

## Лабораторная работа №8

**Тема:** Создание и применение регулярных выражений при разработке программ.

**Цель работы:** Формирование умений и навыков в разработке и использовании регулярных выражений.

**Время на выполнение работы:** 2 часа

Этапы работы:

- I. Ознакомиться с теоретическими сведениями.
- II. Выполнить задания, предложенные преподавателем.
- III. Ответить на контрольные вопросы.

### *I. Краткие теоретические сведения*

Теоретические сведения необходимы для задания 2.

Метод Replace класса Regex позволяет выполнять замену фрагментов текста. Определено несколько перегруженных версий этого метода. Вот как выглядит пример простейшего применения метода в его статическом варианте, заменяющего все вхождения символа \$ символами у.е.

```
string text      = "Салат - $4, борщ -$3. одеколон - $10.";
string text1     = Regex.Replace(text, @"\$", "у.е.");
```

Другие версии метода позволяют задавать любые действия по замене с помощью делегата MatchEvaluator, который вызывается для каждого вхождения фрагмента, совпавшего с заданным регулярным выражением.

### Структура HTML-документа

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
//декларация типа документа, представляет собой объявление, что этот
документ будет определённого типа. Эта декларация позволяет использовать
любые элементы и атрибуты языка HTML.
<html >//необязательные теги начало документа
<head>// заголовок
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251" />
```

```
<title>Название заголовка</title>
</head>
<body> //тело
</body>
</html>
```

### Основное форматирование текста.

*Физическое форматирование:*

1. Тег **<B>** и обязательный парный ему **</B>** - создает жирный текст.  
Атрибуты *title* - показывает текст в виде всплывающей подсказки

2. Тег **<BIG>** и обязательный парный ему **</BIG>** - выводит более крупный текст. Атрибуты *title* - показывает текст в виде всплывающей подсказки

3. Тег **<BLOCKQUOTE>** и обязательный парный закрывающий тег **</BLOCKQUOTE>** - предназначен для включения в документ длинных цитат, создавая отступы с обеих сторон текста и отделяясь от остального текста пустыми строками.

4. Тег **<I>** и обязательный парный ему **</I>** - создает наклонный текст

5. Тег **<U>** и обязательный парный ему **</U>** - указывает, что текст должен быть подчеркнут

6. Тег **<BR>** не требует парного закрывающего тега - вставляет перевод строки

7. Тег **<Font>** и обязательный парный ему **</Font>** - определяет выводимый шрифт. Атрибуты:

*size* - размер текста в пределах от 1 до 7, где 1 - самый мелкий шрифт. По умолчанию равен 3.

*color* - устанавливает цвет текста, используя значение цвета

*face* - название шрифта

Примечание: Атрибуты можно объединять

8. Тег **<Hx>** и обязательный парный ему **</Hx>** - служит для создания заголовка. Всего существует 6 видов заголовков - от H1 до H6. Тегу **<H1>** соответствует самый большой заголовок, тегу **<H6>** - самый маленький

Атрибуты :

*align* - выравнивает заголовок в соответствии со следующими значениями

*center* - по центру

*left* - по левому краю

*right* - по правому краю

*justify* – по ширине

*title* - всплывающая подсказка

9. Тег **<HR>**. Закрывающий тег не требуется - добавляет в HTML документ горизонтальную линию. Перед и после линии помещается пустая строка.

Атрибуты:

*size* - устанавливает высоту(толщину) линии

*width* -устанавливает ширину линии в пикселах или процентах

*noshade* -создает линию без тени

*color* -задает линии определенный цвет

10. Тег **<P>** и необязательный парный ему **</P>** -создает новый параграф. Два или более тега **<P>**, идущих подряд, заменяются одним.

Атрибуты :

*align* -выравнивает заголовок в соответствии со следующими значениями

*center* -по центру

*left* -по левому краю

*right* -по правому краю

*justify* – по ширине

11. Тег **<PRE>** и обязательный парный ему **</PRE>** -обрамляет предварительно отформатированный текст. Преформатированный текст отображается моноширинным шрифтом.

12. Тег **<S>** и обязательный парный ему **</S>** - указывает, что текст должен быть зачеркнут.

13. Тег **<SMALL>** и обязательный парный ему **</SMALL>** - выводит более мелкий текст.Атрибуты: *title* -показывает текст в виде всплывающей подсказки.

14. Тег **<sub>** и обязательный парный ему **</sub>** - приспускает текст.

15. Тег **<sup>** и обязательный парный ему **</sup>** - приподнимает текст.

16. Тег **<tt>** и обязательный парный ему **</tt>** - создает текст, имитирующий стиль печатной машинки.

### Создание программно Html страницы

