

Chương 3 : Truy xuất tài liệu XML với DOM

I. Mô hình đối tượng DOM

1) Mô hình đối tượng

Mục tiêu :

Ôn tập, hệ thống hóa các kiến thức về mô hình đối tượng

=== > Giúp sinh viên tự tìm hiểu và ứng dụng nhanh một công nghệ cụ thể

=== > Mở đầu cho việc trình bày công nghệ DOM

a) Đối tượng

+ **Biến** : Vùng nhớ trong bộ nhớ chính cho phép biểu diễn các thông tin thực tế bên trong phần mềm

+ **Cấu trúc dữ liệu** : Một loại biến đặc biệt với các đặc điểm sau

- Vùng nhớ được cấu trúc bởi (bao gồm bên trong) các "vùng nhớ con"

- Cho phép biểu diễn trọn vẹn/tích hợp các thông tin của một đối tượng trong thực tế

+ **Đối tượng** : Một loại cấu trúc dữ liệu đặc biệt với các đặc điểm sau

- Chỉ cho phép một số hàm (Hàm thành phần) truy xuất đến cấu trúc của các "vùng nhớ con " (Biến thành phần)

- Việc gọi thực hiện các hàm thành phần theo cú pháp đặc biệt

- Cho phép định nghĩa các đối tượng mới dựa trên định nghĩa của các đối tượng đã có

- Cho phép biểu diễn dưới dạng tự nhiên nhất thông tin và các xử lý liên quan một đối tượng trong thực tế

+ **Phân loại đối tượng**

Có rất nhiều cách phân loại đối tượng dựa trên các tiêu chí khác nhau.

Nếu dựa trên tiêu chí về "Nguồn gốc tạo lập" có thể chia các đối tượng thành 2 loại

- Đối tượng thư viện : Các đối tượng "có sẵn" trong các môi trường lập trình

- Đối tượng tự định nghĩa: Các đối tượng do người phát triển phần mềm tự định nghĩa (thiết kế) và thực hiện (lập trình)

b) Đối tượng & Xây dựng phần mềm

Mô hình đối tượng

Hệ thống các đối tượng cho phép biểu diễn các đối tượng trong thực tế

Mô hình đối tượng thư viện :

Hệ thống các đối tượng thư viện cho phép biểu diễn các đối tượng trong thực tế

Mô hình đối tượng tự định nghĩa :

Hệ thống các đối tượng tự định nghĩa cho phép biểu diễn các đối tượng trong thực tế

==== > Nghiên cứu và ứng dụng công nghệ X : Tìm hiểu về mô hình đối tượng thư viện của công nghệ X

Xây dựng ứng dụng phần mềm theo công nghệ X

Sử dụng các đối tượng thư viện của công nghệ X trực tiếp bên trong phần mềm
hay

Xây dựng mô hình đối tượng tự định nghĩa dựa vào mô hình đối tượng thư viện của công nghệ X

c) Mô hình đối tượng dữ liệu

- + **Đối tượng dữ liệu:** Một loại đối tượng đặc biệt cho phép biểu diễn các thông tin đã lưu trữ theo một công nghệ lưu trữ nào đó.
- + **Mô hình đối tượng dữ liệu:** Hệ thống đối tượng cho phép biểu diễn toàn bộ các thông tin đã lưu trữ theo một công nghệ lưu trữ nào đó.
- + **Mô hình đối tượng dữ liệu thư viện:** Hệ thống đối tượng thư viện cho phép biểu diễn toàn bộ các thông tin đã lưu trữ theo một công nghệ lưu trữ nào đó

Ví dụ :

Các mô hình đối tượng dữ liệu thư viện liên quan công nghệ lưu trữ dùng CSDL quan hệ:

ADO (Visual Studio 6.0)
ADO.NET (Visual Studio.NET)
JDO (JDK)

Mô hình đối tượng dữ liệu thư viện liên quan công nghệ lưu trữ dùng XML:

DOM XML

Mô hình đối tượng dữ liệu thư viện liên quan công nghệ lưu trữ dùng Microsoft Office:

Word
Excel

d) Mô hình đối tượng thể hiện

- + **Đối tượng thể hiện:** Một loại đối tượng đặc biệt cho phép biểu diễn các thông tin dưới dạng xem được theo một công nghệ giao diện người dùng
- + **Mô hình đối tượng thể hiện:** Hệ thống đối tượng đặc biệt cho phép biểu diễn toàn bộ các thông tin dưới dạng xem được theo một công nghệ giao diện người dùng nào đó
- + **Mô hình đối tượng thể hiện thư viện:**

Hệ thống đối tượng thư viện cho phép biểu diễn các thông tin dưới dạng xem được theo một công nghệ giao diện người dùng

