

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА ФРАНКА

Факультет прикладної математики та інформатики

Бази даних та інформаційні системи

ЛАБОРАТОРНА РОБОТА №15

Тема: «XSLT — мова перетворення XML-документів».

Виконала:

Ст. Пелещак Вероніка

ПМІ-35с

Тема: «XSLT — мова перетворення XML-документів».

Мета роботи: Ознайомлення з синтаксисом мови XSLT та перетворенням фрагмента XML документа з використанням XSLT.

Завдання: Створити шаблон перетворення XSLT фрагмента свого XML документа з обов'язковим використанням `<value-of>`, `<for-each>`, `<sort >`, `<if>/<choose>`, `<number>`. Використати довільний XSLT processor для перевірки коректності роботи створеного шаблону перетворення.

Контрольні питання:

1. Для чого призначена мова XSLT?

XSLT (Extensible Stylesheet Language Transformations) використовується для перетворення XML-документів у інші формати: HTML, текстові файли або інший XML. Це дозволяє змінювати структуру даних, порядок елементів або зовнішній вигляд без зміни самого XML.

2. За допомогою якого механізму досягається перетворення XML документу?

Перетворення виконує XSLT-процесор. Він читає XML та відповідний XSLT-шаблон і генерує вихідний документ. Механізм працює так: процесор знаходить вузли, що підходять під шаблони, і вставляє у вихідний документ визначене у шаблоні представлення.

3. Яка структура декларації стилю?

Основою є `<xsl:stylesheet>`. Вона містить:

- атрибут `version` (версія XSLT),
- атрибут `xmlns:xsl` (простір імен XSLT),

- опції для виводу (наприклад, метод виводу — HTML або текст),
- набори шаблонів (<xsl:template>), які описують, як обробляти вузли XML.

4. Що таке шаблон перетворення і з яких частин він складається?

Шаблон — це правило, яке визначає, як обробляти конкретні вузли XML.

Складається з двох основних частин:

- match — вираження, яке визначає, які вузли підпадають під шаблон,
- тіло шаблону — набір елементів XSLT, які описують, що робити з вибраними вузлами (наприклад, виводити їх значення, застосовувати інші шаблони, форматувати).

5. Як отримати значення вибраного вузла та додати його до вихідного потоку перетворення?

Для цього використовується <xsl:value-of>. Цей елемент бере текстове значення обраного вузла і вставляє його у вихідний документ у потрібному місці. Це дозволяє відображати дані з XML у HTML або тексті.

6. Який оператор використовується для вибору кожного XML-елемента певного набору вузлів?

Використовується <xsl:for-each>. Він проходить по всіх вузлах заданого набору і для кожного застосовує певне перетворення. Це схоже на цикл у програмуванні.

7. Як можна відсортувати та пронумерувати результати виводу?

- Для сортування застосовується <xsl:sort> всередині <xsl:for-each> або <xsl:apply-templates>, що дозволяє виводити дані у потрібному порядку (за алфавітом, числом, датою).

- Для нумерації використовується `<xsl:number>`, який автоматично генерує порядковий номер елемента у наборі вузлів, незалежно від його ID у XML.

8. Наведіть приклад встановлення умови виводу вмісту файлу XML.

Умови встановлюються через `<xsl:if test="умова">`. Вона перевіряє, чи виконується певна умова для вузла, і лише тоді виводить його значення.

Наприклад:

- `<xsl:if test="priority='High'">`

Ця умова перевіряє, чи пріоритет багу High, і лише тоді виводить його назву або інші дані.

9. Як здійснити вивід за однією з кількох умов?

Для цього використовується конструкція `<xsl:choose>`, яка включає:

- `<xsl:when>` — для перевірки кожної умови,
- `<xsl:otherwise>` — для всіх випадків, що не підходять під жодну умову.

Це дозволяє обробляти різні значення XML і змінювати вихідний документ залежно від умов.

10. Який елемент застосовує правило шаблону до поточного елемента або дочірніх вузлів поточного елемента?

Це `<xsl:apply-templates>`. Він застосовує відповідний шаблон до поточного вузла або його дочірніх елементів. Таким чином реалізується рекурсивне перетворення XML, де кожен елемент обробляється згідно зі своїм шаблоном.

