Работа со строками

1. Создайте проект с именем String

В первом проекте рассмотрим свойства строк

// Задаем строку

string s1 = "hello";

// Обращаемся как к массиву

Console.WriteLine(s1[2]);

// Размер строки

Console.WriteLine(s1.Length);

// Проверка на null

Console.WriteLine(string.IsNullOrEmpty(s1));

// Проверка на пробелы

Console.WriteLine(string.IsNullOrWhiteSpace(s1));

Попробуйте задать другие параметры и проверить работу программы.

1. Конкатенация строк

В этом проекте мы соединим строки в одну. Добавьте в решение проект с именем Concat.

using System;

namespace Concat

{

class Program

{

static void Main(string[] args)

{

string s1 = "hello";

string s2 = "world";

string s3 = s1 + " " + s2; // результат: строка "hello world"

string s4 = String.Concat(s3, "!!!"); // результат: строка "hello world!!!"

Console.WriteLine(s4);

}

}

}

1. Рассмотрим конкатенацию строк с методом Join

using System;

namespace Join\_Concat

{

class Program

{

static void Main(string[] args)

{

string s5 = "C";

string s6 = "добрым";

string s7 = "утром!";

string s8 = "Хорошего";

string s9 = "дня!";

string[] values = new string[] { s5, s6, s7, s8, s9 };

String s10 = String.Join(" ", values);

Console.WriteLine(s10);

}

}

}

Измените количество строк и значения строк. Проверьте работоспособность программы.

1. В данном проекте выполним сравнение строк:

using System;

// Сравнение строк

namespace Compare

{

class Program

{

static void Main(string[] args)

{

string s1 = "hello";

string s2 = "world";

int result = String.Compare(s1, s2);

if (result < 0)

{

Console.WriteLine("Строка s1 перед строкой s2");

}

else if (result > 0)

{

Console.WriteLine("Строка s1 стоит после строки s2");

}

else

{

Console.WriteLine("Строки s1 и s2 идентичны");

}

// результатом будет "Строка s1 перед строкой s2

Измените количество строк и значения строк. Проверьте работоспособность программы.

Как видите, метод String.Compare работает только с двумя строками.

1. В данном проекте рассмотрим методы поиска в строке:

using System;

//Поиск в строке

namespace Search

{

class Program

{

static void Main(string[] args)

{

//Объявляем строку

string s1 = "hello world";

// Объявляем что ищем

char ch = 'o';

// С помощью метода IndexOf мы можем определить индекс первого вхождения отдельного

// символа или подстроки в строке:

int indexOfChar = s1.IndexOf(ch); // равно 4

Console.WriteLine(indexOfChar);

string subString = "wor";

int indexOfSubstring = s1.IndexOf(subString); // равно 6

Console.WriteLine(indexOfSubstring);

}

}

}

Подобным образом действует метод LastIndexOf, только находит индекс последнего вхождения символа или подстроки в строку. Измените программу и проверьте последнее вхождение символа и подстроки.

1. В данном проекте рассмотри разделение строк

using System;

// Разделение строки

namespace Split

{

class Program

{

static void Main(string[] args)

{

//С помощью функции Split мы можем разделить строку на массив подстрок.

//В качестве параметра функция Split принимает массив символов или строк,

//которые и будут служить разделителями

string text = "И поэтому все так произошло";

string[] words = text.Split(new char[] { ' ' });

foreach (string s in words)

{

Console.WriteLine(s);

}

}

}

}

1. Рассмотри метод обрезки строки

using System;

// Обрезка строки

namespace Tpim

{

class Program

{

static void Main(string[] args)

{

string text = " hello world ";

//Функция Trim без параметров обрезает начальные и конечные пробелы и возвращает обрезанную строку.

//Чтобы явным образом указать,

//какие начальные и конечные символы следует обрезать,

//мы можем передать в функцию массив этих символов.

text = text.Trim(); // результат "hello world"

text = text.Trim(new char[] { 'd', 'h' }); // результат "ello worl"

//Обрезать определенную часть строки позволяет функция Substring:

string text1 = "Хороший день";

// обрезаем начиная с третьего символа

text1 = text1.Substring(2);

// результат "роший день"

Console.WriteLine(text1);

// обрезаем сначала до последних двух символов

text1 = text1.Substring(0, text1.Length - 2);

// результат "роший де"

Console.WriteLine(text1);

}

}

}

Самостоятельная работа:

Проверьте работу функций по вставке, удалению и замене символов в строке

Вставка

Для вставки одной строки в другую применяется функция Insert:

|  |  |
| --- | --- |
| 1  2  3  4  5 | string text = "Хороший день";  string subString = "замечательный ";    text = text.Insert(8, subString);  Console.WriteLine(text); |

Первым параметром в функции Insert является индекс, по которому надо вставлять подстроку, а второй параметр - собственно подстрока.

**Удаление строк**

Удалить часть строки помогает метод Remove:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | string text = "Хороший день";  // индекс последнего символа  int ind = text.Length - 1;  // вырезаем последний символ  text = text.Remove(ind);  Console.WriteLine(text);    // вырезаем первые два символа  text = text.Remove(0, 2); |

Первая версия метода Remove принимает индекс в строке, начиная с которого надо удалить все символы. Вторая версия принимает еще один параметр - сколько символов надо удалить.

**Замена**

Чтобы заменить один символ или подстроку на другую, применяется метод Replace:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | string text = "хороший день";    text = text.Replace("хороший", "плохой");  Console.WriteLine(text);    text = text.Replace("о", "");  Console.WriteLine(text); |

Во втором случае применения функции Replace строка из одного символа "о" заменяется на пустую строку, то есть фактически удаляется из текста. Подобным способом легко удалять какой-то определенный текст в строках.

**Смена регистра**

Для приведения строки к верхнему и нижнему регистру используются соответственно функции ToUpper() и ToLower():

|  |  |
| --- | --- |
| 1  2  3  4 | string hello = "Hello world!";    Console.WriteLine(hello.ToLower()); // hello world!  Console.WriteLine(hello.ToUpper()); // HELLO WORLD! |

Задачи:

1. Ввести строку. Посчитать, сколько раз встречается буква «а».
2. В строке, введенной с клавиатуры, удалите символ с позицией 10.
3. В строке, введенной с клавиатуры, замените символ и подстроку.