

宋伟 吴建国等 编著

Visual

中文 Visual Basic 6.0

高级编程



清华大学出版社
<http://www.tup.tsinghua.edu.cn>

中文 Visual Basic 6.0

高级编程

宋伟 吴建国 等 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

Microsoft 的 Visual 系列开发工具,以其独到的特点和优势赢得了越来越多的开发者,Visual Basic 6.0 中文版是 Microsoft 公司最新推出的 6.0 版 Visual 系列产品之一。

本书是一本介绍中文 Visual Basic 6.0 编程技术的中高级读物,内容包括建造和管理 Visual Basic 工程、完善用户界面、用标准控件设计用户界面、处理文本和图形、响应用户输入事件、调试应用程序、Visual Basic 程序设计核心——对象、Visual Basic 高级编程、ActiveX 控件程序设计、Visual Basic 数据访问技术、掌握数据库编程技术、发布软件。

本书适合需要利用 Visual Basic 进行程序设计的有一定基础的专业或非专业软件开发人员,以及大专院校相关专业的师生参考使用。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

中文 Visual Basic 6.0 高级编程/宋伟等编著. —北京:清华大学出版社,1999.4
ISBN 7-302-03422-2

I. 中… II. 宋… III. BASIC 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(1999)第 07794 号

出版者: 清华大学出版社(北京清华大学校内,邮编 100084)

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

印刷者: 中国科学院印刷厂

发行者: 新华书店总店北京发行所

开 本: 787×1092 1/16 **印张:** 23.5 **字数:** 553 千字

版 次: 1999 年 4 月第 1 版 1999 年 4 月第 1 次印刷

书 号: ISBN 7-302-03422-2/TP·1859

印 数: 0001~5000

定 价: 29.80 元

前 言

Visual Basic 的诞生使编程技术向前迈进了一大步,开创了可视化编程的一代先河。在它的带动下,许多优秀的可视化开发工具相继问世。这些开发工具各有千秋,但它们都或多或少地从 Visual Basic 中汲取了营养。随着组件对象技术的不断进步,以及 Internet 应用的不断普及,Visual Basic 也在不断进步。Visual Basic 6.0 及与之相关的 Visual J++ 6.0, Visual C++ 6.0 等开发工具的推出,使可视化开发工具,以及使用可视化开发工具开发 Internet 应用的技术又向前迈进了一大步。

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编写实用的应用程序。从开发个人或小组使用的小工具,到大型企业应用系统,都可用 Visual Basic 来实现。

Visual Basic 由于功能强大、简单易用而有着广泛的用户,尤其是在数据库、多媒体、商用系统的用户界面以及一般应用程序的设计等方面,Visual Basic 更是有其他开发环境所不能取代的优越之处。升级到 6.0 版后,其功能和性能都得到了增强,而汉化后的 Visual Basic 6.0 中文版,由于有全新的中文界面,更是方便了中国用户的使用。

Visual Basic 6.0 有三种版本,各自满足不同的开发需要:“Visual Basic 学习版”可以使编程人员轻松开发 Windows 95 和 Windows NT 应用程序;“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;“企业版”使得专业编程人员能够开发功能强大的分布式应用程序。本书以三种 Visual Basic 6.0 版本的公共方面为主题,系统、全面地讲述了 Visual Basic 6.0 中文版的高级使用方法。

本书旨在为有一定基础的编程人员提供一条进一步提高 Visual Basic 软件开发技术的途径,使之成为能够胜任较高水平应用软件开发工作的专业软件工程师。

参加本书编写工作的有宋伟、吴建国、徐卫、陈敏、刘芳、阎春芳、傅丹镔、葛晓明和周汉挥,孙波、姚志超负责本书的资料收集工作,田枫、刘振明负责本书的策划工作,宋伟、罗薇统稿,蔡汉辉、罗成审校,全书的图片制作由熊鹰、吴仲齐、唐克明负责,排版录入由王力、李扬负责。

