



Лекция № 8



**IT Education
Academy**

WWW.ITEA.UA

C# Base

Урок № 8

XML

План урока

- Понятие XML файла
- Загрузка и чтение из XML файла
- Применение XmlTextReader и XmlTextWriter
- Загрузка содержимого XML из строки, файла и по URL
- Запись комментариев в XML файл
- Запись пространства имен и префиксов в XML файл
- Выборка из XML с помощью XPath. (Запросы XPath)
- Создание файлов конфигурации приложения.
- Работа с реестром Windows.



XML файл



- Текстовый формат, предназначенный для хранения структурированных данных для обмена информацией между программами, а также для создания на его основе более специализированных языков разметки.
- На сегодняшний день **XML** является одним из распространенных стандартов документов, который позволяет в удобной форме сохранять сложные по структуре данные. Поэтому разработчики платформы .NET включили в фреймворк широкие возможности для работы с XML.

Структура XML документа

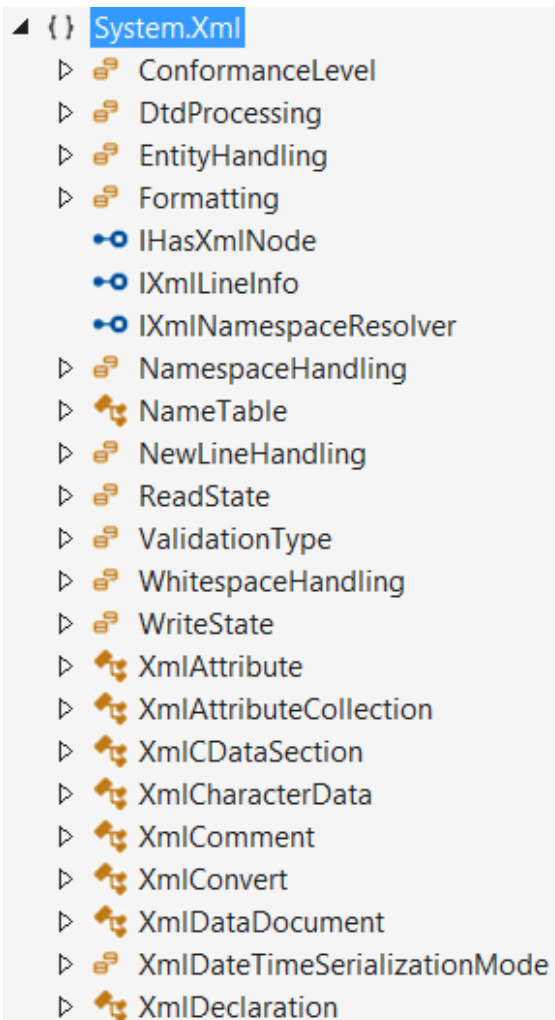
Документ XML:

- **объявление XML** – первая строка документа, в которой указывается версия, кодировка текста и другие зависимости
- **корневой элемент** – является основным телом документа, у каждого документа может быть только один корневой элемент
- **теги** – вложенные элементы документа, могут иметь атрибуты и содержимое
- **комментарии** – текст, размещенный в теге `<!-- -->`

