

21世纪高等院校课程设计丛书

Visual Basic.NET

课程设计案例精编

李 鑫 主 编

黄 卓

戴金波 副主编

王胜峰



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

Visual Basic.NET 是 Microsoft 公司推出的 .NET 系列可视化开发工具。本书以翔实的内容、精选的案例介绍了如何利用 Visual Basic.NET 进行课程学习和软件制作开发, 强调重要知识点的同时又不失全面性。

本书介绍了 16 个大型案例, 各个案例基本独立, 覆盖了 Windows 下编程的大部分内容, 包括图形图像、多媒体、Internet 网络、数据库、程序通信、ActiveX 技术、XML、GDI+ 等技术, 深入浅出地说明了 Visual Basic.NET 最典型的和最有前途的程序设计方法, 其中很多内容是一般性的 Visual Basic.NET 编程书籍所没有涉及到的。本书应用性极强, 案例均可独立运行, 读者也可以根据这些案例进行研究、修改和扩展, 使其符合自己的需要。总之, 本书具有很强的实用性。

本书内容全面、结构清晰、程序案例丰富, 适合作为刚刚学习完 Visual Basic.NET 基础知识后的锻炼参考, 可作为课程设计时的参考用书。也适合作为中、高级用户开发设计时的参考用书。

本书源代码可以从中国水利水电出版社网站 (<http://www.waterpub.com.cn/softdown/>) 上免费下载。

图书在版编目 (CIP) 数据

Visual Basic.NET 课程设计案例精编 / 李鑫主编. —北京: 中国水利水电出版社, 2005

(21 世纪高等院校课程设计丛书)

ISBN 7-5084-3437-4

I. V... II. 李... III. BASIC 语言—程序设计—高等学校—教学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 139151 号

书 名	Visual Basic.NET 课程设计案例精编
作 者	李 鑫 主 编 黄 卓 戴金波 王胜峰 副主编
出版 发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 63202266 (总机)、68331835 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	787mm×1092mm 16 开本 19.5 印张 479 千字
版 次	2006 年 1 月第 1 版 2006 年 1 月第 1 次印刷
印 数	0001—5000 册
定 价	32.00 元

凡购买我社图书, 如有缺页、倒页、脱页的, 本社营销中心负责调换
版权所有·侵权必究

丛书序

课程设计是教学计划中的一个重要环节。通过课程设计，学生能够得到系统的技能训练，从而巩固和加强所学的专业理论知识，其目的是培养学生的综合运用能力，使学生成为具有扎实的理论基础和较强的独立动手能力的专业人才。

随着社会对复合型、应用型人才的需要，各高校对学生在课程设计上的要求越来越高，课程设计的选题也越来越需要结合实际应用。考虑到目前课程设计重视实际操作的需要，我们组织具有丰富实际开发和教学经验的老师编写了本套丛书。书中所选的案例皆取自作者平时所做的项目，具有相当强的实用性和可操作性，其中有些案例项目稍加扩充，即可成为一个功能完整的项目系统。

我们编写本套丛书的目的是给学生和老师在做课程设计的时候提供一个参考，老师可以先让学生按照本书例子的引导完成整个案例的制作，然后要求学生仿照该案例或课后习题有所扩充，最后再独立地以自己的想法做出各具特色的课程设计的实例。因为知识的应用过程就是一个熟练的过程，你可以先照着别人的例子作，在亲自实践后才能了解到其中的实际问题，从而在解决问题中掌握更多实用的知识，并巩固和加强所学的专业理论知识。这样才能在走入工作岗位后更好地将学到的知识应用到实际开发中去。

