

Министерство образования Республики Беларусь
БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ

Кафедра «Таможенное дело»

О.Н. Лабкович

ЯЗЫК РАЗМЕТКИ XML

Методическое пособие
по выполнению лабораторных работ
для студентов дневной формы обучения специальности
1-96 01 01 «Таможенное дело»

Минск
БНТУ
2012

УДК 004.655.2(075.8)

ББК 32.973 – 018.1я7

Л 12

Рецензенты:

начальник управления информационных технологий
таможенной статистики и анализа таможенного комитета

Республики Беларусь *Л.А. Кузьмич*;

д-р физ.-мат. наук, профессор кафедры ЮНЕСКО «Энергосбережение
и возобновляемые источники энергии» БНТУ *М.С. Краков*

Лабкович, О.Н.

Л 12 Язык разметки XML: методическое пособие по выполнению лабораторных работ для студентов дневной формы обучения специальности 1-96 01 01 «Таможенное дело» / О.Н. Лабкович. – Минск: БНТУ, 2012. – 36 с.

ISBN 978-985-525-720-3.

Методическое пособие предназначено для использования в учебном процессе.

Издание содержит основные теоретические сведения и лабораторные работы по языку передачи данных XML.

Методическое пособие имеет практическое назначение.

УДК 004.655.2(075.8)

ББК 32.973 – 018.1я7

Методическое пособие подготовлено при активном участии студентов
Н. Данишевской, В. Коктевой

ISBN 978-985-525-720-3

© Лабкович О.Н., 2012

© БНТУ, 2012

ОГЛАВЛЕНИЕ

ЛАБОРАТОРНАЯ РАБОТА № 1 ОСНОВЫ XML	4
ЛАБОРАТОРНАЯ РАБОТА № 2 СХЕМА DTD	9
ЛАБОРАТОРНАЯ РАБОТА № 3 СХЕМА XSD	16
ЛАБОРАТОРНАЯ РАБОТА № 4 СВЯЗЬ XML С HTML И MS ACCESS	23
ЛИТЕРАТУРА.....	35

ЛАБОРАТОРНАЯ РАБОТА № 1

ОСНОВЫ XML

XML (от англ. eXtensible Markup Language – расширяемый язык разметки) – это язык разметки, фактически представляющий собой свод синтаксических правил.

Язык XML используется для *хранения, передачи и обмена данными*. С помощью XML вы можете осуществлять обмен данными даже между *несовместимыми* системами. В настоящее время компьютеры и базы данных хранят информацию в самых разнообразных форматах. Преобразование данных в XML позволит упростить задачу и сделает данные доступными для самых разнообразных типов программ.

Расширение файла: .xml

Синтаксические правила языка XML однозначны и очень просты. Эти правила легко выучить и легко использовать.

Пример 1

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
<to>Сергей</to>
<from>Наталья</from>
<heading>Напоминание</heading>
<body>Не забудь про наши планы на эти выходные!</body>
</note>
```

Первая строка в документе – это *объявление типа документа* – определяет версию XML и кодировку, которая используется в документе. В нашем случае документ соответствует версии 1.0 спецификации, и использует кодировку UTF-8 (кириллица).

Вторая строка описывает *корневой элемент* документа. Это все равно, что сказать: «этот документ-записка» (note).

Следующие 4 строки описывают 4 *дочерних элемента* к записке (note): кому (to), от кого (from), заголовок (heading), содержание (body).

И, наконец, последняя строка указывает на завершение корневого элемента.

ВАЖНО! В отличие от HTML все элементы XML должны иметь закрывающий тег! Например: `<p>Это первый абзац</p>`

ВАЖНО! Вы могли заметить, что в предыдущем примере объявление типа документа не имеет закрывающий тег. Это не ошибка. Объявление – это не элемент XML документа, а некий *особый элемент*, который не должен иметь закрывающего тега.

Открывающий и закрывающий теги должны быть написаны одинаково с учетом регистра.

В XML документе все элементы обязательно должны быть строго вложенными:

```
<b><i>Этот текст печатается жирным курсивом</i></b>
```

Все XML документы должны иметь корневой элемент. Все XML документы должны содержать единственную пару тегов, определяющую корневой элемент. Все остальные элементы должны быть дочерними элементами этого корневого элемента.

Любой элемент может иметь свой дочерний элемент. Дочерние элементы должны быть строго вложены в родительский элемент:

```
<root>  
<child>  
<subchild>.....</subchild>  
</child>  
</root>
```

В этом примере: root – корневой элемент, child – дочерний элемент, subchild – дочерний элемент дочернего элемента.

В XML значения атрибутов должны записываться в кавычках. Например:

<note date="12/11/22002">

Синтаксис для записи комментариев в XML такой же, как и в HTML.

<!-- Это комментарий -->

XML элементы должны именоваться в соответствии со следующими правилами:

- Имена могут состоять из букв цифр и других символов.
- Имена не могут начинаться с цифры или знака препинания.
- Имена не должны начинаться с последовательности xml (или XML, или Xml и т.д.).
- Имена не могут содержать пробелов.

В спецификации XML разрешается использовать кириллицу для имен элементов, но лучше избегать этого, поскольку различное ПО может неправильно обрабатывать документы, имена в которых написаны не английским алфавитом.

XML элементы могут иметь атрибуты, наподобие атрибутов в HTML. Атрибуты предназначены для указания дополнительной информации об элементе.

ЗАДАНИЯ

1. Создайте такой документ XML, в котором будут представлены такие сведения, как ваше имя, домашний адрес, адрес электронной почты, дата рождения. При указании даты используйте атрибуты. Завершив создание документа, сохраните его с расширением .xml и просмотрите с помощью web-браузера.

2. Ниже приведен пример плохо оформленного экземпляра XML. В этом документе перечисляются два музыкальных альбома и представлены некоторые сведения о них. Зная синтаксис XML, внесите необходимые исправления. Проверьте работоспособность полученного документа.

```
<?xml version=1.0 encoding="UTF-8"?>
<cd number="432">
<title>Africa</Title>
<artist>BeZBileta</artist>
<tracks total=20">
<cd number=97>
<title>HeartBreaker</title>
<Subtitle>Sixteen Classic Performances</subtitle>
<artistPat Benatar</Artist>
<tracks total=16>
</CD>
```

3. Добавьте в исправленный код из задания №2 следующие данные, используя дочерние элементы:

- названия песен из альбома «Africa» (1 – Африка; 5 – Давай останемся; 10 – Красному диску солнца; 11 – Странная жизнь.)
- песня «Красному диску солнца» поется совместно с Сергеем Михалком (feat. Сергей Михалок)
- песня «Африка» имеет длину 4:31 мин, «Красному диску солнца» – 3:53; «Давай останемся» – 5:18;
- песня «Странная жизнь» является бонусом (bonus).

4. Ниже приведен рецепт выпечки хлеба. Опишите его с помощью языка XML. При создании документа используйте атрибуты.

Простой хлеб.

Ингредиенты: 3 стакана муки
25 грамм дрожжей
1,5 стакана теплой воды
1 чайная ложка соли

Смешать все ингредиенты и тщательно замесить. Закрывать тканью и оставить на один час в теплом помещении. Замесить еще раз, положить на противень и поставить в духовку. Выпекать 40 минут.

5. Переделайте созданный ранее документ рецепта так, чтобы каждое новое действие описывалось отдельным элементом. Например, `<step>Смешать все ингредиенты</step>` и т.д.

6. Ознакомьтесь с образцом ГТД и напишите XML документ для нескольких граф согласно своему варианту:

- 1 вариант – графы 1–4;
- 2 вариант – графы 5–8;
- 3 вариант – графы 9–12;
- 4 вариант – графы 13–15а;
- 5 вариант – графы 16–20;
- 6 вариант – графы 21–24;
- 7 вариант – графы 31–34;
- 8 вариант – графы 35–40;
- 9 вариант – графы 41–44;
- 10 вариант – графы 45–48

КОНТРОЛЬНЫЕ ВОПРОСЫ

- 1. Что такое XML?
- 2. Для чего используется XML?
- 3. Как создать XML документ?
- 4. Какую следует задавать кодировку при использовании кириллицы в XML документе?
- 5. Какие типы элементов вы знаете?
- 6. Что такое атрибуты?

ЛАБОРАТОРНАЯ РАБОТА № 2

СХЕМА DTD

Схема DTD (как и любая другая схема) представляет *шаблон* разметки документа, в котором указываются наличие, порядок следования и расположение элементов и их атрибутов в документе XML. Т.е. схемы DTD задают структуру и синтаксис XML документов, а не их содержание.

Можно использовать XML без DTD или любой другой схемы: такие документы будут *формально корректными*, если они удовлетворяют основным требованиям синтаксиса XML. Однако схемы позволяют проверять *правильность содержания* документа на предмет соблюдения некоторых определенных *правил и ограничений*, заданных в схеме DTD. XML-документ считается *действительным*, если в его состав включена схема DTD или XML-схема, а сам документ не противоречит схеме.

Для того чтобы использовать схему DTD в нашем документе, мы можем:

1) описать ее во *внешнем файле* (с расширением .dtd) и при описании схемы DTD просто указать ссылку на этот файл (внешняя схема DTD):

```
<!DOCTYPE имя_элемента SYSTEM "имя_файла.dtd">
```

Например:

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

```
<!DOCTYPE example SYSTEM "example.dtd">
```

```
<example> Это внешняя схема </example>
```

2) непосредственно *внутри* самого документа выделить область, в которой определить нужные правила (внутренняя схема DTD):

```
<!DOCTYPE имя_элемента [
```

```
<!ELEMENT имя_элемента (дочерний_элемент)>
```

```
...
```

```
]>
```

Например:

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

```
<!DOCTYPE example [
```

```
<!ELEMENT example (#PCDATA)>
```

```
]
```

```
<example> Это внутренняя схема </example>
```

Все, что указано в квадратных скобках [...] образует внутреннюю схему DTD.

Все ключевые слова, такие как ELEMENT, #PCDATA, #REQUIRED и т.д. должны указываться с помощью символов верхнего регистра.

Объявление элементов в схеме DTD

Все элементы можно разделить на:

- 1) элементы-контейнеры, которые могут содержать:
 - а) данные (например, текст);
 - б) дочерние элементы;
 - в) и то, и другое (смешанное содержимое);
- 2) пустые элементы.

Объявление элементов имеют следующий вид:

```
<!ELEMENT имя_элемента (содержание)>
```

Например:

```
<!ELEMENT example (#PCDATA)>
```

Содержание элементов:

(#PCDATA)	элемент содержит текстовые данные
(дочерний_элемент)	элемент содержит дочерний элемент
(#PCDATA дочерний_элемент)*	элемент содержит текстовые данные и дочерний элемент
EMPTY	элемент ничего не содержит
ANY	элемент может содержать текстовые данные или дочерние элементы

Объявление атрибутов в схеме DTD

Атрибуты задаются следующим образом:

```
<!ATTLIST имя_элемента  
    имя_атрибута1 (тип) значение атрибута  
    имя_атрибута2 (тип) значение атрибута>
```

Например:

```
<!ATTLIST example  
    number ID #REQUIRED>
```

Стандартные значения атрибутов:

#REQUIRED	атрибут должен быть указан
#FIXED	значение атрибута фиксировано. Если значение атрибута отличается от объявленного, документ не считается правильным
#IMPLIED	атрибут необязателен
“значение_по_умолчанию”	если в документе опустить атрибут, ему будет назначено значение по умолчанию, указанное в кавычках

Существуют 3 типа атрибутов:

- 1) атрибуты, содержащие текстовые данные, указываемые с помощью ключевого слова CDATA;
- 2) маркированные атрибуты, указываемые с помощью определенных ранее маркеров;
- 3)

ID	Уникальным образом идентифицирует элемент (значение атрибута должно начинаться с буквы или символа «_»)
IDREF	Указывает на элементы, содержащие атрибут ID
ENTITY	Ссылка на внешний элемент
NMTOKEN	Содержит буквы, цифры, точки, знаки подчеркивания, переносы и двоеточия, но не пробелы

4) атрибуты с перечислением, предлагающие целый список допустимых значений (при перечислении используется знак |, который воспринимается как логический оператор «или»).

Индикаторы последовательностей и вхождений для схем DTD

,	(a, b, c)	Оператор последовательности разделяет члены списка, когда необходимо последовательное использование членов списка
	(a b c)	Оператор выбора разделяет члены списка, когда необходимо использовать только один из членов списка
	date	Отсутствие какого-либо символа указывает на необходимое вхождение одного и только одного элемента date
?	subject?	Символ указывает на необязательное вхождение (0 или 1 раз) элемента subject
+	paragraph+	Символ указывает на обязательное и повторяющееся вхождение (1 или несколько раз) элемента paragraph
*	brother*	Символ указывает на необязательное и повторяющееся вхождение (0 или несколько раз) элемента brother

Пример 1

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE note [
<!ELEMENT note (message+, outcome+)>
<!ELEMENT message (#PCDATA)>
<!ATTLIST message
    number ID #REQUIRED
    from CDATA #REQUIRED
    alert (несрочно| нормально| срочно) #IMPLIED>
<!ELEMENT outcome (#PCDATA)>
<!ATTLIST outcome
    msg IDREF #IMPLIED>
<!ENTITY today "05.11.09">
    ]>
<note>
    <message number="a1" from="Ольга Петрова"
alert="несрочно">
        &today; Не забудь купить молоко по дороге с работы
домой </message>
        <message number="a2" from="Саша Петров"
alert="срочно">
        &today; Помоги мне сделать домашнее задание
</message>
        <message number="a3" from="Кристина Петрова">
        &today; Поиграй со мной</message>
        <outcome msg="a1">Хорошо, я найду в мага-
зин</outcome>
        <outcome msg="a2">Обязательно помогу</outcome>
        <outcome msg="a3">Конечно, поиграю</outcome>
</note>
```

ЗАДАНИЯ

1. Наберите Пример 1 и просмотрите его с помощью браузера. Выполните следующие изменения в схеме DTD:

1) сделайте так, чтобы атрибуту alert по умолчанию присваивалось значение «нормально» и просмотрите полученный результат;

2) поменяйте дату «05.11.09» на дату выполнения данной лабораторной работы. Обратите внимание, что при просмотре документа символы &today; заменяются значением, указанным вами в DTD схеме;

3) добавьте еще один элемент source, который:

а) является дочерним элементом для элемента note,

б) имеет необязательное вхождение,

с) ничего не содержит,

д) имеет атрибут delivery, которому можно присваивать значения «телефон», «лично» или «e-mail», и которому по умолчанию присваивается значение «e-mail».

4) сделайте схему DTD внешней.

2. В лабораторной работе №1 вы создали XML документ для нескольких граф ГТД (задание № 6). Напишите для этого документа DTD схему.

3. Создайте DTD схему для описания (классификации) товаров товарных позиций ЕТН ВЭД Таможенного союза согласно своему варианту. После этого обменяйтесь созданными схемами с другими вариантами и напишите XML документ для той DTD схемы, которую вы получили.