Ví dụ :

Các mô hình đối tượng thể hiện thư viện liên quan công nghệ giao diện người dùng trên Windows

Form, TreeView, DataGrid, ... (Visual Studio 6.0)
Form, DataGridView , ... (Visual Studio.NET)
JFrame, JTree, JTextBox , ... (JBuilder 10.0, NetBeans, Eclipse)

Mô hình đối tượng thể hiện thư viện trên Web

DOM HTML

2) Khái niệm về DOM

Mục tiêu :Giới thiệu khái niệm cơ bản về DOM (Document Object Model)

- + DOM dưới góc nhìn người xây dựng ứng dụng trên môi trường cụ thể:

DOM là một mô hình đối tượng dữ liệu thư viện cho phép biểu diễn thông tin và các xử lý liên quan một tài liệu XML trong bộ nhớ chính

Ví dụ :

Với các lập trình viên trên môi trường Visual Studio.NET

DOM là thư viện các đối tượng XmlDocument, XmlElement,....các đối tượng này khi cần sử dụng cần phải khai báo theo cú pháp và tên đặt cụ thể của ngôn ngữ đang dùng

using System.Xml ; với C#

Imports System.Xml với VB.NET

Với các lập trình viên trên môi trường JBuilder 10.0

DOM là thư viện các đối tượng XmlDocument, XmlElement ,các đối tượng này khi cần sử dụng cần phải khai báo theo cú pháp cụ thể

```
import javax.xml.parser.*;
```

```
import org.w3c.dom.* ;
```

```
import org.w3c.dom.Node.* ;
```

+ DOM dưới góc nhìn người phát triển thư viện của môi trường phát triển phần mềm

DOM là định chuẩn ràng buộc trên kiến trúc của các mô hình đối tượng thư viện dữ liệu được sử dụng trong các môi trường phát triển phần mềm

=== > Tất cả các mô hình đối tượng thư viện DOM trong các môi trường phát triển phần mềm khác nhau

- Thống nhất về cách sử dụng (Tên , cách gọi hàm, tham số,)

- Chỉ khác biệt nhau về thuật giải xử lý bên trong

Với góc nhìn này DOM chính là hệ thống giao diện lập trình (Interface) mà các đối tượng thư viện phải chấp nhận.

3) Hệ thống các đối tượng DOM

Mục tiêu : Trình bày khái niệm về đối tượng của DOM

Nội dung : Bao gồm 2 phần

Phần 1 : Ôn tập về mô hình đối tượng dữ liệu quan hệ

== > Giúp sinh viên có thể tìm hiểu và sử dụng nhanh các đối tượng thư viện liên quan công nghệ lưu trữ CSDL

== > Mở đầu cho phần trình bày về mô hình DOM

Phần 2 : Trình bày các đối tượng chính của DOM

a) Các đối tượng dữ liệu quan hệ

*** Tổ chức lưu trữ của công nghệ CSDL**

CSDL :

Bao gồm nhiều bảng dữ liệu

Bảng dữ liệu :

- Bao gồm nhiều dòng dữ liệu

- Tất cả các dòng đều có chung cấu trúc theo các cột của bảng

- Cột khóa chính là cột cho phép xác định duy nhất dòng trong bảng và được sử dụng liên kết các dòng của các bảng khác nhau

Dòng dữ liệu :

Lưu trữ các thông tin trong thực tế

*** Các đối tượng chính của mô hình đối tượng dữ liệu quan hệ**

Đối tượng CSDL: Bao gồm nhiều đối tượng bảng dữ liệu

Đối tượng bảng dữ liệu

- Bao gồm nhiều đối tượng dòng dữ liệu
 - Tất cả các đối tượng dòng đều có chung cấu trúc theo các đối tượng cột của đối tượng bảng
 - Đối tượng cột khóa chính là đối tượng cột cho phép xác định duy nhất đối tượng dòng trong đối tượng bảng và được sử dụng liên kết các đối tượng dòng của các đối tượng bảng khác nhau
- đối tượng dòng dữ liệu :

Lưu trữ các thông tin trong thực tế

*** Các đối tượng chính của mô hình đối tượng thư viện ADO.NET**

+ Đối tượng CSDL DataSet

Bao gồm nhiều đối tượng bảng dữ liệu DataTable

+ Đối tượng bảng dữ liệu DataTable

- Bao gồm nhiều đối tượng dòng dữ liệu DataRow
- Tất cả các DataRow đều có chung cấu trúc theo các đối tượng cột DataColumn của DataTable
- Đối tượng cột khóa chính là đối tượng cột cho phép xác định duy nhất đối tượng dòng trong đối tượng bảng và được sử dụng liên kết các đối tượng dòng của các đối tượng bảng khác nhau