本套丛书具有以下特点：

内容全面、综合：精心选取各课题开发中具有代表性的若干个案例，全面覆盖各课题开发技术中的重点和难点，内容详实。

选例实用、典型：案例的选取具有代表性，是在实际开发工作中经常能遇到的。书中所有的例子都经过验证实现，读者可通过实例的学习对相应技术点有清晰直观的了解。

紧扣课设、实用：丛书最大程度地强调课程设计的特色，书中所有的例子尽量做到按学生课程设计的思路编排，以达到更易读懂、更实用的目的。

注释清楚、明了：对于学习过程中易出现问题的部分都加了详细的注释说明，以便学生实际课程设计制作的过程中思路更清晰、明了。

课后练习选取有代表性：为了方便学生练习和老师布置作业，部分案例后还选取了具有代表性的实例题目，并作了简单的实现思路说明，适合不同层次的学生练习。

通过对本套丛书相应课程设计实例的学习，相信您一定能轻松完成自己的课程设计，做出满意的课程设作业。

本套丛书目前涉及到的课程设计课题有：Delphi、Visual FoxPro、Visual Basic、Visual C++、PowerBuilder、C、汇编语言、Java、ASP、SQL Server、Access、JSP、Oracle、数据库技术、网页制作等。在以后的时间里我们将加入更多、更实用的课程设计课题供广大读者和老师参考。也敬请广大读者及时和我们沟通，从而使我们能更好地为大家服务。

真诚地感谢参与本套丛书编写的老师们，是他们认真敬业的精神保证了本套丛书能符

合课程设计的要求，能更好地为读者学习应用。也非常感谢中国水利水电出版社万水公司的图书策划编辑，是他们本着为学生、为读者服务的精神策划了本套实用的课程设计方案精编丛书。

期待各位读者的意见和建议，希望各位不吝赐教，来信请至 xinyuanxuan@263.net。最后祝愿各位读者能通过本套丛书学习到更多更实用的知识，为将来的发展奠一块更好的基石。

丛书编委会
2005 年 10 月

前 言

Visual Basic.NET 作为 Visual Studio.NET 家族中的一分子,是微软公司基于最新的.NET 思想开发的以 Basic 语言为基础的可视化集成开发工具,与 Visual C++.NET、Visual C#.NET 等一起被称为 Visual Studio.NET 的集成开发环境 (IDE)。

Visual Basic.NET 相对于 Visual Basic 来说,在开发环境、调试环境、编程语言和编程方法上都有了很大的改变。为了使广大读者朋友尽快掌握这种新的开发工具,特此编写了《Visual Basic.NET 课程设计案例精编》一书。

本书从具体的案例出发,详细地介绍了利用 Visual Basic.NET 开发各种类型应用程序的步骤、方法和技巧。全书共有 16 个大型案例,其主要内容有:

案例一:采用多方面的界面技巧,开发了一个漂亮的时间管理程序。

案例二:使用 Visual Basic.NET 工具开发了一款注册表管理程序。

案例三:主要介绍在 Visual Basic.NET 中管理线程的方法。

案例四:主要介绍在 Visual Basic.NET 中开发打印机程序的方法。

案例五:使用 Visual Basic.NET 工具开发了一款打字练习程序。

案例六:模仿 Windows 资源管理器使用 Visual Basic.NET 工具开发了一款功能相似的软件。

案例七:主要介绍在 Visual Basic.NET 中开发基于 Web 服务的网络计算器的方法。

案例八:主要介绍在 Visual Basic.NET 中使用数据库开发球员信息管理程序的方法。

案例九:主要介绍在 Visual Basic.NET 中开发网页信息浏览程序的方法。

案例十:主要介绍在 Visual Basic.NET 中开发电子邮件收发系统的方法。

案例十一:主要介绍在 Visual Basic.NET 中使用基于 TCP 协议的网络通信的方法。

案例十二:主要介绍在 Visual Basic.NET 中开发多媒体程序的方法。

案例十三:主要介绍在 Visual Basic.NET 中使用 GDI+ 技术的方法。

案例十四:主要介绍网络订餐订单接收处理系统的开发过程。

案例十五:主要介绍学生成绩信息管理系统的编程技巧和开发过程。

案例十六:主要介绍在 Visual Basic.NET 中创建安装程序的方法。

本书所有源程序都在 Visual Basic.NET 上编译通过,所有案例的开发代码和部分课后作业的源代码可以从中国水利水电出版社网站 (<http://www.waterpub.com.cn/softdown/>) 上免费下载,以备读者参考。

本书由李鑫主编,黄卓、戴金波、王胜峰副主编,参加编写的人员还有晏愈光、杨鑫、张天、付苗苗、赵宇、李晓宁、李英娟、王敬栋、李昆、郝思嘉、房明浩、万春磊、杜波、梁奕斌、徐强、刘燕祎、卢福子、李强等。

