Работа с текстом

№ урока: 4 **Курс:** C# Professional

Средства обучения: Компьютер с установленной Visual Studio

Обзор, цель и назначение урока

Урок направлен на изучение работы со строками, регулярными выражениями, а также кодирования/декодирования текста. Регулярные выражения предоставляют мощный, гибкий и эффективный метод обработки текста. Обширные возможности сопоставления шаблонов, предоставляемые регулярными выражениями, позволяют быстро анализировать большие объемы текста, отыскивая в них определенные символьные последовательности, проверять текст на соответствие определенным заранее требованиям. Системные классы `String` и `StringBuilder` предоставляет множество возможностей для эффективной работы со строками и большими объемами текста. Механизмы, позволяющие выполнять локализацию приложений, определены в пространстве имен `System.Text`.

Изучив материал данного занятия, учащийся сможет:

- Выполнять различные операции с текстом и/или текстовыми файлами, находить в них определенные шаблоны или наоборот, запрещать вводить и сохранять текст согласно установленного шаблона
- Определять кодировку текста, создавать и читать файлы в нестандартных кодировках
- Эффективно работать со строковыми данными, применяя все возможности языка C# для оптимизации производительности при необходимости работы с большими объемами текста

Содержание урока

1. Работа со строками
2. Форматирование вывода
3. Локализация и глобализация
4. Кодирование-декодирование строк. Использование класса `Encoding`
5. Использование класса `StringBuilder`
6. Регулярные выражения и пространство `System.Text.RegularExpressions`
7. Метасимволы и Квантификаторы
8. Метод `Regex.Replace` и использование внутри шаблонных переменных
9. Организация поиска по шаблону. Применение `Match`
10. Построение регулярных выражений
11. Чтение и запись файлов в разных кодировках

Резюме

- Строка (`String`) является упорядоченной коллекцией символов Юникода, используемой для представления текста. Объект `String` является упорядоченной коллекцией объектов `System.Char`, представляющей строку. Значением объекта `String` является содержимое упорядоченной коллекции, и это значение является неизменяемым (т. е. доступным только для чтения).
- Новый экземпляр объекта `String` можно создать следующими способами:
  - Путем присвоения строкового литерала переменной `String`.
  - Путем вызова конструктора класса `String`.
  - С помощью оператора сцепления строк (+) для создания одной строки из любой комбинации экземпляров `String` и строковых литералов.
  - Путем извлечения свойства или вызова метода, который возвращает строку.
  - Путем вызова метода форматирования для преобразования значения или объекта в строковое представление.
- `StringBuilder` – данный класс предоставляет подобный строке объект, значение которого является изменяемой последовательностью знаков. Значение считается

CyberBionic systematics

CyberBionic Systematics ® 2015
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine

t. +380 (44) 361-8473
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com

изменяемым потому, что после создания его можно изменить путем добавления, удаления, замены или вставки знаков.

- Емкостью `StringBuilder` считается максимальное количество знаков, которое экземпляр может хранить в любой момент времени. Емкость больше или равна длине строкового представления значения экземпляра.
- Регулярные выражения – это незаменимый инструмент для многих приложений, в которых ведется работа со строками или анализ объемных блоков текста. Сравнивая с текстом регулярные выражения, состоящие из чисел, букв в определенном регистре или шестнадцатеричных строк, можно принимать решения, влияющие на работу программы.
- Основа обработки текста с помощью регулярных выражений – это подсистема обработки регулярных выражений, представленная в платформе .NET Framework объектом `System.Text.RegularExpressions.Regex`. Минимальный набор сведений, который требуется предоставить подсистеме обработки регулярных выражений для обработки текста, сводится к: созданию шаблона регулярного выражения и применение его к анализируемому тексту.
- Основные метасимволы, для составления шаблона поиска:
  - \d – определяет символы цифр.
  - \D – определяет любой символ, который не является цифрой.
  - \w – определяет любой символ цифры, буквы или подчеркивания.
  - \W – определяет любой символ, который не является цифрой, буквой или подчеркиванием.
  - \s – определяет любой непечатный символ, включая пробел.
  - \S – определяет любой символ, кроме символов табуляции, новой строки и возврата каретки.
  - . – определяет любой символ кроме символа новой строки.
  - \. – определяет символ точки.
- Методы класса `Regex`, позволяют определить, встречается ли во входном тексте шаблон регулярного выражения. Для этого можно использовать метод `IsMatch`. Также можно извлечь из текста одно или все вхождения, соответствующие шаблону регулярного выражения, путем вызова метода `Match` или `Matches`. Первый метод возвращает объект `Match`, предоставляющий сведения о совпадении в тексте. Второй метод возвращает коллекцию `MatchCollection`, в которую входят объекты `Match` для всех совпадений, найденных в проанализированном тексте.
- Заменить текст, соответствующий шаблону регулярного выражения, можно путем вызова метода `Replace`.
- `IsMatch` – метод возвращающий `bool`. `True` - в случае, если шаблон соответствует строке или `false` – в противном случае. Метод `IsMatch` – сравнивает принимаемую в первом параметре строку с шаблоном.
- При проверке вводимых данных рекомендуется начинать регулярные выражения с символа «^» и заканчивать их символом «$». Это гарантирует проверку строки, точно соответствующей заданному шаблону, а не просто содержащей его.
- Кодирование – это процесс преобразования набора символов Юникода в последовательность байтов. Декодирование, наоборот, представляет собой процесс преобразования последовательности закодированных байтов в набор символов Юникода.
- Все данные в текстовых строках и файлах кодируются с использованием одного из стандартов видов кодирования. Почти всегда .NET Framework обрабатывает данные в разных кодировках автоматически. Кодированием-декодированием приходится управлять вручную, в случаях, когда выполняется взаимодействии с унаследованными или UNIX-системами, выполняется чтение-запись файлов на других языках, создаются HTML-страницы, генерируются сообщения электронной почты.
- Класс `System.Text.Encoding` поддерживает статические методы для кодирования-декодирования текста.
- Метод `System.Text.Encoding.GetEncoding` возвращает объект, представляющий текст в заданной кодировке. Метод `Encoding.GetBytes` преобразует строку Unicode в серию байтов с заданной кодировкой.
- Чтобы задать кодировку при записи/чтении файла используют перегруженный конструктор `Stream`, принимающий объект `Encoding`.

