**Лабораторная работа № 6. Строки**

**1 Цель работы**

Изучить методы алгоритмизации и программирования с использованием строк.

**2 Порядок выполнения работы**

Получить задание для выполнения лабораторной работы (раздел 5) согласно своему варианту. Разработать и отладить программу. Составить отчет о лабораторной работе и защитить её у преподавателя.

**3 Содержание отчета**

* наименование и цель работы;
* задание на лабораторную работу согласно варианту;
* схема алгоритма, текст программы на алгоритмическом языке;
* результаты работы программы.

**4 Краткая теория**

В C# для работы со строками используется тип string. Строки представлены в виде последовательности символов Unicode. Максимальный размер одной строки может составлять в памяти 2 ГБ, или около 1 миллиарда символов.

**4.1 Создание строки**

Фактически, любая строка является массивом символов char. Существует множество вариантов создания строк, в коде ниже представлены самые популярные:

using System.Runtime.CompilerServices;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

using System.Xml.Linq;

namespace ExampleApp

{

internal class Program

{

static void Main(string[] args)

{

string firstString = "hello";

string secondString = new String(new char[] { 'w', 'o', 'r', 'l', 'd' });

Console.WriteLine(firstString+ ' ' + secondString);

}

}

}

Стоить помнить, что есть и другие способы инициализировать строку, например, при помощи класса но про них стоит говорить только в контексте объектно-ориентированного программирования, поэтому в коде они не представлены. Например, при помощи класса StringBuilder. Это связано с такой особенностью любой строки, как иммутабельность или ее неизменяемость. При любом изменении строки фактически создается ее измененная копия, а старая строка рано или поздно будет удалена из памяти.

**4.2 Работа со строкой как с массивом**

Так как фактически любая строка представляет из себя набор символов, то в данном контексте ее можно рассматривать как массив символов. В коде ниже представлено обращение к первому символу, получение длины строки (размеров массива) и посимвольный вывод строки.

Ниже представлен код, который позволяет работать со строкой как с массивом символов:

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Reflection.Emit;

using System.Runtime.CompilerServices;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

using System.Xml.Linq;

namespace ExampleApp

{

internal class Program

{

static void Main(string[] args)

{

string str = "hello world";

char firstSymbol = str[0];

int stringLength = str.Length;

for (int i = 0; i < str.Length; i++)

{

Console.WriteLine(str[i]);

}

Console.WriteLine($"Длина строки {stringLength} символов");

Console.WriteLine($"Первый символ: {firstSymbol}");

}

}

}

Строки можно сравнивать между собой при помощи операторов сравнения. Здесь стоит отметить, что в некоторых других языках программирования подобные сравнения устроены по-разному, в виду того, что строки являются ссылочными типами данных, а не типами-значениями:

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Reflection.Emit;

using System.Runtime.CompilerServices;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

using System.Xml.Linq;

namespace ExampleApp

{

internal class Program

{

static void Main(string[] args)

{

string message = "hello world";

string anotherMessage = "Hello world";

string copyFirstMessage = "hello world";

Console.WriteLine(message == anotherMessage); // False, т.к. и ссылки и сообщения разные

Console.WriteLine(message == copyFirstMessage); // True, т.к. сообщения совпадают, но ссылки могут быть на разные объекты

}

}

}

Про ссылки на объекты подробно будет рассказано в контексте объектно-ориентированного программирования.

**4.3 Основные методы, доступные для строк**

Кроме действий, аналогичных массивам, для строк доступны свои действия, которые описаны ниже:

1. Compare: сравнивает две строки с учетом текущей культуры (локали) пользователя;
2. CompareOrdinal: сравнивает две строки без учета локали;
3. Contains: определяет, содержится ли подстрока в строке;
4. Concat: соединяет строки;
5. CopyTo: копирует часть строки, начиная с определенного индекса в массив;
6. EndsWith: определяет, совпадает ли конец строки с подстрокой;
7. Format: форматирует строку;
8. IndexOf: находит индекс первого вхождения символа или подстроки в строке;
9. Insert: вставляет в строку подстроку;
10. Join: соединяет элементы массива строк;
11. LastIndexOf: находит индекс последнего вхождения символа или подстроки в строке;
12. Replace: замещает в строке символ или подстроку другим символом или подстрокой;
13. Split: разделяет одну строку на массив строк;
14. Substring: извлекает из строки подстроку, начиная с указанной позиции;
15. ToLower: переводит все символы строки в нижний регистр;
16. ToUpper: переводит все символы строки в верхний регистр;
17. Trim: удаляет начальные и конечные пробелы из строки.

Concat и Join используются для склеивания двух строк в одну, аналогично оператору +. Отличие состоит в том, что Concat склеивает две строки в одну, а Join может склеивать множество строк, добавляя между ними определенный символ-разделитель.