+ Đối tượng dòng dữ liệu : DataRow

Lưu trữ các thông tin trong thực tế

b) Các đối tượng dữ liệu XML

*** Tổ chức lưu trữ của công nghệ lưu trữ XML**

+ Tài liệu XML :

Bao gồm nhiều thẻ dữ liệu

Bắt buộc phải chứa duy nhất một thẻ gốc

+ Thẻ dữ liệu :

- Có thể bao gồm nhiều thuộc tính
- Có thể bao gồm nhiều thẻ dữ liệu
- Có thể lưu trữ hay không lưu trữ thông tin trong thực tế

+ Thuộc tính:

Lưu trữ các thông tin trong thực tế

*** Các đối tượng chính của mô hình đối tượng dữ liệu XML**

+ Đối tượng tài liệu XML XmlDocument

Bao gồm nhiều đối tượng thẻ dữ liệu XmlElement

Cho phép sử dụng thẻ gốc qua đối tượng DocumentElement

+ Đối tượng thẻ dữ liệu XmlElement

- Bao gồm nhiều đối tượng thuộc tính XmlAttribute
- Bao gồm nhiều đối tượng thẻ con XmlElement
- Có thể có hay không có đối tượng giá trị XmlValue

+ Đối tượng thuộc tính XmlAttribute : Lưu trữ thông tin trong thực tế

Ghi chú :

- Ngoài các đối tượng chính và thông dụng trên, DOM bao hàm các loại đối tượng khác (ít thông dụng hơn) tương ứng với các loại thẻ khác nhau của tài liệu XML như XmlCDataSection, XmlEntity, ... Các đối tượng này cho phép truy xuất thông tin tương ứng với các loại thẻ khác nhau của tài liệu XML
- Tất cả các đối tượng của DOM đều chấp nhận giao diện chung XmlNode tương ứng với hệ thống các hàm xử lý cơ bản trên mọi thẻ của tài liệu XML

*** XmlNode :** Giao diện XmlNode

Hàm	Ý nghĩa	Ghi chú
nodeType	Trả về loại nút	
nodeName	Trả về tên nút	
nodeValue	Trả về giá trị tương ứng	Sử dụng chủ yếu với loại thuộc tính hay nội dung thẻ không có thẻ con
selectSingleNode	Trả về một nút (Node) theo dựa trên chuỗi truy vấn Xpath	
selectNodes	Trả về một tập hợp nút (NodeList) theo dựa trên chuỗi truy vấn Xpath	
childNodes	Trả về tập hợp các nút con (NodeList) của nút đang xét	
parentNode	Trả về nút cha (Node) của nút đang xét	
ownerDocument	Trả về tài liệu (Document) chứa nút đang xét	
appendChild	Bổ sung nút con (Node) vào nút đang xét	
removeChild	Xóa nút con của nút đang xét	
replaceChild	Thay thế một nút con này bằng nút con khác của nút đang xét	
cloneNode	Tạo bản sao của nút đang xét	Sử dụng tham số để quyết định tạo “bản sao cạn” (không xét các nút con) hay “bản sao sâu” (ngược lại)

*** XmlDocument:** Đối tượngXmlDocument

Hàm	Ý nghĩa	Ghi chú
createElement	Trả về nút (Element) với tên nút	
createAttribute	Trả về thuộc tính (Attribute) với tên thuộc tính	
getElementsByTagName	Trả về tập hợp các nút con (NodeList) theo tên trong tham số	Sử dụng cho mọi loại tài liệu có cấu trúc
Load	Tiếp nhận dữ liệu từ nguồn nào đó (tập tin là thông dụng nhất)	Sử dụng cho mọi loại tài liệu có cấu trúc
LoadXML	Tiếp nhận dữ liệu từ chuỗi có cấu trúc tài liệu XML	
Save	Kết xuất dữ liệu đến nguồn nào đó (tập tin là thông dụng nhất)	

*** XmlElement:** Đối tượngXmlElement

Hàm	Ý nghĩa	Ghi chú
getAttribute	Trả về giá trị của thuộc tính có tên trong tham số	
getAttributeNode	Trả về đối tượng thuộc tính (Attr) với tên trong tham số	
setAttribute	Gán giá trị cho thuộc tính có tên trong tham số	Tạo thuộc tính mới nếu chưa có
removeAttribute	Xóa thuộc tính có tên trong tham số	
removeAttributeNode	Xóa đối tượng thuộc tính (Attr) có trong tham số	
getElementsByTagName	Trả về tập hợp các nút con (NodeList) theo tên trong tham số	

II. Một số kỹ thuật lập trình DOM

1) Đọc tài liệu XML

Vấn đề : Cần đọc tài liệu XML trên bộ nhớ phụ vào tạo đối tượng XmlDocument tương ứng