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<ListOfBooks>
  <Books.xml>
    <Title>CLR via C# Рихтер</Title>
    <Price>999</Price>
  </Books.xml>
  <!--This is a comment-->
</ListOfBooks>
```

System.Xml



- [System.Xml](#) - пространство имен, в котором содержатся классы, необходимые для работы с XML файлами.
- В [System.Xml](#) определен ряд классов, которые позволяют манипулировать xml-документом.
- [XmlDocument](#): представляет весь xml-документ;
- [XmlNode](#): представляет узел xml. В качестве узла может использоваться весь документ, так и отдельный элемент;
- [XmlAttribute](#): представляет атрибут элемента

XmlDocument Класс

[XmlDocument](#) представляет XML-документ. Этот класс можно использовать для загрузки, проверки, изменения, добавления и размещения XML-кода в документе.

[Свойства](#)

[Attributes](#) - Возвращает класс [XmlAttributeCollection](#), содержащий атрибуты данного узла.

[ChildNodes](#) - Возвращает все дочерние узлы данного узла.

[NodeType](#) - Возвращает тип текущего узла.

[Value](#) - Возвращает или задает значение узла.

Основные классы для чтения и записи данных XML

- [XmlReader](#) – обеспечивает последовательный доступ только для чтения к потоку XML-данных. Умеет работать только в одном направлении (вперед).

- [XmlWriter](#) – представляет средство записи, обеспечивающее быстрый прямой способ (без кэширования) создания потоков файлов, содержащих XML-данные.

Класс XmlWriter позволяет: проверять допустимость XML-символов и допустимость XML-имен элементов и атрибутов, проверять правильность формата XML-документа, передавать значения, используя типы среды CLR вместо строк, записывать несколько документов в выходной поток, записывать допустимые имена, полные имена и маркеры имен.

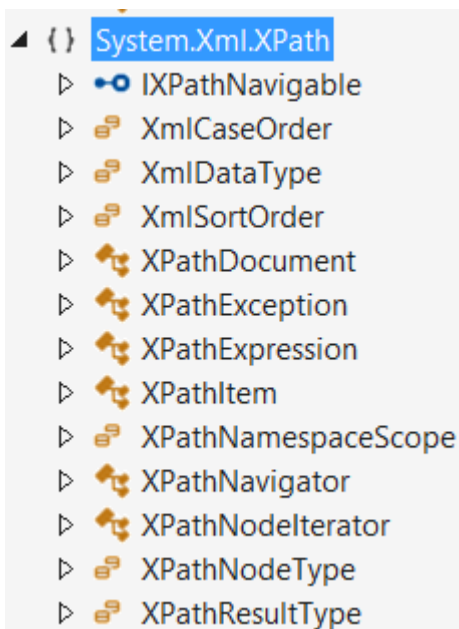
Основные классы для чтения и записи данных XML

[XmlTextReader](#) – расширяет класс XmlReader. Обеспечивает быстрый однонаправленный потоковый доступ к данным XML.