1 вариант – товарная позиция 9201;

2 вариант – товарная позиция 8712;

3 вариант – товарная позиция 6505;

4 вариант – товарная позиция 4415;

- 5 вариант – товарная позиция 6205;
- 6 вариант – товарная позиция 9108;
- 7 вариант – товарная позиция 9301;
- 8 вариант – товарная позиция 0902;
- 9 вариант – товарная позиция 2402;
- 10 вариант – товарная позиция 0906.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Для чего служит DTD схема?
2. Каким образом в DTD схеме задаются элементы и атрибуты?
3. Что могут содержать элементы?
4. С помощью каких ключевых слов задается содержание элементов? Значения атрибутов?
5. С помощью какого символа в DTD схеме задается обязательное вхождение элемента «book»? Обязательное и повторяющееся вхождение элемента? Последовательное вхождение членов списка?

ЛАБОРАТОРНАЯ РАБОТА № 3

СХЕМА XSD

На смену DTD пришел стандарт консорциума W3C – XML Schema, называемый также XSD (XML Schema Definition).

Корневым элементом в схеме XML является элемент Schema, который содержит все остальные элементы в документе схемы. В рамках корневого элемента схемы XSD атрибутом xmlns определяется пространство имен XML Schema, которое содержит элементы и атрибуты XSD схемы.

```
<xsd:schema  
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

Обратите внимание на то, что *все* элементы XSD начинаются с префикса `xsd:`, который указывается для пространства имен XSD, объявленного в корневом элементе экземпляра схемы.

В самом XML-документе, который проверяется с помощью схемы, должно содержаться объявление *пространства имен*. Пространство имен – это именованная совокупность имён элементов и атрибутов, служащая для обеспечения их уникальности в XML-документе. Пространство имен всегда указывается в корневом элементе экземпляра документа с помощью атрибута `xmlns:`

```
xmlns:xsi=«http://www.w3.org/2001/XMLSchema-instance»
```

В схемах XSD элементы, используемые в документах XML, бывают двух типов: *простые* (`simpleType`) и *сложные* (`complexType`). Элементы сложных типов могут содержать атрибуты, а также дочерние элементы. Элементы простых типов такими возможностями не обладают.

Простой тип	Описание
string	буквенно-цифровая строка
integer	...-1,0,1,...
time	13:20:00.000, 13:20:00.000-05:00

date	1999-05-31
gYear	1999
anyURL	http://www.example.com/, http://www.example.com/doc.html#ID5

Элементы XSD-схем:

Элемент	Описание
annotation	Родительский элемент элементов-комментариев <appInfo> и <documentation>
appInfo	Элемент-комментарий. Задаёт титул схемы
attribute	Атрибут
choice	Выбор других элементов. Аналог оператора " " в DTD
complexContent	Ограничения или расширения модели содержимого сложного типа
complexType	Элемент сложного типа
documentation	Элемент-комментарий. Предоставляет информацию о схеме
element	Элемент
restriction	Ограничение элемента
schema	Корневой элемент схемы
sequence	Последовательность других элементов. Аналог оператора "," в DTD
simpleContent	Модель, содержимое которой представляет только символьные данные
simpleType	Элемент простого типа

Ограничения вхождений в схемах XSD

В отличие от других языков описания схем, XSD позволяет вам определить количество вхождений элемента с определенной точностью. Вы можете задать минимальное и максимальное количество вхождений элемента с помощью атрибутов minOccurs и maxOccurs элемента xsd:element соответственно. На возможные значения этих атрибутов налагаются определенные ограничения:

minOccur="0"	вхождение элемента необязательно
minOccur="1"	предусмотрено одно вхождение элемента
maxOccur="1"	
minOccur и maxOccur не указаны	
minOccur="2 (3,4,...)"	минимальное число вхождений равно 2 (3,4,..)
maxOccur="2 (3,4,...)"	максимальное число вхождений равно 2 (3,4,..)
maxOccur="unbounded"	число вхождений не ограничено

Атрибуты в схемах XSD

Объявление атрибутов XML документов в схемах XSD очень похоже на объявление элементов, за исключением того, что атрибуты объявляются с помощью объявлений `attributes`, а не `element`.

Атрибуты-ограничения XSD-схем:

Атрибут	Описание
length	Длина
maxLength	Максимальная длина
maxInclusive	Максимальное значение включительно
minExclusive	Минимальное значение

Иные атрибуты:

Атрибут	Описание
fixed	Фиксированное значение элемента или атрибута
name	Название элемента или атрибута
ref	Задание ссылки на глобально определенный элемент
schemaLocation	Определение местоположения схемы

type	Тип элемента
use	Является элемент обязательным или нет. Возможные значения атрибута use: <ul style="list-style-type: none"> ◆ required – атрибут является обязательным и может иметь любое значение; ◆ optional – атрибут является необязательным и может иметь любое значение; ◆ fixed – значение атрибута фиксировано, и его можно установить с помощью атрибута value; ◆ default – если атрибута нет, его значение равно значению по умолчанию, установленному для атрибута value (если атрибут присутствует, то его значение равно значению, которое присваивается ему в этом документе); ◆ prohibited – атрибут не должен отображаться.
value	Значение элемента схемы
xsi:schemaLocation	Реальное местоположение элемента в XML-документе
xsi:type	Реальный тип элемента в XML-документе

Пример 1

Документ:

```
<?xml version="1.0" encoding="UTF-8"?>
<note
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="3.xsd">
<message number="10" date="2009-10-29" from="Ольга Петрова">
```

```
Не забудь купить молоко по дороге с работы домой
</message>
</note>
```

Схема XSD(3.xsd):

```
<?xml version="1.0" encoding="Windows-1251"?>
<xsd:schema
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:complexType name="messageType">
    <xsd:simpleContent>
      <xsd:restriction base="xsd:string">
        <xsd:attribute name="number" type="xsd:integer"
use="required"/>
        <xsd:attribute name="date" type="xsd:date" use="required"/>
        <xsd:attribute name="from" type="xsd:string" use="required"/>
      </xsd:restriction>
    </xsd:simpleContent>
  </xsd:complexType>
  <xsd:element name="note">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="message" type="messageType"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Анализ

Обратите внимание: элемент simpleContent и элемент complexType не содержат атрибут name. Вы можете указать эти атрибуты, если планируете повторно использовать определения содержания в других местах схемы. Например, элемент complexType называется messageType. Объявление элемента message содержит ссылку (type="messageType") на объявление сложного типа с объявлением атрибута.

С помощью дескриптора `<xsd:restriction base="xsd:string">` объявляется ограничение типа данных для атрибута. Каждый атрибут относится к типу `string`, поэтому в дальнейшем он ограничивается объявлением типов данных (`integer`, `date`, `string`), а также атрибутом `use` (каждый атрибут обязателен в проверяемом документе XML).

Элемент `sequence` включен только для того, чтобы показать типичное расположение и синтаксис, используемые в экземплярах, содержащих несколько дочерних элементов, для чего необходимы ограничения последовательностей.

ЗАДАНИЯ

1. Рассмотрите приведенные в теории примеры. В Примере 1 добавьте элемент `outcome`, содержащий атрибуты `date`, `number`, `whom`. Добавьте для элемента `message` какой-либо атрибут.

2. Перед вами экземпляр документа XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<note
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="имя_файла.xsd">
<notes>
<number/>
<message>Не забудь купить молоко по дороге с работы до-
мой</message>
<message>Я опаздываю к ужину. Пошли за молоком де-
тей</message>
</notes>
<notes>
<number/>
<message>Поехали в Африку</message>
<message>В Африке много вкусных бананов</message>
```

```
<message>В Африке очень тепло</message>
</notes>
</note>
```

Напишите для него схему XSD

3. Наложите на этот документ ряд ограничений:

- в документе допускается не больше двух элементов notes;
- вхождение элементов notes необязательно;
- элемент number должен предшествовать элементу message;
- должен существовать как минимум один элемент message.

4. В прошлых лабораторных работах вы создавали XML документ для нескольких граф ГТД (задание № 6, л.р. № 1) и писали для этого документа DTD схему. Теперь напишите для него XSD схему.

5. Вам будет представлена XSD схема, в соответствии с которой формируется отчет для таможенных органов в виде файла в формате XML декларантами процедур свободной таможенной зоны, переработки на таможенной территории, переработки вне таможенной территории. Сформируйте для этой схемы XML документ.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Для чего нужна XSD схема?
2. Какие типы элементов в XSD схемах вы знаете? В чем их отличие?
3. Как задаются атрибуты XML документа в XSD схемах?
4. Как с помощью XSD схемы задать определенную последовательность элементов в XML документе?

ЛАБОРАТОРНАЯ РАБОТА № 4

СВЯЗЬ XML С HTML И MS ACCESS

Связывание данных

Связывание данных является одним из методов отображения XML-документа с традиционной HTML-страницы. Связывание данных работает только с XML-документом, который симметрично структурирован, т.е. элементы документа могут быть интерпретированы как набор записей и полей. В простейшем случае такой документ состоит из корневого элемента, содержащего множество элементов одинакового типа (записи), каждый из которых имеет одинаковый набор дочерних элементов, причем все они содержат символьные данные (поля).

Основные шаги при связывании данных:

- 1) Установка связи XML-документа с HTML-страницей, на которой вы хотите отобразить данные XML.
- 2) Связывание элементов HTML с элементами XML. Когда вы связываете HTML-элемент с XML-элементом, HTML-элемент автоматически отображает содержимое XML-элемента.

Связывание документа XML с HTML-страницей

Осуществляется с помощью специального элемента HTML – XML 2 способами:

- 1) весь текст XML-документа помещается между начальным и конечным тегами XML.
- 2) атрибуту src элемента XML присваивается URL файла, содержащего данные XML. При этом сам элемент XML остается пустым.

