函数	103
本章习题	75	5.3.4 格式函数	103
第四章 菜单设计	79	5.3.5 随机函数	103
4.1 菜单的规化与设计	79	5.3.6 字符串操作函数	104
4.1.1 菜单的规划	79	5.3.7 日期和时间函数	104
4.1.2 菜单的设计	80	5.4 程序流程控制	105
4.2 弹出式菜单和下接式菜单的创建	82	5.4.1 选择结构程序	105
4.2.1 下拉式菜单	82	5.4.2 循环结构	112
4.2.2 弹出式菜单	85	本章习题	117
4.3 多文档界面 MDI	87	第六章 Visual Basic 中的过程	119
4.3.1 MDI 初步设计	87	6.1 过程概述	119
4.3.2 动态增加子窗体	90	6.2 过程的定义与调用	119
4.3.3 卸载、删除子窗体	91	6.2.1 Sub 过程	119
本章习题	91	6.2.2 事件过程	120
第五章 Visual Basic 程序设计基础	93	6.2.3 通用 Sub 过程	120
5.1 模块与代码窗口	93	6.2.4 调用 Sub 过程	121
5.1.1 模块	93	6.2.5 Function 过程	122
5.1.2 代码窗口	94	6.2.6 过程的创建	124
5.2 Visual Basic 中的数据与运算	95	6.3 过程中的参数传递	125
5.2.1 数据类型	95	6.3.1 形参与实参	126
5.2.2 常量	97	6.3.2 参数的传递	126
5.2.3 变量	98	6.4 变量过程的作用域	127
5.2.4 运算符与表达式	99	6.4.1 变量的作用域	127
5.2.5 关系运算符与关系表达式	101	6.4.2 过程的作用域	129

6.5 程序的调试与排错	130	8.3.1 数据控件	170
6.5.1 调试方法与工具	130	8.3.2 Recordset 对象的属性与方法	172
6.5.2 程序中的排错	130	8.3.3 数据绑定控件	174
本章习题	131	8.3.4 应用示例	174
第七章 文件系统	134	8.4 结构化查询语言 SQL	179
7.1 文件及访问方式	134	8.4.1 数据查询	180
7.1.1 顺序文件	135	8.4.2 数据操作	183
7.1.2 随机文件	143	8.4.3 数据定义	183
7.1.3 二进制文件	144	8.4.4 用可视化数据管理器建立查询	184
7.2 文件管理控件	146	8.5 使用 ADO 控件访问数据库	187
7.2.1 驱动器列表框	146	8.5.1 ADO 对象模型	187
7.2.2 目录列表框	147	8.5.2 Adodc 控件	188
7.2.3 文件列表框	148	8.5.3 应用示例	189
7.3 文件的其他操作	146	8.6 数据环境与数据报表	193
7.3.1 文件的删除	149	8.6.1 数据环境设计器	193
7.3.2 复制和移动文件	150	8.6.2 数据报表设计器	194
7.3.3 文件的更名	151	8.6.3 应用示例	197
7.3.4 运行可执行文件	151	本章习题	203
本章习题	156	* 第九章 Visual Basic 的其他应用	205
第八章 数据库应用程序开发	159	9.1 多媒体控件的应用	205
8.1 数据库基础	159	9.1.1 视频播放器示例	205
8.2 可视化数据管理器	162	9.1.2 MMControl 控件的常用属性与事件	206
8.2.1 新建数据库	162	9.1.3 音频播放器示例	209
8.2.2 打开数据库	163	9.1.4 综合媒体播放器示例	211
8.2.3 添加表	163	9.2 网络应用	214
8.2.4 数据增加、删除、修改	165	9.2.1 网络通信协议的基础和选择	215
8.2.5 数据窗体设计器	169		
8.3 数据控件 Data 和数据绑定控件	170		

