

XML И XSLT.

Современные технологии

обработки данных для

ВЕБ

Игорь Борисов
<http://igor-borisov.ru>

Рассматриваемые темы

- Введение в XML
- Описание структуры документа - DTD
- Описание структуры документа – XML схемы
- Введение в XSLT, XPath
- Основные элементы XSLT
- Дополнительные возможности XSLT
- Практическая работа с XSLT
- Интеграция XML данных

Введение в XML

Игорь Борисов
<http://igor-borisov.ru>

Темы модуля

- Введение в языки разметки
- Представление данных
- Структура XML документа
- XML – представление любых данных
- Синтаксис, грамматика и семантика элементов
- Пространства имен
- Привязка пространств имен
- Применение XML

Введение в языки разметки

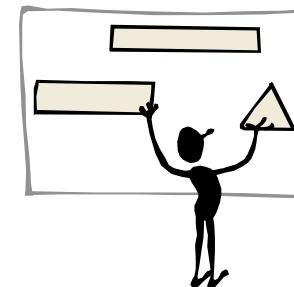
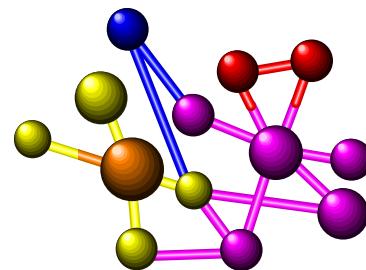
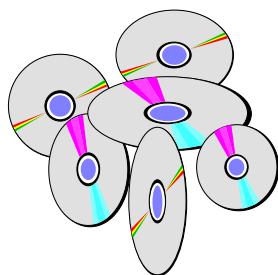
- Язык разметки позволяет представить в виде текстового документа данные (текст, информацию и т.п.)
- Существует множество языков разметки
- Как правило, они не универсальны

XML – это...

- Универсальный, расширяемый язык разметки данных
- Независимость от платформы, операционной системы и программ обработки
- Представление произвольных данных и дополнительной метаинформации

Представление данных

- Документ может содержать...
 - Данные
 - Структуру
 - Уровень представления



Различия XML и HTML

- HTML описывает **ИЗ ЧЕГО СОСТОИТ** и **КАК** отображать документ

```
<TABLE>
  <TR>
    <TD>Name</TD><TD>Jake</TD>
  </TR>
  <TR>
    <TD>Salary</TD><TD>25000</TD>
  </TR>
  <TR>
    <TD>Region</TD><TD>Ohio</TD>
  </TR>
</TABLE>
```

- XML определяет **ЗНАЧЕНИЕ** и **ОТНОШЕНИЕ** данных

```
<employee>
  <name>Jake</name>
  <salary>25000</salary>
  <region>Ohio</region>
</employee>
```

Элементы и тэги

- XML состоит из элементов
- Для определения данных используются тэги

<name>

Lars Peterson

</name>

- Элементы могут быть вложенными

```
<employee>
    <name>Lars Peterson</name>
    <salary>25000</salary>
</employee>
```

Синтаксис, грамматика и семантика элементов

- Синтаксис – правила записи элементов разметки
- Грамматика – отношения элементов разметки друг к другу
- Семантика – смысловые значения элементов разметки

- Чем отличаются теги HTML?
 - и
 - <i> и
 - <p> и <div>

XML разметка

```
<?xml version="1.0" encoding="windows-1251" ?>
<!-- Пример XML разметки -->
<pricelist>
    <book id="1">
        <title>XML и IE5</title>
        <author>Алекс Гомер</author>
        <price currency="RUR">200</price>
        <pubDate>2012-02-25</pubDate>
    </book>
</pricelist>
```

- XML декларация (пролог)
- Комментарий
- Элемент документа (корневой элемент)
- Элемент
- Атрибут
- Текстовые данные

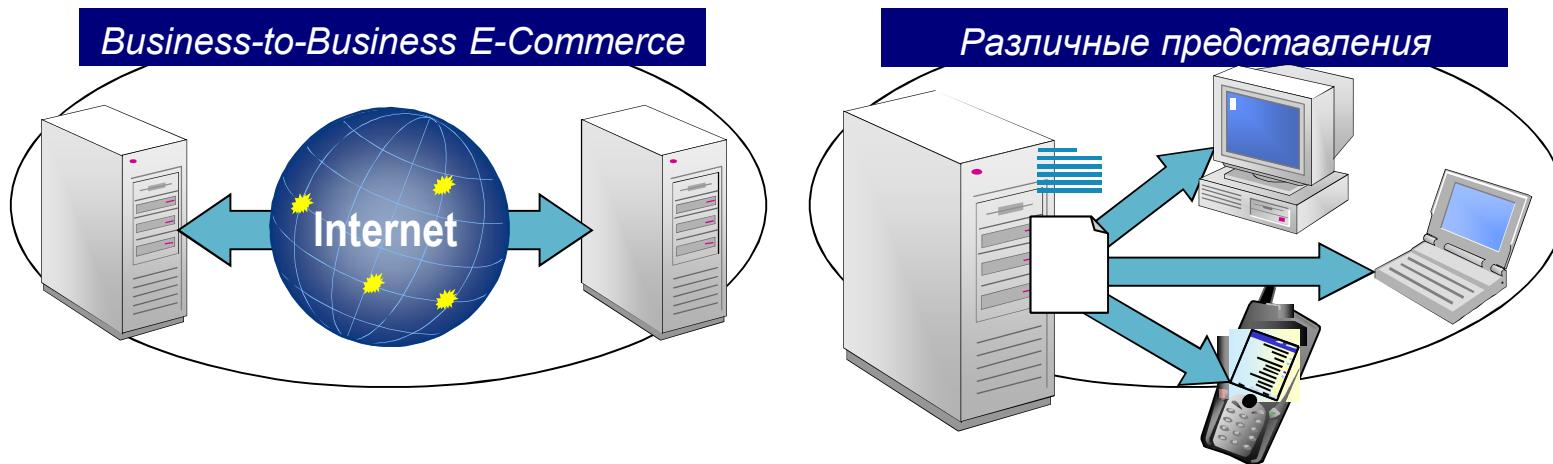
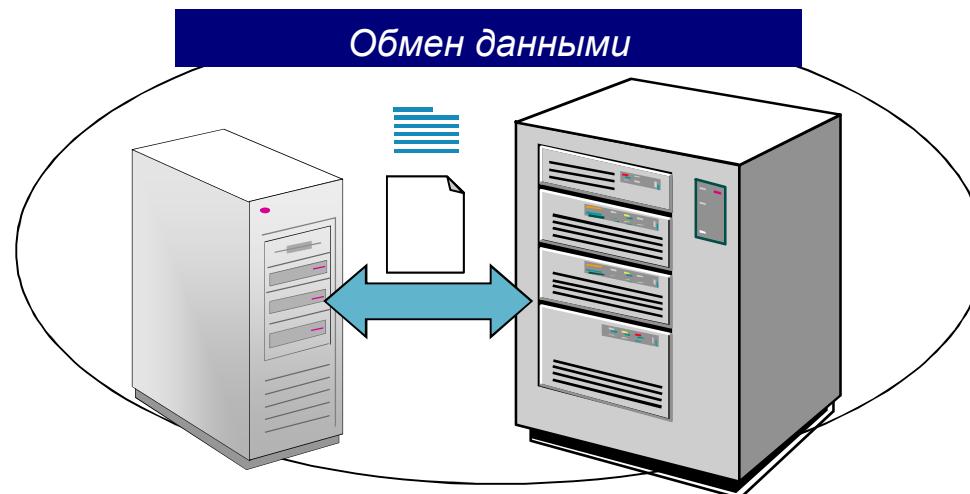
Что ещё может быть в XML?

- Процессинговые инструкции
 - <? ?>
- Описание грамматики
 - DTD или XML-схема
- Пустые (empty) элементы
 - <emptyElement />
- Секция CDATA
 - <description>
 <![[CDATA[Какой-то
 текст]]]>
 </description>

Well-formed document

- Корректный (хорошо сформированный) XML-документ
 - Если документ содержит символы, выходящие за рамки ASCII, необходимо указать кодировку
 - Должен существовать только один корневой элемент, который содержит все остальные элементы
 - XML чувствителен к регистру символов
 - Имя может начинаться с буквы, подчёркивания, двоеточия. Остальными символами имени могут быть те же символы, а также цифры, дефис, точка
 - Теги должны быть правильно вложены друг в друга
 - Все парные теги должны быть закрыты
 - Возможно формирование пустых элементов – без содержимого
 - Значения атрибутов заключаются в одинарные или двойные кавычки
 - У каждого конкретного элемента не должно быть повторяющихся атрибутов

XML-представление любых данных



Лабораторная работа 1.1

Создание простого XML-документа

Упражнение 1: Подготовка к работе

- Откройте проводник Windows и перейдите в раздел диска «C»
(В реальных условиях у Вас при работе с правами пользователя может не быть доступа на запись в другие папки, кроме папки «Мои документы». На этом курсе расположение папок с примерами и лабораторными работами роли не играет, и поэтому мы с Вами используем наиболее короткий для записи путь.)
- Создайте папку «XML»

Упражнение 2: Создание XML документа

- Создайте пустой текстовый документ **lab-1.xml**
- Щелкните по нему правой кнопкой мышки и выберите «**Edit with Notepad++**»
(На этом курсе мы будем использовать бесплатный и очень мощный редактор Notepad++. Получить его бесплатно можно по адресу: <http://notepad-plus.sourceforge.net/ru/site.htm>. Если Вы предпочитаете другой редактор – пользуйтесь другим редактором.)
- Установите кодировку документа **UTF-8**
[Кодировки -> Кодировать в UTF-8]
- Впишите пролог документа:
`<?xml version="1.0" encoding="UTF-8"?>`
- Создайте XML разметку, отражающую ваши занятия на этом курсе, например (будьте внимательны, в примере могут быть опечатки!):
`<timeTable>
 <lesson no=1>
 <date>2012-03-21</date>
 <time>18:30:00</time>
 </lesson>
</TimeTable>`
- Сохраните файл
- Проверьте правильность вашего документа:
 - Для этого можно просто открыть его с помощью браузера...
 - Или в редакторе Notepad++ выбрать следующую возможность:
[Дополнения -> XML Tools -> Check XML syntax now]
(Чтобы эта возможность была Вам доступна необходимо установить дополнение XMLTools для Notepad++. Получить бесплатно это дополнение можно через Менеджер Расширений редактора: Плагины -> Plugin Manager -> Show Plugin Manager)
- Если есть ошибки, исправьте их и сохраните свою работу

Пространства имён

- Устраняют конфликты элементов
- Объявляются с помощью URI
 - Uniform Resource Identifier
- URI:
 - URN
 - urn:schemas-microsoft-com:xml-data
 - URL
 - <http://www.w3.org/1999/XSL/Transform>

Объявление пространств имен

- С помощью атрибута **xmlns**

- <catalog xmlns = "http://megobook.ru/catalog">
 <book>
 <title>Мега Книга</title>
 <author>Иван Петров</author>
 </book>
 </catalog>
 - <catalog>
 <item xmlns = "http://megashop.ru/catalog/music">
 <title>Мега Альбом</title>
 </item>
 <item xmlns = "http://megashop.ru/catalog/book">
 <title>Мега Книга</title>
 </item>
 </catalog>

Объявление пространств имен

- С помощью префикса `xmlns:prefix`

- ```
<c:catalog xmlns:c = "http://megabook.ru/catalog">
 <book>...</book>
</c:catalog>
```
  - ```
<catalog xmlns = "http://megabook.ru/catalog"
          xmlns:m = "http://megashop.ru/catalog/music"
          xmlns:b = "http://megashop.ru/catalog/book">
<m:item>
    <m:title>Мега Альбом</m:title>
</m:item>
<b:item>
    <b:title>Мега Книга</b:title>
</b:item>
</catalog>
```

Лабораторная работа 1.2

Использование пространств имен XML

Упражнение 1: Создание XML-документа с пространствами имен XML

- Откройте файл **lab-1-1.xml** с помощью Notepad++
- Пересохраните его с именем **lab-1-2.xml**
- Измените свой документ так, чтобы он использовал пространство имен <http://www.mysupersite.ru/ns/courses/xml/lab1>
- Проверьте XML-документ на правильность
- Допишите в XML-документ информацию о комплексе, в котором проходят занятия. При этом необходимо использовать следующие данные:

```
<offices xmlns="http://www.mysupersite.ru/ns/courses/xml/office">
    <office code="БК">
        <name>Комплекс "Бакунинский"</name>
        <address>ул. Бакунинская, 71</address>
    </office>
    <office code="РА">
        <name>Комплекс "Радио"</name>
        <address>ул. Радио, д. 24</address>
    </office>
</offices>
```

- Обратите внимание на пространство имен этого документа
- Используйте указанное пространство имен с помощью префикса в своем документе
- Проверьте XML-документ на правильность
- Сохраните свою работу

Примеры XML данных

- XML – очень распространенная технология
- Представление любых данных
- Четко выраженная структура данных
- Сериализация данных

Электронные книги: FB2

II. Открытие профес-
сора Декантора

Сергей Лукьяненко

Name	Size	Author	Title	Year
Сергей Лукьяненко. Близится...	630 KB	Сергей Лукьяненко	Близится утро	1999
Сергей Лукьяненко. Геном	649 KB	Сергей Лукьяненко	Геном	1999
Сергей Лукьяненко. Звезды...	625 KB	Сергей Лукьяненко	Звездная тень	Январь
Сергей Лукьяненко. Звезды...	Type: FictionBook	Сергей Лукьяненко	Звезды – холодны...	Апрель
Сергей Лукьяненко. Лабиринт...	648 KB	Сергей Лукьяненко	Лабиринт Отражение...	Апрель
Сергей Лукьяненко. Линия Грэз	1999	Сергей Лукьяненко	Линия Грэз	1995
Сергей Лукьяненко. Мой Папа - Антибиотик	Genre: SF	Сергей Лукьяненко	Мой Папа - Антибиотик	1992
Сергей Лукьяненко. Ночной дозор	272 KB	Сергей Лукьяненко	Ночной дозор	Январь
Сергей Лукьяненко. Пограничное время	368 KB	Сергей Лукьяненко	Пограничное время	2002
Сергей Лукьяненко. СПЕКТР. Каждый охотник...	991 KB	Сергей Лукьяненко	СПЕКТР. (КАЖДЫЙ ОХОТНИК ЖЕЛАЕТ ЗНАТЬ)	2002
Сергей Лукьяненко. Танцы на снегу	711 KB	Сергей Лукьяненко	Танцы на снегу	2000
Сергей Лукьяненко. Тени Снов	155 KB	Сергей Лукьяненко	Тени Снов	1998

Сергей Лукьяненко, Ник Перумов

Name	Size	Author	Title	Year
Сергей Лукьяненко, Ник Перумов. Не время для драк...	842 KB	Сергей Лукьяненко, Ник Перумов	Не время для драк...	декабрь 2000

Станислав Лем

Name	Size	Author	Title	Year
Станислав Лем. Магелланово облако	762 KB	Станислав Лем	Магелланово облако	1955
Станислав Лем. Сумма технологий	1 272 KB	Станислав Лем	Сумма технологий	1967

Стивен Кинг

Name	Size	Author	Title	Year
Стивен Кинг. Кодовые имена	1 410 KB	Стивен Кинг	Кодовые имена	2002