由于时间仓促,加之作者水平有限,书中难免有不足之处,欢迎广大读者提出宝贵意见,作者 E-mail 地址:nanosnows@126.com。

编 者

2005 年 12 月

目 录

丛书序

前言

案例一 时间管理程序	1
1.1 程序设计目的	1
1.2 程序功能的介绍	1
1.3 程序设计	2
1.3.1 时间管理界面和代码的设计	2
1.3.2 动画时钟的实现	7
1.3.3 闹铃设定界面和代码的设计	11
1.3.4 秒表界面和代码的设计	16
1.4 小结	20
1.5 课后作业	20
案例二 注册表管理程序	24
2.1 程序设计目的	24
2.2 程序功能的介绍	25
2.3 程序设计	26
2.3.1 Registry 类和 RegistryKey 类的简介	26
2.3.2 注册表读写界面和代码的设计	27
2.3.3 CPU 和磁盘信息的获取	31
2.4 小结	33
2.5 课后作业	34
案例三 线程管理程序	36
3.1 程序设计目的	36
3.2 程序功能的介绍	36
3.3 程序设计	37
3.3.1 获取多线程信息	37
3.3.2 多线程应用程序的创建	42
3.3.3 线程池的使用	46
3.4 小结	52
3.5 课后作业	53
案例四 打印机程序	56
4.1 程序设计目的	56

4.2	程序功能的介绍	56
4.3	程序设计	57
4.3.1	设计界面	57
4.3.2	文件读取和写入代码编写	59
4.3.3	打印控件的设置和打印代码的编写	62
4.3.4	脱机打印功能的实现	64
4.4	小结	71
4.5	课后作业	71
案例五	打字练习程序	73
5.1	程序设计目的	73
5.2	程序功能的介绍	73
5.3	程序设计	73
5.3.1	程序界面设计	73
5.3.2	程序代码的设计	75
5.3.3	控件动画的实现	82
5.3.4	控件超链接的实现	82
5.4	小结	84
5.5	课后作业	84
案例六	Windows 资源管理器	88
6.1	程序设计目的	88
6.2	程序功能的介绍	88
6.3	程序设计	89
6.3.1	FSO 对象模型简介	89
6.3.2	Windows 资源管理器程序主界面和代码的设计	91
6.3.3	文件系统在 TreeView 控件中的显示	94
6.3.4	浏览区显示文件夹内容的代码设计	100
6.3.5	菜单栏的界面和代码设计	102
6.3.6	工具栏的界面和代码设计	110
6.3.7	状态栏的界面和代码设计	112
6.4	小结	113
6.5	课后作业	113
案例七	基于 Web 服务的网络计算器	117
7.1	程序设计目的	117
7.2	程序功能的介绍	117
7.3	程序设计	118
7.3.1	Web 服务概述	118
7.3.2	Web 服务的创建	120

7.3.3	网络计算器客户端的界面和代码设计	134
7.4	小结	140
7.5	课后作业	140
案例八	球员信息管理程序	142
8.1	程序设计目的	142
8.2	程序功能的介绍	142
8.3	程序设计	143
8.3.1	数据库的建立	143
8.3.2	主窗口界面及代码设计	144
8.3.3	球员信息编辑窗口的界面和代码设计	148
8.3.4	添加球员窗口的界面和代码设计	153
8.3.5	程序使用	155
8.4	小结	156
8.5	课后作业	156
案例九	网页浏览器程序	159
9.1	程序设计目的	159
9.2	程序功能的介绍	159
9.3	程序设计	159
9.3.1	AxWebBrowser 的简介	159
9.3.2	网页浏览器程序的界面设计	160
9.3.3	程序代码的设计	163
9.3.4	CHM 帮助文件的制作与应用	168
9.4	小结	174
9.5	课后作业	174
案例十	电子邮件发送和接收程序	176
10.1	程序设计目的	176
10.2	程序功能的介绍	176
10.3	程序设计	177
10.3.1	电子邮件发送程序界面的设计	177
10.3.2	电子邮件发送程序代码的设计	179
10.3.3	电子邮件接收程序界面的设计	183
10.3.4	电子邮件接收程序代码的设计	185
10.4	小结	188
10.5	课后作业	189
案例十一	基于 TCP 协议的网络通信	191
11.1	程序设计目的	191
11.2	程序功能的介绍	191