目 录

第一章 Visual Basic 6.0 编程基础	1
1.1 使用代码窗口	1
1.1.1 在代码编辑窗口中编辑代码	2
1.1.2 自动完成编码	2
1.1.3 自定义代码编辑窗口	3
1.2 Visual Basic 的代码编写机制	5
1.3 数字系统及命名的约定	5
1.3.1 什么是数字系统	5
1.3.2 命名约定	6
1.4 常数、变量、属性和赋值	7
1.4.1 常数	7
1.4.2 变量	8
1.4.3 数据类型	11
1.4.4 赋值语句	16
1.5 使用选择和循环控制程序运行	16
1.5.1 选择结构	16
1.5.2 循环结构	19
1.6 使用过程和函数编程	21
1.6.1 Sub 过程	22
1.6.2 Function 过程(函数)	23
1.6.3 使用过程	24
1.6.4 参数传递	26
1.7 使用数组	29
1.7.1 数组的定义	29
1.7.2 多维数组	30
1.7.3 动态数组	31
第二章 工程操作技术	33
2.1 工程概述	33
2.1.1 什么是工程	33
2.1.2 Visual Basic 工程的结构	34

2.2	关于工程的基本操作	35
2.2.1	创建、打开和保存工程	35
2.2.2	向工程中添加文件	36
2.2.3	从工程中删除文件	37
2.2.4	保存文件	37
2.2.5	将文本文件插入代码	37
2.2.6	在工程中添加控件	37
2.2.7	制作可执行文件	39
2.2.8	自定义工程	41
2.3	使用外接程序	41
2.3.1	外接程序管理器	42
2.3.2	使用向导	43
2.4	多工程管理技术	43
2.4.1	添加或删除工程	44
2.4.2	设置启动工程	45
2.5	将工程编译成本地代码	45
2.6	掌握命令行开关	46
2.7	设置应用程序的特征值	47
2.7.1	创建或保存应用程序特征值	47
2.7.2	检索应用程序特征值	48
2.7.3	删除应用程序特征值	49
2.8	资源与资源文件	49
2.9	模 板	50
2.10	条件编译简介	51
第三章	窗体和控件设计	53
3.1	窗体和控件的属性、方法和事件	53
3.2	实现应用程序的窗体	54
3.2.1	窗体的属性	55
3.2.2	窗体的事件和方法	55
3.2.3	创建 Controls 应用程序主窗体	56
3.3	实现应用程序的控件	58
3.3.1	单击按钮执行操作	58
3.3.2	文本控件	61
3.3.3	用于选择的控件	67
3.3.4	显示图片和图形的控件	73
3.3.5	附加控件	78
3.4	几个非常重要的控件	79
3.4.1	焦点	79

3.4.2	Tab 键顺序	80
3.4.3	菜单	80
3.4.4	对话框	82
第四章	利用向导设计程序框架和窗体	84
4.1	利用应用程序向导设计程序框架	84
4.1.1	应用程序向导——介绍	85
4.1.2	应用程序向导——界面类型	85
4.1.3	应用程序向导——菜单	86
4.1.4	应用程序向导——自定义工具栏	87
4.1.5	应用程序向导——资源	87
4.1.6	应用程序向导——Internet 连接性	88
4.1.7	应用程序向导——标准窗体	89
4.1.8	应用程序向导——数据访问窗体	90
4.1.9	应用程序向导——已完成	91
4.2	使用数据窗体向导	92
第五章	完善用户界面	94
5.1	菜单设计技术	94
5.1.1	使用菜单编辑器	94
5.1.2	用菜单编辑器修改菜单	95
5.1.3	确定菜单的级别	95
5.1.4	设计菜单的访问键、快捷键和分隔条	96
5.1.5	显示菜单项的方式	97
5.1.6	编写菜单的事件过程	97
5.1.7	创建弹出式快捷菜单	98
5.2	使用扩充的和外部的控件	99
5.2.1	添加可用控件	100
5.2.2	删除可用控件	101
5.2.3	使用外部对象控件	101
5.3	设计和使用通用对话框	102
5.3.1	使用通用对话框控件	102
5.3.2	“打开”对话框和“另存为”对话框的使用	103
5.3.3	“颜色”对话框的使用	104
5.3.4	“字体”对话框的使用	105
5.3.5	“打印”对话框的使用	107
5.3.6	帮助文件的显示	108
5.4	使用 Windows 95/98 控件	109
5.4.1	ImageList 控件	109
5.4.2	Listview 控件	110