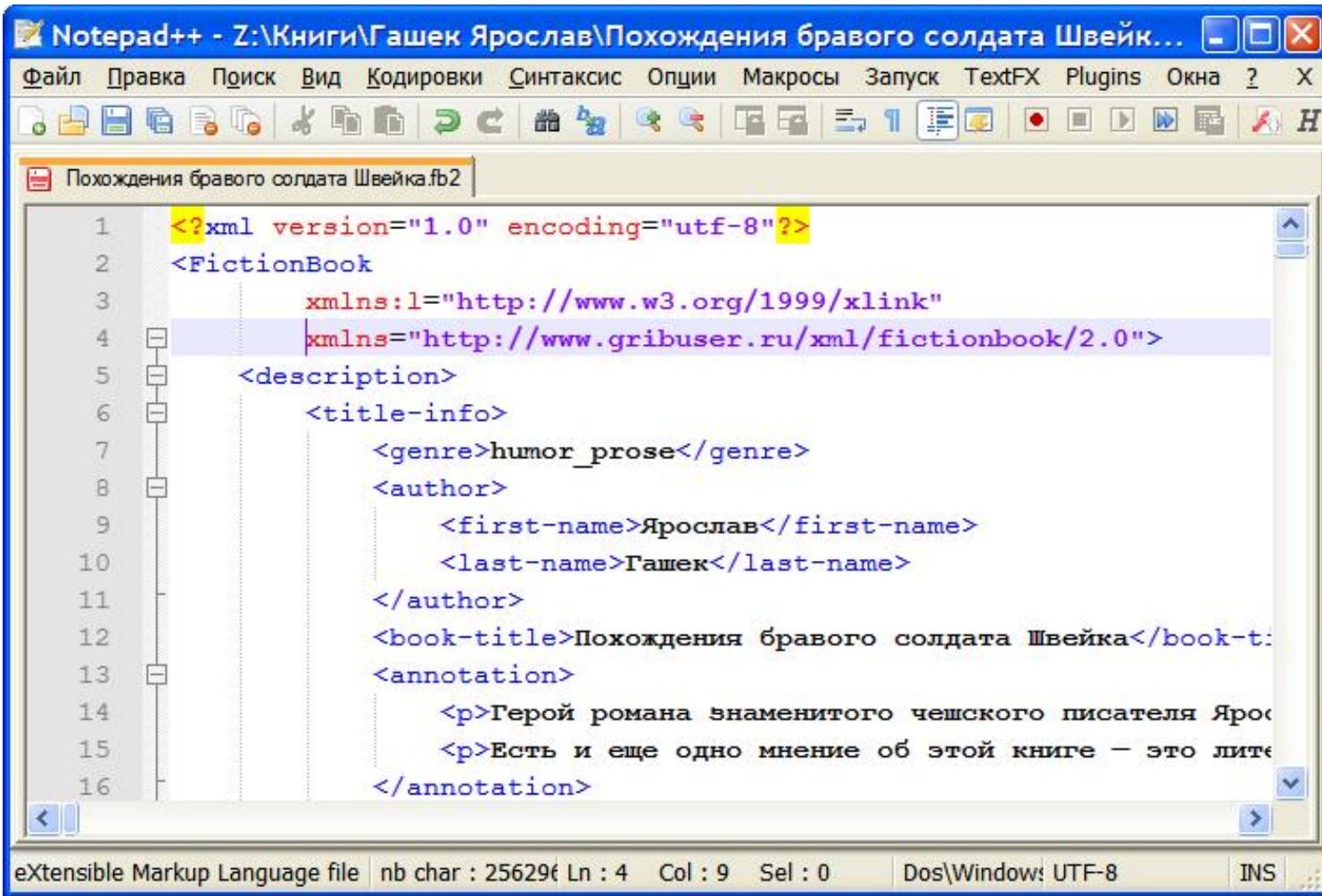
FictionBook
Author: Сергей Лукьяненко
Title: СПЕКТР. (КАЖДЫЙ ОХОТНИК ЖЕЛАЕТ ЗНАТЬ)
Year: 2002
Genre: Science Fiction
Size: 990 KB

10.86% 14/134 12:35

Type: FictionBook Author: Сергей Лукьяненко Title: СПЕКТР. (КАЖДЫЙ ОХОТНИК ЖЕЛАЕТ ЗНАТЬ) Year: 2002 Genre: Sci Fi Size: 990 KB

My Computer

FB2 – XML документ

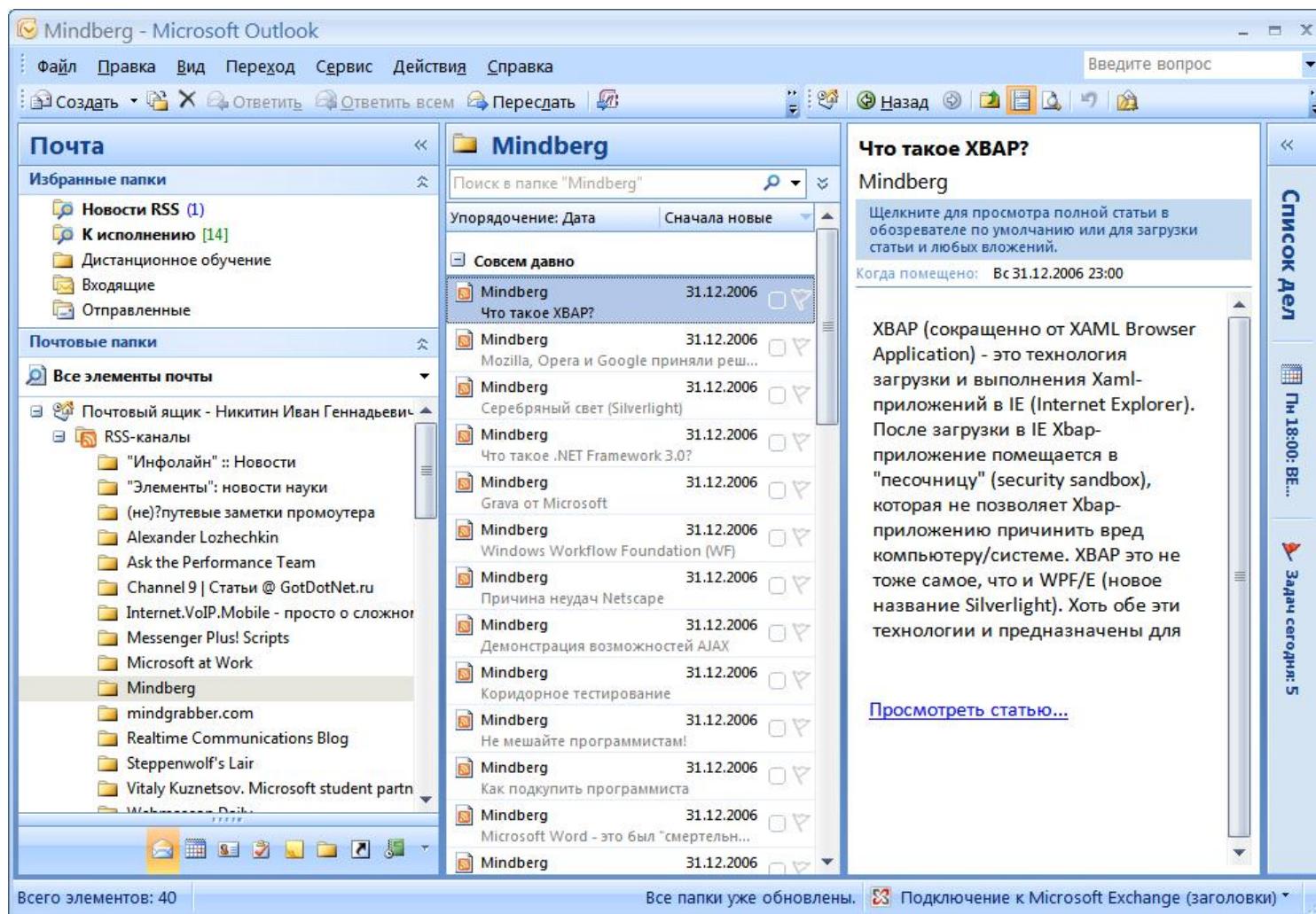


The screenshot shows the Notepad++ text editor displaying an FB2 (FictionBook) XML document. The window title is "Notepad++ - Z:\Книги\Гашек Ярослав\Похождения бравого солдата Швейк...". The menu bar includes Файл, Правка, Поиск, Вид, Кодировки, Синтаксис, Опции, Макросы, Запуск, TextFX, Plugins, Okna, ?, and X. The toolbar contains various icons for file operations like Open, Save, Find, and Replace. The code editor pane shows the following XML structure:

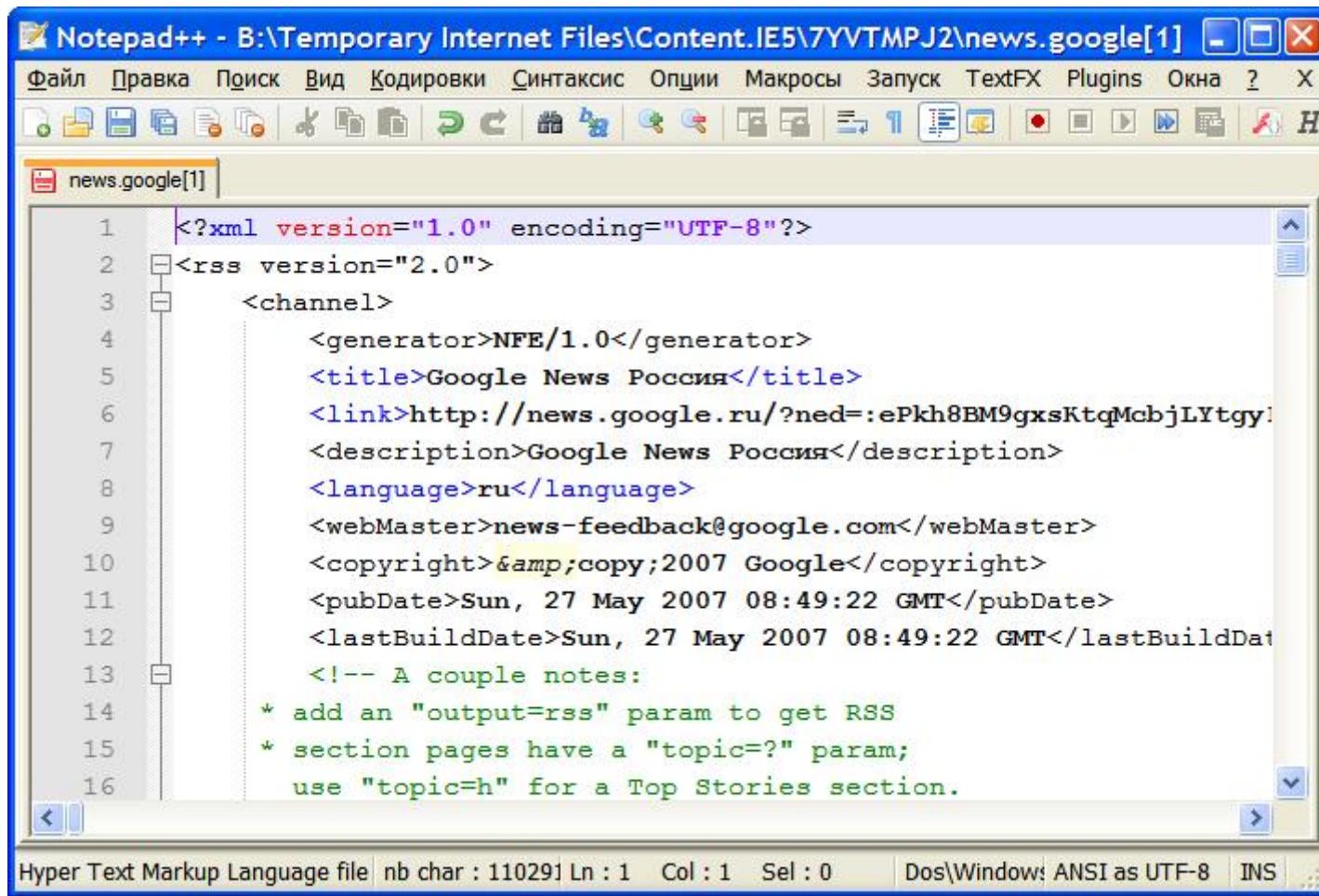
```
<?xml version="1.0" encoding="utf-8"?>
<FictionBook
    xmlns:l="http://www.w3.org/1999/xlink"
    xmlns="http://www.gribuser.ru/xml/fictionbook/2.0">
    <description>
        <title-info>
            <genre>humor_prose</genre>
            <author>
                <first-name>Ярослав</first-name>
                <last-name>Гашек</last-name>
            </author>
            <book-title>Похождения бравого солдата Швейка</book-ti:
            <annotation>
                <p>Герой романа знаменитого чешского писателя Яросла
                <p>Есть и еще одно мнение об этой книге – это лите
            </annotation>
        </title-info>
    </description>
</FictionBook>
```

The status bar at the bottom indicates "eXtensible Markup Language file" and "nb char : 256296 Ln : 4 Col : 9 Sel : 0 Dos\Windows! UTF-8 INS".

RSS: НОВОСТИ



RSS – XML документ



The screenshot shows the Notepad++ application window with the title bar "Notepad++ - B:\Temporary Internet Files\Content.IE5\7YVTMPJ2\news.google[1]". The menu bar includes Файл, Правка, Поиск, Вид, Кодировки, Синтаксис, Опции, Макросы, Запуск, TextFX, Plugins, Окна, ?, and X. The toolbar contains various icons for file operations like Open, Save, Find, and Replace. The main editor area displays an RSS feed XML document named "news.google[1]". The XML code is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0">
  <channel>
    <generator>NFE/1.0</generator>
    <title>Google News Россия</title>
    <link>http://news.google.ru/?ned=:ePkh8BM9gxsKtqMcbjLYtgy:</link>
    <description>Google News Россия</description>
    <language>ru</language>
    <webMaster>news-feedback@google.com</webMaster>
    <copyright>&copy;2007 Google</copyright>
    <pubDate>Sun, 27 May 2007 08:49:22 GMT</pubDate>
    <lastBuildDate>Sun, 27 May 2007 08:49:22 GMT</lastBuildDate>
    <!-- A couple notes:
    * add an "output=rss" param to get RSS
    * section pages have a "topic=?" param;
      use "topic=h" for a Top Stories section.
```

The status bar at the bottom shows "Hyper Text Markup Language file" and "nb char : 110291 Ln : 1 Col : 1 Sel : 0 Dos\Windows: ANSI as UTF-8 INS".

Документы Microsoft Office

Современные XML технологии - Microsoft Word

Главная Вставка Разметка страницы Ссылки Рассылки Рецензирование Вид Надстройки

Шрифт Arial 12

Абзац

Стили Экспресс-стили Изменить стили Редактирование

Увеличить отступ Увеличение уровня отступа абзаца.

Современные XML технологии

Анкета курса
«Современные XML технологии»
«Modern XML technologies»

Цель курса

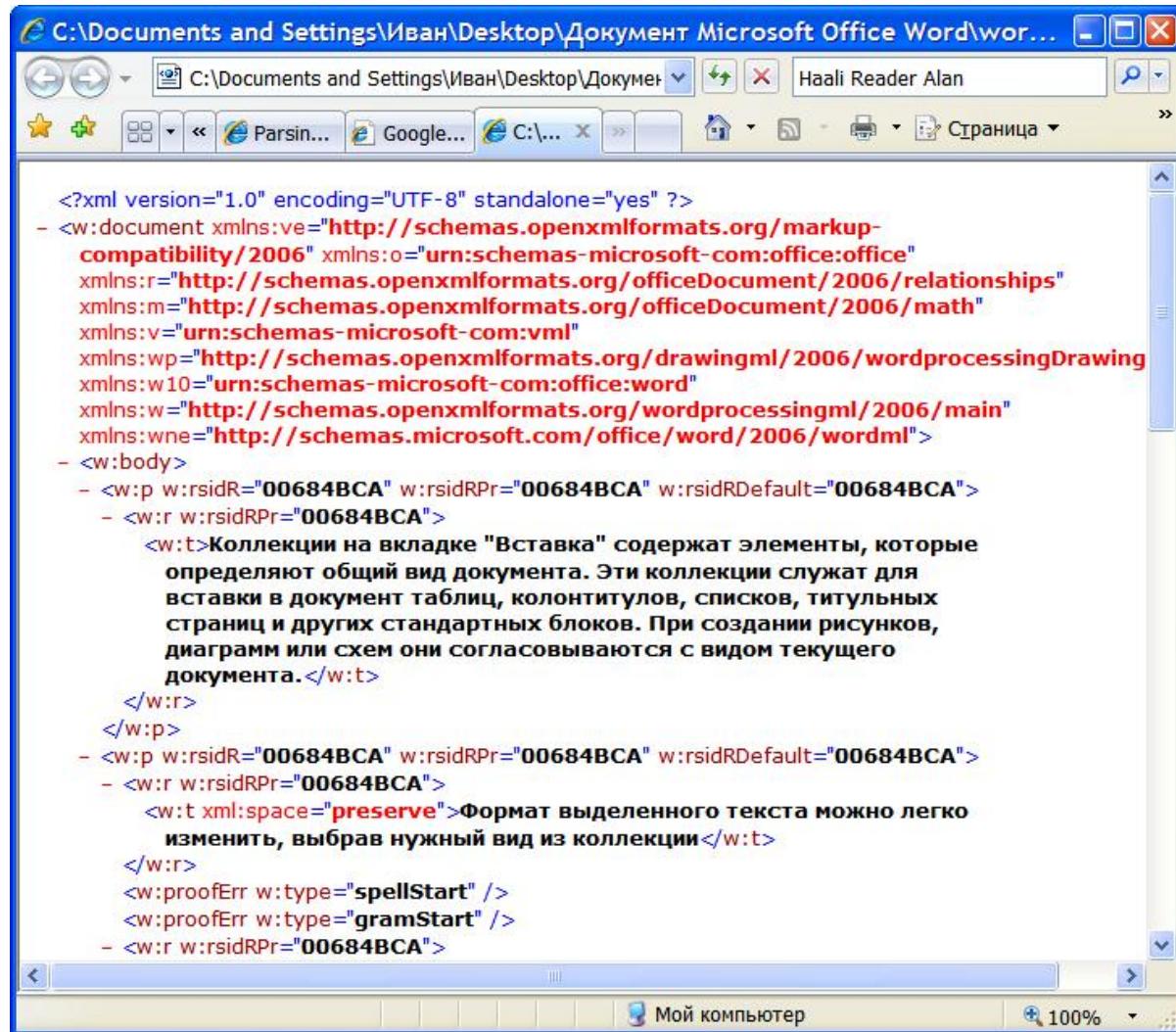
Цель данного курса дать слушателям представление о современных XML технологиях, научить слушателей применять эти технологии на практике, продемонстрировать современные подходы к обработке XML данных. Курс рассчитан на подготовленных слушателей, имеющих опыт в построении сайтов и использовании Веб технологий.

Программа курса

	Тема	Академ. часов	Занятий
1	Введение в XML	8	2
2	DOM	8	2
3	Проверка лекции	8	2

Страница: 1 из 4 Число слов: 951 английский (США) 100%

Office 200x – XML документ



The screenshot shows a window titled 'Haali Reader Alan' displaying an XML document. The document's title bar indicates it is located at 'C:\Documents and Settings\Иван\Desktop\Документ Microsoft Office Word\wor...'. The window has a standard toolbar with icons for back, forward, search, and print. Below the toolbar is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Format', 'Tools', 'Help', and a 'Справка' (Help) dropdown.

The main content area displays the XML code of a Microsoft Word document. The code includes declarations for namespaces such as `<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>`, `<w:document xmlns:ve="http://schemas.openxmlformats.org/markup-compatibility/2006" ...>`, and various element definitions like `<w:body>`, `<w:p>`, and `<w:r>`. The text content within the XML elements is in Russian, describing features of the 'Insert' tab in Word, such as 'Коллекции на вкладке "Вставка" содержат элементы, которые определяют общий вид документа.' (Collections on the 'Insert' tab contain elements that define the overall appearance of the document.)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<w:document xmlns:ve="http://schemas.openxmlformats.org/markup-compatibility/2006" xmlns:o="urn:schemas-microsoft-com:office:office" xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships" xmlns:m="http://schemas.openxmlformats.org/officeDocument/2006/math" xmlns:v="urn:schemas-microsoft-com:vml" xmlns:wp="http://schemas.openxmlformats.org/drawingml/2006/wordprocessingDrawing" xmlns:w10="urn:schemas-microsoft-com:office:word" xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main" xmlns:wne="http://schemas.microsoft.com/office/word/2006/wordml">
<w:body>
<w:p w:rsidR="00684BCA" w:rsidRPr="00684BCA" w:rsidRDefault="00684BCA">
<w:r w:rsidRPr="00684BCA">
<w:t>Коллекции на вкладке "Вставка" содержат элементы, которые определяют общий вид документа. Эти коллекции служат для вставки в документ таблиц, колонтитулов, списков, титульных страниц и других стандартных блоков. При создании рисунков, диаграмм или схем они согласовываются с видом текущего документа.</w:t>
</w:r>
</w:p>
<w:p w:rsidR="00684BCA" w:rsidRPr="00684BCA" w:rsidRDefault="00684BCA">
<w:r w:rsidRPr="00684BCA">
<w:t xml:space="preserve">Формат выделенного текста можно легко изменить, выбрав нужный вид из коллекции</w:t>
</w:r>
<w:proofErr w:type="spellStart" />
<w:proofErr w:type="gramStart" />
<w:r w:rsidRPr="00684BCA">
```

