

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Локтионова Оксана Геннадьевна

Должность: проректор по учебной работе

Дата подписания: 04.08.2021 14:39:31

Уникальный программный ключ:

0b817ca911e6668abb13a5d426d70e5f1c11eabbf73e943df4a4851fda56d089

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра программной инженерии



АЛГОРИТМЫ РАБОТЫ СО СТРОКАМИ НА ЯЗЫКЕ C#
Методические указания по выполнению практических работ по
дисциплине «Алгоритмические языки» для студентов всех форм
обучения направления подготовки бакалавров 45.03.03
«Фундаментальная и прикладная лингвистика»

Курск 2019

УДК 004.43
Составитель Е.А. Петрик

Рецензент
Кандидат технических наук, доцент Т.И.Лапина

Работа со строками на языке С#: методические указания по выполнению практических работ / Юго-Зап. гос. ун-т; сост.: Е. А. Петрик. Курск, 2019. 16 с.: ил.4, табл. 1. Библиогр.: с.16.

Содержат краткие теоретические сведения о строковом типе данных и методах работы с ним, а также приведены примеры и индивидуальные задания для практической работы.

Методические указания соответствуют требованиям программы по направлению подготовки бакалавров: 45.03.03 «Фундаментальная и прикладная лингвистика».

Предназначены для студентов всех форм обучения направления подготовки бакалавров 45.03.03 «Фундаментальная и прикладная лингвистика».

Текст печатается в авторской редакции

Подписано в печать 31.10.2019. Формат 60x84 1/16.
Усл. печ. л. 0,7 Уч. – изд. л. 0,6 .Тираж 100 экз. Заказ 642. Бесплатно.
Юго - Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

Цель работы

Изучение строкового типа данных и методов работы со строками.

Основные понятия

Строка – это последовательная коллекция символов, используемая для представления текста

`string` – ключевое слово для типа данных для задания строк, представляет последовательность, состоящую из нуля или более символов в кодировке Юникод. `string` является псевдонимом для класса `System.String` в .NET.

Строки в `c#` можно создавать различными способами (рисунок 1).

```
string s1 = "Привет! Это строка.";
char[] mas = { 'П', 'р', 'и', 'в', 'е', 'т' };
string s2 = new string(mas);
```

Рисунок 1 – Объявление строк

Для работы со строками в языке `c#` определено множество методов, определенных в классе `System.String`. Описание методов приведено в таблице 1. [1]

Таблица 1 – Методы работы со строками

Наименование	Описание
<code>Chars[Int32]</code>	Возвращает объект <code>Char</code> в указанной позиции в текущем объекте <code>String</code>
<code>Length</code>	Возвращает число знаков в текущем объекте <code>String</code> .
<code>Clone()</code>	Возвращает ссылку на данный экземпляр класса <code>String</code> .
<code>Compare(String, Int32, String, Int32, Int32)</code>	Сравнивает подстроки двух указанных объектов <code>String</code> и возвращает целое число, которое показывает их относительное положение в порядке сортировки.
<code>Compare(String, Int32, String, Int32, Int32, Boolean)</code>	Сравнивает подстроки двух заданных объектов <code>String</code> (с учетом или без учета регистра) и возвращает целое число, которое показывает их относительное положение в порядке сортировки.
<code>Compare(String, Int32, String, Int32, Int32, Boolean, CultureInfo)</code>	Сравнивает подстроки двух заданных объектов <code>String</code> (с учетом или без учета регистра), используя сведения о языке и региональных параметрах, и возвращает целое число, которое показывает их относительное положение в порядке сортировки.