Hướng giải quyết :

Xây dựng hàm đọc tài liệu với

Tham số :đường dẫn của tập tin Xml tương ứng

Kết quả : đối tượng XmlDocument

Ghi chú : để có thể tái sử dụng hàm trên có thể tạo hàm đọc tài liệu

- Bên trong một lớp đối tượng (với khai báo là hàm mức lớp)

Lập trình với C#

```
public XmlDocument Doc_tai_lieu(String duongdan)
{
    XmlDocument doc=null;
    try
    {
        doc.Load(duongdan);
    }
    catch (Exception ex)
    {
        Console.WriteLine("Loi doc tap tin:" + ex.Message);//MessageBox.Show();
    }
    return doc;
}
```

* Bài tập

Mục tiêu :

- Tìm hiểu và sử dụng một số hàm của DOM

Yêu cầu :

Viết chương trình nhập vào đường dẫn của một tập tin Xml và sau đó lần lượt thực hiện các công việc sau

a) Xuất thông tin thống kê

Tổng số các thẻ của tập tin XML

Tổng số thuộc tính của tập tin XML

b) Nhập vào chuỗi Ten và cho biết có bao nhiêu thẻ trong tập tin có Ten tương ứng

2) Ghi tài liệu XML

Vấn đề :

Cần ghi đối tượng XmlDocument vào tập tin trên bộ nhớ phụ

Hướng giải quyết :

Xây dựng hàm ghi tài liệu với

Tham số : đối tượng XmlDocument , đường dẫn của tập tin Xml tương ứng

Kết quả : Giá trị logic

```
public Boolean Ghi_tai_lieu(XmlDocument tai_lieu, String duongdan)
{
    try
    {
        tai_lieu.Save(duongdan);
    }
    catch (Exception ex)
    {
        Console.WriteLine("Loi ghi tap tin tai lieu:" + ex.Message);//MessageBox.Show();
        return false;
    }
    return true;
}
```

- Bài tập:

Mục tiêu :

- Tìm hiểu và sử dụng một số hàm của DOM

Yêu cầu :

Viết chương trình nhập vào đường dẫn tương ứng tập tin Xml và sau đó tạo tập tin (ghi) có tên Nut_la.xml bao gồm tất cả các thẻ không chứa thẻ con của tập tin đang xét

3) Đọc đối tượng từ tập tin XML

Vấn đề :

Cần đọc dữ liệu của đối tượng x thuộc loại X tương ứng với thẻ X trong tập tin XML

- Đọc dữ liệu của đối tượng phân số Ps trong tập tin Phan_so.xml
- Đọc dữ liệu của đối tượng điểm Tam trong tập tin Duong_tron.xml
- Đọc dữ liệu của đối tượng loại phòng trong tập tin Bang_don_gia.xml
- Đọc dữ liệu của đối tượng ngoại tệ trong tập tin Bang_ty_gia.xml

Nhận xét :

Cách đọc và tạo đối tượng x tương ứng thẻ X phụ thuộc vào X có phải là thẻ gốc hay không

Nếu X là thẻ gốc === > Đọc trực tiếp

Nếu X là thẻ con của thẻ gốc == > Phải đọc thẻ gốc và sau đó khởi tạo x từ đối tượng XmlElement tương ứng

Hướng giải quyết với X là thẻ gốc

Xây dựng hàm đọc (hàm mức lớp) của lớp đối tượng XL_X

Tham số : Đường dẫn của tập tin Xml tương ứng

Kết quả : đối tượng x thuộc lớp XL_X

Thuật giải

Khai báo đối tượng Kq

Tai_lieu = đối tượng XmlDocument đọc từ tập tin với Duong_dan

Goc = đối tượng XmlElement tương ứng gốc của Tai_lieu

Gán các biến thành phần của Kq tương ứng các thuộc tính của Goc

Trả Kq

Hướng giải quyết với X là thẻ con Xây dựng hàm khởi tạo (hàm mức lớp) của lớp đối tượng XL_X

Tham số : đối tượng XmlElement Nut

Kết quả : đối tượng x thuộc lớp XL_X

Thuật giải

Khai báo đối tượng Kq

Gán các biến thành phần của Kq tương ứng các thuộc tính của Nut

Trả Kq

4) Ghi đối tượng vào tập tin XML

Vấn đề :

Cần ghi dữ liệu của đối tượng x thuộc loại X vào với thẻ X trong tập tin XML

- Ghi dữ liệu của đối tượng phân số Ps trong tập tin Phan_so.xml
- Ghi dữ liệu của đối tượng điểm Tam trong tập tin Duong_tron.xml
- Ghi dữ liệu của đối tượng loại phòng trong tập tin Bang_don_gia.xml
- Ghi dữ liệu của đối tượng ngoại tệ trong tập tin Bang_ty_gia.xml

