Burndown09 / **WordUnscrambler** Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights

WordUnscrambler / WordUnscrambler / **Constants.cs**

Newer Older

100644 | 24 lines (21 sloc) | 1.05 KB

Added Constants Class and modified the Program class to use ... 1 hour ago

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace WordUnscrambler
{
    class Constants
    {
        public const string Input = "Enter scrambled word(s) manually or as a file: F - file / M - manual";
        public const string EmptError = "String is empty";
        public const string FileName = "Enter full path including the file name: ";
        public const string InstructionsForInput = "Enter word(s) manually (separated by commas if multiple): ";
        public const string InstructionsForInput2 = "Input '/done' when you have completed your list of scrambled words";
        public const string InstructionsForInput3 = "Input '/done' when you have completed your list of none scrambled words";
        public const string InputError = "The entered option was not recognized.";
        public const string Terminate = "The program will be terminated.";
        public const string Prompt = "Please input your word";


        public const string WordList = "wordlist.txt";
    }
}
```

Burndown09 / **WordUnscrambler** Public

<> Code ⊙ Issues Pull requests Actions Projects Wiki Security Insights

WordUnscrambler / WordUnscrambler / FileReader.cs

Newer Older

100644 | 28 lines (23 sloc) | 649 Bytes

Add project files. 3 days ago

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace WordUnscrambler
{
    class FileReader
    {
        public string[] Read(string filename)
        {
            // Implement this.
```

mine changes to program, believe to have finished read metho... 2 days ago

```
            string filepath = filename;
            StreamReader reader = new StreamReader(filepath);

```

Added Constants Class and modified the Program class to use ... 1 hour ago

```
            List<string> listOfWords = new List<string>();
```

mine changes to program, believe to have finished read metho... 2 days ago

```
            while(reader.Peek() >= 0) {
```

Added Constants Class and modified the Program class to use ... 1 hour ago

```
                listOfWords.Append(reader.ReadLine());
```

mine changes to program, believe to have finished read metho... 2 days ago

```


            }
```

Added Constants Class and modified the Program class to use ... 1 hour ago

```
            return listOfWords.ToArray();
```

Add project files. 3 days ago

```

        }
    }
}
```

Burndown09 / **WordUnscrambler** Public

<> Code · Issues · Pull requests · Actions · Projects · Wiki · Security · Insights

WordUnscrambler / WordUnscrambler / **Program.cs**

Newer Older

100644 | 158 lines (123 sloc) | 5.05 KB

Add project files. 3 days ago

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace WordUnscrambler
{
    class Program
    {
        private static readonly FileReader _fileReader = new FileReader();
        private static readonly WordMatcher _wordMatcher = new WordMatcher();

```

mine changes to program, believe to have finished read metho... 2 days ago

```
        static void Main(string[] args){
```

Added Constants Class and modified the Program class to use ... 1 hour ago

```

```

Add project files. 3 days ago

```
            bool repeat = false;
            do
            {
                try
                {
```

mine changes to program, believe to have finished read metho... 2 days ago

```

```

Add project files. 3 days ago

```


```

Added Constants Class and modified the Program class to use ... 1 hour ago

```
                    Console.WriteLine(Constants.Input);
```

Add project files. 3 days ago

```

```

Added Constants Class and modified the Program class to use ... 1 hour ago

```
                    String option = Console.ReadLine() ?? throw new ArgumentNullException(Constants.EmptError);
```

Add project files. 3 days ago

```


                    switch (option.ToUpper())
                    {
                        case "F":
```

Added Constants Class and modified the Program class to use ... 1 hour ago

```
                            Console.WriteLine(Constants.FileName);
```

Add project files. 3 days ago

```
                            ExecuteScrambledWordsInFileScenario();
                            break;
```

```
                        case "M":
```

Added Constants Class and modified the Program class to use ... 1 hour ago

```
                            Console.WriteLine(Constants.InstructionsForInput);
```

Add project files. 3 days ago

```
                            ExecuteScrambledWordsManualEntryScenario();
                            break;
                        default:
```

Added Constants Class and modified the Program class to use ... 1 hour ago

```
                            Console.WriteLine(Constants.InputError);
```

Add project files. 3 days ago