Продолжение таблицы 1

Compare(String, Int32, String, Int32, Int32, CultureInfo, CompareOptions)	Сравнивает подстроки двух заданных объектов String, используя указанные параметры сравнения и сведения о языке и региональных параметрах, которые влияют на сравнение, и возвращает целое число, показывающее связь между двумя подстроками в порядке сортировки.
Compare(String, Int32, String, Int32, Int32, StringComparison)	Сравнивает подстроки двух указанных объектов String с использованием заданных правил и возвращает целое число, которое показывает их относительное положение в порядке сортировки.
Compare(String, String)	Сравнивает два указанных объекта String и возвращает целое число, которое показывает их относительное положение в порядке сортировки.
Compare(String, String, Boolean)	Сравнивает два указанных объекта String (с учетом или без учета регистра) и возвращает целое число, которое показывает их относительное положение в порядке сортировки.
Compare(String, String, Boolean, CultureInfo)	Сравнивает два указанных объекта String (с учетом или без учета регистра), используя сведения о языке и региональных параметрах, и возвращает целое число, которое показывает их относительное положение в порядке сортировки.
Compare(String, String, CultureInfo, CompareOptions)	Сравнивает два заданных объекта String, используя указанные параметры сравнения и сведения о языке и региональных параметрах, которые влияют на сравнение, и возвращает целое число, показывающее связь между двумя строками в порядке сортировки.
Compare(String, String, StringComparison)	Сравнивает два указанных объекта String с использованием заданных правил и возвращает целое число, которое показывает их относительное положение в порядке сортировки.
CompareOrdinal(String, Int32, String, Int32, Int32)	Сравнивает подстроки двух указанных объектов String, вычисляя числовые значения соответствующих объектов Char в каждой подстроке.
CompareOrdinal(String, String)	Сравнивает два указанных объекта String, оценивая числовые значения соответствующих объектов Char в каждой строке.
CompareTo(Object)	Сравнивает данный экземпляр с указанным объектом Object и показывает, расположен ли экземпляр перед, после или в той же позиции в порядке сортировки, что и заданный объект Object.
CompareTo(String)	Сравнивает данный экземпляр с заданным объектом String и показывает, расположен ли данный экземпляр перед, после или на той же позиции в порядке сортировки, что и заданная строка.

Продолжение таблицы 1

Concat(IEnumerable<String>)	Сцепляет элементы созданной коллекции IEnumerable<T>
-----------------------------	--

	типа <code>String</code> .
<code>Concat(Object)</code>	Создает строковое представление указанного объекта.
<code>Concat(Object, Object)</code>	Сцепляет строковые представления двух указанных объектов.
<code>Concat(Object, Object, Object)</code>	Сцепляет строковые представления трех указанных объектов.
<code>Concat(Object, Object, Object, Object)</code>	Сцепляет строковые представления четырех указанных объектов и любые объекты, заданные в необязательном списке параметров переменной длины.
<code>Concat(Object[])</code>	Сцепляет строковые представления элементов указанного массива <code>Object</code> .
<code>Concat(String, String)</code>	Сцепляет два указанных экземпляра <code>String</code> .
<code>Concat(String, String, String)</code>	Сцепляет три указанных экземпляра <code>String</code> .
<code>Concat(String, String, String, String)</code>	Сцепляет четыре указанных экземпляра <code>String</code> .
<code>Concat(String[])</code>	Сцепляет элементы указанного массива <code>String</code> .
<code>Concat<T>(IEnumerable<T>)</code>	Сцепляет элементы реализации <code>IEnumerable<T></code> .
<code>Contains(String)</code>	Возвращает значение, указывающее, встречается ли указанная подстрока внутри этой строки.
<code>Copy(String)</code>	Создает экземпляр <code>String</code> , имеющий то же значение, что и указанный экземпляр <code>String</code> .
<code>CopyTo(Int32, Char[], Int32, Int32)</code>	Копирует заданное число знаков, начиная с указанной позиции в этом экземпляре до указанной позиции в массиве знаков Юникода.
<code>EndsWith(String)</code>	Определяет, совпадает ли конец данного экземпляра строки с указанной строкой.
<code>EndsWith(String, Boolean, CultureInfo)</code>	Определяет, совпадает ли конец данного экземпляра строки с заданной строкой при сравнении с учетом заданного языка и региональных параметров.
<code>EndsWith(String, StringComparison)</code>	Определяет, совпадает ли конец экземпляра строки с заданной строкой при сравнении с учетом заданного параметра сравнения.
<code>Equals(Object)</code>	Определяет, равны ли значения этого экземпляра и указанного объекта, который также должен быть объектом <code>String</code> .
<code>Equals(String)</code>	Определяет, равны ли значения этого экземпляра и указанного объекта <code>String</code> .
<code>Equals(String, String)</code>	Определяет, совпадают ли значения двух указанных объектов <code>String</code> .
<code>Equals(String, String, StringComparison)</code>	Определяет, совпадают ли значения двух указанных объектов <code>String</code> . Параметр определяет язык и региональные параметры, учет регистра и правила сортировки, используемые при сравнении.
<code>Equals(String, StringComparison)</code>	Определяет, равны ли значения этой строки и указанного объекта <code>String</code> . Параметр определяет язык и региональные параметры, учет регистра и правила сортировки, используемые при сравнении.

