

21世纪高等学校规划教材 | 计算机应用



C#课程设计 案例教程

清华大学出版社

21世纪高等学校规划教材 | 计算机应用

C#课程设计 案例教程

万小霞 主编
柳 盛 丁向民 编著

清华大学出版社
北 京

内 容 简 介

本书根据“C# 语言程序设计”理论课教学大纲的要求,结合作者多年程序设计语言教学的经验,给出了C#程序课程设计的主要案例内容。本书从实际应用出发,给出8个案例,每一个案例首先说明设计的目的,然后给出案例要求实现的主要功能,简要描述实现这些功能需要使用的技术,最后给出系统的设计实现步骤。主要实践的编程技术有:常用控件的使用、数据库编程、图像处理技术、网络编程等。

本书可作为在校大学生学习使用C#进行课程设计的参考资料,也可作为非计算机专业学生学习C#语言的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C# 课程设计案例教程/万小霞主编.--北京:清华大学出版社,2012.7

(21世纪高等学校规划教材·计算机应用)

ISBN 978-7-302-28664-6

I. ①C… II. ①万… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2012)第077079号

责任编辑:高买花

封面设计:傅瑞学

责任校对:焦丽丽

责任印制:李红英

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:11.75

字 数:286千字

版 次:2012年7月第1版

印 次:2012年7月第1次印刷

印 数:1~3000

定 价:20.00元

产品编号:041055-01

前言

C# 是 Microsoft 公司发布的一种简洁、类型安全的面向对象程序设计语言。开发人员利用它构建在 .NET Framework 上运行各种安全、可靠的应用程序,如传统的 Windows 客户端应用程序、XML Web Services、分布式组件、数据库应用程序等。

C# 语言继承了 C++ 和 Java 的很多优点,不再支持指针,不支持多重继承。C# 的泛型方法和类型提供了更出色的类型安全性能,迭代器允许自定义迭代行为,LINQ 表达式使强类型查询成为了一流的语言构造。总之,C# 是一种简洁高效的语言,它的灵活性、安全性和技术成熟度已经成为目前网络编程主要技术之一,是目前很流行的网络编程工具。

C# 编程语言是众多高校计算机专业学生的必修或选修课,这对于培养学生的计算机应用能力具有重要的意义。学生学习 C# 编程语言可以掌握 C# 的基本知识,掌握面向对象程序设计和编程的方法,提高开发 .NET 应用程序的能力。

课程设计是高校教学计划中一个重要的环节,课程设计能使得到系统的技能训练,巩固学生所学理论知识,培养学生程序设计的综合能力。本书根据 C# 语言程序设计理论教学大纲和实践教学大纲的要求,介绍了几个课程设计的任务,使学生对 C# 语言讲授的理论知识点加以深入实践。下面对课程设计几个案例进行简单的介绍。

案例 1: 模拟 Windows 操作系统自带的 notepad 应用程序的功能,掌握 RichTextBox 控件的常用方法,掌握通用对话框的使用。

案例 2: 图片浏览器功能,实现对指定目录下的图片文件进行浏览操作,并对打开的图片文件进行简单的图像处理。

案例 3: 模拟 Windows 操作系统自带的 mspaint 应用程序的功能,实现简单的画图功能,主要掌握 GDI+ 的绘图功能。

案例 4: 模拟 Windows 操作系统自带的资源管理器功能,掌握 ListView 控件和 TreeView 控件的使用。

案例 5: 通讯录管理系统,管理联系人信息,学习使用 XML 进行数据的存储以及对 XML 文档的读取和解析。

案例 6: 贪吃蛇游戏实现贪吃蛇的移动,随机给出食物,掌握 Timer、GDI+ 绘图等知识的应用。

案例 7: 学生成绩管理系统,介绍 ADO.NET 数据库开发的基本知识和应用。

案例 8: 简易聊天室的实现,介绍网络编程的基础知识,利用理论知识进行简单的网络通信程序设计。

本书结合案例实例制作,以学生的课程设计为背景,适用于 C# 程序设计的课程设计。

本书由万小霞主编,并由万小霞完成统编定稿工作。参加编写工作的还有柳盛和丁向民。案例 1、案例 3、案例 4、案例 5、案例 6 和案例 8 由万小霞编写,案例 2 由柳盛协助编写,

案例7 由丁向民协助编写。参与本书的写作、整理、调试、校对的还有吴国民、杨梅、周姣、张煌、王从浩、朱晓华、王月、陈丹、陈俊、何宏博、张峻崎、晁成汗等,在此一一表示感谢!