```
<XML id="name" src="URL"></XML>
```

Например:

```
<XML id="myMsg" src="message.xml"></XML>
```

Атрибуту id элемента XML присваивается уникальное имя, которое используется для доступа к XML-документу с HTML-страницы.

Второй способ более соответствует основам философии XML, согласно которой собственно данные (XML-документ) хранятся отдельно от информации по их форматированию и обработке (HTML-страницы), что облегчает работу с XML-документом. Поэтому далее будем использовать именно второй способ связывания.

Связывание элементов HTML с элементами XML

Вы можете осуществлять связывание HTML-элементов с XML-элементами двумя основными способами:

1) Связывание по отдельным записям, что означает связывание нетабличных элементов HTML (например, элементов SPAN, DIV, MARQUE, IMG и т.д.) с XML-элементами таким образом, что за один раз отображается только одна запись.

```
<SPAN datasrc="#ссылка" datafld="имя элемента"></SPAN>
```

Атрибут datasrc (#имя, присвоенное атрибуту id) указывает на источник данных, которые должны связываться с элементом HTML. Атрибут datafld задает конкретное поле, которое должно связываться.

Например:

```
<SPAN datasrc="#myMsg" datafld="from"></SPAN>
```

2) Табличное связывание, что означает связывание HTML-элемента TABLE с данными XML, так что в таблице автоматически отображается весь набор записей, принадлежащих XML-документу. В этом случае данные, принадлежащие каждому из элементов-записей, отображаются в отдельной строке таблицы, а данные дочерних элементов-полей – в отдельном столбце.


```
<TABLE datasrc="#ссылка">
  <TR>
    <TD><SPAN datafld="имя элемента"></SPAN></TD>
  </TR>
</TABLE>
```

Например:

```
<TABLE datasrc="#myMsg">
  <TR>
    <TD><SPAN datafld="from"></SPAN></TD>
  </TR>
</TABLE>
```

Использование одной HTML-таблицы для отображения простого набора записей

Вы можете использовать один HTML-элемент TABLE для отображения XML-документа, в котором данные организованы в виде простого набора записей – т.е. XML-документа, составленного следующим образом:

- корневой элемент (note) содержит множество элементов типа запись (msg);
- каждый элемент типа запись (msg) содержит одинаковый набор элементов типа поле (source, from, message);
- каждый элемент типа поле (source, from, message) содержит только символьные данные.

Пример 1

XML-документ (message_01.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <msg>
    <source>телефон</source>
    <from>Ольга Петрова</from>
    <message> Не забудь купить молоко по дороге с работы
    домой</message>
```

```

</msg>
<msg>
  <source>e-mail</source>
  <from>Саша Петров</from>
  <message>Помоги мне сделать домашнее
задание</message>
</msg>
<msg>
  <source>телефон</source>
  <from>Кристина Петрова</from>
  <message>Поиграй со мной</message>
</msg>
</note>

```

HTML-документ

```

<HTML>
<HEAD>
<TITLE>XML Data Island</TITLE>
</HEAD>
<BODY>
<!--Связывание документа XML с HTML-таблицей-->
<XML id="myMsg" src="message_01.xml"></XML>
<H1>My messages</H1>
<TABLE datasrc="#myMsg" border="6" width="100%" >
  <THEAD style="background-color:aqua">
    <TH>Source</TH>
    <TH>From</TH>
    <TH>Message</TH>
  </THEAD>
  <TR valign="top" align="center">
    <!--Связывание элементов HTML с элементами XML--
>
    <TD><SPAN datafld="source"></SPAN></TD>
    <TD><SPAN datafld="from"></SPAN></TD>

```

```

        <TD><SPAN datafld="message"></SPAN></TD>
    </TR>
</TABLE>
</BODY>
</HTML>

```

Использование вложенных таблиц для отображения иерархической структуры записей

Для отображения XML-документа, элементы которого структурированы как иерархический набор записей используются вложенные таблицы. В иерархическом наборе записей каждая запись может содержать, в дополнение к фиксированному набору полей, переменное число вхождений (нуль или более).

Пример 2

XML-документ (message_02.xml)

```

<?xml version="1.0" encoding="UTF-8"?>
<notes>
<note>
    <alert>срочно</alert>
    <msg>
        <source>телефон</source>
        <from>Ольга Петрова</from>
        <message>Заплати за телефон!</message>
    </msg>
</note>
<note>
    <alert>нормально</alert>
    <msg>
        <source>e-mail</source>
        <from>Саша Петров</from>
        <message>Помоги мне сделать домашнее
задание</message>

```

```

</msg>
<msg>
  <source>телефон</source>
  <from>Ольга Петрова</from>
  <message> Не забудь купить молоко по дороге с работы
домой</message>
</msg>
</note>
<note>
  <alert>несрочно</alert>
  <msg>
    <source>телефон</source>
    <from>Кристина Петрова</from>
    <message>Поиграй со мной</message>
  </msg>
</note>
</notes>

```

HTML-документ

```

<HTML>
<HEAD>
<TITLE>XML Data Island</TITLE>
</HEAD>
<BODY>
<XML id="myMsg" src="message_02.xml"></XML>
<!--Внешняя таблица-->
<TABLE datasrc="#myMsg" BORDER="1">
<THEAD>
<TH>My messages</TH>
  </THEAD>
  <TR>
    <TD><SPAN datafld="alert"></SPAN></TD>
  </TR>
<!--Внутренняя таблица-->