XHTML

World Wide Web Consortium - Mozilla Firefox

Файл Правка Вид Журнал Закладки Инструменты Справка

http://www.w3.org/ Google

S SIPNET Инфолайн Просмотр HTTP загол...

FireFox: Загрузка документ... FireFox: Загрузка документ... World Wide Web Consorti...

W3C® WORLD WIDE WEB
c o n s o r t i u m

Leading the Web to Its Full Potential...

[Activities](#) | [Technical Reports](#) | [Site Index](#) | [New Visitors](#) | [About W3C](#) | [Join W3C](#) | [Contact W3C](#)

The World Wide Web Consortium (W3C) develops interoperable technologies (specifications, guidelines, software, and tools) to lead the Web to its full potential. W3C is a forum for information, commerce, communication, and collective understanding. On this page, you'll find W3C news, links to W3C technologies and ways to get involved. New visitors can find help in [Finding Your Way at W3C](#). We encourage organizations to learn more about W3C and [about W3C Membership](#).

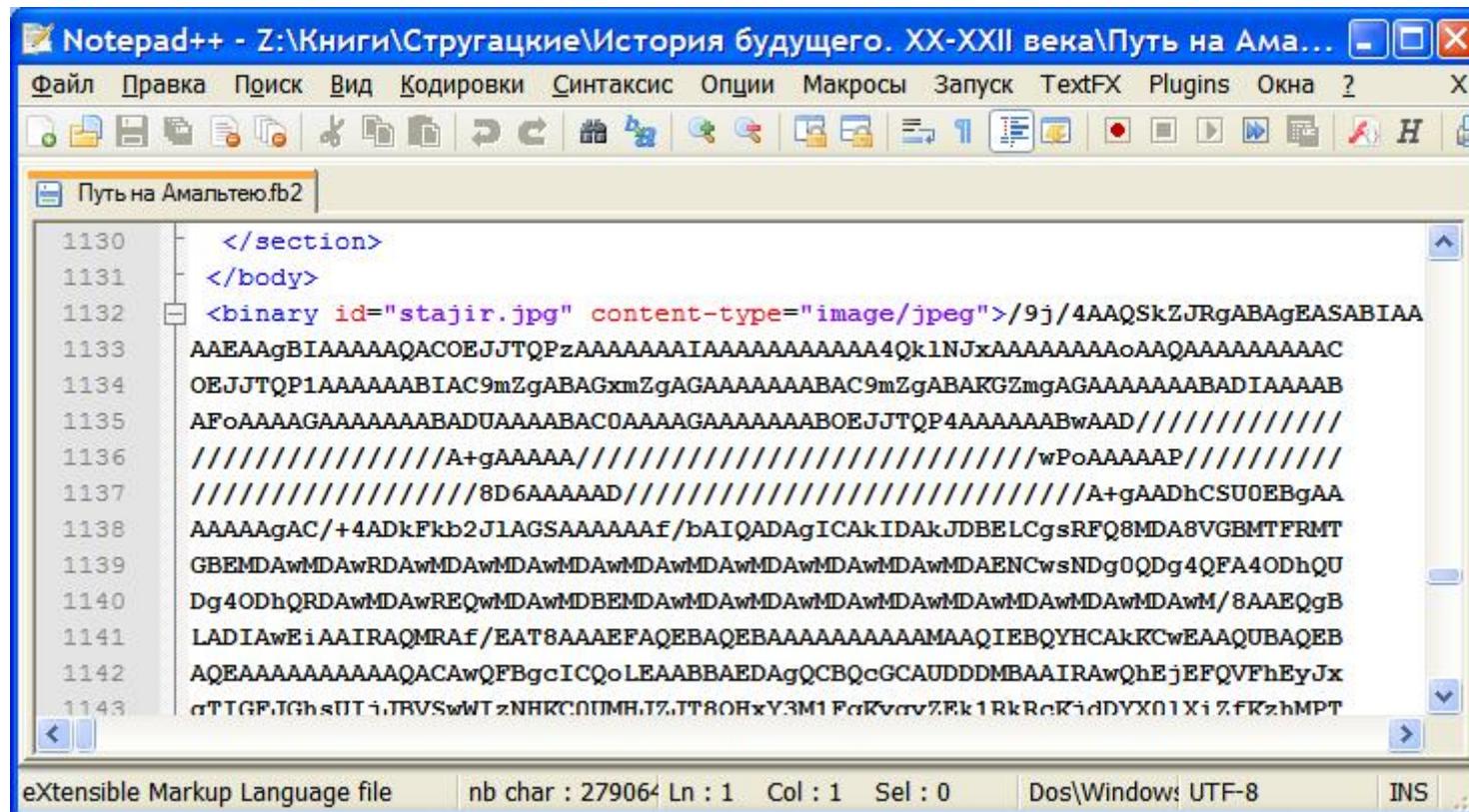
W3C Supporters Help W3C by making a donation through the W3C Supporters Program .	News Web Services Description Language (WSDL) 2.0 Is a Proposed Recommendation 2007-05-23: W3C is pleased to	Search Google Search W3C <input type="text"/> Go Search W3C Mailing Lists Testimonials
---	--	--

Готово Fasterfox

XHTML – XML документ

Представление бинарных данных

- Обычно используется сериализация Base64 (<http://ru.wikipedia.org/wiki/Base64>)



Выводы

- XML – это представление любых данных
- Структура XML-документа отражает структуру данных
- У языка есть синтаксис, грамматика и семантика
- Пространства имён – способ решения конфликтов с именами
- Привязка пространств имён через атрибуты
- XML применяется в самых разных областях

Описание структуры документа - DTD

Игорь Борисов
<http://igor-borisov.ru>

Темы модуля

- Представление документа в виде дерева объектов
- Способы программного анализа документа
- DTD – описание типов документа
- Построение DTD
- Применение DTD
- Проверка структуры с помощью DTD

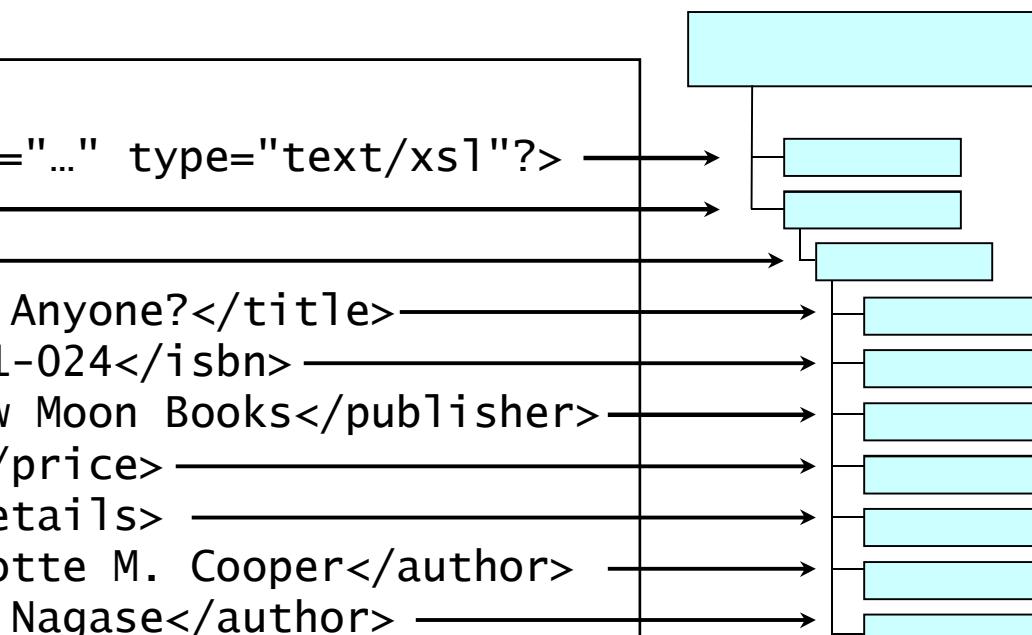
Способы программного анализа документа

- Для программного анализа используются парсеры XML
- Парсер – синтаксический анализатор документа
- SAX – Simple API of XML
- DOM – Document Object Model

DOM представление объектов

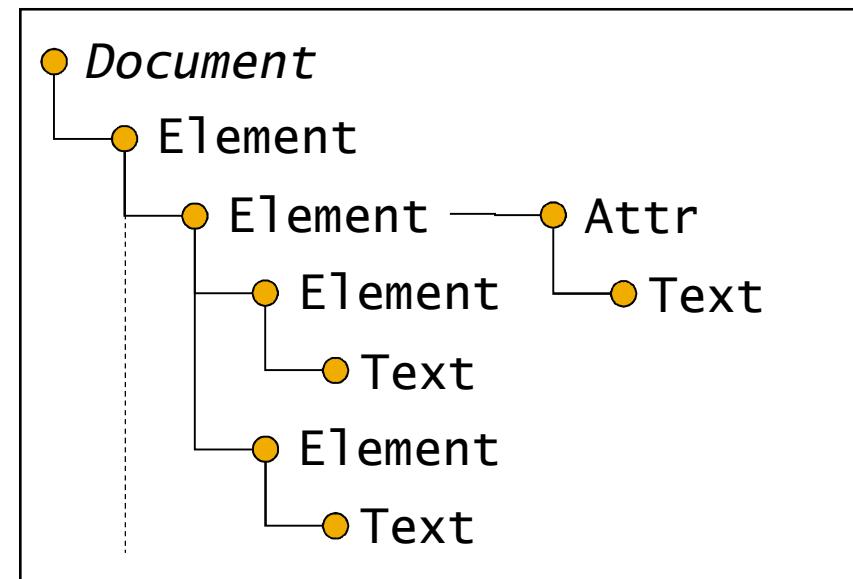
- Любой элемент документа рассматривается парсером как объект в памяти

```
<?xml version="1.0"?>
<?xmlstylesheet href="..." type="text/xsl"?>
<booklist>
  <book>
    <title>Sushi, Anyone?</title>
    <isbn>72-80081-024</isbn>
    <publisher>New Moon Books</publisher>
    <price>14.99</price>
    <details>...</details>
    <author>Charlotte M. Cooper</author>
    <author>Yoshi Nagase</author>
  </book>
  ...
</booklist>
```



DOM структура

- Парсер представляет документ как иреархию объектов
- Объекты DOM – это узлы (node) связанные друг с другом
- Объект Document – основной объект документа
- Другие объекты представляют элементы, текст, атрибуты, комментарии и т.д.



Необходимость проверки грамматики документа

- Документ может быть не предназначен для вашей системы
- Документ может содержать неправильные данные
- Документ может содержать ошибки в структуре

DTD - Document Type Definition

- Язык описания структуры документа
- Описывает:
 - Какие элементы могут присутствовать в документе
 - Повторения элементов
 - Какие атрибуты могут быть у элементов
 - Какие атрибуты обязательны
 - Какие сущности могут применяться

Пример DTD

```
<!ELEMENT pricelist (book+)>
<!ELEMENT book (title, author+, price,
description?)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT price (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ATTLIST price currency CDATA #IMPLIED>
```

Включения DTD

```
<?xml version=1.0" encoding="UTF-8"?>
<!DOCTYPE document [
    <!ELEMENT document (greeting, message)>
    <!ELEMENT greeting (#PCDATA)>
    <!ELEMENT message (#PCDATA)>
]>

<document>
    <greeting> Hello From XML </greeting>
    <message>
        Welcome to the wild
        and woolly world of XML.
    </message>
</document>
```

Внешние определения DTD

```
<?xml version = "1.0" standalone="no"?>

<!DOCTYPE document SYSTEM "mydoc.dtd">

<document>
    <customer>
        ...
    </customer>
</document>
```

Объявление элемента (Списки дочерних элементов)

- <!ELEMENT pricelist (book+)>
- <!ELEMENT book (title, author+, price?)>
- <!ELEMENT title (#PCDATA)>
- <!ELEMENT author (#PCDATA)>
- <!ELEMENT price (#PCDATA)>

- Модификаторы
 - * - ноль или много
 - ? - ноль или один
 - + - один или много

Альтернативы элементов и пустые элементы

- `<!ELEMENT item
(product, number,
(price | chargeacct | sample))>`
- `<!ELEMENT warning EMPTY>`

Объявление атрибута

- <!ATTLIST price
 currency CDATA #REQUIRED
 symbol CDATA #IMPLIED>
- <price
 currency="USD"
 symbol="\$">12.3</price>

Объявления сущностей

- <!ENTITY today "October 15, 2007">
- <!ENTITY company "Roga and Copyta">
- <!ENTITY nbsp "October 15, 2003">

- <?xml version = "1.0"?>
 <TEXT>
 This text about &company;
 is the content.
 </TEXT>

Лабораторная работа 2

Использование DTD в XML-документе

Упражнение 1: XML-документа

- Откройте файл **lab-1-2.xml** с помощью Notepad++
- Пересохраните его с именем **lab-2.xml**
- Удалите из XML документа описания пространств имен
- Сохраните файл и проверьте его правильность

Упражнение 2: Создание DTD описания

- Напишите DTD описание этого файла и проверьте файл на валидность:
[Дополнения -> XML Tools -> Validate now]
- Сохраните результаты своей работы

Выводы

- XML представляется в виде дерева объектов
- Существуют несколько способов программной обработки (SAX, DOM)
- DTD описывает структуру документа
- DTD может применяться для проверки документа на правильность

Описание структуры документа – XML схемы

Игорь Борисов
<http://igor-borisov.ru>

Темы модуля

- Недостатки и проблемы DTD
- XML схемы – как унифицированный способ описания структуры
- Основные элементы XML схемы
- Описания типов, элементов и атрибутов
- Описание сложных типов данных
- Средства создания и работы со схемами
- Проверка документов по XML схеме

Недостатки и проблемы DTD

- Отличный от XML синтаксис языка
- Нельзя поставить документу в соответствие два и более DTD описаний (актуально для составных документов)
- Нет проверки *типов* данных

XML схемы

- Унифицированный способ описания структуры
- <http://www.w3.org/XML/Schema>
- Промышленный стандарт описания XML документа
- Описывает:
 - **словарь** (названия элементов и атрибутов)
 - **модель содержания** (отношения между элементами и атрибутами и их структура)
 - **типы данных**

Пример простой схемы

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="страна" type="Страна"/>
  <xs:complexType name="Страна">
    <xs:sequence>
      <xs:element name="название" type="xs:string"/>
      <xs:element name="население" type="xs:decimal"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```
<страна>
  <название>Франция</название>
  <население>59.7</население>
</страна>
```

Соответствие схемы и документа

■ XML-схема

- <?xml version="1.0"?>
 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 targetNamespace="http://www.camera.org"
 xmlns="http://www.camera.org"
 xmlns:nikon="http://www.nikon.com"
 xmlns:olympus="http://www.olympus.com"
 xmlns:pentax="http://www.pentax.com"
 elementFormDefault="unqualified">

■ Документ

- <?xml version="1.0"?>
 <my:camera xmlns:my="http://www.camera.org"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.camera.org/Camera.xsd">

Привязка схемы к документу без указанного пространства имен

```
<?xml version="1.0" encoding="utf-8"?>
<root
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="myschema.xsd">

    <name>Вася Пупкин</name>
    <myAge>24</myAge>

</root>
```

Структура XML-схемы

```
<xsd:schema  
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">  
  
    <!-- Элементы схемы -->  
    <xsd:annotation>  
        <xsd:documentation>  
            Моя схема  
        </xsd:documentation>  
    </xsd:annotation>  
  
    ...  
  