Хід роботи:

1. Було створено шаблон перетворення XSLT для фрагмента XML-документа з лабораторної роботи №11.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:output method="html" indent="yes" encoding="UTF-8"/>

  <xsl:template match="/">
    <html>
      <head>
        <style>
          table {
            border-collapse: collapse;
            width: 100%;
            font-family: Arial, sans-serif;
          }
          th, td {
            border: 1px solid #ccc;
            padding: 8px;
            text-align: left;
          }
          th {
            background-color: #f0f0f0;
            font-weight: bold;
          }

          tr.critical { background-color: #ff9999; }
          tr.high { background-color: #ffcccb; }
          tr.medium { background-color: #fff5cc; }
          tr.low { background-color: #ccffcc; }

          tr { font-weight: normal; }

          ul { font-family: "Courier New", monospace; margin-top: 20px; }
          li { margin-bottom: 8px; }
        </style>
      </head>
```

```

<body>
  <h2>Bug Report Table</h2>
  <table>
    <tr>
      <th>#</th>
      <th>Software</th>
      <th>Bug Name</th>
      <th>Priority</th>
      <th>Status</th>
      <th>Module</th>
      <th>Date</th>
      <th>Created By</th>
      <th>Assigned To</th>
    </tr>

    <xsl:for-each select="bug_tracking_system/software">
      <xsl:sort select="@id" data-type="number" order="ascending"/>
      <xsl:for-each select="bug">
        <xsl:sort select="@date" data-type="text" order="ascending"/>
        <tr>
          <xsl:attribute name="class">
            <xsl:choose>
              <xsl:when test="priority='Critical'">critical</xsl:when>
              <xsl:when test="priority='High'">high</xsl:when>
              <xsl:when test="priority='Medium'">medium</xsl:when>
              <xsl:otherwise>low</xsl:otherwise>
            </xsl:choose>
          </xsl:attribute>
          <td>
            <xsl:number level="any" count="software" from="bug_tracking_system" format="1"/>
          </td>
          <td><xsl:value-of select="../name"/></td>
          <td><xsl:value-of select="name"/></td>
          <td><xsl:value-of select="priority"/></td>
          <td><xsl:value-of select="status"/></td>
          <td><xsl:value-of select="module"/></td>

```

```

          <td><xsl:value-of select="@date"/></td>
          <td><xsl:value-of select="concat(created_by/first_name, ' ', created_by/last_name)"/></td>
          <td><xsl:value-of select="concat(assigned_to/first_name, ' ', assigned_to/last_name)"/></td>
        </tr>
      </xsl:for-each>
    </xsl:for-each>
  </table>

  <h2>Corrections List</h2>
  <ul>
    <xsl:for-each select="//correction">
      <xsl:sort select="@date" data-type="text" order="descending"/>
      <li>
        <strong>Дата:</strong> <xsl:value-of select="@date"/> <br/>
        <strong>Коментар:</strong> <xsl:value-of select="comment"/> <br/>
        <strong>Виправив:</strong>
        <xsl:value-of select="concat(corrected_by/first_name, ' ', corrected_by/last_name)"/>
        <em> (<xsl:value-of select="corrected_by/email"/>)</em> <br/>
        <strong>Software:</strong> <xsl:value-of select="ancestor::bug/../name"/>
      </li>
    </xsl:for-each>
  </ul>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

2. Для перевірки коректності створеного XSLT-шаблону перетворення використано XSLT-процесор *freeformatter.com*. Результат отримано та збережено з розширенням *.html*.

Option 1: Copy-paste your XML document 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

Choose File no file selected

File encoding
UTF-8

Option 1: Copy-paste your XSL document here (Optional if XSD referred in XML using schemaLocation)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:output method="html" indent="yes" encoding="UTF-8"/>
```

Option 2: Or upload your XSL document

Choose File no file selected

File encoding
UTF-8

Transform XML

Transform XML to new window

[XSLT online course](#)

-Transformed Document-

