

Технологии Java

XML Path Language 1.0 (XPath 1.0)

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

Содержание

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

Часть 1

Введение

XPath

- Язык выбора узлов XML-документов
- XPath рассматривает XML-документ в виде дерева
- Встраиваемый язык
 - XSLT
 - XPather

- Предоставляется внешними средствами
 - Текущий узел (current node)
 - Набор переменных (variable bindings)
 - Библиотека функций (function library)
 - Набор префиксов пространств имен (set of namespace declarations)

Типы данных

- `node-set` – набор узлов
- `boolean` – логический
- `number` – число с плавающей точкой
- `string` – строка (unicode)

Основные конструкции

- **Location path** – пути
 - Выбор набора узлов
- **Expressions** – выражения
 - Вычисления над наборами узлов
- **Functions** – функции
 - Произвольные функции

Часть 2

Location paths

Location Path

- Путь состоит из последовательности шагов
- Типы путей
 - Абсолютный
 - Относительный
- Абсолютный – вычисляется относительно корня документа
 - $('/' | '// ') step (('/' | '// ') step)^* | '/'$
- Относительный – вычисляется относительно текущего узла
 - $step (('/' | '// ') step)^*$

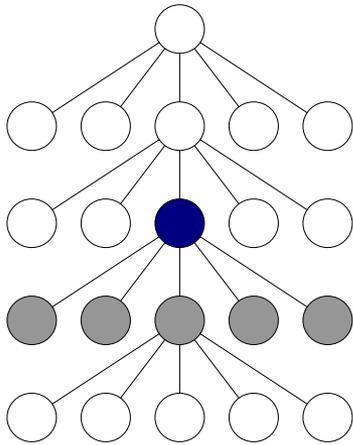
Правила работы

1. Проинициализировать множество текущих узлов (МТУ) текущим узлом (для относительного пути) либо корнем (для абсолютного пути)
2. Для каждого узла из МТУ отложить один шаг и положить результат в множество следующих узлов (МСУ)
3. $\text{МТУ} := \text{МСУ}$
4. Если путь не кончился, перейти к шагу 2
5. Выдать МТУ в качестве результата

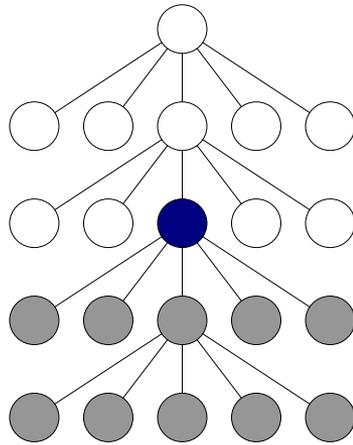
Location step

- Основные части
 - **Axis** (ось) – определяет какие узлы будут рассматриваться с точки зрения текущего узла
 - **Node test** (проверка узла) – типы или имена узлов
 - **Predicates** (предикаты) – условия на узлы (не обязательно)
- Синтаксис
 - *axis '::' node-test ('[' predicate ''])**
- Пример
 - `child::p[position() = 1]`

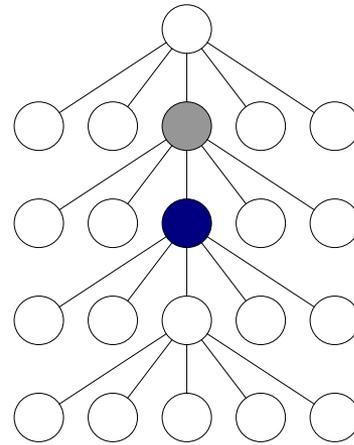
Типы осей (1)



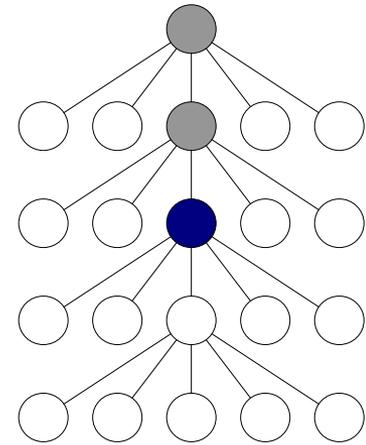
childs



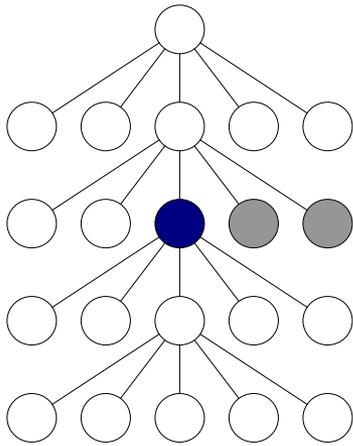
descendant



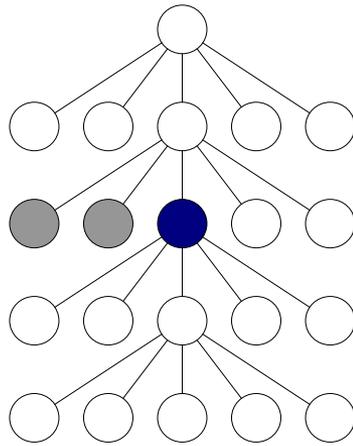
parent



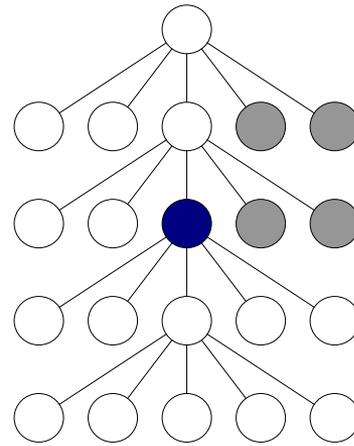
ancestors



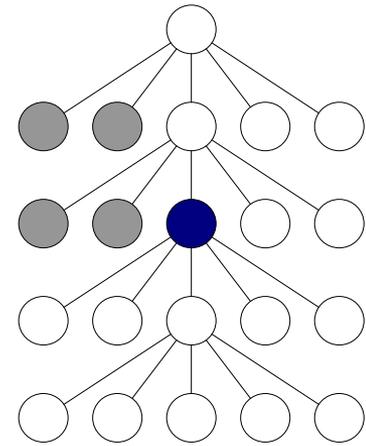
following-sibling



preceding-sibling

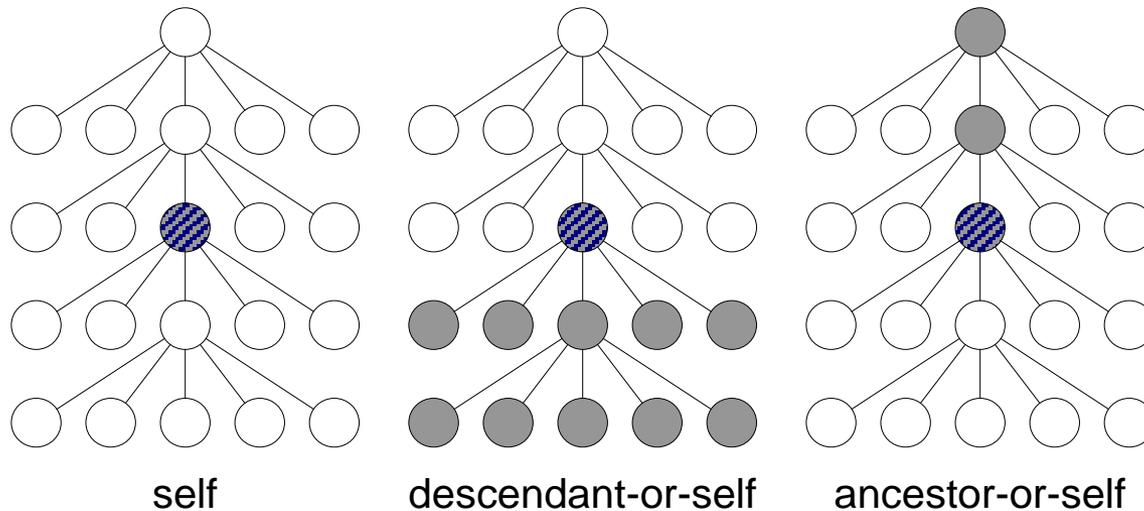


following



preceding

Типы осей (2)



- **attribute** – ось атрибутов
- **namespace** – ось префиксов (не используется)

Node tests

- Каждая ось имеет основной тип узла
 - `attribute` – атрибут
 - `namespace` – пространство имен
 - Остальные – элемент
- Синтаксис
 - `node-name`
 - `*`
 - `('text()' | 'node()' | 'processing-instruction()' | 'comment()')`
 - `'processing-instruction(имя)'`

Примеры

- `child::p` – дети, имеющие имя `p`
- `ancestors::*` – все предки
- `attribute::href` – атрибут `href`
- `attribute::*` – все атрибуты
- `descendants::text()` – все текстовые узлы

Сокращенный синтаксис

- Ось по умолчанию – `children`
- `@` – префикс атрибута
- `[n]` – предикат, выбирающий n-й элемент
- `.` – текущий узел
- `..` – родитель
- `//step` – потомки узла
- Пример `./p/@warning` – сокращение для
 - `self::node()/descendant-or-self::node()/child::p/attribute::warning`

Predicates

- Предикат – произвольное условие на текущий узел
 - Узел берется, если предикат выполняется
- Может быть несколько предикатов
- Примеры
 - `a[@href = "help.html"]`
 - `a[@href = "help.html"][5]`
 - `a[5][@href = "help.html"]`
 - `a[img[@alt]]`
 - `a[@href and @title]`

Часть 3

Выражения

Основные выражения

- `$var` – ссылка на переменную
- `(expr)` – выражение в скобках
- `"literal"` – строка
- `-10` – число
- `id(arg1, arg2, ...)` – вызов функции

Выбор узлов

- | – объединение путей
 - a | img
- /, // – конструирование путей
 - a/img
- Фильтры
 - Основное выражение
 - Фильтр + предикаты

Логические выражения

- Операции
 - `or` – логическое или
 - `and` – логическое и
 - `=`, `!=` – сравнение на равенство (неравенство)
 - `<`, `>`, `<=`, `>=` – сравнения
- Наборы узлов
 - Пустой `false`
 - Не пустой `true`

Сравнения

- При сравнении учитывается тип значений
 - Строки – лексикографически
 - Числа – по величине
- При сравнении двух наборов узлов выражение истинно, если найдется по элементу в первом множестве и втором множестве, связанные отношением
- Значение считается набором узлов из одного узла

Численные выражения

Операции

- +, - – сложение и вычитание
- *, div, mod – умножение, деление, взятие остатка
- - – унарный минус
- Основные выражения

Часть 4

Стандартные функции

Над множествами узлов (1)

- Местоположение
 - `position()` – номер текущего узла в наборе
 - `last()` – номер последнего узла в наборе
 - `count(node-set)` – количество узлов

Над множествами узлов (2)

- Имена
 - `name(node-set?)` – имя текущего / первого в наборе узла
 - `local-name(node-set?)` – локальное имя узла
 - `namespace-uri(node-set?)` – имя пространства имен
- Выбор
 - `node-set id(object)` – выбор элементов по идентификатору

Функции над строками (1)

- `string(object)` – приведение к строке
- `concat(str1, str2, ...)` – конкатенация строк
- `starts-with(s1, s2)` – `s1` начинается с `s2`
- `contains(s1, s2)` – `s1` содержит `s2`
- `string-length(string?)` – длина строки
- `normalize-space(string?)` – нормализация пробелов в строке

Функции над строками (2)

- Подстроки
 - `substring-before(s1, s2)` – часть `s1` до первого вхождения `s2`
 - `substring-after(s1, s2)` – часть `s1` после первого вхождения `s2`
 - `substring(s, off, len?)` – подстрока `s`, начиная с `off`, длиной `len` СИМВОЛОВ
- `translate(s, from, to)` – для каждого СИМВОЛА из `s`: если он содержится во `from`, то заменить его на соответствующий СИМВОЛ из `to`

Логические функции

- `boolean(object)` – преобразование к логическому типу
- `not(boolean)` – логическое отрицание
- `true()` – истина
- `false()` – ложь
- `lang(name)` – проверка языка узла (атрибут `xml:lang`)

Числовые функции

- `number(object?)` – преобразование к числу
- `sum(node-set)` – сумма наборов узлов
- `round(number)` – округление к ближайшему
- `floor(number)` – округление вниз
- `ceiling(number)` – округление вверх

Часть 5

Заключение

- XML Path Language (XPath) version 1.0
// <http://www.w3.org/TR/1999/REC-xpath-19991116>
- Перевод спецификации на русский // <http://www.rol.ru/news/it/helpdesk/xpath01.htm>
- Валиков А. Технология XSLT

Вопросы