11.3	程序设计	191
11.3.1	Socket 简介	191
11.3.2	程序界面设计	193
11.3.3	程序代码设计	194
11.3.4	程序使用	205
11.4	小结	206
11.5	课后作业	206
案例十二	多媒体播放器程序	208
12.1	程序设计目的	208
12.2	程序功能的介绍	208
12.3	程序设计	209
12.3.1	基本播放区界面与代码设计	209
12.3.2	录音区界面与代码设计	211
12.3.3	文件播放信息区的界面和代码设计	217
12.3.4	文本记录区的界面和代码设计	218
12.3.5	音频参数设置区的界面和代码设计	220
12.3.6	菜单栏的设计和代码实现	222
12.3.7	程序使用	226
12.4	小结	227
12.5	课后作业	227
案例十三	GDI+简易画图软件的开发	229
13.1	程序设计目的	229
13.2	程序功能的介绍	229
13.3	程序设计	230
13.3.1	GDI+技术	230
13.3.2	使用鼠标绘图	233
13.3.3	图像处理	237
13.3.4	调色板控件设计	238
13.3.5	主程序界面和代码设计	243
13.4	小结	245
13.5	课后作业	245
案例十四	网络订餐订单处理系统	248
14.1	程序设计目的	248
14.2	程序功能的介绍	249
14.3	程序设计	249
14.3.1	订餐订单发送界面和代码设计	249
14.3.2	订餐订单接收和监测系统代码设计	253

14.3.3	订餐订单响应系统代码设计	254
14.3.4	订单处理系统代码设计	257
14.3.5	程序运行	260
14.4	小结	262
14.5	课后作业	262
案例十五	学生成绩信息管理系统	265
15.1	程序设计目的	265
15.2	程序功能的介绍	265
15.3	程序设计	265
15.3.1	ADO.NET 数据对象模型	265
15.3.2	建立查询	267
15.3.3	程序界面设计	269
15.3.4	程序代码设计	273
15.4	小结	284
15.5	课后作业	284
案例十六	程序打包发行	285
16.1	程序设计目的	285
16.2	程序功能的介绍	285
16.3	程序设计	286
16.3.1	利用安装向导为程序制作安装文件	286
16.3.2	Web 访问数据库应用程序的设计与实现	292
16.3.3	利用 Web 安装项目为 Web 应用程序制作安装文件	299
16.4	小结	301
16.5	课后作业	301

案例一 时间管理程序

1.1 程序设计目的

一般在大的应用程序，尤其是大型的商用程序中，都会为用户提供一些实在而有用的信息，例如当前的时间和日期。

在可视化编程中，日期控件是一种使用非常广泛的控件。在 VB6.0 里，方便的日期时间操作就是 VB 的一个非常强大的功能，在 VB.NET 中这一点变得更加突出。VB.NET 提供了两个时间控件：MonthCalendar 和 DateTimePicker 控件。前者用来包含日期；后者既可以包含日期，又提供了时间输入机制。这大幅简化了程序开发人员在开发与时间有关的应用程序时的工作量。本案例将利用这两个控件进行时间和日期的管理。

一个 Windows 应用程序，需要美观的界面设计，而漂亮的界面又离不开文本、图像、像素、背景要素，VB.NET 就具备了强大的图形图像处理功能。本案例将结合时间处理、数学函数编程和 VB.NET 特有的 GDI+ 技术，绘制一个带有表盘、时钟刻度和指针的时钟动画。

1.2 程序功能的介绍

鼠标双击 Windows 操作系统界面右下角任务栏中的时间，将弹出一个时间管理程序。本案例将模仿该时间管理程序进行开发设计，下面分 4 个部分进行介绍。

1. 日历的显示和编辑

用户运行程序，在“时间管理”标签页的日历显示区中，用户可以根据自己的喜好或者需要，选择不同的日历显示风格和特征；在时间管理显示区中，可以设置数据格式、数据类型等。程序运行后，还将在 Windows 桌面的工具栏上产生一个托盘图标。

2. 时钟动画

程序运行后，在“时间管理”标签页的时间管理显示区中将显示一个时钟动画，带有蓝色的表盘和三根转动的时针、分针和秒针，用户可以从该动画中直接读取当前时间。

3. 闹铃设定

用户如果想要应用程序在某个特定时间段进行某个特殊的提示，可以通过本案例程序的“闹铃设定”标签页的闹铃设定区域进行设置，设置内容包括：日期设定、时间设定和文本提示等。

