

Технологии Java

Описание и проверка структуры XML

<http://kgeorgiy.info/courses/java-advanced/>

Содержание

1. DTD
2. XML Schema
3. Проверка структуры XML
4. Заключение

Часть 1

DTD

Document Type Definition

- DTD – язык описания структуры XML-документов
- Описание не является XML-документом
- DTD не предназначен для описания документов с пространствами имен

Ссылки на файлы

- Синтаксис

- СсылкаНаФайл

- СсылкаНаСистемныйФайл |

- СсылкаНаОбщедоступныйФайл

- СсылкаНаСистемныйФайл

- SYSTEM** "ИмяФайла"

- СсылкаНаОбщеизвестныйФайл

- PUBLIC** "Идентификатор" "ИмяФайла"

- Примеры

- **SYSTEM** "books.xml"

- **PUBLIC** "-//Examples/BookExample" "books.xml"

Указание DTD

- Синтаксис
 - **<!DOCTYPE** ИмяКорневогоЭлемента
СсылкаНаФайл? ВнутреннееОписание?>
 - ВнутреннееОписание ::= [ТелоDTD]
- Примеры
 - **<!DOCTYPE books SYSTEM "books.dtd">**
 - **<!DOCTYPE books PUBLIC "-
//Examples/BookDTD" "books.dtd">**
 - **<!DOCTYPE books [...]>**
 - **<!DOCTYPE books SYSTEM "books.dtd" [...]>**

Описание элемента

- Описание структуры содержимого
 - Описываются возможные вложенные элементы и текст
- Описание атрибутов
 - Описываются имена, типы и значения атрибутов по умолчанию

Описание структуры содержимого

- Синтаксис
 - **<!ELEMENT ИмяЭлемента Содержимое>**
 - **Содержимое**
 - **EMPTY** - без содержимого
 - **ANY** - любое содержимое
 - **Дети** - только вложенные элементы
 - **Смешанное** - вложенные элементы и текст
- Примеры
 - **<!ELEMENT author EMPTY>**
 - **<!ELEMENT text ANY>**

Описание детей

- Синтаксис
 - (Выбор | Последовательность) Количество
 - Дети
 - (Имя | Выбор | Последовательность) Количество
 - Выбор
 - (Дети | Дети | ... | Дети) Количество
 - Последовательность
 - (Дети , Дети , ..., Дети) Количество
 - Количество
 - - Один
 - ? - Ноль или один
 - + - Один и более
 - * - Любое

Примеры описания детей

- Только книги

```
<!ELEMENT library (book)*>
```

- Книги и журналы вперемешку

```
<!ELEMENT library (book | magazine)*>
```

- Сначала книги, потом журналы

```
<!ELEMENT library (book*, magazine*)>
```

- Книги и журналы парами

```
<!ELEMENT library (book, magazine)*>
```

- Блок кода

```
<!ELEMENT body (begin?, (if | while | for)*, end?)>
```

Описание смешанного содержимого

- Синтаксис
 - (**#PCDATA** | Имя | Имя | ... Имя) *
- Примеры
 - (**#PCDATA** | br | emboss | img)*
 - (**#PCDATA**)*
 - (**#PCDATA**)

Описание атрибутов

- Синтаксис
 - **<!ATTLIST** Имя ОписаниеАтрибута***>**
 - **Описание атрибута**
 - Имя ТипАтрибута ЗначениеПоУмолчанию

Типы атрибутов

- Строковые
 - **CDATA**
- Проверяемые
 - **ID** - Идентификатор
 - **IDREF** - Ссылка на идентификатор
 - **IDREFS** - Ссылки на идентификатор
 - **NMTOKEN** - Имя
 - **NMTOKENS** - Имена
- Перечислимые
 - (**Имя | Имя | ... | Имя**)

Значения по умолчанию

- Значение по умолчанию
 - "значение"
- Значение по умолчанию
 - **#FIXED** "значение"
- Без значения по умолчанию
 - **#IMPLIED**
- Обязательно указывать
 - **#REQUIRED**

Пример описания атрибутов

- Рамка

```
<!ATTLIST border
```

ID	ID	#REQUIRED
type	(single double)	"single"
color	CDATA	#REQUIRED
width	CDATA	#IMPLIED
direction	(l tor)	#FIXED "l tor"

```
>
```

Сущности

- Общие сущности
 - `<!ENTITY Имя "значение">`
 - `<!ENTITY Имя СсылкаНаФайл>`
- Сущности-параметры
 - `<!ENTITY % Имя "значение">`
 - `<!ENTITY % Имя СсылкаНаФайл>`
- Примеры
 - `<!ENTITY copyright "Georgiy Korneev">`
 - `<!ENTITY file SYSTEM "books.xml">`
 - `<!ENTITY % statements "(if|while|for)">`

Применение общих сущностей

- Сокращения

```
<!ENTITY copyright "© Georgiy Korneev 2005">
```

```
<p>Copyright: &copyright;</p>
```

- Разбиение XML на файлы

```
<!ENTITY books SYSTEM "books.xml">
```

```
<!ENTITY magazines SYSTEM "magazines.xml">
```

```
<library>
```

```
    &books; &magazines;
```

```
</library>
```