```
public static void CreateHtmlDoc(string TextInBody)
{
    StringBuilder sb = new StringBuilder();
    sb.Append("<html >");
    sb.Append("<head >");
    string meta = @"<meta charset=""UTF-8"">";
    sb.Append(meta);
    sb.Append("<title >");
    sb.Append("</title >");
}
```

```

sb.Append("</head >");
sb.Append("<body >");
sb.Append(TextInBody);
sb.Append("</body >");
sb.Append("</html >");
using(StreamWriter sw = new StreamWriter("MyHtml.html"))
{
    sw.Write(sb.ToString());
    sw.Close();
    Console.WriteLine("Файл создан успешно!");
    System.Diagnostics.Process.Start("MyHtml.html");
}
}

```

### Шаблон для решения 2го задания

```

string text = File.ReadAllText(text.txt);
string pattern = //регулярка для 2го предложения задания по
открытому и закрытому тегу
Regex reg = new Regex(pattern);
bool flag = true;
while (reg.IsMatch(text))
{
    text = flag? reg.Replace(text, "открытый тег",
1):reg.Replace(text, @"закрытый тег", 1);
    flag = !flag;
}

string[] mas = text.Split('\n');
pattern = // регулярка для 1го предложения по применению в
группу помещается символ по которому есть варианты
reg = new Regex(pattern);
for (int i = 0; i < mas.Length; i++ )
{
    if (reg.IsMatch(mas[i]))
    {
        string size = reg.Match(mas[i]).Groups[1].Value;
        mas[i] = reg.Replace(mas[i], @"тег с нужной
настройкой");
        mas[i] = mas[i] + "закрыть тег";
    }
}
text = string.Join("<Br>", mas,0,mas.Length);
GreateHtmlDoc(text);

```

## **II. Выполнение задания**

### **Задание 1.**

Используя регулярные выражения, напишите следующие приложения:

1. Текст находится в текстовом файле. Вывести все слова, начинающиеся с согласных букв русского алфавита.
2. Текст находится в текстовом файле. Вывести все слова, состоящие только из цифр.
3. Текст находится в текстовом файле, Вывести на экран все встречающиеся в данном тексте даты в формате DD.MM.YYYY
4. Текст находится в текстовом файле, Вывести на экран все встречающиеся в данном тексте номера сетевых адаптеров в формате HH-NN-NN-NN-NN-NN (где Н-шестнадцатеричная цифра).
5. Текст находится в текстовом файле, Подсчитать количество в данном файле пустых строк.
6. Текст находится в текстовом файле, Вывести на экран все встречающиеся в данном тексте доменные имена (имена в формате [www.xxx.xxx.ru](http://www.xxx.xxx.ru) , где x-любой символ латинского алфавита, количество символов может быть любым).
7. Текст находится в текстовом файле. Вывести на экран количество предложений в данном тексте.
8. Текст находится в текстовом файле. Переписать содержимое, предварительно заменяя первую букву каждого слова на заглавную.
9. Текст находится в текстовом файле . Вывести все различные слова.
10. Текст находится в текстовом файле . Найти число вхождение в данный текст предлога «не».
11. Текст находится в текстовом файле, имеющий структуру «Фамилия И.О. - \$оклад». Вывести на экран фамилия всех лиц, чей оклад превышает \$9000.
12. Текст находится в текстовом файле, имеющий структуру «Фамилия И.О. – рост см – вес кг». Вывести на экран фамилии всех лиц, чей рост превышает 190 см.
13. Текст находится в текстовом файле. Вывести на экран все строки, содержащие более 30 символов.
14. Текст находится в текстовом файле. Определить, есть ли в файле строки, начинающиеся на букву «Т».