170	9.2.2 WinSock 控件的主要属性、方法与事件	215	10.2 开发实例	225
175	215	10.3 应用程序的编译与发布	261
174	9.2.3 网络通信程序示例	217	10.3.1 生成.exe 文件	261
174	本章习题	222	10.3.2 发布应用程序	261
	第十章 管理信息系统的开发	224	本章习题	262
80	10.1 管理信息系统的开发	224		
183	8.4.2 数据库操作			
183	8.4.3 数据库交互			
184	8.4.4 数据库交互的数据库驱动程序			
187	8.2 使用ADO 进行数据库操作			
187	8.2.1 ADO 对象模型			
188	8.2.2 ADO 控件			
189	8.2.3 数据库连接			
193	8.6 数据库连接与数据库操作			
193	8.6.1 数据库连接与数据库操作			
194	8.6.2 数据库连接与数据库操作			
197	8.6.3 数据库连接			
203	本章习题			
202	*第九章 Visual Basic 的其他应用	202		
202	9.1 多媒体控件的应用	202		
202	9.1.1 多媒体控件应用示例	202		
206	9.1.2 MMControl 控件的常用属性和事件	206		
209	9.1.3 动画播放控件	209		
211	9.1.4 核心控件播放与录制	211		
214	9.2 网络应用	214		
212	9.2.1 网络通信控件的常用属性和事件	212		
150	8.3 数据库与 Data 和数据库控件	150		
192	8.2.4 数据库与数据库控件	192		
193	8.2.3 数据库	193		
193	8.2.2 数据库	193		
193	8.2.1 数据库	193		
193	8.2 数据库与数据库	193		
193	8.1 数据库与数据库	193		
130	第八章 数据库应用程序开发	130		
136	本章习题	136		
131	7.3.4 运行可执行文件	131		
131	7.3.3 文件操作	131		
130	7.3.2 复制与移动文件	130		
129	7.3.1 文件的删除	129		
140	7.3 文本的编辑	140		
148	7.2.3 文本的编辑	148		
147	7.2.2 目录的编辑	147		
140	7.2.1 目录的编辑	140		
130	7.2 文件的操作	130		
132	7.1.1 操作文件	132		
131	7.1 操作文件	131		

第一章

可视化编程简介

本章概述：

本章主要介绍可视化编程的概念及其开发,面向对象程序设计的基本概念,Visual Basic 中的类、对象、属性、方法、事件、事件的过程等基本概念。

学习目标：

- 了解可视化编程的基本概念
- 了解面向对象程序设计与传统程序设计的异同
- 理解可视化编程种类、对象、属性、方法、事件及事件的过程等基本概念

1.1 可视化编程的概念

当我们采用传统的编制程序代码的方式来设计用户图形界面时,不仅需要大量的程序代码对界面进行描述,而且在程序设计过程中看不到实际效果,这就不可避免地造成许多错误,因此需要反复编译、运行、修改、调试,观察运行结果,直到满意为止。应用软件为了适应 Windows 界面环境,使用户界面的开发工作变得越来越复杂,相关代码所占比例也越来越大,因此在 Microsoft 公司推出 Visual Basic 以后,编程人员不再受 Windows 编程的困扰,能够所见即所得地设计标准的 Windows 界面,可视化编程技术受到极大的欢迎。可视化程序设计使用可视化设计工具,使编程人员能直接看到设计制作的图形界面,如一个菜单、一个对话框。而用户需要编写的只是实现程序计算、逻辑判断等程序代码。

可视化(Visual)编程技术把 Windows 界面设计的复杂性“封装”起来,使得编程人员在编写应用程序时,既不需要编写大量的代码来描述组成用户界面的各种对象的外观和位置,也不需要反复地调试运行来调整它们,而只要把由开发工具提供的各种对象拖放到屏幕上并直接调整位置、大小、颜色、字体等即可。程序运行时所见到的结果就是设计时调整的结果,也就是常说的“所见即所得(WYSIWYG—What You See Is What You Get)”。该项技术具有编程简单、自动生成程序代码、效率高的优点,因而在当今的编程语言中被广泛采用。可视化编程即是“所见即所得”技术的重要应用之一。