4. 秒表功能

本案例还制作了一个仿液晶的电子秒表，进入应用程序的“秒表”标签页，该页面上有三个按钮，用户可以进行秒表的运行、暂停和重置设置等。

以上 4 个方面内容是本案例的主要功能，事实上它还是不完善的。熟悉时间管理的用户会发现这个用于学习的时间管理案例程序还缺少很多功能，如时间随着时区变化来调整、与网

络时间的同步等。

1.3 程序设计

1.3.1 时间管理界面和代码的设计

1. MonthCalendar 控件和 DateTimePicker 控件简介

MonthCalendar 控件显示包含本月所有日期的方格，用户可以按月份标题两侧的箭头来变换月份。如图 1-1 所示。



图 1-1 MonthCalendar 控件

DateTimePicker 控件可以看成是 ComboBox 控件和 MonthCalendar 控件的组合，给用户提供下拉式列表的方式，输入特定的日期类型的数据：“下拉式日历”和“时间格式”。前者如同下拉式列表，在输入日期的方块右边有个向下的按钮，当用户按下该按钮时，将显示一个 MonthCalendar 控件；而后者则在输入框的右边有个类似 NumericUpDown 控件的机制，当鼠标指针选择输入方块中的任何单位（如年、月等）时，这个数字可以利用右边的向上向下按钮来增减数值，如图 1-2 所示。DateTimePicker 控件日期和时间的显示格式枚举见表 1-1，该控件的有效格式字符串及说明见表 1-2。控件的模式是由 ShowUpDown 属性来决定的。当 ShowUpDown 属性为 True 时，控件显示为“时间格式”模式，反之则为“下拉式日历”格式。



图 1-2 DateTimePicker 控件的两种模式

表 1-1 DateTimePicker 控件日期和时间的显示格式枚举

成员	说明
Custom	自定义格式显示日期时间值
Long	以用户操作系统设置的长日期格式显示日期时间值
Short	以用户操作系统设置的短日期格式显示日期时间值
Time	以用户操作系统设置的时间格式显示日期时间值

表 1-2 DateTimePicker 控件有效格式字符串及说明

格式字符串	说明
d	一位数或者两位数的天数
dd	两位数的天数。一位数天数的前面加个 0
ddd	三个字符的星期几缩写
dddd	完整的星期几名称
h	12 小时格式的一位数或者两位数
hh	12 小时格式的两位数小时数。一位数数值前面加个 0
H	24 小时格式的一位数或两位数小时数
HH	24 小时格式的两位数小时数。一位数数值前面加个 0
m	一位数或者两位数分钟值
mm	两位数分钟值。一位数数值前面加个 0
M	一位数或两位数月份值
MM	两位数月份值。一位数数值前面加个 0
MMM	三个字符的月份缩写
MMMM	完整的月份名
s	一位数或两位数秒数
ss	两位数秒数。一位数数值前面加一个 0
t	一个字母的 AM/PM 缩写（“AM”显示为“A”）
tt	两个字母的 AM/PM 缩写（“AM”显示为“AM”）
y	一位数的年份（“2001”显示为“1”）
yy	年份的最后两位数（“2001”显示为“01”）
yyy	完整的年份（“2001”显示为“2001”）

2. 时间管理界面设计

运行 VB.NET，选择“文件”→“新建”→“项目”，在弹出的对话框中选择“Windows 应用程序”，将其命名为 TimeManager。在 Form1 窗体中添加一个 TabControl 控件、一个 MonthCalendar 控件、一个 DateTimePicker 控件、一个 TextBox 控件、两个 Label 控件、三个 CheckBox 控件、三个 ComboBox 控件和四个 GroupBox 控件，修改各控件的属性设置情况见表 1-3。调整控件的大小和位置，界面效果如图 1-3 所示。

表 1-3 主窗口控件对象属性列表

对象	控件名称	属性	设置值
GroupBox	GroupBox1	Text	日历
	GroupBox2	Text	时间管理
	GroupBox3/4	Text	选项
ComboBox	ComboBox1/2/3	Text	
CheckBox	CheckBox1	Text	ShowToday
	CheckBox2	Text	ShowTodayCircle
	CheckBox3	Text	ShowWeekNumbers

续表

对象	控件名称	属性	设置值
Label	Label1	Text	DateFormat
	Label2	Text	DateStyle
	Label3	Text	FirstDayofWeek
Form	Form1	Text	时间管理程序
TabControl	TabPage1	Text	时间管理



