**Практическое занятие № 9. Символьные данные и строки. Обработка символьных и строковых данных**

**1 Цель занятия**

Получить практические навыки объявления и обработки символьной информации.

**2 Перечень оборудования и программного обеспечения**

Персональный компьютер

Microsoft Office (Word)

Microsoft Visual Studio

**3 Краткие теоретические сведения**

**3.1 Строки**

В языке C# определен массив char[], и его можно использовать для представления строк постоянной длины. Однако массив char[] нельзя инициализировать строкой символов. В C# не определено преобразование из типа char[] в тип string. У объектов типа string есть динамический метод ToCharArray(), задающий подобное преобразование в char[].

Объекты типа string объявляются с явной или отложенной инициализацией. Чаще всего, при объявлении строковой переменной инициализация задается строковой константой. Также строку можно сконструировать:

* из символа, повторяющегося указанное число раз;
* из массива символов char[];
* из части массива символов.

**3.2 Встроенный сервис по обслуживанию строк**

Строкам соответствует свой класс System.String, который позволяет с ними работать. Методы класса String позволяют выполнять вставку, удаление, замену, поиск вхождения подстроки в строку и многое другое.

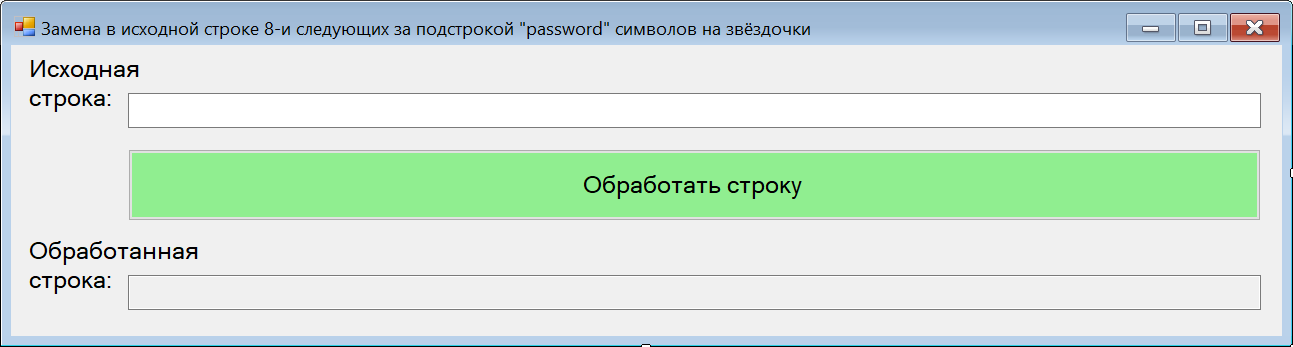
| **Название метода** | **Описание** |
| --- | --- |
| [Clone](http://msdn.microsoft.com/ru-ru/library/system.string.clone.aspx)() | Возвращает ссылку на данный экземпляр типа string |
| [Compare()](http://msdn.microsoft.com/ru-ru/library/84787k22.aspx) | Сравнивает два указанных объекта типа string и возвращает целое число, которое показывает их относительное положение в порядке сортировки |
| [Concat()](http://msdn.microsoft.com/ru-ru/library/khca9w90.aspx) | Создает строковое представление указанного объекта |
| [Contains](http://msdn.microsoft.com/ru-ru/library/dy85x1sa.aspx)() | Возвращает значение, указывающее, содержит ли данная строка заданный объект типа string |
| [Copy](http://msdn.microsoft.com/ru-ru/library/system.string.copy.aspx)() | Создает новый экземпляр типа string, имеющий то же значение, что и заданный экземпляр типа string |
| [EndsWith()](http://msdn.microsoft.com/ru-ru/library/2333wewz.aspx) | Определяет, совпадает ли конец данного экземпляра строки с указанной строкой |
| [Equals(String)](http://msdn.microsoft.com/ru-ru/library/858x0yyx.aspx) | Определяет, равны ли значения двух указанных объектов типа string |
| [Format()](http://msdn.microsoft.com/ru-ru/library/fht0f5be.aspx) | Заменяет элементы формата в указанной строке строковым представлением указанного объекта |
| [IndexOf()](http://msdn.microsoft.com/ru-ru/library/k8b1470s.aspx) | Возвращает индекс с отсчетом от нуля первого вхождения значения указанной строки (или символа Юникода) в данном экземпляре строки |
| [Insert](http://msdn.microsoft.com/ru-ru/library/system.string.insert.aspx)() | Возвращает новую строку, в которой указанная строка вставляется на указанной позиции индекса в данном экземпляре строки |
| [Intern](http://msdn.microsoft.com/ru-ru/library/system.string.intern.aspx)() | Извлекает системную ссылку на указанный объект типа string |
| [IsInterned](http://msdn.microsoft.com/ru-ru/library/system.string.isinterned.aspx) | Извлекает ссылку на указанный объект типа string |
| [IsNullOrEmpty](http://msdn.microsoft.com/ru-ru/library/system.string.isnullorempty.aspx)() | Указывает, действительно ли указанная строка является строкой null или [Empty](http://msdn.microsoft.com/ru-ru/library/system.string.empty.aspx) |
| [Join()](http://msdn.microsoft.com/ru-ru/library/dd783876.aspx) | Сцепляет элементы массива строк или коллекции [IEnumerable<T>](http://msdn.microsoft.com/ru-ru/library/9eekhta0.aspx) типа string, помещая между ними заданный разделитель |
| [LastIndexOf()](http://msdn.microsoft.com/ru-ru/library/1wdsy8fy.aspx) | Возвращает позицию индекса с отсчетом от нуля последнего вхождения указанной строки (или символа Юникода) в данном экземпляре |
| [Normalize()](http://msdn.microsoft.com/ru-ru/library/8eaxk1x2.aspx) | Возвращает новую строку, текстовое значение которой совпадает с данной строкой, а двоичное представление находится в нормализованной форме C Юникода |
| [PadLeft()](http://msdn.microsoft.com/ru-ru/library/92h5dc07.aspx) | Возвращает новую строку, в которой символы данного экземпляра выровнены по правому краю путем добавления слева пробелов или указанного символа Юникода до указанной общей длины |
| [PadRight()](http://msdn.microsoft.com/ru-ru/library/36f2hz3a.aspx) | Возвращает новую строку, в которой символы данной строки выровнены по левому краю путем добавления справа пробелов или указанного символа Юникода до указанной общей длины |
| [Remove()](http://msdn.microsoft.com/ru-ru/library/9ad138yc.aspx) | Возвращает новую строку, в которой были удалены все символы, начиная с указанной позиции и до конца (или указанное число символов в указанной позиции) в текущем экземпляре строки |
| [Replace()](http://msdn.microsoft.com/ru-ru/library/fk49wtc1.aspx) | Возвращает новую строку, в которой все вхождения заданной строки (или символа Юникода) в текущем экземпляре строки заменены другой заданной строкой (или символом Юникода) |
| [Split()](http://msdn.microsoft.com/ru-ru/library/c1bs0eda.aspx) | Возвращает строковый массив, содержащий подстроки данного экземпляра строки, разделенные элементами заданного массива символов Юникода |
| [StartsWith()](http://msdn.microsoft.com/ru-ru/library/baketfxw.aspx) | Определяет, совпадает ли начало данного экземпляра строки с указанной строкой |
| [Substring()](http://msdn.microsoft.com/ru-ru/library/aka44szs.aspx) | Извлекает подстроку из данного экземпляра строки. Подстрока начинается с указанной позиции символа и имеет указанную длину |
| [ToCharArray()](http://msdn.microsoft.com/ru-ru/library/ezftk57x.aspx) | Копирует символы данного экземпляра строки (или подстроки) в массив символов Юникода |
| [ToLower()](http://msdn.microsoft.com/ru-ru/library/e78f86at.aspx) | Возвращает копию данной строки, переведенную в нижний регистр |
| [ToString()](http://msdn.microsoft.com/ru-ru/library/8tc6ws5s.aspx) | Возвращает текущий объект в виде экземпляра типа string; реальное преобразование не осуществляется |
| [ToUpper()](http://msdn.microsoft.com/ru-ru/library/ewdd6aed.aspx) | Возвращает копию этой строки, переведенную в верхний регистр |
| [Trim()](http://msdn.microsoft.com/ru-ru/library/d4tt83f9.aspx) | Удаляет все начальные и конечные вхождения пробела или набора символов, заданного в виде массива, из текущего объекта типа string |
| [TrimEnd](http://msdn.microsoft.com/ru-ru/library/system.string.trimend.aspx)() | Удаляет все конечные вхождения набора символов, заданного в виде массива, из текущего объекта типа string |
| [TrimStart](http://msdn.microsoft.com/ru-ru/library/system.string.trimstart.aspx)() | Удаляет все начальные вхождения набора символов, заданного в виде массива, из текущего объекта типа string |