可视化编程一般是基于事件驱动的原理。使用过办公软件的人必定深有体会,若用鼠标单

击工具栏上某个按钮,就会完成相应的操作。比如,单击“保存”按钮,就会将文件保存,单击一下“另存为”菜单项,就会弹出“另存为”对话框。这是由于这些对象(按钮或菜单项)触发了一个事件来执行相应的过程,从而完成某种特定的功能。

1.2 面向对象的程序设计

传统的过程式程序设计随着软件危机和应用系统的不断膨胀显得越来越力不从心。随着20世纪70年代面向对象的编程语言的出现,以及C++的发展成熟,面向对象程序设计思想得到广泛的认同和普及。至20世纪90年代,各种程序语言和工具都引入了这一思想。其优越性是有目共睹的,因此成为这一时代软件产业的主体技术。

面向对象程序设计 OOP (Object - Oriented Programming) 是一种以对象为基础,以事件驱动程序执行的程序设计技术。OOP 将一个应用程序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关联的事件过程,通过对象对所发生的事件产生响应来执行相应的事件过程,引发对象状态的改变,从而达到运算、处理的目的。程序员在应用程序中只需说明对象应完成的任务,该任务通常通过编程来完成,采用结构化程序设计的方法。

面向对象程序设计起源于 Smalltalk 语言,它引入了新的概念和思维方式,为使软件在程序设计阶段能够模仿真实世界建立模型,它对系统的复杂性进行了概括、抽象和分类,使软件的设计与实现形成了一个由抽象到具体、由简单到复杂的循序渐进的过程,从而解决了大型软件研制中存在的效率低、质量难以保证、调试复杂、维护困难等一系列问题。因此,近年来面向对象的程序设计得到了广泛应用。目前在 Windows 环境下常用的面向对象程序设计开发工具有 Visual Basic、Visual J++、Visual C++、Visual FoxPro、PowerBuilder、Delphi 等。面向对象程序设计的优点如下:

- (1) 符合人们习惯的思维方式,便于分析复杂而多变的问题。
- (2) 易于维护软件和增减功能。
- (3) 采用继承的方式缩短程序开发所花费的时间。
- (4) 与可视化技术相结合,改善了工作界面。

OOP 的这些特点使面向对象程序设计与面向过程程序设计有很大的不同,具体体现在以下这些方面:

- (1) 设计程序不采用顺序性的结构,而是采用对象本身的属性与方法来解决问题。
- (2) 在解决问题的过程中,可以直接在对象中设计事件处理程序,而不用调用子过程,方便地实现了用户自由无顺序的操作。
- (3) 数据与程序不是分离的,数据是特定对象的数据,也只有对象的函数或过程才能对数据进行处理。一个对象中的函数或过程共享对象的数据,解决了因调用子过程而出现大量数据传递的情况(如函数返回值和较多参数)。
- (4) OOP 非常适合于 Windows 环境下的程序开发。它可以充分利用 Windows 的各种资源来构造应用程序,但需要程序员比较熟悉 Windows。

面向对象编程技术和可视化编程开发环境的结合,改变了应用软件只有经过专门技术训练的专业编程人员才能开发的状况。它使软件开发变得容易,从而扩大了软件开发队伍。由于大



量软件模块的重用和可视控件的引入,技术人员在掌握这些技术之后,就能有效地提高应用软件的开发效率,缩短开发周期,降低了开发成本,并且使应用软件界面风格统一,增加了易用性。

1.3 类、对象、属性及方法

在面向对象的程序设计中,包括了类、对象、属性、方法等基本概念。

1. 类(Class)

类是面向对象可视化编程中最基本的概念之一,是一组具有相同属性和相同操作的对象的集合。例如,可以把“书”看成一个“类”,它是由许多本具体的书抽象而来的一般概念。所有的书籍都具有类似的功能(供人阅读、学习)及相似的属性(书名、书号、作者、出版社、出版日期、价格等),但却又不是完全一样的。把“书籍”作为一个类,则具体某本书就是“书籍”这个类的一个实例、一个对象。所以说,类是一个集合,也是其实例的模板。

2. 对象(Object)

对象的概念来源于生活,它是面向对象理论的基石。面向对象是一种思维方式,按这种思维方式,任务事物都可以理解为对象。在现实生活中,我们随时随地都在和对象打交道,从天体太阳、月亮、地球,到城市上海、北京,再到生活中常见的物品报纸、衣服等都可以看作对象。对象是具体存在的,每个对象都是唯一的,有着自己独特的个性。类则是具有相同性质的对象的集合。对象和类的区别,在于前者是具体的,而后者则是抽象的。比如“人”可以被称为类,平常我们谈到人类时,绝不是指某一具体的人或某一群具体的人,而我们每一个具体的人,只能被看作是对象。

由此,可得出结论,类是对所有具有共同特征的事物的抽象思维,它是一个抽象的概念,相当于一个模具;而对象是一个类中某个具体实体,是一个具体存在的、客观的实体,或者说是类的一个实例。

在面向对象程序设计中,对象的概念就是对现实世界中对象的模型化,它是代码和数据的组合,同样具有自己的状态和行为。对象是既包含数据,又包含对数据进行操作的方法,并将其封装起来的一个逻辑实体。对象的状态(例如球的质地、颜色、大小)用数据来表示,称为对象的属性;而对象的行为(例如球的滚动、停止或旋转)用对象中的代码来实现,称为对象的方法。总之,任何对象都由状态(也称为属性、数据)和行为(也称为方法、操作)组成。因此也可以理解为对象是一些属性、方法和事件的集成。

3. 属性(Property)

不同的对象具有不同的特征。就像现实生活中的书、飞机、钢琴、计算机等等,千差万别,这是显而易见的。在面向对象的程序设计中,对象的特征用一些数据项来描述,这些数据项取不同值,对象便具有不同的特征,这些特征称为属性。有些属性项属于公共属性,比如标题属性、名称属性等;有些属性则是某一个对象所特有的。不同的属性能够控制对象的外观和行为,比如把一个人作为一个对象,那么姓名、身高、体重都是这个对象的属性。其中有些属性是只读并且无法改变的,而有些则可以通过设定来改变。这就好像录音机的出厂日期、型号等属性是无法改变的,但面板上所显示的时间则可通过设定来改变。

在可视化编程中,常见的属性有标题(Caption)、名称(Name)、背景色(Backcolor)、字体大小(Fontsize)、是否可见(Visible)等。通过修改或设置某些属性便能有效地控制对象的外观。每一种对象都有一组特定的属性。如文本框的属性有名称、文本内容、最大字符数、多行等。

属性值的设置或修改可以通过属性窗口来进行,也可以通过编程的方法在程序运行的时候来改变对象的属性。一般来说,每个对象的属性都有一组默认值。对大多数属性都可采用系统提供的默认值,当默认值不能满足要求时可以由用户自己来设置所需的属性值。对象的属性反映了对象的状态。在程序系统中,属性也就是对象所拥有的数据。

4. 方法(Method)

方法是指要进行的操作。如在 Visual Basic 中,方法是由系统提供的一种专门的子程序,用来完成一定的操作:显示对象、显示图像及移动、打印、绘画等。

方法是特定对象的一部分,正如属性和事件是对象的一部分一样。有些方法可能适用于多种甚至所有对象,而有些方法可能只适用于少数几种对象。例如,Visual Basic 中的 Print 就是一种“方法”,是用来输出信息的专用子程序。当把它用于不同的对象时,可以在不同的设备上输出信息。

例如对于“人”这个对象,可以有很多“方法”,如跑步、游泳等,方法是一个简单的、不必知道细节的、无法改变的事件。同样,方法也不是随意的,某些对象有一些特定的方法。

方法是用来处理操作对象的途径。对象通常会提供一些方法,以便应用程序可以使用对象所提供的服务。例如,录音机提供了“播放”、“停止”、“暂停”、“快进”、“快退”等操作按钮,而这些按钮其实就相当于录音机提供的方法。用户只要按下这些按钮,就可以得到录音机所提供的播放、停止、暂停播放、快进等服务。

