**数据结构课程设计**

**题 目: 文章编辑**

**班 级：**

**学 生 姓 名：**

**学 生 学 号：**

**2022 年 2月 24日**

目录

[1. 开发环境 3](#_Toc477208932)

[2. 程序的功能 3](#_Toc477208933)

[3. 代码清单 3](#_Toc477208934)

[4. 运行界面 14](#_Toc477208935)

## 开发环境

C# WPF，.NET Framework 4.8，VS2022

**郑重声明：**本程序是在之前代码上的升级改造，**绝对不存在**炒冷饭行为！！！

## 功能描述（内容仅供参考）

本程序模仿了经典的系统实用程序——记事本，提供具有打字，保存，查看，统计功能的图形用户界面，支持实时预览统计信息，并且在文件保存时提供自动的提示信息，提醒用户当心误操作。

此次项目是在作业3的基础上扩充完善而成的，主要的变化如下：

1.用户界面由固定式布局改为响应式布局，可以适配窗口大小的调整和变化，而不会出现放大窗口后出现大面积留白的现象。

2.用户输入文字，窗口下方状态栏实时显示空格，英文字母，数字，其他，已及总字数，也可以点击统计按钮查看

3.提供就地保存和另存为功能

4.当用户修改内容后忘记保存尝试关闭窗口时，会阻止用户操作，要求用户选择是否保存更改。

本程序用到的数据结构和算法：

数据结构：本程序将文章分为若干行，每行保存至一个string（本质上就是个线性表）中，各行之间按线性表组织

在查找单词的过程中，采用了经典的KMP模式匹配算法，可以有效提高查找效率

## 代码清单

|  |
| --- |
| Service\SearchService.cs |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace 作业6\_单词统计小程序升级版.service  {  public class SearchService  {  public static readonly int MAX\_LINE = 1000;  private int[] next = new int[MAX\_LINE];  //每一行开启模式匹配之前都来设置一下这next数组  private void setNext(string source, int[] next)  {  next[0] = -1;  int n=source.Length;  int j = -1;  for (int i = 1; i < n; i++)  {  while (j != -1 && source[i] != source[j + 1])  {  j = next[j];  }  if(source[i] == source[j + 1])  {  j++;  }  next[i] = j;  }  }  //统计一行出现模式串的位置及次数  //返回每一行出现的位置集合  public List<int> kmpSearch(string text, string pattern, out int cnt)//这样其实有点不太好，但为了一次性完成任务，就先忍忍啦  {  setNext(text, this.next);  int countLine = 0;  List<int> list = new List<int>();  int n = text.Length, m = pattern.Length;  int j = -1;  for (int i = 0; i < n; i++)  {  while (j != -1 && text[i] != pattern[j + 1])  {  j = next[j];  }  if(text[i]==pattern[j + 1])  {  j++;  }  if (j == m - 1)  {  int index = i - m + 1;  list.Add(index);  countLine++;  j = next[j];//回去，等着下一次匹配  }  }  cnt = countLine;  return list;  }    //考虑所有的行  public List<List<int>> searchAll(string[] lines,string word, out int total)  {  List<List<int>> list = new List<List<int>>();  int cnt = 0;  for(int i = 0; i < lines.Length; i++)  {  var ans = kmpSearch(lines[i], word, out int curcnt);  cnt += curcnt;  list.Add(ans);  }  total = cnt;  return list;  }  }  } |

|  |
| --- |
| Service\CountService.cs |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace 作业6\_单词统计小程序升级版.service  {  public class CountService  {  public List<Int32> getSymbolCounts(string[] lines)  {  int space = 0, letter = 0, others = 0, digits = 0;  foreach (string line in lines)  {  //对每一行  for (int i = 0; i < line.Length; i++)  {  if (line[i] == ' ' || line[i] == '\t')  {  space++;  }  else if (('a' <= line[i] && line[i] <= 'z') || ('A' <= line[i] && line[i] <= 'Z'))  {  letter++;  }  else if('0' <= line[i] && line[i] <= '9')  {  digits++;  }  else  {  others++;  }  }  }  List<Int32> list = new List<Int32>();  list.Add(space);  list.Add(letter);  list.Add(others);  list.Add(digits);  return list;  }  }  } |