Класс String не разрешает изменять существующие объекты. Для этой цели имеется другой класс – класс StringBuilder, который позволяет исправить данный недостаток. Этот класс принадлежит к изменяемым классам и его можно найти в пространстве имен System.Text.

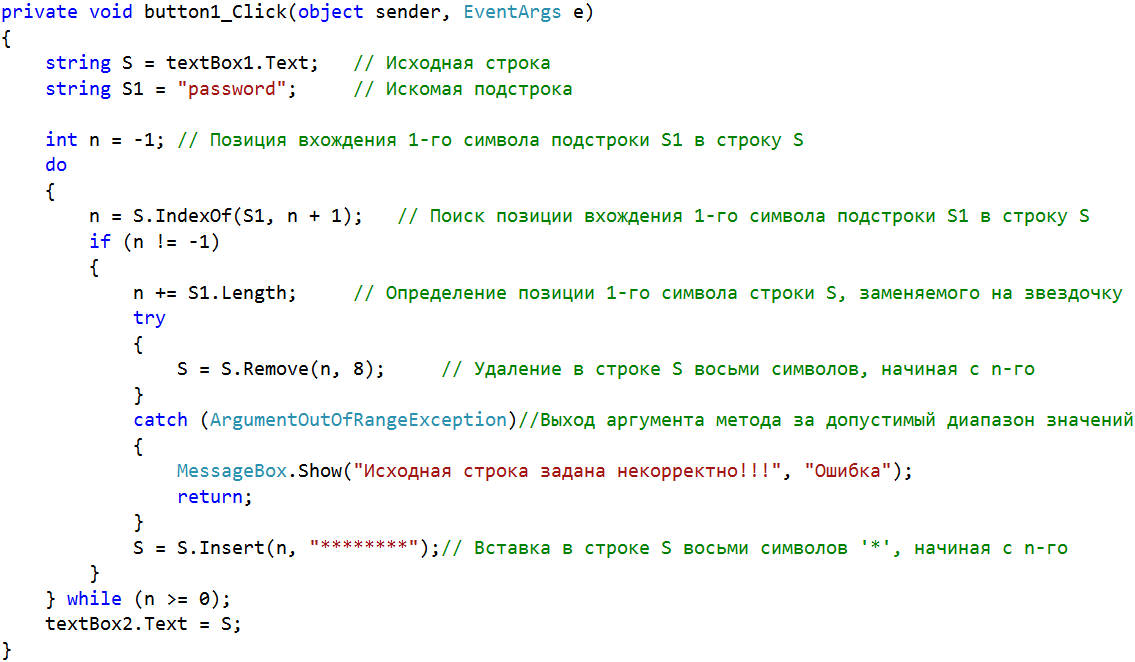
**Задание.** Дана непустая строка *S*, содержащая произвольное число раз подстроку «password». Заменить в заданной строке восемь следующих за подстрокой «password» символов на звездочки, применяя методы класса String.

