

**Бази даних та інформаційні системи**

**ЛАБОРАТОРНА РОБОТА №6**

**Мова виразів XPath**

Виконав:

Ст. Прізвище Ім'я

Група .....

**Тема:** Мова виразів XPath.

**Мета роботи:** Вивчення мови виразів XPath та створення виразів XPath для ефективного використання XSLT та XQuery.

### Теоретичний матеріал:

Мова XPath визначена W3C наступними документами:

- Документ [XML Path Language \(XPath\) 3.1](#) W3C Recommendation 21 March 2017, що визначає специфікацію мови XPath, яка дозволяє обробляти значення, що відповідають моделі даних, визначеній у [XQuery and XPath Data Model \(XDM\) 3.1](#).
- Документ [XQuery and XPath Functions and Operators 3.1](#). W3C Recommendation 21 March 2017, що визначає, крім іншого, функції та оператори на вузлах та послідовностях вузлів.

Назва мови XPath походить від її найбільш відмінної риси, виразу шляху, який забезпечує засіб ієрархічної адресації вузлів у XML-дереві. Основна мета XPath - звернення до вузлів дерев XML, іншими словами для навігації по елементах та атрибутах у документах XML.

Результатом виразу XPath може бути набір вузлів (node-set) з вхідних документів, атомарні значення, такі як цілі числа, рядки (string) та булеві (boolean) true чи false.

- XPath використовує компактний не-XML-синтаксис (щоб полегшити використання XPath в межах значень URI та атрибутів XML).
- XPath отримав свою назву завдяки використанню позначень шляху, як в URL-адресах для навігації по ієрархічній структурі XML-документа.
- XPath містить понад 200 вбудованих функцій.
- XPath був визначений одночасно з XSLT (1999 рік).
- Спочатку XPath було розроблено для підтримки XSLT та XPointer (мова вказівника XML, що використовується для XLink, XInclude тощо). Сьогодні XPath також використовує XQuery. Взагалі, будь-який вираз, який є синтаксично дійсним і успішно виконується в XPath 3.1 та XQuery 3.1, поверне однаковий результат в обох мовах. Є кілька винятків із цього правила:
  - оскільки XQuery розширює заздалегідь визначені посилання на сутності та посилання на символи, а XPath цього не робить, вирази, що містять їх, дають різні результати на двох мовах. Наприклад, значення літерального рядка "&"; є & в XQuery, і & в XPath;
  - якщо ввімкнено режим сумісності XPath 1.0, XPath поводить себе по-різному від XQuery в кількох випадках, які перелічені в [H.3.2 Incompatibilities when Compatibility Mode is false](#).
- Система типів XPath 3.1 заснована на XML-схемі. Це визначається реалізацією, чи базується система типів на основі [\[XML Schema 1.0\]](#) або [\[XML Schema 1.1\]](#).

### 1. Основні поняття

Основним будівельним блоком XPath є вираз, який є рядком символів. Мова містить декілька видів виразів, які можуть бути побудовані з ключових слів, символів та операндів. Взагалі операнди виразу - це інші вирази. XPath дозволяє виразам вкладатись із повною загальністю.

Як і XML, XPath є чутливою до регістру мовою. Ключові слова в XPath використовують малі символи та незарезервовані, тобто імена в виразах XPath можуть бути такими ж, як і ключові слова мови, за винятком наступних нефіксованих імен функцій:

*array, attribute, comment, document-node, element, empty-sequence, function, if, item, map, namespace-node, node, processing-instruction, schema-attribute, schema-element, switch, text, typeswitch.*

Хоча ключові слова *switch* і *typeswitch* не використовуються в XPath, вони вважаються зарезервованими іменами функцій для сумісності з XQuery.

### **1.1. Порядок документа**

Порядок, що називається порядком документа, визначається усіма вузлами, доступними під час обробки заданого виразу, які можуть складатися з одного або декількох дерев (документів або фрагментів). Стабільний порядок документа означає, що відносний порядок двох вузлів не зміниться під час обробки заданого виразу.