图 1-3 程序主界面图

3. 时间管理代码设计

程序的主界面设计完毕，下一步编写代码。首先添加窗体加载时所处理的代码，即为三个 ComboBox 控件添加项目，代码如下：

```
Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles MyBase.Load
...
Me.ComboBox1.Items.Add("long")
Me.ComboBox1.Items.Add("short")
Me.ComboBox1.Items.Add("time")
Me.ComboBox1.Items.Add("custom1")
Me.ComboBox1.Items.Add("custom2")
Me.ComboBox1.Items.Add("custom3")
Me.ComboBox1.Items.Add("custom4")
'添加日期显示格式
Me.ComboBox2.Items.Add("下拉式")
Me.ComboBox2.Items.Add("单击式")
'添加日历风格
Me.ComboBox3.Items.Add("星期日")
Me.ComboBox3.Items.Add("星期一")
Me.ComboBox3.Items.Add("星期二")
Me.ComboBox3.Items.Add("星期三")
Me.ComboBox3.Items.Add("星期四")
Me.ComboBox3.Items.Add("星期五")
Me.ComboBox3.Items.Add("星期六")
'添加起始星期日
```

```

Me.ComboBox1.SelectedIndex = 0
Me.ComboBox2.SelectedIndex = 0
Me.ComboBox3.SelectedIndex = 0
Me.TextBox1.Text = TimeOfDay
ComboDate.Text = Date.Now.Date
'默认设为当前日期

```

```
...
```

```
End Sub
```

接下来编写改变 DateTimePicker 控件数据格式显示的代码。该控件数据格式显示的改变是通过 ComboBox1 控件的 SelectedIndexChanged 事件来实现的，代码如下：

```

Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles ComboBox1.SelectedIndexChanged
    Select Case Me.ComboBox1.SelectedIndex
        Case 0
            Me.DateTimePicker1.Format = DateTimePickerFormat.Long
        Case 1
            Me.DateTimePicker1.Format = DateTimePickerFormat.Short
        Case 2
            Me.DateTimePicker1.Format = DateTimePickerFormat.Time
        Case 3
            Me.DateTimePicker1.Format = DateTimePickerFormat.Custom
            Me.DateTimePicker1.CustomFormat = "MMMM dd, yyyy - dddd"
        Case 4
            Me.DateTimePicker1.Format = DateTimePickerFormat.Custom
            Me.DateTimePicker1.CustomFormat = "ddd,hh:mm:ss,tt"
        Case 5
            Me.DateTimePicker1.Format = DateTimePickerFormat.Custom
            Me.DateTimePicker1.CustomFormat = "dddd,HH:mm:ss,tt"
        Case 6
            Me.DateTimePicker1.Format = DateTimePickerFormat.Custom
            Me.DateTimePicker1.CustomFormat = "dddd,HH:m:ss,t"
    End Select
    Me.TextBox1.Text = TimeOfDay
    '显示当天的时间
End Sub

```

同样为 ComboBox2 控件和 ComboBox3 控件添加代码。前者是改变 DateTimePicker 控件的数据类型，后者是改变 MonthCalendar 控件所显示的每个星期中天数显示的顺序，代码如下：

```

Private Sub ComboBox2_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles ComboBox2.SelectedIndexChanged
    Select Case Me.ComboBox2.SelectedIndex
        Case 0
            Me.DateTimePicker1.ShowUpDown = False
        Case 1
            Me.DateTimePicker1.ShowUpDown = True
    End Select
    Me.TextBox1.Text = TimeOfDay
End Sub

Private Sub ComboBox3_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles ComboBox3.SelectedIndexChanged

```

```

Select Case Me.ComboBox3.SelectedIndex
    Case 0
        Me.MonthCalendar1.FirstDayOfWeek = Day.Sunday
        '选择星期日, 也是默认模式, 以下类似
    Case 1
        Me.MonthCalendar1.FirstDayOfWeek = Day.Monday
    Case 2
        Me.MonthCalendar1.FirstDayOfWeek = Day.Tuesday
    Case 3
        Me.MonthCalendar1.FirstDayOfWeek = Day.Wednesday
    Case 4
        Me.MonthCalendar1.FirstDayOfWeek = Day.Thursday
    Case 5
        Me.MonthCalendar1.FirstDayOfWeek = Day.Friday
    Case 6
        Me.MonthCalendar1.FirstDayOfWeek = Day.Saturday
End Select
Me.TextBox1.Text = TimeOfDay
End Sub