CyberBionic systematics

CyberBionic Systematics ® 2015
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine

t. +380 (44) 361-8473
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com

Закрепление материала

- Что такое регулярные выражения и для чего они применяются?
- Как создать шаблон регулярного выражения?
- В каких случаях необходимо использовать метод `IsMatch`?
- Что такое кодирование и декодирование. В каких случаях его необходимо применять?
- Что такое локализация? Какие методы для работы с локалями вам известны?
- Как можно определить культуру, установленную в системе в текущий момент времени?
- Какие возможности форматирования вывода поддерживает C#?
- Обоснуйте необходимость применения класса `StringBuilder` для работы с большими объемами текста.

Дополнительное задание

Напишите консольное приложение, позволяющие пользователю зарегистрироваться под «Логином», состоящем только из символов латинского алфавита, и пароля, состоящего из цифр и символов.

Самостоятельная деятельность учащегося

Задание 1
Выучите основные конструкции и понятия, рассмотренные на уроке.

Задание 2
Напишите программу, которая бы позволила вам по указанному адресу web-страницы выбирать все ссылки на другие страницы, номера телефонов, почтовые адреса и сохраняла полученный результат в файл.

Задание 3
Напишите шуточную программу «Дешифратор», которая бы в текстовом файле могла бы заменить все предлоги на слово «ГАВ!».

Задание 4
Создайте текстовый файл-чек по типу «Наименование товара – 0.00 (цена) грн.» с определенным количеством наименований товаров и датой совершения покупки. Выведите на экран информацию из чека в формате текущей локали пользователя и в формате локали en-US.

Задание 5
Зайдите на сайт MSDN.
Используя поисковые механизмы MSDN, найдите самостоятельно описание темы по каждому примеру, который был рассмотрен на уроке, так, как это представлено ниже, в разделе «Рекомендуемые ресурсы», описания данного урока. Сохраните ссылки и дайте им короткое описание.

Рекомендуемые ресурсы

MSDN: Системный класс `String`
http://msdn.microsoft.com/ru-ru/library/system.string.aspx

MSDN: Системный класс `StringBuilder`
http://msdn.microsoft.com/ru-ru/library/system.text.stringbuilder.aspx

MSDN: Пространство имен `System.Globalization`
http://msdn.microsoft.com/ru-ru/library/system.globalization.aspx

MSDN: Регулярные выражения (Visual Studio)
http://msdn.microsoft.com/ru-ru/library/2k3te2cs.aspx

CyberBionic systematics
CyberBionic Systematics ® 2015
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine
t. +380 (44) 361-8473
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com

MSDN: Регулярные выражения в .NET Framework
http://msdn.microsoft.com/ru-ru/library/hs600312.aspx

MSDN: Пространство имен `System.Text.RegularExpressions`
http://msdn.microsoft.com/ru-ru/library/system.text.regularexpressions.aspx

MSDN: Практическое руководство. Поиск строк с помощью регулярных выражений (Руководство по программированию в C#)
http://msdn.microsoft.com/ru-ru/library/ms228595.aspx

CyberBionic systematics

CyberBionic Systematics ® 2015
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine

t. +380 (44) 361-8473
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com