5.4.3	ProgressBar 控件	111
5.4.4	Slider 控件	111
5.4.5	StatusBar 控件	111
5.4.6	TabStrip 控件	113
5.4.7	ToolBar 控件	113
5.4.8	TreeView 控件	115
5.4.9	Rich TextBox 控件的使用	115
5.4.10	SSTab 控件的使用	116
5.5	设计多窗体及多文档界面	117
5.5.1	单文档界面(SDI)和多文档界面(MDI)	117
5.5.2	多文档用户界面	119
5.5.3	使用 MDI 窗体及其子窗体	120
5.5.4	MDI 用户界面的菜单设计方法	122
第六章	用标准控件设计用户界面	125
6.1	Visual Basic 的标准控件	125
6.1.1	内部控件	125
6.1.2	标准 ActiveX 控件	126
6.1.3	控件的命名规则	127
6.1.4	使用控件值	127
6.1.5	使用控件数组	128
6.2	文件系统控件	135
6.2.1	检查文件系统	135
6.2.2	驱动器列表框	136
6.2.3	目录列表框	137
6.2.4	文件列表框	138
6.2.5	组合文件系统控件	138
6.2.6	文件搜索器应用程序	139
6.3	Timer 控件	145
6.3.1	示例应用程序:Alarm.vbp	146
6.3.2	响应 Timer 事件	147
6.4	Frame 控件和 OLE 容器控件	148
6.4.1	Frame 控件	148
6.4.2	OLE 容器控件	149
6.5	CommonDialog 控件	149
6.5.1	“打开”和“另存为”对话框	150
6.5.2	“颜色”对话框	151
6.5.3	“字体”对话框	152
6.5.4	“打印”对话框	153

6.5.5	用 ShowHelp 方法显示帮助文件	154
6.6	滚动条控件	155
6.6.1	滚动条控件的工作原理	155
6.6.2	创建可滚动的图形窗口	156
第七章	处理文本和图形	159
7.1	字体和文本操作	159
7.1.1	使用字体	159
7.1.2	文本操作	160
7.2	数字、日期和时间的格式处理	163
7.3	使用剪贴板对象实现数据传送	165
7.3.1	使用剪贴板处理文本	165
7.3.2	在剪贴板中使用多种格式	166
7.3.3	检查剪贴板中的数据格式	167
7.4	坐标系统简介	168
7.4.1	改变对象的坐标系统	169
7.4.2	转换坐标标尺的刻度模式	171
7.5	使用图形控件	171
7.5.1	添加图片	172
7.5.2	窗体和控件的图形属性	173
7.5.3	刷新图形	173
7.5.4	裁剪区域	174
7.5.5	动态移动控件	175
7.5.6	动态改变控件大小	176
7.5.7	产生简单动画效果	177
7.6	用图形方法创建图形	177
7.6.1	图形方法的绘图原理	178
7.6.2	画点	178
7.6.3	画直线	179
7.6.4	画方框	180
7.6.5	画圆	181
7.6.6	画椭圆	182
7.6.7	在任意位置画图	183
7.6.8	指定线宽	183
7.6.9	指定线型	184
7.6.10	在窗体加载时创建图形	185
7.7	颜色操作	185
7.7.1	使用 RGB 函数	185
7.7.2	使用颜色属性	186

7.8 使用图片对象	186
第八章 响应用户输入事件	188
8.1 响应鼠标事件	188
8.1.1MouseDown 事件	189
8.1.2MouseMove 事件	190
8.1.3MouseUp 事件	191
8.1.4检测鼠标按钮(button 参数)	192
8.2 响应键盘事件	195
8.2.1检测 Shift、Ctrl 和 Alt 键的状态	195
8.2.2低级键盘事件处理	196
8.2.3处理 KeyPress 事件	197
8.2.4处理 KeyUp 和 KeyDown 事件	198
8.2.5处理窗体级键盘事件	199
8.3 拖放操作	201
8.3.1启动自动拖放模式	201
8.3.2改变拖动图标	201
8.3.3放下对象时的响应	202
8.3.4启动拖动控制	202
8.3.5调整控件位置	204
8.4 OLE 拖放操作	204
8.4.1自动拖放和手工拖放	206
8.4.2启动自动 OLE 拖放	206
8.4.3DataObject 对象	207
8.4.4OLE 拖放操作的工作原理	208
8.4.5启动手工 OLE 拖动	209
8.4.6将 OLE 拖放源放到 OLE 拖放目标上	210
8.4.7从 Windows“资源管理器”中拖动文件	211
第九章 调试应用程序	213
9.1 处理应用程序中的错误	213
9.1.1错误处理代码	213
9.1.2设计错误处理程序	215
9.2 Visual Basic 的调试工具	218
9.2.1错误分类	218
9.2.2调试工具的功能	219
9.2.3避免在应用程序中产生错误	219
9.3 应用程序所处的三种模式	220
9.3.1标识当前模式	220
9.3.2模式切换	221