[XmlTextWriter](#) – расширяет класс XmlWriter. Обеспечивает быструю однонаправленную генерацию потоков данных XML.

[XmlNode](#) – предоставляет отдельный узел в XML-документе.

XPath



XPath представляет язык запросов в XML. Он позволяет выбирать элементы, соответствующие определенному селектору.

[System.Xml.XPath](#) - Пространство имен, позволяющее использовать возможности языка XPath (XML Path Language).

Содержит класс [XPathDocument](#), позволяющий создавать экземпляры [XPathNavigator](#).

[XPathNavigator](#) – предоставляет курсор-ориентированную модель для просмотра и редактирования XML данных.

Linq to Xml

Еще один подход к работе с Xml представляет технология [LINQ to XML](#). Вся функциональность LINQ to XML содержится в пространстве имен [System.Xml.Linq](#). Основные классы этого пространства имен:

[XAttribute](#): представляет атрибут xml-элемента

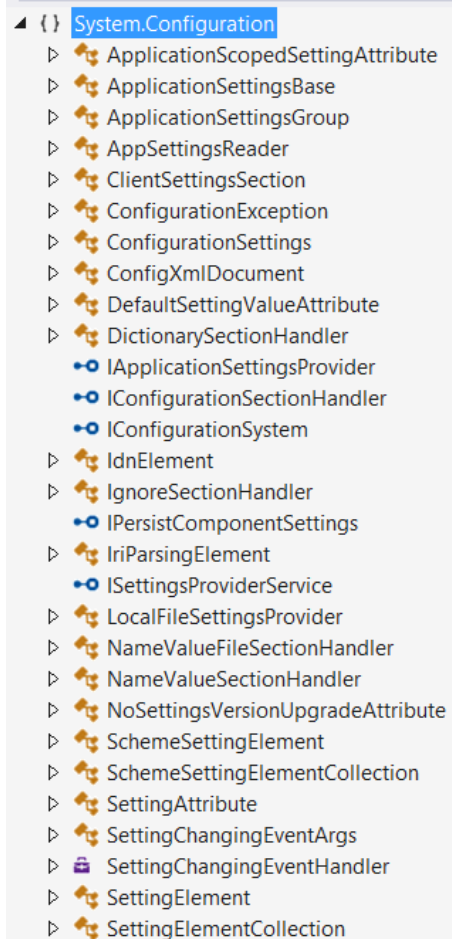
[XComment](#): представляет комментарий

[XDocument](#): представляет весь xml-документ

[XElement](#): представляет отдельный xml-элемент

Ключевым классом является [XElement](#), который позволяет получать вложенные элементы и управлять ими

Файлы конфигураци



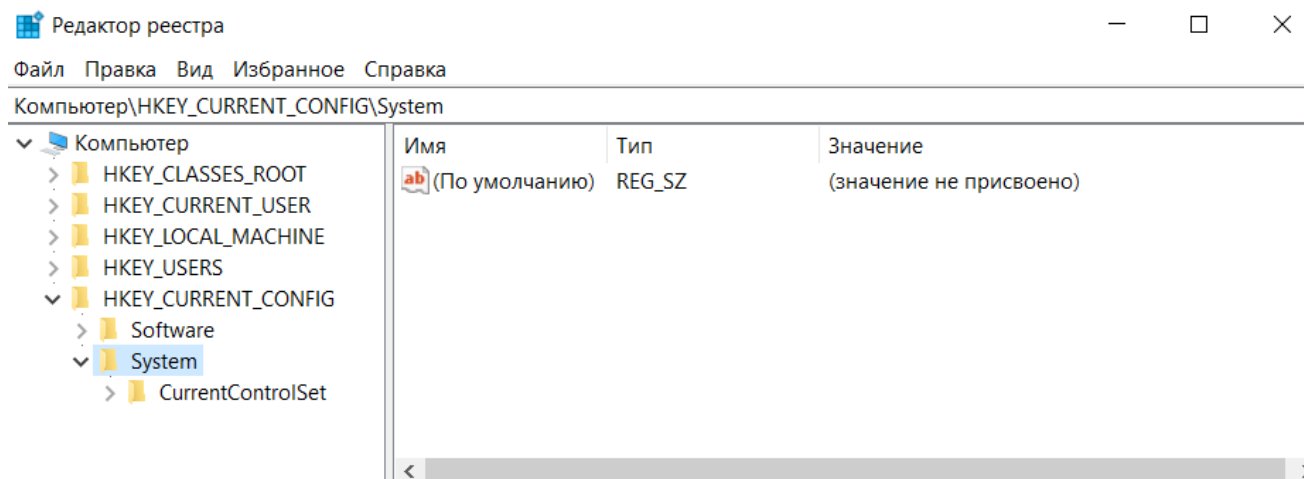
Пространство имен [System.Configuration](#) содержит типы, которые обеспечивают программную модель для работы с данными конфигурации.

Классы:

- [ConfigurationSettings](#) - поддерживает чтение разделов конфигурации и общих параметров конфигурации;
- [ConfigurationManager](#) - предоставляет доступ к файлам конфигурации для клиентских приложений. Этот класс не наследуется;

Реестр

- Реестр Windows или системный реестр ([Windows Registry](#)) — иерархически построенная база данных параметров и настроек в большинстве операционных систем Microsoft Windows.
- Реестр содержит информацию и настройки для аппаратного обеспечения, программного обеспечения, профилей пользователей, предустановки. Большинство изменений в Панели управления, ассоциации файлов, системные политики, список установленного ПО фиксируются в реестре.