本书编写过程中参考了相关文献和资料,在此对原作者深表感谢。另外,互联网是本书的另一个重要参考资料的来源。由于网上许多资料无法找到其出处,所以书中如有内容涉及相关人士的知识产权,请给予谅解并及时与我们联系。

由于本人水平有限,加之时间仓促,书中错误和不足之处在所难免,恳请专家、读者批评指正。Email: wxiaoxia_su@163.com。

编 者

目 录

案例1 文本编辑器	1
1.1 功能要求	1
1.2 相关知识介绍	1
1.2.1 RichTextBox 控件	1
1.2.2 OpenFileDialog /SaveFileDialog 对话框	2
1.2.3 PrintPreviewDialog 对话框	3
1.2.4 PrintDialog 对话框	3
1.2.5 PrintDocument 对话框	3
1.2.6 PageSetupDialog 对话框	3
1.2.7 FontDialog 对话框	3
1.2.8 ColorDialog 对话框	4
1.3 程序设计	4
1.3.1 用户界面设计	4
1.3.2 主要功能模块的代码	7
1.3.3 测试运行	20
1.4 小结	20
案例2 图片浏览器	21
2.1 功能要求	21
2.2 相关知识介绍	21
2.2.1 图像显示	21
2.2.2 图像处理	24
2.3 程序设计	26
2.3.1 需求分析	26
2.3.2 功能划分和界面设计	26
2.3.3 主要功能模块的代码	28
2.3.4 程序运行效果	36
2.4 小结	37
案例3 画图板的实现	38
3.1 功能要求	38
3.2 相关知识介绍	38

3.2.1	GDI+概述	38
3.2.2	GDI+绘图基础	38
3.2.3	GDI+基本绘图方法	40
3.3	程序设计	41
3.3.1	需求分析	41
3.3.2	系统设计	42
3.3.3	界面设计	43
3.3.4	主要功能模块的代码	46
3.3.5	程序运行效果	62
3.4	小结	62
案例4	模拟资源管理器	63
4.1	功能要求	63
4.2	相关知识介绍	63
4.2.1	TreeView 控件	63
4.2.2	ListView 控件	65
4.2.3	File 类和 FileInfo 类	67
4.2.4	Directory 类和 DirectoryInfo 类	71
4.3	程序设计	74
4.3.1	用户界面设计	74
4.3.2	主要功能模块的代码	77
4.3.3	测试运行	97
4.4	小结	98
案例5	通讯录管理	99
5.1	功能要求	99
5.2	相关知识介绍	99
5.2.1	XML 基础知识	99
5.2.2	XML 文档处理	101
5.3	程序设计	104
5.3.1	需求分析	104
5.3.2	功能划分和界面设计	104
5.3.3	主要功能模块的代码	109
5.3.4	程序运行效果	116
5.4	小结	118
案例6	贪吃蛇	119
6.1	功能要求	119
6.2	相关知识介绍	119

6.2.1	System. Threading. Timer	119
6.2.2	System. Web. UI. Timer	119
6.2.3	System. Timers. Timer	120
6.2.4	System. Windows. Forms. Timer	120
6.3	程序设计	121
6.3.1	需求分析	121
6.3.2	概要设计	121
6.3.3	详细设计	122
6.3.4	主要功能模块的代码	123
6.3.5	测试运行	129
6.4	小结	130
案例7	学生成绩管理系统	131
7.1	功能要求	131
7.2	相关知识介绍	131
7.2.1	ADO. NET 结构	131
7.2.2	SQL Server 数据提供程序	132
7.3	程序设计	136
7.3.1	需求分析	136
7.3.2	系统设计	136
7.3.3	主要功能模块的代码	144
7.3.4	程序运行效果	154
7.4	小结	157
案例8	简易聊天室的实现	158
8.1	功能要求	158
8.2	相关知识介绍	158
8.2.1	网络编程基础知识	158
8.2.2	Socket 编程	160
8.3	程序设计	163
8.3.1	需求分析	163
8.3.2	功能划分和界面设计	163
8.3.3	主要功能模块的代码	165
8.3.4	程序运行效果	172
8.4	小结	174
参考文献		175