Применение сущностей-параметров

- Структура программы

```
<!ENTITY % statements "if | while | for">
```

```
<!ELEMENT while (%statements;)*>
```

```
<!ELEMENT for (%statements;)*>
```

```
<!ELEMENT if (then, else?)*>
```

```
<!ELEMENT then (%statements;)*>
```

```
<!ELEMENT else (%statements;)*>
```

- Разбиение DTD на файлы

```
<!ENTITY % books SYSTEM "books.dtd">
```

```
%books;
```

DTD и пространства имен

- В DTD можно указывать название элементов и атрибутов с двоеточиями
- Пространства имен задаются с помощью fixed-атрибутов

Пример: DTD и пространства имен

- Библиотека

```
<!ELEMENT lib:book (lib:author)*>
```

```
<!ATTLIST lib:book
```

```
  xmlns:lib          CDATA          #FIXED
```

```
    "http://www.example.com/library"
```

```
  caption            CDATA          #REQUIRED
```

```
  isbn               CDATA          #IMPLIED
```

```
>
```

Часть 2

XML Schema

XML Schema

- XML Schema – язык описания структуры XML-документов
- Описание является XML-документом
- XML Schema предназначена для описания документов с пространствами имен
- XML Schema позволяет выразить все то же, что и DTD
- Пространство имен
<http://www.w3.org/2001/XMLSchema>

Типы

- Описывают множество значений
- Простые типы (`simpleType`)
 - Строка символов
 - Атрибуты, текст
- Сложные типы (`complexType`)
 - Элементы и текст
 - Элементы

Простые типы

- Стандартные
- Ограниченные (**restricted**)
 - Ограничение другого типа
- Списочные (**list**)
 - Список элементов другого типа
- Объединение (**union**)
 - Объединение простых типов

Стандартные типы (1)

- Строковые
 - `string`, `normalizedString`, `token`
- Неограниченные целые
 - `integer`, `positiveInteger`, `nonPositiveInteger`, `negativeInteger`, `nonNegativeInteger`
- Ограниченные целые
 - `(unsigned)Byte`, `(unsigned)Short`, `(unsigned)Int`, `(unsigned)Long`
- Дробные
 - `decimal`, `float`, `double`

Стандартные типы (2)

- Дата и время
 - `dateTime`, `date`, `time`, `duration`
- Имена
 - `name`, `QName`, `NCName`
- Из DTD
 - `ID`, `IDREF`, `NMTOKEN`
- Специальные
 - `boolean`, `anyURI`, `language`

Объявление ограниченного типа

- Синтаксис

```
<xsd:simpleType name="Имя">
```

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

```
    Ограничения
```

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

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

Пример: объявление типа

- Серийный номер

```
<xsd:simpleType name="serial">  
  <xsd:restriction base="xsd:integer">  
    <xsd:minInclusive value="10000"/>  
    <xsd:maxInclusive value="99999"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

- Телефон

```
<xsd:simpleType name="phone">  
  <xsd:restriction base="xsd:string">  
    <xsd:pattern value="\d{3}-\d{2}-\d{2}"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

Объявление списочного типа

- Синтаксис

```
<xsd:simpleType name="Имя">  
  <xsd:list itemType="БазовыйТип">  
</xsd:simpleType>
```

- Примеры

```
<xsd:simpleType name="serialList">  
  <xsd:list itemType="serial"/>  
</xsd:simpleType>  
<xsd:simpleType name="phoneList">  
  <xsd:list itemType="phone"/>  
</xsd:simpleType>
```

Объявление типа-объединения

- Синтаксис

```
<xsd:simpleType name="ИмяТипа">  
  <xsd:union memberTypes="СПИСОК ТИПОВ"/>  
</xsd:simpleType>
```

- Пример

```
<xsd:simpleType name="phoneSerial">  
  <xsd:union memberTypes="serial phone"/>  
</xsd:simpleType>
```

Объявление сложного типа

- Синтаксис

```
<xsd:complexType  
  name = "ИмяТипа "  
  mixed = "boolean"
```

>

ОписаниеСодержимого

ОписаниеАтрибута*

```
</xsd:complexType>
```

Описание атрибута

- Синтаксис

<xsd:attribute

name = "Имя"

type = "ИмяТипа"

default = "ЗначениеПоУмолчанию"

fixed = "ЗначениеПоУмолчанию"

use = "optional | prohibited | required"

ref = "ИмяАтрибута"

>ОбъявлениеПростогоТипа</xsd:attribute>

- Пример

```
<xsd:attribute name="serial" type="serial"  
  use="required"/>
```


Описание содержимого

- Синтаксис
 - Описание содержимого
 - `<xsd:choice>Содержимое</xsd:choice>`
 - `<xsd:sequence>Содержимое</xsd:sequence>`
 - `<xsd:any/>`
 - `<xsd:all>Содержимое</xsd:all>`
 - Содержимое
 - ОписаниеЭлемента
 - ОписаниеГруппы

