КПІ ім. Ігоря Сікорського

Кафедра ІПІ

ЗВІТ

про виконання комп’ютерного практикуму № 1

з кредитного модуля

«Основи програмування-2. Методології програмування»

Тема: Текстові файли

Варіант №3

Виконала:

студентка 1-го курсу

гр. ІП-з21 ФІОТ

Гавриленко Даяна Юріївна

Київ 2023

1. Умова завдання:

Изображение выглядит как текст, Шрифт

Автоматически созданное описание

1. Текст програми на мові C#:

Program.cs

using System;  
using System.Collections.Generic;  
using System.Text;  
using System.IO;  
  
namespace Lab1;  
  
internal static class Program  
{  
 static void Main(string[] args)  
 {  
 // створення екземплярів класів  
 FileManager fileManager = new FileManager();  
 TaskManager taskManager = new TaskManager();  
 ConsoleManager consoleManager = new ConsoleManager();  
   
 // створення констант, що містять імена текстових файлів  
 const string firstNameOfFile = "set\_file";  
 const string secondNameOfFile = "get\_file";  
  
 consoleManager.PrintInfoForUser();  
 string[] textFromUser = consoleManager.WorkWithConsoleText();  
  
 fileManager.GenerateFile(firstNameOfFile, textFromUser);  
 string textFromFirstFile = fileManager.ReadAllFromFile(firstNameOfFile);  
  
 fileManager.GenerateFile(secondNameOfFile,  
 taskManager.DefineLongestInText(taskManager.SplitToNewLines(textFromFirstFile)));  
 fileManager.PrintFromFile(firstNameOfFile);  
 fileManager.PrintFromFile(secondNameOfFile);  
 }  
}

ConsoleManager.cs

namespace Lab1;  
  
internal class ConsoleManager  
{  
 public void PrintInfoForUser() //вивести інформацію для вводу користувачу  
 {  
 Console.WriteLine("Enter your text:\t" +  
 "(Please, press \"Ctrl + E\" to end entering text)");  
 }  
  
 public string[] WorkWithConsoleText() //отримати текст від користувача  
 {  
 List<string> text = new List<string>();  
 bool continueEntering = true;  
 string str = "";  
  
 void AddNewString() // додавання значення до списку та очищення строки  
 {  
 if (str != "")  
 {  
 text.Add(str);  
 str = "";  
 }  
 }  
  
 while (continueEntering)  
 {  
 ConsoleKeyInfo keyInfo = Console.ReadKey();  
  
 switch (keyInfo.Key)  
 {  
 case ConsoleKey.E when keyInfo.Modifiers == ConsoleModifiers.Control:  
 AddNewString();  
 Console.Write(' ');  
 continueEntering = false;  
 break;  
 case ConsoleKey.Enter:  
 Console.CursorTop++; // позиція курсору у консолі перейде на наступний рядок у тому самому стовпці  
 break;  
 case ConsoleKey.Backspace:  
 str = str.Substring(0, str.Length - 1);   
 Console.Write(' ');  
 Console.CursorLeft--; // позиція курсору у консолі перейде на один символ ліворуч у тому самому рядку  
 break;  
 default:  
 str += keyInfo.KeyChar;  
 break;  
 }  
 }  
  
 return text.ToArray();  
 }  
}

FileManager.cs

namespace Lab1;  
  
internal class FileManager  
{  
 public void GenerateFile(string fileName, string[] text) // створення і заповнення файлу  
 {  
 StreamWriter streamWriter = new StreamWriter(fileName, append: false); // Параметр append: false вказує, що файл буде перезаписаний, а не дописаний  
   
 foreach (string line in text)  
 {  
 streamWriter.WriteLine(line);  
 }  
  
 streamWriter.Close();  
 }  
  
 public string ReadAllFromFile(string fileName) // зчитування тексту з файлу  
 {  
 StreamReader streamReader = new StreamReader(fileName);  
 string text = streamReader.ReadToEnd();  
   
 streamReader.Close();   
   
 return text;  
 }  
  
 public void PrintFromFile(string fileName) // виведення тексту з файлу  
 {  
 StreamReader streamReader = new StreamReader(fileName);  
   
 Console.WriteLine($"\nFile name is {fileName}");  
 Console.WriteLine(streamReader.ReadToEnd());  
   
 streamReader.Close();  
 }  
}

TaskManager.cs

namespace Lab1;  
  
internal class TaskManager  
{  
 public string[] SplitToNewLines(string str) //розбивання речення по рядочкам  
 {  
 str = str.Replace(".", ".\n");  
 string[] sentences = str.Split('\n', StringSplitOptions.RemoveEmptyEntries); // розбиття рядка на масив підрядків за символом нового рядка та видалення порожніх елементів  
   
 return sentences;  
 }  
   
 public string FindShortestInString(string str) //знайти найкоротше слово, після речення поставити довжину цього слова та саме слово  
 {  
 str = str.Trim();  
 string[] words = str.Split(new char[] {' ', ',', '.', ';'}, StringSplitOptions.RemoveEmptyEntries);  
 string minWord = words.First();  
 int lenOfMinWord = minWord.Length;  
  
 foreach (string word in words) // знаходження найкоротшого слова  
 {  
 if (word.Length < lenOfMinWord)  
 {  
 minWord = word;  
 lenOfMinWord = word.Length;  
 }  
 }  
  
 return $"{str} {lenOfMinWord}-{minWord}";  
 }  
  
 public string[] DefineLongestInText(string[] text) //виконати завдання для всіх рядків  
 {  
 string[] resultText = new string[text.Length];  
  
 for (int i = 0; i < text.Length; i++)  
 {  
 resultText[i] = FindShortestInString(text[i]);  
 }  
  
 return resultText;  
 }  
}

1. Відеокопія результатів роботи програми:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание