

Технологии Java

XML Transformations 1.0 (XSLT 1.0)

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

Содержание

1. Введение
2. Пути
3. Выражения
4. Функции
5. Заключение

Часть 1

Введение

XML Transformations

- Язык преобразования XML-документов
 - XML
 - HTML
 - Текст
- Основан на XPath

XSL и XSLT

- Extensible Stylesheet Language (XSL)
 - Правила отображения
 - Произвольный XML
 - Произвольный носитель
- Этапы отображения



Объявление преобразования

- `<xsl:stylesheet`
 `id = id`
 `extension-element-prefixes = tokens`
 `exclude-result-prefixes = tokens`
 `version = number>`
- `<xsl:transform>`
- Пространство имен
 - <http://www.w3.org/1999/XSL/Transform>

Обработка пробелов

- Удаление последовательностей пробелов
 - `<xsl:strip-spaces elements=names />`
- Сохранение последовательностей пробелов
 - `<xsl:preserve-spaces elements=names />`
- Умолчания
 - Исходный документ – сохранять
 - Преобразование – удалять (кроме `xsl:text`)

Часть 2

Шаблоны

Правила преобразования

- Узлы, соответствующих условию `match` заменяются содержимым шаблона
- `<xsl:template`
 - `match = pattern`
 - `priority = number`
 - `mode = qname>`
- Примеры
 - `<xsl:template match="a[@title = ``]"/>`
 - `<xsl:template match="*/>`
 - `<xsl:template match="text()"/>`

Применение преобразований

- Применить преобразования к выбранным узлам
 - `<xsl:apply-templates`
 select = node-set-expression
 mode = qname>

Выбор применяемого правила

- Совпадение режимов
- Правила с максимальным приоритетом
 - * -0.25
 - `text()` -0.5
- В порядке объявления (не обязательно)

Композиция преобразований

- Включение преобразования
 - `<xsl:include href=uri />`
- Импортирование преобразования
 - `<xsl:import href=uri />`
 - Уменьшает приоритет шаблонов

Правила по умолчанию

- Рекурсивное применение к элементам с сохранением режима
- Копирование текста и атрибутов
- Игнорирование инструкций по обработке и комментариев

Именованные шаблоны

- Определение

```
<xsl:template name=qname>
```

```
  [<xsl:param name=qname select=default/>]*
```

- Вызов

```
<xsl:call-template name=name>
```

```
  <xsl:with-param name=qname>
```

```
    value
```

```
  </xsl:with-template>
```

Часть 3

Содержание шаблона

Создание элементов и атрибутов

- Создает элемент

```
<xsl:element  
  name = {qname}  
  namespace = {uri-reference}>
```

- Создает атрибут

```
<xsl:attribute  
  name = {qname}  
  namespace = {uri-reference}>
```


Простое создание элементов

- Элемент
 - ``
- Вычисление выражений в атрибутах
 - ``

Создание текста

- Создает текстовый узел

```
<xsl:text disable-output-escaping=yesno>  
  text</xsl:text>
```

- Вычисление выражений

```
<xsl:value-of  
  select=value  
  disable-output-escaping=yesno  
>
```

- Вывод чисел `xsl:number`

Копирование

- Копирование имени элемента
 - `<xsl:copy>`
- Копирование набора узлов
 - `<xsl:copy-of select=nodeset/>`

Обход набора узлов

- Применяет правило для узлов из набора

```
<xsl:for-each select=nodeset>
```

```
  <xsl:sort
```

```
    select = string-expression
```

```
    data-type = { "text" | "number" | ... }
```

```
    order = { "ascending" | "descending" }
```

```
  />
```

УСЛОВИЯ

- Простое условие

```
<xsl:if test=expression>
```

- Выбор

```
<xsl:choose>
```

```
  <xsl:when test=expression>...</xsl:when>
```

```
  <xsl:when test=expression>...</xsl:when>
```

```
  <xsl:otherwise>...</xsl:otherwise>
```

```
</xsl:choose>
```

Переменные

- Неизменяемые переменные
- Простая переменная
 - `<xsl:variable name=name select=value/>`
- Переменная с шаблоном
 - `<xsl:variable name=name>...</xsl:variable>`
- Параметры преобразования
 - `<xsl:param name=name select=value/>`

Часть 4

Дополнительные ВОЗМОЖНОСТИ

Дополнительные функции

- Текущий узел
 - `current()`
- Генерация уникального идентификатора
 - `generate-id(nodeset)`
- Загрузка документа
 - `document(uri)`
- Форматирование числа
 - `format-number(number, format)`

Ключи

- Определение ключа
 - `<xsl:key`
 - `name = qname`
 - `match = pattern`
 - `use = expression />`
- Использование ключа
 - `key(string, query)`

Вывод

- Инструкция по выводу
 - `<xsl:output`
 - `method = "xml" | "html" | "text"`
 - `encoding = string`
 - `omit-xml-declaration = "yes" | "no"`
 - `doctype-public = string`
 - `doctype-system = string`
 - `/>`

Часть 5

Заключение

- XSL Transformations (XSLT) Version 1.0
// <http://www.w3.org/TR/1999/REC-xslt-19991116>
- Перевод спецификации // <http://www.rol.ru/news/it/helpdesk/xslt01.htm>
- Валиков А. Технология XSLT

Вопросы