```
<html>
<head>
  <style>
    table {
      border-collapse: collapse;
      width: 100%;
      font-family: Arial, sans-serif;
    }
    th, td {
      border: 1px solid #ccc;
      padding: 8px;
      text-align: left;
    }
    th {
      background-color: #f0f0f0;
      font-weight: bold;
    }
    tr.critical { background-color: #ff9999; }
```

[Copy](#)[Save](#)

Результат:

Bug Report Table

#	Software	Bug Name	Priority	Status	Module	Date	Created By	Assigned To
1	TaskMaster	Несправність інтерфейсу	Low	In Progress	UI	2024-11-02	Olena Bilous	Noah Davis
1	TaskMaster	Новий баг	Low	New	UI	2025-03-18	Ivan Sokolov	Maria Kozak
1	TaskMaster	Некоректне відображення таблиці	High	New	Reports	2025-09-21	Ivan Sokolov	Maria Kozak
1	TaskMaster	Новий баг з високим пріоритетом	High	New	UI	2025-09-22	Ivan Sokolov	Maria Kozak
1	TaskMaster	Bug from T1	Medium	New	Core	2025-09-22	Ivan Sokolov	Maria Kozak
1	TaskMaster	Bug from T1	Medium	New	Core	2025-09-22	Ivan Sokolov	Maria Kozak
1	TaskMaster	Новий баг з високим пріоритетом	High	New	UI	2025-09-23	Ivan Sokolov	Maria Kozak
2	EpicQuest	Помилка в системі аутентифікації	Critical	New	Authentication	2025-01-26	Ivan Sokolov	Dmytro Melnyk
3	FocusFlow	Помилка при запуску програми	Critical	Fixed	Core	2025-03-01	Maria Kozak	Olena Bilous
4	ChatLink	Система не зберігає налаштування	Medium	Fixed	Settings	2024-08-27	Olena Bilous	Dmytro Melnyk
5	SocialSphere	Нерівний розмір шрифту	Low	New	UI	2025-02-21	Maria Kozak	Anna Petrovna
6	PixelCraft	Помилка збереження	High	In Progress	Export	2025-04-12	Liam Johnson	Liam Johnson
7	Quantum Query	Некоректний пошук	Medium	Fixed	Search	2025-01-13	Maria Kozak	Noah Davis
8	Echo Wave	Обрізаний експорт аудіо	High	In Progress	Export	2025-03-14	Olivia Brown	Ivan Sokolov
9	DeepCanvas	Проблема з пензлем	Medium	New	UI	2025-04-11	Dmytro Melnyk	Dmytro Melnyk

Corrections List

- Дата:** 2025-09-21
Коментар: Баг виправлено: додано обробку винятків при запуску.
Виправив: Ivan Sokolov (ivansokolov88@gmail.com)
Software: FocusFlow
- Дата:** 2025-04-15
Коментар: Зміни на сервері дозволяють зберігати файли без пошкоджень, додано перевірку цілісності.
Виправив: Liam Johnson (liam.j@gmail.com)
Software: PixelCraft
- Дата:** 2025-04-12
Коментар: Покращено обробку пензля для зміни розміру, тепер пензель адаптується до розмірів екрану.
Виправив: Dmytro Melnyk (dmytromelnuk2000@gmail.com)
Software: DeepCanvas
- Дата:** 2025-03-19
Коментар: Додано перевірку коректності збереження даних перед експортом.
Виправив: Ivan Sokolov (ivansokolov88@gmail.com)
Software: Echo Wave
- Дата:** 2025-03-03
Коментар: Система аутентифікації була вдосконалена для кращої безпеки.
Виправив: Olena Bilous (olenabilous@gmail.com)
Software: SocialSphere
- Дата:** 2025-03-01
Коментар: Шрифт було адаптовано для різних розмірів екранів.
Виправив: Maria Kozak (mariakozak@gmail.com)
Software: TaskMaster
- Дата:** 2025-02-09
Коментар: Додано можливість скидання пароля при аутентифікації через Google.
Виправив: Ivan Sokolov (ivansokolov88@gmail.com)
Software: EpicQuest
- Дата:** 2025-01-22
Коментар: Виправлено помилку пошуку, тепер результати повертаються коректно.
Виправив: Emma Stone (emma.stone@gmail.com)
Software: Quantum Query
- Дата:** 2024-11-15
Коментар: Змінено дизайн кнопок на головній сторінці.
Виправив: Ivan Sokolov (ivansokolov88@gmail.com)
Software: FocusFlow

Висновок: У ході виконання лабораторної роботи було створено шаблон перетворення XSLT для фрагмента XML-документа, який дозволяє відображати вибрані дані у вигляді таблиці та списку з використанням форматування, виділення рядків і колонок, сортування та нумерації. Для реалізації роботи застосовано основні конструкції XSLT: `<xsl:value-of>`, `<xsl:for-each>`, `<xsl:sort>`, `<xsl:if>/<xsl:choose>`, `<xsl:number>`. Перевірку коректності роботи шаблону виконано за допомогою XSLT-процесора (freeformatter.com), результат збережено у форматі HTML та відкрито в браузері, що підтвердило правильність відображення даних відповідно до поставлених умов.