Продолжение таблицы 1

<code>Format(IFormatProvider,</code>	Заменяет элементы формата в указанной строке строковым
--------------------------------------	--

String, Object)	представлением соответствующего объекта. Параметр предоставляет сведения об особенностях форматирования, связанных с языком и региональными параметрами.
Format(IFormatProvider, String, Object, Object)	Заменяет элементы формата в строке строковым представлением двух указанных объектов. Параметр предоставляет сведения об особенностях форматирования, связанных с языком и региональными параметрами.
Format(IFormatProvider, String, Object, Object, Object)	Заменяет элементы формата в строке строковым представлением трех указанных объектов. Параметр предоставляет сведения об особенностях форматирования, связанных с языком и региональными параметрами.
Format(IFormatProvider, String, Object[])	Заменяет элементы формата в строке строковым представлением соответствующих объектов в указанном массиве. Параметр предоставляет сведения об особенностях форматирования, связанных с языком и региональными параметрами.
Format(String, Object)	Заменяет один или более элементов формата в строке строковым представлением указанного объекта.
Format(String, Object, Object)	Заменяет элементы формата в строке строковым представлением двух указанных объектов.
Format(String, Object, Object, Object)	Заменяет элементы формата в строке строковым представлением трех указанных объектов.
Format(String, Object[])	Заменяет элемент формата в указанной строке строковым представлением соответствующего объекта в указанном массиве.
GetEnumerator()	Извлекает объект, который может выполнять итерацию отдельных знаков данной строки.
GetHashCode()	Возвращает хэш-код для этой строки.
GetType()	Возвращает объект Type для текущего экземпляра. (Inherited from Object)
GetTypeCode()	Возвращает TypeCode для класса String.
IndexOf(Char)	Возвращает индекс с отсчетом от нуля первого вхождения указанного символа Юникода в данной строке.
IndexOf(Char, Int32)	Возвращает индекс с отсчетом от нуля первого вхождения указанного символа Юникода в данной строке. Поиск начинается с указанной позиции знака.
IndexOf(Char, Int32, Int32)	Возвращает индекс с отсчетом от нуля первого вхождения указанного символа в данном экземпляре. Поиск начинается с указанной позиции знака; проверяется заданное количество позиций.
IndexOf(String)	Возвращает индекс с отсчетом от нуля первого вхождения значения указанной строки в данном экземпляре.

Продолжение таблицы 1

IndexOf(String, Int32)	Возвращает индекс с отсчетом от нуля первого вхож-
------------------------	--

