

Женщины продолжают подниматься по карьерной лестнице, занимая должности глав государств, лидеров корпораций и авторитетных руководителей СМИ, но тот факт, что их меньше, позволяет оценивать их на порядок ниже, только потому, что они женщины.

Вывод:

Как можно заметить, жизнь женщин далеко не так безоблачна, как можно судить по картинам салонных художников. Быть может, роскошные шелковые платья скрывают следы синяков, а нежные матери, трогательно обнимающие своих детей, через несколько лет будут рыдать в зале суда. Тем не менее, они не сдавались, но продолжали бороться за свои права – те права, которыми мы пользуемся сегодня.

Библиографический список

1. Английская литература с профессором Беллуччи: величайшие авторы. (URL: <https://englishlitwithprofbellucci.blogspot.com/2017/04/greatest-authors.html>)
2. Взаимоотношения мужчин и женщин в Викторианскую эпоху. (URL: <https://proza.ru/2013/12/11/68>)
3. Ломаем стереотипы: великие женщины- ученые Викторианской эпохи. (URL: <https://www.colors.life/post/527149/>)
4. Роль женщин в Викторианскую эпоху . (URL: <https://anna-warwick.livejournal.com/162589.html>)
5. Философия: женский род. Гарриет Тейлор - Милль. (URL: <https://www.colta.ru/articles/she/25674-filosofiya-zhenschiny-garriet-teylor-mill-video>)
6. 8 женщин- ученых изменивших мир (URL: <https://style.rbc.ru/people/5cfa0a3c9a79479ffbd87848>)
7. Успешные мировые писательницы современности. (URL: https://aminoapps.com/c/knizhnyi-klub/page/blog/samye-uspeshnye-mirovyepisatel'nitsy/YjX_gDQUbuv0Z4b4ZaXqrl0LZXI84wP8Le0)

УДК 004.04.004.045

Гусева М.А.

РАБОТА С ИЗМЕНЯЕМЫМИ СТРОКАМИ НА C# WORKING WITH VARIABLE STRINGS IN C #

*Филиал «Протвино» государственного университета «Дубна»
Секция «Информационные технологии»*

Автор: Гусева Мария Андреевна, студентка 1 курса направления «Информатика и вычислительная техника» филиала «Протвино» государственного университета «Дубна».

Научный руководитель: Кульман Татьяна Николаевна, кандидат технических наук, доцент кафедры информационных технологий филиала «Протвино» государственного университета «Дубна».

Author: Guseva Maria Andreevna, first year student of the direction "Informatics and computer engineering" of the branch "Protvino" state University "Dubna".

Scientific adviser: Kulman Tatiana Nikolaevna, candidate of technical sciences, associate professor of the department information technology of the branch "Protvino" state University "Dubna".

Аннотация

Рассматривается принцип работы класса StringBuilder в C#. Изучается различие классов String и StringBuilder. Исследуются методы класса StringBuilder.

Abstract

Explains how the StringBuilder class works in C#. Explore the difference between the String and StringBuilder classes.

Ключевые слова: String, StringBuilder, класс, объект, строка, метод, операция.

Keywords: String, StringBuilder, class, object, string, method, operation.

Цель работы: изучить различия между классами String и StringBuilder, выявить особенности класса String, рассмотреть методы класса StringBuilder, исследовать практическое применение класса StringBuilder.

Задачи, необходимые для достижения цели:

1. Изучение классов String и StringBuilder.
2. Исследование различий классов.
3. Изучение методов и свойств класса StringBuilder.

Изучение классов String и StringBuilder

Рассматривается язык программирования C#. String – класс с различными методами, созданный для работы со строками. Он включен в стандартную библиотеку языка C#. Его относят к числу важнейших классов многих сред программирования. Максимальный размер объекта String в памяти 2 ГБ, или примерно 1 миллиард символов. String является неизменным типом данных, то есть каждая операция, которая изменяет строку, на самом деле создает новую, что является крайне неэффективным для больших текстов, поскольку занимает большой объем памяти.

Строковый класс StringBuilder позволяет компенсировать этот недостаток, он предоставляет динамическую строку, которая меняется в зависимости от действий пользователя, и занимает меньше памяти. Класс StringBuilder находится в пространстве имен System.Text.

После всех изменений можно преобразовать объект StringBuilder в String. Это преобразование выполняется путем вызова метода StringBuilder.ToString().

У класса StringBuilder методов значительно меньше, чем у класса String. Поскольку класс StringBuilder создавался с целью изменения значения строки, в нём определены такие операции как вставка, удаление и замена подстрок, но нет методов, подобных поиску вхождения, которые можно выполнять над обычными строками.

Технология работы обычно такова:

- конструируется строка класса StringBuilder,
- выполняются операции, требующие изменения значения,
- полученная строка преобразуется в строку класса String – далее выполняются операции, не требующие изменения значения строки.

Исследование различий классов

Класс String рекомендуется применять в следующих ситуациях:

1. При небольшом объеме изменений в строках.
2. При фиксированном количестве операций объединения.
3. Когда приходится производить масштабные операции поиска, например, используя метод IndexOf().