</xsd:schema>
```

Описание простых элементов

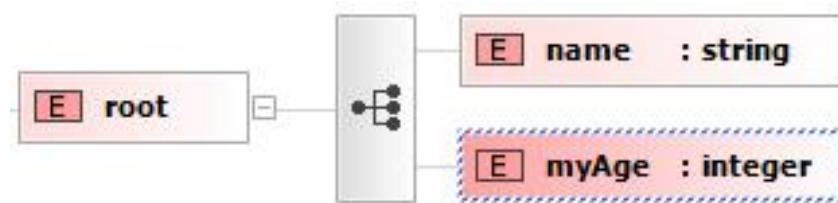
```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
    xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="name" type="xs:string" />

</xs:schema>
```

Описание сложных элементов

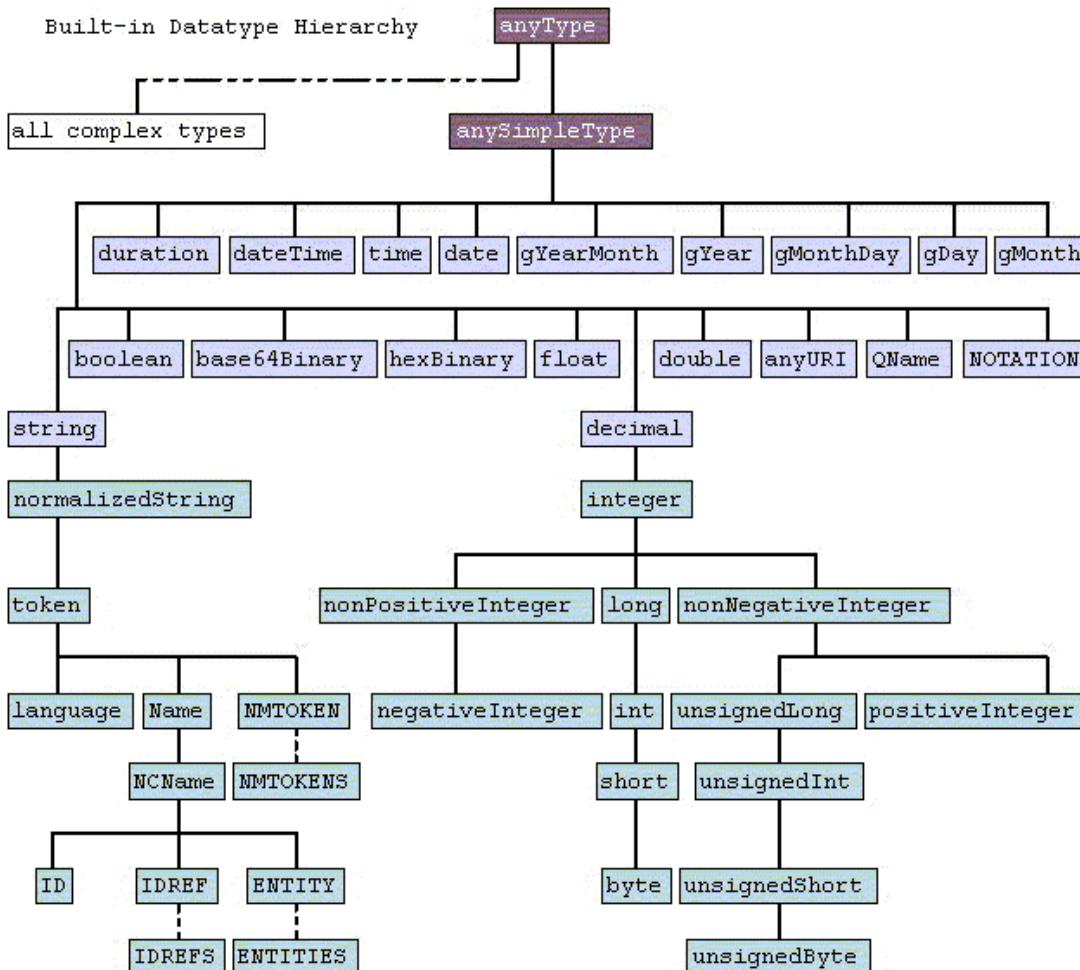
```
<?xml version="1.0" encoding="utf-8" ?>
<xss:schema
    xmlns:xss="http://www.w3.org/2001/XMLSchema">
<xss:element name="root">
    <xss:complexType>
        <xss:sequence>
            <xss:element name="name" type="xss:string"/>
            <xss:element name="myAge" type="xss:integer"/>
        </xss:sequence>
    </xss:complexType>
</xss:element>
</xss:schema>
```



Типы данных делятся на:

- simpleType
- complexType
 - Поименованный
 - Безымянный

Типы данных, иерархия базовых типов



<http://www.w3.org/TR/xmlschema-2/>

Поименованные типы

```
<xsd:complexType name="PersonType">
  <xsd:sequence>
    <xsd:element name="name" type="xsd:string"/>
    <xsd:element name="address" type="xsd:string"/>
    <xsd:element name="note" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="person" type="PersonType' />
```

Контейнеры комплексных типов

- Контейнеры-объединители
(композиторы)
 - sequence
 - choice
 - all

Лабораторная работа 3.1

Использование простой XML-схемы в XML-документе

Упражнение 1: Создание простой XML-схемы

- Откройте папку с лабораторными работами
- Создайте текстовый файл с именем **lab-3-1.xsd**
- Откройте этот файл с помощью Notepad++
- Установите кодировку **UTF-8**
- Напишите простую XML-схему для документа, содержащего данные:
 - **Имя**
 - **Фамилия**
 - **Возраст**
 - **Дата рождения**
- Сохраните файл **lab-3-1.xsd**
- Проверьте синтаксис схемы на правильность

Упражнение 2: Создание XML-документа по XML-схеме

- Создайте документ **lab-3-1.xml**, соответствующий созданной XML-схеме
- Проверьте XML-документ на правильность
- Подключите созданную схему к Вашему XML-файлу и проверьте его на соответствие схеме
- Сохраните результат Вашей работы

Количество вхождений элементов

```
<xsl:persons>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="person" type="PersonType"
                    minOccurs="0"
                    maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsl:persons>
```

Определения атрибутов

```
<xsd:complexType name="address">
  <xsd:sequence>
    <xsd:element name="name" type="xsd:string"/>
    <xsd:element name="street"
                  type="xsd:string" />
    <xsd:element name="city" type="xsd:string" />
  </xsd:sequence>
  <xsd:attribute name="phone"
                 type="xsd:string"
                 use="optional"/>
</xsd:complexType>
```

Атрибуты простых типов

```
<xsd:element name="phone">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="mobile"
                      type="xsd:string"
                      use="optional"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
```

Группы элементов

```
<xs:group name="acct">
  <xs:sequence>
    <xs:element name="descript" type="xs:string"/>
    <xs:element name="number" type="xs:string"/>
    <xs:element name="balance" type="xs:decimal"/>
  </xs:sequence>
</xs:group>
```

```
<xs:complexType name="checkingAcct">
  <xs:sequence>
    <xs:group ref="acct" />
  </xs:sequence>
  <xs:attribute name="status" type="xs:string"/>
</xs:complexType>
```

Расширения типов

```
<xsd:complexType name="acct">
  <xsd:sequence>
    <xsd:element name="description" type="xsd:string"/>
    <xsd:element name="number" type="xsd:string"/>
    <xsd:element name="type" type="acctTypeCode"/>
    <xsd:element name="balance" type="xsd:decimal"/>
  </xsd:sequence>
  <xsd:attribute name="status" type="xsd:string"/>
</xsd:complexType>

<xsd:complexType name="savingsAcct">
  <xsd:complexContent>
    <xsd:extension base="acct">
      <xsd:sequence>
        <xsd:element name="minimumBalance"
                     type="xsd:decimal" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

Расширения простых типов

```
<xsd:simpleType name="loginType">
```

```
  <xsd:restriction base="xsd:string">
```

```
    <xsd:minLength value="3" />
```

```
    <xsd:maxLength value="12" />
```

```
  </xsd:restriction>
```

```
</xsd:simpleType>
```

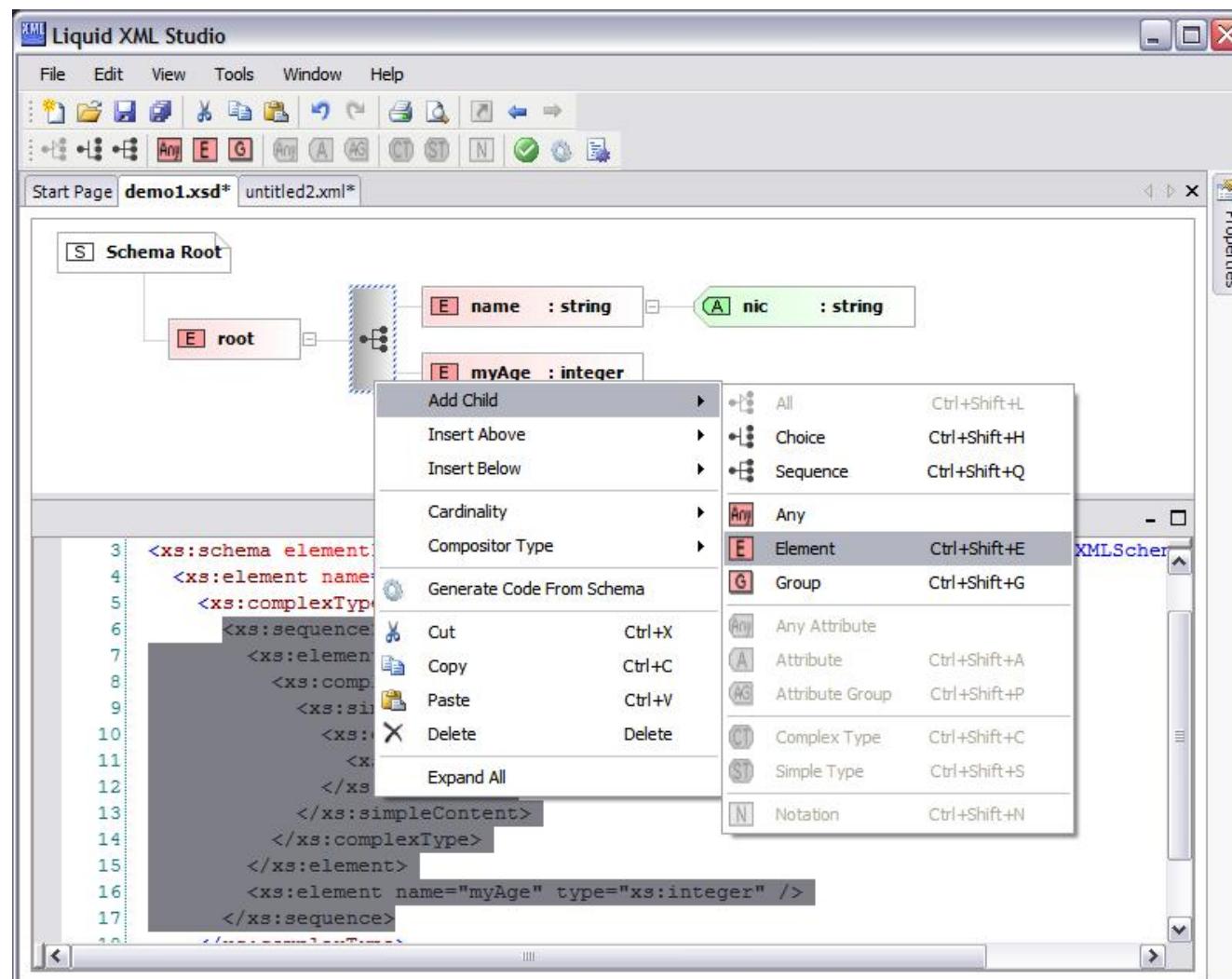
Лабораторная работа 3.2

Модификация XML-схемы своего XML-документа

Упражнение 1: Создание XML-схемы

- Откройте папку с лабораторными работами
- Откройте файл **lab-2.xml** и сохраните его с именем **lab-3-2.xml**
- Создайте текстовый файл с именем **lab-3-2.xsd**
- Создайте XML-схему для документа с Вашим расписанием занятий
- Подключите созданную схему к документу **lab-3-2.xml** и проверьте его на валидность
- Сохраните результат Вашей работы

Средства создания и работы со схемами



Выводы

- DTD описания имеют некоторые серьезные ограничения
- XML схемы могут описывать сложные типы данных и структуру документов
- XML схемы являются промышленным способом описания и проверки XML документов
- Существуют специализированные графические редакторы XML схем

Введение в XSLT и XPath

Игорь Борисов
<http://igor-borisov.ru>

Темы модуля

- XSLT как средство преобразований XML
- Создание XSL документа
- Программные средства преобразований XSLT
- Шаблоны и шаблонные правила
- Язык XPath
- Оси выборки
- Предикаты
- Типы данных и функции XPath

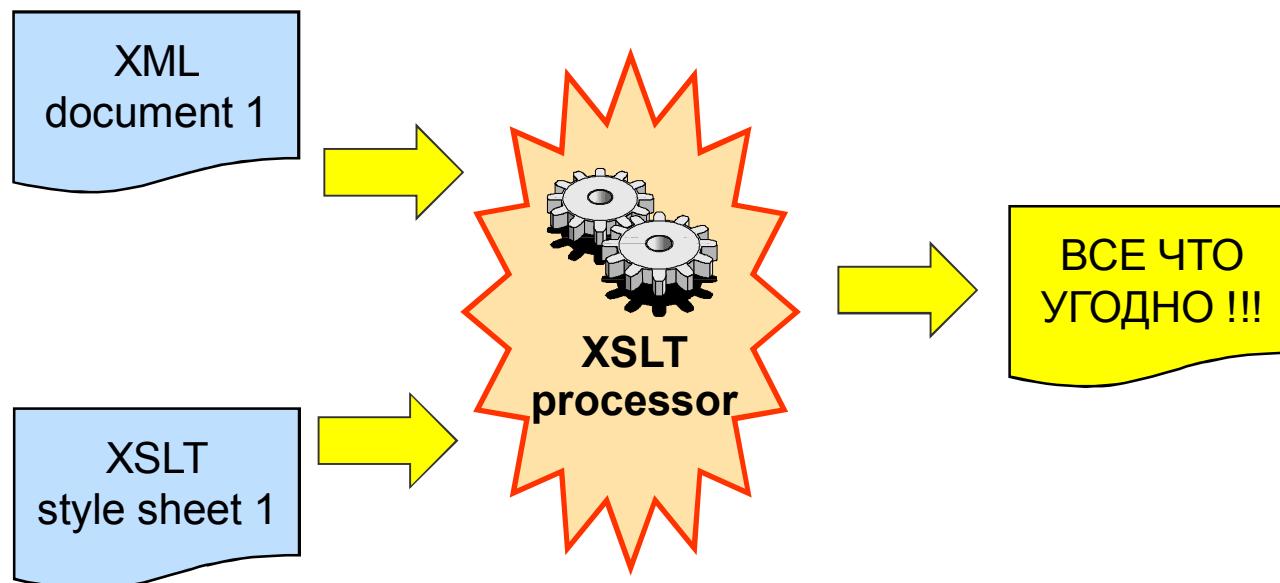
Проблемы представления XML

С этим XML-файлом не связана ни одна таблица стилей. Ниже показано дерево элементов.