**Решение.**

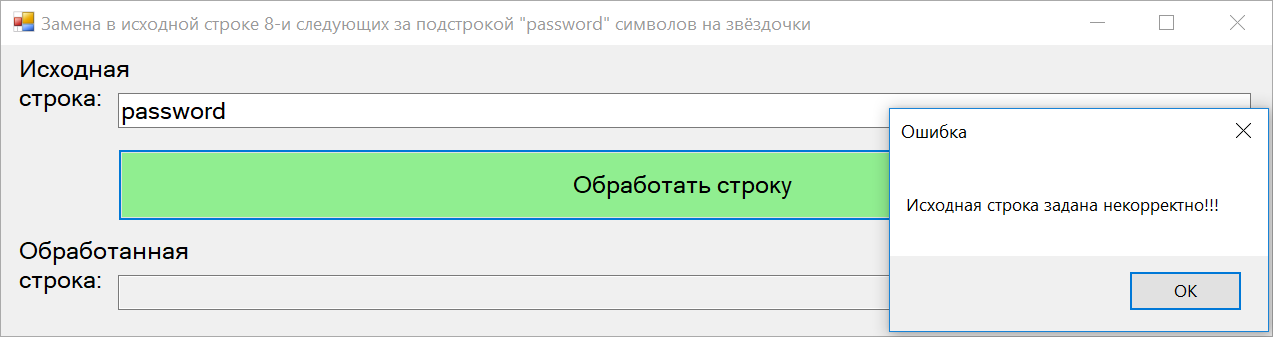
а) Интерфейс приложения Windows Forms:

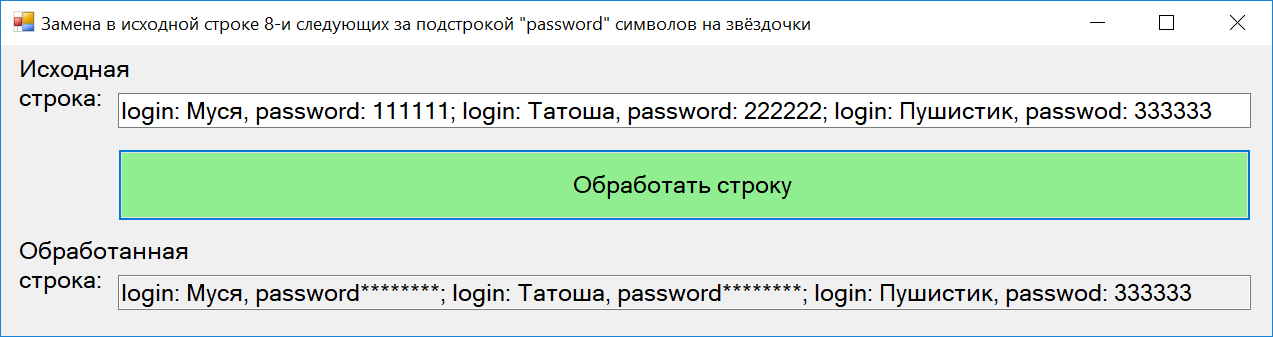


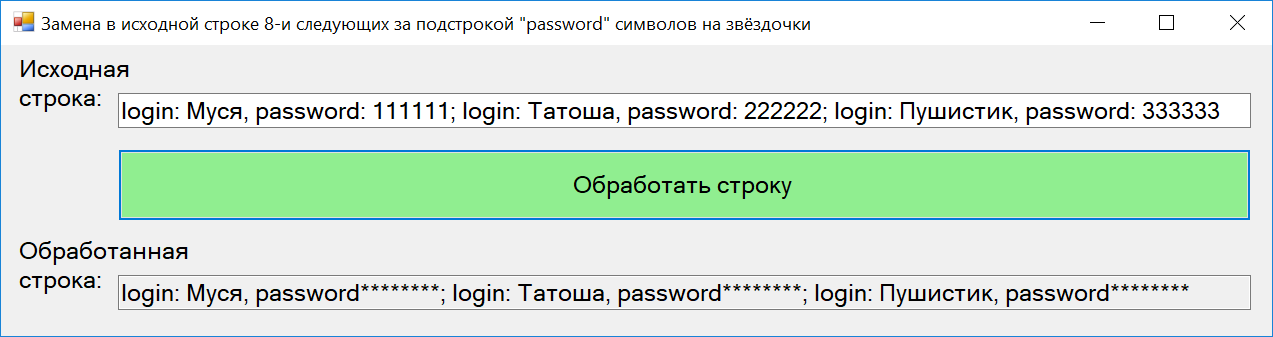
Код, выполняемый при щелчке на кнопку «Обработать строку»:



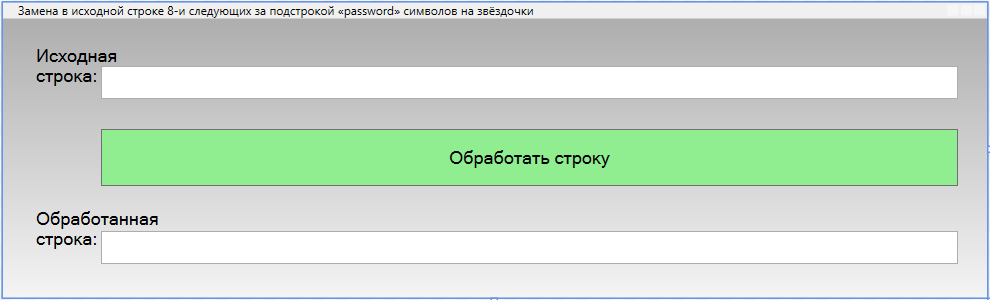
Скрины отлаженной программы:



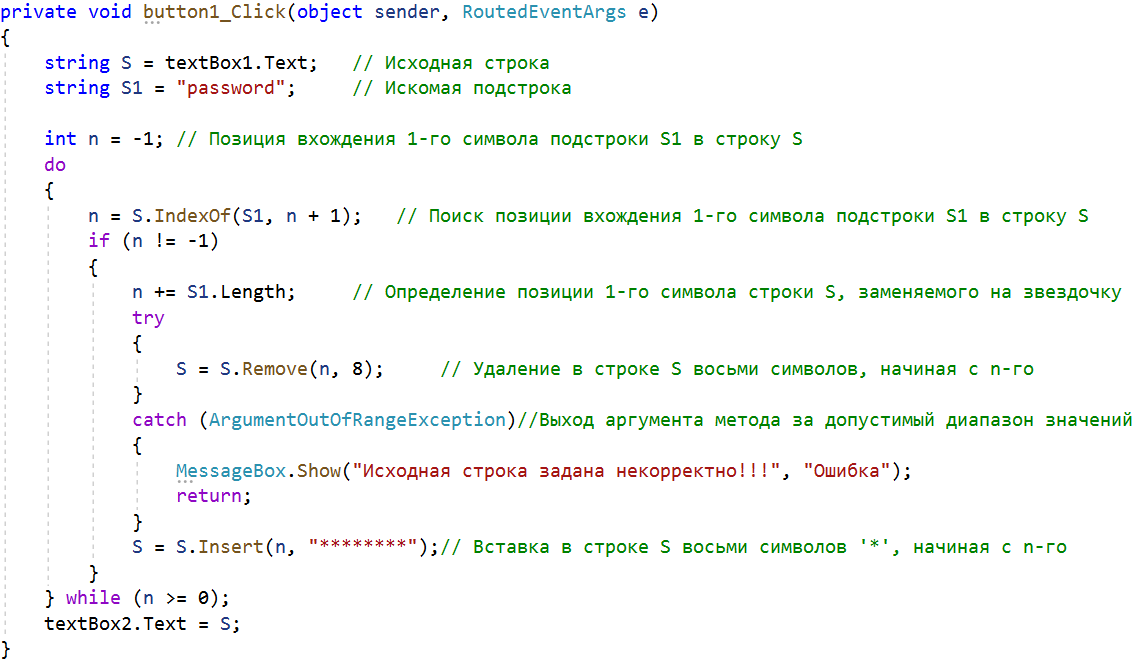




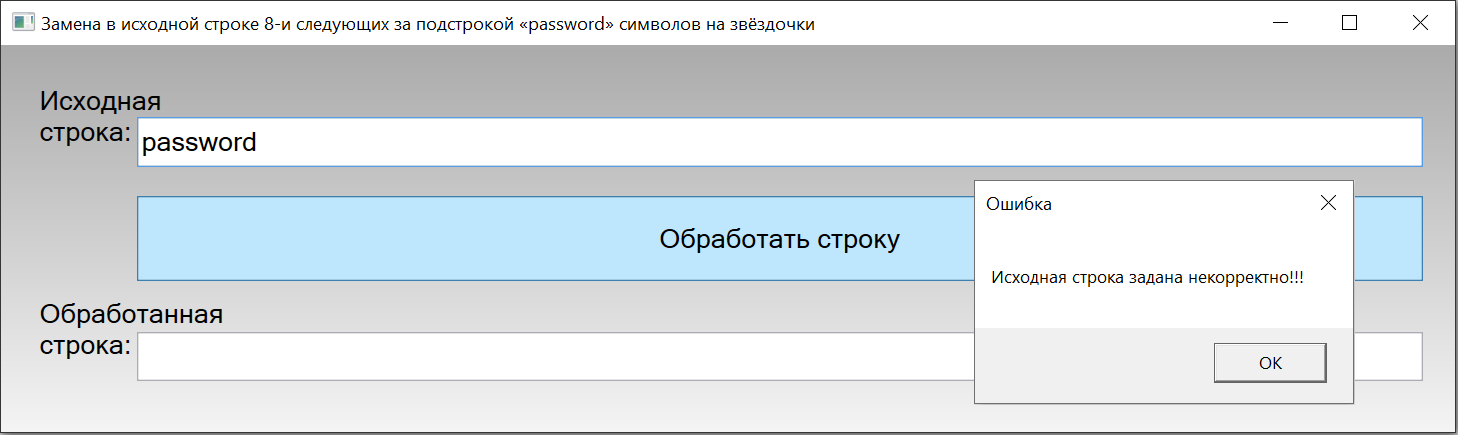
б) Интерфейс приложения WPF:

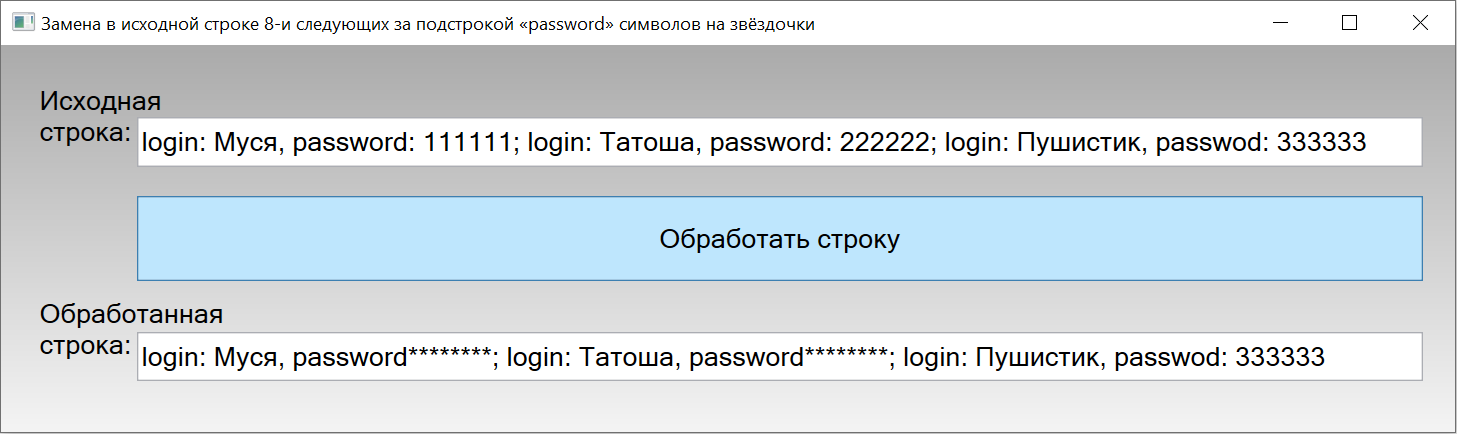


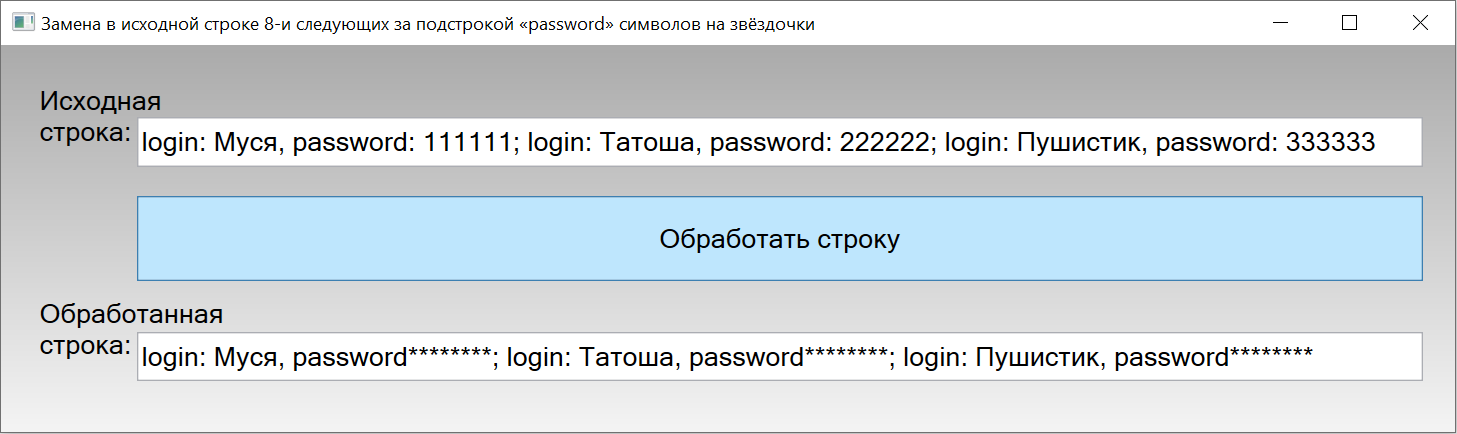
Код, выполняемый при щелчке на кнопку «Обработать строку»:



Скрины отлаженной программы:

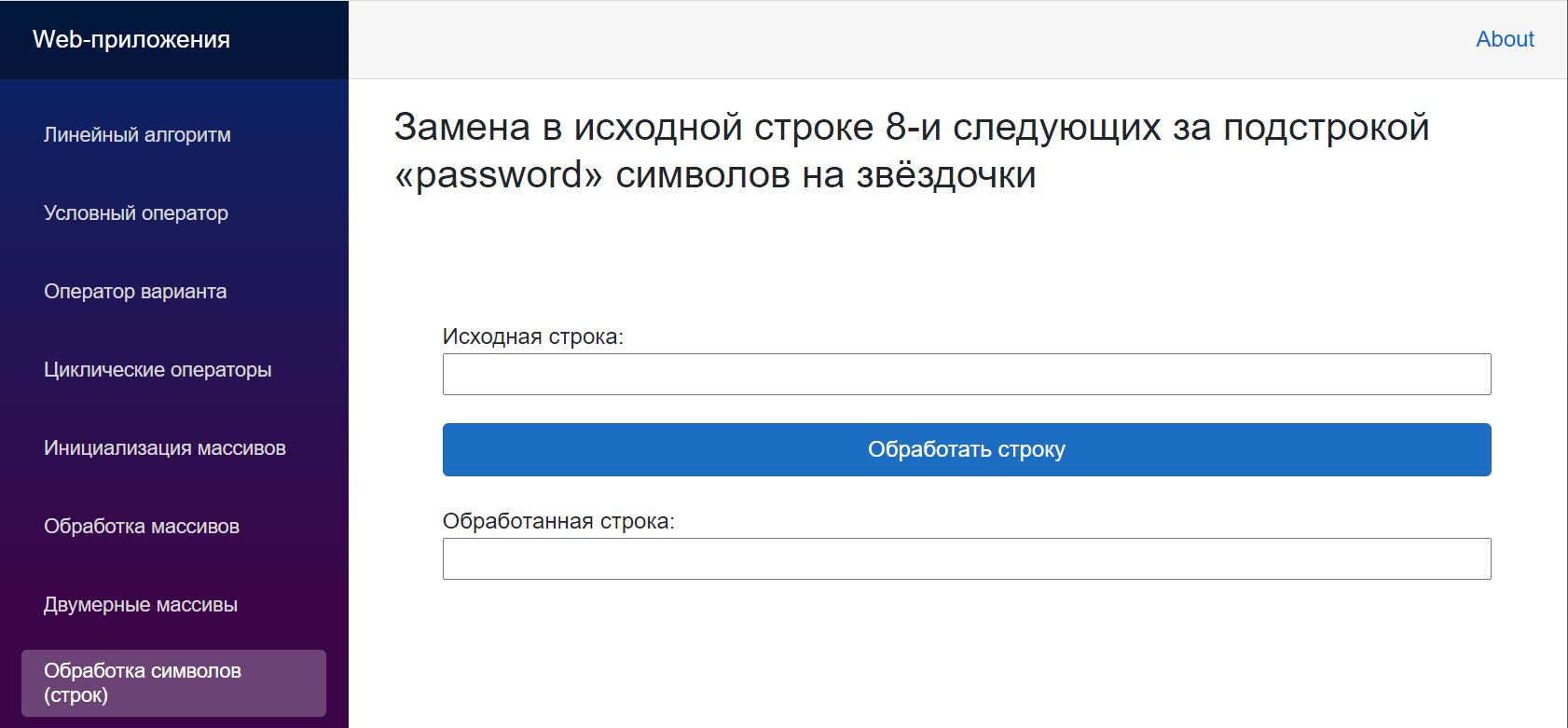




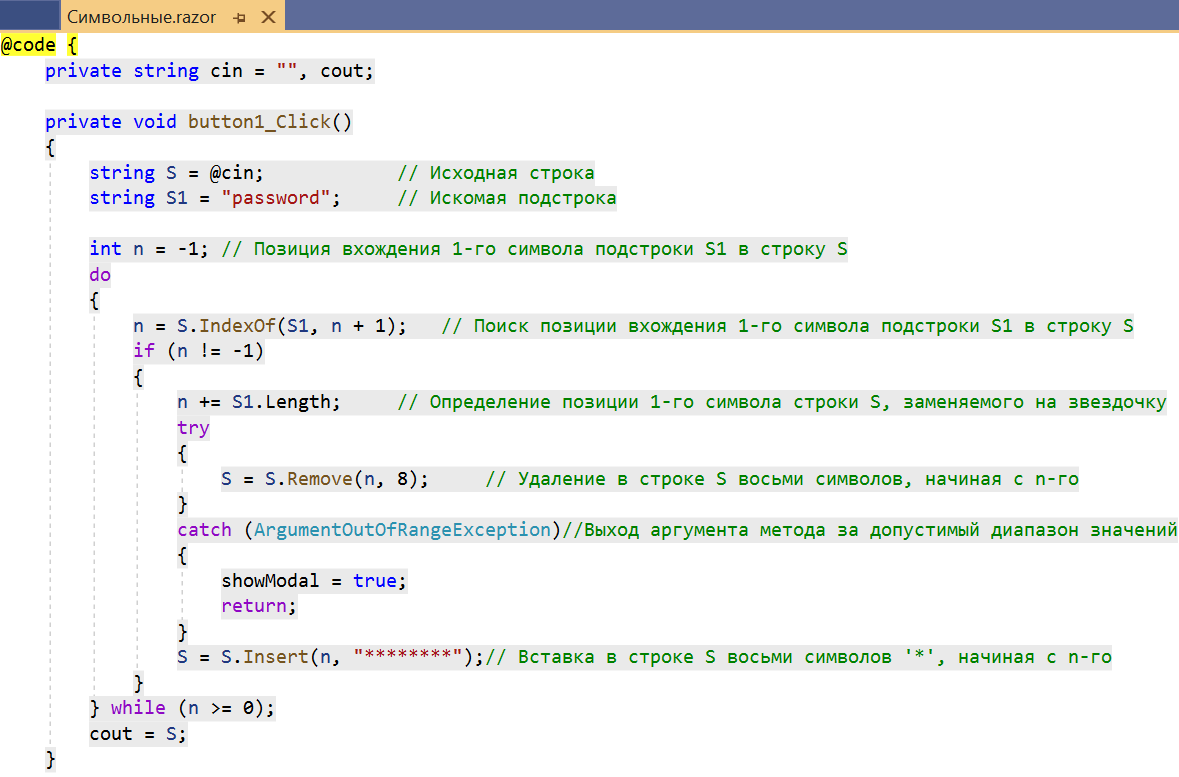


в) Интерфейс приложения WebAssembly Blazor:





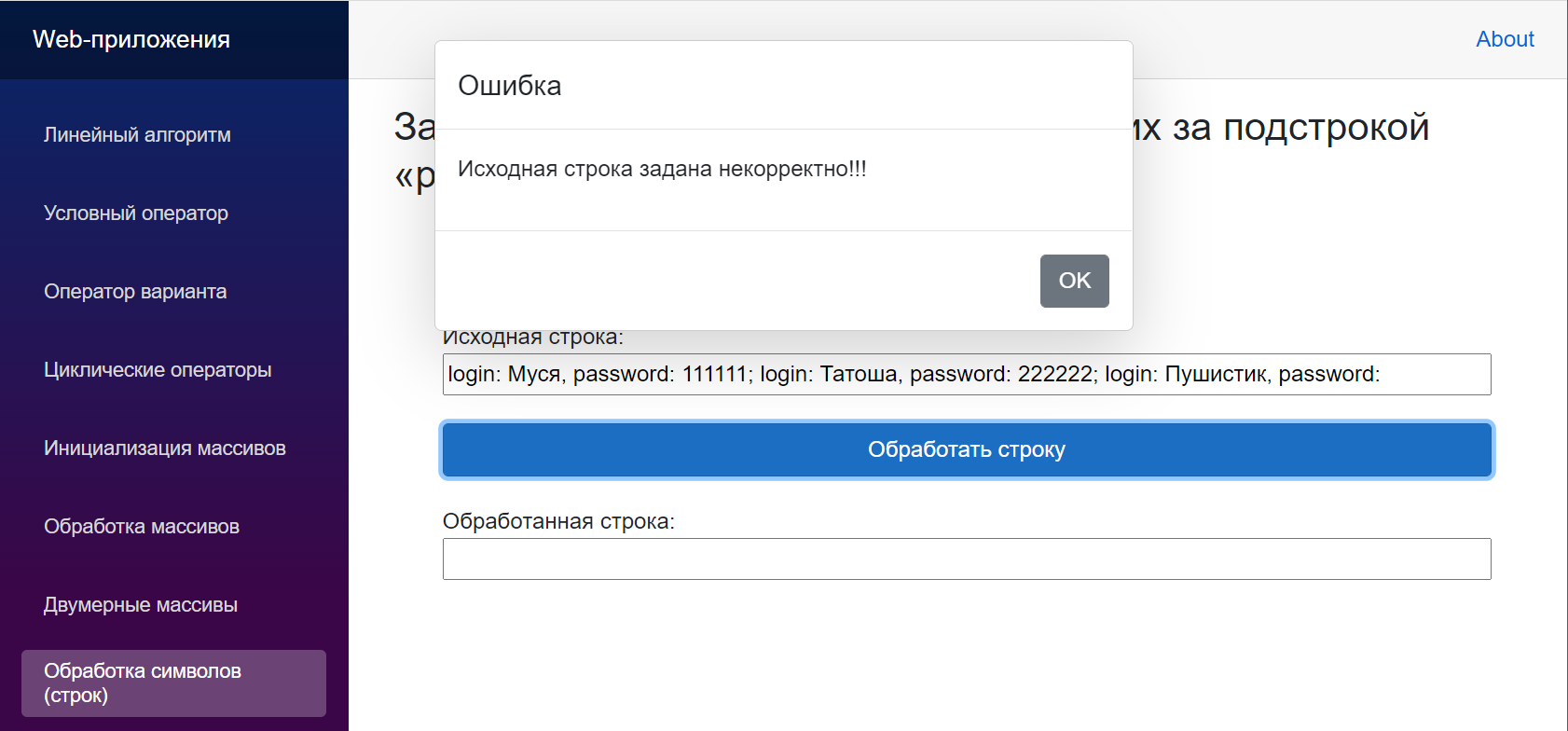
Коды приложения:

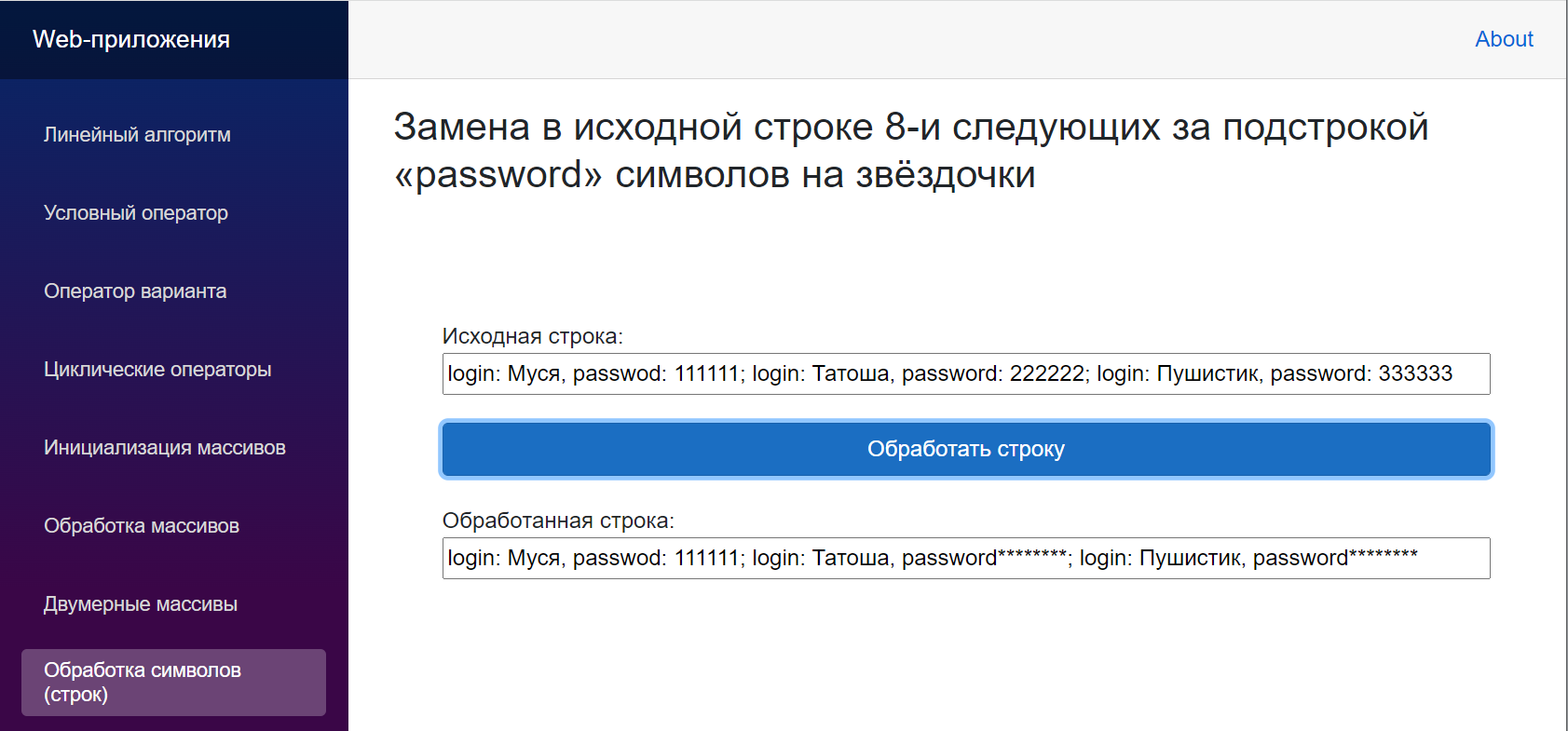


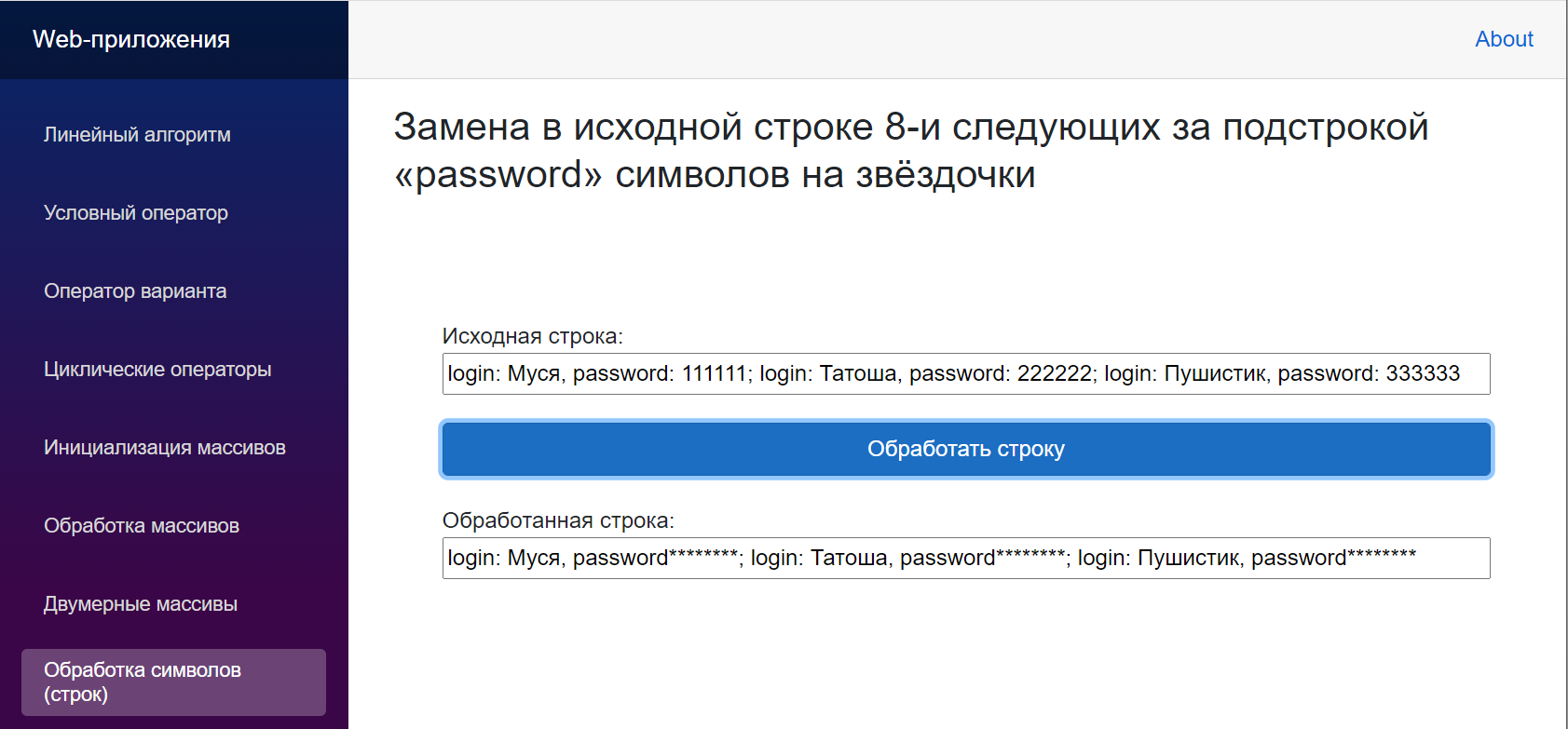




Скрины отлаженной программы:







**4 Порядок выполнения работы**

4.1 Изучить теоретические сведения и задание к работе.

4.2 В соответствии с вариантом задания создать форму приложения Windows Forms (WPF, WebAssembly Blazor).

4.3 В соответствии с вариантом задания разработать и отладить код приложения Windows Forms (WPF, WebAssembly Blazor), применяя методы класса String и используя исключения, выбрасываемые, если форматы или значения входных данных не соответствуют спецификациям параметров обрабатывающих их методов.

**5 Содержание отчета**

5.1 Название работы.

5.2 Цель работы.

5.3 Программные коды и скрины работы разработанного приложения Windows Forms в соответствии с вариантом задания.

**6 Варианты заданий**

6.1 Дано четное число *N* (> 0) и символы *C*1 и *C*2. Вывести строку длины *N*, которая состоит из чередующихся символов *C*1 и *C*2, начиная с *C*1.

6.2 Дана строка. Вывести строку, содержащую те же символы, но расположенные в обратном порядке.

6.3 Дана непустая строка *S*. Вывести строку, содержащую символы строки *S*, между которыми вставлено по одному пробелу.

6.4 Дана непустая строка *S* и целое число *N* (> 0). Вывести строку, содержащую символы строки *S*, между которыми вставлено по *N* символов «\*» (звездочка).

6.5 Дана строка. Подсчитать количество содержащихся в ней цифр.

6.6 Дана строка. Подсчитать количество содержащихся в ней прописных латинских букв.

6.7 Дана строка. Преобразовать в ней все строчные латинские буквы в прописные.