Класс StringBuilder же используется в случаях:

1. Когда предполагается, что приложению придется сделать множество операций изменения.
2. Если ожидается, что приложение внесет неизвестное количество изменений[2].

Основная функциональность класса String раскрывается через его методы, среди которых можно выделить следующие:

- *Concat()*: объединение строк.
- *CopyTo()*: копирует часть строки, начиная с определенного индекса.
- *Format()*: форматирует строку.
- *IndexOf()*/*LastIndexOf()*: находит индекс первого(последнего) вхождения символа или подстроки.
- *Join()*: соединяет элементы строк.
- *Split()*: разделяет одну строку на массив строк.
- *ToLower()*/*ToUpper()*: переводит все символы строки в нижний(верхний) регистр[3].

Многие методы класса String изменяют строку, однако в больших объемах текста это

непродуктивно.

Изучение методов и свойств класса `StringBuilder`

`StringBuilder` имеет два основных свойства:

1. *Length* – реальная длина строки,
2. *Capacity* – выделенная память.

В приведенной ниже программе, строке `StringBuilder` присвоено начальное значение и указан объём выделяемой памяти 32. При добавлении нового текста длина строки становится равной 36 и автоматически объём удваивается до 64. Следующее добавление текста не выводит за рамки выделенной памяти и объём остаётся равным 64 (Рис.1).

```
using System;
using System.Text;
namespace ConsoleApp3
{
    class Program
    {
        static void Main(string[] args)
        {
            StringBuilder sb = new StringBuilder("Тема: ", 32);
            Console.WriteLine(sb);
            Console.WriteLine("Количество символов в строке: {0}", sb.Length);
            Console.WriteLine("Выделенная память на строку: {0}", sb.Capacity);

            sb.Append("Работа с изменяемыми строками ");
            Console.WriteLine(sb);
            Console.WriteLine("Количество символов в строке: {0}", sb.Length);
            Console.WriteLine("Выделенная память на строку: {0}", sb.Capacity);

            sb.Append("в C#");
            Console.WriteLine(sb);
            Console.WriteLine("Количество символов в строке: {0}", sb.Length);
            Console.WriteLine("Выделенная память на строку: {0}", sb.Capacity);
        }
    }
}
```

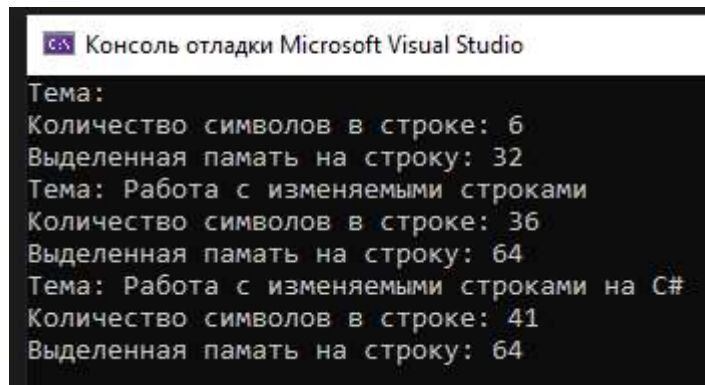


Рис. 1. Вид консольного приложения

Класс `StringBuilder` использует такие методы как:

- *Append()*: добавляет сведения в конец текущего объекта `StringBuilder`.
- *Insert()*: вставляет подстроку в объект `StringBuilder`, начиная с определенного индекса.
- *Remove()*: удаляет определенное количество символов, начиная с определенного индекса.

- *Replace()*: заменяет все вхождения определенного символа или подстроки на другой символ или подстроку.
 - *AppendFormat()*: добавляет подстроку в конец объекта `StringBuilder[1]`.
- Рассмотрим эти методы на примерах небольших программ.

Append() (Рис.2).

```
using System;
using System.Text;
```

```
namespace ConsoleApp3
{
    class Program
    {
        static void Main(string[] args)
        {
            StringBuilder sb = new StringBuilder("Приветствую всех присутствующих");
            Console.WriteLine(sb);
            sb.Append(" гостей");
            Console.WriteLine(sb);
        }
    }
}
```

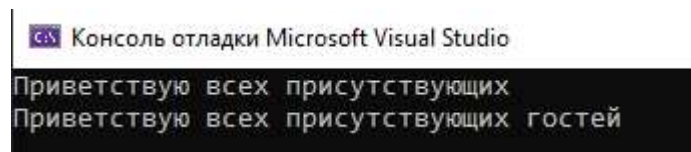


Рис. 2. Вид консольного приложения. Метод Append

Insert() (Рис.3)

```
using System;
using System.Text;
```

```
namespace ConsoleApp3
{
    class Program
    {
        static void Main(string[] args)
        {
            StringBuilder sb = new StringBuilder("Приветствую всех присутствующих");
            Console.WriteLine(sb);
            sb.Insert(11, " и премного уважаю");
            Console.WriteLine(sb);
        }
    }
}
```