同理,只要通过对象对外提供的方法,就可以得到它的服务,根本不需要知道对象内部的实际运作方式。所以,用面向对象的程序设计方法来开发应用软件,不仅可以提高效率,更重要的是可以保证软件的质量。因为,用户仅需知道怎样调用对象提供的服务(功能)就可以了,而不必从头开始设计和编写应用软件中需要的所有功能。

1.4 事件及事件的过程

面向对象程序设计语言是有事件驱动的,程序设计人员只需编写响应用户动作的程序,而不必考虑每个步骤按什么顺序执行。在这种机制下,不必编写一个大型程序,而是建立一个由若干个微小程序组成的应用程序,这些微小的应用程序都可以由用户操作来触发。这样就使编程工作变得比较简单。下面具体讲述几个相关概念。

1. 事件(Event)

如前所述,一本书是一个对象,那么我们可把翻开书视为一个事件。事件就是在对象上所发生的事情,是一种特定的操作,是指由系统事先设定的、能被对象识别和响应的动作。一个事件可以在键盘上按键、在窗体上单击鼠标、窗体打开或关闭或者向文本框中输入一个值。例如,在 Visual Basic 中,鼠标的单击(Click)、双击(Doubleclick)、窗体的装载(Load)等都是事件。通常情况下,事件可以由用户操作引起的,也可以是来自系统、其他应用程序或应用程序内部消息触发。



每一种对象能识别的事件是不同的。例如窗体能识别单击和双击事件,而命令按钮能识别单击事件却不能识别双击事件。但是,很多事件类型为大多数对象所共有。例如命令按钮和窗体都可以对单击(Click)事件做出响应。

每个对象都可以对被称为事件的动作进行识别和响应。事件可以由一个用户的动作产生,如单击鼠标或按下一个键,也可以由程序代码或系统产生,如计时器。在多数情况下,事件是通过用户的交互操作产生的。例如,对于一部电话来说,当用户拿起听筒时,便触发了一个事件;同样地,当用户拨号打电话时,也触发了若干事件。

2. 事件过程(Event Procedure)

当事件被触发时,对象就会对该事件作出响应。比如,单击工具栏上的“保存”按钮,计算机将会执行一系列相应的操作,这些操作就是对象对事件作出的响应。响应某个事件所执行的程序代码称为事件过程,也称为事件子程序。一个对象能识别一个或多个事件,因此可以使用一个或多个事件过程对用户或系统的事件作出响应。但在程序中究竟使用多少事件过程,则要求设计者根据程序的具体要求来确定。相同事件发生在不同对象上所得到的反应是不一样的,因为这些事件的事件过程是不同的。面向对象程序设计的主要工作,就是采用程序代码为对象编写事件过程。

采用面向对象程序设计语言编制的程序,包括若干个规模较小的事件子程序(事件过程),每一个事件过程又由一个相应的事件来驱动。程序运行时,会等待某个事件的发生(一般由用户操作来触发),然后为响应该事件去执行该事件的事件过程。当执行完某一事件过程后,程序又会等待下一事件的发生,直至运行了结束程序运行的事件而使程序结束。

可见,事件过程需要经过事件的触发才能被执行,程序的这种执行方式被称为事件驱动方式。由于各事件的发生顺序是任意的,所以,程序的执行并没有固定的顺序,而是由事件控制整个程序的执行流程。事件的发生是由用户操作来触发的,所以,程序的执行流程事实上是由用户来控制的。

事件方法与事件过程的不同之处在于:

- (1) 事件过程是对某个事件的响应,而方法不能响应某个事件。
- (2) 用户必须考虑响应事件的过程,但不必考虑方法的实现过程。
- (3) 用户一般编写事件过程的程序代码,但对方法的使用只能按照语言的约定直接调用它们。

3. 事件驱动

在面向对象的程序设计中,程序执行后系统等待某个事件的发生,然后去执行处理此事件的事件过程,待事件过程执行完后,系统又处于等待某事件发生的状态,这就是事件驱动程序设计方式。这些事件驱动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可能都是不同的。