	дения значения указанной строки в данном экземпляре. Поиск начинается с указанной позиции знака.
IndexOf(String, Int32, Int32)	Возвращает индекс с отсчетом от нуля первого вхождения значения указанной строки в данном экземпляре. Поиск начинается с указанной позиции знака; проверяется заданное количество позиций.
IndexOf(String, Int32, Int32, StringComparison)	Возвращает индекс с отсчетом от нуля первого вхождения указанной строки в текущем объекте String. Параметры задают начальную позицию поиска в текущей строке, количество проверяемых знаков текущей строки и тип поиска.
IndexOf(String, Int32, StringComparison)	Возвращает индекс с отсчетом от нуля первого вхождения указанной строки в текущем объекте String. Параметры задают начальную позицию поиска в текущей строке и тип поиска.
IndexOf(String, StringComparison)	Возвращает индекс с отсчетом от нуля первого вхождения указанной строки в текущем объекте String. Параметр определяет тип поиска заданной строки.
IndexOfAny(Char[])	Возвращает индекс с отсчетом от нуля первого обнаруженного в данном экземпляре символа из указанного массива символов Юникода.
IndexOfAny(Char[], Int32)	Возвращает индекс с отсчетом от нуля первого обнаруженного в данном экземпляре символа из указанного массива символов Юникода. Поиск начинается с указанной позиции знака.
IndexOfAny(Char[], Int32, Int32)	Возвращает индекс с отсчетом от нуля первого обнаруженного в данном экземпляре символа из указанного массива символов Юникода. Поиск начинается с указанной позиции знака; проверяется заданное количество позиций.
Insert(Int32, String)	Возвращает новую строку, в которой указанная строка вставляется в указанной позиции индекса в данном экземпляре.
Intern(String)	Извлекает системную ссылку на указанный объект String.
IsInterned(String)	Извлекает ссылку на указанный объект String.
IsNormalized()	Указывает, находится ли данная строка в форме нормализации Юникода С.
IsNormalized(NormalizationForm)	Указывает, находится ли данная строка в заданной форме нормализации Юникода.
IsNullOrEmpty(String)	Указывает, действительно ли указанная строка является строкой null или пустой строкой ("").
IsNullOrWhiteSpace(String)	Указывает, имеет ли указанная строка значение null, является ли она пустой строкой или строкой, состоящей только из символов-разделителей.

Продолжение таблицы 1

Join(String, IEnumerable<String>)	Сцепляет элементы созданной коллекции IEnumerable<T> типа String, помещая между ними заданный разделитель.
Join(String, Object[])	Сцепляет элементы массива объектов, помещая между ними заданный разделитель.
Join(String, String[])	Сцепляет все элементы массива строк, помещая между ними заданный разделитель.
Join(String, String[], Int32, Int32)	Сцепляет указанные элементы массива строк, помещая между ними заданный разделитель.
Join<T>(String, IEnumerable<T>)	Сцепляет элементы созданной коллекции, помещая между ними заданный разделитель.
LastIndexOf(Char)	Возвращает позицию индекса с отсчетом от нуля последнего вхождения указанного символа Юникода в пределах данного экземпляра.
LastIndexOf(Char, Int32)	Возвращает позицию индекса с отсчетом от нуля последнего вхождения указанного символа Юникода в пределах данного экземпляра. Поиск начинается с указанной позиции символа и выполняется в обратном направлении до начала строки.
LastIndexOf(Char, Int32, Int32)	Возвращает позицию индекса с отсчетом от нуля последнего вхождения указанного символа Юникода в подстроке в пределах данного экземпляра. Поиск начинается с указанной позиции символа и выполняется в обратном направлении до начала строки для заданного числа позиций символов.
LastIndexOf(String)	Возвращает позицию индекса с отсчетом от нуля последнего вхождения указанной строки в данном экземпляре.
LastIndexOf(String, Int32)	Возвращает позицию индекса с отсчетом от нуля последнего вхождения указанной строки в данном экземпляре. Поиск начинается с указанной позиции символа и выполняется в обратном направлении до начала строки.
LastIndexOf(String, Int32, Int32)	Возвращает позицию индекса с отсчетом от нуля последнего вхождения указанной строки в данном экземпляре. Поиск начинается с указанной позиции символа и выполняется в обратном направлении до начала строки для заданного числа позиций символов.
LastIndexOf(String, Int32, Int32, StringComparison)	Возвращает позицию индекса с отсчетом от нуля последнего вхождения указанной строки в данном экземпляре. Поиск начинается с указанной позиции символа и выполняется в обратном направлении до начала строки для заданного числа позиций символов. Параметр определяет тип сравнения для выполнения во время поиска заданной строки.

