# Лабораторная работа № 3.2 Функции для работы со строками

**Цель лабораторной работы**

Ознакомиться со способами инициализации строк, и организацией ввода-вывода строк, с методами класса System.String.

**Постановка задачи**

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 сроки 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:

public char this[int index] {get;}

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

**Задание на лабораторную работу**

Разработать приложение, реализующие методы класса System.String, обладающее заданной в вариантах функциональностью.

**Варианты заданий на лабораторную работу**

|  |  |
| --- | --- |
| **№** | **Задание** |
|  | Расстояние между двумя словами равной длины – это количество позиций, в которых различаются эти слова. В заданном предложении найти пару слов с максимальным расстоянием. |
|  | Отредактировать заданное предложение, удаляя из него все слова с нечетными номерами и переворачивая слова с четными номерами. |
|  | Задан текст. Переписать его таким образом, чтобы предложения шли в обратном порядке, при этом внутри предложения слова шли в обратном порядке. Каждое слово в предложении должно начинаться с заглавной буквы. Предложения в новом тексте должны заканчиваться теми же знаками препинания, что и в исходном тексте. |
|  | Задан текст. Переписать его таким образом, чтобы сначала шли все гласные буквы, затем знаки препинания, а затем согласные буквы в том порядке в каком они встречаются в тексте. |
|  | Задан текст. Удалить из него все знаки препинания и переставить в нем пробелы таким образом, чтобы длина всех слов, кроме быть может последнего, была одинаковой. |
|  | Задан текст. Поменять в нем местами две гласные, которые встречаются чаще остальных двумя согласными, которые встречаются чаще остальных. |
|  | Дана строка, содержащая текст, вывести на экран те слова которые читаются одинаково справа налево и слева направо (т.е. являются палиндромом). |
|  | Упорядочить слова заданного текста по алфавиту. |
|  | Задан текст. Удалить из него все знаки препинания. Дописать в его конец минимальное количество символов так, чтобы полученный текст был палиндромом (пробелы не учитывать). |
|  | Ввести строку, в которую могут входить только цифры и символы {', ', '; ' , '\*', '@'}. Вывести на экран сколько раз встречается последовательность символов ',@,' |
|  | Ввести строку, в которой могут встречаться только буквы и цифры. Вывести на экран такие группы, в которых цифра 1 встречается не менее 3 раз. Под группой считать такую последовательность цифр, которая заключена между буквами. |
|  | Из введенного текста вывести на экран только те слова, которые симметричны (RADAR, ANNA) |
|  | Из введенного текста вывести на экран только те слова, в которых буквы упорядочены в порядке, обратном алфавитному (ROD, SON). |
|  | Сколько чисел между n и m (n<m) состоит из нечётных цифр и сколько из различных цифр? |
|  | Задан текст, содержащий только буквы. Переписать его таким образом, чтобы сначала шли все гласные буквы, а затем согласные буквы в том порядке в каком они встречаются алфавите. |
|  | Отредактировать заданное предложение, удаляя из него все слова с, входящей в них произвольной буквой (буква задается пользователем), и переворачивая в обратном порядке слова без вхождения этой буквы. |
|  | Строка содержит произвольный текст. Проверить совпадают ли в ней самые короткие слова (короткое слово – это слово с наименьшим количеством символов). |
|  | Задан текст. Удалить из него все знаки препинания и пробелы. Дописать в его конец столько раз слово 'ой!' сколько символов было удалено. |
|  | Задан текст. Дописать к слову длинной, заданной пользователем , слово, которое вводит пользователь. Все остальные -удалить. |
|  | Из введенного текста вывести на экран только те слова, в которых буквы упорядочены в алфавитном порядке (dor, nos). |
|  | Ввести строку, в которой могут встречаться только буквы и цифры. Вывести на экран такие группы, в которых последовательность букв 'ок' встречается не менее 2 раз. Под группой считать такую последовательность букв, которая заключена между цифрами. |
|  | Будем считать расстоянием между двумя одинаковыми словами – количество символов между ними. В заданном предложении найти расстояние между всеми соседними одинаковыми словами. |
|  | Даны две строки А и В. Составьте программу, выводящую на экран совпадающие подстроки длиной 2. |
|  | Строка содержит произвольный текст. Проверить совпадают ли в ней самые короткие слова (короткое слово – это слово с наименьшим количеством символов). |
|  | Задан текст. Удалить из него самое длинное и самое короткое слово. Во всех остальных словах вхождение буквы 'А' увеличить вдвое. |
|  | Дан текст. Найти самое короткое слово, заменить его словом «вот!». В остальных словах текста заменить первые символы на «вот!», оставшиеся символы – не изменять. |
|  | Будем считать расстоянием между двумя словами – количество символов между этими словами. В заданном предложении найти расстояние между первым и последним вхождениями слова, введенного пользователем. |

# Содержание пояснительной записки

1. Постановка задачи. Приводится теоретический материал, использованный при написании приложения.

2. Формулировка задания и вариант. Приводится задание на лабораторную работу и вариант этого задания.

3. Описание выполняемых действий. Необходимо привести описание последовательности разработки программы, реализации используемых методов, алгоритмов, блок-схем.

4. Анализ результатов. Привести анализ входных и выходных данных. Показать результаты выполнения программного кода. Предоставить скриншоты обработки тестовых примеров. Сделать выводы.

5. Листинг программы. Привести листинг разработанного программного кода, содержание файлов входных и выходных данных.

# Используемое программное обеспечение

1. Среда программирования MS Visual Studio Community 2017 (Свободно распространяемое программное обеспечение (в учебных целях));
2. Microsoft Office Standard 2007 (Open License: 42267924);
3. Браузер (Свободно распространяемое программное обеспечение).