

Лабораторная работа 7. СТРОКИ И РЕГУЛЯРНЫЕ ВЫРАЖЕНИЯ

ЦЕЛЬ РАБОТЫ: Изучение классов языка программирования C# для работы со строками:

1. Работа с функциями класса System.String и класса System.Text.StringBuilder
2. Работа с регулярными выражениями. Поиск в тексте фрагментов по определенному шаблону

ПОДГОТОВКА К РАБОТЕ: Изучить методы классов System.String, System.Text.StringBuilder и System.Text.RegularExpressions.

1. Примеры программ для работы со строками

Пример программы для работы с классом String

Рассмотрим пример программы шифрования методом Цезаря (модификация метода). Исходная строка модифицируется следующим образом – каждый символ заменяется на другой с ASCII-кодом, следующим в алфавитном порядке.

Последовательность действий:

1. Создайте приложение оконное приложение, изображенное на рисунке 1

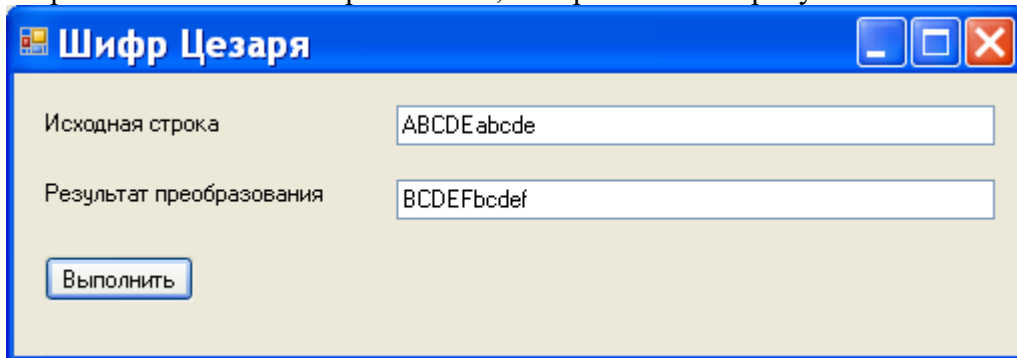


Рис. 1. Пример программы шифрования по методу Цезаря

Список используемых элементов управления приведен в таблице 1.

Табл.1. Список используемых элементов управления

Элемент управления	Класс	Описание
Label1	Label	Метка «Исходная строка»
Label2	Label	Метка «Результат преобразования»
textBox1	TextBox1	Окно ввода исходной строки
textBox2	TextBox2	Окно вывода результата
Button1	Button	Командная кнопка «Выполнить»

2. Для события Click кнопки Button1 запишите следующий программный код:

```
private void button1_Click(object sender, EventArgs e)
```

```
{  
    string s = textBox1.Text;  
    for (int i='z';i>=(int)'a';i--)  
    {  
        char old1=(char) i;  
        char new1=(char) (i+1);  
        s=s.Replace(old1,new1);  
    }  
    for (int i='Z';i>=(int)'A';i--)  
    {  
        char old1=(char) i;  
        char new1=(char) (i+1);  
        s=s.Replace(old1,new1);  
    }  
    textBox2.Text=s;  
}
```

Класс String весьма мощный класс, но он является неизменяемым (о чем говорилось в разделе 1). То есть если однажды строковый объект инициализирован, он уже не может быть изменен. Методы и операции, модифицирующие содержимое строк, на самом деле создают новые строки, копируя при необходимости старые. В данном примере метод Replace() при вызове каждый раз создает новую строку. Следовательно, в результате такого процесса шифрования появляются строковые объекты, которые будут храниться в куче до тех пор пока сборщик мусора не уничтожит их.

Для того чтобы справиться с этой проблемой и предусмотрен класс StringBuilder.

Пример работы с классом StringBuilder

Работу с классом StringBuilder рассмотрим на предыдущем примере. Оконное приложение изображено на рисунке 3.1, а список элементов управления приведен в таблице 3.1.