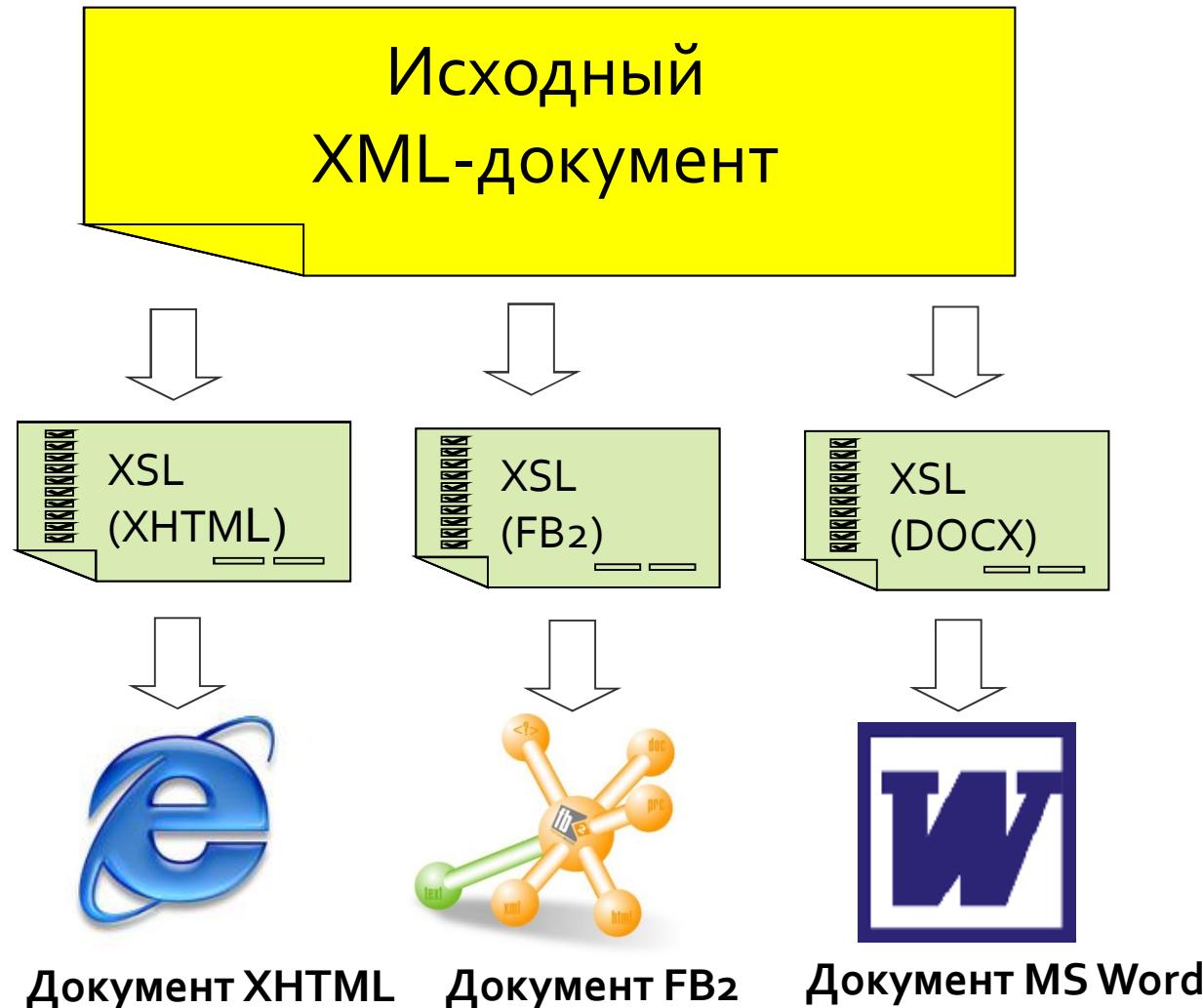
```
- <pricelist>
  - <book id="1">
    <title>XML и IE5</title>
    <author>Алекс Гомер</author>
    <price>200</price>
  </book>
  <!-- Комментарий -->
  - <book>
    <title>Dimanic HTML</title>
    <author>Алекс Хоумер</author>
    <price>120</price>
  </book>
  - <book id="3">
    <title>Технология XSLT</title>
    <author>Алексей Валиков</author>
    <price>150</price>
  </book>
```

XSLT как средство преобразований XML

- eXtensible Stylesheet Language
- XSL Transformation

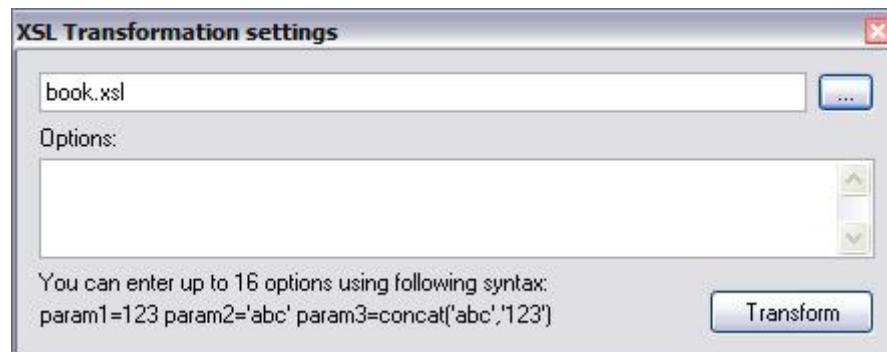


XSLT в действии



Средства преобразований XSL

- В браузере
 - <?xml-stylesheet type="text/xsl" href="style.xsl"?>
- В командной строке
 - c:\> msxsl books.xml books.xsl -o result.txt
- Программные средства



Создание XSL-документа

- XSL - XML-документ
- Все элементы принадлежат пространству имен:
<http://www.w3.org/1999/XSL/Transform>
- В XSL-документе могут использоваться любые пространства имен

```
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    ...
</xsl:stylesheet>
```

Шаблоны и шаблонные правила

- Шаблон (шаблонное правило) – правило обработки части XML документа

```
<xsl:template match="Xpath_выражение">  
    тело шаблона  
</xsl:template>
```

- Вывод значения

```
<xsl:template match="Xpath_выражение">  
    <xsl:value-of select="Xpath_выражение">  
</xsl:template>
```

Лабораторная работа 4.1

Использование простого XSL-преобразования

Упражнение 1: Создание XSL-файла

- Откройте папку с лабораторными работами
- Создайте текстовый файл с именем **lab-4-1.xsl**
- Откройте этот файл с помощью **Notepad++** и установите кодировку **UTF-8**
- Впишите в файл следующий текст и используйте его в качестве шаблона для других файлов XSL:

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

```
<!-- Шаблон корневого элемента -->
<xsl:template match="/">
</xsl:template>
```

```
</xsl:stylesheet>
```

- Напишите простую таблицу преобразования, выводящую фразу «Привет, МИР!»

Упражнение 2: XSL-преобразование XML-документа

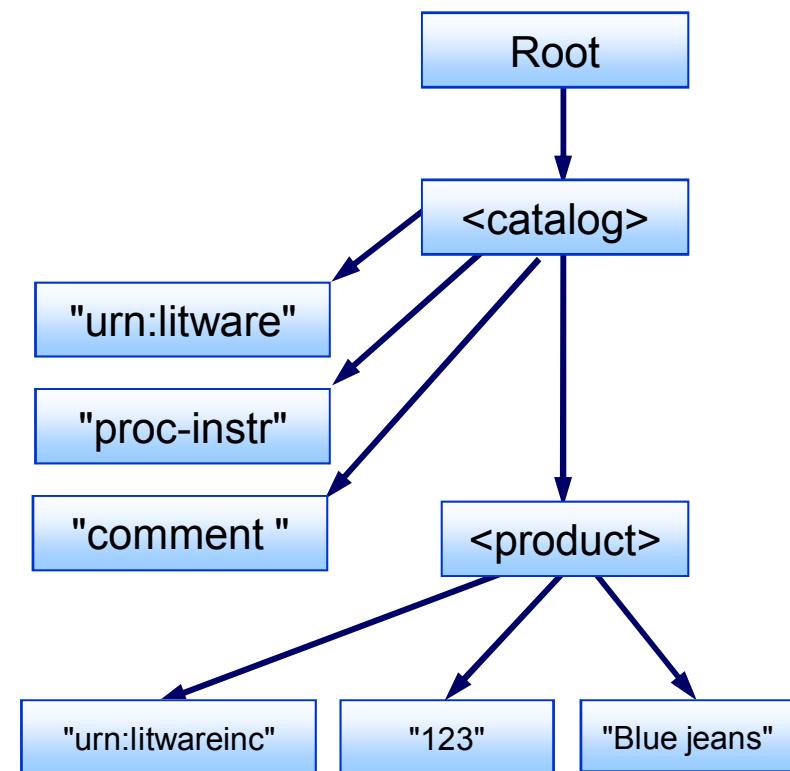
- Откройте файл **lab-1.xml** и сохраните его с именем **lab-4-1.xml**
- Подключите к нему таблицу **lab-4-1.xsl** и убедитесь в работоспособности преобразования:
 - Для этого можно открыть файл **lab-4-1.xml** с помощью браузера...
 - Или в редакторе Notepad++ выбрать следующую возможность:
[Дополнения -> XML Tools -> XSL Transformation]
- Сохраните результат Вашей работы

Введение в XPath выражения

- XPath выражения похожи на пути к файлам
 - <xsl:template match="job/title">
- / - абсолютный путь
 - <xsl:template match="/employee/job/title">
- // - поиск элементов на любой глубине
 - <xsl:template match="/employee//name">
- * - любой элемент
 - <xsl:template match="*>

XPath дерево документа и типы узлов

```
<catalog xmlns="urn:litwareinc">  
  
<?proc instr?>  
  
<!--comment-->  
  
<product code="123">  
    Blue jeans  
</product>  
  
</catalog>
```

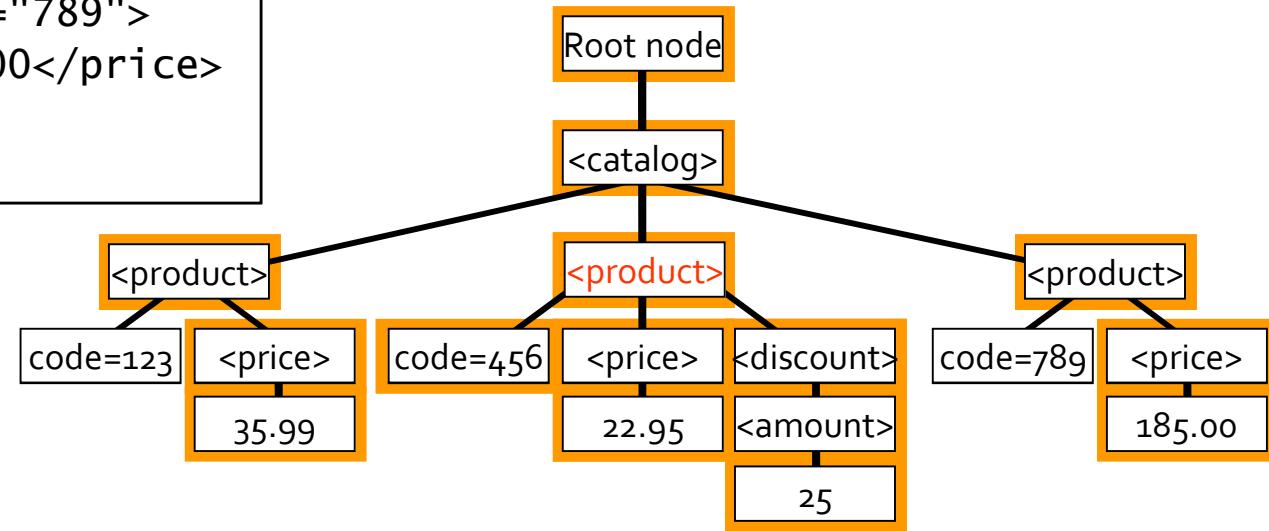


Оси выборки (axes)

```
<catalog>
  <product code="123">
    <price>35.99</price>
  </product>
  <product code="456">
    <price>22.95</price>
    <discount>
      <amount>25</amount>
    </discount>
  </product>
  <product code="789">
    <price>185.00</price>
  </product>
</catalog>
```

self
child
parent
attribute
descendant
descendant-or-self
ancestor
ancestor-or-self

following
following-sibling
preceding
preceding-sibling



Использование осей в XPath

- Многие выражения имеют сокращенную запись

Ось	Полная запись	Сокращенная запись
child	child::price	price
attribute	attribute::code	@code
self	self::node()	.
parent	parent::node()	..
descendant-or-self	/descendant-or-self()/price	//price

- Некоторые оси не имеют сокращенной записи
 - `ancestor::node()`

Выборки по осям

■ Выборка узла по имени

- Имя узла
- Все узлы
- Имя и пространство
- Все узлы в пространстве

Child examples

price

Attribute examples

@code

*

@*

aprefix:price

@aprefix:code

aprefix:*

@aprefix:*

■ Выборка узлов по типу

- `axis::node-type-function()`

`text()`

`processing-instruction()`

`node()`

`comment()`

Предикаты

- Выражение может иметь предикаты
 - `axis::node-test[predicate]...`
- Фильтрация
 - Положение узла: `product[last()]`
 - Содержание узла: `product[@code="123"]`
 - Присутствие узла: `product[@code]`
- Несколько предикатов
 - `product[3][@discount]`

Операции в XPath

- Логические операции
 - A **and** B, A **or** B
- Арифметические операции
 - A **+** B, A **-** B, A ***** B, A **div** B,
A **mod** B, - A
- Операции сравнения
 - A **=** B, A **!=** B, A **>** B, A **<** B,
A **>=** B, A **<=** B

Типы данных и функции XPath

- Булевские функции
- Числовые функции
- Строковые функции
- Функции множеств узлов

Булевские функции

- `boolean(object)`
- `not(boolean)`
- `true()`
- `false()`

Числовые функции

- `number(object)`
- `sum(node-set)`
- `floor(number)`
- `ceiling(number)`
- `round(number)`

Строковые функции

- `string(object)`
- `concat(string, string, string*)`
- `starts-with(string, string)`
- `contains(string, string)`
- `substring-before(string, string)`
- `substring-after(string, string)`
- `substring(string, number, number?)`
- `string-length(string?)`
- `normalize-space(string?)`
- `translate(string, string, string)`

ФУНКЦИИ МНОЖЕСТВ УЗЛОВ

- last()
- position()
- count(node-set)
- local-name(node-set?)
- namespace-uri(node-set?)
- name(node-set?)

Лабораторная работа 4.2

XPath и XSLT

Упражнение 1: Использование XPath для преобразования

- Откройте папку с лабораторными работами
- Откройте файл **lab-1.xml** и сохраните его с именем **lab-4-2.xml**
- Откройте файл **lab-4-1.xsl** и сохраните его с именем **lab-4-2.xsl**
- Подключите к файлу **lab-4-2.xml** таблицу **lab-4-2.xsl** и убедитесь в работоспособности преобразования
- Напишите в корневом шаблоне файла **lab-4-2.xsl** оператор, выводящий на экран общее число занятий в документе **lab-4-2.xml**
- Напишите в корневом шаблоне файла **lab-4-2.xsl** оператор, выводящий на экран первое занятие в документе **lab-4-2.xml**
- Напишите в корневом шаблоне файла **lab-4-2.xsl** оператор, выводящий на экран время начала второго занятия в документе **lab-4-2.xml**
- Проверьте свою работу
- Сохраните результат Вашей работы

Выводы

- XSLT – мощное средство преобразования XML данных практически к любому виду
- XSL файл – это набор шаблонных правил

- XPath – язык выборки узлов в дереве
- Оси выборки XPath – это адресация к другим элементам из текущего контекста
- XPath обладает рядом функций

Основные элементы XSLT

Игорь Борисов
<http://igor-borisov.ru>

Темы модуля

- Использование шаблонов
- Создание узлов-элементов
- Создание узлов-атрибутов
- Создание текстовых узлов
- Создание комментариев и инструкций обработки
- Копирование узлов
- Управляющие конструкции
- Сортировка значений

Использование шаблонов

```
<xsl:template  
    name="name"  
    match="pattern"  
    mode="mode"  
    priority="number">  
    <!--  
        Content: (<xsl:param>*, template)  
    -->  
  
</xsl:template>
```

Выборка узлов и применение шаблонов

```
<xsl:apply-templates  
    select="expression"  
    mode="name">  
  
<!--  
    Content: (xsl:sort|xsl:with-param)*  
-->  
  
</xsl:apply-templates>
```

Именованные шаблоны

```
<xsl:template name="tmpName"
  <!--
    Content:(<xsl:param>*,template)
  -->
```

```
</xsl:template>
```

```
<xsl:call-template name="tmpName">
  <!-- Content:xsl:with-param* -->
</xsl:call-template>
```

Создание узлов-элементов

```
<xsl:element  
    name="name"  
    namespace="URI"  
    use-attribute-sets="namelist">  
  
    <!-- Content:template -->  
  
</xsl:element>
```

Создание узлов-атрибутов

```
<xsl:attribute  
    name="name"  
    namespace="URI">
```

```
<!-- Content:template -->
```

```
</xsl:attribute>
```

Создание наборов атрибутов

```
<xsl:attribute-set  
    name="name">  
  
    <!-- Content:xsl:attribute* -->  
  
</xsl:attribute-set>
```

Создание текстовых узлов

```
<xsl:text  
    disable-output-escaping="yes | no">  
  
    <!-- Content:#PCDATA -->  
  
</xsl:text>
```

Создание комментариев и инструкций обработки

```
<xsl:comment>
    <!-- Content:template -->
</xsl:comment>
```

```
<xsl:processing-instruction
    name="process-name">
    <!-- Content:template -->
</xsl:processing-instruction>
```

Лабораторная работа 5.1

Использование операторов XSLT создания узлов

Упражнение 1: Создание XML-документа

- Откройте папку с лабораторными работами
- Создайте пустой текстовый файл с именем **lab-5-1.xml**
- Откройте файл **lab-5-1.xml** с помощью Notepad++
- Установите кодировку документа **UTF-8**
- Наберите следующий текст:

```
<?xml version="1.0" encoding="UTF-8"?>
<lab5_1>
    <element name="computer" value="Intel Core Duo" />
    <element name="monitor" value="19 in." />
    <element name="printer" value="LBP2900" />
</lab5_1>
```
- Проверьте XML на правильность синтаксиса

