Лабораторная работа № 11 «Описание структуры XML документа средствами DTD»

**Аннотация:**Цель работы: ознакомиться с конструкциями языка XML и его реализациями, со способами описания структуры документа средствами DTD. Изучить способы обработки XML документов

#### Примеры выполнения лабораторной работы можно посмотреть в папке с лабораторной работой

#### Синтаксис XML

Любой XML-документ состоит из следующий частей:

* Необязательный пролог.
* Тело документа.
* Необязательный эпилог, следующего за деревом элементов.

Рассмотрим каждую из частей более подробно.

#### Пролог

Пролог состоит из нескольких частей:

1. необязательное объявление XML (XML Declaration) которое заключено между символами <?...?>. Объявление содержит:
   * пометку xml и номер версии (version) спецификации XML;
   * указание на кодировку символов (encoding), в которой написан документ (по умолчанию encoding="UTF-8");
   * параметр standalone который может принимать значения "yes" или "no" (по умолчанию standalone="yes"). Значение "yes" показывает, что в документе содержатся все требуемые декларации элементов, a "no" - что нужны внешние определения DTD.

Все это вместе может выглядеть следующим образом:

<?xml version ="1.0" encoding-"windows-1251" standalone="yes"?>.

Важно отметить, что в объявлении XML только атрибут version является обязательным, все остальные атрибуты могут быть опущены и, следовательно, принимать значения по умолчанию. Так же нужно помнить, что все эти атрибуты следует указывать только в приведенном выше порядке.

1. комментарии.
2. команды обработки.
3. символы пустых пространств.

Необязательное *объявление типа документа*, DTD (Document *Type Declaration*) которое заключено между символами <!DOCTYPE...> и может занимать несколько строк. В этой части объявляются теги, использованные в документе, или приводится ссылка на файл, в котором записаны такие объявления.

После *объявление типа документа* так же могут следовать комментарии, команды обработки и символы пустых пространств.

Поскольку все эти части необязательны, пролог может быть опущен.

#### Тело документа

Тело документа, состоит из одного или больше элементов. В правильно оформленном XML документе элементы формируют простое иерархическое дерево, в котором обязательно присутствует корневой элемент (root element) в который вложены все остальные элементы документа. Имена элементов должны быть уникальны в пределах документа. Имя корневого элемента считается именем всего документа и указывается во второй части пролога после слова Doctype.

Элемент начинается открывающим тегом, затем идет необязательное содержимое элемента, после чего записывается закрывающий тег (в отличие от HTML наличие закрывающего тега обязательно, исключением являются элементы без содержания, так называемые пустые элементы, которые могут быть записаны в сокращенной форме: <имя\_элемента/> ). В качестве содержимого элемента могут выступать:

1. другие элементы
2. символьные данные
3. ссылки на символы

Для того, что бы вставить в текст документа некоторый символ, который, например не присутствует в раскладке клавиатуры либо может быть неправильно истолкован анализатором, используют ссылки на символы. Ссылка на символ обязательно начинается со знака "амперсанта" и заканчивается точкой с запятой. Ссылки на символы записываются в следующем виде:

&# код\_символа\_в\_Unicode;.

Код символа можно записать и в шестнадцатеричном виде. В этом случае перед ним ставится символ "x":

&#xШестнадцатеричный\_код\_символа;.

1. ссылки на сущности

Ссылки на сущности позволяют включать любые строковые константы в содержание элементов или значение атрибутов. Ссылки на сущности, как и ссылки на символы, начинающиеся с амперсанта, после которого идет имя сущности и заканчивающиеся точкой с запятой:

&имя\_сущности;.

Ссылки на сущности указывают программе-анализатору подставить вместо них строку символов заранее заданную в определении типа документа.

1. комментарии

Если надо вставить в текст документа комментарий либо сделать какой-то фрагмент "невидимым" для программы-анализатора, то его оформляют следующим образом:

<!--…текст комментария…-->

1. разделы CDATA

Секция CDATA используется, для того чтобы задать область документа, которую при разборе анализатор будет рассматривать как простой текст, игнорируя любые инструкции и специальные символы. Программа-анализатор не разбивает секцию CDATA на элементы, а считает ее просто набором символов. В отличие от комментариев, содержание данной секции не игнорируется, а передается без изменений на выход программы анализатора, благодаря чему его можно использовать в приложении.