У дереві порядок документування задовольняє наступним обмеженням:

- Кореневий вузол - це перший вузол.
- Кожен вузол є перед усіма його дітьми та нащадками.
- Вузли простору імен відразу слідує за вузлом елемента, з яким вони асоціюються.
- Вузли атрибутів відразу слідує за вузлами простору імен елемента вузла, з яким вони асоціюються.
- Відносний порядок братів і сестер - це порядок, в якому вони відбуваються у *children* властивості свого батьківського вузла.
- Діти та нащадки трапляються до наступних братів і сестер.

Відносний порядок вузлів у різних деревах є стабільним, але залежним від реалізації, за умови наступного обмеження: якщо будь-який вузол у даному дереві T1 знаходиться перед будь-яким вузлом у іншому дереві T2, то всі вузли у дереві T1 знаходяться перед усіма вузлами в дереві T2.

В моделі даних значення завжди є послідовністю. Послідовність являє собою упорядкований набір з нуля або більше елементів.

Елемент є або атомарне значення, вузол або функція.

Вузол є екземпляром одного з семи видів вузлів: *елемент, атрибут, текст, простір імен, інструкція по обробці, коментар та документ*.

Для прикладу розглянемо наступний документ XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book>
    <title lang="en">Harry Potter</title>
    <price>29.99</price>
  </book>
  <book>
    <title lang="en">Learning XML</title>
    <price>39.95</price>
  </book>
</bookstore>
```

Приклади вузлів у зазначеному документі XML:

<code>&lt;bookstore&gt;</code>	(вузол кореневий елемент)
<code>&lt;price&gt;29.99&lt;/price&gt;</code>	(вузол елемент)
<code>lang="en"</code>	(вузол атрибут)

Атомарні значення - це вузли, у яких немає дітей чи батьків.

Наприклад: 29.99, "en"

## 1.2. Взаємозв'язок вузлів

- Батьківський (Parent) Кожен елемент та атрибут мають одного батьківського вузла. У наведеному вище прикладі елемент book - батько елемента title та price.
- Діти (Children) Вузли елементів можуть мати нуль, одного або декількох дітей: title та price є дітьми елемента book.
- Брати і сестри Це вузли, які мають одного з батьків. У прикладі це назва та ціна.
- Предки Батьківський вузол, батьківський батько тощо. У прикладі предками елемента title є елемент book та елемент bookstore.
- Нащадки Діти вузла, діти дітей тощо. У прикладі нащадками елемента bookstore є елементи book, title та price

## 2. Синтаксис XPath

### 2.1. Вибір вузлів

XPath використовує вирази шляху для вибору вузлів у документі XML. Вузол вибирається шляхом слідування шляху або кроків. Нижче наведено найбільш корисні вирази шляху у короткій формі позначення:

*nodename* Вираз шляху, що складається лише з однієї назви вузла (напр. *nodename*) дозволяє вибрати всі вузли з цим іменем

*/* на початку виразу шляху - це аббревіатура для початкового кроку Значення цього початкового кроку в тому, що шлях повинен розпочатися в кореновому вузлі дерева, яке містить контекстний вузол.

*//* на початку виразу шляху означає встановлення початкового вузла послідовності яка містить корінь дерева, в якому знайдено контекстний вузол, плюс усі вузли, що походять від цього кореня. Ця послідовність вузлів використовується як вхід до наступних кроків у виразі шляху. Повне позначення *descendant-or-self::node()*

*.* Вибирає поточний вузол; повне позначення *self::node()*

*..* Вибір батька поточного вузла (на один крок назад/вверх); *parent::node()*

*@* Вибір вузла атрибута; *attribute::*

Вираз *child::* може опускатися, для NAME, але не для COUNTRY *child::COUNTRY/child::NAME*

Наступні вирази ідентичні:

*/child::book/child::author/child::name/attribute::lastName*

*/book/author/name/@lastName*

Корінь документа завжди є контекстом за замовчуванням. Контекст - це поточний вузол або набір вузлів, щодо яких розраховується наступний крок.

Нижче наведено деякі вирази шляху та результат виразів:

<i>bookstore</i>	Вибирає всі вузли з назвою bookstore