|  |
| --- |
| SearchWindow.xaml |
| <Window x:Class="作业6\_单词统计小程序升级版.SearchWindow"  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"  xmlns:local="clr-namespace:作业6\_单词统计小程序升级版"  mc:Ignorable="d"  Title="查找" Height="293" Width="394" WindowStyle="ToolWindow">  <Grid>  <Button x:Name="btnSearch" Content="查找" HorizontalAlignment="Left" Height="50" Margin="139,201,0,0" VerticalAlignment="Top" Width="110" Click="BtnSearch\_Click"/>  <TextBox x:Name="txbPattern" HorizontalAlignment="Left" AcceptsTab="True" Height="29" Margin="137,39,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="224"/>  <Label Content="要查的文本：" HorizontalAlignment="Left" Height="29" Margin="10,39,0,0" VerticalAlignment="Top" Width="122"/>  <GroupBox HorizontalAlignment="Left" Height="123" Header="结果" Margin="10,73,0,0" VerticalAlignment="Top" Width="351">  <ScrollViewer>  <TextBox x:Name="txbResult" TextWrapping="Wrap" Text="" IsReadOnly="True" />  </ScrollViewer>  </GroupBox>  </Grid>  </Window> |

|  |
| --- |
| SearchWindow.xaml.cs |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows;  using System.Windows.Controls;  using System.Windows.Data;  using System.Windows.Documents;  using System.Windows.Input;  using System.Windows.Media;  using System.Windows.Media.Imaging;  using System.Windows.Shapes;  using 作业6\_单词统计小程序升级版.service;  namespace 作业6\_单词统计小程序升级版  {  /// <summary>  /// SearchWindow.xaml 的交互逻辑  /// </summary>  public partial class SearchWindow : Window  {  public SearchWindow()  {  InitializeComponent();  }  private void BtnSearch\_Click(object sender, RoutedEventArgs e)  {  SearchService service = new SearchService();  String article=(App.Current.MainWindow as MainWindow).txbArticle.Text;  string[] vs = article.Split('\n');  var ans=service.searchAll(vs, txbPattern.Text.Trim(), out int total);  txbResult.Text = "";//清除上一次结果  StringBuilder sb=new StringBuilder();  sb.Append("总次数：" + total + "\n");  int index = 1;  foreach(var x in ans)  {  if (x != null && x.Count != 0)  {  sb.Append("在第" + index + "行共出现" + x.Count + "次：\n");  sb.Append("分别是第");  foreach (var y in x)  {  sb.Append(y + 1 + " ");//数组是从0开始数的，为了符合人们的阅读习惯，因此输出从1开始  }  sb.Append("列\n");  }  index++;  }  txbResult.Text = sb.ToString();  }  }  } |

|  |
| --- |
| MainWindow.xaml |
| <Window x:Class="作业6\_单词统计小程序升级版.MainWindow"  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"  xmlns:local="clr-namespace:作业6\_单词统计小程序升级版"  mc:Ignorable="d"  Title="较为完备的记事本（响应式布局+字数统计+防误操作提示）" Height="450" Width="800" Closing="Window\_Closing">  <Grid>  <Grid.RowDefinitions>  <RowDefinition Height="2.2\*"/>  <RowDefinition Height="18\*"/>  <RowDefinition Height="1\*"/>  </Grid.RowDefinitions>  <DockPanel Grid.Row="0" Margin="5,5,5,5" Background="AliceBlue">  <Button x:Name="btnOpen" Content="打开" Width="60" Margin="5,5,5,5" Click="btnOpen\_Click"/>  <Button x:Name="btnSaveNotRename" Content="存储" Width="60" Margin="5,5,5,5" Click="btnSaveNotRename\_Click"/>  <Button x:Name="btnSave" Content="另存为" Width="60" Margin="5,5,5,5" Click="btnSave\_Click"/>  <Button x:Name="btnSearch" Content="查找" Width="60" Margin="5,5,5,5" Click="BtnSearch\_Click"/>  <Button x:Name="btnCount" Content="字数统计" Width="60" Margin="5,5,5,5" Click="btnCount\_Click"/>  <Border BorderBrush="Black" BorderThickness="1" >  <Label Content="说明：暂时仅支持UTF - 8编码的纯文本文件！" HorizontalAlignment="Center" VerticalAlignment="Center"/>  </Border>  </DockPanel>    <DockPanel Grid.Row="1" LastChildFill="True" Margin="5,5,5,5">  <ScrollViewer DockPanel.Dock="Top" HorizontalAlignment="Stretch" Margin="0,0,0,0" VerticalAlignment="Stretch" >  <TextBox x:Name="txbArticle" x:FieldModifier="public" TextWrapping="Wrap" AcceptsReturn="True" AcceptsTab="True" Text="" TextChanged="txbArticle\_TextChanged"/>  </ScrollViewer>  </DockPanel>  <DockPanel Grid.Row="2">  <TextBlock DockPanel.Dock="Left" Text=" 空格数："/>  <TextBlock x:Name="txbSpaces" DockPanel.Dock="Left" Text="0"/>  <TextBlock DockPanel.Dock="Left" Text=" 英文字母数："/>  <TextBlock x:Name="txbLetters" DockPanel.Dock="Left" Text="0"/>  <TextBlock DockPanel.Dock="Left" Text=" 数字总数："/>  <TextBlock x:Name="txbDigits" DockPanel.Dock="Left" Text="0"/>  <TextBlock DockPanel.Dock="Left" Text=" 其他字符总数："/>  <TextBlock x:Name="txbOthers" DockPanel.Dock="Left" Text="0"/>  <TextBlock DockPanel.Dock="Left" Text=" 文章总字数："/>  <TextBlock x:Name="txbAll" DockPanel.Dock="Left" Text="0"/>  <TextBlock DockPanel.Dock="Left"/>  </DockPanel>  </Grid>  </Window> |

