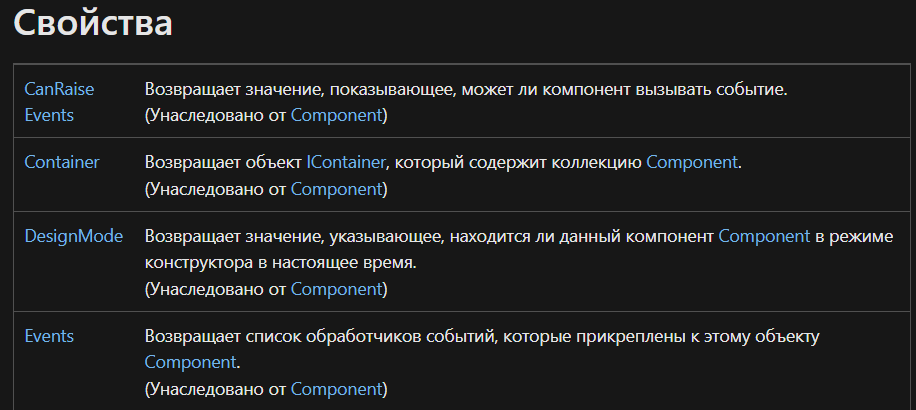
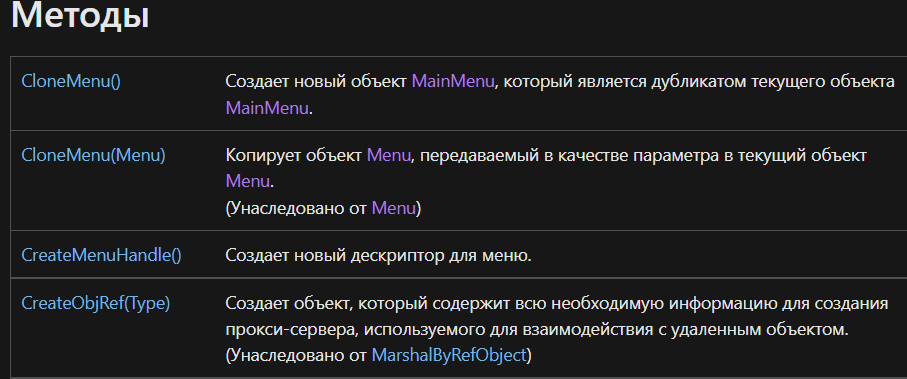
1. **Назовите классы, которые используются для создания меню. Перечислите свойства и методы.**
2. MenuStrip (унаследован от toolStrip) выступает своего рода контейнером для отдельных пунктов меню, которые представлены объектом **ToolStripMenuItem**

свойства компонента MenuStrip:

* Dock: прикрепляет меню к одной из сторон формы
* LayoutStyle: задает ориентацию панели меню на форме.
* TextDirection: задает направление текста в пунктах меню

1. MainMenu

ззззззз



**2. Что может содержать строка состояния? Какие есть методы управления строкой состояния?**

StatusStrip представляет строку состояния, во многом аналогичную панели инструментов ToolStrip. Строка состояния предназначена для отображения текущей информации о состоянии работы приложения или результат действий пользователя.

StatusStrip может содержать различные элементы. В режиме дизайнера мы можем добавить следующие типы элементов:

* StatusLabel: метка для вывода текстовой информации. Представляет объект ToolStripLabel
* ProgressBar: индикатор прогресса. Представляет объект ToolStripProgressBar
* DropDownButton: кнопка с выпадающим списком по клику. Представляет объект ToolStripDropDownButton
* SplitButton: еще одна кнопка, во многом аналогичная DropDownButton. Представляет объект ToolStripSplitButton