<i>/bookstore</i>	Вибирає кореневий елемент bookstore. Примітка. Якщо шлях починається з косої риски (/), він завжди представляє абсолютний шлях до елементу!
<i>bookstore/book</i>	Вибирає всі елементи book, які є дітьми елемента bookstore
<i>//book</i>	Вибирає всі елементи book незалежно від того, де вони знаходяться в документі
<i>bookstore//book</i>	Вибирає всі елементи book, які є нащадком елемента bookstore, незалежно від того, де вони знаходяться під елементом bookstore
<i>//@lang</i>	Вибирає всі атрибути, названі lang

Вираз шляху, який *починається* з " /" або " //", вибирає вузли, починаючи з *кореня дерева*, що містить елемент контексту; його часто називають *абсолютним* виразом шляху.

*Відносний* вираз шляху - це вираз шляху, який вибирає вузли в дереві, виконуючи ряд кроків, починаючи з *контекстного вузла* (який, на відміну від абсолютного виразу шляху, може бути будь-яким вузлом у дереві).

Кожне *непочаткове* виникнення "/" у виразі шляху фактично замінюється під час обробки виразу шляху. Наприклад, bookstore//book це скорочений термін, child::bookstore /descendant-or-self::node()/child::book і так вибирають усіх book нащадків bookstore дітей, залишаючи послідовність кроків, розділених на "/". Але, вираз шляху //book [1] зовсім НЕ означає те ж саме, як вираз шляху /descendant::book [1]. Останній вибирає перший нащадковий book елемент; перший вибирає всі book елементи нащадків, які є першими book дітьми відповідних батьків.

Наступний приклад ілюструє використання відносних виразів шляху.

child::bookstore /child::book

Вибирається дочірній елемент book дочірнього елемента bookstore елемента вузла контексту; тобто book - онука елемента контекстного вузла, який має bookstore батька.

Приклади:

<b>child::book</b>	Вибирає всі вузли book, які є дітьми поточного вузла
<b>attribute::lang</b>	Вибирає атрибут lang поточного вузла
<b>child::*</b>	Вибирає всі дочірні елементи поточного вузла
<b>attribute::*</b>	Вибирає всі атрибути поточного вузла
<b>child::text()</b>	Вибирає всі текстові вузли дітей поточного вузла
<b>child::node()</b>	Вибирає всіх дітей поточного вузла
<b>descendant::book</b>	Вибирає всіх нащадків book поточного вузла
<b>ancestor::book</b>	Вибирає всіх предків book поточного вузла
<b>ancestor-or-self::book</b>	Вибирає всіх предків book поточного вузла - і поточний, також якщо це book вузол
<b>child::* / child::price</b>	Вибирає всіх price онуків поточного вузла

В обох випадках шлях розташування складається з одного або декількох кроків, кожен розділений косою рисою:

*Абсолютний шлях розташування:* /step/step/...

*Відносний шлях розташування:* step/step/...

Кожен крок оцінюється по відношенню до вузлів у поточному наборі вузлів.

Крок складається з:

- вісь (визначає залежність дерева між вибраними вузлами та поточним вузлом)
- тест вузла (ідентифікує вузол у межах осі)
- нуль або більше предикатів (для подальшого уточнення вибраного набору вузлів)

Синтаксис кроку місцезнаходження:

axisname::nodetest[predicate]

## 2.2. Предикати

Предикати використовуються для пошуку конкретного вузла або вузла, який містить конкретне значення. Предикати завжди вбудовані в квадратні дужки. Наприклад:

/bookstore/book[1] Вибирає перший елемент book, який є дочірнім елементом bookstore. Примітка: У IE 5,6,7,8,9 перший вузол є [0], але згідно W3C,

це [1]. Щоб вирішити цю проблему в IE, встановіть *SelectionLanguage* на *XPath: In JavaScript: xml.set Property("SelectionLanguage","XPath");*

`/bookstore/book[last()]` Вибирає останній елемент `book`, який є дочірнім елементом `bookstore`

`/bookstore/book[last()-1]` Вибирає передостанній елемент `book`, який є дочірнім елементом `bookstore`

`/bookstore/book[position()<3]` Вибирає перші два елементи `book`, які є дітьми `bookstore`

`//title[@lang]` Вибирає всі елементи `title`, які мають атрибут з назвою `lang`

