**数据结构课程设计**

**题 目: 单词检索统计程序**

**班 级：**

**学 生 姓 名：**

**学 生 学 号：**

**2022 年 3月 12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

Visual Studio 2022, C# WPF, .NET Framework 4.8

## 功能描述

串是非数值处理中的主要对象，如在信息检索、文本编辑、符号处理等许多领域，得到越来越广泛的应用。在串的基本操作中，在主串中查找模式串的模式匹配算法是文本处理中最常用、最重要的操作之一，称为模式匹配或串匹配，就是求子串在主串中首次出现的位置。朴素模式匹配算法的基本思路是将给定字串与主串从第一个字符开始比较，找到首次与子串完全匹配的子串为止，并记住该位置。但为了实现统计子串出现的个数，不仅需要从主串的第一个字符位置开始比较，而且需要从主串的任一位置检索匹配字符串。

这是一个具有查找统计功能的简单记事本程序，可以简单的编辑文本文件，打开已有文件和将文件存盘。查找功能可以针对输入的字符串（包括空格和tab键，但不包括换行符），求出出现的总次数，在出现的每一行的行号以及在改行的出现次数，在该行出现的所有列的列号。

本项目的设计要求可以分为三个部分实现：其一，建立一个文本文件，文件名由用户浏览选择；其二，给定单词计数，输入一个不含换行符的单词，统计输出该单词在文本中的出现次数；其三，检索给定单词，输入一个单词，检索并输出该单词所在的行号、该行中出现的次数以及在该行中的相应位置

## 代码清单

查找部分核心算法（KMP模式匹配法）：

|  |
| --- |
| service\Service.cs |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace 作业3\_单词统计小程序.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;  }  }  } |

查找对话框及其交互逻辑

|  |
| --- |
| SearchWindow.xaml |
| <Window x:Class="作业3\_单词统计小程序.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3\_单词统计小程序"  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="" Width="316" IsReadOnly="True" Height="95" />  </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3\_单词统计小程序.service;  namespace 作业3\_单词统计小程序  {  /// <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();  }  }  } |

主窗口及其交互逻辑

|  |
| --- |
| MainWindow.xaml |
| <Window x:Class="作业3\_单词统计小程序.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3\_单词统计小程序"  mc:Ignorable="d"  Title="简易记事本" Height="450" Width="800">  <Grid>  <Border BorderBrush="Black" BorderThickness="1" HorizontalAlignment="Center" Height="327" Margin="0,76,0,0" VerticalAlignment="Top" Width="780">  <ScrollViewer HorizontalAlignment="Center" Height="309" Margin="0,9,0,0" VerticalAlignment="Top" Width="760">  <TextBox x:Name="txbArticle" x:FieldModifier="public" TextWrapping="Wrap" AcceptsReturn="True" AcceptsTab="True" Text="" Width="760"/>  </ScrollViewer>  </Border>  <WrapPanel HorizontalAlignment="Center" Height="61" Margin="0,10,0,0" VerticalAlignment="Top" Width="780">  <Button x:Name="btnOpen" Content="打开" Height="51" Width="110" Margin="5,5,5,5" Click="btnOpen\_Click"/>  <Button x:Name="btnSave" Content="另存为" Height="51" Width="110" Margin="5,5,5,5" Click="btnSave\_Click"/>  <Button x:Name="btnSearch" Content="查找" Height="51" Width="110" Margin="5,5,5,5" Click="BtnSearch\_Click"/>  <Border BorderBrush="Black" BorderThickness="1" Width="420">  <Label Content="说明：暂时仅支持UTF - 8编码的纯文本文件！" HorizontalAlignment="Center" VerticalAlignment="Center"/>  </Border>  </WrapPanel>  </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;  namespace 作业3\_单词统计小程序  {  /// <summary>  /// MainWindow.xaml 的交互逻辑  /// </summary>  public partial class MainWindow : Window  {  private string fileName;  public MainWindow()  {  InitializeComponent();  }  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;  try  {  File.WriteAllText(fileName, txbArticle.Text, Encoding.UTF8);  }  catch (Exception ex)  {  MessageBox.Show(ex.Message);  }  }  }  }  } |

## 运行界面（通过图例展示操作流程）

打开程序，在编辑框里面输入一些文本

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

随便，这只是个例子

点击查找，弹出如图所示的对话框

图形用户界面, 应用程序, 电子邮件

描述已自动生成

输入一些内容，点击查找，下方文本框内将显示结果

图形用户界面, 文本, 电子邮件

描述已自动生成

图形用户界面

描述已自动生成

另一组输入，也是类似效果，如果查不到，直接显示0

图形用户界面

描述已自动生成

点击打开和另存为还可以持久化数据

社交网络的手机截图

描述已自动生成