案

例 1

文本编辑器

设计目的

- 熟练掌握 .NET 基本结构和 C# 语言的基本知识
- 掌握面向对象程序设计的基本思路和方法
- 掌握菜单、工具栏以及常用控件的使用
- 掌握 RichTextBox 控件的使用
- 掌握通用对话框的使用

1.1 功能要求

“记事本”是一个基本的文本编辑器，它可以用于编辑简单的文档或创建网页文件。本案例设计实现一个具有记事本基本功能的文本编辑器，实现对文本文件和 rtf 格式文件进行创建、打开、保存、另存等操作，并能对选中的文本简单地进行字体设置，实现对文本串进行查找、替换操作，以及打印预览和打印的功能。

1.2 相关知识介绍

1.2.1 RichTextBox 控件

.NET 提供的 TextBox 控件和 RichTextBox 控件都可以作为文本编辑控件。而 RichTextBox 控件允许用户输入和编辑文本的同时还提供了比普通 TextBox 控件更高级的格式特征。RichTextBox 控件提供了数个有用的特征，可以在控件中安排文本的格式，对选中的文本进行格式设置。使用 RichTextBox 控件可以打开和保存 RTF 文件或普通的 ASCII 文本文件，支持剪贴板和 OLE 对象的 OLE 拖放功能。RichTextBox 控件支持几乎所有的 TextBox 控件中的属性、事件和方法。常用的 RichTextBox 控件的属性和方法介绍如下：

WordWrap 属性：指示多行编辑控件是否自动换行。

ScrollBars 属性：定义控件滚动条的行为，RichTextBox 控件的滚动条行为有 6 种值设置，说明如表 1-1 所示。

表 1-1 ScrollBars 值及其说明

值	说 明
Both(默认值)	当文本超过控件的宽度或长度时,才显示水平滚动条或垂直滚动条或两个滚动条都显示
None	从不显示任何类型的滚动条
Horizontal	当文本超过控件的宽度时,显示水平滚动条(必须将 WordWrap 属性设置为 false,才会出现这种情况)
Vertical	当文本超过控件的高度时,显示垂直滚动条
ForcedHorizontal	当 WordWrap 属性设置为 false 时,显示水平滚动条。在文本未超过控件的宽度时,该滚动条显示为浅灰色
ForcedVertical	始终显示垂直滚动条。在文本未超过控件的长度时,该滚动条显示为浅灰色

Font 属性: 获取或设置在控件中显示文本时所使用的字体。

Lines 属性: 获取或设置文本框控件中的文本行。

SelectedText 属性: 获取或设置 RichTextBox 内选定的文本。

SelectionStart 属性: 获取或设置文本框中选定的文本起始点。

SelectionLength 属性: 获取或设置控件中选定的字符数。

SelectionFont 属性: 获取或设置当前选定文本或插入点的字体。

SelectionColor 属性: 获取或设置当前选定文本或插入点的文本颜色。

Find()方法: 在 RichTextBox 的内容内搜索文本。

GetLineFromCharIndex()方法: 从 RichTextBox 控件文本内的指定字符位置检索行号。

GetFirstCharIndexFromLine()方法: 检索给定行第一个字符的索引。

LoadFile()方法: 将文件的内容加载到 RichTextBox 控件中。

SaveFile()方法: 将 RichTextBox 的内容保存到文件。

Undo()方法: 撤销文本框中的上一个编辑操作。

Redo()方法: 重新应用控件中上次撤销的操作。

Cut()方法: 将文本框中的当前选定内容移动到【剪贴板】。

Copy()方法: 将文本框中的当前选定内容复制到【剪贴板】。

Paste()方法: 将【剪贴板】的内容粘贴到控件中。

SelectAll()方法: 选定文本框中的所有文本。

1.2.2 OpenFileDialog /SaveFileDialog 对话框

OpenFileDialog 是打开文件对话框,允许用户选择打开一个或多个文件。SaveFileDialog 为保存文件对话框,它允许用户指定用于保存文件的选项。这两个类的使用方法基本相似,主要的属性和方法说明如下:

Filter 属性: 获取或设置当前文件名筛选器字符串,该字符串决定对话框的“另存为文件类型”或“文件类型”框中出现的选择内容。

FilterIndex 属性: 获取或设置文件对话框中当前选定筛选器的索引。

FileName 属性: 获取或设置一个包含在文件对话框中选定的文件名字符串。

MultiSelect 属性：获取或设置一个值，该值指示对话框是否允许选择多个文件。

ShowDialog()方法：运行通用对话框。

OpenFile()方法：打开用户选定的具有只读权限的文件。

1.2.3 PrintPreviewDialog 对话框

Document 属性：获取或设置要预览的文档。

Margin 属性：获取或设置控件的边距。

Location 属性：获取或设置该控件的左上角相对于其容器的左上角的坐标。

Left 属性：获取或设置控件左边缘与其容器的工作区左边缘之间的距离。

ShowDialog()方法：将窗体显示为模式对话框。

1.2.4 PrintDialog 对话框

Document 属性：获取或设置一个值，指示用于获取 PrinterSettings 的 PrintDocument。

AllowCurrentPage 属性：获取或设置一个值，该值指示是否显示【当前页】选项按钮。

AllowPrintToFile 属性：获取或设置一个值，该值指示是否启用【打印到文件】复选框。

PrinterSettings 属性：获取或设置对话框修改的打印机设置。

Reset()方法：将所有选项、最后选定的打印机和页面设置重新设置为其默认值。

ShowDialog()方法：用默认的所有者运行通用对话框。

1.2.5 PrintDocument 对话框

Print()方法：开始文档的打印进程。

BeginPrint 事件：在调用 Print 方法时并且在打印文档的第一页之前发生。

EndPrint 事件：打印完文档的最后一页时发生。

PrintPage 事件：当需要为当前页打印的输出时发生。

1.2.6 PageSetupDialog 对话框

Document 属性：获取或设置一个值，指示从中获取页面设置的 PrintDocument。

PageSettings 属性：获取或设置一个值，指示要修改的页设置。

PrinterSettings 属性：获取或设置用户单击对话框中【打印机】按钮时修改的打印机设置。

Reset()方法：将所有选项重新设置为其默认值。

ShowDialog()方法：运行通用对话框。

1.2.7 FontDialog 对话框

Font 属性：获取或设置选定的字体。

MaxSize 属性：获取或设置用户可选择的最小磅值。

MinSize 属性：获取或设置用户可选择的最大磅值。

ShowDialog()方法：运行通用对话框。

1.2.8 ColorDialog 对话框

Color 属性：获取或设置用户选定的颜色。

Reset()方法：将所有选项重新设置为其默认值，将最后选定的颜色重新设置为黑色，将自定义颜色重新设置为其默认值。

ShowDialog()方法：用默认的所有者运行通用对话框。

1.3 程序设计

根据人机交互的需要，设计以菜单触发事件的应用程序，对整个界面进行设计，依次将各功能进行划分，编写代码。

1.3.1 用户界面设计

为了掌握记事本的基本功能，本程序主要包括 2 个窗体，一个是主窗体，一个是查找替换窗体，下面对这两个窗体进行界面设计。

1. 主窗体界面设计

窗体运行的界面图如图 1-1 所示，从窗体界面的菜单和工具栏上执行相关操作，可以对文件执行打开、保存、另存为等操作。对已经打开的文件，可以对选中文字的字体进行设置，如复制、剪切、粘贴等操作。向主窗体中添加的控件以及对这些控件的设置说明如表 1-2 所示。