|  |
| --- |
| MainWindow.xaml.cs |
| using Microsoft.Win32;  using System;  using System.Collections.Generic;  using System.IO;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows;  using System.Windows.Controls;  using System.Windows.Data;  using System.Windows.Documents;  using System.Windows.Input;  using System.Windows.Media;  using System.Windows.Media.Imaging;  using System.Windows.Navigation;  using System.Windows.Shapes;  using 作业6\_单词统计小程序升级版.service;  namespace 作业6\_单词统计小程序升级版  {  /// <summary>  /// MainWindow.xaml 的交互逻辑  /// </summary>  public partial class MainWindow : Window  {  private string fileName;  private bool issaved = true;  CountService service = new CountService();  public MainWindow()  {  InitializeComponent();  }  private void saveByteData()  {  try  {  File.WriteAllText(fileName, txbArticle.Text, Encoding.UTF8);  issaved = true;  }  catch (Exception ex)  {  MessageBox.Show(ex.Message);  }  }  private void BtnSearch\_Click(object sender, RoutedEventArgs e)  {  SearchWindow searchWindow = new SearchWindow();  searchWindow.Owner = this;  searchWindow.Show();  }  private void btnOpen\_Click(object sender, RoutedEventArgs e)  {  var openDialog = new OpenFileDialog  {  InitialDirectory = Environment.CurrentDirectory,  Filter = "文本文件(\*.txt)|\*.txt"  };  if (openDialog.ShowDialog() == true)  {  fileName = openDialog.FileName;  txbArticle.Text = File.ReadAllText(fileName, Encoding.UTF8);  }  }  private void btnSave\_Click(object sender, RoutedEventArgs e)  {  var saveDialog = new SaveFileDialog  {  InitialDirectory = Environment.CurrentDirectory,  Filter = "文本文件(\*.txt)|\*.txt",  FileName = fileName  };  if (saveDialog.ShowDialog() == true)  {  fileName = saveDialog.FileName;  saveByteData();  }  }  private void btnSaveNotRename\_Click(object sender, RoutedEventArgs e)  {  if(fileName == null || fileName.Equals(""))  {  btnSave\_Click(sender, e);  }  try  {  File.WriteAllText(fileName, txbArticle.Text, Encoding.UTF8);  issaved = true;  }  catch (Exception ex)  {  MessageBox.Show(ex.Message);  }  }  private void txbArticle\_TextChanged(object sender, TextChangedEventArgs e)  {  issaved = false;  var article = txbArticle.Text;  var collections = article.Split('\n');  List<Int32>list = service.getSymbolCounts(collections);  txbSpaces.Text = list[0].ToString();  txbLetters.Text = list[1].ToString();  txbDigits.Text = list[3].ToString();  txbOthers.Text = list[2].ToString();  txbAll.Text = article.Count().ToString();  }  private void Window\_Closing(object sender, System.ComponentModel.CancelEventArgs e)  {  if(!issaved)  {  var choice = MessageBox.Show("有未保存的内容，是否在关闭程序前保存" + ((fileName == null || fileName.Equals("")) ? "" : fileName) + "?", "系统提示", MessageBoxButton.YesNoCancel, MessageBoxImage.Question);  if (choice == MessageBoxResult.Yes)  {  if (fileName == null || fileName.Equals(""))  {  var saveDialog = new SaveFileDialog  {  InitialDirectory = Environment.CurrentDirectory,  Filter = "文本文件(\*.txt)|\*.txt",  FileName = fileName  };  if (saveDialog.ShowDialog() == true)  {  fileName = saveDialog.FileName;  saveByteData();  }  e.Cancel = true;  }  else  {  saveByteData();  }    }  else if(choice == MessageBoxResult.Cancel)  {  e.Cancel = true;  }  }  }  private void btnCount\_Click(object sender, RoutedEventArgs e)  {  var article = txbArticle.Text;  var collections = article.Split('\n');  List<Int32> list = service.getSymbolCounts(collections);  StringBuilder sb = new StringBuilder();  sb.AppendLine("空格数：" + list[0]);  sb.AppendLine("英文字母数：" + list[1]);  sb.AppendLine("数字个数：" + list[3]);  sb.AppendLine("其他字符数：" + list[2]);  sb.AppendLine("文章总字数" + article.Length);  MessageBox.Show(sb.ToString(), "统计结果", MessageBoxButton.OK, MessageBoxImage.Information);  }  }  } |

