

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА ФРАНКА

Факультет прикладної математики та інформатики

Бази даних та інформаційні системи

ЛАБОРАТОРНА РОБОТА №14

Тема: «Мова виразів XPath».

Виконала:

Ст. Пелещак Вероніка

ПМІ-35с

Тема: «Мова виразів XPath».

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

Завдання: Написати 5-6 виразів шляху XPath у відповідності до свого XML документа (обов'язково використати функції, наприклад count, sum, avg чи інші).

Контрольні питання:

1. Що означає назва мови XPath?

XPath (XML Path Language) — це мова, яка використовується для **навігації по елементах XML-документа** за допомогою шляхів, подібно до того, як ми використовуємо шляхи в файловій системі.

2. Що таке порядок документа і яким обмеженням він має задовільняти?

Порядок документа — це **послідовність, у якій вузли з'являються в XML**.
Обмеження:

- кореневий елемент і його нащадки мають бути впорядковані за тим, як вони записані в документі;
- не може бути “стрибків” — кожен вузол з'являється лише після свого попередника;
- порядок визначає, що “раніше” і “пізніше” при навігації по дереву XML.

3. Який взаємозв'язок вузлів XML документа?

XML утворює **дерево вузлів**:

- один **кореневий вузол**;

- кожен вузол може мати **батька**, **дітей** і **сусідів**;
- типи зв'язків: *parent* (батько), *child* (дитина), *sibling* (сусід), *ancestor* (предок), *descendant* (нащадок).

4. Які вирази шляху у короткій формі використовуються для вибору вузлів?

Короткі (скорочені) вирази:

- / — від кореня (абсолютний шлях);
- // — усі вузли, незалежно від рівня;
- . — поточний вузол;
- .. — батьківський вузол;
- @ — атрибут.

5. Що таке абсолютний і відносний вирази шляху?

- **Абсолютний** починається з / — тобто від кореня.

Наприклад: /bug_tracking_system/software/bug

- **Відносний** починається з поточного вузла, без /.

Наприклад: bug/description — відносно вузла software.

6. Наведіть приклад предикатів для пошуку конкретного вузла або вузла, який містить конкретне значення.

- //bug[@id=5] — знайти баг із id = 5
- //bug[name='Несправність інтерфейсу'] — знайти вузол, де name має конкретне значення.

7. Наведіть приклад вибору невідомих XML-вузлів.

Якщо наперед не знаєте назву елемента — можна використати *:

- `//software/*` — усі дочірні елементи `software`;
- `//@*` — усі атрибути в документі.

8. Як у виразі XPath можна вибрати кілька шляхів?

За допомогою символу `|` (об'єднання шляхів): `//bug/name | //bug/priority`. Це вибере `name`, і `priority` для кожного багу.

Хід роботи:

1. Підраховує кількість унікальних прізвищ користувачів, які створили баги:

Option 1: Copy-paste your XML here

```
<bug_tracking_system>
<software id="1">
  <name>TaskMaster</name>
  <version>1.0.0</version>
  <category>Utility</category>
```

Option 2: Or upload your XML file

File encoding

Choose File

no file selected

UTF-8



XPath expression

`count(distinct-values(//bug/created_by/last_name))`



Include 'XML Item Type' in output

-XPath Result-

Int64='6'

2. Обчислює середню кількість багів на одне програмне забезпечення:

Option 1: Copy-paste your XML here

```
<bug_tracking_system>
<software id="1">
  <name>TaskMaster</name>
  <version>1.0.0</version>
  <category>Utility</category>
</software>
</bug_tracking_system>
```

Option 2: Or upload your XML file

Choose File no file selected

File encoding UTF-8

XPath expression

avg(for \$s in //software return count(\$s/bug))

☒ Include 'XML Item Type' in output

-XPath Result-

Decimal='1.6666666666666667'

3. Для кожного елемента <software> виводить його назву та кількість багів:

Option 1: Copy-paste your XML here

```
<bug_tracking_system>
<software id="1">
  <name>TaskMaster</name>
  <version>1.0.0</version>
  <category>Utility</category>
</software>
</bug_tracking_system>
```

Option 2: Or upload your XML file

Choose File no file selected

File encoding UTF-8

XPath expression

for \$s in //software return concat(\$s/name, ': ', count(\$s/bug))

☒ Include 'XML Item Type' in output

-XPath Result-

```
String='TaskMaster: 7'  
String='EpicQuest: 1'  
String='FocusFlow: 1'  
String='ChatLink: 1'  
String='SocialSphere: 1'  
String='PixelCraft: 1'  
String='Quantum Query: 1'  
String='Echo Wave: 1'  
String='DeepCanvas: 1'
```

4. Вибирає імена тих багів, де опис (<description>) має довжину більше ніж 55 символів:

Option 1: Copy-paste your XML here

```
<bug_tracking_system>  
<software id="1">  
  <name>TaskMaster</name>  
  <version>1.0.0</version>  
  <category>Utility</category>
```

Option 2: Or upload your XML file

File encoding

Choose File no file selected

UTF-8 ▼

XPath expression

```
//bug[string-length(description) > 55]//name
```

☒ Include 'XML Item Type' in output

-XPath Result-

```
Element='<name>Некоректне відображення таблиці</name>'  
Element='<name>Система не зберігає налаштування</name>'  
Element='<name>Нерівний розмір шрифту</name>'  
Element='<name>Помилка збереження</name>'  
Element='<name>Обрізаний експорт аудіо</name>'
```

5. Знаходити максимальну кількість виправлень серед усіх багів:

Option 1: Copy-paste your XML here

```
<bug_tracking_system>
  <software id="1">
    <name>TaskMaster</name>
    <version>1.0.0</version>
    <category>Utility</category>
```

Option 2: Or upload your XML file

File encoding

Choose File

no file selected

UTF-8



XPath expression

max(//bug/count(history_corrections/correction))



Include

-XPath Result-

Int64='2'

6. Виводить імена всіх багів, у яких пріоритет High або Critical:

Option 1: Copy-paste your XML here

```
<bug_tracking_system>
  <software id="1">
    <name>TaskMaster</name>
    <version>1.0.0</version>
    <category>Utility</category>
```

Option 2: Or upload your XML file

File encoding

Choose File

no file selected

UTF-8



XPath expression

//bug[priority='High' or priority='Critical']/name



Include

-XPath Result-

```
Element='<name>Новий баг з високим пріоритетом</name>'  
Element='<name>Новий баг з високим пріоритетом</name>'  
Element='<name>Некоректне відображення таблиці</name>'  
Element='<name>Помилка в системі аутентифікації</name>'  
Element='<name>Помилка при запуску програми</name>'  
Element='<name>Помилка збереження</name>'  
Element='<name>Обрізаний експорт аудіо</name>'
```

Висновок: Вивчення XPath допомагає легко знаходити і обробляти потрібні дані в XML. Використання різних виразів і функцій робить роботу з XSLT та XQuery швидшою і зручнішою.