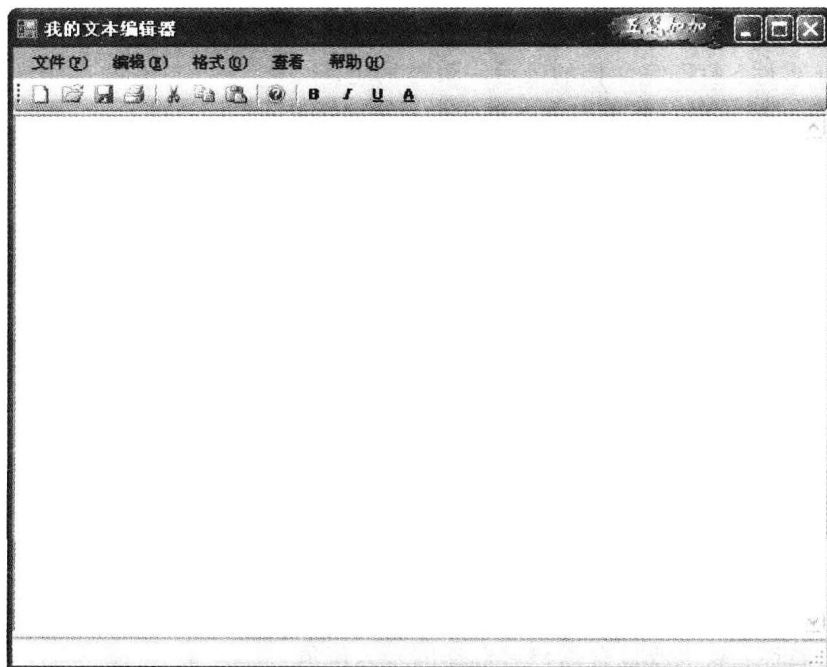


图 1-1 文本编辑器运行主界面

表 1-2 主窗体中控件对象属性表

控件类	属性	设 置 值	控 件 说 明
MenuStrip	Name	MenuStrip1	用于创建主菜单
	Items	见表 1-3~表 1-5 的设置	用于设置菜单项
ToolStrip	Name	ToolStrip1	用于创建工具栏
	Items	见表 1-6 的设置	用于设置工具栏按钮
OpenFileDialog	Name	OpenFileDialog1	【打开】对话框
	Filter	文本文件(*.txt) *.txt 所有文件(*.*) *.*	打开文件的类型设置
SaveFileDialog	Name	SaveFileDialog1	【打开文件】对话框
StatusStrip	Name	StatusStrip1	状态栏
	Items	StatusLabel	显示状态栏信息
FontDialog	Name	FontDialog1	打开【设置字体】对话框
PrintDocument	Name	PrintDocument1	显示给用户的文档
PageSetupDialog	Name	PageSetupDialog1	【页面设置】对话框
PrintPreviewDialog	Name	PrintPreviewDialog1	【页面预览】对话框
PrintDialog	Name	PrintDialog1	【打印】对话框
ColorDialog	Name	ColorDialog1	打开【颜色设置】对话框
ContextMenuStrip	Name	ContextMenuStrip1	弹出式对话框
	Items	见表 1-7 的设置	

表 1-3 【文件】菜单下的子菜单项

主菜单项	Text 属性	子 菜 单 项	Text 属性
File_ToolStripMenuItem	文件(&F)	File_MenuItemNew	新建(&N)
		File_MenuItemOpen	打开(&O)
		File_MenuItemSave	保存(&S)
		File_MenuItemSaveAs	另存为(&A)
		File_MenuItemPageSetup	页面设置(&U)
		File_MenuItemPrintPreview	打印预览(&V)
		File_MenuItemPrint	打印(&P)
		File_MenuItemExit	退出(&X)

表 1-4 【编辑】菜单下的子菜单项

主菜单项	Text 属性	子 菜 单 项	Text 属性
Edit_ToolStripMenuItem	编辑(&E)	Edit_MenuItemUndo	撤销(&U)
		Edit_MenuItemRedo	重复(&R)
		Edit_MenuItemCut	剪切(&T)
		Edit_MenuItemCopy	复制(&C)
		Edit_MenuItemPaste	粘贴(&P)
		Edit_MenuItemDelete	删除(&D)
		Edit_MenuItemFind	查找(&F)
		Edit_MenuItemReplace	替换(&E)
		Edit_MenuItemSelectAll	全选(&L)