### **Задание 2**

Используя возможности регулярных выражений по замене текста необходимо создать html страницу на основе текстового файла. Используя особое форматирование(указанно в варианте).

Для создания документа используется метод указанный [теории](#).

1. В начале строки текста может встретиться (Pc), где P означает тег P, а c – первая буква слова(center,left,right,justify) для выравнивания. Означает применить ко всей строке тега P. В тексте встречается (i) по 2 раза. Означает применение тега I.

2. В начале строки текста может встретиться (Hx), где x – номер заголовка. Означает применить ко всей строке заголовков Hx. В тексте встречается (sm) по 2 раза. Означает применение тега Small.

3. В начале строки текста может встретиться (FcX), где F означает тег font, c – color, а X – первая буква цвета(Yellow,Green,Red,Blue). Означает применить ко всей строке тег Font. В тексте встречается (u) по 2 раза. Означает применение тега U.

4. В начале строки текста может встретиться (Fsx), где F означает тег font, s – size, а x – размер шрифта. Означает применить ко всей строке тег Font. В тексте встречается (P) по 2 раза. Означает применение тега Pre.

5. В начале строки текста может встретиться (Ffx), где F означает тег font, f – face, а x – первая буква имени шрифта(Arial,Times New Roman, Calibri, Mistral) . Означает применить ко всей строке тег Font. В тексте встречается (s) по 2 раза. Означает применение тега S.

6. В начале строки текста может встретиться (Hsx), где H означает тег HR, s – size, а x – размер . Означает применить ко всей строке тег HR. В тексте встречается (b) по 2 раза. Означает применение тега B.

7. В начале строки текста может встретиться (Hwx), где H означает тег HR, w – width, а x – ширина в пикселах . Означает применить ко всей строке тег HR. В тексте встречается (t) по 2 раза. Означает применение тега tt.

8. В начале строки текста может встретиться (Hcx), где H означает тег HR, c – color, а x – первая буква цвета(White,Green,Red,Black). Означает применить ко всей строке тег HR. В тексте встречается (su) по 2 раза. Означает применение тега sup.

9. В начале строки текста может встретиться (FcX), где F означает тег font, c – color, а X – первая буква цвета(Aqua,Purple, Orange, Gold). Означает применить ко всей строке тег Font. В тексте встречается (su) по 2 раза. Означает применение тега sub.

10. В начале строки текста может встретиться (H1a), где H1 означает тег H1, а a – первая буква слова(center,left,right,justify) для выравнивания. Означает применить ко всей строке тег H. В тексте встречается (b) по 2 раза. Означает применение тега BLOCKQUOTE.

11. Коляго. В начале строки текста может встретиться (Ffx), где F означает тег font, f – face, а x – первая буква имени шрифта(Arial,Times New Roman, Calibri, Mistral) Означает применить ко всей строке тег Font. В тексте встречается (su) по 2 раза. Означает применение тега sub.

### ***III. Контрольные вопросы***

1. Способы создания класса Rgex.
2. Что такое квантификаторы (рассказать про все).
3. Методы для проверки совпадений.
4. Спецсимволы (рассказать про все).

### ***Литература***

1. Полный справочник по С#. Г. Шилдт. Издательский дом «Вильямс», 2004.
2. С# в подлиннике. Наиболее полное руководство. Х. Дейтел.
3. С# в задачах и примерах. Культин. Н.Б.
4. С# учебный курс. Г.Шилдт. СПб.: Питер, 2002.
5. С# программирование на языке высокого уровня Павловская Т.А. СПб.: БХВ-Петербург.