Nhận xét :

Cách ghi đối tượng x tương ứng thẻ X phụ thuộc vào X có phải là thẻ gốc hay không

Nếu X là thẻ gốc ==> Tạo thẻ X và Ghi trực tiếp

Nếu X là thẻ con của thẻ gốc ==> Chỉ tạo thẻ XmlElement tương ứng và sau đó bổ sung vào thẻ gốc trước khi ghi

Hướng giải quyết với X là thẻ gốc

Xây dựng hàm ghi của lớp đối tượng XL_X

Tham số : đường dẫn của tập tin Xml tương ứng

Kết quả : Giá trị logic

Thuật giải

Khai báo biến logic Kq

Tai_lieu = đối tượng XmlDocument

Goc = đối tượng XmlElement được tạo từ Tai_lieu

Gán giá trị các thuộc tính của Goc tương ứng biến thành phần của đối tượng

Bổ sung Goc vào Tai_lieu

Kq = Kết quả khi ghi Tai_lieu với Duong_dan

Trả Kq

Hướng giải quyết với X là thẻ con

Xây dựng hàm tạo nút của lớp đối tượng XL_X

Tham số : đối tượng XmlDocument Tai_lieu Kết

quả : đối tượng XmlElement

Thuật giải

Khai báo đối tượng XmlElement Kq

Kq = đối tượng XmlElement tạo ra từ Tai_lieu

Gán các thuộc tính của Kq tương ứng các biến thành phần của đối tượng

Trả Kq

*** Thẻ hiện cây đối tượng**

Vấn đề :

Cần thể hiện các thông tin của tập tin Xml dưới dạng cây tương ứng cấu trúc tổ chức các thẻ

Ví dụ :

Thẻ hiện cây Công ty – đơn vị

Thẻ hiện cây Trường - Khối - Lớp

Thẻ hiện cây các số nguyên

Hướng giải quyết chung :

Sử dụng đối tượng thẻ hiện cây trong thư viện đối tượng giao diện người dùng

Hướng giải quyết cụ thể với Visual Studio.NET Sử

dụng đối tượng thẻ hiện TreeView

III. Ngôn ngữXPath

Mục tiêu :

- Mở đầu về ngôn ngữ XPath trong ngữ cảnh kết hợp với DOM

==> Sẽ tiếp tục trình bày chi tiết về hệ thống các hàm khi kết hợp với XSLT (Chương 4)

- Cung cấp kiến thức, khái niệm tổng quát về ngôn ngữ truy vấn thông tin

==> Giúp sinh viên tự tìm hiểu và sử dụng tốt các ngôn ngữ truy vấn khác

Nội dung :

Truy vấn thông tin : Ý niệm chung về xử lý truy vấn thông tin

Ngôn ngữ truy vấn thông tin : Ý niệm chung về các ngôn ngữ truy vấn thông tin

Ngôn ngữ truy vấn XPath : Giới thiệu về Xpath, một loại ngôn ngữ truy vấn thông tin với tài liệu XML

Chuỗi định vị : Trình bày chi tiết cách định vị các nút trong Xpath (nội dung chính)

Chuỗi lọc : Giới thiệu sơ lược về cách lọc dữ liệu

1) Truy vấn thông tin

Truy vấn thông tin

- Một trong các loại xử lý quan trọng và rất thông dụng

- Loại xử lý cho phép trích rút thông tin về tập hợp con thông tin của một tập hợp thông tin nào đó

Ví dụ :

- Mảng 1 chiều các số nguyên là tập hợp các thông tin với các thành phần con là các số nguyên

- Mảng 2 chiều các phân số là tập hợp các thông tin với các thành phần con trực tiếp là các phân số.

Thành phần con này lại được cấu trúc từ 2 thành phần con khác là tử số, mẫu số

- Hệ thống các tập tin trong một đĩa là tập hợp các thông tin với các thành phần con là các tập tin, thư mục. Các thành phần này được cấu trúc theo dạng cây

- Cơ sở dữ liệu của phần mềm quản lý nhân sự là tập hợp thông tin với các thành phần con là các bảng dữ liệu (quan hệ) : NHAN_VIEN, TRINH_DO, DON_VI, v... . Các bảng này có cấu trúc và liên kết với nhau theo các ý niệm trong mô hình quan hệ

- Tài liệu XML (với mô hình DOM) là một tập hợp thông tin với các thành phần con là các nút. Các nút này được sắp xếp theo dạng cây

Tập hợp con thông tin

Khái niệm cho phép mô tả kết quả của việc truy vấn thông tin trên tập hợp thông tin gốc. Tập hợp con này cũng bao gồm các thành phần con , các thành phần con cũng có cách biểu diễn và cấu trúc riêng.