Программный код для события Click кнопки Button1 имеет следующий вид:

```
private void button1_Click(object sender, EventArgs e)
{
    StringBuilder s = new StringBuilder(textBox1.Text);
    for (int i = 'z'; i >= (int)'a'; i--)
    {
        char old1 = (char)i;
        char new1 = (char)(i + 1);
        s = s.Replace(old1, new1);
    }
    for (int i = 'Z'; i >= (int)'A'; i--)
    {
        char old1 = (char)i;
        char new1 = (char)(i + 1);
        s = s.Replace(old1, new1);
    }
    textBox2.Text = s.ToString();
}
```

В данном программном фрагменте используется метод StringBuilder.Replace(), который выполняет ту же работу, что и String.Replace(), но без копирования строки.

Пример работы с регулярными выражениями

Рассмотрим пример программы, выполняющей подсчет количества слов в текстовом файле.

Последовательность действий:

3. Создайте приложение оконное приложение, изображенное на рисунке 2.

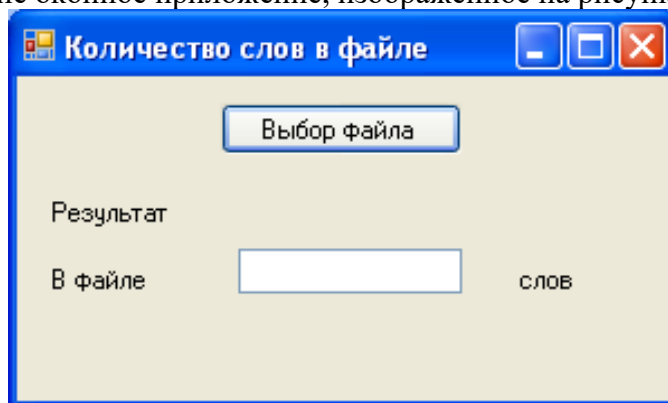


Рис..2. Пример приложения

Список используемых элементов управления приведен в таблице 2.

Табл.2. Список используемых элементов управления

Элемент управления	Класс	Описание
Button1	Button	Командная кнопка «Выбор файла»
Label1	Label	Метка «Результат»
Label2	Label	Метка «слов»
Label3	Label	Метка «в файле»
openFileDialog1	openFileDialog	Невидимый элемент управления для выбора текстового файла

- Для работы с файлами и регулярными выражениями добавьте пространство имен `using System.IO` и `using System.Text.RegularExpressions`;
- Для события Click кнопки Button1 запишите следующий программный код:

```
private void button1_Click(object sender, EventArgs e)
{
    // Диалог выбора файла
    if(openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        // Открытие выбранного файла
        StreamReader f = new StreamReader(openFileDialog1.FileName);
        String pattern = @"\"b(\w+)\";
        String s;
        int total = 0;
        Regex r = new Regex(pattern);
        // Чтение выбранного файла по строкам
        while ((s = f.ReadLine()) != null)
        {
            Match m = r.Match(s);
            // Поиск регулярного выражения в очередной строке
            while (m.Success)
            {
                total++;
                m = m.NextMatch();
            }
        }
        f.Close();
        textBox1.Text = Convert.ToString(total);
    }
}
```

2 Задания для самостоятельной работы

1) Задания для работы со строками (использовать класс String или StringBuilder)

- Ввести текст, состоящий из нескольких предложений. Вывести каждое слово в обратном порядке.
- Написать программу, которая вводит текст, состоящий из нескольких предложений, и выводит на экран только предложения, содержащие введенное с клавиатуры слово.
- Написать программу, которая вводит текст, состоящий из нескольких предложений, и выводит на экран только строки, содержащие двузначные числа.
- Ввести текст. Вывести слова, которые отличны от последнего слова и в них нет повторяющихся букв

