Міністерство освіти України

Національний технічний університет "ХПІ"

кафедра "Інформатики та інтелектуальної власності"

**Звіт**

**Лабораторна робота 1**

з дисципліни "Кросплатформне програмування"

Виконав: студент групи КН-1223в

Япринцев М.А.

Перевірив:

Івашко А.В.

Харків 2025

Зміст

[«Робота з текстовою інформацією у .NET Framework» 3](#_Toc192600066)

[РОЗРОБКА ЗАСТОСУНКА 4](#_Toc192600067)

[ВИСНОВКИ 6](#_Toc192600068)

[ДОДАТОК А 7](#_Toc192600069)

**«Робота з текстовою інформацією у .NET Framework»**

**Мета роботи**: освоєння методики обробки текстової інформації.

**Завдання:**

1. Розробити алгоритм розв’язання задачі, запропонованої для програмування.
2. Здійснити програмну реалізацію розробленого алгоритму.
3. Здійснити налаштування програми, виправивши синтаксичні та логічні помилки.
4. Підібрати тестові дані для перевірки програми, включаючи виняткові випадки.
5. Відповісти на контрольні запитання.
6. Здати викладачу працездатну програму з демонстрацією її роботи на декількох варіантах вихідних даних.

**Варіант виконаної роботи :**

Варіант № 9. Нехай надано рядок, що складається зі слів, відокремлених одне від одного одним і більше пропусками. Визначити довжину найкоротшого із слів, що містяться в ньому.

**РОЗРОБКА ЗАСТОСУНКА**

Код застосунку наданий у Додатку А.

1. **Підключення необхідних бібліотек**

using System; – підключає базову бібліотеку, яка містить основні функції для роботи з введенням/виведенням та іншими операціями.

using System.Linq; – підключає бібліотеку LINQ, що надає зручні методи для роботи з колекціями, такими як Min(), Where(), Select() тощо.

1. **Оголошення класу та методу Main**

class Program – визначає клас Program, який міститиме основний код програми.

static void Main() – точка входу в програму. Це головний метод, який виконується при запуску програми.

1. **Налаштування кодування для коректного відображення Unicode-символів**

System.Console.OutputEncoding – встановлює кодування для виводу в консоль, щоб підтримувати українські символи.

System.Console.InputEncoding – встановлює кодування для коректного зчитування введених даних.

1. **Зчитування рядка від користувача**

Console.WriteLine("Введіть рядок:"); – виводить запрошення для введення тексту.

string input = Console.ReadLine(); – зчитує введений користувачем рядок.

1. **Перевірка на порожній або пустий рядок**

string.IsNullOrWhiteSpace(input) – перевіряє, чи є рядок порожнім або містить тільки пробіли.

Якщо так, то виводиться відповідне повідомлення і виконання програми припиняється (return).

1. **Обробка рядка та визначення довжини найкоротшого слова**

input.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries) Метод Split() розділяє рядок input на окремі слова за пробілом.

new char[] { ' ' } – вказує, що роздільником є пробіл. StringSplitOptions.RemoveEmptyEntries – видаляє порожні елементи, що можуть з'явитися через декілька пробілів підряд.

У результаті виходить масив слів. .Min(word => word.Length) LINQ-метод Min() знаходить мінімальне значення в колекції. word => word.Length – вираз, що отримує довжину кожного слова. У підсумку знаходиться довжина найкоротшого слова в рядку.

1. **Виведення результату**

Використовується інтерпольований рядок $"...", що дозволяє вставляти змінні прямо в текст.

Виводиться довжина найкоротшого слова.

**ВИСНОВКИ**

У ході виконання лабораторної роботи була розглянута та реалізована задача обробки текстової інформації у середовищі .NET Framework. Було розроблено програму, яка визначає довжину найкоротшого слова у введеному рядку, використовуючи можливості мови програмування C# та бібліотеки LINQ.

В процесі виконання роботи були виконані наступні етапи:

1. Розробка алгоритму вирішення задачі, що включала визначення способу розділення рядка на слова та знаходження найкоротшого слова;
2. Реалізація алгоритму за допомогою мови C# із застосуванням методів класу string та функціональних можливостей LINQ, зокрема Split(), Min();
3. Виконано налагодження коду, що дозволило виправити синтаксичні та логічні помилки;
4. Проведено тестування програми на різних вхідних даних, включаючи граничні та виняткові випадки, такі як порожній рядок, рядок із множинними пробілами та різними довжинами слів;
5. Отримано коректні результати роботи програми, що підтвердило правильність реалізованого алгоритму;

Таким чином, у ході лабораторної роботи було закріплено навички роботи з текстовою інформацією у C#, використання методів обробки рядків та застосування бібліотеки LINQ для оптимального пошуку потрібних значень. Програма працює ефективно та демонструє практичне використання методів роботи з текстовими даними у середовищі .NET Framework.

**ДОДАТОК А**

using System;

using System.Linq;

class Program

{

static void Main()

{

//Варіант 9.

// Нехай надано рядок, що складається зі слів, відокремлених одне від

// одного одним і більше пропусками.Визначити довжину найкоротшого

// із слів, що містяться в ньому.

System.Console.OutputEncoding = System.Text.Encoding.Unicode;

System.Console.InputEncoding = System.Text.Encoding.Unicode;

Console.WriteLine("Введіть рядок:");

string input = Console.ReadLine();

if (string.IsNullOrWhiteSpace(input))

{

Console.WriteLine("Рядок порожній або містить лише пробіли.");

return;

}

int minLength = input.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries)

.Min(word => word.Length);

Console.WriteLine($"Довжина найкоротшого слова чи слів: {minLength}");

}

}