Program Cover Sheet

|  |
| --- |
| Name: Austin Rippee |
| Assignment: Assignment 3 |
| List any parts of the assignment that do not work/were not completed:  When adding values to the array of string, the array added the spaces as another “value” which added 1 more to the total unique words than necessary  Stars weren’t identical to what the program needed. I couldn’t get the max and min value to be counted as a double.  Lowest word utilization was continuously counted as 2 instead of 1 and I couldn’t find where the problem was occurring. Also with the lowest, I couldn’t print out every value from the array string, only the first value.  Finally, the average word utilization wasn’t working properly (again with the min and max not allowing it to be a double really screwed me over but this way it worked better than the last way I tried which offered a lot less) |

|  |
| --- |
| Instructor’s Comments: |
| Grade: |

Program Submission Requirements: (1) all files, zipped and uploaded to Canvas and (2) a completed cover sheet, program execution screenshots and source code printed, **stapled** and turned in during class. Failure to follow the submission requirements will result in points lost on that particular assignment.

imports System.IO

'------------------------------------------------------------

'- File Name : frmConsoleApp.frm -

'- Part of Project: Main -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: February 1st, 2022 -

'------------------------------------------------------------

'- File Purpose: -

'- This file contains the main application form where the -

'- user will input a path address in which locates a file

'- and does some processing in which it will count the amount

'- of words is in the file along with writing it out to a report

'------------------------------------------------------------

'- Program Purpose: -

'- -

'- Takes in a flile, reads each line and displays each unique

'- word and the amount of times that word appears then prints

'- it out to a report file where it can be displayed

'------------------------------------------------------------

'- Global Variable Dictionary (alphabetically): -

'- arrDistinctWordList - creates the array of the distinct words

'- arrFileWords - creates the array of all the words in the file

'- dblAvgNum - finds the average

'- intMax - max value occurrance

'- intMaxStars - max amount of stars for a line

'- intMinStarNum - finds min stars correlated with min value occurrance

'- intNumWords - accumulates number of distinct words there are

'- intOtherStarNum - finds the stars correlated with any other occurance other than max and min

'- regex - regular expressions syntax to split the test file

'- srReader - used as a streamreader

'- srSourcePath - reads in what the source path is

'- strConsoleReport - reads in the report file

'- strFile - sets all values in the file to lowercase

'- strFindMax - find values with max occurance

'- strFindMin - find values with min occurance

'- strNumStars - stores the strdup stars

'- strReadTestFile - reads in line by line from the source path

'- strRPTFilePath - sets the report file path as a string

'- strSourcePath - gets the source path of the user input

'- strTxtFileArr() - Reads in the text file as an array of strings

'- strTxtFileName - gets the file name the source path is pointing to

'- strUserInput - reads in what path the user wants to read the report from

'- swFileStream - creates a streamwriter to write to the report file

'------------------------------------------------------------

Module frmConsoleApp

Sub Main()

Console.Title = "Word Analysis Profiler Application" 'Changes program title

Console.BackgroundColor = ConsoleColor.White 'Sets background color to white

Console.Clear() 'Clears the console

Console.ForegroundColor = ConsoleColor.DarkBlue 'Sets font color to dark blue

Console.WriteLine("Please enter the path and name of the file to process: ") 'Intial line asking for the file

'Initializes the String Array in which the text file will be stored

Dim strTxtFileArr() As String

'Sets the source path as what the user enters

Dim strSourcePath As String = Console.ReadLine()

'Gets the file name of the source path the user entered

Dim strTxtFileName As String = System.IO.Path.GetFileName(strSourcePath)

If System.IO.File.Exists(strSourcePath) Then

Dim strReadTestFile As String

'Reads in line by line the file in which is in the source path

Using srReader As StreamReader = New StreamReader(strSourcePath)

strReadTestFile = srReader.ReadLine

End Using

'Reads in the entire test file

Dim srSourcePath As StreamReader = New StreamReader(strSourcePath)

Dim regex As System.Text.RegularExpressions.Regex

'converts it to all lowercase

Dim strFile As String = srSourcePath.ReadToEnd().ToLower

'Splits the file to a new line

strTxtFileArr = regex.Split(strFile, "\s")

' A for loop to sort the array of string in alphabetical order and removes any periods

For i = 0 To strTxtFileArr.Length - 1

If strTxtFileArr(i).Trim <> "" Then

'initializes temp string array to store the word values

Dim temp() As String = Split(strTxtFileArr(i))

'sorts by alphabetical order

strTxtFileArr.Sort(strTxtFileArr)

'removes periods from the words that have them

strTxtFileArr(i) = temp(0).Trim().Replace(".", "")

End If

Next

For i = 0 To strTxtFileArr.Length - 1

If strTxtFileArr(i).Trim <> "" Then

'initializes temp string array to store the word values

Dim temp() As String = Split(strTxtFileArr(i))

'removes commas from the words that have them

strTxtFileArr(i) = temp(0).Trim().Replace(",", "")

End If

Next

'creates an object that takes in the test file and groups it by distinct words and counts the amount of words

Dim arrFileWords = strTxtFileArr.GroupBy(Function(x) x).Select(Function(words) New With {words.Key, Key .Count = words.Count()})

'Creates an object that stores each distinct word

Dim arrDistinctWordList = arrFileWords.Select(Function(x) x.Key)

'Counts the amount of unique words in the distinct word list

Dim intNumWords As Integer

For Each item In arrDistinctWordList

intNumWords = arrFileWords.Count

Next

Console.WriteLine()

Console.WriteLine("Processing Completed...")