9.4	Visual Basic 的调试技术	221
9.4.1	使用调试窗口	221
9.4.2	使用中断模式	223
9.4.3	运行应用程序的特定部分	228
第十章	Visual Basic 程序设计核心——对象	231
10.1	Visual Basic 对象概述	231
10.1.1	Visual Basic 中的对象	231
10.1.2	使用 Visual Basic 对象	232
10.1.3	对象之间的关系	233
10.2	在 Visual Basic 中创建对象	235
10.2.1	使用对象变量	235
10.2.2	区别特定对象类型和一般对象类型	235
10.2.3	使用 New 关键字创建新对象	236
10.2.4	释放对象	237
10.2.5	将对象传递给过程	237
10.3	使用对象浏览器查找对象信息	238
10.4	创建并应用类	240
10.4.1	应用类	240
10.4.2	定制窗体类	241
10.4.3	使用类模块定义类	242
10.5	类的属性、方法和事件	244
10.5.1	类的属性	245
10.5.2	类的方法	247
10.5.3	将属性和方法设置为缺省的	248
10.5.4	友元属性和方法	248
10.5.5	在类中添加事件	249
10.6	一个创建自己的集合类的实例	254
10.7	Visual Basic 中的多态	257
10.7.1	Visual Basic 提供多态的方法	258
10.7.2	创建并实现接口	258
10.7.3	将属性添加到指定接口中	261
第十一章	Visual Basic 高级编程	262
11.1	使用多工程	262
11.2	应用程序设置值	264
11.2.1	创建和保存应用程序设置值	264
11.2.2	检索应用程序设置值	264
11.2.3	删除应用程序设置值	265
11.3	程序编译技术	266

11.3.1	使用条件编译	266
11.3.2	使用资源文件	267
11.3.3	使用模板	267
11.3.4	使用命令行开关	269
11.3.5	将工程编译成本地代码	270
11.4	创建自定义数据类型	271
11.4.1	声明自定义类型的变量	271
11.4.2	包含数组的自定义类型	272
11.4.3	向过程传递自定义类型	273
11.4.4	包含对象的自定义类型	273
11.4.5	嵌套数据结构	273
11.5	使用枚举数据类型	274
11.6	用集合替代数组	275
第十二章	ActiveX 控件程序设计	277
12.1	ActiveX 控件简介	277
12.1.1	加载 ActiveX 控件的方法	278
12.1.2	ActiveX 控件的图标及功能	278
12.1.3	ActiveX 控件类	280
12.2	用 SysInfo 控件检测系统事件	281
12.2.1	检测当前操作系统及其版本	281
12.2.2	检测屏幕大小和分辨率的变化	282
12.2.3	管理硬件配置和即插即用设备	283
12.2.4	监视电池电源的状态	284
12.3	创建控件的基础知识	286
12.3.1	控件创建术语	286
12.3.2	UserControl 对象	287
12.3.3	创建 ActiveX 控件的方法	288
12.3.4	创建控件的步骤	289
12.4	创建 ActiveX 控件	289
12.4.1	创建 ControlDemo 工程	289
12.4.2	添加 TestCtlDemo 工程	291
12.4.3	在设计时运行 ShapeLabel 控件	292
12.4.4	UserControl 对象的生存周期	294
12.4.5	绘制 ShapeLabel 控件	295
12.4.6	保存 ShapeLabel 控件的属性值	297
12.4.7	给 ShapeLabel 控件添加属性页	299
12.4.8	把一个事件添加到 ShapeLabel 控件中	303
12.4.9	编译 ControlDemo 部件	305

第十三章 Visual Basic 数据访问技术	307
13.1 用 Data(数据)控件访问数据库	307
13.1.1 Data 控件的功能	307
13.1.2 数据绑定控件	308
13.1.3 绑定控件的类型	309
13.1.4 创建简单的数据库应用程序	310
13.1.5 添加、更新和删除记录	312
13.2 外部数据绑定控件	312
13.2.1 DBListBox 和 DBComboBox 控件	313
13.2.2 DBGrid 控件	315
13.2.3 MSFlexGrid 控件	317
13.3 编写代码访问数据库记录	321
13.3.1 定位记录集	321
13.3.2 查找记录	322
13.3.3 更新记录集	324
13.3.4 控制事务	326
13.3.5 确认数据库变化	327
13.3.6 Update 方法	328
13.4 访问外部文件中的数据	329
13.4.1 用于访问文件的函数和语句	329
13.4.2 访问顺序文件	330
13.4.3 访问随机文件	331
13.4.4 访问二进制文件	333
第十四章 掌握数据库编程技术	335
14.1 Visual Basic 数据库系统	335
14.1.1 Visual Basic 数据库模型	335
14.1.2 关系数据库的设计	336
14.2 创建数据库	340
14.2.1 创建数据库的步骤	340
14.2.2 添加索引和关系	341
14.2.3 创建 Biblio.mdb 数据库	343
14.2.4 从外部数据库中附加表	345
14.3 修改数据库	346
14.3.1 向数据库中添加表	346
14.3.2 删除表	346
14.3.3 向表中添加字段	347
14.3.4 更改或删除字段	347
14.3.5 向表中添加索引	347