表 1-5 【格式】【查看】【帮助】菜单下的子菜单项

主菜单项	Text 属性	子 菜 单 项	Text 属性
Format_MenuItem	格式(&O)	Format_MenuItemFont	字体...
		Format_MenuItemWrapLine	自动换行
Check_MenuItem	查看(&V)	Check_MenuItemStatus	状态栏
Help_MenuItem	帮助(&H)	Help_MenuItemTheme	帮助主题

表 1-6 工具栏 ToolStrip1 的 Items 中添加项的说明

添加项类型	名 称	ToolTip Text 属性	功 能 说 明	Image
Button	NewToolStripButton	新建(N)	新建文件	
Button	OpenToolStripButton	打开(O)	打开已有文件	
Button	SaveToolStripButton	保存(S)	保存打开的文件	
Button	PrintToolStripButton	打印(P)	打印打开的文件	
Button	CutToolStripButton	剪切(U)	将选择的文字剪切到剪贴板中	
Button	CopyToolStripButton	复制(C)	将选择的文字复制到剪贴板	
Button	PasteToolStripButton	粘贴(P)	将剪贴板中的文字粘贴在插入点	
Button	HelpToolStripButton	帮助(L)	打开帮助文档	
Button	BoldToolStripButton	B	将选中的文字加粗显示	
Button	ItalicToolStripButton	I	将选中的文字倾斜显示	
Button	UnderlineToolStripButton	U	将选中的文字加下划线	
Button	FontColorToolStripButton	A	对选中的文字设置颜色	

添加 MenuItem, 各个 MenuItem 设置如表 1-7 所示。

表 1-7 弹出式菜单 ContextMenuStrip1 的 Items 设置

名 称	Text 属性	功 能 说 明
CutToolStripMenuItem	剪切(&T)	将选择的文字剪切到剪贴板, 功能与菜单及工具栏上的剪切功能相同
CopyToolStripMenuItem	复制(C)	将选择的文字复制到剪贴板
PasteToolStripMenuItem	粘贴(&P)	将剪贴板的数据插入到当前插入点
DelToolStripMenuItem	删除(&D)	删除选择的文字

2. 【查找替换】窗体的界面设计

【查找替换】窗体的界面如图 1-2 所示。此窗体用于对打开的文本内容进行查找或替换, 界面中添加两个标签, 两个文本框和两个按钮。

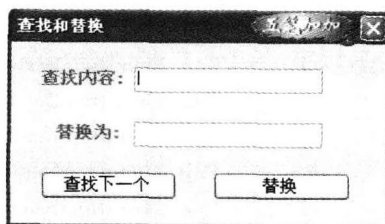


图 1-2 【查找替换】窗体界面图

各个控件属性的设置如表 1-8 所示。

表 1-8 【查找替换】窗体属性及各控件属性设置

对象	属 性	设 置 值	功 能 说 明
Form	Name	FormFindReplace	窗体名称
	Caption	查找和替换	窗体的标题
Label	Name	Label1	标签名称
	Caption	查找和替换	标签标题
Label	Name	Label2	标签名称
	Caption	替换为	标签标题
TextBox	Name	TextBox1	输入查找的文本串
TextBox	Name	TextBox2	输入替换的文本串
CommandButton	Name	BtnSearch	查找下一个和文本框 TextBox1 中的文本相同的文字
	Caption	查找下一个	
CommandButton	Name	BtnReplace	将打开的文件中具有和 TextBox1 中相同的文本替换成 TextBox2 中的文本
	Caption	替换	

1.3.2 主要功能模块的代码

1. 主窗体中代码的实现

主窗体定义的成员变量如下：

```
string s_FileName = "";           //打开的文件名
int FindPostion = 0;             //查找的位置
bool bSave = false;              //文件是否已经保存
StringReader streamToPrint = null;
Font printFont;
```