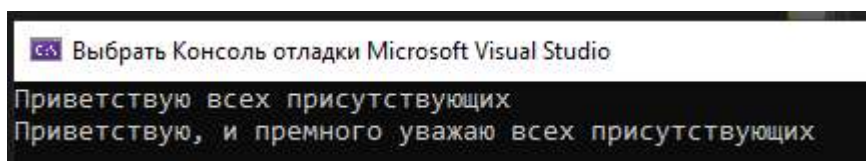


Рис. 3. Вид консольного приложения. Метод Insert

Remove() (Рис.4)

```
using System;
using System.Text;

namespace ConsoleApp3
{
    class Program
    {
        static void Main(string[] args)
        {
            StringBuilder sb = new StringBuilder("Приветствую всех присутствующих");
            Console.WriteLine(sb);
            sb.Remove(11, 19);
            Console.WriteLine(sb);
        }
    }
}
```

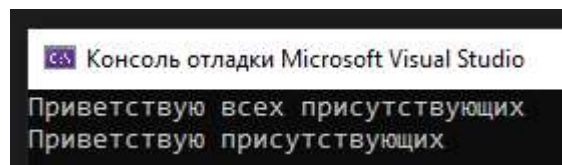


Рис. 4. Вид консольного приложения. Метод Remove

Replace() (Рис.5)

```
using System;
using System.Text;

namespace ConsoleApp3
{
    class Program
    {
        static void Main(string[] args)
        {
            StringBuilder sb = new StringBuilder("Приветствую всех присутствующих");
            Console.WriteLine(sb);
            sb.Replace("присутствующих", "пришедших");
            Console.WriteLine(sb);
        }
    }
}
```

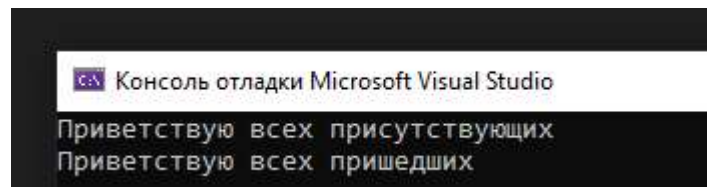


Рис. 5. Вид консольного приложения. Метод Replace

AppendFormat() (Рис.6)

```
using System;
using System.Text;
namespace ConsoleApp3
{
```

```

class Program
{
    static void Main(string[] args)
    {
        StringBuilder sb = new StringBuilder("Приветствую всех присутствующих");
        Console.WriteLine(sb);StringBuilder a = new StringBuilder(" на нашем");
        sb.AppendFormat("{0} мероприятия", a);
        Console.WriteLine(sb);
    }
}
}

```

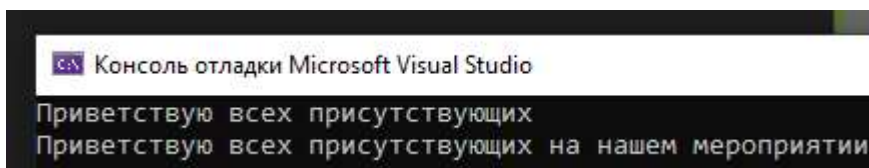


Рис. 6. Вид консольного приложения. Метод AppendFormat

Выводы. Применение класса StringBuilder значительно сокращает затрачиваемую память на обработку больших текстов, поэтому, с точки зрения изменений текста, является эффективным по сравнению с классом String. Выполнены все поставленные задачи, а именно:

1. Изучены классы String и StringBuilder.
2. Исследованы различия классов.
3. Изучены методы и свойства класса StringBuilder.

Библиографический список

1. Уроки по C# и платформе .NET Framework -- Класс StringBuilder -- https://professorweb.ru/my/csharp/charp_theory/level4/4_8.php
2. StringBuilder Класс -- <https://docs.microsoft.com/ru-ru/dotnet/api/system.text.stringbuilder?view=net-5.0#Converting>
3. METANIT.COM -- Работа со строками -- Строки и класс System.String -- <https://metanit.com/sharp/tutorial/7.1.php>

УДК 7.013

Дарнейко Е.К.

ВЛИЯНИЕ ДРЕВНЕРИМСКОГО ЯЗЫЧЕСТВА И СРЕДНЕВЕКОВОГО ХРИСТИАНСТВА НА ХУДОЖЕСТВЕННУЮ КУЛЬТУРУ INFLUENCE OF ANCIENT LANGUAGE AND MEDIEVAL CHRISTIANITY FOR ART CULTURE

*Филиал «Протвино» государственного университета «Дубна»
Секция «Социальные и гуманитарные науки»*

Автор: Дарнейко Евгений Константинович, студент 2 курса направления «Информатика и вычислительная техника» филиала «Протвино» государственного университета «Дубна».

Научный руководитель: Ардашева Маргарита Евгеньевна, кандидат педагогических наук, доцент кафедры общеобразовательных дисциплин филиала «Протвино» государственного университета «Дубна».

Author: Darneiko Evgeniy Konstantinovich, 2d year student of the direction "Informatics and Computer Engineering" of the branch "Protvino" of the State University "Dubna".