1.5 对象、属性、事件和方法之间的关系

1. 对象、事件和方法三者之间的关系

在可视化程序设计中,完成某种功能可通过触发对象事件和调用对象方法来实现,而触发事

件和调用对象方法都必须基于某个对象。触发某个事件实际上就是执行一个事件过程的代码段,在该代码段中又可以随意调用对象方法,而调用对象方法不可能触发某个事件。

2. 属性和方法的区别与联系

对象的属性是描述对象的一组特征。如设置对象的名称、标题、颜色、大小等,有的可直接在设计界面时通过属性窗口进行设置,有的可以在事件过程或通用过程代码中进行设置。对象的方法是在对象上实施某种操作的功能,不能在设计界面时调用对象方法,一般在过程代码中可随意调用,根据需要还可以添加适当的参数。设置对象属性和调用对象方法均可在代码设计中进行。

我们可以将“学生”理解为对象,那么这个“学生”具有姓名、年龄、性别、穿衣等特征,这些特征就是这个学生的属性。此外,这个学生还有一些属于自己的行为,比如打架、乱投东西等,这些行为就可以称为“属于这个学生的方法”了。

那么什么是这个学生的事件呢?这个例子不太好列举(因为人毕竟不是 Visual Basic),比如某些同学受到强光的刺激会眨眼。这里的“强光”就是事件,而“眨眼”这个动作是由强光所触发的,这就叫事件驱动。

当然,上述实例基本上说明了面向对象思维方式的实质,即当我们看一个事物时,通常是把这个事物“打成一个包,把属于它的东西都包在一个包袱里”,这在理论上被称为“封装”,而对象则被称为属性和方法的封装体。

本章习题

一、填空题

1. 对象具有属性和_____。
2. 对象的属性是用_____来描述的。
3. 对象是既包含_____,又包含对_____的方法,并将其封装起来的一个逻辑实体。
4. 事件过程需要经过事件的触发才能被执行,程序的这种执行方式被称为_____。
5. 类是一组具有相同属性和相同操作的_____的集合。

二、选择题

1. 英文缩写 OOP 的含义是()。
 - A. 面向对象程序设计
 - B. 对象链接
 - C. 对象嵌入
 - D. 对象链接与嵌入
2. 可视化编程一般基于()的原理。
 - A. 事件过程
 - B. 事件驱动
 - C. 方法
 - D. 事件
3. 属性是用来描述()的一个特性。
 - A. 类
 - B. 方法
 - C. 过程
 - D. 对象



4. 事件是指由系统事先设定的、能被()识别和响应的动作。
- A. 类 B. 方法 C. 过程 D. 对象
5. 方法是特定()的一部分。
- A. 事件过程 B. 事件驱动
C. 对象 D. 事件

第二章

三、问答题

1. 面向对象的程序设计的含义是什么?
2. 事件过程的含义是什么?
3. 怎样理解可视化设计?
4. 可视化编程最主要的特点是什么?
5. 面向对象程序设计的开发工具有哪些?
6. 面向对象程序设计的优点有哪些?
7. 列举在 Windows 操作系统中常见的对象。
8. 什么是类、对象、属性、方法?
9. 事件及事件过程的概念是什么?
10. 事件方法与事件过程的不同之处是什么?

本章概述

本章主要介绍了 Visual Basic 的编程特点、开发环境、事件驱动编程、面向对象编程、Windows 操作系统中常见的对象、类、对象、属性、方法、事件及事件过程的概念、事件方法与事件过程的不同之处。