`//title[@lang='en']` Вибирає всі елементи `title`, які мають атрибут `"lang"` зі значенням `"en"`

`/bookstore/book[price>35.00]` Вибирає всі елементи `book` елемента `bookstore`, які мають `price` елемент, значення якого перевищує 35,00

`/bookstore/book[price>35.00]/title` Вибирає всі `title` елементи `book` елементів `bookstore`, які мають `price` елемент, значення якого перевищує 35,00

### 2.3. Вибір невідомих вузлів

Підстановочні символи XPath можна використовувати для вибору невідомих XML-вузлів.

\* відповідає будь-якому вузлу елемента

@\* відповідає будь-якому вузлу атрибута

node() відповідає будь-якому вузлу будь-якого типу

Наприклад:

`/bookstore/*` вибирає всі дочірні вузли елемента `bookstore`

`//*` вибирає всі елементи в документі

`//title[@*]` вибирає всі `title` елементи, які мають принаймні один атрибут будь-якого типу

### 2.4. Вибір декількох шляхів

За допомогою | оператора в виразі XPath можна вибрати кілька шляхів.

`//book/title | //book/price` Вибирає всі `title` ТА `price` елементи всіх елементів `book`

`//title | //price` Вибирає всі елементи `title` ТА `price` в документі

`/bookstore/book/title | //price` Вибирає всі `title` елементи `book` елемента `bookstore` елемента і всі `price` елементи в документі.

### 2.5. Оці XPath

Вісь представляє відношення до контекстного (поточного) вузла і використовується для пошуку вузлів відносно цього вузла на дереві.

AxisName	Result
ancestor	Вибирає всіх предків (батьків, бабусь і дідусів тощо) поточного вузла
ancestor-or-self	Вибирає всіх предків поточного вузла та самого поточного вузла
attribute	Вибирає всі атрибути поточного вузла
child	Вибирає всіх дітей поточного вузла
descendant	Вибирає всіх нащадків (дітей, онуків тощо) поточного вузла
descendant-or-self	Вибирає всіх нащадків поточного вузла та самого поточного вузла
following	Вибирає все в документі після закриття тегу поточного вузла
following-sibling	Вибирає всіх братів і сестер після поточного вузла

namespace	Вибирає всі вузли простору імен поточного вузла
parent	Вибирає батька поточного вузла
preceding	Вибирає всі вузли, що з'являються перед поточним вузлом у документі, крім предків, атрибутивних вузлів та вузлів простору імен
preceding-sibling	Вибирає всіх братів і сестер перед поточним вузлом
self	Вибирає поточний вузол

Для осей, що найчастіше використовуються, існують скорочення:

attribute:: - можна замінити на "@"  
 child:: - часто просто опускають  
 descendant:: - можна замінити на "///"  
 parent: - можна замінити на «..»  
 self:: - можна замінити на "."

Посилання для ознайомлення з теоретичним матеріалом теми «Мова виразів XPath»:

- Beginning XML, 5-edition © 2012 Joe Fawcett, Liam R.E. Quin, Danny Ayers
- XPath Tutorial W3C: [https://www.w3schools.com/xml/xpath\\_intro.asp](https://www.w3schools.com/xml/xpath_intro.asp)

Перелік понять, з якими необхідно ознайомитись для виконання лабораторної роботи:

1. Основні поняття мови виразів XPath
  - 1.1. Порядок документа
  - 1.2. Взаємозв'язок вузлів
2. Синтаксис XPath
  - 2.1. Вибір вузлів
  - 2.2. Предикати
  - 2.3. Вибір невідомих вузлів
  - 2.4. Вибір декількох шляхів
  - 2.5. Осі XPath
  - 2.6. Функції XPath

### Хід роботи

1. Опрацювати теоретичний матеріал.
2. Написати декілька виразів шляху XPath у відповідності до свого XML документа (обов'язково використати функції, наприклад count, sum, avg чи інші).
3. Використати on-line [XPath Tester / Evaluator](#) для перевірки коректності створених виразів шляху.
4. Оформити звіт про виконання лабораторної роботи, який має містити:
  - титульну сторінку;
  - тему, мету та завдання лабораторної роботи;
  - короткий перелік та опис створених виразів шляху XPath;
  - навести скріни екрану з кодом та результатом їх перевірки XPath Tester.
5. Завантажити в канал “БД. Лабораторна робота” своєї команди в Teams.