

ЛАБОРАТОРНАЯ РАБОТА №4

Курс «Основы программирования»

Тема: Работа со строками в C#. Знакомство с регулярными выражениями.

Цель: Научиться работать с типом string на языке C#, а также со строкой как с массивом символов; научиться писать простейшие регулярные выражения для поиска подстроки в строке.

Темы для предварительной проработки ^[устно]:

- Типы System.String и System.Text.StringBuilder в языке C# .NET.
- Регулярные выражения.

Общие задания ^[код] :

1. Ввести с клавиатуры текст предложения, завершить точкой. Вывести на консоль все символы, которые входят в этот текст ровно по одному разу. Решить задачу 2 способами: через обработку строки как массива символов и с помощью методов класса string.
2. Ввести с клавиатуры текст предложения, завершить точкой. Сформировать новую строку на основе исходной, в которой после каждого слова в скобках указать номер слова в предложении (слова разделяются запятыми, пробелами или тире). Например, если введено «Донецк – прекрасный город», результирующая строка должна выглядеть так: «Донецк(1) – прекрасный(2) город(3)». Решить задачу 2 способами: через обработку строки как массива символов и с помощью методов класса string.
3. Ввести текст из нескольких слов, завершить точкой. Сформировать новую строку на основе исходной путем перестановки слов в обратной последовательности. Решить задачу 2 способами: через обработку строки как массива символов и с помощью методов классов string и StringBuilder.
4. Ввести с клавиатуры 7 строк, занести их в массив. Вывести все строки, в которых содержится хотя бы одно слово, оканчивающееся на “.com” (регистр символов не важен; слова разделяются пробелами, запятыми или точками). Также вывести номер строки, содержащей наименьшее число пробелов. Решить 2 способами: через обработку строки как массива символов и с помощью методов класса string.
5. Ввести с клавиатуры текст. Программно найти в нем и вывести отдельно все слова, которые начинаются с большого латинского символа (от A до Z) и заканчиваются 2 цифрами, например «Petr93», «Johnny70», «Service02». Решить 2 способами: через обработку строки как массива и с помощью *регулярных выражений*.
6. Ввести строку вида « 15 + 36 = 51 » (количество пробелов может быть разным, числа – целые и могут быть отрицательны). С помощью *регулярных выражений* разобрать эту строку и занести в переменные типа int оба операнда и сумму. Вывести все переменные на консоль.
7. Дан треклист – массив из 10 строк следующего вида:

1. Gentle Giant – Free Hand [6:15]
2. Supertramp – Child Of Vision [07:27]
3. Camel – Lawrence [10:46]
4. Yes – Don't Kill The Whale [3:55]
5. 10CC – Notell Hotel [04:58]
6. Nektar – King Of Twilight [4:16]
7. The Flower Kings – Monsters & Men [21:19]
8. Focus – Le Clochard [1:59]
9. Pendragon – Fallen Dream And Angel [5:23]
10. Kaipa – Remains Of The Day (08:02)

Написать программу, которая обрабатывает весь треклист, суммирует время звучания песен и выводит результат на экран, а также отображает самую длинную и самую короткую песню в списке и пару песен с минимальной разницей во времени звучания.

Индивидуальные задания ^[КОД] :

1. Написать программу, позволяющую шифровать и расшифровывать строки символов на основе 3 симметричных алгоритмов шифрования (*прилож. А*).
2. Написать программу обработки текста, в соответствии с вариантом. Решить задачу 2 способами: через обработку строки как массива символов и с помощью методов классов string и/или StringBuilder.
3. Написать регулярные выражения для поиска подстроки в строке по правилу или шаблону, в соответствии с вариантом.

Контрольные вопросы ^[ОТЧЕТ] :

1. Как в C# .NET может быть представлена строка символов?
2. Кратко опишите особенности и основные методы класса String.
3. Что делают методы split() и join()?
4. Укажите особенности сравнения строк в C# .NET.
5. Что такое абстрактное синтаксическое дерево? Дерево разбора?
6. Приведите определение и пример РБНФ-выражения.
7. Что такое регулярные выражения? Для чего они используются?
8. Опишите основные синтаксические элементы регулярных выражений.
9. Укажите разницу между «жадными», «ленивыми» и «собственническими» квантификаторами в регулярных выражениях.

Рекомендуемые источники:

- [1] Шилдт Г. C# 4.0. Полное руководство. – М.: Издательский дом «Вильямс», 2011. – 1056с.
- [2] Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: Построение и анализ. – М.: МЦНМО, 1999. – 960с.
- [3] Ахо А., Хопкрофт Д., Ульман Д. Структуры данных и алгоритмы. – М.: Издательский дом «Вильямс», 2001. – 384с.
- [4] Кнут Д. Искусство программирования, том 1. Основные алгоритмы. – М.: Издательский дом «Вильямс», 2000. – 720с.
- [5] Златопольский Д.М. Сборник задач по программированию. – СПб.: БХВ-Петербург, 2007. – 240с.