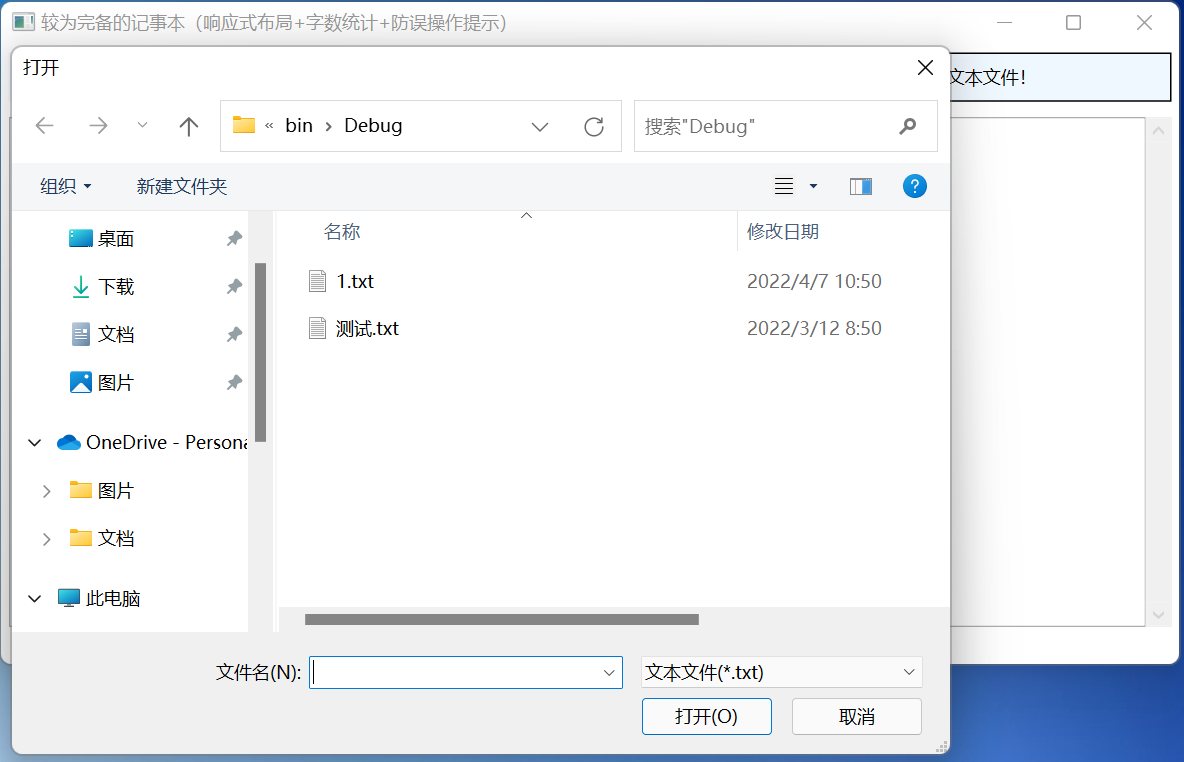
## 运行界面（通过图例展示操作流程）

主界面：

图形用户界面, 文本, 应用程序

描述已自动生成

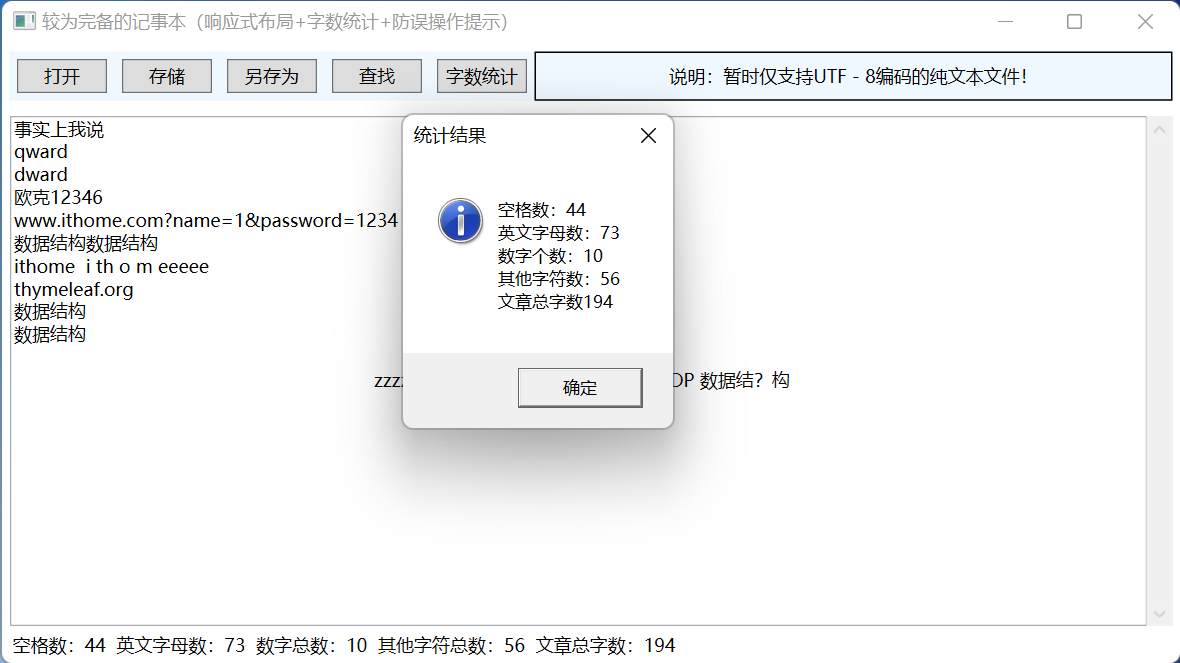
