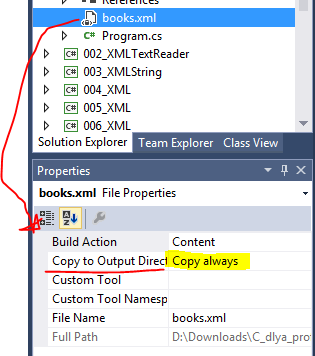
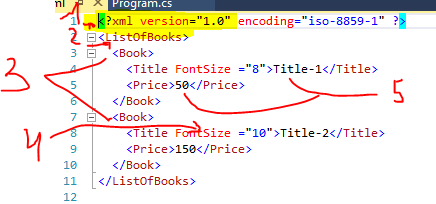


В окне свойств для файла book.xml, свойству - Copy to Output Directory, присвойте значение - Copy always.

Это необходимо чтобы файл book.xml копировался в папку рядом с \*.exe файлом.



1. Декларация документа (версия и кодировка)
2. Корневой узел (может быть только один)
3. Два элемента
4. Атрибут
5. Заглавие книги

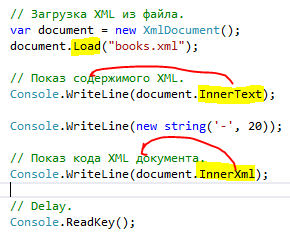


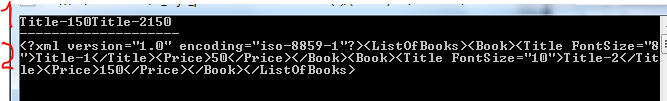
XmlDocument()

document.Load("books.xml"); - загрузка документа

document.InnerText – показ содержимого (вернет конкатенацию всех значений)

document.InnerXml – показ кода XML

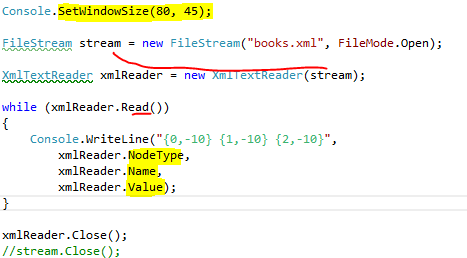


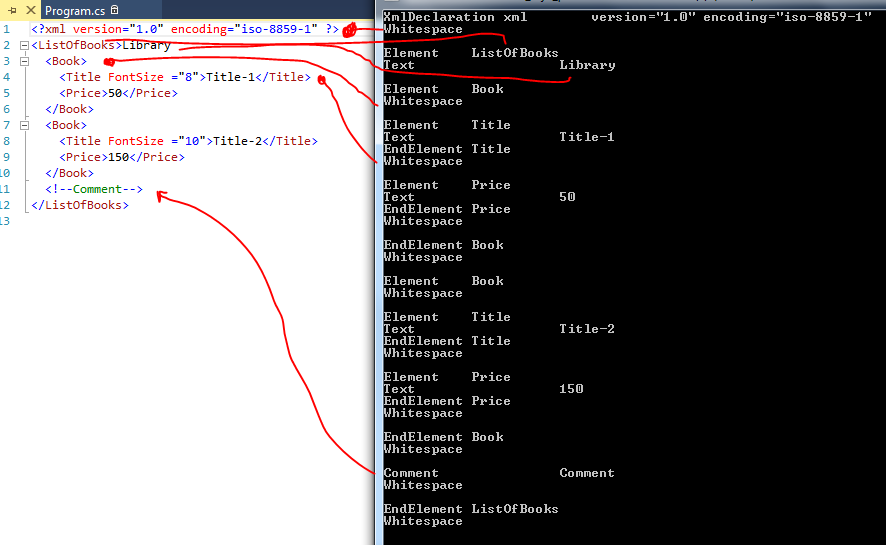


XmlTextReader

new XmlTextReader(stream); - передается побитовый поток (создается декорированный поток)

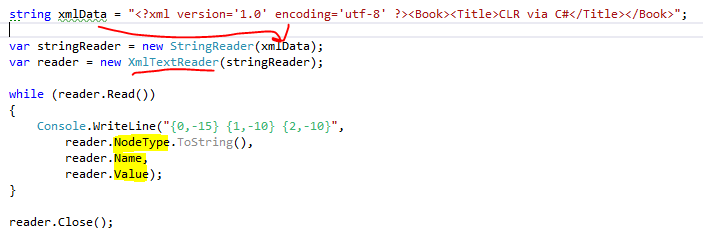
xmlReader.Read() – будет возвращать true пока может читать