Compare используется для сравнения строк, как оператор ==, но с учетом конкретных языковых особенностей. Также этот метод позволяет определить, какая из строк расположена раньше в лексилогическом порядке.

Методы IndexOf, LastIndexOf, StartsWith, EndsWith можно определить, имеется ли в исходной строке искомая подстрока.

Метод Split работает как Join но в обратную сторону, он позволяет разбить одну большую строку на множество подстрок по символу-разделителю, например, по наличию символа пробел.

Методы Trim, TrimStart и TrimEnd позволяют образать строку от ненужных символов в начале, в конце или и в начале и в конце строки сразу. Если необходимо выделить некоторую строку, тогда используется метод Substring.

Для вставки подстроки в строку в определенную позицию, используется метод Insert. Если же необходимо заменить одну подстроку на другую в строке, существует метод Replace, а для удаления подстроки из строки используется метод Remove.

И наконец, существуют способы привести все символы в строке к нижнему или верхнему регистру при помощи методов ToLower и ToUpper соответственно.

**5 Варианты заданий для выполнения работы**

1. Для заданной последовательности символов, за которой следует точка (в сам текст точка не входит), переменной *f* присвоить значение *true,* если буква *а* встречает­ся чаще, чем буква *b,* и присвоить значение *false* в противном случае;
2. Для заданной последовательности символов, за которой следует точка (в сам текст точка не входит), подсчитать, сколько раз встречается слово *key.*
3. Для заданной последовательности символов, за которой следует точка (в сам текст точка не входит), проверить, правильно ли в данной последовательности рас­ставлены круглые скобки.
4. Для заданной последовательности символов, за которой следует точка (в сам текст точка не входит), определить, является ли данная последовательность символов правильной записью целого числа (возможно, со знаком).
5. Последовательность символов содержит слова из латин­ских букв, соседние слова отделены друг от друга запятой, за послед­ним символом — точка. Определить общее число символов.
6. Последовательность символов содержит слова из латин­ских букв, соседние слова отделены друг от друга запятой, за послед­ним символом — точка. Определить количество слов, которые начинаются с буквы *а.*
7. Последовательность символов содержит слова из латин­ских букв, соседние слова отделены друг от друга запятой, за послед­ним символом — точка. Определить количество слов, которые оканчиваются буквой *w.*
8. Последовательность символов содержит слова из латин­ских букв, соседние слова отделены друг от друга запятой, за послед­ним символом — точка. Определить количество слов, которые начинаются и оканчива­ются одной и той же буквой.
9. Напишите программу, которая принимает на входе текст, содержащий последовательность заглавных и строчных букв, затем печатает этот текст только литерами заглавных букв.
10. В данном тексте найти наибольшее количество цифр, идущих в нём подряд.
11. В данном тексте определить наличие символов, отличных от букв и пробела.
12. В данном тексте заменить буквы N, D, и Y соответственно на Н, Д, и У.
13. Для каждого символа данного текста указать, сколько раз он встречается в тексте.
14. Выяснить, является ли данный текст десятичной записью целого числа.
15. Напечатать в алфавитном порядке все различные русские буквы, входящие в данный текст.
16. Для данного текста проделайте следующие действия: для каждого из слов укажите, сколько раз оно встречается сре­ди слов текста.
17. Для данного текста проделайте следующие действия: найдите слова, начинающиеся и оканчивающиеся одной и той же буквой.
18. Для данного текста проделайте следующие действия: найдите самое длинное слово.
19. Для данного текста проделайте следующие действия: укажите слова, в которых доля гласных максимальна.
20. Для данного текста проделайте следующие действия: удалите заданное слово.
21. Для данного текста проделайте следующие действия: удалите слова, встречающиеся более одного раза.
22. Проверьте, имеется ли в заданном тексте баланс открываю­щих и закрывающих скобок.
23. Даны два текста. Для них найдите самое длинное общее слово обоих тестов.
24. Даны два текста. Для них найдите самое короткое из слов первого текста, которого нет во втором.
25. Замените в данном тексте каждую из групп одинаковых слов, стоящих рядом, одной точкой.
26. Замените в тексте каждую точку троеточием.
27. Если в данном тексте нет символа «\*», то оставьте его без изменений, иначе каждую из букв, предшествующих вхождению символа «\*», необходимо заменить символом «/».
28. Если в данном тексте есть буквы, то каждый из символов, следующих за первой группой букв, замените точками. В противном случае текст оставьте без изменений.
29. Найдите количество предложений в заданном тексте.
30. Дан непустой текст из заглавных русских букв, за которым следует точка. Определить, упорядочены ли буквы по алфавиту.