Приложение А. Варианты индивидуальных заданий.

Задание 1*.

№	Симметричный алгоритм шифрования
1	Шифр Атбаш (http://ru.wikipedia.org/wiki/Атбаш)
2	Шифр Вернама (http://ru.wikipedia.org/wiki/Шифр_Вернама)
3	Шифр простой одинарной перестановки (http://ru.wikipedia.org/wiki/Перестановочный_шифр)
4	Шифр Полибия (http://ru.wikipedia.org/wiki/Квадрат_Полибия)
5	Шифр Гронсфельда (http://ru.wikipedia.org/wiki/Полиалфавитный_шифр)
6	Книжный шифр (http://ru.wikipedia.org/wiki/Книжный_шифр)
7	Шифр Цезаря (http://ru.wikipedia.org/wiki/Шифр_Цезаря)
8	Шифр Виженера (http://ru.wikipedia.org/wiki/Полиалфавитный_шифр)
9	Шифр Плейфера (http://ru.wikipedia.org/wiki/Шифр_Плейфера)
10	Шифр Тритемиуса (http://ru.wikipedia.org/wiki/Шифр_Тритемиуса)

Вариант	Алгоритмы шифрования
1	1, 2, 3
2	4, 5, 6
3	7, 8, 9
4	1, 4, 10
5	2, 5, 9
6	3, 6, 8
7	4, 7, 8
8	5, 6, 10
9	1, 3, 7
10	2, 7, 10

Задание 2.

Вариант 1.

Ввести с клавиатуры текст. Сформировать новую строку из исходной путем удаления из нее всех слов, которые начинаются с того же символа, которым и заканчиваются (независимо от регистра).

Вариант 2.

Ввести с клавиатуры текст. Сформировать новую строку из исходной путем удаления из нее всех слов, которые не могут быть использованы в качестве названий переменных в C#.

Вариант 3.

Ввести с клавиатуры текст. Сформировать новую строку из исходной путем замены в ней всех слов, которые в своей середине содержат «+», на слово «CONCAT».

Вариант 4.

Ввести с клавиатуры текст. Сформировать новую строку из исходной путем обрезания каждого слова до длины 8 (если длина слова меньше 8, то заполнить оставшиеся символы знаком «*»).

Вариант 5.

Ввести с клавиатуры латинский текст. Сформировать новую строку из исходной путем замены всех больших букв в коротких словах (менее 6 букв) на символ «_».

Вариант 6.

Ввести с клавиатуры текст. Сформировать новую строку из исходной путем замены всех слов “AND” на “&” и “OR” на “|” (независимо от регистра символов).

Вариант 7.

Ввести с клавиатуры текст. Сформировать новую строку из исходной путем удаления из нее всех слов, которые и начинаются, и заканчиваются на один из символов a, o, e, i, u.

Вариант 8.

Ввести с клавиатуры текст. Сформировать новую строку из исходной путем замены всех цифр от 1 до 5 соответствующими словами – one, two, three, four, five.

Вариант 9.

Ввести с клавиатуры текст. Сформировать новую строку из исходной путем замены слов one, two, three, four, five (если встречаются в тексте) на соответствующие цифры 1,2,3,4,5.

Вариант 10.

Ввести с клавиатуры текст. Сформировать новую строку из исходной путем удаления из нее всех слов, которые содержат пары подряд идущих символов (например, sleep, parrot и т.д.).

Задание 3.

Вариант 1.

Найти в тексте все даты – подстроки вида «11-05-2014».

Вариант 2.

Найти в тексте все значения времени – подстроки вида «23:15:59».

Вариант 3.

Найти в тексте все IPv4-адреса – подстроки вида «192.168.5.48».

Вариант 4.

Найти в тексте все строки вида «*type* *x* = *value*», где *type* – это тип (может принимать значение int, short или byte), *x* – любое слово, *value* – любое число.

Вариант 5.

Найти в тексте все номера телефонов – подстроки вида «(000)1112233» или «(000)111-22-33».

Вариант 6.

Найти в тексте все строки вида «*x*: *type* [*N*]», где *type* – это тип (может принимать значение int, short или byte), *x* – любое слово, *N* – любое положительное целое число.

Вариант 7.

Найти в тексте все «смайлы» – подстроки вида «:)», «:-)», «)))» (количество скобок может быть любым, начиная с 1).

Вариант 8.

Найти в тексте все логические выражения – подстроки вида «*x* && *y*», «*x* & *y*», где *x* и *y* – любые слова. Количество пробелов может быть также любым.

Вариант 9.

Найти в тексте все донецкие почтовые индексы – подстроки вида «83000, Донецк» (первые 2 символа строго закреплены).

Вариант 10.

Найти в тексте все полные имена директорий Windows – подстроки вида «C:\Dir\SubDir3».