Секция CDATA начинается со строки <![CDATA[ после которой записывается содержимое секции. Завершается секция двумя закрывающими квадратными скобками и знаком "меньше":

<![CDATA[ содержание секции ]]>

1. инструкции по обработке.

Инструкции по обработке содержат указания программе анализатору документа XML. Инструкции по обработке заключаются между символами <? и ?>. Сразу за начальным вопросительным знаком записывается имя программного модуля, которому предназначена инструкция. Затем, через пробел, идет сама инструкция, передаваемая программному модулю. Сама инструкция это обычная строка, которая не должна содержать набор символов "?>", означающий конец инструкции. Примером инструкции по обработке может служить строка объявления XML:

<?xml version="1.0" encoding="windows-1251"?>

Эта инструкция предназначена программе, обрабатывающей документ XML. Инструкция передает ей номер версии и кодировку, в которой записан документ.

Открывающие теги либо теги пустых элементов в XML могут содержать атрибуты, представляющие собой пару имя=значение. В одном открывающем теге разрешается использовать только один экземпляр имени атрибута. Атрибуты могут содержать ссылки на объекты, ссылки на символы, текстовые символы. В отличие от языка HTML, в XML значения атрибутов обязательно надо заключать в апострофы ('), либо в кавычки ("). Таким образом, атрибут может быть записан в одном из двух форматов:

имя\_атрибута="значение\_атрибута"

имя\_атрибут= 'значение\_атрибута'.

### Описание структуры документа средствами DTD

Для связывания декларации DTD с экземпляром документа в версии *XML* 1.0 предлагается специальная декларация DOCTYPE.

Декларация DOCTYPE содержит *ключевое слово* DOCTYPE, за которым следует имя корневого элемента документа, а затем конструкция с декларациями содержания.

Можно написать внешнее *подмножество* деклараций в отдельном файле DTD, включить внутреннее *подмножество* в тело декларации DOCTYPE или сделать то и другое. В последнем случае (смешение внутренних и внешних *DTD*) во внутренних *DTD* могут быть заданы новые декларации или переписаны те, что содержатся во внешних (*по* определению спецификации *XML* анализаторы сначала читают внутреннее *подмножество*, и потому содержащиеся там декларации пользуются приоритетом).

Блок внутренней декларации разметки тега DOCTYPE состоит из левой квадратной скобки, списка деклараций и правой квадратной скобки:

<! DOCTYPE root\_element\_name […здесь находятся декларации внутреннего подмножества ... ]>

Для внешних DTD декларация DOCTYPE состоит из обычного ключевого слова и имени корневого элемента, за которым следует еще одно *ключевое слово* SYSTEM либо PUBLIC, обозначающее источник внешнего определения DTD, а за ним - *локализация* этого определения.

Допустимое в документе *XML* содержание определяется с помощью четырех типов декларации разметки в *DTD*. В следующей далее таблице показаны связанные с этими декларациями ключевые слова и их значения. Первые два типа связаны с информацией, которую мы рассчитываем найти в документе *XML*, - элементами и атрибутами. Последние два типа используются для поддержки. Особенно облегчают жизнь разработчика словаря *XML* сущности. Как правило, они состоят из содержания, которое настолько часто используется в DTD или документе, что оправдывает создание специальной декларации. Применение этой декларации напоминает оператор include в языках C/C++, когда в качестве замены для содержания используется имя. Нотации описывают содержание, разработанное не на языке *XML*. Используются они для того, чтобы объявить конкретный *класс* данных и связать его с внешней программой.

|  |  |
| --- | --- |
| **Конструкция DTD** | **Значение** |
| ELEMENT | Декларация типа элемента XML |
| ATTLIST | Декларация атрибутов, которые могут быть назначены конкретным типам элементов, а также разрешенных значений этих атрибутов |
| ENTITY | Декларация повторно используемого содержания |
| NOTATION | Декларация форматирования для внешнего содержания, которое не должно быть проанализировано (например, двоичные данные), а также для внешних приложений, обрабатывающих содержание |

Рассмотрим более подробно первые два типа декларации разметки в *DTD*.

