

Лабораторная работа №4

Обработка строк

Строки и класс System.String

Довольно большое количество задач, которые могут встретиться при разработке приложений, так или иначе связано с обработкой строк - парсинг веб-страниц, поиск в тексте, какие-то аналитические задачи, связанные с извлечением нужной информации из текста и т.д. Поэтому в этом плане работе со строками уделяется особое внимание.

В языке C# строковые значения представляет тип `string`, а вся функциональность работы с данным типом сосредоточена в классе `System.String`. Собственно, `string` является псевдонимом для класса `System.String`. Объекты этого класса представляют текст как последовательность символов Unicode. Максимальный размер объекта `String` может составлять в памяти 2 ГБ, или около 1 миллиарда символов.

Создание строк

Создавать строки можно, как используя переменную типа `string` и присваивая ей значение, так и применяя один из конструкторов класса `String`:

```
string s1 = "hello";
string s2 = null;

string s3 = new String('a', 6); // результатом будет строка "aaaaaa"
string s4 = new String(new char[]{'w', 'o', 'r', 'l', 'd'});
```

Конструктор `String` имеет различное число версий. Так, вызов конструктора `new String('a', 6)` создаст строку "aaaaaa". И так как строка представляет ссылочный тип, то может хранить значение `null`.

Строка как набор символов

Так как строка хранит коллекцию символов, в ней определен индексатор для доступа к этим символам. Применяя индексатор, мы можем обратиться к строке как к массиву символов и получить по индексу любой из ее символов:

```
string s1 = "hello";
char ch1 = s1[1]; // символ 'e'
Console.WriteLine(ch1);
```

```
Console.WriteLine(s1.Length);
```

Используя свойство `Length`, как и в обычном массиве, можно получить длину строки.

Основные методы строк

Основная функциональность класса `String` раскрывается через его методы, среди которых можно выделить следующие:

- **Compare**: сравнивает две строки с учетом текущей культуры (локали) пользователя
- **CompareOrdinal**: сравнивает две строки без учета локали
- **Contains**: определяет, содержится ли подстрока в строке
- **Concat**: соединяет строки
- **CopyTo**: копирует часть строки или всю строку в другую строку
- **EndsWith**: определяет, совпадает ли конец строки с подстрокой
- **Format**: форматирует строку
- **IndexOf**: находит индекс первого вхождения символа или подстроки в строке
- **Insert**: вставляет в строку подстроку
- **Join**: соединяет элементы массива строк
- **LastIndexOf**: находит индекс последнего вхождения символа или подстроки в строке
- **Replace**: замещает в строке символ или подстроку другим символом или подстрокой
- **Split**: разделяет одну строку на массив строк
- **Substring**: извлекает из строки подстроку, начиная с указанной позиции
- **ToLower**: переводит все символы строки в нижний регистр
- **ToUpper**: переводит все символы строки в верхний регистр
- **Trim**: удаляет начальные и конечные пробелы из строки

Операции со строками

Конкатенация

Конкатенация строк или объединение может производиться как с помощью операции `+`, так и с помощью метода `Concat`:

```
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);
```

Метод `Concat` является статическим методом класса `String`, принимающим в качестве параметров две строки. Также имеются другие версии метода, принимающие другое количество параметров.

Для объединения строк также может использоваться метод `Join`:

```
string s5 = "apple";
string s6 = "a day";
string s7 = "keeps";
string s8 = "a doctor";
string s9 = "away";
string[] values = new string[] { s5, s6, s7, s8, s9 };

String s10 = String.Join(" ", values);
// результат: строка "apple a day keeps a doctor away"
```

Метод `Join` также является статическим. Используемая выше версия метода получает два параметра: строку-разделитель (в данном случае пробел) и массив строк, которые будут соединяться и разделяться разделителем.

Сравнение строк

Для сравнения строк применяется статический метод `Compare`:

```
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"

```

Данная версия метода Compare принимает две строки и возвращает число. Если первая строка по алфавиту стоит выше второй, то возвращается число меньше нуля. В противном случае возвращается число больше нуля. И третий случай - если строки равны, то возвращается число 0.

В данном случае так как символ h по алфавиту стоит выше символа w, то и первая строка будет стоять выше.

Поиск в строке

С помощью метода IndexOf мы можем определить индекс первого вхождения отдельного символа или подстроки в строке:

```

string s1 = "hello world";
char ch = 'o';
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.

Еще одна группа методов позволяет узнать начинается ли строка на определенную подстроку. Для этого предназначены методы StartsWith и EndsWith. Например, у нас есть задача удалить из папки все файлы с расширением exe:

```

string path = @"C:\SomeDir";

string[] files = Directory.GetFiles(path);

```

```

for (int i = 0; i < files.Length; i++)
{
    if (files[i].EndsWith(".exe"))
        File.Delete(files[i]);
}

```

Разделение строк

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. Например, подсчитаем количество слов в строке, разделив ее по пробельным символам:

```

string text = "И поэтому все так произошло";