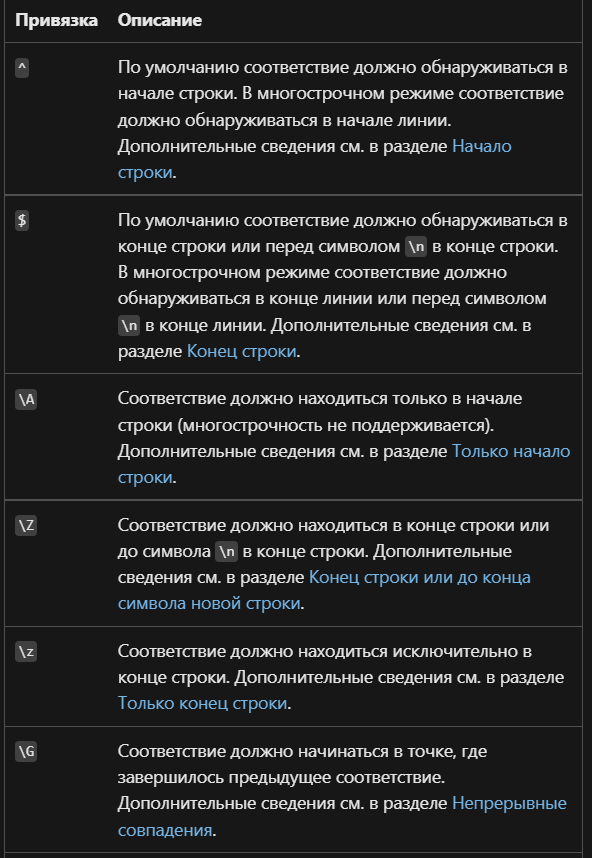
**3. Что такое регулярные выражения? Где и как их можно использовать?**

Регулярные выражения – это язык для описания текста и внесения в него изменений. Регулярное выражение применяется к строке. Результатом применения является фрагмент строки, либо новая строка, либо группы подстрок, либо логический результат – в зависимости от того, какая операция выполняется.

Регулярные выражения очень мощный и в то же время простой механизм обработки текстовой информации. На данный момент наиболее полно они реализованы в язык Perl, хотя возникли гораздо раньше. Для работы с регулярными выражениями в С# существует класс System.Text.RegularExpressions.Regex. Многие методы этого класса также существуют в двух версиях – статической и экземпляра. У класса два конструктора с одним параметром строкового типа, определяющего правило обработки и с двумя – второй в этом случае задает параметры регулярного выражения (аналог опций в Perl).

**4. Что такое привязки (якоря) в RegEx? Приведите примеры.**

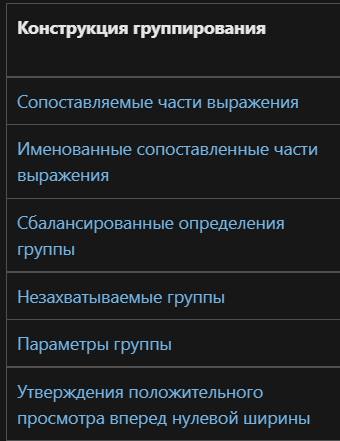
Привязки (или атомарные утверждения нулевой ширины) указывают положение в строке, где должно быть найдено соответствие. При использовании привязки в выражении поиска обработчик регулярных выражений не проходит по строке и не потребляет символы; он ищет соответствия только в заданном местоположении.

****

**5. Зачем используют конструкции группирования? Приведите примеры.**

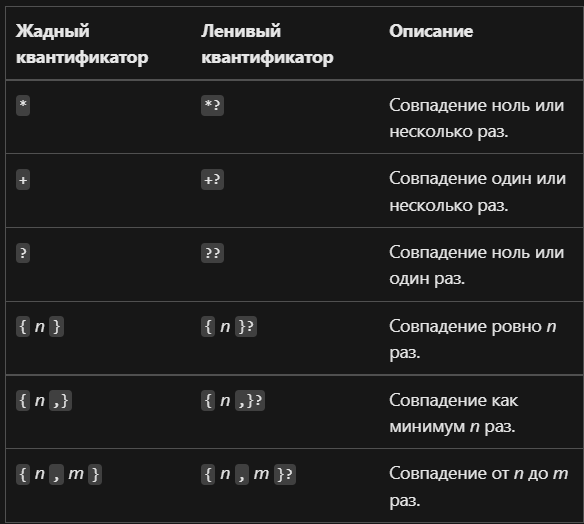
Конструкции группирования отображают части выражений регулярных выражений и захватывают части строки входной строки. Конструкции группирования можно использовать в следующих целях.