程序运行过程中由相应事件触发执行相关的动作。程序代码主要是由事件过程和自定义的方法组成，下面依次对这些方法加以介绍。

1) IfSaveOldFile()方法

这个方法是个自定义方法，用于判断文件是否需要保存。如果一个打开的文件在关闭之前有修改，程序提示是否保存，然后进入下一步操作。代码如下：

```
public bool IfSaveOldFile()
{
    bool ReturnValue = true;
    if (bSave)
    {
        System.Windows.Forms.DialogResult dr;
        string[] str = this.Text.Split('-'); //用特定的字符来划分字符串
        dr = MessageBox.Show("文件" + str[0] + " 文字已经改变.\n 想保存文件吗?", "记事本", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Question);
        switch (dr)
        {
```

```

        case System.Windows.Forms.DialogResult.Yes:
            bSave = false;
            if (s_FileName.Length != 0)
            {
                richTextBox1.SaveFile(s_FileName,
                    RichTextBoxStreamType.PlainText);
            }
            else
            {
                SaveFileDialog saveFileDialog1 = new SaveFileDialog();
                if (saveFileDialog1.ShowDialog() == DialogResult.OK)
                {
                    s_FileName = saveFileDialog1.FileName;
                    richTextBox1.SaveFile(s_FileName,
                        RichTextBoxStreamType.PlainText);
                }
            }
            ReturnValue = true;
            break;
        case System.Windows.Forms.DialogResult.No:
            bSave = false;
            ReturnValue = true;
            break;
        case System.Windows.Forms.DialogResult.Cancel:
            ReturnValue = false;
            break;
    }
}
return ReturnValue;
}

```

2) FindRichTextBoxString()方法

FindRichTextBoxString()方法用于在 RichTextBox 中查找是否和给定文本串相同的文本串。如果找到,则将此文本串在 RichTextBox 中高亮度显示。代码如下:

```

public void FindRichTextBoxString(string FindString)
{
    if (FindPostion >= richTextBox1.Text.Length)
    {
        MessageBox.Show("已到文本底部,再次查找将从文本开始出查找", "提示",
            MessageBoxButtons.OK);
        FindPostion = 0;
        return;
    }
    FindPostion = richTextBox1.Find(FindString, FindPostion,
        RichTextBoxFinds.MatchCase);
    if (FindPostion == -1)
    {
        MessageBox.Show("未找到匹配字符串,再次查找将从文本开始处查找", "提示",
            MessageBoxButtons.OK);
        FindPostion = 0;
    }
}

```



```
    }  
    else  
    {  
        richTextBox1.Focus();  
        FindPostion += FindString.Length;  
    }  
}
```

3) ReplaceRichTextBoxString()方法

ReplaceRichTextBoxString()方法用于文本串的替换操作,使用参数的信息替换 richTextBox 中选中的文本。代码如下:

```
public void ReplaceRichTextBoxString(string ReplaceString)  
{  
    if (richTextBox1.SelectedText.Length != 0)  
        richTextBox1.SelectedText = ReplaceString;  
}
```

4) WrapLine()方法

WrapLine()方法用于设置自动换行。当用户在 RichTextBox 中输入文本时,如果超过一行,不需要手动换行,程序调用 WrapLine()方法实现自动换行。代码如下:

```
public void WrapLine()  
{  
    //使用 richtextbox 的自动换行  
    this.richTextBox1.WordWrap = !(this.richTextBox1.WordWrap);  
}
```

5) Cleation()方法

使用 Windows 系统中的 NotePad 应用程序时,如果编辑时支持自动换行,则状态栏不可用。当不支持自动换行时,状态栏可用。Cleation()方法给出自动换行和状态栏之间联系。代码如下:

```
public void Cleation()  
{  
    //若不选中自动换行,则状态栏可用  
    if (Format_MenuItem mWrapLine.Checked == false)  
    {  
        Check_MenuItemStatus.Enabled = true;  
    }  
    else  
    {  
        Check_MenuItemStatus.Enabled = false;  
    }  
}
```

6) 事件过程

文本编辑器中主要的事件是由 RichTextBox、菜单、工具栏按钮的相关操作形成的,下面分别加以描述。