

### Лабораторная работа № 3. Отображение XML с использованием XSLT

XSL является приложением XML, т. е. XSL-таблица представляет собой корректно сформированный XML-документ, который отвечает правилам XSL. Подобно любому XML-документу, XSL-таблица стилей содержит простой текст, и вы можете создать ее с помощью вашего любимого текстового редактора.

Связывание XSL-таблицы стилей с XML-документом. Вы можете связать XSL-таблицу стилей с XML-документом, включив в документ инструкцию по обработке `xml-stylesheet`, которая имеет следующую обобщенную форму записи:

```
<?xml-stylesheet type="text/xsl" href="3.xslt"?>
```

Таблица стилей при этом должна размещаться на том же домене, что и XML-документ, с которым вы ее связываете.

Если вы не связали XML-документ ни с CSS-таблицей, ни с XSL-таблицей стилей, Internet Explorer 5 отобразит документ с помощью встроенной XSL-таблицы, которая используется по умолчанию. Эта таблица стилей отображает исходный XML-текст в виде дерева с возможностью свертывания/развертывания уровней.

В отличие от CSS, содержащей правила, XSL-таблица стилей включает один или несколько шаблонов, каждый из которых содержит информацию для отображения в определенной ветви элементов в XML-документе.

Каждая XSL-таблица стилей должна иметь элемент Документ, известный как корневой элемент, является XML-элементом верхнего уровня, который содержит все остальные элементы.

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

Элемент Документ `xsl:stylesheet` служит не только хранилищем других элементов, но также идентифицирует документ как XSL-таблицу стилей. Все XSL-элементы принадлежат пространству имен `xsl` – т. е. вы предваряете имя каждого XSL-элемента префиксом `xsl:`, обозначающим пространство имен.

Элемент Документ `xsl:stylesheet` XSL-таблицы стилей должен содержать один или несколько шаблонов элементов, которые для краткости будем называть шаблонами.

```
<xsl:template match="/">
  <!-- дочерние элементы ... -->
</xsl:template>
```

Браузер использует шаблон для отображения определенной ветви элементов в иерархии XML-документа, с которым вы связываете таблицу стилей. Атрибут `match` шаблона указывает на определенную ветвь.

Значение атрибута `match` носит название образца (pattern). Образец в данном примере ("/") представляет корневой элемент всего XML-документа.

Каждая XSL-таблица стилей должна содержать один и только один шаблон с атрибутом `match`, который имеет значение "/".

Корневой образец ("/") не представляет элемент Документ (или корневой элемент) XML-документа. Он представляет весь документ, для которого элемент Документ является дочерним. (Т. е. он аналогичен корневому узлу Document в объектной модели документа DOM)

Пример использования XSL-таблицы стилей для предоставления информации в виде таблицы:

```
<?xml version="1.0" encoding="UTF-8"?>
    <xsl:stylesheet version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:template match="/">
    <html>
    <body>
    <h2> Список специальностей факультета ИТ</h2>
        <table border="1">
            <tr bgcolor="#9acd32">
                <th style="text-align:center">Специальность</th>
                <th style="text-align:center">Срок обучения</th>
                <th style="text-align:center">Предметы ЦТ</th>
                <th style="text-align:center">План набора</th>
                <th style="text-align:center">Проходной балл</th>
            </tr>
            <xsl:for-each select="FACULTY/SPECIALIZATION">
                <tr>
                    <td><xsl:value-of select="NAME"></td>
                    <td><xsl:value-of select="TIME"></td>
                    <td><xsl:value-of select="EXAM"></td>
                    <td><xsl:value-of select="PAGES"></td>
                    <td><xsl:value-of select="PASSING"></td>
                </tr>
            </xsl:for-each>
        </table>
    </body>
    </html>
    </xsl:template>
    </xsl:stylesheet>
```

Оператор пути в значении атрибута `select` относится к текущему элементу. Каждый контекст внутри XSL-таблицы стилей относится к текущему элементу. Если вы опустите атрибут `select` для XSL-элемента

value-of, элемент будет осуществлять вывод текстового содержимого плюс текстовое содержимое всех дочерних элементов в текущий элемент.

Порядок элементов value-of в шаблоне определяет порядок, в котором браузер отображает эти элементы. Таким образом, даже из этой простой таблицы стилей вы можете понять, что XSL-таблица стилей является гораздо более гибкой, чем CSS, которая всегда отображает элементы в том порядке, в котором они следуют в документе.

Элемент **for-each** выполняет две основные задачи:

- осуществляет вывод блока элементов, содержащихся внутри элемента for-each, повторяя его для каждого XML-элемента в документе, отвечающего образцу, присвоенному атрибуту select элемента for-each;

- внутри элемента for-each задает текущий элемент, устанавливаемый атрибутом select элемента for-each.

Не нужно включать в XSL-шаблон элементы, представляющие элементы HTML или BODY, которые являются стандартными составными частями HTML-страницы, поскольку браузер сам эффективно их формирует.

Каждый из элементов, представляющих HTML-разметку, должен быть корректно сформированным XML-элементом, а также стандартным HTML-элементом. Не забывайте, что XSL-таблица стилей является XML-документом.

### Задание к лабораторной работе № 3

Оформите задание лабораторной работы № 1 через подключение XSLT.