14.3.6 删除索引	347
14.4 数据库的维护	348
14.4.1 映射数据库	348
14.4.2 压缩数据库	350
14.4.3 修复数据库	351
第十五章 发布软件	352
15.1 Visual Basic 应用软件的组成	352
15.2 安装盘制作向导的使用	352
15.2.1 安装向导的启动	352
15.2.2 安装向导每一步的操作	353
15.3 生成从属文件	358
15.3.1 从属文件解剖	358
15.3.2 创建从属文件	360

第一章 Visual Basic 6.0 编程基础

本章将介绍用 Visual Basic 6.0 进行程序设计的核心内容,即 Visual Basic 的语法规则、Visual Basic 6.0 代码窗口的使用 and 编辑选项以及 Visual Basic 6.0 的新特性。本章的 1.1 节主要介绍在 Visual Basic 6.0 中使用代码窗口的方法和技巧以及 Visual Basic 6.0 代码窗口的一些编辑选项,以后各节着重介绍了 Visual Basic 语言的各要素,这些对于编写事件处理程序将有重要的作用。当然,如果用户对于以前版本的 Visual Basic 或 Basic 语言比较熟悉,这部分内容将会比较容易理解。

1.1 使用代码窗口

本节将系统地介绍代码窗口的使用 and 编辑选项。

在工程管理窗口中选择一个窗体对象或一个模块,然后单击显示代码按钮,代码窗口就会显示出来。当然,在设计窗体上或某个控件上双击也可以使代码窗口显示出来。代码窗口是用来编写事件处理过程或用户自定义过程的,当用户处理多窗体或较复杂的程序代码时,还可以同时打开多个代码窗口。图 1.1 就是一个代码窗口的例子。

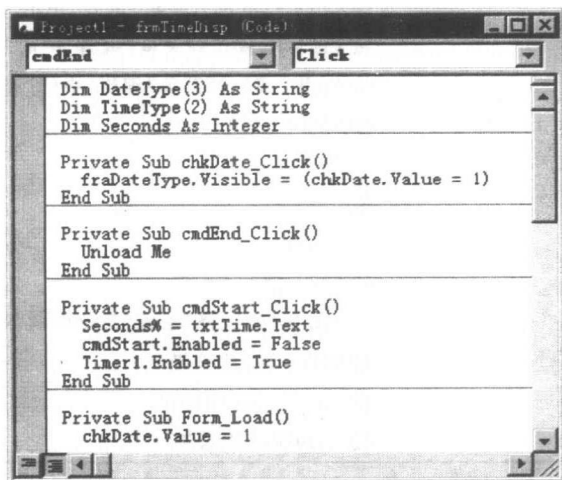


图 1.1 代码窗口

在代码窗口的顶部有两个下拉列表框,其中左边的是对象选择框,右边的是事件选择框,再下面就是代码编辑窗口,并配有滚动条。与以往版本不同的是在代码窗口的底部有

两个显示选择按钮,分别用于选择单一过程显示和全部过程显示。这是 Visual Basic 6.0 的一个新特性,对于方便地使用不同的方式查看程序代码有很大的好处。当选择单一过程显示时,在代码窗口中只显示一个过程;当选择全部过程显示时,在代码窗口中显示整个窗体或模块中的代码,两个过程的代码用一条横线分隔。

1.1.1 在代码编辑窗口中编辑代码

用户在代码编辑窗口中编辑代码时,可以使用所有通用的 Windows 编辑操作。“编辑”菜单中的大多数命令都是针对代码编辑窗口而设的,因此,在编辑代码时,可以经常使用这些命令。