学习目标

- 了解 Visual Basic 6.0 的特点、开发环境、事件驱动编程、面向对象编程、Windows 操作系统中常见的对象、类、对象、属性、方法、事件及事件过程的概念、事件方法与事件过程的不同之处。
- 掌握 Visual Basic 6.0 的编程特点、开发环境、事件驱动编程、面向对象编程、Windows 操作系统中常见的对象、类、对象、属性、方法、事件及事件过程的概念、事件方法与事件过程的不同之处。
- 掌握 Visual Basic 6.0 的编程特点、开发环境、事件驱动编程、面向对象编程、Windows 操作系统中常见的对象、类、对象、属性、方法、事件及事件过程的概念、事件方法与事件过程的不同之处。
- 掌握 Visual Basic 6.0 的编程特点、开发环境、事件驱动编程、面向对象编程、Windows 操作系统中常见的对象、类、对象、属性、方法、事件及事件过程的概念、事件方法与事件过程的不同之处。
- 掌握 Visual Basic 6.0 的编程特点、开发环境、事件驱动编程、面向对象编程、Windows 操作系统中常见的对象、类、对象、属性、方法、事件及事件过程的概念、事件方法与事件过程的不同之处。

5.1 Visual Basic 简介

5.1.1 Visual Basic 的发展过程

Basic 诞生于上个世纪 60 年代初期,是 Beginner's All-purpose Symbolic Instruction Code (初学者通用符号指令代码)的缩写,它是专门为初学者设计的高级语言,以其简单易学、使用方便的优点,为计算机的推广普及起到了重要作用。但随着计算机技术的飞速发展,硬件性能的增强,以及 Windows 操作系统的流行,Basic 的优点得不到发挥,缺点也逐步显现出来。Microsoft 公司于 1991 年推出了开发 Windows 应用程序而设计的大容量语言工具 Visual Basic (简称 VB),它具有友好的图形用户界面(Graphic User Interface,简称为 GUI)的可操作程序语言,其中“Visual”是“可视化”、“形象化”的意思。Visual Basic 的语言与 Basic 语言的基本相同,因此 Visual Basic 也具有简单易用的特点。Visual Basic 不仅仅是一种语言,也是一种开发工具。它提供了一套可视化设计工具,大大简化了 Windows 程序界面的设计工作。同时编译器采用了面向对象的编程技术,使得 Visual Basic 有了很大的不同。应用 Visual Basic 可以方便地完成从小型应用到大型的多媒体应用,这使 Visual Basic 成为 Windows 应用开发系统的开发工具。

1991 年,微软公司推出了 Visual Basic 1.0。

第二章

Visual Basic 的开发环境

本章概述:

本章主要介绍了 Visual Basic 的功能特点、安装及集成开发环境、基本概念及窗体对象等基本控件的常用属性、方法和事件。然后通过开发一个简单的程序,对 Visual Basic 的程序设计有一个全面的了解,以便为今后开发复杂的应用程序打下良好的基础。

学习目标:

- 了解 Visual Basic 6.0 的特点、安装与启动
- 熟悉集成开发环境的应用
- 掌握可视化程序设计的基本方法和步骤
- 理解工程的含义及对工程的管理方法
- 初步掌握应用程序的打包与发布的方法

2.1 Visual Basic 简介

2.1.1 Visual Basic 的发展过程

Basic 诞生于上个世纪 60 年代初期,是 Beginner's All - purpose Symbolic Instruction Code(初学者通用符号指令代码)的缩写,它是专门为初学者设计的高级语言,以其简单易学、使用方便的特点,为计算机的推广普及起到了重要作用。但随着计算机技术的发展,硬件功能的增强,以及 Windows 操作系统的流行,Basic 的优点得不到发挥,缺点也逐渐显现出来。Microsoft 公司于 1991 年推出了为开发 Windows 应用程序而设计的强有力的编程工具 Visual Basic(简称为 VB),它是具有很好的图形用户界面(Graphic User Interface,简称为 GUI)的可视化程序设计语言,其中“Visual”是“可视化”、“形象化”的意思。Visual Basic 的语法与 Basic 语言的语法基本相同,因此 Visual Basic 也具有易学易用的特点。Visual Basic 不仅仅是一种语言,也是一种开发工具。它提供了一套可视化设计工具,大大简化了 Windows 程序界面的设计工作。同时编程系统采用了面向对象和事件驱动机制,与传统 Basic 有了很大的不同。应用 Visual Basic 可以方便地完成从小型应用程序到大型的数据库管理、多媒体信息处理、Internet 应用程序等系统的开发。

1991 年,微软公司推出了 Visual Basic 1.0。