В документе *XML* должен быть описан каждый элемент. Объявление элемента начинается с символов <!ELEMENT, после которых через *пробел*идет имя элемента и его содержимое. Заканчивается объявление символом "больше". *По* своему содержанию элементы делятся на четыре группы.

1. Пустой элемент - может иметь атрибуты, но не содержит текст или порожденные элементы. Объявляется следующим образом: после имени элемента указывается ключевое слово EMPTY.
2. Пример:

<!ELEMENT element\_name EMPTY>

1. Элемент содержит только порожденные элементы, но не текст. Объявляется следующим образом: после имени элемента в скобках через запятую перечисляются все вложенные элементы. Причем вложенные элементы должны следовать в XML документе в том порядке, в каком они перечислены в объявлении.
2. Пример:

<ELEMENT element\_name ( elem\_1,elem\_2)>

1. Элемент содержит не только порожденные элементы, но и текст. Объявляется следующим образом: после имени элемента в скобках указывается ключевое слово #PCDATA, после которого через запятую, как и в предыдущем случае, перечисляются все вложенные элементы (если они имеются).
2. Пример:
3. <ELEMENT element\_name (#PCDATA, elem\_1,elem\_2)>

<ELEMENT element\_name (#PCDATA)>

1. Элемент, открытый для любого содержания. Объявляется следующим образом: после имени элемента указывается ключевое слово ANY.
2. Пример:

ELEMENT element\_name ANY>

Иногда из нескольких вложенных элементов или списков (*список* элементов указанных в скобках) может встретиться только один. В таком случае их имена перечисляются через вертикальную черту( | ). Например:

<!ELEMENT element\_name (elem\_1,(elem\_2|elem\_3))> - элемент element\_name должен

содержать элемент elem\_1, а затем либо elem\_2, либо elem\_3.

Элементы появляются именно в таком порядке.

Если вложенный элемент можно записать в объявляемом элементе несколько раз, то необходимо это указать использую звездочку, плюс или вопросительным знак.

? - элемент или *список* может встретиться нуль или один раз;

\* - элемент или *список* может встретиться нуль или несколько раз;

+ - элемент или *список* может встретиться один или несколько раз.

После объявления элемента объявляются его атрибуты. Все атрибуты одного элемента объявляются сразу, одним списком. *Список* начинается с символов <!ATTLIST, после них через *пробел* следует имя элемента, к которому относятся атрибуты. Затем идет имя атрибута, его тип или *список* значений, которые он может принимать (все значения перечисляются через вертикальную черту, в скобках), признак обязательности присутствия атрибута в элементе или *значение* *по* умолчанию (это *значение* будет использовано, если *атрибут* не записан явно в *XML*документе).

Тип атрибута записывается одним из ключевых слов:

1. CDATA - строка символов.
2. ID - уникальный идентификатор, однозначно определяющий элемент, в котором встретился этот атрибут; значения такого атрибута не должны повторяться в документе. Они играют ту же роль, что и первичные ключи в таблицах баз данных.
3. IDREF - идентификатор, содержащий одно из значений атрибутов типа id, используется в качестве ссылки на другие элементы.
4. *IDREFS* - идентификатор, содержащий набор значений атрибутов типа id, перечисленных через пробелы; тоже используется в качестве ссылки сразу на несколько элементов.
5. ENTITY - имя непроверяемой анализатором сущности объявленных в этом же описании DTD.
6. ENTITIES - имена непроверяемых сущностей.

NMTOKEN - *слово*, содержащее только символы, применяемые в именах. Атрибуты этого типа могут содержать имена других элементов или атрибутов, например, для того чтобы ссылаться на них.

1. NMTOKENS - слова, перечисленные через пробелы.
2. NOTATION - обозначение расшифрованное в описании DTD.

Признак обязательности записывается с использование ключевых слов:

1. #REQUIRED - атрибут надо обязательно записывать в элементе;
2. #IMPLIED - атрибут необязателен, у него нет значения по умолчанию;
3. #FIXED - у атрибута есть только одно значение, которое записывается тут же, через пробел.

Пример:

<!ATTLIST city type (город | поселок | деревня) "город">

<!ATTLIST pre xml:lang NMTOKEN "ru\_RU">

<!ATTLIST pre xml:space (default | preserve) "preserve">