Описание элемента

- Синтаксис

<element

name = "Имя"

type = "ИмяТипа"

minOccurs = "число"

maxOccurs = "число | **unbounded**"

ref = "ИмяЭлемента"

/>ОбъявлениеТипа**</element>**

- Пример

```
<xsd:element name="address" type="address"
  maxOccurs="unbounded"/>
```

Описание групп

- Синтаксис

```
<group
  name      = "Имя"
  ref       = "ИмяГруппы"
  maxOccurs = "число | unbounded"
  minOccurs = "число"
>ОписаниеСодержимого</group>
```

- Пример

```
<xsd:group name="contact"><xsd:choice>
  <xsd:element name="phone" type="phone"/>
  <xsd:element name="address" type="address"/>
</xsd:choice></xsd:group>
```

Пример: описание сложного типа

- Адрес

```
<xsd:complexType name="address">  
  <xsd:sequence>  
    <xsd:element name="city" type="xsd:string"/>  
    <xsd:element name="street" type="xsd:string"/>  
    <xsd:element name="building" type="xsd:string"/>  
    <xsd:element name="office" type="xsd:integer"/>  
  </xsd:sequence>  
</xsd:complexType>
```

Описание структуры

- Синтаксис

```
<xsd:schema
```

```
    targetNamespace      = "NS"
```

```
    elementFormDefault  = "(un)qualified"
```

```
    attributeFormDefault = "(un)qualified"
```

```
>
```

Описание Типов, Атрибутов, Элементов

```
</xsd:schema>
```

Пример: пространства имен

```
<xsd:schema targetNamespace="ns"
  elementFormDefault="unqualified"
  attributeFormDefault="unqualified"
>
  <xsd:element name="contact" type="ns:contact"/>
  <xsd:complexType name="contact"><xsd:choice>
    <xsd:element name="address" type="ns:address"/>
    <xsd:element name="phone" type="ns:phone"/>
  </xsd:choice></xsd:complexType>
  <xsd:simpleType name="phone">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="\d{3}-\d{2}-\d{2}"/>
    </xsd:restriction>
  </xsd:simpleType>
  ...
</xsd:schema>
```

Пример: пространства имен

- `elementFormDefault="unqualified"`
`<contact>`
 `<address>St. Petersburg, ...</address>`
`</contact>`
- `elementFormDefault="qualified"`
`<ns:contact`
 `xmlns:ns="ns"`
`>`
 `<ns:phone>123-45-67</ns:phone>`
`</ns:contact>`

Разбиение на файлы

- Возможно включения одних схем в другие

- Синтаксис

```
<xsd:include
  namespace      = "URI"
  schemaLocation = "URL"
/>
```

- Пример

```
<xsd:schema targetNamespace="namespace">
  <xsd:include
    namespace      ="contact"
    schemaLocation ="file:/contact.xsd"
  />
</xsd:schema>
```


Аннотации

- Определение типов, элементов, атрибутов могут содержать аннотации
- Синтаксис
 - **<annotation>**Описания**</annotation>**
 - Описания
 - **<document>**текст**</document>**
 - **<appinfo>**текст**</appinfo>**

Связывание документа со схемой

- Пример

```
<contact
```

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

```
  xsi:schemaLocation="contact.xsd"
```

```
>
```

```
  ...
```

```
</contact>
```

Часть 3

Проверка структуры XML

Проверка по DTD

- Свойство `DocumentBuilderFactory`
 - `validating` – производить ли проверку
- Метод `DocumentBuilder`
 - `isValidating()` – производится ли проверка
- Свойство `SAXParserFactory`
 - `validating` – производить ли проверку
- Свойство `SAXParser`
 - `isValidating()` – производится ли проверка

Проверка по XML Schema

- Пакет `javax.xml.validation`
- Класс `Schema`
- Установка схемы
 - Свойство `DocumentBuilderFactory.schema`
 - Свойство `SAXParserFactory.schema`

Создание Schema

- Класс `SchemaFactory`
- Методы
 - `newInstance(type)` – создание фабрики
 - `newSchema(...)` – создание экземпляра схемы
 - `setFeature(uri, value)` – установить настройку
- Типы схем
 - Класс `XMLConstants`
 - `W3C_XML_SCHEMA_NS_URI`
 - `http://www.w3.org/2001/XMLSchema`

Пример: задание схемы

```
SchemaFactory sfactory =  
    SchemaFactory.newInstance(  
        XMLConstants.W3C_XML_SCHEMA_NS_URI);  
Schema schema = sfactory.newSchema(  
    new File("books.xsd"));
```

```
SAXParserFactory factory =  
    SAXParserFactory.newInstance();  
factory.setSchema(schema);  
factory.setNamespaceAware(true);  
factory.setValidating(true);  
SAXParser parser = factory.newSAXParser();
```

Часть 6

Заключение

Ссылки

- Extensible Markup Language 1.1 // <http://www.w3.org/TR/2004/REC-xml11-20040204/>
- XML Schema Part 0: Primer // <http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/>
- XML Schema Part 1: Structures // <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>
- XML Schema Part 2: Datatypes // <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>

Вопросы