Console.WriteLine()

Console.WriteLine("Please enter the path and name of the report file to generate: ")

'Sets the report file path to what the user enters

Dim strRPTFilePath As String = Console.ReadLine()

'Creates a file stream that writes to the file

Dim swFileStream As System.IO.StreamWriter

'Checks if user has entered any data

If strRPTFilePath = "" Then

MsgBox("Sorry, you have entered a wrong path name. Please restart the program and try again.")

Else

'Opens the report file and is set to false so it doesn't append so it can be used over and over without confliction with another report file

swFileStream = My.Computer.FileSystem.OpenTextFileWriter(strRPTFilePath, False)

'sets title of report file

swFileStream.WriteLine(vbTab + vbTab + vbTab + "Word Analysis Statistics")

swFileStream.WriteLine()

'displays line of unique words

swFileStream.WriteLine("There were a total of " + CStr(intNumWords) + " unique words encountered")

swFileStream.WriteLine()

For Each words In arrFileWords

' Initializes strNumStars for use as a string

Dim strNumStars As String = ""

'Minimum amount of stars as an integer

Dim intMinStarNum As Integer

'Not min or max number of stars as an integer

Dim intOtherStarNum As Integer

'Creates integer to keep track of max stars

Dim intMaxStars As Integer = 97

Dim intMinStars As Integer = 0

'Checks if user has the max amount of unique words

If CStr(words.Count) = arrFileWords.Count.ToString.Max Then

strNumStars = StrDup(intMaxStars, "\*")

ElseIf CStr(words.Count) = arrFileWords.Count.ToString.Min Then

'Checks if user has the minimum amount of unique words

intMinStarNum = CInt(Microsoft.VisualBasic.AscW(arrFileWords.Count.ToString.Min)) \ CInt(Microsoft.VisualBasic.AscW(arrFileWords.Count.ToString.Max))

intMinStars = intMaxStars \* 0.66

strNumStars = StrDup(intMinStars, "\*")

Else

'Checks the rest of the cases (in the test file case, only other option would be 2)

intOtherStarNum = words.Count \ CInt(AscW(arrFileWords.Count.ToString.Max))

intMinStars = intMaxStars \* 0.33

strNumStars = StrDup(intMinStars, "\*")

End If

'Displays the line for each unique word with the word to uppercase with the count formatted in the thousands and the number of stars

swFileStream.WriteLine(String.Format("{0, -15}", words.Key.ToUpper) & ": " & Format(words.Count, "0000 ") & CStr(strNumStars))

Next

swFileStream.WriteLine()

'Creates the average number as a double

Dim dblAvgNum As Double

For Each words In arrFileWords

' Creates the average (an attempt was made to create it as a double)

Dim intMax As Integer = Microsoft.VisualBasic.AscW(arrFileWords.Count.ToString.Max)

dblAvgNum = words.Count \ intNumWords

Next

'Displays average word utilization

swFileStream.WriteLine("Average Word Utilization: " + CStr(dblAvgNum))

'Creates strings for the words with the min and max correlated with them

Dim strFindMax As String = ""

Dim strFindMin As String = ""

'Counts the amount of

For Each words In arrFileWords

If CStr(words.Count) = arrFileWords.Count.ToString.Max Then

'Gets all words with the max count

strFindMax = words.Key

End If

Next

'Displays max and max values

swFileStream.WriteLine("Highest Word Utilization: " + arrFileWords.Count.ToString.Max + " on " + strFindMax.ToUpper)

For Each words In arrFileWords

If CStr(words.Count) = arrFileWords.Count.ToString.Min Then

'An attempt at combining a string of every single value with the min count

'strFindMin = regex.Replace(strFindMin, "\t\n\r", "")

strFindMin = String.Join(",", words.Key)

End If

Next

' Displays min and min values

swFileStream.WriteLine("Lowest Word Utilization: " + arrFileWords.Count.ToString.Min + " on " + strFindMin.ToUpper)

swFileStream.WriteLine()

swFileStream.Close()

Console.WriteLine()

Console.WriteLine("Report File Generation Completed...")

Console.WriteLine()

Console.WriteLine("Would you like to see the report file? [Y/n]")

'Takes the user's option y or n and stores it

Dim strUserInput As String = Console.ReadLine()

'Converts the input to all lowercase for easier processing

strUserInput = strUserInput.ToLower()

If strUserInput = "y" Then

Dim strConsoleReport As String

'Reads in the entire report line by line and displays it

Using srReader As StreamReader = New StreamReader(strRPTFilePath)

'reads through the entire report file and reads in line by line

strConsoleReport = srReader.ReadToEnd

End Using

'Displays the report file

Console.WriteLine(strConsoleReport)

ElseIf strUserInput = "n" Then

'The user selects no so it skips to the end of the file

MsgBox("See you later, then.")

Else

' If the user doesn't select y or n then the user needs to end the program

MsgBox("That is not a valid option. Exiting program.")

Console.WriteLine()

Console.ReadLine()

End If

End If

Else

'If the user enters a path that does no exist, the user is prompted to exit the program

Console.WriteLine("Sorry, you have entered a wrong path name. Please restart the program and try again.")

Console.ReadLine()

End If

'Prompts the user to close out the application

Console.WriteLine("Application has completed. Press any key to end.")

Console.ReadLine()

End Sub

End Module

Table

Description automatically generated