```
                            repeat = true;
                            break;
                    }

                    Console.ReadLine();

                }


                catch (ArgumentNullException ex)
                {
```

Added Constants Class and modified the Program class to use ... 1 hour ago

```
                    Console.WriteLine(Constants.EmptError + ex.Message);
```

Add project files. 3 days ago

```
                    repeat = true;

                }
                catch (Exception ex)
                {
```

Added Constants Class and modified the Program class to use ... 1 hour ago

```
                    Console.WriteLine(Constants.Terminate + ex.Message);
```

Add project files. 3 days ago

```

                }
            } while (repeat);
        }

        private static void ExecuteScrambledWordsInFileScenario()
        {
            var filename = Console.ReadLine();
            string[] scrambledWords = _fileReader.Read(filename);
            DisplayMatchedUnscrambledWordsFileScenerio(scrambledWords);
        }

        private static void ExecuteScrambledWordsManualEntryScenario()
        {
            List<string> listOfScrambledWords = new List<string>();
```

Added Constants Class and modified the Program class to use ... 1 hour ago

```
            Console.WriteLine(Constants.InstructionsForInput2);
```

Add project files. 3 days ago

```
            string finished = "";
            do
            {
```

Added Constants Class and modified the Program class to use ... 1 hour ago

```
                Console.WriteLine(Constants.Prompt);
```

Add project files. 3 days ago

```

                string userInput = Console.ReadLine();

                if (userInput.Equals("/done")) {
                    finished = userInput;

                }
                else {
                    listOfScrambledWords.Append(userInput);
                }

            }   while(!finished.Equals("/done"));

            string[] scrambledwords = new string[listOfScrambledWords.Count];

            for (int i = 0; i < scrambledwords.Length; i++) {
                scrambledwords[i] = listOfScrambledWords[i];
            }

            List<string> listOfWords = new List<string>();

```

Added Constants Class and modified the Program class to use ... 1 hour ago

```
            Console.WriteLine(Constants.InstructionsForInput3);
```

Add project files. 3 days ago

```
            string done = "";

            do
            {
```

Added Constants Class and modified the Program class to use ... 1 hour ago

```
                Console.WriteLine(Constants.Prompt);
```

Add project files. 3 days ago

```
                string input = Console.ReadLine();
                if (input.Equals("/done"))
                {
                    done = "/done";
                }
                else
                {
                    listOfWords.Append(input);
                }

            } while (!done.Equals("/done"));

            string[] wordList = new string[listOfWords.Count];

            for (int i = 0; i < wordList.Length; i++)
            {
                wordList[i] = listOfWords[i];
            }
            DisplayMatchedUnscrambledWordsManualScenerio(scrambledwords, wordList);
        }

        private static void DisplayMatchedUnscrambledWordsFileScenerio(string[] scrambledWords)
        {
            //read the list of words from the system file.
```

Added Constants Class and modified the Program class to use ... 1 hour ago

```
            string[] wordList = _fileReader.Read(Constants.WordList);
```

Add project files. 3 days ago

```

            //call a word matcher method to get a list of structs of matched words.
            List<MatchedWord> matchedWords = _wordMatcher.Match(scrambledWords, wordList);
```

Added Constants Class and modified the Program class to use ... 1 hour ago

```
            foreach (var matchedWord in matchedWords)
            {
                int i = 0;
                Console.WriteLine("the scrambled word" + matchedWords[i].ScrambledWord + "matched with" +
                    matchedWords[i].Word);
                i++;
            }
```

Add project files. 3 days ago

```
        }

        private static void DisplayMatchedUnscrambledWordsManualScenerio(string[] scrambledWords, string[] wordList)
        {

            //call a word matcher method to get a list of structs of matched words.
            List<MatchedWord> matchedWords = _wordMatcher.Match(scrambledWords, wordList);
```

Worked on displaying the matched words 2 days ago

```

            foreach(var matchedWord in matchedWords){
                    int i = 0;
                Console.WriteLine("the scrambled word" + matchedWords[i].ScrambledWord + "matched with" +
                    matchedWords[i].Word);
                   i++;
            }

```

Add project files. 3 days ago

```
        }
    }
}
```