Упражнение 2: Создание XSL-документа

- Создайте пустой текстовый файл с именем **lab-5-1.xsl**
- Откройте файл **lab-5-1.xsl** с помощью Notepad++
- Установите кодировку документа **UTF-8**
- Наберите следующий текст:

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:output
        method="xml"
        encoding="UTF-8" />

    <xsl:template match="/">

    </xsl:template>
</xsl:stylesheet>
```
- Проверьте XML на правильность синтаксиса

Упражнение 3: Преобразование с использованием операторов создания узлов

- Допишите XSL-файл так, чтобы получить из XML-документа **lab-5-1.xml** XML-документ следующего вида:

```
<?xml version="1.0" encoding="UTF-8" ?>
<labResult>
    <!-- Всего 3 элемента -->
    <computer>Intel Core Duo</computer>
    <monitor>19 in.</monitor>
    <printer>LBP2900</printer>
</labResult>
```

Копирование узлов

```
<xsl:copy-of select="expression"/>
```

```
<xsl:copy  
use-attribute-sets="name-list">  
    <!-- Content:template -->  
</xsl:copy>
```

Управляющие конструкции

- Условная обработка
- Оператор выбора
- Циклическая обработка
- Рекурсивные правила

Условная обработка

```
<xsl:if test="expression">  
    <!-- Content: template -->  
</xsl:if>
```

Оператор выбора

```
<xsl:choose>
  <xsl:when test="boolean-expression">
    <!-- Content: template -->
  </xsl:when>
  <xsl:when test="boolean-expression">
    <!-- Content: template -->
  </xsl:when>
  <xsl:otherwise>
    <!-- Content:template -->
  </xsl:otherwise>
</xsl:choose>
```

Циклическая обработка

```
<xsl:for-each select="expression">  
    <!-- Content:(xsl:sort*,template) -->  
</xsl:for-each>
```

Рекурсивные правила

```
<xsl:template match="myElement"
  ...
    <xsl:apply-templates
      select="myElement" />
  ...
</xsl:template>
```

Сортировка значений

```
<xsl:sort  
    select="expression"  
    lang="language-code"  
    data-type="text|number|qname"  
    order="ascending|descending"  
    case-order="upper-first|lower-first"/>
```

Лабораторная работа 5.2

XSLT преобразование с управляющими конструкциями

Упражнение 1: Создание XML-документа

- Откройте папку с лабораторными работами
- Создайте пустой текстовый файл с именем **lab-5-2.xml**
- Откройте файл **lab-5-2.xml** с помощью Notepad++
- Установите кодировку документа **UTF-8**
- Наберите следующий текст:

```
<?xml version="1.0" encoding="UTF-8"?>
<lab5_2>
    <element name="computer" value="Intel Core Duo" />
    <element name="monitor" value="19 in." />
    <element name="printer" value="LBP2900" />
    <element name="keyboard" value="104 keys" />
    <element name="mouse" value="optical" />
</lab5_2>
```
- Проверьте XML на правильность синтаксиса

Упражнение 2: Создание XSL-документа

- Создайте пустой текстовый файл с именем **lab-5-2.xsl**
- Откройте файл **lab-5-2.xsl** с помощью Notepad++
- Установите кодировку документа **UTF-8**
- Наберите следующий текст:

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:output
        method="html"
        encoding="UTF-8" />

    <xsl:template match="/">
        </xsl:template>
</xsl:stylesheet>
```
- Проверьте XML на правильность синтаксиса

Упражнение 3: Преобразование с использованием управляющих конструкций

- Допишите XSL-файл так, чтобы получить из XML-документа **lab-5-2.xml** HTML-документ со списком элементов, при этом каждый нечетный элемент должен выделяться серым фоном текста.

Выводы

- Создание узлов позволяет формировать выходной документ любой сложности
- Комментарии и инструкции по обработке полезны при выводе XML документов
- Управляющие конструкции позволяют гибко менять выходной документ
- Сортировка значений позволяет выводить данные в требуемом виде

Дополнительные возможности XSLT

Игорь Борисов
<http://igor-borisov.ru>

Темы модуля

- Псевдонимы и пространства имен
- Ключи и выборка узлов по ключу
- Управление выводом документа
- Формирование различных выходных файлов
- Использование нескольких входных документов
- Дополнительные функции XSLT

Использование пространств имен

- Перед использованием пространств имен их следует определить:

```
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:fb2="http://www.gribuser.ru/xml/fictionbook/2.0"
    xmlns:lab="http://www.mysite.ru/xml/labs">
```

- XPath

```
/fb2:FictionBook/fb2:description/
    fb2:title-info/fb2:author
```

Псевдонимы и пространства имен

```
<xsl:namespace-alias  
stylesheet-prefix="prefix|#default"  
result-prefix="prefix|[#default"/>
```

Ключи и выборка узлов по ключу

- Определение ключа

- `<xsl:key`

```
    name="keyName"  
    match="pattern"  
    use="expression"/>
```

- Выборка по ключу

- `key('keyName' , 'value')`

Пример выборки

- <pricelist>
 <book id="1">
 <title>XML</title>
 <author>Алекс Гомер</author>
 </book>
 ...
 </pricelist>
- <xsl:key name="ixBookTitle"
 match="/pricelist/book" use="title" />
- <h2>Книги XML</h2>
 <xsl:apply-templates
 select="key('ixBookTitle', 'XML')"/>

Составные ключи

- ```
<xsl:key name="ixBookTitleAuthor"
 match="/pricelist/book"
 use="concat(title, author)" />
```
- ```
<h2>Книги XML</h2>
<xsl:apply-templates
          select="key('ixBookTitleAuthor',
concat( 'XML', ' Алекс Хоумер'))" />
```

Лабораторная работа 6.1

Выборка узлов по ключу

Упражнение 1: Создание XML-документа

- Откройте папку с лабораторными работами
- Создайте пустой текстовый файл с именем **lab-6-1.xml**
- Откройте файл **lab-6-1.xml** с помощью Notepad++
- Установите кодировку документа **UTF-8**

- Наберите следующий текст:

```
<?xml version="1.0" encoding="UTF-8"?>
<lab6_1>
    <course id="XML">
        <title>XML и XSLT. Современные технологии обработки данных для Web и Office 2007</title>
        <keywords>
            <keyword>XML</keyword>
            <keyword>XML Schema</keyword>
            <keyword>XPath</keyword>
            <keyword>XSLT</keyword>
        </keywords>
        <teachers>
            <teacher>Борисов И.О.</teacher>
            <teacher>Никитин И.Г.</teacher>
        </teachers>
    </course>
    <course id="AJAX">
        <title>AJAX. Разработка web-приложений для Web 2.0</title>
        <keywords>
            <keyword>AJAX</keyword>
            <keyword>XML</keyword>
            <keyword>XML Schema</keyword>
            <keyword>JavaScript</keyword>
            <keyword>XSLT</keyword>
        </keywords>
        <teachers>
            <teacher>Борисов И.О.</teacher>
            <teacher>Никитин И.Г.</teacher>
        </teachers>
    </course>
    <course id="PHP">
        <title>Профессиональное программирование на PHP 5</title>
        <keywords>
            <keyword>PHP</keyword>
            <keyword>XML</keyword>
            <keyword>XSLT</keyword>
        </keywords>
        <teachers>
            <teacher>Борисов И.О.</teacher>
            <teacher>Шуйков С.Ю.</teacher>
        </teachers>
    </course>
</lab6_1>
```

- Сохраните файл и проверьте XML на правильность синтаксиса

Упражнение 2: Создание XSL-документа

- Создайте пустой текстовый файл с именем **lab-6-1.xsl**
- Откройте файл **lab-6-1.xsl** с помощью Notepad++
- Установите кодировку документа **UTF-8**

Упражнение 3: XSLT преобразование в HTML для вывода списка курсов

- Создайте **корневой шаблон** преобразования
- Создайте **шаблон** для одного курса
- Создайте **ключ** для выбора курсов по преподавателю
- Создайте **ключ** для выбора курсов по ключевому слову
- Выведите все курсы, которые читает преподаватель **Никитин И.Г.**
(Подсказка: Сначала используем нужный ключ, а потом от результата выборки прыгаем на два этажа выше и получаем курс)
- Выведите все курсы, в которых рассматривается тема **XML**
(Подсказка: Смотрите подсказку для предыдущего задания)
- Выведите все курсы, которые читает преподаватель **Шуйков С.Ю.**, и в которых есть тема **XSLT**. Вам захочется сделать ключ и по ключевым словам, и по преподавателям. Но такой ключ сделать нельзя. Что делать? Подумайте.
(Подсказка: Можно сначала сделать выборку по ключу, а уже результат этой выборки отфильтровать предикатом)
- Сохраните свою работу

Управление выводом документа

```
<xsl:output  
    method="xml|html|text|name"  
    version="string"  
    encoding="string"  
    omit-xml-declaration="yes|no"  
    standalone="yes|no"  
    doctype-public="string"  
    doctype-system="string"  
    cdata-section-elements="namelist"  
    indent="yes|no"  
    media-type="string"/>
```

Формирование различных выходных файлов

```
<xsl:output method="text" encoding="utf-8" />

<xsl:output method="html"
    doctype-public="-//W3C//DTD HTML 4.01//EN"
    doctype-system="http://www.w3.org/TR/html4/
                      strict.dtd"

<xsl:output method="xml"
    doctype-public="-//W3C//DTD XHTML 1.0
                      Transitional//EN"
    doctype-system="http://www.w3.org/1999/xhtml"
    indent="yes"/>
```

Лабораторная работа 6.2

XSLT преобразование в различные форматы

Упражнение 1: XSLT преобразование в HTML для вывода списка курсов

- Откройте с помощью Notepad++ файл **Lab-6-1.xml** и сохраните его с именем **Lab-6-2.xml**
- Создайте новый файл **Lab-6-2-html.xsl**
- Установите кодировку документа **UTF-8**
- Создайте **корневой шаблон** преобразования
- Создайте **шаблон** для одного курса
- Сформируйте вывод в формат **HTML 4**
- Проверьте свою работу

Упражнение 2: XSLT преобразование для вывода списка курсов в текстовый файл

- Пересохраните файл **Lab-6-2-html.xsl** с именем **Lab-6-2-txt.xsl**
- Выведите список курсов в текстовый файл, используя утилиту командной строки **msxsl.exe**
(Эту утилиту можно бесплатно загрузить по адресу <http://www.microsoft.com/Downloads/details.aspx?FamilyID=2fb55371-c94e-4373-b0e9-db4816552e41&displaylang=en> или просто набрать в поисковой системе Google запрос: "msxsl download")

Упражнение 3: XSLT преобразование в XML для вывода списка курсов

- Пересохраните файл **Lab-6-2-html.xsl** с именем **Lab-6-2-xml.xsl**
- Напишите XSLT преобразование для вывода списка курсов в XML документ следующего вида:

```
<lab62Result>
    <course id="XML" teacher="Борисов И.О." />
    ...
</lab62Result>
```
- Для тестирования используйте утилиту командной строки **msxsl.exe** или расширение Notepad++ **XML Tools**
- Если остается время — допишите вывод атрибута **keywords** в элемент **course**, так чтобы в нем оказался список ключевых слов разделенный запятыми.

Использование нескольких входных документов

- include()
- import()
- document()

<xsl:copy-of

```
select="document('my.html')/html/body/h1"/>
```

Дополнительные функции XSLT

- `current()`
 - `<xsl:template match="item">`
`<xsl:element`
`name="/root/element[@id=current()/@id]">`
`...`
- `generate-id()`
 - `<xsl:value-of`
`select="generate-id(/catalog/book)">`

Выводы

- Пространства имен должны быть явно указаны
- Ключи и выборка узлов по ключу – очень эффективное средство выборки данных
- Управление выводом документа позволяет формировать различные выходные данные
- Использование нескольких входных документов расширяет возможности преобразования

Практическая работа с XSLT

Игорь Борисов
<http://igor-borisov.ru>

Темы модуля

- Переменные и параметры
- Выборка уникальных значений
- Группировка Мюнха
- Выборка узлов с множественной группировкой
- XSL Formatting Objects (обзорно)

Переменные и параметры

```
<xsl:variable  
    name="name"  
    select="expression">  
    <!-- Content:template -->  
</xsl:variable>
```

Использование переменных

```
<xsl:variable name="header">
  <tr>
    <th>Element</th>
    <th>Description</th>
  </tr>
</xsl:variable>

<table>
  <xsl:copy-of select="$header" />
  <xsl:for-each select="root/item">
</table>
```

Параметры и шаблоны

```
<xsl:param  
    name="paramName"  
    select="expression">  
    <!-- Content:template -->  
</xsl:param>
```

```
<xsl:call-template name="show_title">  
    <xsl:with-param name="paramName" />  
</xsl:call-template>
```

Выборка уникальных значений

- Группировка – выборка уникальных элементов из набора
- Задача группировки найти множество элементов с одинаковым значением и взять первый элемент множества
- Первый элемент - это элемент не имеющий предыдущего:
 - preceding-sibling::item[...условие...] == null

Группировка Мюнха (Steve Muench)

- Ключи! С помощью ключей можно выбрать множество узлов по их свойствам
- Функция `generate-id(множество)` возвращает ID первого элемента множества

Пример группировки Мюнха

```
<xsl:key name="ixAuthor"  
match="/pricelist/book/author" use=". " />
```

```
<xsl:variable name="authors"  
select="/pricelist/book/  
author[generate-id(.) =  
generate-id(key('ixAuthor', .))]" />
```

Лабораторная работа 7

Выборка узлов с множественной группировкой

Упражнение 1: Создание XML-документа

- Откройте редактор Notepad++
- Создайте новый файл и сохраните его под именем **lab-7.xml**
- Установите кодировку документа **UTF-8**
- Наберите следующий текст:

```
<?xml version="1.0" encoding="utf-8" ?>
<lab7>
    <item city="Москва" org="000 Рога и Копыта" title="Кепка" value="120" />
    <item city="Москва" org="000 Рога и Копыта" title="Велосипед" value="5" />
    <item city="Москва" org="000 Рога и Копыта" title="Футблока" value="10" />

    <item city="Москва" org="000 Светлана" title="Кепка" value="120" />
    <item city="Москва" org="000 Светлана" title="Футблока" value="10" />

    <item city="Москва" org="000 Ариадна" title="Велосипед" value="120" />
    <item city="Москва" org="000 Ариадна" title="Футблока" value="10" />

    <item city="Петербург" org="000 Стрелка" title="Кепка" value="120" />
    <item city="Петербург" org="000 Стрелка" title="Велосипед" value="5" />
    <item city="Петербург" org="000 Стрелка" title="Футблока" value="10" />

    <item city="Петербург" org="000 Светлана" title="Кепка" value="120" />
    <item city="Петербург" org="000 Светлана" title="Футблока" value="10" />

    <item city="Петербург" org="000 Петр I" title="Велосипед" value="120" />
    <item city="Петербург" org="000 Петр I" title="Футблока" value="10" />
</lab7>
```
- Сохраните файл
- Проверьте XML на правильность

Упражнение 2: XSLT преобразование в HTML для вывода списка товаров

- Создайте новый файл и сохраните его под именем **lab-7.xsl**
- Установите кодировку документа **UTF-8**
- Создайте **корневой шаблон** преобразования
- Создайте **ключ** для выбора городов
- Создайте **составной ключ** для выбора компании в городе
- Выберите и сохраните в переменную (#1) все города (уникальные значения)
- Проходя циклом по набору уникальных городов выберите и сохраните в переменную (#2) все уникальные компании (организации в этом городе)

- Проходя циклом по компаниям в указанном городе (переменная #1) выберите все товары с сортировкой по названию и выведите их в браузер
- Посчитайте промежуточные итоги: общее количество товаров для каждой компании и общее количество товара для каждого города
- Сохраните и проверьте свою работу