* Сопоставление с частью выражения, которая повторяется во входной строке.
* Применение квантификатору к части выражения с несколькими языковыми элементами регулярных выражений.
* Добавление в строку части выражения, которая возвращается методами [Regex.Replace](https://docs.microsoft.com/ru-ru/dotnet/api/system.text.regularexpressions.regex.replace) и [Match.Result](https://docs.microsoft.com/ru-ru/dotnet/api/system.text.regularexpressions.match.result) .
* Извлечение отдельных частей выражения из свойства [Match.Groups](https://docs.microsoft.com/ru-ru/dotnet/api/system.text.regularexpressions.match.groups) и обработка их по отдельности от всего текста.



**6. Что такое квантор или множители? Приведите примеры.**

Квантификаторы определяют количество экземпляров символа, группы или класса символов, которое должно присутствовать во входных данных, чтобы было зафиксировано совпадение.



Количества n и m являются целочисленными константами. Обычно квантификаторы "жадные" — они заставляют обработчик регулярных выражений выделить максимально возможное число вхождений определенного шаблона. Добавление символа ? в квантификатор делает его отложенным (или ленивым). Это приводит к тому, что обработчик регулярных выражений пытается сопоставить так мало вхождений, как это возможно.

**7. Напишите регулярное выражение для проверки номера MTC (Velcom).**

"^(\+375|80)(29|25|44|33)(\d{3})(\d{2})(\d{2})$"

**8. Напишите регулярное выражение для проверки паспорта.**

"^(\(МР|AB|BM|HB|KH|MC|KB|PP|SP|DP)(\d{7})$"

**9. Напишите регулярное выражение для проверки даты.**

"^[0-9]?[0-9]-[0-9]{2}-[0-9]{4}$"

Формат dd-dd-dddd

[0-9]{4}-(0[1-9]|1[012])-(0[1-9]|1[0-9]|2[0-9]|3[01])

Формат dddd-dd-dd

**10. Напишите регулярное выражение для проверки УДК**

"[0-9]+\.?/?-?"

**11. Напишите регулярное выражение для проверки Фамилии.**

^([a-zA-Z]{2,}\s[a-zA-Z]{1,}'?-?[a-zA-Z]{2,}\s?([a-zA-Z]{1,})?)

**12. Напишите регулярное выражение для проверки пароля, логина.**

Первый символ всегда буква, от 2 до 20 символов

^[a-zA-Z][a-zA-Z0-9-\_\.]{1,20}$

^(?=.\*?[A-Z])(?=.\*?[a-z])(?=.\*?[0-9])(?=.\*?[#?!@$%^&\*-]).{8,}$

• По крайней мере одна английская буква в верхнем регистре

• По крайней мере одна английская буква в нижнем регистре

• По крайней мере одна цифра

• По крайней мере один специальный символ

• Минимум 8 в длину

**13. Разберите**

"?<=<img .\*?src\s\*=\s\*"")[^""]+(?="".\*?>)" – путь к картинке

"[a-zA-Zа-яА-Я,-;:]{5,50}" – буквы и знаки пунктуации из диапазонов, суммарная длина лексемы от 5 до 50 символов

/^[a-z0-9\_-]{3,16}$/ - начало и конец строки, буквы и цифры, аналогично \w (оно тоже включает нижнее подчеркивание), суммарная длина лексемы от 3 до 16 символов, ПАРОЛЬ

/^#?([a-f0-9]{6}|[a-f0-9]{3})$/ - кодирование цвета, шестнадцатеричное

/^(https?:\/\/)?([\da-z\.-]+)\.([a-z\.]{2,6})([\/\w \.-]\*)\*\/?$/ - URL https адрес

/^(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$/ - IP-адрес

/^<([a-z]+)([^<]+)\*(?:>(.\*)<\/\1>|\s+\/>)$/ - HTML-теги

**ДЛЯ ЛАБЫ**

Валидация (проверка) данных, вводимых пользователями.