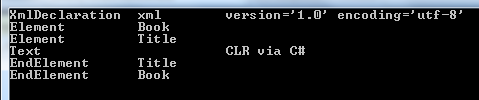




XmlTextReader – если документ находиться в строке

Строка передается в поток StringReader и после поток передается в XmlTextReader



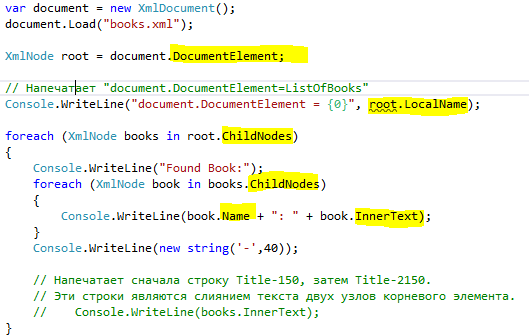


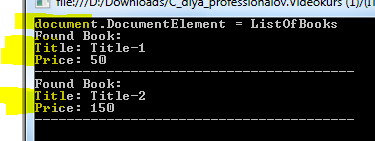
XmlNode

Обход всех элементов XML файла

document.DocumentElement – возвращает корневой элемент

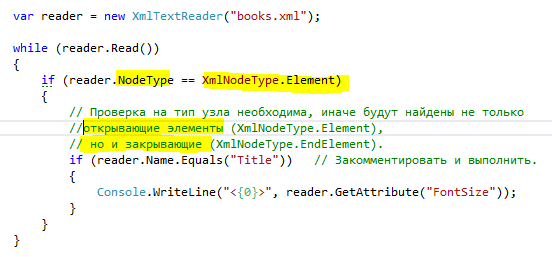
root.ChildNodes – возвращает коллекцию всех дочерних элементов корневого документа





Ищем только нужные нам элементы

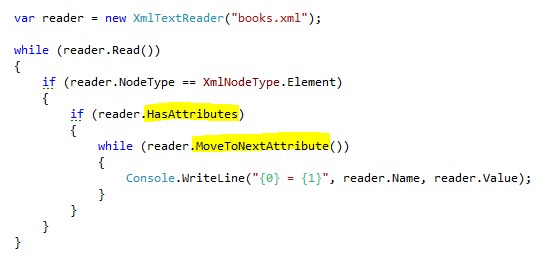
reader.GetAttribute("FontSize") – получение атрибута элемента

Вывод значений всех атрибутов

reader.HasAttributes – есть ли атрибут

reader.MoveToNextAttribute() – прочитали, и если возвращает true то атрибут есть



XmlTextWriter

– создать и записать XML документ

new XmlTextWriter("books.xml", null); - название документа, 2й параметр кодировка документа (кодировка по умолчанию)

или - new XmlTextWriter("books.xml", Encoding.GetEncoding(1251));

xmlWriter.WriteStartDocument();- запишет в файл декларацию Заголовок XML - <?xml version="1.0"?>

xmlWriter.WriteStartElement("ListOfBooks"); - открыли элемент, запишет элемент // Корневой элемент - <ListOfBooks> (обязательно нужно закрывать в конце кода)

xmlWriter.WriteStartAttribute("FontSize"); - к открытому элементу будет добавлен атрибут // Атрибут – FontSize (обязательно нужно закрывать)

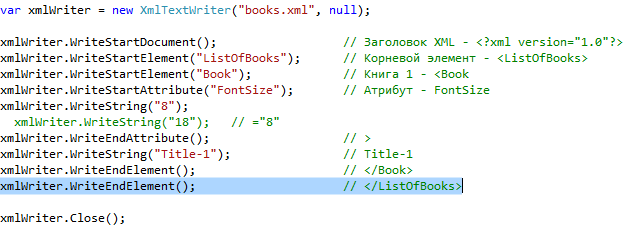
xmlWriter.WriteString("8"); - записать значение атрибута

xmlWriter.WriteEndAttribute(); - закрыть открытый атрибут

xmlWriter.WriteString("Title-1"); - записать значение элемента который сейчас открыт и на котором находимся // Title-1

xmlWriter.WriteEndElement(); - закрыть открытый элемент // </ListOfBooks>

xmlWriter.WriteComment("Строка комментария."); - добавить комментарий



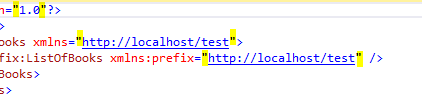
Запись XML документа в форматированном виде

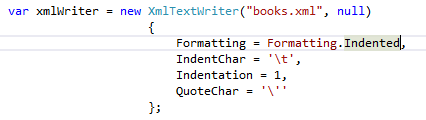
Formatting = Formatting.Indented – запись будет осуществлена в форматированном виде

IndentChar = '\t' – отступ следующего уровня вложенности

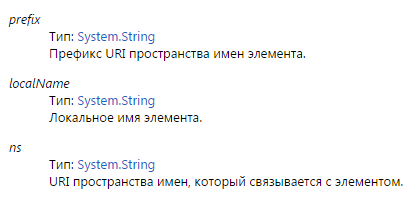
Indentation = 1 – количество символов которые будут разделять один уровень вложенности от другого

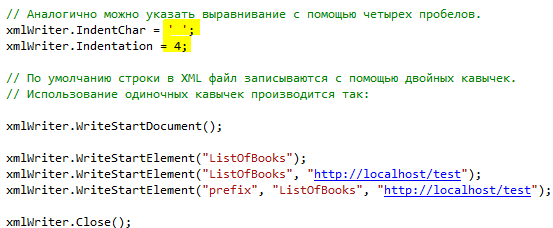
QuoteChar = '\'' – одиночная кавычка (по умолчанию двойная)



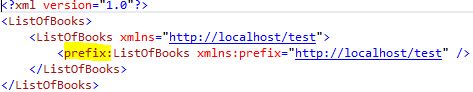


Перегрузка WriteStartElement





результат





XPathDocument - Единственное назначение XPathDocument - создание навигатора.

При создании навигатора при помощи XPathDocument возможно выполнять только чтение.

Предоставляет быстрый, доступное только для чтения представление xml-документа в памяти с использованием модели данных XPath.

var document = new XPathDocument("books.xml"); - создание документа XPath

XPathNavigator

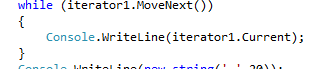
XPathNavigator navigator = document.CreateNavigator(); - возвращает экземпляр класса XPathNavigator

XPathNodeIterator

// Прямой запрос XPath.

XPathNodeIterator iterator1 = navigator.Select("ListOfBooks/Book/Title"); - передается путь к элементу который интересует



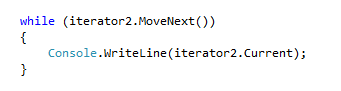


Выведет два Title т.к. в ListOfBooks два Book и в них есть Title 

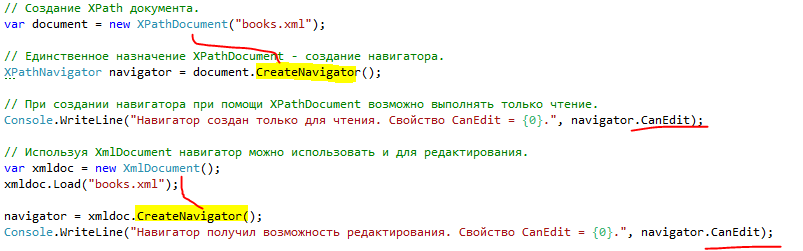
XPathExpression – обращение к конкретному элементу (начало элементов с 1)

XPathExpression expr = navigator.Compile("ListOfBooks/Book[2]/Price");- возвращает экземпляр класса XPathExpression

XPathNodeIterator iterator2 = navigator.Select(expr);

Используя XmlDocument навигатор можно использовать для чтения и для редактирования.



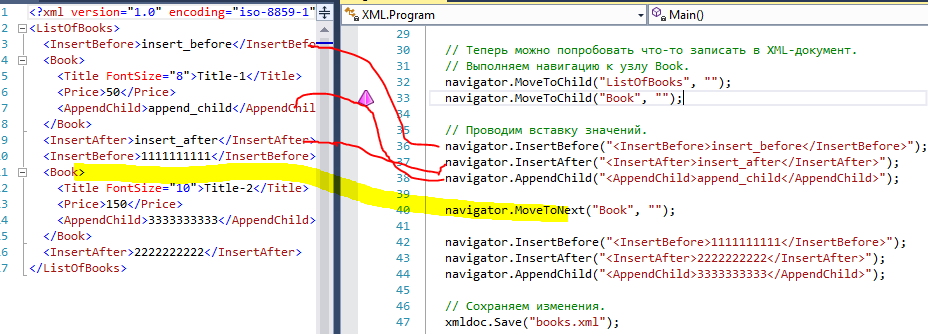
navigator.MoveToChild("ListOfBooks", ""); - выполняем навигацию к узлу

navigator.MoveToChild("Book", ""); - выполняем навигацию к узлу

navigator.InsertBefore(" – вставить до элемента на котором стоим

navigator.InsertAfter("<– вставить после элемента на котором стоим

navigator.AppendChild("<– вставить в этот элемент на котором стоим



Обязательно Сохраняем изменения.

xmldoc.Save("books.xml");