打开并编辑文档



图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

字数统计：

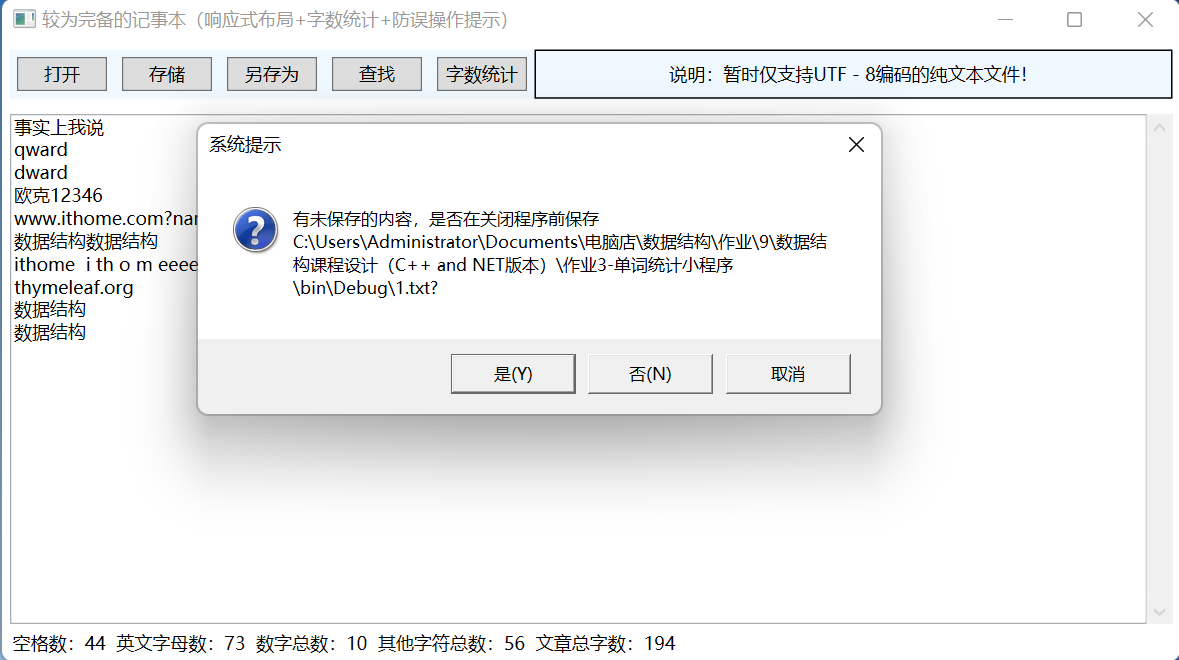