LastIndexOf(String, Int32, StringComparison)	Возвращает индекс с отсчетом от нуля последнего вхождения указанной строки в текущем объекте String. Поиск начинается с указанной позиции символа и выполняется в обратном направлении до начала строки. Параметр определяет тип сравнения для выполнения во время поиска заданной строки.
LastIndexOf(String, StringComparison)	Возвращает индекс с отсчетом от нуля последнего вхождения указанной строки в текущем объекте String. Параметр определяет тип поиска заданной строки.
LastIndexOfAny(Char[])	Возвращает позицию индекса с отсчетом от нуля последнего вхождения в данном экземпляре какого-либо одного или нескольких символов, указанных в массиве символов Юникода.
LastIndexOfAny(Char[], Int32)	Возвращает позицию индекса с отсчетом от нуля последнего вхождения в данном экземпляре какого-либо одного или нескольких символов, указанных в массиве символов Юникода. Поиск начинается с указанной позиции символа и выполняется в обратном направлении до начала строки.
LastIndexOfAny(Char[], Int32, Int32)	Возвращает позицию индекса с отсчетом от нуля последнего вхождения в данном экземпляре какого-либо одного или нескольких символов, указанных в массиве символов Юникода. Поиск начинается с указанной позиции символа и выполняется в обратном направлении до начала строки для заданного числа позиций символов.
MemberwiseClone()	Создает неполную копию текущего объекта Object. (Inherited from Object)
Normalize()	Возвращает новую строку, текстовое значение которой совпадает с данной строкой, а двоичное представление находится в нормализованной форме С Юникода.
Normalize(NormalizationForm)	Возвращает новую строку, текстовое значение которой совпадает с данной строкой, а двоичное представление находится в заданной нормализованной форме Юникода.
PadLeft(Int32)	Возвращает новую строку, в которой знаки данного экземпляра выровнены по правому краю путем добавления слева символов-разделителей до указанной общей длины.
PadLeft(Int32, Char)	Возвращает новую строку, в которой знаки данного экземпляра выровнены по правому краю путем добавления слева пробелов или указанного знака Юникода до указанной общей длины.
PadRight(Int32)	Возвращает новую строку, в которой знаки данной строки выровнены по левому краю путем добавления справа пробелов до указанной общей длины.

Продолжение таблицы 1

PadRight(Int32, Char)	Возвращает новую строку, в которой знаки данной строки вы-
-----------------------	--

	ровнены по левому краю путем добавления справа пробелов или указанного знака Юникода до указанной общей длины.
Remove(Int32)	Возвращает новую строку, в которой были удалены все символы, начиная с указанной позиции и до конца в текущем экземпляре.
Remove(Int32, Int32)	Возвращает новую строку, в которой было удалено указанное число символов в указанной позиции.
Replace(Char, Char)	Возвращает новую строку, в которой все вхождения заданного знака Юникода в текущем экземпляре заменены другим заданным знаком Юникода.
Replace(String, String)	Возвращает новую строку, в которой все вхождения заданной строки в текущем экземпляре заменены другой заданной строкой.
Split(Char[])	Разбивает строку на подстроки в зависимости от символов в массиве.
Split(Char[], Int32)	Разбивает строку на максимальное число подстрок в зависимости от символов в массиве. Можно также указать максимальное число возвращаемых подстрок.
Split(Char[], Int32, StringSplitOptions)	Разбивает строку на максимальное число подстрок в зависимости от символов в массиве.
Split(Char[], StringSplitOptions)	Разбивает строку на подстроки в зависимости от символов в массиве. Можно указать, включают ли подстроки пустые элементы массива.
Split(String[], Int32, StringSplitOptions)	Разбивает строку на максимальное число подстрок в зависимости от строк в массиве. Можно указать, включают ли подстроки пустые элементы массива.
Split(String[], StringSplitOptions)	Разбивает строку на подстроки в зависимости от строк в массиве. Можно указать, включают ли подстроки пустые элементы массива.
StartsWith(String)	Определяет, совпадает ли начало данного экземпляра строки с указанной строкой.
StartsWith(String, Boolean, CultureInfo)	Определяет, совпадает ли начало экземпляра строки с заданной строкой при сравнении с учетом заданного языка и региональных параметров.
StartsWith(String, StringComparison)	Определяет, совпадает ли начало этого экземпляра строки с заданной строкой при сравнении с учетом заданного параметра сравнения.
Substring(Int32)	Извлекает подстроку из данного экземпляра. Подстрока начинается в указанном положении символов и продолжается до конца строки.
Substring(Int32, Int32)	Извлекает подстроку из данного экземпляра. Подстрока начинается с указанной позиции знака и имеет указанную длину.
ToCharArray()	Копирует знаки данного экземпляра в массив знаков Юникода.