6.8 Дана строка. Преобразовать в ней все строчные буквы в прописные, а прописные — в строчные.

6.9 Дана строка, изображающая целое положительное число. Вывести сумму цифр этого числа.

6.10 Дана строка, изображающая арифметическое выражение вида «<цифра>±<цифра>±…±<цифра>», где на месте знака операции «±» находится символ «+» или «–» (например, «4+7–2–8»). Вывести значение данного выражения (целое число).

6.11 Дана строка, изображающая двоичную запись целого положительного числа. Вывести строку, изображающую десятичную запись этого же числа.

6.12 Дана строка, изображающая десятичную запись целого положительного числа. Вывести строку, изображающую двоичную запись этого же числа.

6.13 Дан символ *C* и строки *S*, *S*0. После каждого вхождения символа *C* в строку *S* вставить строку *S*0.

6.14 Даны целые положительные числа *N*1 и *N*2 и строки *S*1 и *S*2. Получить из этих строк новую строку, содержащую первые *N*1 символов строки *S*1 и последние *N*2 символов строки *S*2 (в указанном порядке).

6.15 Даны строки *S* и *S*0. Найти количество вхождений строки *S*0 в строку *S*.

6.16 Дан символ *C* и строка *S*. Удвоить каждое вхождение символа *C* в строку *S*.

6.17 Даны строки *S*, *S*1 и *S*2. Заменить в строке *S* первое вхождение строки *S*1 на строку *S*2.

**7 Используемая литература**

7.1 Г. Гагарина, Е. В. Кокорева, Б. Д. Виснадул Технология разработки программного обеспечения. Форум, Инфра-М, 2009

7.2 Эндрю Троелсен Язык программирования С# 2010 и платформа .NET 4.0, М., ВИЛЬЯМС, 2011

7.3 А. Марченко C#. Введение в программирование, М, Вильямс, 2009

7.4 http://msdn.microsoft.com/ru-ru/library/67ef8sbd.aspx.