5. Написать программу, которая вводит текст, состоящий из нескольких предложений, и выводит на экран слова, начинающиеся с гласных букв.
6. Написать программу, которая вводит текст, состоящий из нескольких предложений, и определяет, сколько в нем слов, состоящих не из трех букв.
7. Написать программу, которая вводит текст, состоящий из нескольких предложений, и выводит его на экран, меняя местами каждые два соседних слова.
8. Написать программу, которая вводит текст, состоящий из нескольких предложений, и определяет, сколько в нем слов, состоящих не более чем из четырех букв.
9. Ввести текст. Вывести все различные слова.
10. Написать программу, которая вводит текст, состоящий из нескольких предложений, и выводит его на экран, после каждого предложения добавляя, сколько раз встретилось в нем введенное с клавиатуры слово
11. Ввести текст. Вывести слова, встречающиеся в тексте по два раза.
12. Написать программу, которая вводит текст, состоящий из нескольких предложений, и выводит на экран только предложения, состоящие из заданного количества слов.
13. Дан текст. Определить количество слов, в которых содержится хотя бы одна заданная буква.
14. Написать программу, которая вводит текст, состоящий из нескольких предложений, и выводит на экран слова текста, начинающиеся и оканчивающиеся на гласные буквы.
15. Ввести текст. Определить количество слов, начинающихся и оканчивающихся одной и той же буквой.
16. Ввести текст. Определить количество слов, являющихся палиндромами.
17. Написать программу, которая вводит текст, состоящий из нескольких предложений, находит самое длинное слово и определяет, сколько раз оно встретилось в тексте.
18. Ввести текст. Вывести все слова, предварительно удалив из них все предыдущие вхождения последней буквы.
19. Даны две строки. Определить, можно ли, переставляя символы в первой строке, получить вторую строку.
20. Ввести текст. Определить количество слов, которые содержат заданное число вхождений заданной буквы

2) Задания на регулярные выражения

Используя регулярные выражения, напишите следующие приложения:

1. Дан текстовый файл. Вывести все слова, состоящие только из цифр.
2. Дан текстовый файл, Вывести на экран все встречающиеся в данном файле даты в формате DD.MM.YYYY.

3. Дан текстовый файл, имеющий структуру «Фамилия И.О. – улица – номер дома – квартира – номер телефона». Вывести на экран фамилии всех абонентов, проживающих на улице Мира.
4. Дан текстовый файл. Вывести все двузначные числа, содержащиеся в файле.
5. Дан текстовый файл, Вывести на экран все встречающиеся в данном файле номера сетевых адаптеров в формате НН-НН-НН-НН-НН-НН (где Н-шестнадцатеричная цифра).
6. Дан текстовый файл, Вывести на экран все встречающиеся в данном файле доменные имена (имена в формате www.xxx.xxx.ru , где x-любой символ латинского алфавита, количество символов может быть любым).
7. Дан текстовый файл. Вывести на экран все встречающиеся в данном файле IP адреса.
8. Дан текстовый файл, Вывести на экран все встречающиеся в данном файле даты в формате YYYY-MM-DD.
9. Дан текстовый файл. Используя регулярные выражения, в каждой строке удалить все символы, стоящие перед группой из шести цифр.
10. Дан текстовый файл. Составить регулярное выражение для проверки правильности адреса электронной почты.
11. Дан текстовый файл. Составить регулярное выражение, определяющее, является ли данная строка шестнадцатеричным идентификатором цвета в HTML (где #FFFFFF- для белого, #000000 для черного, #FF0000 для красного и.т.д.).
12. Дан текстовый файл, Вывести на экран все встречающиеся в данном файле номера автомобилей в формате xYyyxx, где x – буква, y – цифра.
13. Дан текстовый файл. Вывести на экран все встречающиеся в данном файле адреса электронной почты.
14. Дан текстовый файл. Составить регулярное выражение для проверки, является ли заданная строка IP адресом.
15. Дан текстовый файл, Вывести на экран все встречающиеся в данном файле двойные фамилии (фамилия может состоять только из букв, между которыми расположено тире, первые буквы – заглавные).
16. Дан текстовый файл. Вывести на экран все встречающиеся в данном файле номера мобильных телефонов.
17. Дан текстовый файл. Получить все подстроки между круглыми и квадратными скобками.
18. Дан текстовый файл f. Выяснить, верно ли, что в данном файле больше групп букв, чем групп цифр.
19. Дан текстовый файл, имеющий структуру «Фамилия И.О. – рост см – вес кг». Вывести на экран фамилии всех лиц, чей рост превышает 190 см.
20. Дан текстовый файл, имеющий структуру «Фамилия И.О. – рост см – вес кг». Вывести на экран фамилии всех лиц, чей вес превышает 100 кг.