查找：

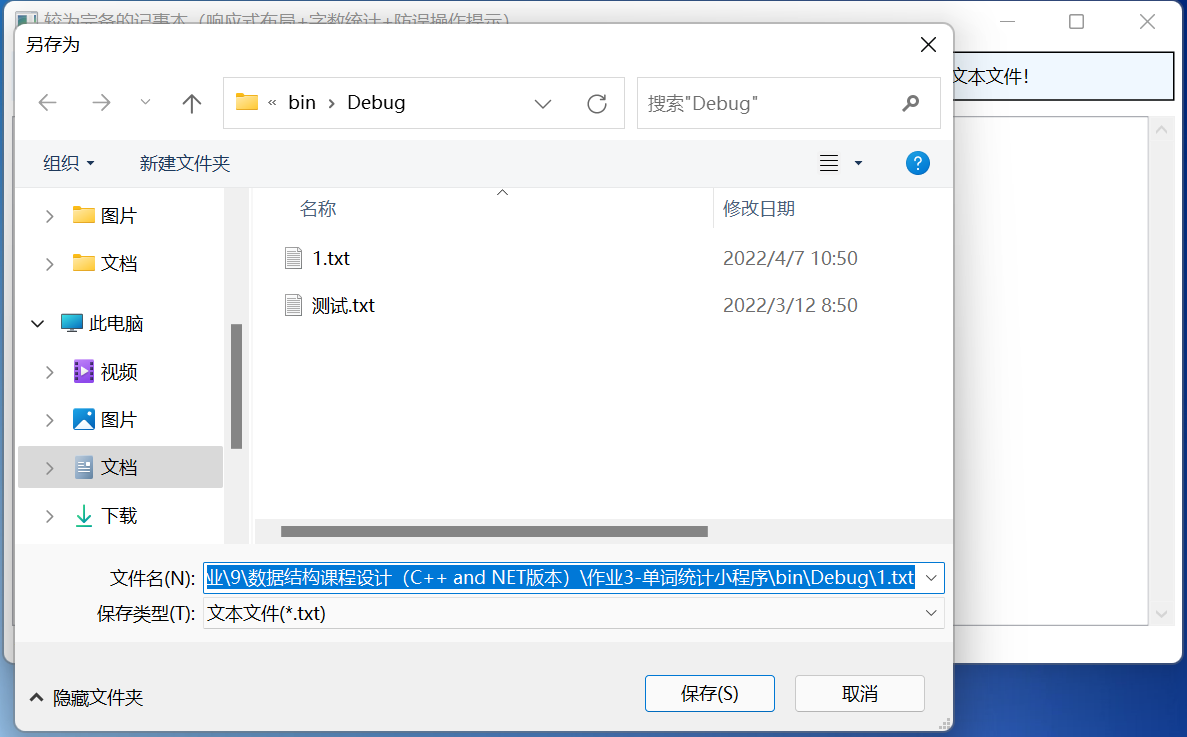
图形用户界面

描述已自动生成

如果不小心关了还会提醒你：



点击另存为的效果：



响应式布局（以最大化为例）