```

```

<TR>
  <TD>
    <TABLE datasrc="#myMsg" datafld="msg">
      <THEAD>
        <TH>Source</TH>
        <TH>From</TH>
        <TH>Message</TH>
      </THEAD>
      <TR align="center">
        <TD><SPAN datafld="source"></SPAN></TD>
        <TD><SPAN datafld="from"></SPAN></TD>
        <TD><SPAN datafld="message"></SPAN></TD>
      </TR>
    </TABLE>
  </TD>
</TR>
</TABLE>
</BODY>
</HTML>

```

Вот как будет выглядеть документ на HTML-странице:

My messages		
срочно		
Source	From	Message
телефон	Ольга Петрова	Заплати за телефон!
нормально		
Source	From	Message
e-mail	Саша Петров	Помоги мне сделать домашнее задание
телефон	Ольга Петрова	Не забудь купить молоко по дороге с работы домой
несрочно		
Source	From	Message
телефон	Кристина Петрова	Поиграй со мной

Связывание HTML-элементов с XML-атрибутами

При связывании данных атрибут трактуется как дочерний элемент. Например, элемент `msg` содержит атрибут `time`:

```
<msg time="12.00"/>
```

Эта запись трактуется, как если бы атрибут `time` был дочерним элементом `msg`:

```
<msg><time>12.00</time></msg>
```

Следовательно, вы можете получить доступ к значению атрибута с использованием обычной техники связывания данных.

Например:

```
<SPAN datasrc="#myMsg" datafld="time"></SPAN>
```

Следует учитывать, однако, что при добавлении атрибута к одному из *элементов-полей* в XML-документе, например, добавлении атрибута `number` к полю `message`

```
<message number="a1">
```

Не забудь купить молоко по дороге с работы домой

```
</message>
```

после связывания данных элемент `message` будет интерпретироваться следующим образом:

```
<message>
```

```
<number>a1</number>
```

Не забудь купить молоко по дороге с работы домой

```
</message>
```

В этом случае набор записей превратится в иерархический набор, а не в простой набор записей, и вам придется отображать вложенные записи с использованием вложенной таблицы. Чтобы иметь возможность отобразить как символьные данные (Не забудь купить молоко по дороге с работы домой), так и атрибут (`a1`) следует использовать специальное имя `$TEXT` для обращения ко всем символьным данным элемента, не включая при этом значений атрибута.

```

<TD>
<!--Обязательно укажите элемент-поле, которому принад-
лежит атрибут -->
<TABLE datasrc="#myMsg" datafld="message">
<TR>
<!--Ссылка на текст "Не забудь купить молоко по дороге с
работы домой" -->
    <TD><SPAN datafld="$TEXT"></SPAN></TD>
    <TD><SPAN datafld="number"></SPAN></TD>
</TR>
</TABLE>
</TD>

```

Импорт XML документов в БД MS Access

Для того чтобы импортировать несколько XML документов в одну таблицу базы данных MS Access необходимо выполнить следующие действия:

- 1) Файл – Внешние данные – Импорт
- 2) В строке Тип файлов указать XML (*.xml, *.xsd)
- 3) а) Выбрать необходимую XSD схему
б) Если XSD схемы нет, то выбрать необходимый XML-документ и во вкладке Параметры выбрать параметры импорта – Только структура
- 4) Нажать ОК.

Затем необходимо повторить вышеописанные действия, только на вкладке Параметры выбрать параметры импорта – Добавить данные в таблицы.

ЗАДАНИЯ

1. Перед вами экземпляр документа XML:
 <?xml version="1.0" encoding="UTF-8"?>
 <Тов_поз>

```
<код>1801 00 000 0</код>
<наименование>Какао-бобы, целые или дробленые, сырые
или жареные</наименование>
<доп_ед>-</доп_ед>
<ставка>0</ставка>
</Тов_поз>
```

Используя связывание данных по отдельным записям, свяжите данные документа XML с элементами HTML так, чтобы получился такой результат:

Единый таможенный тариф Таможенного союза

Код ТН ВЭД: 1801 00 000 0

Наименование позиции: Какао-бобы, целые или дробленые, сырые или жареные

Доп.ед.изм.: -

Ставка ввозной таможенной пошлины: 0

2. Измените предыдущий XML-документ так, чтобы:
 - корневым элементом был элемент ЕТТ;
 - элемент ЕТТ содержал набор из 3-4 элементов-записей (элементы Тов_поз);
 - каждый из элементов-записей имел одинаковый набор элементов-полей, которые содержат только символьные данные (код, наименование, доп_ед, ставка).

Для заполнения XML документа используйте товары Группы 18 единой ТН ВЭД ТС.

Свяжите данные из полученного XML-документа с элементами HTML, используя HTML-элемент TABLE.

3. Преобразуйте документ XML так, чтобы он имел следующую структуру:
 - корневой элемент ЕТТ содержит группу записей Разделы;

- каждая запись Разделы начинается с поля Раздел, которое содержит только символьные данные, далее следует вложенная запись Группы;
- каждая запись Группы начинается с поля Группа, которое содержит только символьные данные;
- затем следуют вложенные записи Тов_поз;
- каждая вложенная запись Тов_поз имеет 3 элемента поля (код, наименование, доп_ед, ставка).