ПРИЛОЖЕНИЕ А. Справочные данные для лабораторной работы № 7

Табл. А.1. Регулярные выражения. Классы символов

Класс символов	Описание	Пример
.	Любой символ, кроме \n	Выражение <code>c.t</code> соответствует фрагментам <code>cat</code> , <code>cut</code> , <code>clt</code> , <code>c{t}</code> и т. д.
[]	Любой одиночный символ из последовательности, записанной внутри скобок. Допускается использование диапазонов символов	Выражение <code>c[au]t</code> соответствует фрагментам <code>cat</code> , <code>cut</code> и <code>clt</code> , а выражение <code>c[a-z]t</code> — фрагментам <code>cat</code> , <code>cbt</code> , <code>cct</code> , <code>celt</code> , ..., <code>czt</code>
[^]	Любой одиночный символ, не входящий в последовательность, записанную внутри скобок. Допускается использование диапазонов символов	Выражение <code>c[^au]t</code> соответствует фрагментам <code>cbt</code> , <code>c2t</code> , <code>cXt</code> и т. д., а выражение <code>c[^a-zA-Z]t</code> — фрагментам <code>sit</code> , <code>clt</code> , <code>cЧt</code> , <code>c3t</code> и т. д.
\w	Любой алфавитно-цифровой символ, то есть символ из множества прописных и строчных букв и десятичных цифр	Выражение <code>c\wt</code> соответствует фрагментам <code>cat</code> , <code>cut</code> , <code>clt</code> , <code>cЮt</code> и т. д., но не соответствует фрагментам <code>c{t}</code> , <code>c;t</code> и т. д.
\W	Любой <i>не</i> алфавитно-цифровой символ, то есть символ, не входящий в множество прописных и строчных букв и десятичных цифр	Выражение <code>c\Wt</code> соответствует фрагментам <code>c{t}</code> , <code>c;t</code> , <code>c t</code> и т. д., но не соответствует фрагментам <code>cat</code> , <code>cut</code> , <code>clt</code> , <code>cЮt</code> и т. д.
\s	Любой пробельный символ, например символ пробела, табуляции (<code>\t</code> , <code>\v</code>), перевода строки (<code>\n</code> , <code>\r</code>), новой страницы (<code>\f</code>)	Выражение <code>\s\w\w\w\s</code> соответствует любому слову из трех букв, окруженному пробельными символами
\S	Любой <i>не</i> пробельный символ, то есть символ, не входящий в множество пробельных	Выражение <code>\s\S\S\s</code> соответствует любым двум непробельным символам, окруженным пробельными
\d	Любая десятичная цифра	Выражение <code>c\dт</code> соответствует фрагментам <code>c1t</code> , <code>c2t</code> , ..., <code>c9t</code>
\D	Любой символ, не являющийся десятичной цифрой	Выражение <code>c\Dт</code> не соответствует фрагментам <code>c1t</code> , <code>c2t</code> , ..., <code>c9t</code>

Табл. А.2. Регулярные выражения. Уточняющие метасимволы

Метасимвол	Описание
^	Фрагмент, совпадающий с регулярным выражением, следует искать только в начале строки
\$	Фрагмент, совпадающий с регулярным выражением, следует искать только в конце строки