Продолжение таблицы 1

ToCharArray(Int32, Int32)	Копирует знаки из указанной подстроки данного экземпляра в
---------------------------	--

	массив знаков Юникода.
ToLower()	Возвращает копию этой строки, переведенную в нижний регистр.
ToLower(CultureInfo)	Возвращает копию этой строки, переведенную в нижний регистр, используя правила определения регистра заданного языка и региональных параметров.
ToLowerInvariant()	Возвращает копию этого объекта String , переведенную в нижний регистр, используя правила учета регистра инвариантного языка и региональных параметров.
ToString()	Возвращает этот экземпляр String; реальное преобразование не осуществляется.
ToString(IFormatProvider)	Возвращает этот экземпляр String; реальное преобразование не осуществляется.
ToUpper()	Возвращает копию этой строки, переведенную в верхний регистр.
ToUpper(CultureInfo)	Возвращает копию этой строки, переведенную в верхний регистр, используя правила определения регистра заданного языка и региональных параметров.
ToUpperInvariant()	Возвращает копию этого объекта String , переведенную в верхний регистр, используя правила учета регистра инвариантного языка и региональных параметров.
Trim()	Удаляет все начальные и конечные символы-разделители из текущего объекта String.
Trim(Char[])	Удаляет все начальные и конечные вхождения набора знаков, заданного в виде массива, из текущего объекта String.
TrimEnd(Char[])	Удаляет все конечные вхождения набора знаков, заданного в виде массива, из текущего объекта String.
TrimStart(Char[])	Удаляет все начальные вхождения набора знаков, заданного в виде массива, из текущего объекта String.

Пример выполнения задания

Задание

В заданной строке, состоящей из не более 40 слов из заглавных и строчных русских букв, разделенных пробелами, максимальная длина слов в которой 8 символов, найти повторяющиеся слова и удалить повторения.

Алгоритм решения задачи

Для решения задачи необходимо сначала проверить, удовлетворяет ли всем условиям входная строка. Для этого в строке необходимо проверить отсутствие запрещенных символов, а потом преобразовать в массив слов. Метод `String.Split` возвращает строковый массив, содержащий подстроки данного экземпляра, разделенные элементами заданной строки или массива знаков Юникода (в нашем случае этот элемент – пробел). Далее необходимо вычислить количество слов в строке и проверить, не превышает ли их количество 40, а также длину каждого слова (≤ 8 символов). После этого осуществить сравнение слов в полученном массиве друг с другом и удалить повторяющиеся слова.

Листинг программы представлен на рисунке 2.