另外,在进行代码编辑的过程中,还可以使“编辑”工具栏显示在屏幕上。“编辑”工具栏中的工具按钮都是为代码编辑而设置的,如图 1.2 所示。

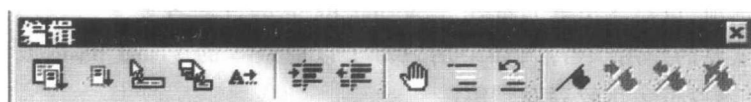


图 1.2 “编辑”工具栏

使用这些工具可以方便地进行代码的编辑操作。另外,在编辑代码过程中,还可以使用一些常用的快捷键,表 1.1 中列出了一些常用编辑操作的方法。

表 1.1 常用的编辑操作

意义	操作
查看代码窗口	按 F7 键
查找	按 Ctrl + F, 或选择菜单“编辑”中的“查找”命令
替换	按 Ctrl + H, 或选择菜单“编辑”中的“替换”命令
查找下一个	按 F3 键, 或选择菜单“编辑”中的“查找下一个”命令
查找上一个	按 Shift + F3 组合键
向右移动一个单词	按 Ctrl + →
向左移动一个单词	按 Ctrl + ←
移到行尾	按 End 键
移到行首	按 Home 键
缩进	按 Tab 键, 或选择“编辑”菜单中的“缩进”命令
取消缩进	按 Shift + Tab 组合键
到窗口顶部	按 Ctrl + Home 组合键
到窗口底部	按 Ctrl + End 组合键

1.1.2 自动完成编码

Visual Basic 能自动填充语句、属性和参数的性能使得编写代码更加方便,这被称之为“智能感知”功能。这使得在输入代码时,编辑器能列举适当的属性、语句和函数原型或枚

举出变量的值。通过“工具”菜单上的“选项”命令打开“选项”对话框,在“选项”对话框的“编辑器”选项卡上选择适当的选项,可以决定是允许还是禁止各代码的设置值。具体操作见下面的说明。

在代码中输入一控件名时,“自动列出成员”特性会列出这个控件的下拉式属性表。键入属性名的前几个字母,就会从表中选中该名字,按 Tab 键将完成这次输入,而按空格键,则在插入空格之前,也会自动完成当前的输入。当不能确认给定的控件有什么样的属性或对控件的属性不特别清楚时,这个选项是非常有帮助的。即使选择了禁止“自动列出成员”特性,也可以使用 Ctrl + J 组合键得到这种性能。

“自动显示快速信息”功能显示语句和函数的语法。当输入合法的 Visual Basic 语句或函数名之后,该语句或函数的语法立即显示在当前行的下面,并用粗体字显示它的第一个参数。在输入第一个参数值之后,第二个参数又出现了,同样也是粗体字。“自动显示快速信息”也可以用 Ctrl + I 组合键得到。

1.1.3 自定义代码编辑窗口

Visual Basic 6.0 允许用户自定义代码窗口的显示格式和操作风格。具体方法是选择菜单“工具”中的“选项”命令,系统显示名为“选项”的对话框,如图 1.3 所示。这里有两个选项卡与代码窗口有关,一个是“编辑器”选项卡,另一个是“编辑器格式”选项卡。

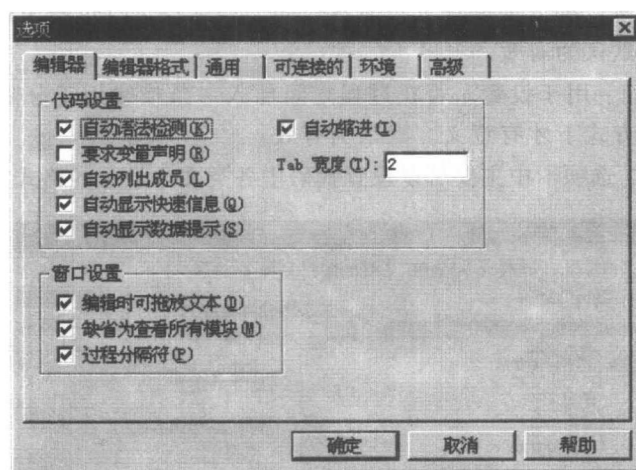


图 1.3 编辑器选项的设置

在“编辑器”选项卡中主要可以设置“代码设置”和“窗口设置”两类共 10 个选项。它们可分成两类,分别是:

1. 代码设置

(1) 自动语法检测: 设置是否进行自动语法检查,当这一选项有效时,在代码窗口中每结束一行代码的编辑,系统会自动检查这一行代码的语法正确性,如果不符合语法规则,系统会提示用户。默认为有效。

(2) 要求变量声明: 设置是否每个变量都需要严格定义才能使用。熟悉以前 Basic 版