Cấu trúc của thành phần con thông thường là trùng với cấu trúc trong tập hợp gốc nhưng không nhất thiết.

Ví dụ :

- Tập hợp các số nguyên dương trong mảng 1 chiều a các số nguyên
- Tập hợp các phân số có giá trị lớn hơn 1 trong mảng 2 chiều các phân số
- Tập hợp các tập tin có tên bắt đầu với chuỗi ký tự A
- Tập hợp các nhân viên có tuổi từ 15 đến 20 trong danh sách nhân viên
- Tập hợp các nút không có nút con (nút lá) trong tập tin Xml

2) Ngôn ngữ truy vấn thông tin

Một trong các loại ngôn ngữ đặc tả

- Cho phép đặc tả tập hợp con các thông tin cần truy vấn dưới dạng một chuỗi : Chuỗi truy vấn
- Tùy theo dạng thông tin cần truy vấn, có rất nhiều ngôn ngữ truy vấn đã được đề xuất.

Với các cấu trúc dữ liệu mảng, ngôn ngữ truy vấn rất đơn giản với chuỗi truy vấn chỉ bao gồm một chỉ số (mảng 1 chiều) hay biểu thức gồm 2 thành phần : chỉ số dòng, chỉ số cột (mảng 2 chiều)

Ví dụ :

Truy xuất phần tử thứ 3 trong mảng 1 chiều a

a[3]

Truy vấn phần tử thuộc dòng 2, cột 4 trong mảng 2 chiều b

b[2][4]

Với hệ thống tập tin, chuỗi truy vấn có dạng đường dẫn bao hàm bên trong tên ổ đĩa , tên các thư mục và tên tập tin

Ví dụ :

Truy xuất tập tin THONG_BAO.Txt trong thư mục gốc đĩa C

C:\THONG_BAO.Txt

Truy xuất các tập tin trong thư mục A là con thư mục gốc đĩa C

C:\A*.*

Với cơ sở dữ liệu quan hệ, ngôn ngữ truy vấn được sử dụng thông dụng là ngôn ngữ SQL. Chuỗi truy vấn trong ngôn ngữ này có cấu trúc phức tạp hơn rất nhiều so với các ví dụ trên (và như thể khả năng truy vấn cũng tăng lên rất nhiều)

Ví dụ :

Truy xuất hồ sơ các nhân viên có đơn vị là đơn vị X với mã số là 5

```
Select * From NHAN_VIEN Where MDV=5
```

Với tập tin XML, mô hình DOM cho phép truy vấn thông tin với ngôn ngữ truy vấn **Xpath**. Chuỗi truy vấn trong Xpath có cấu trúc tương tự như đường dẫn (của hệ thống tập tin) nhưng phức tạp hơn rất nhiều với khái niệm về các trục định vị và bộ lọc (sẽ trình bày chi tiết sau)

Ví dụ :

Truy xuất các nút có giá trị là số nguyên dương trong cây các số nguyên

```
//Nut[@Gia_tri>0]
```

Các thành phần chính trong ngôn ngữ truy vấn

Mỗi ngôn ngữ truy vấn sẽ có dạng khác nhau về cấu trúc của chuỗi truy vấn. Tuy nhiên do cùng mục tiêu là truy vấn thông tin, các chuỗi truy vấn bao hàm bên trong 2 thành phần chính sau:

1. Thành phần định vị

2. Thành phần lọc

*** Thành phần định vị :**

Cho phép đặc tả “vị trí” của thông tin cần truy vấn. Kết quả của việc định vị là một tập hợp con các thông tin thuộc “vị trí” đang xét.

Thành phần định vị chỉ là bước đầu tiên trong quá trình truy vấn, sau bước này thông thường cần sử dụng thành phần lọc để mô tả chi tiết hơn thông tin cần truy vấn

Mỗi ngôn ngữ truy vấn sẽ dùng một số các từ khóa riêng cho phép đặc tả thành phần này

Ví dụ :

- Ngôn ngữ truy vấn tập tin dùng từ khóa chính là các ký hiệu \ (con trực tiếp) để định vị các tập tin, thực mục cần truy vấn trong chuỗi đường dẫn
- Ngôn ngữ SQL dùng 2 từ khóa chính là From , Select để định vị các mẫu tin cần truy vấn
- Ngôn ngữ Xpath cho phép định theo đường dẫn với ký hiệu / và mở rộng với các khái niệm về trục định vị (từ khóa descendant định vị thành phần con theo mọi cấp, từ khóa following-sibling định thành phần con là “anh/ em” của thành phần đang xét , v.v...)