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

foreach (string s in words)
{
    Console.WriteLine(s);
}

```

Задания на обработку строк:

Задание 1: написать программу согласно заданию.

№ вар.	Задача
1	В заданной строке посчитать количество русских букв «А».
2	В заданной строке поменять местами первый и последний символ строки.
3	В заданной строке вместо первого символа поставить пробел, а вместо последнего – точку.
4	В заданной строке удалить первый символ.
5	В заданной строке добавить в конец строки символ «>».
6	В заданной строке посчитать количество символов цифр (от «0» до «9»).
7	Определить, есть ли в заданной строке символ цифры. Вывести сообщение об этом.
8	В заданной строке вместо каждого пробела поставить символ «&».
9	В заданной строке удалить последний символ.
10	В заданной строке удалить все пробелы.
11	В заданной строке удалить второй и четвертый по счету символы.
12	Найти сумму кодов всех символов заданной строки.
13	Удалить первое слово заданной строки. Разделителем слов считается пробел.
14	В заданной строке посчитать количество точек и запятых.

15	В заданной строке посчитать количество круглых и квадратных скобок.
16	Взять заданную строку в кавычки вида «».
17	В конце заданной строки дописать символы «_», доведя длину строки до 25.
18	Для заданной строки определить входит ли в нее хотя бы одна русская буква.
19	Для заданной строки определить входит ли в нее хотя бы одна латинская буква.
20	В заданной строке найти месторасположение первого символа «*».
21	Удалить в заданной строке все, кроме первого и последнего символа.
22	Для заданной строки, определить какие символы встречаются чаще «!» или «?».
23	В заданной строке определить, совпадает ли количество круглых открывающихся и круглых закрывающихся скобок.
24	В заданной строке после каждой точки поставить еще одну точку.
25	В заданной строке после каждой точки поставить пробел.
26	В строке заменить каждый символ «!» символом « » (пробел).
27	Определить, входит ли в данную строку пробел.
28	В заданной строке посчитать количество сочетаний символов «-+».
29	В заданной строке удалить все русские буквы.
30	В заданной строке удалить все латинские буквы.

Задание 2: написать программу согласно заданию.

№ вар.	Задача
1	В заданной строке заменить подряд идущие пробелы на один пробел.
2	В заданной строке посчитать количество слов. Разделителем слов считается один или несколько пробелов.
3	В заданной строке заменить каждую русскую букву символом «*».
4	В заданной строке удалить все латинские буквы.
5	Дана строка. Дописать в конец строки ее длину.
6	В заданной строке дописать после каждого символа «*» символ «=».
7	В заданной строке вставить перед каждым символом «!» символ «,»
8	Заданы две одинаковые по длине строки. Построить новую строку, в которой на четных местах расположены элементы первой строки, а на нечетных – элементы второй строки.
9	В заданной строке заменить каждый пробел двумя пробелами.
10	Задана строка. Построить новую строку, в которой все символы записаны в обратном порядке.
11	В заданной строке удалить первый символ «*», который найдется строке.
12	В заданной строке удалить последний символ « » (пробел), который найдется в строке.
13	В заданной строке заменить каждый символ «№» строкой «номер по порядку».

14	В заданной строке перед каждой большой латинской буквой поставить « » (точка, пробел).
15	Разрядить пробелами заданную строку.
16	В заданной строке заменить сочетания символов «-+» символом «0».
17	В заданной строке заменить каждый символ строки следующим по коду символом. Букву «я» заменить пробелом.
18	В заданной строке поменять каждый символ строки предыдущим по таблице кодирования символом.
19	В заданной строке после каждого символа вставить число, соответствующее коду этого символа.
20	Получить из заданной строки две строки, состоящие из символов первой строки, имеющих соответственно четные и нечетные индексы.
21	Заданы две строки одинаковой длины. Построить новую строку, состоящую из символов как одной, так и другой строки, чередующихся между собой.
22	В строке после каждого слова дописать ее длину.
23	Задана строка. Получить предпоследнее слово этой строки. Разделите лем слов считаются один или несколько пробелов.
24	В заданной строке заменить четное количество подряд идущих пробелов символом «Ч», а нечетное – символом «Н».
25	Для заданной строки после каждой русской буквы поставить ее код, а после каждой латинской – символ «».
26	В заданной строке, после каждого слова вида «sin», «cos» или «log», поставить скобку «(».

Задание 3. Обработка строк с использованием оператора цикла с параметром

1. Дано предложение. Составить программу, которая печатает "столбиком" все вхождения в предложение некоторого символа.

2. Дано предложение. Вывести "столбиком" его третий, шестой и т. д. символы.

3. Дано предложение. Вывести все буквы м и н в нем.

4. Дано предложение. В нем слова разделены одним пробелом (начальные и конечные пробелы и символ "-" в предложении отсутствуют). Определить количество слов в предложении.

5. Дано предложение. В нем слова разделены одним или несколькими пробелами (символ "-" в предложении отсутствует). Определить количество слов в предложении. Рассмотреть два случая:

1) начальные и конечные пробелы в предложении отсутствуют;

2) начальные и конечные пробелы в предложении имеются.

6. Дан текст. Подсчитать общее число вхождений в него символов "+" и "-".

7. Дано предложение. Составить программу, которая выводит все вхождения в предложение двух заданных символов.
8. Дано предложение. Вывести все имеющиеся в нем буквосочетания nn.
9. Дано предложение. Вывести "столбиком" все его буквы и, стоящие на четных местах.
10. Дано предложение. Вывести "столбиком" его первый, второй, пятый, шестой, девятый, десятый и т. д. символы.
11. Дано предложение. Определить число букв o в нем.
12. Дано предложение. Определить число пробелов в нем.
13. Дано предложение. Определить число вхождений в него некоторого символа.
14. Дано предложение. Определить долю (в %) букв a в нем.
15. Дан текст. Сколько раз в нем встречается символ "+" и сколько раз символ "*"?
16. Дано предложение. Определить, сколько в нем одинаковых соседних букв.

Задание 4. Обработка строк с использованием операторов цикла с условием

1. Дано предложение. Напечатать все его символы, предшествующие первой запятой. Рассмотреть два случая:
 - 1) известно, что в предложении запятые имеются;
 - 2) в предложении запятых может не быть.
2. Дано предложение, в котором имеется несколько букв e. Найти:
 - а) порядковый номер первой из них;
 - б) порядковый номер последней из них.
3. Дано предложение. Определить, есть ли буква a в нем. В случае положительного ответа найти также порядковый номер первой из них.
4. Дано слово. Проверить, является ли оно "перевертышем" (перевертышем называется слово, читаемое одинаково как с начала, так и с конца).
5. Дан текст. Определить количество букв и в первом предложении. Рассмотреть два случая:
 - 1) известно, что буквы и в этом предложении есть;
 - 2) букв и в тексте может не быть.
6. Дана последовательность символов, в начале которой имеется некоторое количество одинаковых символов. Определить это количество. Рассмотреть два случая:

- 1) известно, что не все символы последовательности одинаковые;
- 2) все символы последовательности могут быть одинаковыми.
7. Даны два слова. Определить, сколько начальных букв первого слова совпадает с начальными буквами второго слова. Рассмотреть два случая:
 - 1) известно, что слова разные;
 - 2) слова могут быть одинаковыми.
8. Дано предложение, в котором нет символа "-". Определить количество букв о в первом слове. Учесть, что в начале предложения могут быть пробелы.
9. Дано предложение. Определить количество букв н, предшествующих первой запятой предложения. Рассмотреть два случая:
 - 1) известно, что запятые в предложении есть;
 - 2) запятых в предложении может не быть.
10. Дано предложение. Определить порядковые номера первой пары одинаковых соседних символов. Если таких символов нет, то должно быть напечатано соответствующее сообщение.
11. Дано предложение. Определить, есть ли в нем буквосочетания чу или щу. В случае положительного ответа найти также порядковый номер первой буквы первого из них.
12. Дана последовательность слов. Проверить, правильно ли в ней записаны буквосочетания жи и ши.
13. Дана последовательность слов. Проверить, правильно ли в ней записаны буквосочетания ча и ща. Исправить ошибки.
14. Дано предложение. Напечатать все символы, расположенные между первой и второй запятой. Если второй запятой нет, то должны быть напечатаны все символы, расположенные после единственной имеющейся запятой.
15. Дано предложение, в котором имеются одна буква с и одна буква т. Определить, какая из них встречается раньше (при просмотре слова слева направо).

Задание 5. Изменение исходных строк

1. Дано предложение. Все буквы е в нем заменить буквой и.
2. Дано предложение. Все пробелы в нем заменить символом " _".
3. Дано предложение. Все его символы, стоящие на четных местах, заменить буквой ы.
4. Дано предложение. Все его символы, стоящие на третьем, шестом, девятом и т. д. местах, заменить буквой а.

5. Дано предложение. Заменить в нем все вхождения буквосочетания *ах* на *ух*.
6. Дано предложение. Заменить в нем все вхождения буквосочетания *да* на *не*.
7. Дано предложение. Заменить в нем все вхождения буквосочетания *про* на *нет*.
8. Дано предложение. Заменить в нем все вхождения буквосочетания *бит* на *рог*.
9. Дано предложение. Заменить в нем все вхождения подстроки *s1* на подстроку *s2*.
10. Символьной строке *s* по ошибке вместо *опечатка* присвоено значение *очепатка*. Изменить значение *s* так, чтобы ошибки не было.
11. Дано слово. Поменять местами его вторую и пятую буквы.
12. Дано слово. Поменять местами его третью и последнюю буквы.
13. Дано слово. Поменять местами его *m*-ю и *n*-ю буквы.
14. Дано слово из четного числа букв. Поменять местами первую букву со второй, третью — с четвертой и т. д.
15. Дано слово из четного числа букв. Поменять местами его половины следующим способом: первую букву поменять с последней, вторую — с предпоследней и т. д.
16. Дано слово из 12-ти букв. Переставить в обратном порядке буквы, расположенные между второй и десятой буквами (т. е. с третьей по девятую).