Разделы реестра

- [HKEY_CLASSES_ROOT](#) – содержит информацию о ассоциациях расширения файлов и зарегистрированных объектах.
- [HKEY_CURRENT_USER](#) – содержит настройки для текущего пользователя.
- [HKEY_LOCAL_MACHINE](#) – содержит настройки локального компьютера.
- [HKEY_USERS](#) – содержит настройки пользователя по умолчанию.
- [HKEY_CURRENT_CONFIG](#) – содержит информацию о настройках, которые не являются специфическими для пользователя.
- [HKEY_PERFORMANCE_DATA](#) – содержит информацию о производительности программных компонентов.

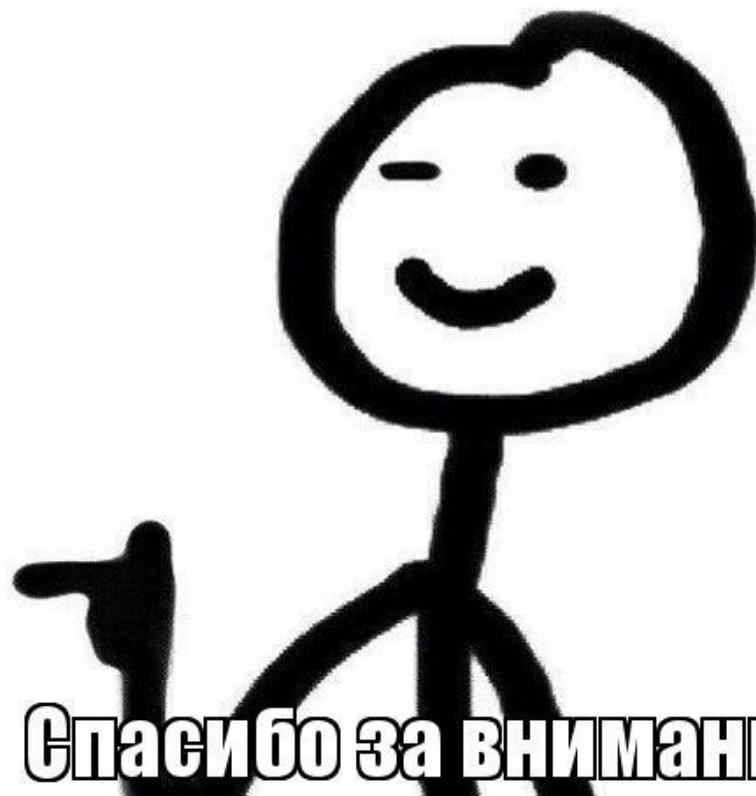
Registry

[Registry](#) – этот класс предоставляет набор стандартных корневых разделов, находящихся в реестре компьютеров, работающих под управлением Windows. Реестр является средством хранения сведений о приложениях, пользователях и стандартных системных параметрах. Например, приложения используют реестр для хранения сведений, которые необходимо сохранить после закрытия приложения и к которым необходимо получать доступ при перезагрузке приложения. Например, можно сохранять цветовые настройки, положение или размер окна. Для разных пользователей эти сведения могут сохраняться в различных местах реестра.

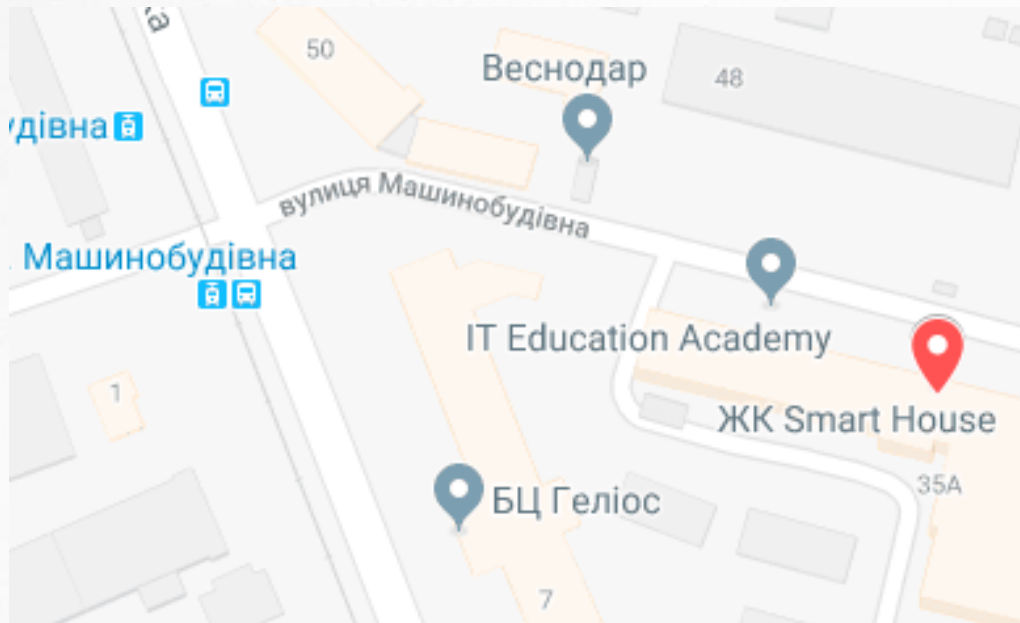
Чтение из реестра осуществляется при помощи метода [GetValue](#), узел должен быть предварительно открыт при помощи метода [OpenSubKey](#).

Запись данных в реестр осуществляется при помощи метода [SetValue](#), узел должен существовать и быть предварительно открытым при помощи метода [OpenSubKey](#) с указанием признака «открытие для записи».

Презентация окончена.



Спасибо за внимание!



КОНТАКТНЫЕ ДАННЫЕ

ITEA

ЖК “Smart House”, ул.
 Машиностроительная, 41
 (м.Берестейская)

ЖК «Корона» улица
 Срибнокильская, 1
 м. Позняки

+38 (044) 599-01-79

facebook.com/Itea

info@itea.ua

itea.ua