***Thành phần lọc:**

Cho phép lọc kết quả của bước định vị với việc mô tả các tính chất của thông tin cần truy vấn thông qua biểu thức lọc

Biểu thức lọc thông dụng có dạng biểu thức logic (tương tự trong ngôn ngữ lập trình nhưng với một số giới hạn)

Ví dụ :

- Ngôn ngữ truy vấn tập tin không có ý niệm rõ nét về biểu thức lọc mà chỉ cho phép sử dụng một số ký tự đặc biệt (ký tự ? , ký tự *) để mô tả chi tiết về tên (tập tin , thực mục) cần truy vấn
- Ngôn ngữ SQL dùng từ khoá Where kết hợp biểu thức logic để cho phép lọc các mẫu tin cần truy vấn
- Ngôn ngữ Xpath sử dụng biểu thức lọc có dạng [Biểu thức điều kiện] để cho phép lọc các nút cần truy vấn

3) Ngôn ngữ truy vấn XPath

Một trong các ngôn ngữ truy vấn với

- Tập hợp thông tin bao gồm các đối tượng của mô hình DOM

- Tập hợp con thông tin : Danh sách các đối tượng của mô hình DOM

Chuoi_dinh_vi_1 Chuoi_loc_1 Chuoi_dinh_vi_2 Chuoi_loc_2 Chuoi_dinh_vi_n Chuoi_loc_n

***Chuoi_dinh_vi :**

Cho phép xác định tập hợp các nút có quan hệ (thông dụng là con) so với một tập hợp các nút X cho trước

Ví dụ :

/AAA/BBB/CCC --- > Tập hợp tất cả các nút tương ứng thẻ CCC

Là con của BBB với BBB là con của AAA , với AAA là con của gốc

//CCC ---- > Tập hợp tất cả các nút tương ứng thẻ CCC

BBB/CCC --- > Tập hợp tất cả các nút tương ứng thẻ CCC

Là con của BBB với BBB là con của nút ngữ cảnh

***Chuoi_loc :**

Có dạng sau

[Bieu_thuc_loc]

Bieu_thuc_loc là biểu thức logic cho phép xác định một tập hợp con các nút của tập hợp các nút X cho trước. Tập hợp con này là chính là tập hợp các nút của X thỏa điều kiện trong Bieu_thuc_loc.

Ví dụ :

/AAA/BBB/CCC[@x > 5] --- > Tập hợp tất cả nút tương ứng thẻ CCC

- Có thuộc tính x lớn hơn 5

- Là con của thẻ BBB, thẻ BBB là con của thẻ AAA, thẻ AAA là con của gốc

/AAA/BBB[@y='bbb']/CCC[@x>5] ---- > Tập hợp tất cả nút tương ứng thẻ CCC

- Có thuộc tính x lớn hơn 5

- Là con của thẻ BBB với thẻ BBB

- Có giá trị thuộc tính y là 'bbb'

- Là con của thẻ AAA, thẻ AAA là con của gốc

- Ví dụ minh họa

Ví dụ 1 :

Xét Tài liệu Xml với đặc tả DTD như sau

```
<?xmlversion="1.0"encoding="utf-8"?>
```

```
<!DOCTYPE TRUONG [
```

```
<!ELEMENT TRUONG (KHOI)+ >
```

```
<ATTLIST TRUONG
```

```
    Ten          CDATA #REQUIRED
```

```
>
```

```
<!ELEMENT KHOI (LOP)+ >
```

```
<ATTLIST KHOI
```

```
    Ten          CDATA #REQUIRED
```

```
>
<!ELEMENT LOP (HOC_SINH)+ >
<ATTLIST LOP
    Ten          CDATA #REQUIRED
    Si_so        CDATA #REQUIRED
    <!-- Si_so : A_int      -->
```

```
>
<!ELEMENT HOC_SINH EMPTY >
<ATTLIST HOC_SINH
    Ho_ten       CDATA #REQUIRED
    Gioi_tinh    CDATA #REQUIRED
    <!-- Si_so : A_int      -->
```

```
>
]>
```

Với nút ngữ cảnh là nút gốc (nút có tên là TRUONG).