На рисунках 3 и 4 представлены примеры выполнения программы.

```

1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10 using System.Text.RegularExpressions;
11
12 namespace WindowsFormsApp6
13 {
14     Ссылка: 3
15     public partial class Form1 : Form
16     {
17         ссылка: 1
18         public Form1()
19         {
20             InitializeComponent();
21         }
22         ссылка: 1
23         private void Button1_Click(object sender, EventArgs e)
24         {
25             string s = textBox1.Text; //ввод данных
26             foreach (char c in s)
27                 if (!(c == 32 || ((c >= 1040) && (c <= 1103)))) //поиск запрещенных символов
28                 {
29                     label1.Text = "ошибочный символ " + c;
30                     return; //выход
31                 }
32             string[] mas = s.Split(' '); //слова, разделенные знаком пробела, записываются в массив строк mas
33             if (mas.Length > 40)
34             {
35                 label1.Text = "Слов более 40 ";
36                 return; //выход
37             }
38             foreach (string s1 in mas)
39                 if (s1.Length > 8)
40                 {
41                     label1.Text = "Символов более 8 ";
42                     return; //выход
43                 }
44             for (int i = 0; i < mas.Length; i++)
45                 for (int j = 0; j < mas.Length; j++)
46                     if (i != j && (mas[i] == mas[j]))
47                         mas[j] = "";
48             string rstr = String.Join(" ", mas);
49             string pattern = @"\s+";
50             string replacement = " ";
51             Regex rx = new Regex(pattern);
52             label1.Text = rx.Replace(rstr, replacement);
53         }
54     }
55 }
56