XSL Formatting Objects (обзорно)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
    <fo:layout-master-set>
        <fo:simple-page-master master-name="A4">
            <!-- Page template goes here -->
        </fo:simple-page-master>
    </fo:layout-master-set>
    <fo:page-sequence master-reference="A4">
        <!-- Page content goes here -->
    </fo:page-sequence>
</fo:root>
```

Выводы

- Переменные XSLT очень удобны для выборки данных
- Переменные не могут изменять свое значение
- Параметры используются в именованных шаблонах
- Группировка Мюнха – быстрый способ сформировать группированные данные
- XSL FO – способ формирования печатных листов

Интеграция XML данных

Игорь Борисов
<http://igor-borisov.ru>

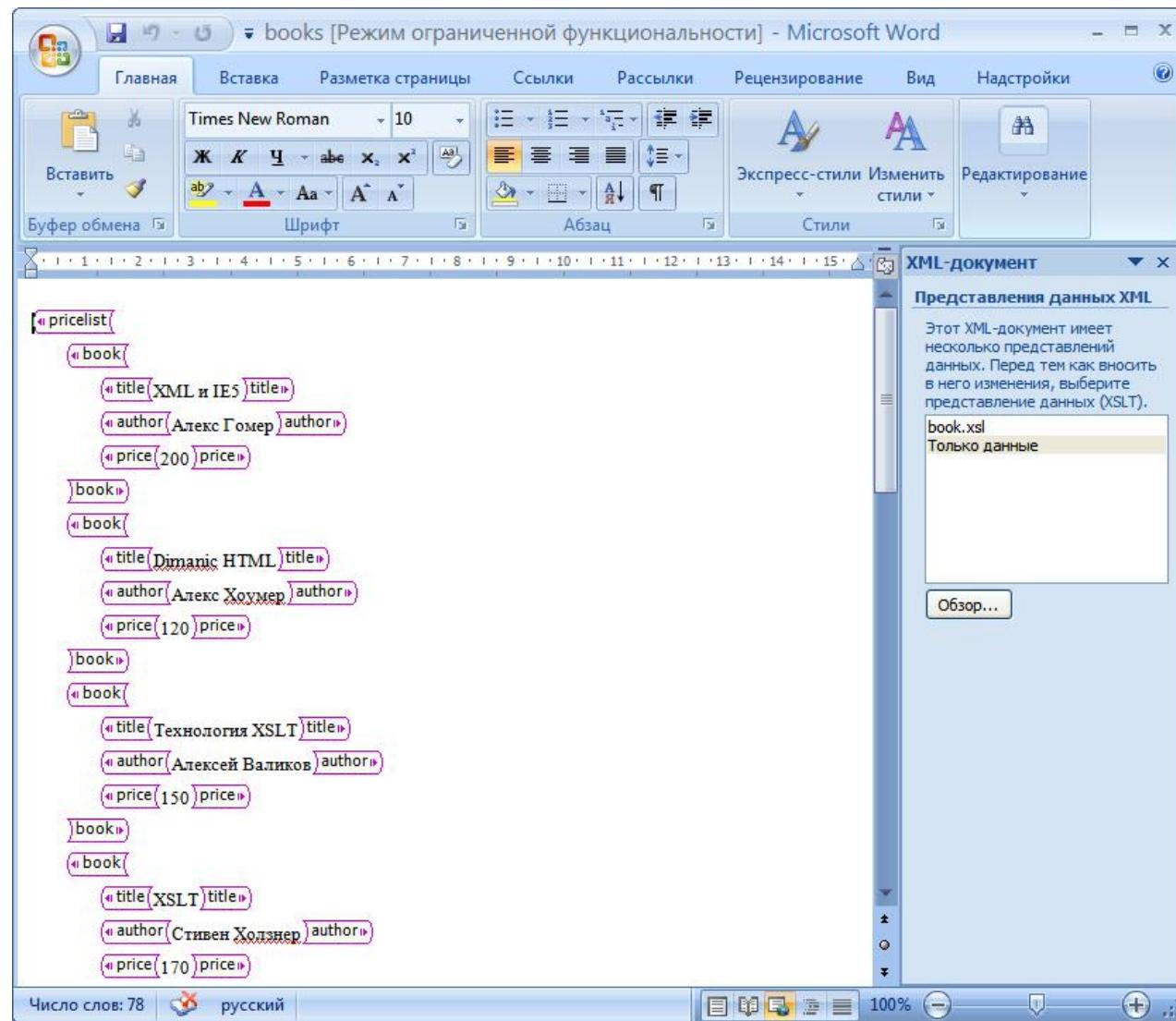
Темы модуля

- Использование XML
- Манипуляция XML данными в Microsoft Office 2010
- Интеграция данных с Microsoft Office Excel 2010
- Интеграция с Microsoft Office Access 2010
- Интеграция с Microsoft Office Word 2010
- Использование Microsoft Office InfoPath 2010

Использование XML

- XLink, XPointer, XQuery
 - <http://www.w3.org/Consortium/siteindex.html>
- RSS
 - <http://web.resource.org/rss/1.0/spec>
- FB2
 - <http://www.gribuser.ru/xml/fictionbook/index.html>
- WSDL, SOAP, XML-RPC
 - <http://www.w3.org/2002/ws/>
- MathML
 - <http://www.w3.org/Math/>
- SVG
 - <http://www.w3.org/Graphics/SVG/>

Интеграция с MS Word



Интеграция данных с MS Excel

Книга2 - Microsoft Excel

Главная Вставка Разметка страницы Формулы Данные Рецензирование Вид Надстройки Конструктор Работа с т...

Шрифт Выравнивание Число Ячейки Редактирование

Вставить Удалить Формат Ячейки Редактирование

A2 XML и IE5

	A	B	C	D
1	title	author	price	
2	XML и IE5	Алекс Гомер	200	
3	Dimanic HTML	Алекс Хоумер	120	
4	Технология XSLT	Алексей Валиков	150	
5	XSLT	Стивен Холзнер	170	
6	Практическая криптография	Михаил Масленников	200	
7	Peest Windows XP	Джерри Хонейкэтт	250	
8	Windows XP. Средства повышения производительности	Уолтер Брюс	250	
9	Windows Script Host для Windows 2000/XP	Андрей Попов	220	
10	Excel 2002. Библия пользователя	Джон Уокенбах	260	
11	XML. Справочник	Сандра Э. Эдди	100	
12	Разработка защищенных Web-приложений на платформе .NET	Майкл Ховард	250	
13	WAP-программирование на языке WML	Михаэль Зеебергер-Вай	150	
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				

Источник XML

Карты XML в этой книге:
pricelist_карта

pricelist
└── book
 ├── title
 ├── author
 └── price

Чтобы сопоставить повторяющиеся элементы, перетащите их из дерева на лист, где должны будут отображаться заголовки данных.

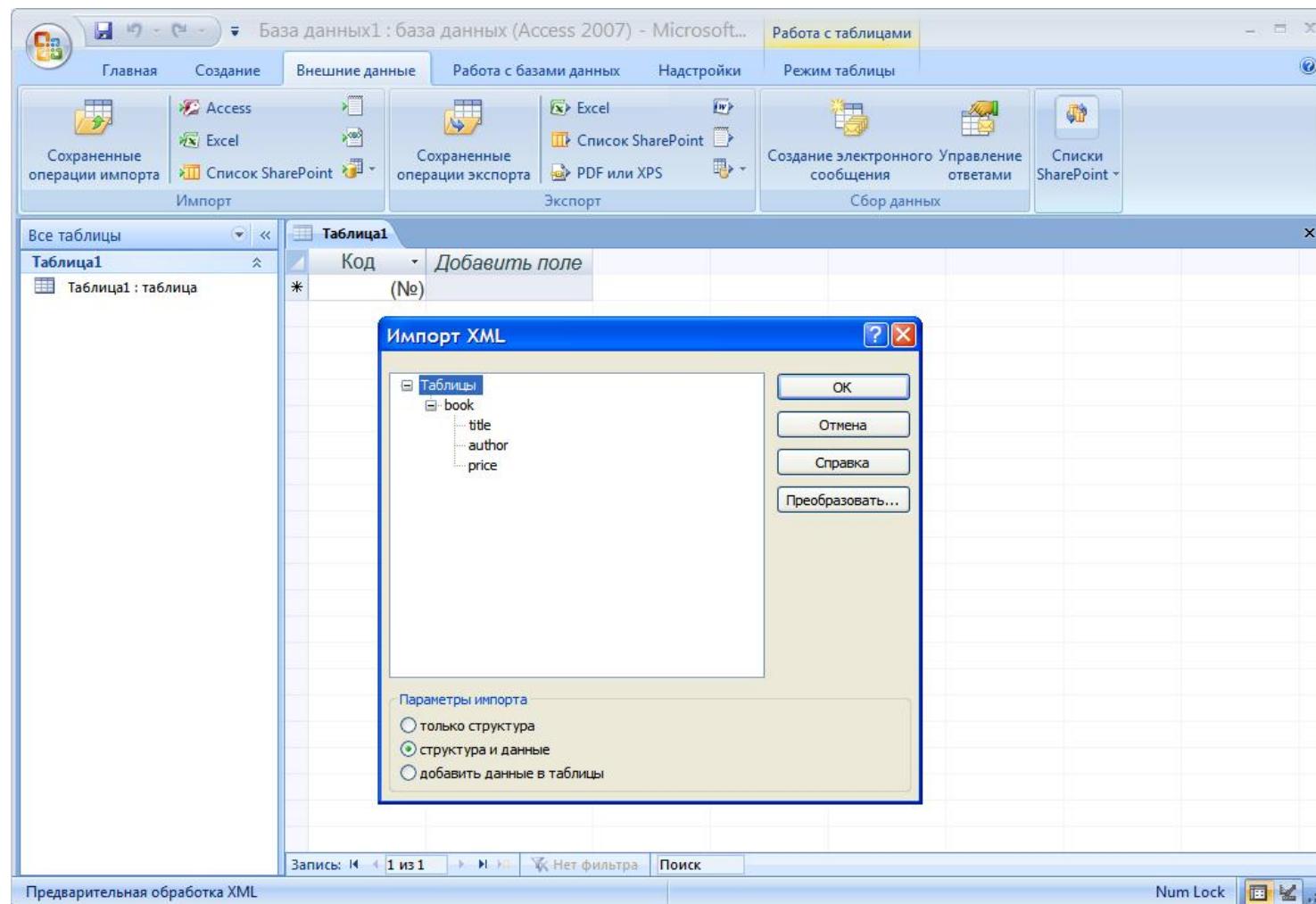
Чтобы импортировать данные XML, щелкните сопоставленную ячейку XML правой кнопкой мыши и выберите пункт "XML", а затем команду "Импорт".

Параметры

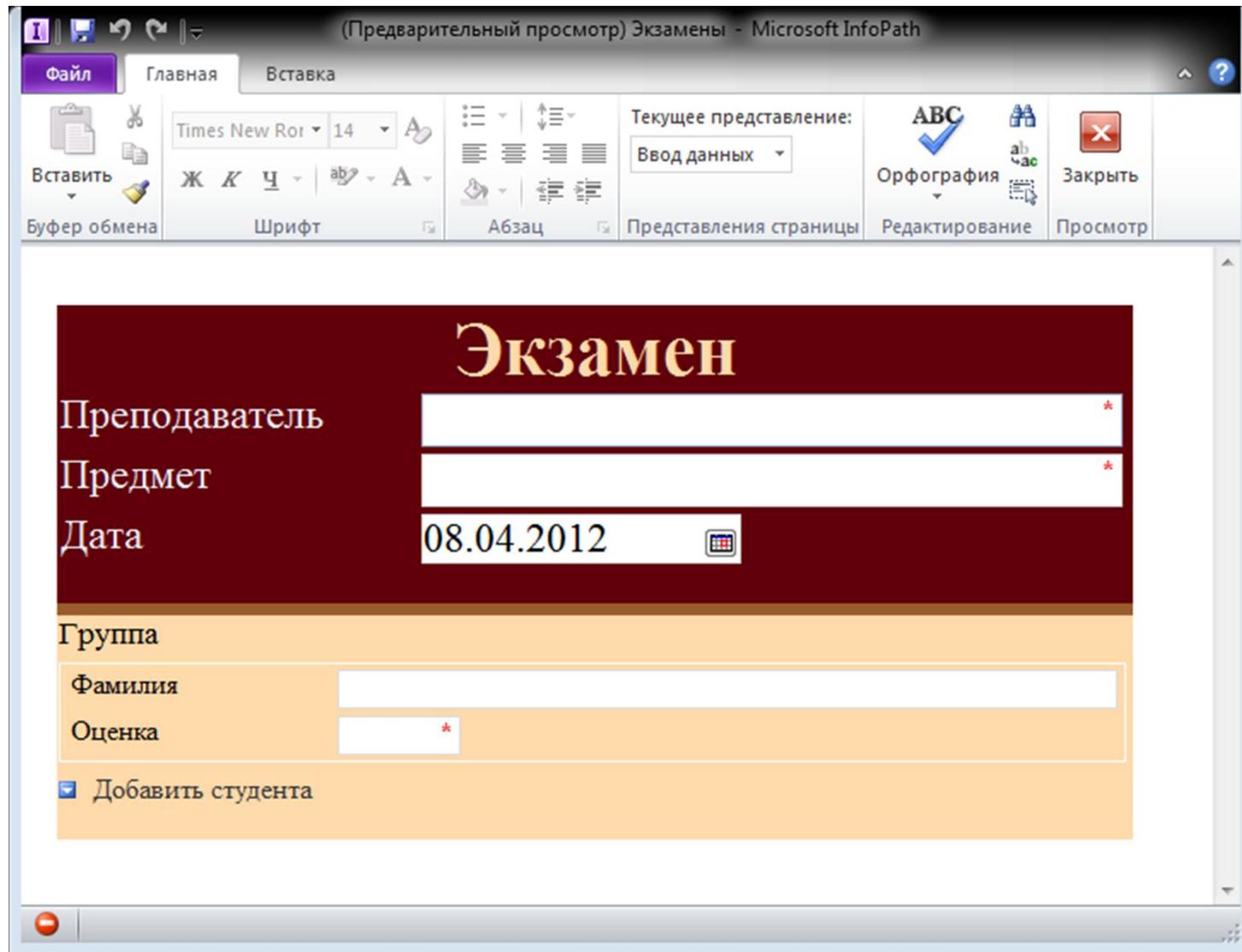
Лист1

Готово Количество: 12 100%

Интеграция с MS Access



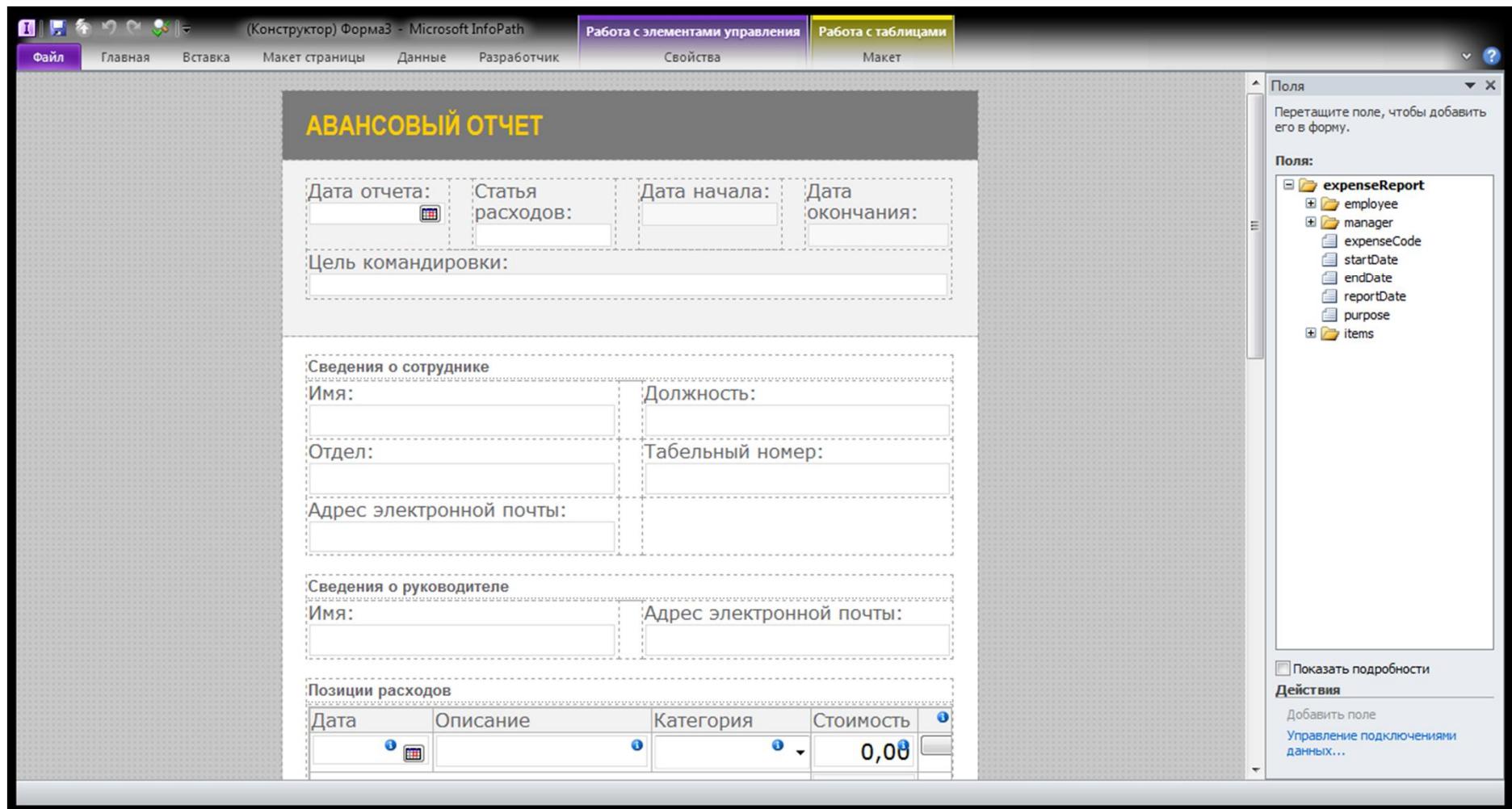
Интеграция с MS InfoPath



Формы Office InfoPath 2010

- Быстро создаются (практически без программирования)
- Могут повторно использоваться (формат данных – XML)
- Легко собираются в едином месте хранения (библиотеки форм InfoPath)
- Для работы может не требоваться само приложение InfoPath (!) – Forms Server
- Легко интегрируются в существующие бизнес-процессы

Создание шаблона формы



Готовый отчет

(Предварительный просмотр) Авансовый отчет - Microsoft InfoPath

Файл Главная Вставка ?

АВАНСОВЫЙ ОТЧЕТ

Дата отчета: Статья расходов:
Дата начала: Дата окончания:

Цель командировки:

Сведения о сотруднике

Имя: Должность:
Отдел: Табельный номер:
Адрес электронной почты:

Сведения о руководителе

Имя: Адрес электронной почты:

Позиции расходов

Дата	Описание	Категория	Стоимость
<input type="button" value="..."/>	<input type="text"/>	<input type="text"/>	0,00 <input type="button" value=">>"/>

Шаги создания шаблона формы

- Создание макета
- Определение элементов управления
- Определение схемы данных
- Определение представлений формы
- Проверка формы и публикация

Создание макета формы

(Конструктор) Форма2 - Microsoft InfoPath

Файл Главная Вставка Макет страницы Данные Разработчик

Times New Roma 12 Экспресс-стили
Ж К Ч а в Начертания
Буфер обмена Формат текста

Элементы управления
Добавить Управление правилами Правила

Редактирование Просмотр
Форма

Наши книги

<i>Id</i>	<i>Название</i>	<i>Автор</i>	<i>Цена</i>

Повторяющаяся таблица

Поля

Перетащите поле, чтобы добавить его в форму.

Поля:

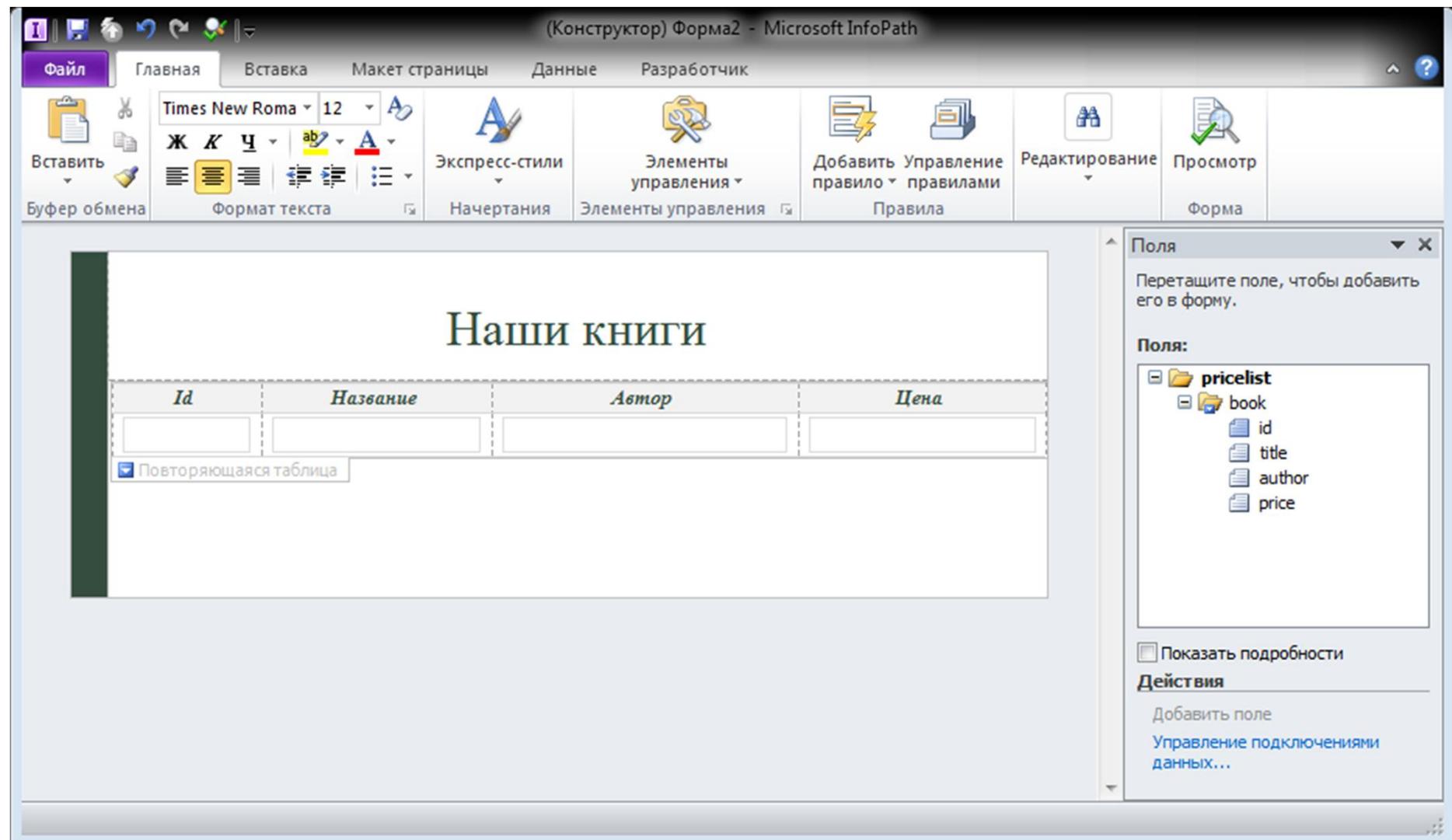
- pricelist
 - book
 - id*
 - title*
 - author*
 - price*

Показать подробности

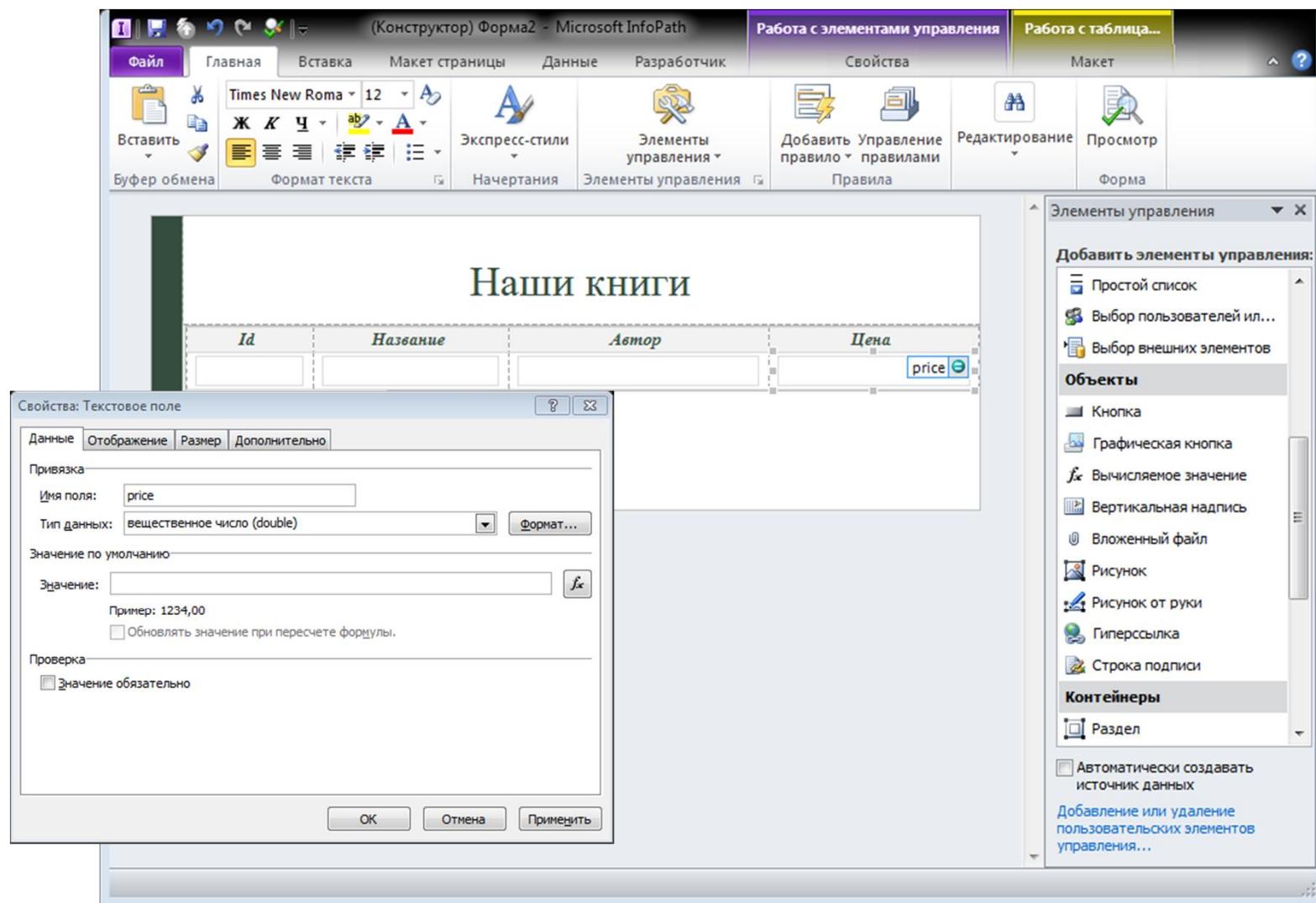
Действия

Добавить поле

Управление подключениями данных...

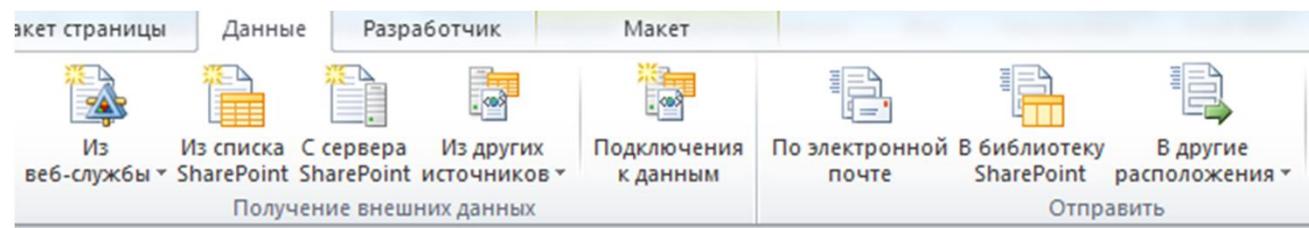


Определение элементов управления

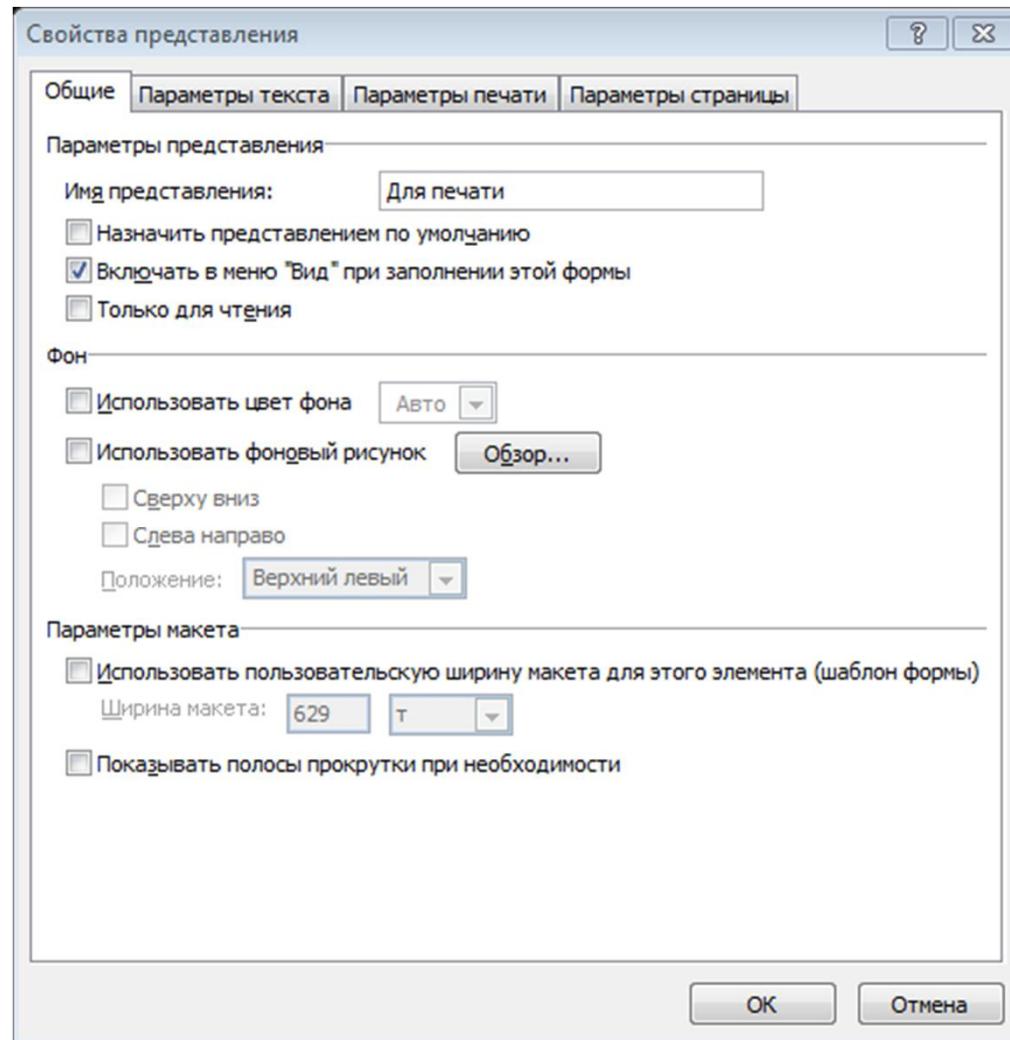


Определение схемы данных

- Схема данных (источник данных) – это логическое представление вводимых пользователем данных
- Источник данных определяет типы элементов, их необходимость, их вложенность
- Форма может иметь несколько источников данных



Определение представлений



Лабораторная работа 8

Создание простого приложения Microsoft Office InfoPath 2010

Упражнение 1: Создание нового документа InfoPath на основе существующего XML-документа

- Выберите в меню Windows
[Start (Пуск) -> All Programs (Все программы) -> Microsoft Office → Microsoft InfoPath Designer 2010]
- Выберите пункт меню «Создать» и в группе «Дополнительные шаблоны форм» щелкните два раза левой кнопкой мыши по пункту «XML или схема»
- В диалоговом окне «Мастер источника данных» нажмите кнопку «Обзор...» и выберите в качестве примера файл **lab-7.xml** из предыдущей лабораторной работы
- Нажмите кнопку «Далее >»
- Выберите «Нет» и нажмите кнопку «Готово»
- В диалоговом окне нажмите кнопку «Нет»

Упражнение 2: Создание формы InfoPath

- На правой панели «Поля» разверните элемент **item** в структуре данных (нажав на иконку [+])
- Щелкните правой кнопкой мыши по элементу **value** и выберите пункт контекстного меню «Свойства»
- Выберите из списка «Тип данных» значение «целое число (integer)»
- Нажмите кнопку «OK»
- Выберите пункт главного меню
[Макет страницы -> Шаблоны макетов страниц -> Только заголовок]
- Щелкните по фразе «Щелкните, чтобы добавить название» и введите «Наши товары»
- На правой панели «Поля» щелкните мышкой по элементу **item** и перетащите его на область «Добавить таблицы»
- Выберите «Повторяющаяся таблица»
- Удалите элемент **Item** из таблицы, выделив столбец и щелкнув правой кнопкой мышки по пункту «Удалить столбцы»

- Щелкните правой кнопкой мыши по колонке с элементом **Org** и выберите пункт контекстного меню
[Вставка -> Столбцы справа]
- Перетащите элемент **value** в новый самый крайний столбец
- Удалите первый пустой столбец
- Поменяйте местами столбы **Org** и **Title**
- Напишите название столбцов по-русски: «Город», «Организация», «Название», «Количество»
- Щелкните правой кнопкой мышки по текстовому полю **Город (@city)** и выберите в контекстном меню
[Изменить элемент управления -> Раскрывающийся список]
- Щелкните правой кнопкой мышки по списку **Город (@city)** и выберите в контекстном меню
[Раскрывающийся список: Свойства...]
- Нажмите на кнопку «Добавить» и введите значение «Москва»
- Нажмите кнопку «OK»
- Аналогично добавьте города «Петербург» и «Бобруйск»
- Нажмите кнопку «OK»
- В главном меню программы выберите
[Файл -> Сохранить]
- Если появится диалоговое окно, нажмите кнопку «OK»
- Сохраните шаблон формы в папку лабораторных работ с именем «Моя форма»
- Закройте программу InfoPath Designer 2010

Упражнение 3: Использование формы InfoPath

- В папке лабораторных работ дважды щелкните по файлу «Моя форма»
- Заполните и сохраните форму с произвольным именем
- Откройте сохраненный файл с помощью Notepad++ и убедитесь, что это ваш файл лабораторных работ (структура документа)
- Закройте все приложения

Выводы

- XML используется повсеместно
- Microsoft Office 2010 очень тесно связан с XML
- XML можно легко подключить к различным офисным приложениям
- Microsoft Office InfoPath 2010 – XML приложение для сбора и обработки бизнес-информации
- Формы InfoPath быстро создаются практически без программирования