Используя вложенные таблицы, отобразите XML-документ на HTML-странице так, чтобы получился такой результат:

Единый таможенный тариф Таможенного союза

Раздел IV Готовые пищевые продукты; алкогольные и безалкогольные напитки и уксус; табак и его заменители			
Группа 18 Какао и продукты из него			
Код ТН ВЭД	Наименование позиции	Доп. ед. изм.	Ставка ввозной таможенной пошлины
1801 00 000 0	Какао-бобы, цельные или дробленые, сырые или жареные	-	0
1802 00 000 0	Шелуха, оболочки, кожица и прочие отходы какао	-	5
1803	Какао-паста, обезжиренная или необезжиренная:		
1803 10 000 0	– необезжиренная	-	5
1803 20 000 0	– частично или полностью обезжиренная	-	5

4. Измените предыдущий документ так, чтобы информация о номере раздела (Раздел IV) перешла из текстового содержимого элемента Раздел в атрибут данного элемента. Используя технику связывания атрибутов с HTML-элементами, отобразите измененный документ на HTML-странице.

5. В задании № 6 лабораторной работы № 1 вы создали XML-документ. Отобразите его на HTML-странице.

6. Создайте еще несколько XML документов на основании XSD схемы из задания № 5 лабораторной работы № 3.

Импортируйте созданные вами документы в базу данных Access. К полученной таблице создайте запросы:

1) Запрос на выборку с группировкой, который выводит поле CUNN и подсчитывает количество ГТД, по которым вывозится готовая продукция с территории СЭЗ, для каждого резидента СЭЗ;

2) Запрос на выборку с условиями отбора, который выводит Дату размещения товара в свободной таможенной зоне и подсчитывает количество ГТД, по которым товары были размещены в СТЗ за 2009 год;

3) Параметрический запрос, который запрашивает уникальный номер плательщика резидента СЭЗ и выводит поля CUNN, GTDRAZR и GTDGDF.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие XML документы можно отобразить на HTML-странице?
2. Какие два этапа необходимо пройти, чтобы отобразить XML-документ на HTML-странице?
3. Какими способами можно связать документ XML с HTML-страницей? Элементы HTML с элементами XML?
4. Как импортировать XML-документ в базу данных?

ЛИТЕРАТУРА

1. Холзнер, С. XML. Энциклопедия / С. Холзнер. – 2-е изд. – СПб.: Питер, 2004. – 1101 с.
2. Шеперд, Д. Освой самостоятельно XML за 21 день / Д. Шеперд. – 2-е изд. – М.: Издательский дом «Вильямс», 2002. – 432 с.
3. Пособие по XML / пер. с англ. А. Сычев [Электронный ресурс] – Режим доступа: <http://www.sytchev.ru/xml.index>. – Дата доступа: 31.03.2010.
4. XML // Википедия – свободная энциклопедия [Электронный ресурс] – Режим доступа: <http://ru.wikipedia.org/wiki/XML>. – Дата доступа: 24.03.2010.
5. Лекция: Типизированный объект DataSet. Чтение и запись XML-документов / Интернет университет информационных технологий [Электронный ресурс] – Режим доступа: <http://www.intuit.ru/department/se/msadonet/11/>. – Дата доступа: 31.03.2010.
6. Учебник по XML // Realcoding [Электронный ресурс] – Режим доступа: <http://www.realcoding.net/article/rubric/webtech/XMLbook>. – Дата доступа: 24.03.2010.
7. Электронное декларирование // Государственный таможенный комитет Республики Беларусь [Электронный ресурс] – Режим доступа: <http://www.gtk.gov.by/ru/eldeclaration>. – Дата доступа: 24.03.2010.
8. Связывание данных // Realcoding [Электронный ресурс] – Режим доступа: <http://www.realcoding.net/teach/xml/g18/g18-1.html>. Дата доступа: 23.03.2010.
9. Язык XML – практическое введение// Knigodrom [Электронный ресурс] – Режим доступа: <http://knigodrom.net/xml/index.html>. Дата доступа: 23.03.2010.

Учебное издание

ЛАБКОВИЧ Ольга Николаевна

ЯЗЫК РАЗМЕТКИ XML

Методическое пособие
по выполнению лабораторных работ
для студентов дневной формы обучения специальности
1-96 01 01 «Таможенное дело»

Технический редактор О.В. Песенько

Подписано в печать 31.01.2012.

Формат 60x84 1/16. Бумага офсетная.

Отпечатано на ризографе. Гарнитура Таймс.

Усл. печ. л. 2,09. Уч.-изд. л. 1,64. Тираж 100. Заказ 747.

Издатель и полиграфическое исполнение:

Белорусский национальный технический университет.

ЛИ № 02330/0494349 от 16.03.2009.

Проспект Независимости, 65. 220013, Минск.



Министерство образования
Республики Беларусь

**БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Кафедра «Таможенное дело»

О.Н. ЛАБКОВИЧ

ЯЗЫК РАЗМЕТКИ XML

Методическое пособие

Минск
БНТУ
2012