```

Рисунок 2 – Листинг программы

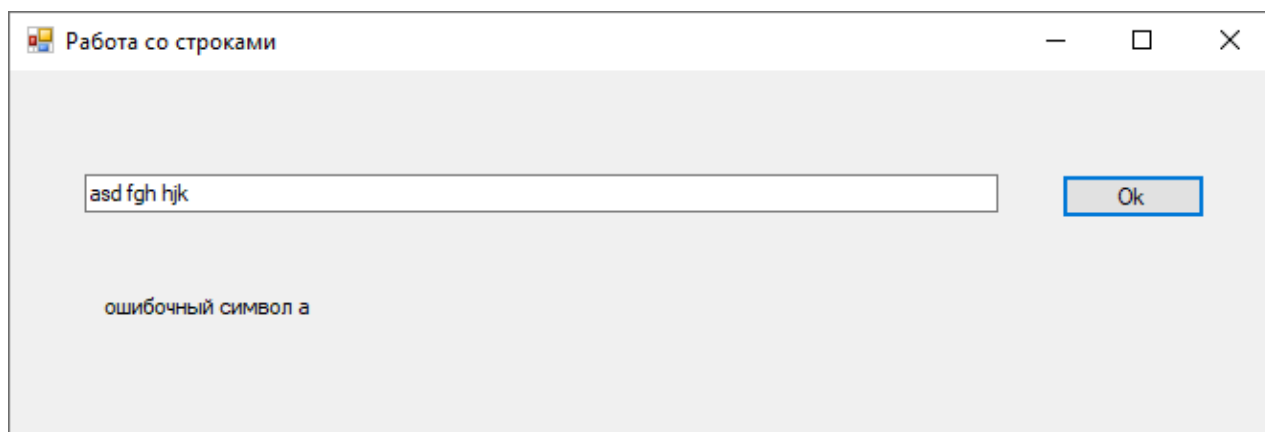


Рисунок 3 – Пример работы программы

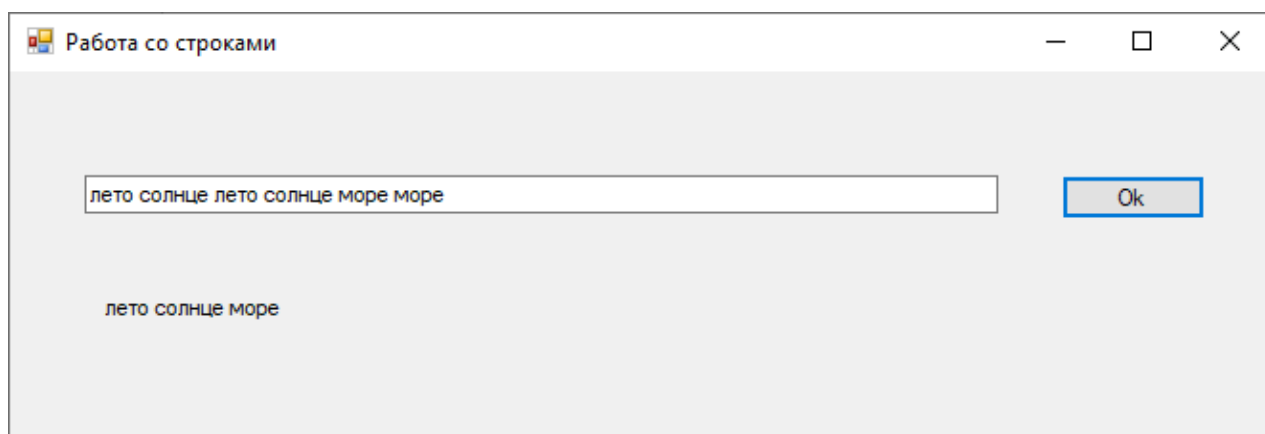


Рисунок 4 – Пример работы программы

Индивидуальные задания для лабораторной работы

1. В заданной строке, состоящей из не более 50 слов слов из заглавных и строчных латинских букв, разделенных запятыми, максимальная длина слов в которой 10 символов, удалить все заглавные буквы из слов.

2. Из заданной строки, состоящей из не более 30 слов слов из строчных латинских букв и цифр, разделенных запятыми, максимальная длина слов в которой 8 символов, составить новый массив из чисел, каждое из которых образуется из цифр отдельного слова входной последовательности.

3. Дана строка, состоящая из не более 40 слов из строчных латинских букв и цифр, разделенных запятыми, максимальная длина слов в которой 7 символов, составить новую строку, в которой слова из входной последовательности будут упорядочены по алфавиту.

4. Дана строка, состоящая из не более 60 слов из строчных русских букв, разделенных двоеточием, максимальная длина слов в которой 9 символов, составить новую строку, в которой слова из входной последовательности будут упорядочены по увеличению длин слов.

5. Из слов заданной строки, состоящей из не более 50 слов из заглавных и строчных латинских букв, разделенных знаками «точка с запятой», максимальная длина слов в которой 15 символов, составить новую строку, предварительно удалив среднюю букву из слов нечетной длины.

6. Дана строка, состоящая из не более 40 слов из строчных и заглавных русских букв, разделенных запятой, максимальная длина слов в которой 10 символов, составить новую строку, в которой в словах из входной последовательности необходимо удалить гласные буквы.

7. Дана строка, состоящая из не более 50 слов из строчных латинских букв и цифр, разделенных запятыми, максимальная длина слов в которой 10 символов, составить новую строку, в которой слова из входной последовательности будут упорядочены по алфавиту в обратном порядке.

8. Дана строка, состоящая из не более 60 слов из строчных русских букв, разделенных пробелами, максимальная длина слов в которой 20 символов, составить новую строку, в которой слова из входной последовательности будут упорядочены по уменьшению длин слов.

Содержание отчета

1. Титульный лист.
2. Текст задания.
3. Описание данных программы.
4. Листинг программы.
5. Примеры работы программы.

Контрольные вопросы

1. Что такое строка?
2. Опишите варианты объявления строк.
3. Какие существуют методы для работы со строками?
4. Как работает метод `String.Split`?
5. Приведите примеры использования методов для работы с подстрокой?

Список литературы

1. `String Class` [Электронный ресурс] – Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/api/system.string?view=netframework-4.8> (дата обращения: 21.08.19)
2. Голицына О. Л. Программное обеспечение [Текст] : учебное пособие / О. Л. Голицына, Т. Л. Партыка, И. И. Попов. - 3-е изд., перераб. и доп. - М. : Форум, 2010. - 201 с.
3. Иванова Г. С. Технология программирования [Текст] : учебник / Г. С. Иванова. – М. : Кнорус, 2011. – 336 с.