Chuỗi truy vấn Xpath đặc tả tập hợp các lớp có số trên 40, có dạng như sau
child::KHOI/child::LOP[@Si_so>40]

Ví dụ 2:

Chuỗi truy vấn Xpath cho phép đặc tả tập hợp con các số nguyên dương trong tài liệu XML biểu diễn thông tin về cây các số nguyên có dạng sau (với nút ngữ cảnh là nút gốc)

```
descendant::SO_NGUYEN[@Gia_tri >0]
```

***Chuỗi định vị:**

Xpath cho phép định vị theo 2 cách

Cách 1 : Dùng trực định vị với tên cụ thể

Cách 2 : Dùng dạng tốc ký với các từ khóa thay thế tên trực định vị

Cách 1 là dạng cơ sở, dạng xử lý trực tiếp của các thành phần xử lý chuỗi Xpath (bộ xử lý phân tích , bộ xử lý định vị, xử lý bộ lọc). Tuy nhiên cách này không cho phép mô tả một cách ngắn gọn so với cách 2

Cách 2 là cách rất thông dụng vì cho phép mô tả một cách ngắn gọn, súc tích. Tuy nhiên không phải tất cả các trục đều có từ khóa thay thế nên trong một số trường hợp nhất định cách 1 là cách duy nhất có thể sử dụng

Ví dụ 1 : Với tài liệu XML biểu diễn thông tin các học sinh (DTD phía trên)

Thay vì sử dụng chuỗi định vị theo cách 1

```
child::KHOI/child::LOP[@Si_so>40]
```

Có thể sử dụng dạng tốc ký trong chuỗi định vị như sau

```
KHOI/LOP[@Si_so>40]
```

Hay

```
/TRUONG/KHOI/LOP[@Si_so>40]
```

Các dạng sử dụng này sẽ cho kết quả hoàn toàn trùng khớp với kết quả của dạng phía trên

Ví dụ 2 : Với tài liệu XML biểu diễn thông tin về biểu thức số học

Thay vì sử dụng chuỗi định vị theo cách 1
descendant::SO_NGUYEN[@Gia_tri
>0]

Có thể dùng dạng tốc ký

//SO_NGUYEN[@Gia_tri >0]

(Từ khóa // là dạng viết tắt cho trục descendant::)

Dạng sử dụng này sẽ cho kết quả hoàn toàn trùng khớp với kết quả của dạng phía trên

- **Định vị theo tên trục**

Cú pháp (dạng thông dụng)

Chuỗi định vị theo tên trục có dạng chung sau

Ten_truc:: Ten_nut

Ý nghĩa : đặc tả các tập hợp con các nút có tên là Ten_nut và có vị trí tương đối so với tập hợp nút đang xét

(kết quả trung gian) theo ý nghĩa của Ten_truc

Tên trục	Ý nghĩa	Ghi chú
self	Chính nút đang xét	Ít sử dụng
child	Con trực tiếp	Rất thông dụng
parent	Cha trực tiếp	
descendant	Tất cả con theo mọi cấp (không bao gồm nút đang xét)	Rất thông dụng
descendant-or-self	Tất cả con theo mọi cấp (bao gồm nút đang xét)	
ancestor	Tất cả cha theo mọi cấp (không bao gồm nút đang xét)	
ancestor-or-self	Tất cả cha theo mọi cấp (bao gồm nút đang xét)	
following-sibling	Tất cả anh em cùng cha(phía sau nút đang xét)	Ít sử dụng
preceding-sibling	Tất cả anh em cùng cha(phía trước nút đang xét)	Ít sử dụng
following	Tất cả anh em cùng cha (phía sau nút đang xét) cùng với các con theo mọi cấp	Ít sử dụng
preceding	Tất cả anh em cùng cha (phía trước nút đang xét) cùng với các con theo mọi cấp	Ít sử dụng

Ghi chú : Cho phép dùng ký tự * để mô tả một nút bất kỳ

- **Định vị theo tốc ký**

Cú pháp (dạng thông dụng)

Chuỗi định vị theo tên trục có dạng chung sau

Tu_khoa Ten_nut

Ý nghĩa : đặc tả các tập hợp con các nút có tên là Ten_nut và có vị trí tương đối so với tập hợp nút đang xét

Ý nghĩa của Ten_truc tương ứng với Tu_khoa

Tên trục	Từ khóa	Ghi chú
self	,	Ít sử dụng
child	/	Rất thông dụng, có thể dùng xác định nút ngữ cảnh chính là Document
parent	..	
descendant	//	Rất thông dụng

***Chuỗi lọc**

Chuỗi lọc **XXX[Bieu_thuc_loc]**

Cho phép đặc tả điều kiện lọc trên các thuộc tính của nút XXX đang xét

Bieu_thuc_loc có cú pháp hoàn toàn tương tự cú pháp của biểu thức điều kiện trong ngôn ngữ lập trình if (Biểu thức điều kiện)

```
{ // Các lệnh  
}
```

với một số khác biệt mà trong đó quan trọng nhất là

Biểu thức điều kiện của ngôn ngữ lập trình bao gồm các biến

Biểu thức điều kiện của XPath bao gồm các thuộc tính (cú pháp @Ten_thuoc_tinh)