```

接下来为 **CheckBox** 控件的 **CheckStateChanged** 方法添加相关代码, 主要功能是以不同的方式来显示日历中某个特定的日期:

```

Private Sub CheckBox1_CheckStateChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CheckBox1.CheckStateChanged
    If Me.CheckBox1.Checked Then
        Me.MonthCalendar1.ShowToday = True
        '显示今天日期
    Else
        Me.MonthCalendar1.ShowToday = False
    End If
End Sub

Private Sub CheckBox2_CheckStateChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CheckBox2.CheckStateChanged
    If Me.CheckBox2.Checked Then
        Me.MonthCalendar1.ShowTodayCircle = True
        '显示今天前面的圆
    Else
        Me.MonthCalendar1.ShowTodayCircle = False
    End If
End Sub

Private Sub CheckBox3_CheckStateChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CheckBox3.CheckStateChanged
    If Me.CheckBox3.Checked Then
        Me.MonthCalendar1.ShowWeekNumbers = True
        '显示星期数
    Else
        Me.MonthCalendar1.ShowWeekNumbers = False
    End If
    Me.TextBox1.Text = TimeOfDay
End Sub

```


至此，时间管理部分的界面和代码设计已经基本完成，下一步我们将学习利用 VB.NET 进行应用程序托盘图标的制作。

4. 应用程序托盘图标的实现

许多知名软件，如诺顿等，在启动后都会在 Windows 窗口右下角的任务栏中出现一个托盘图标，以表示程序正在运行。托盘图标方便了用户对程序的操作。如何在自己的程序里实现托盘图标的制作呢？

在 VB 开发环境中，托盘图标的制作是通过 API 函数来实现的，过程比较烦琐。VB.NET 提供了专门的托盘图标生成控件 `NotifyIcon`，可以很方便地制作应用程序的托盘图标。其操作步骤如下：

往项目的设计界面上添加一个 `NotifyIcon` 控件。在 `NotifyIcon` 的属性栏里，单击 `Icon` 属性旁边的  按钮，在弹出的对话框中选择一个合适的图标并添加到该控件的属性中。程序运行后，就可以在窗体的右下角的任务栏里发现程序的图标。

对于成熟的应用程序，鼠标停留在托盘图标上时，将会出现一个文本提示。这是通过 `NotifyIcon` 控件的 `Text` 属性来实现的，本案例将通过下面的代码来实现：

```
Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs)
    Handles MyBase.Load
```

```
...
```

```
    NotifyIcon1.Text = Date.Now.ToString
```

```
    '鼠标停留在托盘图标上时，弹出的文本将显示当前时间
```

```
End Sub
```

现在，时间管理程序的功能已经完成，按 F5 键运行程序，就可以在程序界面上选取不同的显示格式和风格，如图 1-4 所示。

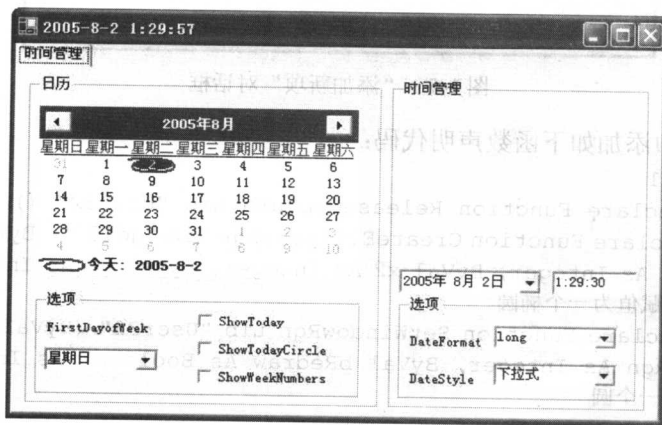


图 1-4 程序运行效果

下一步的任务是利用 VB.NET 的 GDI+ 技术和数学函数在图 1-4 的时间管理区中添加一个动画时钟。

1.3.2 动画时钟的实现

本案例动画时钟的设计思想如下：整个时钟基于 VB.NET 的 `PictureBox` 控件，首先添加一个 `PictureBox1` 控件，利用 Windows API 函数来生成一个圆形的表盘；然后使用 GDI 技术实