Công nghệ XML và ứng dụng

2007

Đề cương

1. Mục tiêu

- Cung cấp các kiến thức cơ bản về công nghệ XML
- Rèn luyện các kỹ năng lập trình xử lý trên tài liệu XML

== > Ứng dụng tốt các công nghệ DTD, DOM, XSLT trong quá trình thiết kế và thực hiện phần mềm

2. Nội dung

Chương 1 : Mở đầu
1. Các ví dụ mở đầu
2. Tài liệu XML
3. Công nghệ XML
4. Giới thiệu về DOM

Chương 2 : Đặc tả cấu trúc và nội dung tài liệu XML
1. Các khái niệm
2. Đặc tả cấu trúc tài liệu XML với DTD
3. Đặc tả cấu trúc tài liệu XML với Xml-Schema

Chương 3 : Truy xuất tài liệu XML với DOM
1. Giới thiệu chung về DOM
2. Các thao tác cơ bản
3. Ngôn ngữ XPath
4. DOM và các công nghệ khác

Chương 4 : Biến đổi tài liệu XML với XSLT
1. Giới thiệu chung về XSLT
2. Hệ thống các thẻ cơ bản

3. Môi trường thực hành

Visual Studio.NET 2005

4. Tài liệu tham khảo

- Sách :
  Sách về môi trường Visual Studio.NET (phần liên quan XML)
  Sách kỹ thuật liên quan XML,DOM,Xpath,XSLT

- Giáo trình :
  Giáo trình "Công nghệ XML và ứng dụng " - Nguyễn tiến Huy

5. Thi

Thi lý thuyết : 5 điểm
Thi thực hành : 5 điểm

Chương 1 : Mở đầu

Giới thiệu chung về chương 1

I. Các ví dụ mở đầu

Mục tiêu :
- Minh họa việc sử dụng tập tin Xml để lưu trữ thông tin các đối tượng trong thực tế
- Mở đầu cho việc trình bày các khái niệm về tài liệu Xml cùng với định chuẩn Xml

1. Phân số

Tập tin Phan_so.xml biểu diễn thông tin về phân số 4/7 trên bộ nhớ phụ

```
<?xml version="1.0" encoding="utf-8" ?>
<PHAN_SO Tu_so="4" Mau_so="7" />
```

2. Dãy số nguyên

Tập tin Day_so.xml biểu diễn thông tin về dãy các số nguyên
-4,22,-3,15,7,12

```
<?xml version="1.0" encoding="utf-8" ?>
<DAY_SO>
 <SO Gia_tri="-4" />
 <SO Gia_tri="22" />
 <SO Gia_tri="-3" />
 <SO Gia_tri="15" />
 <SO Gia_tri="7" />
 <SO Gia_tri="12" />
</DAY_SO>
```

3. Đường tròn

Tập tin Duong_tron.xml biểu diễn thông tin về đường tròn C có tậm O(2,1) và bán kính R=4

```
<?xml version="1.0" encoding="utf-8" ?>
<DUONG_TRON Ban_kinh="4">
    <DIEM x="2"y="1" />
</DUONG_TRON>
```

4. Bảng đơn giá thuê phòng

Tập tin Bang_don_gia.Xml biểu diễn thông tin về bảng đơn giá thuê phòng của khách sạn

```
<?xml version="1.0" encoding="utf-8" ?>
<KHACH_SAN Ten="X" Dia_chi="123 ABC" >
 <LOAI_PHONG Ten="Loại A" Don_gia="280000" />
 <LOAI_PHONG Ten="Loại B" Don_gia="240000" />
 <LOAI_PHONG Ten="Loại C" Don_gia="180000" />
```

```
 <LOAI_PHONG  Ten="Loại đặc biệt"  Don_gia="320000" />
</KHACH_SAN>
```

5. Bảng tỷ giá

Tập tin Bang_ty_gia.xml biểu diễn thông tin bảng tỷ giá các ngoại tệ

```
<?xml  version="1.0"  encoding="utf-8" ?>
<BANG_TY_GIA  Ngay="14/6/2007">
 <NGOAI_TE  Ten="Đô la Mỹ"  Ky_hieu="USD"
                Mua_tien_mat="16103"  Mua_chuyen_khoan="16124"  Ban="16127" />
 <NGOAI_TE  Ten="Bảng Anh"  Ky_hieu="GBP"
                 Mua_tien_mat="31604"  Mua_chuyen_khoan="31699"  Ban="31910" />
 <NGOAI_TE  Ten="Đô la HongKong"  Ky_hieu="HKD"
                 Mua_tien_mat="2030"  Mua_chuyen_khoan="2054"  Ban="2074" />
</BANG_TY_GIA>
```

6. Kết quả xổ số

Tập tin Ket_qua_xo_so.xml biểu diễn thông tin kết quả xổ số tỉnh Bình thuận

```
<?xml  version="1.0"  encoding="utf-8" ?>
<KET_QUA  Ngay="14/6/2007"  Tinh="Bình Thuận" >
 <GIAI  Ten="Giải tám" >
   <SO  Gia_tri="98" />
 </GIAI>
 <GIAI Ten="Giải bảy" >
   <SO  Gia_tri="709" />
 </GIAI>
 <GIAI  Ten="Giải sáu" >
   <SO  Gia_tri="6137" />
   <SO  Gia_tri="0429" />
   <SO  Gia_tri="1351" />
 </GIAI>
 <GIAI  Ten="Giải năm" >
   <SO  Gia_tri="1268" />
 </GIAI>
 <GIAI  Ten="Giải bốn" >
   <SO  Gia_tri="00044" />
   <SO  Gia_tri="74572" />
   <SO  Gia_tri="49335" />
   <SO  Gia_tri="38644" />
   <SO  Gia_tri="74331" />
   <SO  Gia_tri="05085" />
   <SO  Gia_tri="66071" />
 </GIAI>
 <GIAI  Ten="Giải ba" >
   <SO  Gia_tri="20579" />
   <SO  Gia_tri="49194" />
 </GIAI>
```

```
 <GIAI  Ten="Giải nhì" >
   <SO  Gia_tri="54621" />
 </GIAI>
 <GIAI  Ten="Giải nhất" >
   <SO  Gia_tri="58998" />
 </GIAI>
 <GIAI  Ten="Giải đặc biệt" >
   <SO  Gia_tri="02700" />
 </GIAI>
</KET_QUA>
```

7. Mạch điện

Tập tin Mach_dien.xml biểu diễn thong tin về mạch điện với các điện trở

```
<?xml  version="1.0"  encoding="utf-8" ?>
<MACH_NOI_TIEP>
 <DIEN_TRO  Gia_tri="4" />
 <MACH_SONG_SONG>
   <DIEN_TRO  Gia_tri="2" />
   <MACH_NOI_TIEP>
     <DIEN_TRO  Gia_tri="2" />
     <MACH_SONG_SONG>
       <DIEN_TRO  Gia_tri="3" />
       <DIEN_TRO  Gia_tri="3" />
     </MACH_SONG_SONG>
     <DIEN_TRO  Gia_tri="4" />
   </MACH_NOI_TIEP>
   <MACH_NOI_TIEP>
     <DIEN_TRO  Gia_tri="2" />
     <DIEN_TRO  Gia_tri="1" />
   </MACH_NOI_TIEP>
 </MACH_SONG_SONG>
 <DIEN_TRO  Gia_tri="8" />
 <DIEN_TRO  Gia_tri="6" />
</MACH_NOI_TIEP>
```

II. Tài liệu XML

Mục tiêu :

- Trình bày khái niệm và ý nghĩa sử dụng của tài liệu Xml
- Trình bày chi tiết về định chuẩn Xml

1. Khái niệm về tài liệu XML

Văn bản có cấu trúc theo định chuẩn XML
cho phép biểu diễn thông tin về các đối tượng trong thực tế

Đối tượng x thuộc loại X trong thực tế
==== > Thẻ X trong tài liệu Xml

Thuộc tính a của đối tượng x thuộc loại X trong thực tế
=== > Thuộc tính a của thẻ X trong tài liệu Xml

Ví dụ 1 :

Phân số 4/5 trong thực tế
== > Thẻ <PHAN_SO Tu_so="4" Mau_so="5" />

Ví dụ 2 :

Dãy các số nguyên a bao gồm các số nguyên 1,4,5,-3
=== > Thẻ <DAY_SO>
<SO Gia_tri="1" />
<SO Gia_tri="4" />
<SO Gia_tri="5" />
<SO Gia_tri="-3" />
</DAY_SO>

2. Định chuẩn XML

*** Qui định 1 : Hệ thống các thẻ đánh dấu**

Các thẻ đánh dấu trong ngôn ngữ theo định chuẩn XML chỉ bao gồm 2 loại : Thẻ có nội dung và thẻ rỗng.
Các thẻ có nội dung có dạng
<Tên> Nội dung </Ten>
Các thẻ rỗng có dạng
<Tên />
Các thẻ (nói chung) có thể có hoặc không các thuộc tính với các tên khác nhau (trong cùng thẻ).
Thuộc tính trong một thẻ có dạng
Ten_thuoc_tinh=”Gia_tri”
Hay
Ten_thuoc_tinh=’Gia_tri’

Ví dụ : với tài liệu XML

```
<?xml version="1.0" encoding="utf-8"?>
<DUONG_TRON Ban_kinh="5">
    <DIEM x="4" y="2"/>
</DUONG_TRON>
```

Thẻ có nội dung là thẻ DUONG_TRON
Thẻ rổng là thẻ DIEM
Thẻ DUONG_TRON có 1 thuộc tính là Ban_kinh
Thẻ DIEM có 2 thuộc tính là x, y

*** Qui định 2 : Quan hệ lồng nhau (chứa trong) giữa các thẻ có nội dung**

Nội dung bên trong thẻ có nội dung có thể là các thẻ khác (có nội dung hay rổng). Khi thẻ A có nội dung là thẻ B ta gọi

Thẻ A là thẻ cha của B , thẻ A chứa thẻ B

Thẻ B là thẻ con của A, thẻ B được chứa trong A

Qui định 2 yêu cầu các thẻ với quan hệ lồng nhau hoàn toàn. Khi thẻ A là thẻ cha của thẻ B, A phải chứa phần bắt đầu và cả phần kết thúc của B

Ví dụ :

Thẻ A là thẻ cha của B với dạng lồng nhau hoàn toàn (hợp lệ)

```
<A>
        <B> ….</B>
</A>
```

Thẻ A là thẻ cha của B với dạng lồng nhau không hoàn hoàn toàn (không hợp lệ)

```
<A>
        <B> ….</A>
</B>
```

*** Qui định 3 : Thẻ đánh dấu gốc**

Qui định 3 yêu cầu tài liệu XML phải có duy nhất (một và chỉ một) thẻ chứa (trực tiếp hay gián tiếp) tất cả các thẻ còn lại (nếu có)

Ví dụ : Tài liệu XML sau biểu diễn thông tin 2 đường tròin là không hợp lệ vì không có thẻ gốc

```
<?xml   version="1.0"  encoding="utf-8"?>
<DUONG_TRON   Ban_kinh="5">
    <DIEM  x="4"   y="2"/>
</DUONG_TRON>

<DUONG_TRON  Ban_kinh="5">
    <DIEM  x="4"  y="2"/>
</DUONG_TRON>
```

III. Công nghệ XML

Mục tiêu :

- Trình bày khái niệm và ý nghĩa của công nghệ Xml
- Trình bày các khả năng ứng dụng cơ bản của công nghệ Xml

1. Khái niệm về công nghệ Xml

Nội dung
- Mô tả tóm tắt về các loại công nghệ trong lĩnh vực công nghệ thông tin
- Trình bày quá trình hình thành của công nghệ Xml và quan hệ với các loại công nghệ khác

a. Công nghệ thông tin

- Ngành khoa học ngiên cứu về việc xây dựng các hệ thống tin học tương ứng với hệ thống thực tế
- Ngành khoa học nghiên cứu về việc biểu diễn và xử lý thông tin của hệ thống tin học tương ứng với các thông tin và nghiệp vụ của hệ thống thực tế

Hai hướng nghiên cứu chính trong Công nghệ thông tin
- Công nghệ phần cứng
- Công nghệ phần mềm

* Công nghệ xử lý thông tin

- Một trong các hướng nghiên cứu chính của công nghệ phần mềm
- Ngành khoa học nghiên cứu về các mô hình, phương phàp, kỹ thuật xử lý thông tin

Các mô hình xử lý (chung) chính trong Công nghệ xử lý thông tin
- Thủ tục/Hàm (Procedure/Function)
- Đơn thể (Module)
- Đối tượng (Object)
- Thành phần (Component)
- Dịch vụ (Service)
- ….

* Công nghệ biểu diễn thông tin

- Một trong các hướng nghiên cứu chính của công nghệ phần mềm
- Ngành khoa học nghiên cứu về các mô hình, phương phàp, kỹ thuật biểu diễn thông tin

Các mô hình biểu diễn (chung) chính trong Công nghệ biểu diễn thông tin
- Tập tin (File) : Thông tin lưu trữ trên bộ nhớ phụ
- Cơ sở dữ liệu (Database) : Thông tin lưu trữ trên bộ nhớ phụ
- Cấu trúc dữ liệu (Data Structure) : Thông tin xử lý trong bộ nhớ chính
- Trang Web (WebPage) : Thông tin thể hiện
- Luồng dữ liệu (Data Stream) : Thông tin trao đổi nội bộ bên trong một hệ thống tin học hay giữa các các hệ thống tin học
- ….

b. Công nghệ Xml

Khái niệm về Công nghệ XML
- Thuộc loại công nghệ biểu diễn thông tin
- Hình thành từ nhu cầu và vấn đề cần giải quyết của việc trao đổi thông tin
- Có phạm vi nghiên cứu và ứng dụng trên tất cả mô hình biểu diễn của công nghệ biểu diễn thông tin
- Có hướng nghiên cứu cho phép ứng dụng một mô hình xử lý thông tin mới thuộc về công nghệ xử lý thông tin

*** Nhu cầu trao đổi thông tin**

1. Trao đổi thông tin nội bô bên trong hệ thống tin học

Sự phát triển về qui mô, độ phức tạp, phạm vi sử dụng của các hệ thống tin học dẫn đến sự phân rả hệ thống cần xây dựng thành các hệ thống con (kiến trúc đa tầng là một ví dụ điển hình về sự phân rả như thế)

=== > Nhu cầu về trao đổi thông tin bên trong các hệ thống con

2. Trao đỗi thông tin giữa các hệ thống tin học

- Sự phát triển của Internet và các ứng dụng trên Web , đặc biệt là các ứng dụng trong lĩnh vực thương mại điện tử

==== > Nhu cầu về trao đổi thông tin giửa các ứng dụng này

- Các yêu cầu cao về chất lượng phần mềm (Tiện dụng, Tương thích, Bảo mật, v.v…) khả năng đáp ứng cao, chuyên biệt của một số hệ thống sẳn có (WebBrowser, Excel, Word, Fax, v.v…)

==== > Nhu cầu về trao đổi thông tin giữa hệ thống đang xây dựng và các hệ thống có sẳn

*** Mô hình trao đổi thông tin**

Mô hình trao đổi thông tin trước khi XMl ra đời chủ yếu dựa trên công nghệ về luồng dữ liệu (Data Stream) với 2 dạng chính

- Dạng nhị phân : Dữ liệu trao đổi là chuỗi các byte theo một cấu trúc và ngữ nghĩa riêng biệt của từng ứng dụng
- Dạng văn bản : Dữ liệu trao đổi là chuỗi các ký tự theo cách mã hóa chung nhưng cấu trúc và ngữ nghĩa vẫn là riêng biệt cho từng ứng dụng

Cả 2 dạng trao đổi trên đều không thích hợp với các nhu cầu phía trên
với cùng khuyết điểm :
"Thông tin cần trao đổi có cấu trúc và ngữ nghĩa riêng biệt theo từng ứng dụng "

=== > Nhu cầu về một định chuẩn chung khi trao đổi thông tin

*** Sự ra đời của Công nghệ XML**

Công nghệ XML ra đời là kết quả của các nghiên cứu về dạng biểu diễn thông tin khi cần trao đổi giữa các hệ thống tin học. Dạng biểu diễn cần thỏa mản các yêu cầu sau

1) Cho phép trao đổi trên phạm vi rộng (Internet)
2) Dễ dàng trong việc kết xuất và tiếp nhận khi trao đổi
3) Tuân theo một định chuẩn chung được chấp nhận và hổ trợ của nhiều môi trường phát triển phần mềm

Ghi chú :

Công nghệ XML đã ra đời và đề xuất một dạng biểu diễn thích hợp cho các yêu cầu trên (tài liệu XML). Tuy nhiên với bản chất hình thành của mình, phạm vi ứng dụng của các tài liệu XML không chỉ dừng ở việc trao đổi thông tin mà bao hàm cả các vấn đề biểu diễn thông tin khác như : Lưu trữ thông tin , cấu trúc dữ liệu , thể hiện thông tin , v.v.. (chi tiết trong phần ứng dụng của XML)

2. Một số ứng dụng của công nghệ Xml

Nội dung :

Trình bày các hướng ứng dụng chính của công nghệ Xml

*** Trao đổi thông tin**

Ứng dụng Xml trong việc trao đổi thông tin

Trao đổi thông tin là xuất phát điểm cho sự hình thành của công nghệ XML.
Trao đổi thông tin là ứng dụng chính yếu nhất của XML
Có 2 dạng trao đổi thông tin chính
Dạng 1 : Trao đổi thông tin nội bộ giữa các thành phần của cùng hệ thống tin học
Dạng 2 : Trao đỗi thông tin giữa các hệ thống tin học khác nhau

XML có thể ứng dụng tốt cho cả 2 dạng trao đổi thông tin trên

A có nhu cầu trao đổi thông tin với B

Dạng 1 :
- A, B là 2 thành phần bên trong một hệ thống tin học (giao diện hay xử lý hay lưu trữ)
- Tài liệu XML đuợc thiết kế cho việc sử dụng nội bộ

Dạng 2 :
- A là hệ thống tin học đang xem xét
- B là hệ thống đã có trước với khả năng chuyên biệt nào đó
- A phải sử dụng tài liệu XML có cấu trúc do do B đề xuất

Ví dụ : XML có thể được sử dụng để
- Trao đổi thông tin giữa các tầng của một ứng dụng đuợc thiết kế theo mô hình kiến trúc đa tầng
- Trao đổi thông giữa một tấng với hệ thống khác bên ngoài

Cụ thể có thể
- Sử dụng XML trao đổi thông tin giữa hệ thống lưu trữ dữ liệu (thông thướng là hệ quản trị cơ sỡ dữ liệu) và tầng xử lý lưu trữ dữ liệu
- Sử dụng XML trao đỏi thông tin giữa tầng dữ liệu và tầng xử lý nghiệp vụ
- Sử dụng XML trao đỏi thông tin giữa tầng xử lý nghiệp vụ và tậng thể hiện
- Sử dụng XML trao đỏi thông tin giữa các tầng xử lý nghiệp vụ (khi hệ thống có nhiều tầng xử lý nghiệp vụ)

*** Lưu trữ thông tin**

Ứng dụng Xml trong việc lưu trữ thông tin

Có 3 cách ứng dụng chính của XML để lưu trữ dữ liệu bên trong hệ thống tin học
Cách 1 : Chỉ sử dụng các tập tin XML để lưu trữ dữ liệu
Cách 2 : Một số dữ liệu lưu trữ dưới dạng tập tin XML, một số khác lưu trữ bên trong cơ sỡ dữ liệu
Cách 3 : Lưu trữ toán bộ bên trong cơ sở dữ liệu, tài liệu XML khi đó đuợc nhúng vào nội dung các bảng dữ liệu

Cách 1 :
- Ưu điểm chính :
  Không cần sự hổ trợ của các hệ quản trị cơ sở dữ liệu
  == > Dễ cài đặt, triển khai
- Khuyết điểm chính :
  Tính hiệu quả không cao khi khối lượng dữ liệu l
- Nhận xét :

Các phần mềm trò chơi là ứng viên tốt cho ứng dụng XMl theo cách 1

Các phần mềm quản lý không thích hợp cho cách ứng dụng này

Rất thích hợp cho các ứng dụng trên các mội trường tin học không có (hoặc chưa có) hệ quản trị cơ sỡ dữ liệu như : Điện thoại di động, Máy công cụ , v.v…

Cách 2, 3 :

- Ưu điểm chính :
  Có thể kết hợp tốt ưu điểm của cả 2 mô hình lưu trữ thông tin : XML , Cơ cở dữ liệu
- Khuyết điểm chính :
  Cần có sự cân nhắc và quyết định đúng đắn loại thông tin nào sẽ đùng hình thức lưu trữ nào
- Nhận xét :
  Cách 2 là cách sử dụng phổ biến nhất hiện nay
  Cấu hình của hệ thống tin học (phân hệ ứng dụng) là loại thông tin thướng được chọn để lưu trữ theo dạng tài liệu XML

*** Cấu trúc dữ liệu**

Ưng dụng Xml với các cấu trúc dữ liệu

Với mô hình DOM (được giới thiệu tóm tắt trong phần kế tiếp và chi tiết trong chương 3), có thể sử dụng tài liệu XMLnhư một loại cấu trúc dữ liệu động với nhiều ưu điểm

- Đọc/Ghi dễ dàng

Các cấu trúc dữ liệu động như Mảng động (Dynamic Array), Xâu (List), Ngăn xếp (Stack), Hàng đợi (Queue), Cây (Tree), … có nhiều tính năng tốt trong việc biểu diễn và xử lý thông tin trong bộ nhớ chính. Tuy nhiên việc đọc/ghi thông tin của các cấu trúc dữ liệu này từ/vào bộ nhớ phụ (thông thướng thông qua tập tin) là không đơn giãn và thường phải thực hiện gián tiếp với một bộ đọc ghi. Tài liệu XML có thể sử dụng để cài đặt lại hấu hết các cấu trúc dữ liệu động trên (với một số chức năng bổ sung vào DOM qua cơ chế kế thừa hay bao bọc của hướng đối tượng !!!) và đặc biệt là việc đọc/ghi rất dễ dàng (thông thường chỉ là một lệnh gọi hàm đơn giãn)

Ví dụ : Với VB.NET

đọc tài liệu XML

```
Tai_lieu.Load(Ten_tap_tin_XML)
```

ghi tài liệu XML

```
Tai_lieu.Save (Ten_tap_tin_XML)
```

- Khả năng truy vấn cao

Việc truy vấn các thành phần/tập họp thành phần của các cấu trúc dữ liệu động phía trên thông thường phải thông qua các vòng lặp duyệt đến từng phần tử. Với tài liệu XML, có thể sử dụng ngôn ngữ truy vấn Xpath để truy xuất đến thành phần/tập họp thành phần một cách rất dễ dàng (và thông thường cũng chỉ thông qua một lệnh gọi hàm đơn giãn)

Ví dụ : Với VB.Net , giả sử có Tai_lieu tương ứng thông tin về cây các số nguyên. Để lập danh sách các nút (thành phần của cây) có giá trí dương

```
Nut_duong=Tai_lieu.SelectNodes("//Phan_tu[@Gia_tri >0]")
```

*** Xử lý thông tin**

Ung dụng Xml khi xử lý thông tin

Như đã trình bày trong các phần trên , công nghệ XML được xếp vào loại công nghệ biểu diễn thông tin, và như thể các hướng ứng dụng chính của XML đều nhắm vào giải quyết/ cải tiến các vấn đề về biểu diễn thông tin trên các loại hình biểu diễn khác nhau.

Tuy nhiên một trong các khả năng ứng dụng khá thú vị và có nhiều hứa hẹn sẽ phát triển mạnh trong tương lại liên quan đến công nghệ xử lý thông tin với việc đề xuất một mô hình xử lý thông tin mới theo hướng đặc tả thay vì lập trình (chi tiết về hướng ứng dụng này được trình bày chi tiết trong chương cuối)

Ý tưởng xuất phát từ việc tài liệu XML cho phép biểu diễn rất tốt các văn bản có cấu trúc. Và chương trình nguồn trong các ngôn ngữ lập trình cũng là các văn bản có cấu trúc. Có nên hay không ? tạo ra một ngôn ngữ lập trình mới tương tự như ngôn ngữ lập trình hiện nay nhưng với các từ khóa là các thẻ đánh dấu (ví dụ <for> ….</for>, <function>….<function>)

Câu trả lời rất tiếc là không, vì lập trình trên một ngôn ngữ như thế là rất khó khăn, không tự nhiên theo các thuật giải đã đề xuất.

Ý tưởng về một ngôn ngữ lập trình mới theo định chuẩn XML không thành công, nhưng nếy thay vì sử dụng ngôn ngữ lập trình mà lại sử dụng ngôn ngữ đặc tả (chỉ mô tả mà không đi vào chi tiết thuật giải) thì kết quả có được rất khả quan. Ngôn ngữ đặc tả XSLT đã ra đời trong bối cảnh như thế.)

Với XSLT có thể xây dựng một chương trình theo hướng đặc tả với các thẻ xử lý có ngữ nghĩa rất cao (mà đặc biệt là một cơ chế mới về vòng lặp) và tính dễ mang chuyễn tốt nhất có thể có (vì đầy là ngôn ngữ XML). XSLT còn có nhiều đặc tính thú vị khác sẽ đuợc trình bày chi tiết trong chương cuối

IV. Giới thiệu về DOM

Khái niệm về DOM (Document Object Model)

- Mô hình đối tượng cho phép xử lý trên tài liệu XML từ các ngôn ngữ lập trình
- Cấu trúc dữ liệu động biểu diễn thông tin của văn bản có cấu trúc nói chung và tài liệu XML nói riêng

DOM bao gồm hệ thông các đối tượng thư viện cho phép truy xuất nội dung của tài liệu Xml
Toàn bộ tập tin Xml trên bô nhớ phụ
=== > Đối tượng XmlDocument của DOM
Mỗi thẻ bên trong tập tin Xml
=== > Đối tượng XmlElement của DOM
Mỗi thuộc tính của thẻ
==== > Đối tương XmlAttribute của DOM

Ví dụ 1 :
Tập tin Duong_tron.xml biểu diễn thông tin về đường tròn C có tậm O(2,1) và bán kính R=4

```
<?xml   version="1.0"   encoding="utf-8" ?>
<DUONG_TRON   Ban_kinh="4">
   <DIEM   x="2"y="1" />
</DUONG_TRON>
```

Tương ứng
1 đối tượng XmlDocument

2 đối tượng XmlElement
3 đối tượng XmlAttribute

Ví dụ 2 :
Tập tin Bang_don_gia.Xml biểu diễn thông tin về bảng đơn giá thuê phòng của khách sạn

```
<?xml  version="1.0"  encoding="utf-8" ?>
        <KHACH_SAN  Ten="X"  Dia_chi="123 ABC" >
            <LOAI_PHONG  Ten="Loại A"  Don_gia="280000" />
            <LOAI_PHONG  Ten="Loại B"  Don_gia="240000" />
            <LOAI_PHONG  Ten="Loại C"  Don_gia="180000" />
            <LOAI_PHONG  Ten="Loại đặc biệt"  Don_gia="320000" />
        </KHACH_SAN>
```

Tương ứng
1 đối tượng XmlDocument
5 đối tượng XmlElement
10 đối tượng XmlAttribute

1. Các thao tác cơ bản

Trình bày các thao tác cơ bản nhất về DOM

* Khai báo sử dụng thư viện DOM

Khai báo sử dụng DOM
VB6
    Sử dụng thư viện Microsoft XML, v50
        Khai báo trong chức năng Project- References
VB.NET
    Sử dụng thư viện Xml
        Imports System.Xml
C#
    Sử dụng thư viện Xml
        using System.Xml ;

* Khai báo sử dụng tài liệu Xml

Khai báo sử dụng tài liệu XML
VB6
    Khai báo đối tượng thuộc lớp DOMDocument
        Dim Tai_lieu As New DOMDocument
VB.NET
    Khai báo đối tượng thuộc lớp XmlDocument
        Dim Tai_lieu As New XmlDocument
C#
    Khai báo đối tượng thuộc lớp XmlDocument
        XmlDocument Tai_lieu = new XmlDocument ();

* Đọc - Ghi tài liệu Xml

Đọc/ghi tài liệu Xml

VB6

Đọc : Kq= Tai_lieu.Load (Ten_tap_tin)

Ghi : Tai_lieu.Save(Ten_tap_tin)

VB.NET

Đọc : Tai_lieu.Load (Ten_tap_tin)

Ghi : Tai_lieu.Save(Ten_tap_tin)

C#

Đọc : Tai_lieu.Load (Ten_tap_tin) ;

Ghi : Tai_lieu.Save(Ten_tap_tin) ;

*** Truy xuất nút gốc**

Truy xuất đến nút gốc của tài liệu XML

VB6

Khai báo đối tượng thuộc giao diện IXMLDOMElement và nhận đối tượng từ chức năng tương ứng trong Tai_lieu

```
Dim Goc As IXMLDOMElement
Set Goc= Tai_lieu.documentElement
```

VB.NET

Khai báo đối tượng thuộc giao diện XmlElement và nhận đối tượng từ chức năng tương ứng trong Tai_lieu

```
Dim Goc As XmlElement
Goc=Tai_lieu.DocumentElement
```

C#

Khai báo đối tượng thuộc giao diện XmlElement và nhận đối tượng từ chức năng tương ứng trong Tai_lieu

```
XmlElement Goc ;
Goc=Tai_lieu.DocumentElement;
```

*** Truy xuất nút con trực tiếp của nút**

Truy xuất đến nút con trực tiếp của một nút

VB6

Khai báo đối tượng Nut_con thuộc giao diện IXMLDOMElement và nhận đối tượng từ chức năng tương ứng của Nut_cha

```
Dim Nut_con As IXMLDOMElement
Set Nut_con= Nut_cha.selectSingleNode(Ten_nut_con)
```

VB.NET

Khai báo đối tượng Nut_con thuộc giao diện XmlElement và nhận đối tượng từ chức năng tương ứng của Nut_cha

```
Dim Nut_con As XmlElement
Nut_con= Nut_cha.SelectSingleNode(Ten_nut_con)
```

C#

Khai báo đối tượng Nut_con thuộc giao diện XmlElement và nhận đối tượng từ chức năng tương ứng của Nut_cha

```
XmlElement Nut_con ;
```

Nut_con= Nut_cha.SelectSingleNode(Ten_nut_con) ;

*** Truy xuất thuộc tính của nút**

Truy xuất đến thuộc tính của một nút

VB6

Nhận giá trị :

Gia_tri=Nut.getAttribute(Ten_thuoc_tinh)

Cập nhật/bổ sung thuộc tính với giá trị

Nut.setAttribute Ten_thuoc_tinh, Gia_tri

VB.NET

Nhận giá trị :

Gia_tri=Nut.GetAttribute(Ten_thuoc_tinh)

Cập nhật/bổ sung thuộc tính với giá trị

Nut.SetAttribute (Ten_thuoc_tinh, Gia_tri)

C#

Nhận giá trị :

// Nhận một số nguyên

Gia_tri=int.Parse(Nut.GetAttribute(Ten_thuoc_tinh)) ;

Cập nhật/bổ sung thuộc tính với giá trị

Nut.SetAttribute(Ten_thuoc_tinh, Gia_tri.ToString()) ;

*** Truy xuất nội dung của nút**

Truy xuất đến nội dung (giá trị chuỗi) của một nút

VB6

Nhận nội dung :

Gia_tri=Nut.nodeValue

Cập nhật nội dung

Nut.nodeValue=Gia_tri

VB.NET

Nhận nội dung :

Gia_tri=Nut.InnerText

Cập nhật nội dung

Nut.InnerTextt=Gia_tri

C#

Nhận nội dung :

// Nhận số thực

Gia_tri=Double.Parse(Nut.InnerText);

Cập nhật nội dung

Nut.InnerTextt=Gia_tri.ToSstring();

*** Tạo nút mới**

Tạo nút mới

VB6

Khai báo đối tượng Nut thuộc giao diện IXMLDOMElement

và nhận đối tượng từ chức năng tương ứng của Tai_lieu
```
Dim Nut As IXMLDOMElement
Set Nut= Tai_lieu.CreateElement(Ten_nut)
```

VB.NET

Khai báo đối tượng Nut thuộc giao diện XmlElement và nhận đối tượng từ chức năng tương ứng của Tai_lieu
```
Dim Nut As XmlElement
Nut= Tai_lieu.CreateElement(Ten_nut)
```

C#

Khai báo đối tượng Nut thuộc giao diện XmlElement và nhận đối tượng từ chức năng tương ứng của Tai_lieu
```
XmlElement Nut ;
Nut= Tai_lieu.CreateElement(Ten_nut) ;
```

*** Bổ sung nút vào nút cha**

Bổ sung nút con vào nút cha

VB6
```
Nut_cha.appendChild Nut_con
```

VB.NET
```
Nut_cha.AppendChild( Nut_con)
```

C#
```
Nut_cha.appendChild (Nut_con);
```

2. Ví dụ minh họa

Giới thiệu chung về các ví dụ minh họa

*** Đọc phân số**

Cách 1 : Chỉ sử dụng hàm Main

```
Imports System.Xml
Module Doc_phan_so_Main
   Public Sub Main()
      Dim Tu_so, Mau_so As Integer
      Dim Duong_dan As String = "..\..\Du_lieu\Phan_so.xml"
      Dim Tai_lieu As New XmlDocument
      Tai_lieu.Load(Duong_dan) ' Chưa xử lý lỗi
      Dim Goc As XmlElement = Tai_lieu.DocumentElement
      Tu_so = Goc.GetAttribute("Tu_so")
      Mau_so = Goc.GetAttribute("Mau_so")
      Dim Chuoi As String = "Phân số : "
      Chuoi &= Tu_so & "/" & Mau_so
      Console.Write(Chuoi)
```

```
        Console.ReadLine()
    End Sub
End Module
========================
```
Cách 2 : Sử dụng kiểu & hàm tự định nghĩa
```
Imports System.Xml
Module Doc_phan_so_Kieu_Ham
    Structure PHAN_SO
        Public Tu_so As Integer
        Public Mau_so As Integer
    End Structure
    Public Sub Main()
        Dim Ps As PHAN_SO
        Dim Duong_dan As String = "..\..\Du_lieu\Phan_so.xml"
        Ps = Doc_phan_so(Duong_dan)
        Dim Chuoi As String = "Phân số : " & Chuoi_phan_so(Ps)
        Console.Write(Chuoi)
        Console.ReadLine()
    End Sub
    Public Function Doc_phan_so(ByVal Duong_dan As String) As PHAN_SO
        Dim Kq As PHAN_SO
        Dim Tai_lieu As New XmlDocument
        Tai_lieu.Load(Duong_dan) ' Chưa xử lý lỗi
       Dim Goc As XmlElement = Tai_lieu.DocumentElement
        Kq.Tu_so = Goc.GetAttribute("Tu_so")
        Kq.Mau_so = Goc.GetAttribute("Mau_so")
        Return Kq
    End Function
    Public Function Chuoi_phan_so(ByVal Ps As PHAN_SO) As String
        Dim Kq As String = ""
        Kq &= Ps.Tu_so & "/" & Ps.Mau_so
        Return Kq
    End Function
End Module
```

- Đọc phân số VB.NET

Cách 1 : Chỉ sử dụng hàm Main

```
Imports System.Xml
Module Doc_phan_so_Main
    Public Sub Main()
        Dim Tu_so, Mau_so As Integer
        Dim Duong_dan As String = "..\..\Du_lieu\Phan_so.xml"
        Dim Tai_lieu As New XmlDocument
        Tai_lieu.Load(Duong_dan) ' Chưa xử lý lỗi
        Dim Goc As XmlElement = Tai_lieu.DocumentElement
        Tu_so = Goc.GetAttribute("Tu_so")
        Mau_so = Goc.GetAttribute("Mau_so")
        Dim Chuoi As String = "Phân số : "
```

```
        Chuoi &= Tu_so & "/" & Mau_so
        Console.Write(Chuoi)
        Console.ReadLine()
    End Sub
End Module
========================
```

Cách 2 : Sử dụng kiểu & hàm tự định nghĩa

```
Imports System.Xml
Module Doc_phan_so_Kieu_Ham
    Structure PHAN_SO
        Public Tu_so As Integer
        Public Mau_so As Integer
    End Structure
    Public Sub Main()
        Dim Ps As PHAN_SO
        Dim Duong_dan As String = "..\..\Du_lieu\Phan_so.xml"
        Ps = Doc_phan_so(Duong_dan)
        Dim Chuoi As String = "Phân số : " & Chuoi_phan_so(Ps)
        Console.Write(Chuoi)
        Console.ReadLine()
    End Sub
    Public Function Doc_phan_so(ByVal Duong_dan As String) As PHAN_SO
        Dim Kq As PHAN_SO
        Dim Tai_lieu As New XmlDocument
        Tai_lieu.Load(Duong_dan) ' Chưa xử lý lỗi
       Dim Goc As XmlElement = Tai_lieu.DocumentElement
        Kq.Tu_so = Goc.GetAttribute("Tu_so")
        Kq.Mau_so = Goc.GetAttribute("Mau_so")
        Return Kq
    End Function
    Public Function Chuoi_phan_so(ByVal Ps As PHAN_SO) As String
        Dim Kq As String = ""
        Kq &= Ps.Tu_so & "/" & Ps.Mau_so
        Return Kq
    End Function
End Module
```

- Đọc phân số C#

Cách 1 : Chỉ sử dụng hàm Main

```
using System;
using System.Xml;
public class Doc_phan_so_Main
{
    public static void Main()
    {
        int Tu_so, Mau_so;
        String Duong_dan = "..\\..\\Du_lieu\\Phan_so.xml";
        XmlDocument Tai_lieu = new XmlDocument();
```

```
        Tai_lieu.Load (Duong_dan) ; // Chưa xử lý lỗi
        XmlElement Goc=Tai_lieu.DocumentElement ;
        Tu_so =int.Parse (Goc.GetAttribute ("Tu_so")) ;
        Mau_so =int.Parse (Goc.GetAttribute ("Mau_so"));
        String Chuoi = "Phân số :";
        Chuoi += Tu_so + "/" + Mau_so;
        Console.Write(Chuoi);
        Console.ReadLine();
    }
}
```

Cách 2 : Sử dụng kiểu & hàm tự định nghĩa

```
using System;
using System.Xml;
public class Ghi_phan_so_Kieu_Ham
{
    public struct PHAN_SO
    {
        public int Tu_so; // > 0
        public int Mau_so; // > 0
    }
    public static void Main()
    {
        PHAN_SO Ps;
        Ps = Nhap_phan_so();
        String Duong_dan = "..\\..\\Du_lieu\\Phan_so.xml";
        if (Ghi_phan_so(Ps, Duong_dan))
            Console.Write("Đã ghi ");
        else
            Console.Write("Lỗi khi ghi ");
        Console.ReadLine();
    }
    public static PHAN_SO Nhap_phan_so()
    {
        PHAN_SO Kq;
        Console.Write("Tử số:");
        Kq.Tu_so = int.Parse(Console.ReadLine()); // Chưa xử lý lỗi
        Console.Write("Mẩu số:");
        Kq.Mau_so = int.Parse(Console.ReadLine()); // Chưa xử lý lỗi
        return Kq;
    }
    public static Boolean Ghi_phan_so(PHAN_SO Ps, String Duong_dan )
    {
        Boolean Kq = true;
        XmlDocument Tai_lieu = new XmlDocument();
        XmlElement Goc = Tai_lieu.CreateElement("PHAN_SO");
        Goc.SetAttribute("Tu_so", Ps.Tu_so.ToString());
        Goc.SetAttribute("Mau_so", Ps.Mau_so.ToString());
        Tai_lieu.AppendChild(Goc);
```

```
        // Ghi tài liệu Xml

        Tai_lieu.Save(Duong_dan);

        return Kq;
    }
}
```

*** Đọc đường tròn**

```
Imports System.Xml
Module Doc_duong_tron
    Structure DIEM
        Public x As Double
        Public y As Double
    End Structure
    Structure DUONG_TRON
        Public Tam As DIEM
        Public Ban_kinh As Double
    End Structure
    Public Sub Main()
        Dim Dt As DUONG_TRON
        Dim Duong_dan As String = "..\..\Du_lieu\Duong_tron.xml"
        Dt = Doc_duong_tron(Duong_dan)
        Dim Chuoi As String = "Phương trình dường tròn : " & Chuoi_duong_tron(Dt)
        Console.Write(Chuoi)
        Console.ReadLine()
    End Sub
    Public Function Doc_duong_tron(ByVal Duong_dan As String) As DUONG_TRON
        Dim Kq As DUONG_TRON
        Dim Tai_lieu As New XmlDocument
        Tai_lieu.Load(Duong_dan) ' Chưa xử lý lỗi
        Dim Goc As XmlElement = Tai_lieu.DocumentElement
        Kq.Ban_kinh = Goc.GetAttribute("Ban_kinh")
        Dim Nut As XmlElement = Goc.ChildNodes(0)
        Kq.Tam.x = Nut.GetAttribute("x")
        Kq.Tam.y = Nut.GetAttribute("y")
        Return Kq
    End Function
    Public Function Chuoi_duong_tron(ByVal Dt As DUONG_TRON) As String
        Dim Kq As String = ""
        Kq &= String.Format(" (x -{0})^2 + ( y -{1})^2 = {2} ^2", Dt.Tam.x, Dt.Tam.y, Dt.Ban_kinh)
        Return Kq
    End Function
End Module
```

*** Đọc dãy số**

```
Imports System.Xml
```

```
Module Doc_day_so

    Public Sub Main()
        Dim a As ArrayList
        Dim Duong_dan As String = "..\..\Du_lieu\Day_so.xml"
        a = Doc_day_so(Duong_dan)
        Dim Chuoi As String = "Dãy số : " & Chuoi_day_so(a)
        Console.Write(Chuoi)
        Console.ReadLine()
    End Sub
    Public Function Doc_day_so(ByVal Duong_dan As String) As ArrayList
        Dim Kq As New ArrayList
        Dim Tai_lieu As New XmlDocument
        Tai_lieu.Load(Duong_dan) ' Chưa xử lý lỗi
        Dim Goc As XmlElement = Tai_lieu.DocumentElement
        For Each Nut As XmlElement In Goc.ChildNodes
            Kq.Add(Nut.GetAttribute("Gia_tri"))
        Next
        Return Kq
    End Function
    Public Function Chuoi_day_so(ByVal a As ArrayList) As String
        Dim Kq As String = ""
        For Each So As Integer In a
            Kq &= So & " "
        Next
        Return Kq
    End Function
End Module
```

*** Ghi phân số**

Cách 1 : Chỉ sử dụng hàm Main

```
Imports System.Xml
Module Ghi_phan_so_Main
    Public Sub Main()
        Dim Tu_so, Mau_so As Integer
        'Nhập liệu
        Console.Write("Tử số:")
        Tu_so = Integer.Parse(Console.ReadLine) ' Chưa kiểm tra
        Console.Write("Mẩu số:")
        Mau_so = Integer.Parse(Console.ReadLine) ' Chưa kiểm tra

        ' Tạo tài liệu Xml \
        Dim Tai_lieu As New XmlDocument
        Dim Goc As XmlElement = Tai_lieu.CreateElement("PHAN_SO")
        Goc.SetAttribute("Tu_so", Tu_so)
        Goc.SetAttribute("Mau_so", Mau_so)
        Tai_lieu.AppendChild(Goc)
        'Ghi
        Dim Duong_dan As String = "..\..\Du_lieu\Phan_so.xml"
```

```
        Tai_lieu.Save(Duong_dan)
        Console.ReadLine()
    End Sub
End Module
==============================
Cách 2 :  Sử dụng kiểu & hàm tự định nghĩa
Imports System.Xml
Module Ghi_phan_so_Kieu_Ham
    Structure PHAN_SO
        Public Tu_so As Integer
        Public Mau_so As Integer
    End Structure
    Public Function Nhap_phan_so() As PHAN_SO
        Dim Kq As PHAN_SO
        Console.Write("Tử số")
        Kq.Tu_so = Integer.Parse(Console.ReadLine) ' Chưa kiểm tra
        Console.Write("Mẩu số")
        Kq.Mau_so = Integer.Parse(Console.ReadLine) ' Chưa kiểm tra
        Return Kq
    End Function
    Public Function Ghi_phan_so(ByVal Ps As PHAN_SO, ByVal Duong_dan As String) As Boolean
        Dim Kq As Boolean = True
        Dim Tai_lieu As New XmlDocument
        Dim Goc As XmlElement = Tai_lieu.CreateElement("PHAN_SO")
        Goc.SetAttribute("Tu_so", Ps.Tu_so)
        Goc.SetAttribute("Mau_so", Ps.Mau_so)
        Tai_lieu.AppendChild(Goc)
        Tai_lieu.Save(Duong_dan)
        Return Kq
    End Function
    Public Sub Main()
        Dim Ps As PHAN_SO
        Ps = Nhap_phan_so()
        Dim Duong_dan As String = "..\..\Du_lieu\Phan_so.xml"
        Ghi_phan_so(Ps, Duong_dan)
        Console.ReadLine()
    End Sub
End Module
```

- Ghi phân số VB.NET

```
Cách 1 :  Chỉ sử dụng hàm Main
Imports System.Xml
Module Ghi_phan_so_Main
    Public Sub Main()
        Dim Tu_so, Mau_so As Integer
        'Nhập liệu
        Console.Write("Tử số:")
        Tu_so = Integer.Parse(Console.ReadLine) ' Chưa kiểm tra
        Console.Write("Mẩu số:")
```

```
        Mau_so = Integer.Parse(Console.ReadLine) ' Chưa kiểm tra

        ' Tạo tài liệu Xml \
        Dim Tai_lieu As New XmlDocument
        Dim Goc As XmlElement = Tai_lieu.CreateElement("PHAN_SO")
        Goc.SetAttribute("Tu_so", Tu_so)
        Goc.SetAttribute("Mau_so", Mau_so)
        Tai_lieu.AppendChild(Goc)
        'Ghi
        Dim Duong_dan As String = "..\..\Du_lieu\Phan_so.xml"
        Tai_lieu.Save(Duong_dan)
        Console.ReadLine()
    End Sub
End Module
==============================
Cách 2 :  Sử dụng kiểu & hàm tự định nghĩa
Imports System.Xml
Module Ghi_phan_so_Kieu_Ham
    Structure PHAN_SO
        Public Tu_so As Integer
        Public Mau_so As Integer
    End Structure
    Public Function Nhap_phan_so() As PHAN_SO
        Dim Kq As PHAN_SO
        Console.Write("Tử số")
        Kq.Tu_so = Integer.Parse(Console.ReadLine) ' Chưa kiểm tra
        Console.Write("Mẩu số")
        Kq.Mau_so = Integer.Parse(Console.ReadLine) ' Chưa kiểm tra
        Return Kq
    End Function
    Public Function Ghi_phan_so(ByVal Ps As PHAN_SO, ByVal Duong_dan As String) As Boolean
        Dim Kq As Boolean = True
        Dim Tai_lieu As New XmlDocument
        Dim Goc As XmlElement = Tai_lieu.CreateElement("PHAN_SO")
        Goc.SetAttribute("Tu_so", Ps.Tu_so)
        Goc.SetAttribute("Mau_so", Ps.Mau_so)
        Tai_lieu.AppendChild(Goc)
        Tai_lieu.Save(Duong_dan)
        Return Kq
    End Function
    Public Sub Main()
        Dim Ps As PHAN_SO
        Ps = Nhap_phan_so()
        Dim Duong_dan As String = "..\..\Du_lieu\Phan_so.xml"
        Ghi_phan_so(Ps, Duong_dan)
        Console.ReadLine()
    End Sub
End Module
```

- Ghi phân số C#

Cách 1 : Chỉ sử dụng hàm Main

```
using System;
using System.Xml;
public class Ghi_phan_so_Main
{
   public static void Main()
   {
      int Tu_so, Mau_so;
      // Nhập liệu
      Console.Write ("Tử số:");
      Tu_so =int.Parse (Console.ReadLine() ) ; // Chưa xử lý lỗi
      Console.Write("Mẩu số:");
      Mau_so =int.Parse (Console.ReadLine() ) ; // Chưa xử lý lỗi

      // Tạo nội dung tài liệu Xml
      XmlDocument Tai_lieu = new XmlDocument();
      XmlElement Goc=Tai_lieu.CreateElement ("PHAN_SO");
      Goc.SetAttribute ("Tu_so",Tu_so.ToString ());
      Goc.SetAttribute ("Mau_so",Mau_so.ToString ());
      Tai_lieu.AppendChild (Goc) ;

      // Ghi tài liệu Xml
      String Duong_dan = "..\\..\\Du_lieu\\Phan_so.xml";
      Tai_lieu.Save (Duong_dan) ;

      Console.ReadLine();
   }
}
```

==========

Cách 2 : Sử dụng kiểu & hàmtự định nghĩa

```
using System;
using System.Xml;
public class Ghi_phan_so_Kieu_Ham
{
   public struct PHAN_SO
   {
      public int Tu_so; // > 0
      public int Mau_so; // > 0
   }
   public static void Main()
   {
      PHAN_SO Ps;
      Ps = Nhap_phan_so();
      String Duong_dan = "..\\..\\Du_lieu\\Phan_so.xml";
      if (Ghi_phan_so(Ps, Duong_dan))
         Console.Write("Đã ghi ");
      else
```

```
            Console.Write("Lỗi khi ghi ");
        Console.ReadLine();
    }
    public static PHAN_SO Nhap_phan_so()
    {
        PHAN_SO Kq;
        Console.Write("Tử số:");
        Kq.Tu_so = int.Parse(Console.ReadLine()); // Chưa xử lý lỗi
        Console.Write("Mẩu số:");
        Kq.Mau_so = int.Parse(Console.ReadLine()); // Chưa xử lý lỗi
        return Kq;
    }
    public static Boolean Ghi_phan_so(PHAN_SO Ps, String Duong_dan )
    {
        Boolean Kq = true;
        XmlDocument Tai_lieu = new XmlDocument();
        XmlElement Goc = Tai_lieu.CreateElement("PHAN_SO");
        Goc.SetAttribute("Tu_so", Ps.Tu_so.ToString());
        Goc.SetAttribute("Mau_so", Ps.Mau_so.ToString());
        Tai_lieu.AppendChild(Goc);

        // Ghi tài liệu Xml

        Tai_lieu.Save(Duong_dan);

        return Kq;
    }
}
```

*** Ghi đường tròn**

```
Imports System.Xml
Module Ghi_duong_tron
    Structure DIEM
        Public x As Double
        Public y As Double
    End Structure
    Structure DUONG_TRON
        Public Tam As DIEM
        Public Ban_kinh As Double
    End Structure
    Public Function Nhap_duong_tron() As DUONG_TRON
        Dim Kq As DUONG_TRON
        Console.Write("Hoành độ tâm:")
        Kq.Tam.x = Double.Parse(Console.ReadLine) ' Chưa kiểm tra
        Console.Write("Tung độ tâm:")
        Kq.Tam.y = Double.Parse(Console.ReadLine) ' Chưa kiểm tra
        Console.Write("Bán kính :")
        Kq.Ban_kinh = Double.Parse(Console.ReadLine) ' Chưa kiểm tra
        Return Kq
```

```
    End Function
    Public Function Ghi_duong_tron(ByVal Dt As DUONG_TRON, ByVal Duong_dan As String) As
Boolean
        Dim Kq As Boolean = True
        Dim Tai_lieu As New XmlDocument
        Dim Goc As XmlElement = Tai_lieu.CreateElement("DUONG_TRON")
        Goc.SetAttribute("Ban_kinh", Dt.Ban_kinh)
        Tai_lieu.AppendChild(Goc)
        Dim Nut As XmlElement = Tai_lieu.CreateElement("DIEM")
        Nut.SetAttribute("x", Dt.Tam.x)
        Nut.SetAttribute("y", Dt.Tam.y)
        Goc.AppendChild(Nut)
        Tai_lieu.Save(Duong_dan)
       Return Kq
    End Function
    Public Sub Main()
        Dim Dt As DUONG_TRON
        Dt = Nhap_duong_tron()
        Dim Duong_dan As String = "..\..\Du_lieu\Duong_tron.xml"
        Ghi_duong_tron(Dt, Duong_dan)
        Console.ReadLine()
    End Sub
End Module
```

*** Ghi dãy số**

```
Imports System.Xml
Module Ghi_day_so

    Public Function Nhap_day_so() As ArrayList
        Dim Kq As New ArrayList
        Console.Write("Dãy số")
        Dim Chuoi As String = Console.ReadLine
        Dim M As String() = Chuoi.Split(",") ' Chưa kiểm tra
        For Each Con As String In M
            Kq.Add(Integer.Parse(Con)) ' Chưa kiểm tra
        Next
        Return Kq
    End Function
    Public Function Ghi_day_so(ByVal a As ArrayList, ByVal Duong_dan As String) As Boolean
        Dim Kq As Boolean = True
        Dim Tai_lieu As New XmlDocument
        Dim Goc As XmlElement = Tai_lieu.CreateElement("DAY_SO")
        Tai_lieu.AppendChild(Goc)
        For Each So As Integer In a
            Dim Nut As XmlElement = Tai_lieu.CreateElement("SO")
            Nut.SetAttribute("Gia_tri", So)
            Goc.AppendChild(Nut)
        Next
        Tai_lieu.Save(Duong_dan)
```

```
        Return Kq
    End Function
    Public Sub Main()
        Dim a As ArrayList
        a = Nhap_day_so()
        Dim Duong_dan As String = "..\..\Du_lieu\Day_so.xml"
        Ghi_day_so(a, Duong_dan)
        Console.ReadLine()
    End Sub
End Module
```

V. Bài tập

Bài tập
Giới thiệu chung về các bài tập

1. Biểu diễn thông tin với Xml

* Tam giác

Yêu cầu :
Biểu diễn thông tin tam giác ABC với A(1,0) , B(-8,3), C(4,4) với Xml

Bài giải :

```
<TAM_GIAC>
  <DIEM Ten="A" x="1" y="0" />
  <DIEM Ten="B" x="-8" y="3" />
  <DIEM Ten="C" x="4" y="4" />
</TAM_GIAC>
```

* Đa thức

Yêu cầu :
Biểu diễn thông tin đa thức
P(x) = 4x5 -7x3 +2x2 +4 với Xml
Bài giải :

```
<DA_THUC  Ten="P"  Bien_so="x">
 <DON_THUC  He_so="4"   So_mu="5" />
 <DON_THUC  He_so="-7"  So_mu="3" />
 <DON_THUC  He_so="2"   So_mu="2" />
 <DON_THUC  He_so="4"   So_mu="0" />
</DA_THUC>
```

* Ma trận

Yêu cầu :
Biểu diễn thông tin ma trận các số nguyên

```
1   2   -4   6
8   0    7   12
0   9   11   -3
```

với Xml

Bài giải :

Cách 1 :

```
<MA_TRAN>
  <SO y="0" x="0" Gia_tri="1" />
  <SO y="0" x="1" Gia_tri="2" />
  <SO y="0" x="2" Gia_tri="-4" />
  <SO y="0" x="3" Gia_tri="6" />

  <SO y="1" x="0" Gia_tri="8" />
  <SO y="1" x="1" Gia_tri="0" />
  <SO y="1" x="2" Gia_tri="7" />
  <SO y="1" x="3" Gia_tri="12" />

  <SO y="2" x="0" Gia_tri="0" />
  <SO y="2" x="1" Gia_tri="9" />
  <SO y="2" x="2" Gia_tri="11" />
  <SO y="2" x="3" Gia_tri="-3" />

</MA_TRAN>
```

*** Bảng điểm danh**

Yêu cầu :
Biểu diễn thông tin bảng điểm danh

Bảng điểm danh học sinh lớp !0A
Tháng 11/2007

Họ tên Vắng có phép Vắng không phép
...
....

với Xml

Bải giải :

```
<BANG_DIEM_DANH  Ten_lop="10A"   Thang="11"  Nam="2007">
 <HOC_SINH  Ho_ten="Trần văn Long" Vang_co_phep="0"  Vang_khong_phep="2" />
 <HOC_SINH  Ho_ten="Lê thị bé Nhỏ"  Vang_co_phep="0"  Vang_khong_phep="0" />
 <HOC_SINH  Ho_ten="Nguyễn văn A" Vang_co_phep="0" Vang_khong_phep="0" />
 <HOC_SINH  Ho_ten="Hồ thị Đẹp"      Vang_co_phep="0"  Vang_khong_phep="0" />
 <HOC_SINH  Ho_ten="Lê văn Tốt"      Vang_co_phep="1"  Vang_khong_phep="4" />
</BANG_DIEM_DANH>
```

*** Bàn cờ carô**

Yêu cầu :
Biểu diễn thông tin về các quân cờ của một bàn cờ carô

Hướng dẫn :

Sử dụng tập tin Caro.xml với thẻ gốc BAN_CO
(có thẻ chỉ có 1 thuộc tính So_dong
hay chỉ có 1 thuộc tính So_cot
hay cả 2 thuộc tính So_dong, So_cot
hay không có thuộc tính nào hết !!!)

Thẻ BAN_CO gồm nhiều thẻ QUAN_CO
(Các thuộc tính của QUAN_CO tương ứng thông tin về quấn cờ)

*** Sơ đồ ghế**

Yêu cầu :
Biểu diễn thông tin về sơ đồ ghế của một sân khấu biểu diễn bất ký (Ví dụ Idecaf, Hòa Bình, Nhà văn hóa quận 1, Nhà hát thành phố, v.v..)

Hướng dẫn :

Sử dụng tập tin So_do_ghe.xml với thẻ gốc SO_DO
(có thẻ chỉ có 1 thuộc tính So_dong
hay chỉ có 1 thuộc tính So_cot
hay cả 2 thuộc tính So_dong, So_cot
hay không có thuộc tính nào hết !!!)

Thẻ SO_DO gồm nhiều thẻ GHE
(Các thuộc tính của GHE tương ứng thông tin của ghế)

*** Bảng lịch tàu thống nhất**

Yêu cầu :
Biểu diễn thông tin về bảng lịch tàu thống nhất

Ghi chú :
Đây là bài tập khó dành cho các sinh viên tự xếp mình vào loại khá, giỏi

*** Sơ đồ các chuyến bay nội địa**

Yêu cầu :
Biểu diễn thông tin về sơ đồ các chuyến bay nội địa của VN

Ghi chú :
Đây là bài tập khó dành cho các sinh viên tự xếp mình vào loại giỏi

2. Lập trình với DOM

*** Tính giá trị đơn thức**

Yêu cầu :
Viết chương trình tính giá trị đơn thức P(x) =axn (có thông tin đuợc lưu trữ duới dạng tập tin Xml) với x0 cho trước

Hướng dẫn thiết kế chương trình :

Biểu diễn thông tin :
- Bô nhớ phụ :
  Sử dụng tập tin Xml với một thẻ duy nhất DON_THUC (gồm 2 thuộc tính)
- Bô nhớ chính :
  Sử dụng 3 biến
    P : DON_THUC
    x0 : Số thực
    Kq : Số thực
  với DON_THUC là kiểu cấu trúc gồm 2 thành phần

Biểu diễn xử lý :
Sử dụng các hàm sau
Hàm xử lý chính Main
  Khai báo các biến P,x0,Kq
  P = Dữ liệu đọc từ tập tin Xml
  x0 = Dữ liệu nhập từ người dùng
  Kq = Giá trị của P với x0
  Xuất Kq

Hàm Doc_don_thuc
  Tham số : Chuỗi Duong_dan đến tấp tin Xml
  Kết quả : Đon thức tương ứng
Hàm Nhap_so_thuc
  Tham số : Chuỗi Ghi_chu
  Kết quả : Số thực nhập từ người dùng
Ham Gia_tri
  Tham số : Đơn thức P, Giá trị x0
  Kết quả : Giá trị của P với x0

VB.NET

```
Imports System.Xml
Module Tinh_gia_tri_don_thuc_Kieu_Ham
    Structure DON_THUC
        Public He_so As Double
        Public So_mu As Integer ' >=0
    End Structure
    Public Sub Main()
        Dim Duong_dan As String = "..\..\Du_lieu\Don_thuc.xml"
        Dim P As DON_THUC
        Dim x0 As Double

        Dim Kq As Double

        ' Đọc đơn thức
        ' Nhập x0

        '  Tính Kq
```

```
        ' Xuất Kq
    End Sub
    Public Function Doc_don_thuc(ByVal Duong_dan As String) As DON_THUC
        Dim Kq As DON_THUC
        ' Đọc dữ liệu của tập tin vào đối tượng Tai_lieu XmlDocument

        ' Gán giá trị của kết quả từ gốc của Tai_lieu
        Return Kq
    End Function
    Public Function Gia_tri(ByVal P As DON_THUC, ByVal x0 As Double) As Double
        Dim Kq As Double
        Kq = P.He_so * Math.Pow(x0, P.So_mu)
        Return Kq
    End Function
    Public Function Chuoi_don_thuc(ByVal P As DON_THUC) As String
        Dim Kq As String = ""
        Kq &= P.He_so & "x^" & P.So_mu ' Chưa xem xét các trường hợp đặc biệt
        Return Kq
    End Function
    Public Function Nhap_so_thuc(ByVal Ghi_chu As String) As Double
        Dim Kq As Double
        Console.Write(Ghi_chu)
        Kq = Double.Parse(Console.ReadLine) ' Chưa xử lý lỗi
        Return Kq
    End Function
End Module

C#
using System;
using System.Xml;
public class Tinh_gia_tri_don_thuc_Kieu_Ham
{
    public struct DON_THUC
    {
        public Double He_so;
        public int So_mu; // > 0
    }
    public static void Main()
    {
        String Duong_dan = "..\\..\\Du_lieu\\Don_thuc.xml";

        DON_THUC P;
        Double x0;

        Double Kq;

         // Đọc đơn thức P
        // Nhập giá trị x0
```

```
        // Tính Kq

       // Xuất Kq
    }
    public static DON_THUC Doc_don_thuc(String Duong_dan )
    {
        DON_THUC Kq;
        // Đọc dữ liệu của tập tin vào đối tượng Tai_lieu XmlDocument

        // Gán giá trị cho Kq từ nút gốc của Tai_lieu
        return Kq;
    }
    public static Double Gia_tri(DON_THUC P, Double x0 )
    {
        Double Kq;
        Kq = P.He_so * Math.Pow(x0, P.So_mu);
        return Kq;
    }
    public static String Chuoi_don_thuc(DON_THUC P)
    {
        String Kq = "";
        Kq += P.He_so;
        Kq += "x^";
        Kq += P.So_mu; // Chưa xem xét các trường hợp đặc biệt
        return Kq;
    }
    public static Double Nhap_so_thuc(String Ghi_chu)
    {
        Double Kq;
        Console.Write(Ghi_chu);
        Kq = Double.Parse(Console.ReadLine()); // Chua xem xét người dùng nhập chuỗi
        return Kq;
    }
}
```

*** Giải phương trình bậc 2**

Yêu cầu :
Viết chương trình giải phương trình bậc 2
ax2+bx+c=0 (a khác 0)
có thông tin được lưu trữ duới dạng tập tin Xml

Hướng dẫn thiết kế chương trình :

Biểu diễn thông tin
- Bô nhớ phụ :
  Sử dụng tập tin Xml với một thẻ duy nhất TAM_THUC (gồm 3 thuộc tính)
- Bô nhớ chính :
  Sử dụng 3 biến
    P : TAM_THUC

Ng : Mảng 1 chiều(dãy) các số thực với kích thước tối đa 2
với TAM_THUC là kiểu cấu trúc gồm 3 thành phần
Biểu diễn xử lý :
Sử dụng các hàm sau
Hàm xử lý chính Main
Khai báo các biến P,Ng
P = Tam thức đọc từ tập tin Xml
Ng = Nghiệm của P
Chuoi = Chuỗi tương ứng nghiệm Ng
Xuất Chuoi
Hàm Doc_tam_thuc
Tham số : Chuỗi Duong_dan đến tấp tin Xml
Kết quả : Tam thức tương ứng
Hàm Giai_phuong_trinh
Tham số : Tam thức P
Kết quả : Mảng 1 chiều(dãy) các số thực với kích thước tối đa 2
Ham Chuoi_nghiem
Tham số : Mảng 1 chiều(dãy) các số thực với kích thước tối đa 2
Kết quả : Chuỗi tương ứng

VB.NET

```
Imports System.Xml
Module Giai_phuong_trinh_bac_2_Kieu_Ham
    Structure TAM_THUC
        Public a As Double ' khác 0
        Public b As Double
        Public c As Double
    End Structure
    Public Sub Main()
        Dim Duong_dan As String = "..\..\Du_lieu\Tam_thuc.xml"
        Dim P As TAM_THUC

        Dim Ng As New ArrayList

        ' Đọc tam thức P

        ' Tính nghiệm Ng

        ' Xuất nghiệm Ng
    End Sub
    Public Function Doc_tam_thuc(ByVal Duong_dan As String) As TAM_THUC
        Dim Kq As TAM_THUC
        Dim Tai_lieu As New XmlDocument
        Tai_lieu.Load(Duong_dan) ' Chưa xử lý lỗi
        Dim Goc As XmlElement = Tai_lieu.DocumentElement
        Kq.a = Goc.GetAttribute("a")
        Kq.b = Goc.GetAttribute("b")
        Kq.c = Goc.GetAttribute("c")
        Return Kq
    End Function
```

```
    Public Function Giai_phuong_trinh(ByVal P As TAM_THUC) As ArrayList
        Dim Kq As New ArrayList
        Dim Delta As Double = P.b * P.b - 4 * P.a * P.c
      ' Tính nghiệm và đưa vào Kq dựa trên xét dấu của Delta
        Return Kq
    End Function
    Public Function Chuoi_tam_thuc(ByVal P As TAM_THUC) As String
        Dim Kq As String = ""
        Kq &= String.Format("{0}x^2 + {1}x +{2}", P.a, P.b, P.c)
        ' Chưa xem xét trình bày số âm, 0
        Return Kq
    End Function
    Public Function Chuoi_nghiem(ByVal Ng As ArrayList) As String
        Dim Kq As String = ""
        If Ng.Count = 0 Then
            Kq = "Phương trình vô nghiệm"
        ElseIf Ng.Count = 1 Then
            Kq = String.Format("Phương trình có nghiệm kép x1=x2={0:F2}", Ng(0))
        ElseIf Ng.Count = 2 Then
            Kq = String.Format("Phương trình có 2 nghiệm x1={0:F2} , x2={1:F2}", Ng(0), Ng(1))
        End If
        Return Kq
    End Function
End Module
```

C#

```
using System;using System.Xml;using System.Collections; // Khai báo thư viện hàm cho phép sử dụng
ArrayList
public class Giai_phuong_trinh_bac_2_kieu_ham
{
    public struct TAM_THUC
    {
        public Double a; // Khác 0
        public Double b;
        public Double c;
    }
    public static void Main()
    {
        String Duong_dan = "..\\..\\Du_lieu\\Tam_thuc.xml";

        TAM_THUC P;

        ArrayList Ng;

          // Đọc tam thức P

         // Tính nghiệm Ng

         // Xuất nghiệm Ng
     }
```

```
    public static TAM_THUC Doc_tam_thuc(String Duong_dan)
    {
        TAM_THUC Kq;
        XmlDocument Tai_lieu = new XmlDocument();
        Tai_lieu.Load(Duong_dan); // Chưa xử lý lỗi
        XmlElement Goc = Tai_lieu.DocumentElement;
        Kq.a = Double.Parse(Goc.GetAttribute("a"));
        Kq.b = Double.Parse(Goc.GetAttribute("b"));
        Kq.c = Double.Parse(Goc.GetAttribute("c"));
        return Kq;
    }
    public static ArrayList Giai_phuong_trinh(TAM_THUC P)
    {
        ArrayList Kq = new ArrayList();
        Double Delta = P.b * P.b - 4 * P.a * P.c;

       // Tính nghiệm và đưa vào Kq dựa trên xét đấu của Delta
        return Kq;
    }
    public static String Chuoi_tam_thuc(TAM_THUC P)
    {
        String Kq = "";
        Kq = String.Format("{0}x^2 + {1}x + {2}", P.a, P.b, P.c);
          // Chưa xem xét các trường hợp đặc biệt
        return Kq;
    }
    public static String Chuoi_nghiem(ArrayList Ng )
    {
        String Kq = "";
        if (Ng.Count == 0)
            Kq = "Phương trình vô nghiệm";
        else if (Ng.Count == 1)
            Kq = String.Format("phương trình có nghiệm kép x1=x2={0:F2}", Ng[0]);
        else if (Ng.Count == 2)
            Kq = String.Format("phương trình có 2 nghiệm x1{0:F2} , x2={1:F2}", Ng[0],Ng[1]);
        return Kq;
    }
}
```

*** Tính giá trị hàm số**

Yêu cầu :
Viết chương trình tính giá trị hàm số f(x)
  f(x) = (a1x2+b1x+c1) /(a2x2+b2x+c2)
có thông tin được lưu trữ duới dạng tập tin Xml
với x0 cho trước

Hướng dẫn thiết kế chương trình :

Biểu diễn thông tin :

- Bô nhớ phụ :
  Sử dụng tập tin Xml với thẻ gốc HAM_SO bao gồm 2 thẻ conTAM_THUC
- Bô nhớ chính :
  Sử dụng 3 biến
    f : HAM_SO
    x0 : Số thực
    Kq : Số thực
  với HAM_SO là kiểu cấu trúc gồm 2 thành phần tương ứng với 2 tam thức P,Q

Biểu diễn xử lý :
Sử dụng các hàm sau
Hàm xử lý chính Main
  Khai báo các biến f,x0,Kq
  f = Hàm số đọc từ tập tin Xml
  x0 = Dữ liệu nhập từ người dùng
  Kq = Giá trị của f với x0
  Xuất Kq

Hàm Doc_ham_so
  Tham số : Chuỗi Duong_dan đến tấp tin Xml
  Kết quả : Hàm số tương ứng
Hàm Nhap_so_thuc
  Tham số : Chuỗi Ghi_chu
  Kết quả : Số thực nhập từ người dùng
Ham Gia_tri
  Tham số : Hàm số f , Giá trị x0
  Kết quả : Giá trị của f với x0

*** Tính tiền thuê phòng**

Yêu cầu :
Viết chương trình tính tiến thuê phòng khi biết số ngày thuê lọai phòng dựa trên bảng đơn giá đã được lưu trữ dưới dạng tập tin Xml

Hướng dẫn thiết kế chương trình :

Biểu diễn thông tin :
- Bô nhớ phụ :
  Sử dụng tập tin Bang_don_gia.Xml với thẻ gốc BANG_DON_GIA bao gồm các thẻ con LOAI_PHONG (có các thuộc tính Ten, Don_gia)
- Bô nhớ chính :
  Sử dụng 3 biến
    Bdg: Mảng 1 chiều(dãy) các LOAI_PHONG
    So_ngay_thue : Số nguyên >0
    ChI_so : Số nguyên tương ứng số thứ tự của loại phòng thuê trong Bdg
      (Chi_so>=0 và Chi_so < Số các loại phòng)
    Tien_phai_tra : Số nguyên
  với LOAI_PHONG là kiểu cấu trúc

Biểu diễn xử lý :
Sử dụng các hàm sau
Hàm xử lý chính Main

Khai báo các biến Bdg,So_ngay_thue,Chi_so,Tien_phai_tra
Bdg = Bảng đơn giá đọc từ tập tin Xml
So_ngay_thue = Dữ liệu nhập từ người dùng
ChI_so= Dữ liệu nhập từ người dùng

Tien_phai_tra = Tiền phải trả với So_ngay_thue và loại phòng Bdg[Chi_so]
Xuất Tien_phai_tra

Hàm Doc_bang_don_gia
Tham số : Chuỗi Duong_dan đến tấp tin Xml
Kết quả : Bảng đơn giá tương ứng
Hàm Nhap_so_nguyen
Tham số : Chuỗi Ghi_chu , Can_duoi, Can_tren
Kết quả : Số nguyên n nhập từ người dùng với Can_duoi<=n<=Can_tren
Ham Tinh_tien
Tham số : Bảng đơn giá , Số ngày thuê, Chỉ số của loại phòng
Kết quả : Tiền phải trả

*** Đổi ngoại tệ**

Yêu cầu :
Viết chương trình đổi ngoại tệ khi biết số tiền cần đổi, loại ngoại tệ, hình thức đổi dựa trên bảng tỷ giá đã được lưu trữ dưới dạng tập tin Xml

Hướng dẫn thiết kế chương trình :

Biểu diễn thông tin :
- Bô nhớ phụ :
Sử dụng tập tin Bang_ty_gia.Xml với thẻ gốc BANG_TY_GIA bao gồm các thẻ con NGOAI_TE (có các thuộc tính Ten, Mua_tien_mat,Mua_chuyen_khoan, Ban)
- Bô nhớ chính :
Sử dụng 3 biến
Btg: Mảng 1 chiều(dãy) các NGOAI_TE
So_tien_doi : Số nguyên >0
Hinh_thuc_doi : Chuỗi với 1 trong 3 giá trị "MTM", "MCK", "BAN"
Ngoai_te_doi : Ngoại tệ cần đổi
Tien_doi_duoc : Số thực

với NGOAI_TE là kiểu cấu trúc
Biểu diễn xử lý :
Sử dụng các hàm sau
Hàm xử lý chính Main
Khai báo các biến Btg,So_tien_doi, Ngoai_te_doi,Tien_doi_duoc
Btg = Bảng tỷ giá đọc từ tập tin Xml
So_tien_doi = Dữ liệu nhập từ người dùng
Ngoai_te_doi= Ngoại tệ được chọn từ người dùng
Hinh_thuc_doi=Hình thức đổi được chọn từ người dùng

Tien_doi_duoc = Tiền đổi được với
Bảng tỷ giá,Số tiền đổi, Ngoại tệ đổi và hình thức đổi

Xuất Tien_doi_duoc

Hàm Doc_bang_ty_gia
Tham số : Chuỗi Duong_dan đến tấp tin Xml
Kết quả : Bảng tỷ giá tương ứng
Hàm Nhap_so_nguyen
Tham số : Chuỗi Ghi_chu , Can_duoi, Can_tren
Kết quả : Số nguyên n nhập từ người dùng với Can_duoi<=n<=Can_tren
Hàm Chon_ngoai_te
Tham số : ???
Kết quả : Ngoại tệ được chọn
Hàm Chon_hinh_thuc
Tham số :
Kết quả : Hình thức đổi được chọn
Ham Doi_tien
Tham số : ???
Kết quả : Số tiền đổi được

*** Dò vé số**

Yêu cầu :
Viết chương trình dò vé số dựa trên kết quả xổ số đã được lưu trữ dưới dạng tập tin Xml

Hướng dẫn thiết kế chương trình :

Biểu diễn thông tin :
- Bô nhớ phụ :
Sử dụng tập tin Kqxs.Xml với thẻ gốc KET_QUA bao gồm các thẻ con GIAI (có các thuộc tính Ten, Tien_thuong) , thẻ GIAI bao gồm các thẻ con SO (với thuộc tính Gia_tri)
- Bô nhớ chính :
Sử dụng 3 biến
Kqxs: Mảng 1 chiều(dãy) các GIAI
Ve_so : Số nguyên
Tien_thuong : Số nguyên
với GIAI là kiểu cấu trúc tương ứng cấu trúc tập tin Xml

Biểu diễn xử lý :
Sử dụng các hàm sau
Hàm xử lý chính Main
Khai báo các biến Kqxs ,Ve_so, Tien_thuong
Kqxs = Kết quả xổ số đọc từ tập tin Xml
Ve_so = Dữ liệu nhập từ người dùng

Tien_thuong= Tổng tiền thưởng với kết quả xổ số và Ve_so
(Một vé số có thể trúng thưởng nhiều giải)
Xuất Tien_thuong

Hàm Doc_ket_qua_xo_so
Hàm Nhap_so_nguyen
Ham Tinh_tien_thuong

*** Tính tổng trở**

Yêu cầu :
Viết chương trình tính tổng trở của mạch điện chỉ gồm các điện trở (có thông tin lưu trữ dưới dạng tập tin Xml)

Ghi chú : Bài tập khó dành cho các sinh viên tự xếp mình vào loại khá giỏi

Gọi ý :
Kỹ thuật đệ qui
hay
Kỹ thuật lập trình hướng đối tượng với kỹ thuật đa xạ

*** Sơ đồ chuyến bay**

Yêu cầu :
Viết chương xuất sơ đồ các chuyến bay với thông tin đã được lưu trữ trên một tập tin Xml

Ghi chú :
Bài tập khó dành cho các sinh viên tự xếp mình vào loại giỏi

Chương 2 : Đặc tả nội dung & cấu trúc tài liệu XML

Mục tiêu :
- Trình bày các khả năng của công nghệ Xml khi được ứng dụng trong giai đoạn thiết kế thành phần dữ liệu
- Rèn luyện kỹ năng đặc tả tài liệu Xml biểu diễn thông tin của đối tượng trong thực tế
- Bước đầu ứng dụng Xml trong ứng dụng nhỏ, đơn giãn

I. Mở đầu

1. Nội dung tài liệu XML

Khái niệm về nội dung tài liệu Xml
Nội dung của tài liệu XML bao gồm 2 phần

Nội dung chính

Hệ thống các thẻ đánh dấu (có hay không có nội dung) tương ứng với các thông tin cần biểu diễn

Nội dung phụ

Hệ thống các thẻ khác có ý nghĩa bổ sung, tăng cường một số thông tin về tài liệu XML. Các thẻ này có tác dụng giúp cho việc sử dụng, xử lý trên tài liệu XML tốt hơn trong một số trường hợp nhất định

Các thẻ bên trong nội dung phụ bao gồm loại sau
- Thẻ khai báo tham số
- Thẻ chỉ thị xử lý
- Thẻ ghi chú
- Thẻ CDATA
- Thẻ khai báo cấu trúc
- Thẻ khai báo thực thể

Các thẻ khai báo tham số, thẻ chỉ thị xử lý,thẻ ghi chú và thẻ CDATA có ý nghĩa sử dụng đơn giản sẽ được diễn giải chi tiết ngay trong phần sau

Thẻ khai báo cấu trúc liên quan đến cấu trúc tài liệu XML với nhiều khái niệm khác. Thẻ này sẽ được trình bày chi tiết trong phần "Đặc tả cấu trúc với DTD"

Thẻ khai báo thực thể liên quan đến nhiều kỹ thuật khác nhau có thẻ áp dụng trên tài liệu XML. Thẻ này sẽ được trình bày chi tiết trong phần "Kỹ thuật đặc tả nội dung tài liệu XML"

* Thẻ khai báo tham số

Thẻ khai báo tham số
Thẻ khai báo tham số cho phép mô tả thêm một số thông tin chung (tham số) về tài liệu XML ngoài các thông tin đã biểu diễn trong nội dung chính.

Dạng khai báo chung như sau

```
<?xml Ten_1="Gia_tri_1" Ten_2="Gia_tri_2" …  ?>
```

Ten_1, Ten_2, … là các tên của các tham số và Gia_tri_1, Gia_tri_2, … là các giá trị tương ứng. Cho đến hiện nay có 3 tham số được dùng là version, encoding, và standalone. Tham số version bắt buộc phải có nếu các tham số khác được sử dụng

Tham số version : Khai báo về phiên bản của định chuẩn XML được sử dụng
Ví dụ :

Tài liệu XML thuộc đỉnh chuẩn 1.0

<?xml version=”1.0” ?>

Tham số encoding : Khai báo về cách mã hóa các ký tự trong tài liệu
Ví dụ

Tài liệu XML sử dụng cách mã hóa Unicode ký hiệu utf-8

<?xml version=”1.0” encoding=”utf-8” ?>

Tài liệu XML sử dụng cách mã hóa Unicode ký hiệu utf-16

<?xml version=”1.0” encoding=”utf-16” ?>

Tham số standalone : Khai báo về liên kết của tài liệu XML và các tài liệu khác. Tham số này chỉ có 2 giá trị hợp lệ là “yes” , “no”. Giá trị định sẳn là “no”
Ví dụ :

Tài liệu XML có liên kết với các tài liệu khác

<?xml standalone=”yes” ?>

Tài liệu XML không có liên kết với các tài liệu khác

<?xml version=”1.0” standalone=”no” ?>

*** Thẻ chỉ thị xử lý**

Thẻ chỉ thị xử lý
Ý nghĩa chung của các thẻ chỉ thị xử lý là cho phép mô tả thêm một số thông tin (liên quan xử lý) về tài liệu XML có ý nghĩa riêng với một công cụ xử lý nào đó. Đây chính là một phương pháp cho phép mở rộng, bổ sung các xử lý riêng vào một lớp tài liệu XML cùng thuộc một hệ thống phân lớp nào đó

Dạng khai báo chung như sau

<? Bo_xu_ly Du_lieu ?>

Bo_xu_ly là ký hiệu của bộ xử lý sẽ tiến hành một số xử lý nào đó trên tài liệu XML . Du_lieu là thông tin được gởi đến Bo_xu_ly

Ví dụ:

<?xml-stylesheet type=”text/css” href=”Dinh_dang.css” ?>

Là thẻ chỉ thị cần xử lý định dạng thể hiện tài liệu XML với “chương trình định dạng ” theo ngôn ngữ css được lưu trữ bên trong tập tin Dinh_dang.css

Thẻ này sẽ có ý nghĩa với một số trình duyệt Web như IE (phiên bản 50. về sau), Netscape (phiên bản 6.0 về sau)

*** Thẻ ghi chú**

Thẻ ghi chú
Thẻ ghi chú cho phép bổ sung các thông tin ghi chú có ý nghĩa đối với con người và hoàn toàn không có ý nghĩa với các hệ thống xử lý tài liệu XML

Dạng khai báo chung như sau

<-- Nội dung ghi chú -->

*** Thẻ CDATA**

Thẻ CDATA
Thẻ CDATA có ý nghĩa yêu cầu các bộ phân tích tài liệu XML bỏ qua và không phân tích vào nội dung bên trong của thẻ này. Tác dụng của thẻ là cho phép sử dụng trực tiếp bên trong thẻ một số ký hiệu không được phép nếu sử dụng bên ngoài (ví dụ các ký tự "<" , ">" , …)

Dạng khai báo chung như sau

<![CDATA [Nội dung]]>

2. Cấu trúc tài liệu XML

Khái niệm về cấu trúc tài liệu XML
- Chỉ tương ứng cấu trúc của nội dung chính
- Cách thức tổ chức, sắp xếp của các thẻ (có hay không có nội dung) trong nội dung chính

Khái niệm về đặc tả cấu trúc tài liệu XML
- Mô tả ngắn gọn, chính xác cấu trúc tài liệu XML
- Mô tả ngắn gọn, chính xác cách thức tổ chức, sắp xếp của các thẻ

*** Ngôn ngữ đặc tả cấu trúc**

Có rất nhiều ngôn ngữ đặc tả được đề xuất để mô tả cấu trúc tài liệu Xml như DTD, XML Schema, XMl-Data, Schematron , RELAX NG, v,v.. .Trong số đó có 2 ngôn ngữ thông dụng là DTD, XML Schema

Đặc điểm của DTD
- Ra đời rất sớm
- Cho phép mô tả văn bản có cấu trúc bất kỳ
- Đơn giản, dễ học và sử dụng
- Chỉ cho phép đặc tả một số "kiểu dữ liệu đơn giản" trong nội dung chính của tài liệu XML

Đặc điểm của XML Schema
- Được đề xuất bởi W3C
- Chỉ áp dụng cho tài liệu XML
- Khó học và sử dụng so với DTD
- Cho phép đặc tả chi tiết về các "kiểu dữ liệu" được sử dụng trong nội dung chính của tài liệu XML

Ví dụ : Với tài liệu Xml

```
<?xml version="1.0" encoding="utf-8"?>
<PHAN_SO>
    <Tu_so> 4 </Tu_so>
    <Mau_so> 3 </Mau_so>
</PHAN_SO>
```

Đặc tả với DTD

```
<!DOCTYPE PHAN_SO [
<!ELEMENT PHAN_SO (Tu_so, Mau_so) >
<!ELEMENT Tu_so #PCDATA >
        <!-- Tu_so : Số nguyên  // >0      -->
<!ELEMENT Mau_so #PCDATA>
        <!-- Mau_so : Số nguyên  // >0   -->
]>
```

Đặc tả với Xml Schema

```
<?xmlversion="1.0"encoding="utf-8"?>
<xs:schemaid="PHAN_SO"
            targetNamespace="http://tempuri.org/PHAN_SO.xsd"
            xmlns:xs="http://www.w3.org/2001/XMLSchema">

      <xs:elementname="PHAN_SO"     type="PHAN_SO"/>

      <xs:complexTypename="PHAN_SO">
            <xs:sequence>
              <xs:elementname="Tu_so"type="SO_NGUYEN_DUONG"
                        minOccurs="1"maxOccurs="1"/>
              <xs:elementname="Mau_so"type=" SO_NGUYEN_DUONG "
                        minOccurs="1"maxOccurs="1" />
            </xs:sequence>
        </xs:complexType>

        <xs:simpleTypename="SO_NGUYEN_DUONG">
           <xs:restrictionbase="xs:int">
             <xs:minExclusivevalue="0"/>
           </xs:restriction>
        </xs:simpleType>
</xs:schema>
```

*** Sử dụng đặc tả cấu trúc**

Ý nghĩa của đặc tả cấu trúc

Có 2 trường hợp chính cần thiết sử dụng các tài liệu đặc tả cấu trúc

- Trường hợp 1 : Sử dụng cho việc trao đổi thông tin người – người
- Trường hợp 1 : Sử dụng cho việc trao đổi thông tin người – hệ thống xử lý

Trường hợp 1 là trường hợp thông dụng nhất, với trường hợp này tài liệu đặc tả cấu trúc

- Được sử dụng như phương tiện giao tiếp giữa các chuyên viên tin học có liên quan đến tài liệu XML tương ứng
- Có thể được lưu trữ theo bất kỳ định dạng nào thích hợp cho việc sử dụng (trình bày, xem báo cáo , v.v..)

Ví dụ : Có thể sử dụng các tài liệu đặc tả cấu trúc (DTD/ XML Schema trên) trong

- Hồ sơ thiết kế phần mềm hay giáo trình này (theo dạng tập tin của Microsoft Word)
- Tài liệu mô tả cách thức trao đổi thông tin giữa các chuyên viên tin cùng xây dựng các phần mềm bài tập phân số

Ghi chú :

- Đây là trường hợp sử dụng chính và thông dụng nhất
- Đây là trường hợp dễ sử dụng nhất vì không yêu cầu thật chặt chẽ về cú pháp. Nếu trong tài liệu đặc tả cấu trúc có sai sót một ít về cú pháp thì người sử dụng cũng có thể hiểu hay cũng có thể phát hiện và trao đổi lại với người tạo lập
- Với trường hợp này, tùy vào từng trường hợp cụ thể với các một số qui ước riêng mang tính cục bộ trong một nhóm người nào đó, có thể mở rộng các ngôn ngữ đặc tả cấu trúc hiện có để bổ sung thêm các từ vựng, cú pháp và ngữ nghĩa riêng

Trường hợp 2 chỉ được sử dụng khi

- Có hệ thống xử lý (phần mềm, hàm , đối tượng thư viện) "hiểu" và thực hiện các xử lý tương ứng nào đó với tài liệu đặc tả cấu trúc (xử lý thông dụng nhất là kiểm tra một tài liệu XML có theo đúng cấu trúc được mô tả trong tài liệu đặc tả cấu trúc hay không.)
- Thật sự có nhu cầu cần đến các xử lý của hệ thống xử lý nói trên

Ví dụ :

- Có thể sử dụng các tài liệu đặc tả cấu trúc (DTD/ XML Schema trên) với bộ phân tích XmlTextReader trong VB.NET để yêu cầu bộ phân tích này kiểm tra tính hợp lệ của tài liệu XML. Tuy nhiên, một cách tổng quát xử lý kiểm tra này không thật sự cần thiết !!!
- Với các ứng dụng thương mại điện tử việc trao đổi các tài liệu XML liên quan các nghiệp vụ thương mại (thông tin về các mặt hàng, đơn đặt hàng , phiếu giao hàng, v.v…) đặt ra nhu cầu thật sự về việc kiểm tra một tài liệu XML có đúng theo cấu trúc mong đợi hay không. Với ngữ cảnh này nhất thiết phải tạo lập và sử dụng các bộ phân tích cú pháp thích hợp để tiến hành kiểm tra tính hợp lệ và xử lý tương ứng

Ghi chú :

- Trường hợp này yêu cầu tài liệu đặc tả cấu trúc phải tuân thủ hoàn toàn theo ngôn ngữ đặc tả cấu trúc tương ứng, mọi sai sót về cú pháp sẽ không được bộ phân tích cú pháp chấp nhận
- Cần cân nhắc khi sử dụng tài liệu đặc tả cấu trúc trong trường hợp này vì một trong các đặc điểm quan trọng trong tiếp cận của XML là

"Cho phép đặc tả nội dung mà không nhất thiết đặc tả cấu trúc "

II. Một số kỹ thuật đặc tả nội dung

1. Sử dụng thẻ thực thể

Kỹ thuật sử dụng thẻ thực thể
Ý nghĩa chung các thẻ khai báo thực thể là cho phép tài liệu XML tham chiếu đến một tập hợp các giá trị đã chuẩn bị trước dưới dạng một tên gợi nhớ (tên thực thể).

Mỗi cách thức tham chiếu và "loại" của tập hợp giá trị được tham chiếu tương ứng với một ý nghĩa/mục tiêu (dạng sử dụng) riêng và sẽ yêu cầu dạng thẻ khai báo thực thể thích hợp.

Có 4 dạng sử dụng chính các thực thể

- Dạng 1 : Tham chiếu đến một chuỗi giá trị bên trong tài liệu XML đang xem xét
- Dạng 2 : Tham chiếu đến các ký tự đặc biệt được định nghĩa trước
- Dạng 3 : Tham chiếu đến một tập hợp các giá trị bên ngoài tài liệu
- Dạng 4 : Tham chiếu đến một tài liệu XML khác

Cách thức khai báo và sử dụng chung các thẻ khai báo thực thể (cho cả 4 dạng trên) như sau

Khai báo

```
<!DOCTYPE Ten_goc [
        Khai báo thực thể X
        Khai báo thực thể Y
] >
```

Sử dụng

&X; < -- Sử dụng tham chiếu của X -->

&Y; <-- Sử dụng tham chiếu của Y -->

*** Dạng 1**

Tham chiếu đến một chuỗi giá trị bên trong tài liệu XML đang xem xét
Ý nghĩa :
- Tăng cường tính dễ đọc của tài liệu XML
- Tăng cường tính dễ bảo trì của tài liệu XML

Dạng khai báo và sử dụng :

```
    <?xml   version="1.0"    encoding="utf-8"?>
  <!DOCTYPE Goc [
      <!ENTITY Ten_1 “Chuoi_1” >
      <!ENTITY Ten_2 “Chuoi_2” >
    ]>
    <Goc>
          <A X=”&Ten_1;”>
                <B>
                      &Ten_2;
                </B>
          </A>

          <C Y=”&Ten_1;”> &Ten_2; </C>

          <D Y> &Ten_2; </C>

        </Goc>
```

Tài liệu XML trên khai báo và sử dụng 2 thực thể
Thực thể thứ 1 : Tên là Ten_1 và được sử dụng trong thuộc tính của 2 thẻ A,C
Thực thể thứ 2 : Tên là ten_2 và được sử dụng trong nội dung của 3 thẻ B,C,D

Việc sử dụng thực thể trong trường hợp này giúp
- Tài liệu dễ đọc hơn : Dùng gián tiếp các tên Ten_1, ten_2 với ngữ nghĩa cụ thể nào đó thay vì trực tiếp các Chuoi_1, Chuoi_2
- Tài liệu dễ bảo trì hơn khi cần thay đổi nội dung của Chuoi_1, Chuoi_2 (chỉ cần thay đổi trong khai báo)

*** Dạng 2**

Tham chiếu đến các ký tự đặc biệt được định nghĩa trước

Ý nghĩa :
Cho phép sử dụng một số ký tự đặc biệt

Sử dụng ký tự đặc biệt được định nghĩa trước

< Ký tự <
> Ký tự >
" Ký tự nháy kép “
' Ký tự nháy đơn ‘
& Ký tự &

Sử dụng các ký tự thông qua mã số trong cách mã hóa

Nếu dùng hệ thập phân
&#So_thap_phan;
Ký tự có mã số là số thập phân
Ví dụ :
0 Ký số 0
a Ký tự a
Nếu dùng hệ thập lục phân
&#xSo_thap_luc_phan;
Ký tự có mã số là số thập lục phân
Ví dụ :
0 Ký số 0
A Ký tự a

*** Dạng 3**

Tham chiếu đến một tập họp các giá trị bên ngoài tài liệu XML
Ý nghĩa :
Cho phép tham chiếu đến tập tin chứa giá trị cần sử dụng nào đó. Các giá trị này không nhất thiết theo định chuẩn XML.
Cách sử dụng này của thực thể thông thường để bổ sung vào nội dung các hình ảnh, âm thanh, v.v.v.

Dạng khai báo thông dụng :
<!ENTITY Ten_thuc_the SYSEM Ten_tap_tin >

Thực thể Ten_thuc_the tham chiếu đến tập tin có vị trí được cho bởi Ten_tap_tin
Ghi chú :
- Ten_tap_tin bao hàm cả đường dẫn
- Có thể dùng địa chỉ URL như Ten_tap_tin

Ví dụ :
Giả sử đã có tập tin Hinh.jpg lưu trữ hình ảnh một nhân viên trong thư mục hiện hành

```
<!DOCTYPE NHAN_VIEN [
<!ENTITY  Hinh_nhan_vien  SYSTEM  “Hinh.jpg” >
]>
<NHAN_VIEN Hinh=”&Hinh_nhan_vien;” ….>
    ….

</NHAN_VIEN>
```

*** Dạng 4**

Tham chiếu đến một tài liệu XML khác
Ý nghĩa :
Cho phép phân rả tài liệu XML thành các tài liệu con được lưu trữ trong các tập tin độc lập.

Dạng khai báo thông dụng : tương tự dạng trên
<!ENTITY Ten_thuc_the SYSEM Ten_tap_tin >

Ví dụ :

Giả sử đã có các tập tin Thu_tien_1.xml , Thu_tien_2.xml , …. Thu_tien_12.xml lưu trữ thông tin về các phiếu thu tiền trong các tháng 1,2,..12 của năm đang xét.
Tập tin Thu_tien.xml lưu trữ thông tin về các phiếu thu trong năm đang xét như sau

```
<!DOC_TYPE THU_TIEN [
<!ENTITY Thu_tien_1 SYSTEM “Thu_tien_1.xml” >
<!ENTITY Thu_tien_2 SYSTEM “Thu_tien_2.xml” >

…
<!ENTITY Thu_tien_12 SYSTEM “Thu_tien_12.xml” >
]>
<THU_TIEN>
    &Thu_tien_1;
    &Thu_tien_2;
    …
    &Thu_tien_12;
</THU_TIEN>
```

2. Sử dụng tên

Kỹ thuật sử dụng tên thẻ

Tên thẻ , tên các thuộc tính trong tài liệu XML thuộc về 1 trong 2 loại sau
Loại 1 : Tên không tiền tố
Loại 2 : Tên có tiền tố

Tên không tiền tố
Mô tả đầy đủ các qui tắc đặt tên cho các tên thẻ, thuộc tính là công việc không đơn giản và đặc biệt là cũng không cần thiết.
Một cáchtóm tắt (và tất nhiên chưa đầy đủ !) tên

là chuỗi bao gồm các ký tự chữ (a-z, A-Z), ký số (0-9) và một số ký tự khác như ‘–‘ , “_” , “.”.

Tên có tiền tố

Tên có tiền tố có dạng 2 chuỗi ký tự cách nhau bới ký tự ‘:’

Chuoi_tien_to : Chuoi_ten

Ví dụ :

<A:MAT_HANG …./>
<B:MAT_HANG …/>

Thẻ A:MAT_HANG tương ứng thông tin về mặt hàng trong công ty A. Thẻ B:MAT_HANG tương ứng thông tin về mặt hàng trong công ty B. 2 thẻ này có thể có các thuộc tính khác nhau.

Sử dụng tên có tiền tố :

Nếu chỉ sử dụng tài liệu XML đơn lẻ, riêng cho ứng dụng cục bộ thì không cần thiết dùng tiền tố trong tên. Tuy nhiên nếu cần thiết tiếp nhận, kết xuất toàn bộ/một phần tài liệu XML từ/đến một ứng dụng khác (rất thông dụng trong thương mại điện tử) việc sử dụng tên với tiền tố là rất cần thiết.

Tiền tố của tên sẽ đùng để phân biệt được nguồn gốc của một thẻ trong tài liệu XML được tạo thành từ nhiều tài liệu XML khác có các thẻ trùng phân tên không tiền tố

Thuộc tính xmlns

Xét tài liệu XML với việc sử dụng các tiền tố A tương ứng tên công ty A trong giao dịch thương mại điện tử

```
<A:The_goc>
    <A:The_1 …. / >
    <A:The_2>
        Nội dung
    </A:The_2>
    …
    <A:The_2>
        Nội dung
    </A:The_2>
    …
    <A:The_3 …./>

</A:The_goc>
```

Việc sử dụng thuộc tính xmlns cho phép đơn giản hóa tài liệu XML trên

```
<The_goc xmlns=”A” >
    <The_1 …. / >
    <The_2>
        Nội dung
    <The_2>
    …
    <The_2>
        Nội dung
    <The_2>
```

<The_3 …./>

<The_goc>

Cú pháp khai báo của thuộc tính xmlns như sau
Dạng 1 :

<Ten_the xmlns=”Chuoi_tien_to” ….>
…
</Ten_the>

Khai báo trên có ý nghĩa rằng tất cả các tên bên trong thẻ đang xét (bao hàm chính thẻ này) nếu không có tiền tố thì tiền tố chính là Chuoi_tien_to

Dạng 2 :

<Ten_the xmlns:Chuoi_tien_to_1=”Chuoi_tien_to_2” ….>
…
</Ten_the>

Khai trên có ý nghĩa rằng tất cả các tên bên trong thẻ đang xét (bao hàm chính thẻ này) nếu có tiền tố là chuoi_tien_to_1 thì tiền tố thực sự là Chuoi_tien_to_2

Khai báo trên thông thướng được dùng khi trong tài liệu XML có sử dụng đồng thời nhiều tiền tố khác nhau và khai báo dạng 1 chỉ có thể áp dụng được với một trong số các tiền tố đó và các tiền tố còn lại thì lại quá dài (nhưng tại sao lại đặt dài ?)

Không gian tên (namespace)

Bản chất của không gian tên chính là Chuoi_tien_to được sử dụng trong một số các tài liệu XML

Không gian tên là chuoi_tien_to với đặc điểm quan trọng như sau : Chuoi_tien_to phải là duy nhất trên phạm vi toàn cầu

Đặc điểm trên nhằm bảo đảm rằng khi một đơn vị/ứng dụng sử dụng các tài liệu XML của mình với Chuoi_tien_to thì không có tài liệu XML nào của các đơn vị/ứng dụng khác trên phạm vi toàn cấu sử dụng Chuoi_tien_to đó.

Với đặc điểm trên thông thường không gian tên được chọn là chuỗi tương ứng với một địa chỉ URL của một tên miền trong định vị của thế giới Internet

Cách dùng trên lý giải vì sao lại phải dùng các Chuoi_tien_to rất dài trong tài liệu XML.

III. Đặc tả cấu trúc với DTD

Đặc tả cấu trúc tài liệu XML với DTD

Có nhiều dạng khác nhau cho phép khai báo (đặc tả) cấu trúc của tài liệu XML

Dạng 1 : Khai báo cấu trúc tài liệu XML được lưu trữ ngay bên trong chính tài liệu XML đó

<!DOCTYPE Ten_the_goc [
Đặc tả cấu trúc nội dung các thẻ
Đặc tả thuộc tính các thẻ
]>

Dạng 2 : Khai báo cấu trúc tài liệu XML được lưu trữ bên ngoài dưới dạng một tập tin chứa Đặc tả cấu trúc nội dungcác thẻ , Đặc tả thuộc tính các thẻ

<!DOCTYPE Ten_the_goc SYSTEM Ten_tap_tin >

Ví dụ :

<!DOCTYPE DUONG_TRON SYSTEM "DUONG_TRON.dtd" >

Dạng 3 : Khai báo cấu trúc tài liệu XML đã được chuẩn hóa , có phạm vi sử dụng rộng rải. dạng này thường đuợc dùng với các ngôn ngữ XML chung có phạm vi áp dụng toàn cầu như MathML, VML, XHTML, v.v…

<!DOCTYPE Ten_the_goc PUBLIC Chuoi_nhan_dang >

1. Đặc tả cấu trúc nội dung các thẻ

Cú pháp chung đặc tả cấu trúc nội dung của một thẻ như sau
<!ELEMENT Ten_the Bieu_thuc_dac_ta__cau_truc_noi_dung >

Bieu_thuc_cau_truc_dac_ta_noi_dung có thể chỉ là một từ khoá

Bieu_thuc_cau_truc_dac_ta_noi_dung cũng có thể bao gồm nhiều từ khóa khác mô tả cách bố trí, sắp xép các thành phần con bên trong thẻ
Với A, B là 2 thẻ con của thẻ X

- A, B A, B sắp xếp theo thứ tự tuần tự A đến B
- A* A có thể lặp lại ít nhất 0 lần
- B+ B có thể lặp lại ít nhất 1 lần
- A? A có thể có hay không có
- A|B Có thể chọn sử dụng A hay B

*** Dạng 1**

Từ khóa ANY : Thẻ có nội dung bất ký theo định chuẩn XML
Ví dụ :

<!ELEMENT X ANY >

X có thể chứa nội dung bất kỳ. Thông thường cách khai báo này chỉ để mô tả sự tồn tại của X bên trong một thẻ khác

Từ khóa EMPTY : Thẻ không có nội dung
Ví dụ :

<!ELEMENT PHAN_SO EMPTY >

PHAN_SO không thể có nội dung mà chỉ có thể có các thuộc tính

Từ khóa #PCDATA : Thẻ với nội dung là chuỗi văn bản
Ví dụ :

<!ELEMENT Ho_ten (#PCDATA) >

Ho_ten có nội dung là chuỗi và không thể chứa các thẻ khác. Đây là một trong các giới hạn chính của DTD vì không cho phép mô tả chi tiết về "kiểu" hay "loại" của chuỗi văn bản.

Với DTD muốn mô tả chi tiết hơn có thể dùng thẻ ghi chú
Ví dụ :

```
<!ELEMENT He_so (#PCDATA) >
    <!-- He_so : A_Float -->
```

*** Dạng 2**

Bieu_thuc_dac_ta_cau_truc_noi_dung cũng có thể bao gồm nhiều từ khóa khác mô tả cách bố trí, sắp xép các thành phần con bên trong thẻ
Với A, B là 2 thẻ con của thẻ X

- A, B A, B sắp xếp theo thứ tự tuần tự A đến B
- A* A có thể lặp lại ít nhất 0 lần
- B+ B có thể lặp lại ít nhất 1 lần
- A? A có thể có hay không có
- A|B Có thể chọn sử dụng A hay B

- Tuần tự

Dạng tuần tự : Các thẻ con chỉ có thể xuất hiện 1 lần duy nhất và phải theo đúng thứ tự xuất hiện trong biểu thức

Cú pháp :

```
<!ELEMENT Ten_the (Ten_the_1, Ten_the_2, ….) >
```

Ý nghĩa :
The_1, The_2, ..., The_k phải xuất hiện một lần duy nhất theo đúng thứ tự trên
Ví dụ :

```
<!ELEMENT DON_THUC(He_so,So_mu) >
```

Thẻ DON_THUC phải bao hàm bên trong 2 thẻ con He_so,So_mu theo đúng thứ tự trên

Ghi chú :

- Các thẻ bên trong có thể có tên trùng nhau

Ví dụ :

```
<!ELEMENT TAM_GIAC (DIEM,DIEM,DIEM) >
```

Thẻ TAM_GIAC phải bao hàm bên trong đúng 3 thẻ con với tên thẻ là DIEM

- Có thể sử dụng từ khóa #PCDATA trong biểu thức tuần tự (và các loại biểu thức khác)

Ví dụ :

```
<!ELEMENT X (#PCDATA,A,#PCDATA)>
```

Thẻ X phải bao gồm 3 thành phần :
Thành phần thứ 1 là chuỗi văn bản
Thành phần thức 2 là thẻ có tên A
Thành phần thứ 3 là chuỗi văn bản

- Tùy chọn

Dạng tùy chọn : Thẻ con có thể được sử dụng hay không sử dụng
Cú pháp (dạng đơn giản) :

```
<!ELEMETNT Ten_the (Ten_the_con ?) >
```

Thẻ đang xét có thể chứa 1 hay 0 lần xuất hiện của thẻ có tên là Ten_the_con

Ví dụ :

<!ELEMETNT DON_THUC (Ten?) >

Thẻ DON_THUC có thẻ chứa hay không thẻ Ten

Ghi chú :

- Có thể kết hợp với biểu thức tuần tự

<!ELEMENT X (A,B?,C) >

Thành phần đầu tiên của thẻ X là thẻ A, kế đến có thẻ có hay không có thẻ B và thành phần cuối cùng phải là C

- Có thể cho phép tùy chọn một tập họp các thẻ

<!ELEMENT X (A,B,C)? >

X có thể bao hàm bên trong các thẻ A,B,C (theo thứ tự trên) hay cũng có thẻ không chứa bất kỳ thẻ nào

Dạng chọn : Bắt buộc chọn một thẻ con để sử dụng trong tập họp thẻ cho trước

Cú pháp (dạng đơn giản) :

<!ELEMETNT Ten_the(Ten_the_1|Ten_the_2|..|Ten_the_k) >

Thẻ đang xét bắt buộc phải chứa duy nhất một trong các thẻ có tên Ten_the_1 hay ten_the_2, hay ... Ten_the_k

Ghi chú :

- Có thể kết hợp với biểu thức tuần tự

<!ELEMENT X (A,B|C,D) >

Thành phần đầu tiên của thẻ X là thẻ A, kế đến là thẻ B hay thẻ C và thành phần cuối cùng phải là D

- Có thể cho phép chọn một tập họp các thẻ

<!ELEMENT X ((A,B) | (C,D)) >

X có thể bao hàm bên trong cặp thẻ A,B (theo thứ tự trên) hay cặp thẻ C,D (theo thứ tự trên)

- Lặp

Dạng lặp ít nhất 0 lần : Các thẻ con có thể lặp lại nhiều lần hay có thẻ không có lần nào

Cú pháp :

<!ELENEMT Ten_the (Ten_the_con*) >

Ý nghĩa :

Thẻ đang xét có thẻ bao hàm bên trong nhiều thẻ có tên là Ten_the_con hay cũng có thể là thẻ rổng (không có nội dung)

Ví dụ :

<!ELEMENT LOP (HOC_SINH*) >

Thẻ LOP có thẻ chứa nhiều thẻ HOC_SINH hay không có thẻ HOC_SINH nào

Ghi chú :

- Có thể mô tả lặp đồng thời nhiều thẻ con

<!ELEMENT X (A,B,C)* >

Các thẻ A,B,C theo thứ tự trên có thẻ lặp lại ít nhất 0 lần trong thẻ X

- Có thể kết hợp với biểu thức tuần tự

Ví dụ :

<!ELEMENT X (A,B*,C) >

Thẻ X có thành phần đầu tiên là thẻ A, kế đến có thể có nhiều hay 0 lần lặp của thẻ B và cuối cùng là thẻ C

- Có thể kết hợp với biểu thức tùy chọn

Ví dụ :

<!ELEMENT X (A,B*,C?,D) >

Thẻ X có thành phần đầu tiên là thẻ A, kế đến có thể có nhiều hay 0 lần lặp của thẻ B, kế đến có thể có hay không thẻ C và cuối cùng là thẻ D

- Có thể kết hợp với biểu thức chọn

Ví dụ :

<!ELEMENT X (A|B,C*,D) >

Thẻ X có thành phần đầu tiên là thẻ A hay thẻ B , kế đến có thể có nhiều hay 0 lần lặp của thẻ B và cuối cùng là thẻ D

Dạng lặp ít nhất 1 lần : Các thẻ con có thể lặp lại nhiều lần và ít nhất là một lần

Cú pháp :

<!ELENEMT Ten_the (Ten_the_con+) >

Ý nghĩa :

Thẻ đang xét có thẻ bao hàm bên trong ít nhất một thẻ có tên là Ten_the_con

Ví dụ :

<!ELEMENT DA_THUC (DON_THUC+) >

Thẻ DATHUC phải bao hàm bên trong ít nhất một thẻ DON_THUC

Ghi chú :

- Có thể mô tả lặp đồng thời nhiều thẻ con

<!ELEMENT CT_HOA_DON
(Mat_hang,So_luong,Don_gia) + >

Các thẻ CT_HOA_DON phải bao hàm ít nhất 3 thẻ Mat_hang,So_luong,Don_gia

- Có thể kết hợp với biểu thức tuần tự

Ví dụ :

<!ELEMENT DA_GIAC (DIEM,DIEM,DIEM+) >

Các thẻ DA_GIAC phải bao hàm ít nhất 3 thẻ DIEM

- Có thể kết hợp với biểu thức tùy chọn

Ví dụ :

<!ELEMENT BIEU_THUC (Ten?,PHAN_SO+) >

Thẻ BIEU_THUC có thẻ chứa hay không thành phần đầu là thẻ Ten và kế đến ít nhất một thẻ PHAN_SO

- Có thể kết hợp với biểu thức chọn

Ví dụ :

<!ELEMENT X (A|B,C+,D) >

Thẻ X có thành phần đầu tiên là thẻ A hay thẻ B , đến ít nhất một thẻ B và cuối cùng là thẻ D

2. Đặc tả thuộc tính của thẻ

Cú pháp khai báochung :

Cú pháp khai báo các thuộc tính của thẻ tương tự như cú pháp khai báo kiểu cấu trúc trong ngôn ngữ lập trình

<!ATTLIST Ten_the
Ten_thuoc_tinh_1 Kieu_1 Tham_so_1

```
        Ten_thuoc_tinh_2     Kieu_2 Tham_so_2
        ...
        Ten_thuoc_tinh_k     Kieu_k Tham_so_k
>
```

Ý nghĩa :

Ten_the : tên thẻ cần khai báo các thuộc tính

Ten_thuoc_tinh_1,Ten_thuoc_tinh_2, ...Ten_thuoc_tinh_k : Tên các thuộc tính của thẻ đang khai báo

Kieu_1,Kieu_2, ..., Kieu_k : Mô tả tập hợp các giá trị mà thuộc tính có thể nhận

Tham_so_1,Tham_so_2,.., Tham_so_k : Mô tả một số tính chất trên thuộc tính tương ứng

Ví dụ : Đặc tả cấu trúc tài liệu XML biểu diễn thông tin về biểu thức phân số P = 4/5 + 6/7*1/3 - 10/3 +11/2*2/3

```
<?xml   version="1.0"   encoding="utf-8"?>
<!DOCTYPE BIEU_THUC [
<!ELEMENT BIEU_THUC (PHAN_SO | TICH_SO)+ >
<ATTLIST BIEU_THUC
        Ten             CDATA        #IMPLIED
                <!-- Ten : A_String        -->
>
<!ELEMENT PHAN_SO EMPTY >
<ATTLIST PHAN_SO
        Tu_so           CDATA        #REQUIRED
        <!-- Tu_so : A_Int     -->
        Mau_so                CDATA        #REQUIRED
        <!-- Mau_so : A_Int // >0 -->
>
<!ELEMENT TICH_SO (PHAN_SO)+ >
]>
```

<u>* Kiểu</u>

Kiểu : Mô tả tập hợp các giá trị của thuộc tính

Có nhiều cách khác nhau cho phép mô tả tập hợp các giá trị có thể có của một thuộc tính. Phần sau chỉ giới thiệu 2 cách mô tả chính và thông dụng. Đề biết thêm chi tiết về các cách mô tả khác xin tham khảo các tài liệu chuyên biệt về DTD

Cách 1 : Dùng từ khoá CDATA

Cú pháp :

```
        <!ATTLIST Ten_the
         ...
        Ten_thuoc_tinh        CDATA
        ...
        >
```

Ý nghĩa :

Tập họp các giá trị của huộc tính với khai báo CDATA chính là tập họp các chuỗi. Đây là trường hợp sử dụng thông dụng nhất, và đây cũng là một trong các giới hạn của DTD vì không cho phép mô tả chi tiết hơn về kiểu của thuộc tính. Tương tự như nội dung văn bản của thẻ, để mô tả thêm thông tin cần sử dụng các ghi chú

Ví dụ : Đặc tả cấu trúc tài liệu XMl biểu diễn phương trình đường thẳng trong mặt phẳng

```
<?xml  version="1.0"   encoding="utf-8"?>
<!DOCTYPE DUONG_THANG [

<!ELEMENT DUONG_THANG EMPTY>
        <!—phương trình ax + by+c =0     -->

<ATTLIST DUONG_THANG
        Ten             CDATA          #IMPLIED
                <!-- Ten : A_String    -->
        a               CDATA          #REQUIRED
                <!-- a : A_Float     -->
        b               CDATA          #REQUIRED
                <!-- b : A_Float     -->
        c               CDATA          #REQUIRED
                <!-- c : A_Float     -->
>
        <!—a,b không đồng thời là 0      -->

]>
```

Cách 2 : Dùng biểu thức liệt kê

Cú pháp :

```
        <!ATTLIST Ten_the
         ...
        Ten_thuoc_tinh ( Gia_tri_1,Gia_tri_2,...._gia_tri_k)
        ...
        >
```

Ý nghĩa :

Tập hợp các giá trị có thể có của thuộc tính đang xét chính là tập hợp các giá trị được liệt kê Gia_tri_1,Gia_tri_2, ...,Gia_tri_k. Các giá trị này là các chuỗi ký tự

Ví dụ : Đặc tả cấu trúc tài liệu XML biểu diễn thông tin về phiếu điểm của một học sinh

```
<?xml  version="1.0"   encoding="utf-8"?>

<!DOCTYPE PHIEU_DIEM [

<!ELEMENT PHIEU_DIEM (HOC_SINH, DIEM_SO+ ) >

<!ELEMENT HOC_SINH EMPTY >
<ATTLIST HOC_SINH
        Ho_ten          CDATA                       #REQUIRED
                <!-- Ho_ten : A_String         -->
        Ngay_sinh       CDATA                       #REQUIRED
                <!--Ngay_sinh : A_Date    -->
         Xep_loai       (“Giỏi”, “Khá”, Trung bình”, Yếu”) #IMPLIED
>
```

```
<!ELEMENT DIEM_SO EMPTY >
<ATTLIST DIEM
        Ten_mon                 CDATA          #REQUIRED
                <!-- Ten_mon : A_String    -->
        Gia_tri           CDATA                 #REQUIRED
                <!-- Gia_tri : A_Float // từ 0 đến 10     -->
>
]>
```

*** Tham số**

Tham_so : Mô tả tính chất của thuộc tính
Có nhiều cách khác nhau cho phép mô tả tập hợp các giá trị có thể có của một thuộc tính. Phần sau chỉ giới thiệu 3 cách mô tả chính và thông dụng. Để biết thêm chi tiết về các cách mô tả khác xin tham khảo các tài liệu chuyên biệt về DTD

Cách 1 : Dùng từ khóa #REQUIRED
Cú pháp :

```
        <!ATTLIST Ten_the
         ...
        Ten_thuoc_tinh Kieu   #REQUIRED
        ...
        >
```

Ý nghĩa :
Thuộc tính đang xét là thuộc tính bắt buộc phải có. Đây là cách sử dụng phổ biến nhất

Ví dụ : Đặc tả cấu trúc tài liệu XML biểu diễn thông tin về các đơn thức với tên bắt buộc phải có

```
<?xml   version="1.0"   encoding="utf-8"?>

<!DOCTYPE DON_THUC [

<!ELEMENT DON_THUC (He_so, So_mu ) >
<ATTLIST DON_THUC
        Ten             CDATA          #REQUIRED
        <!-- Ten : A_String        -->
        Bien_so         CDATA          #REQUIRED
        <!-- Bien_so: A_String    -->
>
<!ELEMENT He_so   #PCDATA >
        <!-- He_so : A_Float             -->

<!ELEMENT So_mu #PCDATA >
        <!-- So_mu : A_Int // >=0       -->
]>
```

Cách 2 : Dùng từ khóa #IMPLIED
Cú pháp :

```
        <!ATTLIST Ten_the
         ...
```

```
        Ten_thuoc_tinh Kieu   #IMPLIED
        ...
        >
```

Ý nghĩa :

Thuộc tính đang xét là tùy chọn và không bắt buộc phải có

Ví dụ : Đặc tả cấu trúc tài liệu XML biểu diễn thông tin về tam thức P(x) =2x2 -4x +6

```
<?xml   version="1.0"   encoding="utf-8"?>

<!DOCTYPE TAM_THUC [

<!ELEMENT TAM_THUC (DON_THUC,DON_THUC,DON_THUC) >
<ATTLIST TAM_THUC
        Ten             CDATA         #IMPLIED
                <!-- Ten : A_String        -->
        Bien_so         CDATA         “x”
                <!-- Bien_so: A_String // định sẵn là x   -->
>
<!ELEMENT DON_THUC EMPTY >
<ATTLIST DON_THUC
        He_so           CDATA         #REQUIRED
                <!-- He_so : A_Float // Khác 0 nếu So_mu=2    -->
        So_mu           (0,1,2)           #REQUIRED
                <!-- So_mu : A_Int // =0,1,2 và khác nhau     -->
>
]>
```

Cách 3 : Dùng từ khóa #FIXED

Cú pháp :

```
        <!ATTLIST Ten_the
         ...
        Ten_thuoc_tinh Kieu     #FIXED Gia_tri
        ...
        >
```

Ý nghĩa :

Thuộc tính đang xét phải có giá trị cố định là Gia_tri. Trường hợp này ít được sử dụng

Ví dụ :

Đặc tả cấu trúc tài liệu XML biểu diễn thông tin về các đơn thức chỉ với biến số x

```
<?xm  lversion="1.0"   encoding="utf-8"?>

<!DOCTYPE DON_THUC [

<!ELEMENT DON_THUC  (He_so, So_mu ) >
<ATTLIST DON_THUC
        Ten             CDATA   #REQUIRED
```

```
        <!-- Ten : A_String        -->
        Bien_so         CDAT    #FIXED  “x”
        <!-- Bien_so: A_String    -->
>

<!ELEMENT He_so   (#PCDATA)  >
        <!-- He_so : A_Float             -->

<!ELEMENT So_mu (#PCDATA) >
        <!-- So_mu : A_Int // >=0       -->
]>
```

IV. Đặc tả cấu trúc với XML-Schema

Đặc tả cấu trúc tài liệu XML với Xml-Schema

XML Schema thuộc họ các ngôn ngữ XML nên khai báo XML Schema chính là tạo lập tài liệu XML mà nội dung chính là các thẻ đánh dấu, các thẻ này sẽ mô tả cho cấu trúc các thẻ của một tài liệu XML khác. Cấu trúc chung (thông dụng) của các tài liệu trong XML Shema như sau

```
<?xmlversion="1.0"encoding="utf-8"?>
<xs:schema     xmlns:xs="http://www.w3.org/2001/XMLSchema">
        Đặc tả các thẻ
        Đặc tả các kiểu
</xs:schema>
```

Với DTD, Đặc tả cấu trúc tài liệu XML bao gồm 2 phần : Đặc tả cấu trúc nội dung các thẻ , Đặc tả thuộc tính các thẻ. Thông tin về một thẻ được mô tả qua 2 phần tách biệt nhau : Đặc tả cấu trúc nội dung mô tả cách sắp xếp các thành phần bên trong của thẻ đang xét, Đặc tả thuộc tính mô tả hệ thống các thuộc tính của thẻ đang xét.

Với XML Schema, thông tin về một thẻ được mô tả tập trung qua một ý niệm duy nhất là kiểu. Mỗi thẻ sẽ có tương ứng một kiểu. Đặc tả kiểu mô tả kiểu của thẻ cùng với một số tính chất khác. Đặc tả kiểu mô tả các thông tin về các thẻ thuộc kiểu (có thể có nhiều thẻ cùng thuộc một kiểu) bao hàm cả các thông tin về cách sắp xếp các thành phần bên trong của thẻ và hệ thống các thuộc tính của thẻ.

Ví dụ

```
<?xml version="1.0" encoding="utf-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="DA_THUC" type="K_DA_THUC"/>

  <xs:complexType name="K_DA_THUC">
     <xs:sequence>
        <xs:element name="DON_THUC"
                    type="K_DON_THUC"
                    minOccurs="1"/>
     </xs:sequence>
     <xs:attribute name="Ten" type="xs:string" />
     <xs:attribute name="Bien_so" type="xs:string"/>
  </xs:complexType>
```

```
  <xs:complexType name="K_DON_THUC">
      <xs:attribute name="He_so" type="xs:float"/>
      <xs:attribute name="So_mu" type="SO_TU_NHIEN"/>
  </xs:complexType>

  <xs:simpleType name="SO_TU_NHIEN">
       <xs:restriction base="xs:int">
           <xs:minInclusive value="0"/>
       </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

Ý nghĩa của đặc tả :

```
   <xs:element name="DA_THUC" type="K_DA_THUC"/>
```

Tài liệu XML có thẻ gốc là DA_THUC thẻ này có kiểu là kiểu phức hợp với tên là K_DA_THUC (có thể dùng cùng tên là DA_THUC)

```
    <xs:complexType name="K_DA_THUC">
      <xs:sequence>
         <xs:element name="DON_THUC"
                     type="K_DON_THUC"
                     minOccurs="1"/>
      </xs:sequence>
      <xs:attribute name="Ten" type="xs:string" />
      <xs:attribute name="Bien_so" type="xs:string"/>
   </xs:complexType>
```

Kiểu phức hợp K_DA_THUC bao gồm bên trong

- Thẻ DON_THUC có kiểu là kiểu phức hợp với tên là K_DON_THUC và thẻ DON_THUC phải xuất hiện ít nhất 1 lần trong các thẻ có kiểu là K_DA_THUC
- 2 thuộc tính :
  Ten với kiểu là kiểu cơ sở dạng chuỗi
  Bien_so với kiểu là kiểu cơ sở dạng chuỗi

=== > Tóm tắt : Thẻ DA_THUC phải bao hàm bên trong ít nhất một thẻ DON_THUC và có 2 thuộc tính Ten,Bien_so

```
<xs:complexType name="K_DON_THUC">
      <xs:attribute name="He_so" type="xs:float"/>
      <xs:attribute name="So_mu" type="SO_TU_NHIEN” /> </xs:complexType>
```

Kiểu phức hợp K_DON_THUC chỉ bao gồm bên trong cá thuộc tính
  He_so có kiểu là kiểu cơ sỡ loại số thực
  So_mu có kiểu là kiểu đơn giản với tên SO_TU_NHIEN

== > Tóm tắt : Thẻ DON_THUC là thẻ không có nội dung và có 2 thuộc tính “ He_so,So_mu

```
<xs:simpleType name="SO_TU_NHIEN">
      <xs:restriction base="xs:int">
          <xs:minInclusive value="0"/>
      </xs:restriction>
</xs:simpleType>
```

Kiểu đơn giản SO_TU_NHIEN chính là kiểu cơ sở số nguyên với hạn chế : giá trị phải lớn hơn hay bằng 0
=== > Thuộc tính So_mu của thẻ DON_THUC phải là một số nguyên không âm

1. Đặc tả kiểu

Đặc tả kiểu
XML Schema có 3 loại kiểu chính :
- Loại 1 : Kiểu định nghĩa sẵn (BultinType)
- Loại 2 : Kiểu đơn giản (simpleType)
- Loại 3 : Kiểu phức hợp (complexType).

Tùy thuộc vào loại thẻ cần mô tả (theo cách phân loại sẽ trình bày sau) loại kiểu tương ứng sẽ được sử dụng.

* Kiểu định nghĩa sẵn

Kiểu định nghĩa sẵn (thư viện)
Khái niệm :
Là các kiểu được xây dựng, định nghĩa sẵn trong XML Schema. Các kiểu này tương tự như các kiểu cơ sở trong ngôn ngữ lập trình
Có tên trong danh sách các kiểu cơ sở của XML Schema

Danh sách các kiểu cơ sở :

Một số kiểu cơ sở thông dụng

Ten_kieu_co_so	Ý nghĩa
string	Chuỗi ký tự
int, integer	Số nguyên
float	Số thực chính xác đơn
double	Số thực chính xác kép
boolean	Giá trị logic
date	ngày
month	Tháng
ID	Chuỗi định danh
binary	Dữ liệu nhị phân

Ý nghĩa sử dụng :

Được sử dụng để mô tả trực tiếp kiểu của các thuộc tính hay của thẻ thỏa 2 điều kiện :
Điều kiện 1 : Không có thuộc tính
Điều kiện 2 : Không chứa thẻ khác (nội dung là chuỗi văn bản) và có miền giá trị (tập hợp giá trị có thể có) thích hợp với kiểu

Với các thẻ có thuộc tính hay có chứa thẻ khác, bắt buộc phải dùng kiểu phức hợp vì kiểu cơ sở và kiểu đơn giản không cho phép mô tả thông tin về thuộc tính , thẻ con bên trong

Cú pháp :

Khi dùng với thẻ
<xs:element name=”Ten_the” type=”Ten_kieu_co_so” ... />

Khi dùng với thuộc tính
<xs:attribute name=”Ten_thuoc_tinh” type=”Ten_kieu_co_so” .. />

Ví dụ :

<xs:element name="Ho_ten" type="xs:string" />
Thẻ Ho_ten không có thuộc tính, không chứa thẻ con và có nội dung là chuỗi văn bản

<xs:element name="Ngay_sinh" type="xs:date" />
Thẻ Ngay_sinh không có thuộc tính, không chứa thẻ con và có nội dung tương ứng một ngày

<xs:attribute name="He_so" type="xs:float"/>
Thuộc tính He_so phải là số thực

<xs:attribute name="x" type="xs:int"/>
Thuộc tính x phải là số nguyên

<xs:attribute name="f" type="xs:boolean"/>
Thuộc tính f phải là giá trị logic

*** Kiểu đơn giãn**

Kiểu đơn giản (simpleType)
Khái niệm :
Là các kiểu do người dùng định nghĩa dựa trên các kiểu cơ sở có sẵn trong XML Schema.

Ý nghĩa sử dụng :
Được sử dụng để mô tả trực tiếp kiểu của các thuộc tính hay các thẻ thỏa 2 điều kiện :
Điều kiện 1 : Không có thuộc tính
Diều kiện 2 : Không chứa thẻ khác (nội dung là chuỗi văn bản) và có miền giá trị (tập họp giá trị có thể có) là tập con của miền giá trị một kiểu cơ sở nào đó

Tương tự như với kiểu cơ sở, các thẻ có thuộc tính hay thẻ có chứa thẻ con khác, nhất thiết phải dùng kiểu phức hợp vì kiểu cơ sở và kiểu đơn giản không cho phép mô tả thêm thông tin về thuộc tính ,thẻ con bên trong

Cú pháp : (dạng đơn giản và thông dụng)
<xs:simpleType name="Ten_kieu">
<xs:restriction base="Ten_kieu_co_so">
Giới hạn (ràng buộc) trên miền giá trị
</xs:restriction>
</xs:simpleType>

Ten_kieu : Tên của kiểu đơn giản
Ten_kieu_co_so : Tên của kiểu cơ sở tương ứng

Giới hạn (ràng buộc) trên miền giá trị : Có nhiều dạng giới hạn (ràng buộc) khác nhau cho phép mô tả chi tiết miền giá trị của kiểu cơ sở (đây chính là một trong các thế mạnh của XML Shema so với DTD).

Giáo trình chỉ giới hạn xem xét và trình bày tóm tắt 2 loại ràng buộc chính và thông dụng : Ràng buộc về cận trên các kiểu cơ sở loại số (số nguyên, số thực) , ràng buộc loại liệt kê trên kiểu cơ sở. Để biết thêm chi tiết về các các ràng buộc khác xin tham khảo các tài liệu khác chuyên biệt về XML Shema.
Giới hạn (ràng buộc) về cận trên kiểu cơ sở loại số :

Có 4 thẻ chính được sử dụng để cho phép xác định các cận (cận trên, cận dưới) của kiểu cơ sở đang xét .

Dạng khai báo chung các ràng buộc loại này như sau

```
<xs:simpleType  name="Ten_kieu">
        <xs:restriction  base="Ten_kieu_co_so_loai_so">
            Khai báo cận dưới
            Khai báo cận trên
        </xs:restriction>
</xs:simpleType>
```

Khai báo cận dưới : Sử dụng từ khoáminInclusive (cận dưới cho phép sử dụng biên), minExclusive (cận dưới không cho phép sử dụng biên)

Cú pháp

```
<xs:minInclusive  value="Gia_tri_bien_duoi"/>
```

Kiểu đơn giản đang xét có miền giá trị là tập họp các số x thỏa điều kiện x thuộc miền giá trị của kiểu cơ sở

x >= Gia_tri_bien_duoi

```
<xs:minExclusive  value="Gia_tri_bien_duoi"/>
```

Kiểu đơn giản đang xét có miền giá trị là tập họp các số x thỏa điều kiện x thuộc miền giá trị của kiểu cơ sở

x > Gia_tri_bien_duoi

Ví dụ :

```
<xs:simpleType name="SO_THUC_DUONG">
    <xs:restriction base="xs:float">
        <xs:minInclusive value="0" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="SO_NGUYEN_DUONG">
    <xs:restriction base="xs:int">
        <xs:minInclusive value="0" />
    </xs:restriction>
</xs:simpleType>
```

Khai báo cận trên : Sử dụng từ khoámaxInclusive (cận trên cho phép sử dụng biên), maxExclusive (cận trên không cho phép sử dụng biên)

Cú pháp

<xs:maxInclusive value="Gia_tri_bien_tren"/>
Kiểu đơn giản đang xét có miền giá trị là tập hợp các số x thỏa điều kiện x thuộc miền giá trị của kiểu cơ sở
x < Gia_tri_bien_tren

<xs:maxExclusive value="Gia_tri_bien_tren"/>
Kiểu đơn giản đang xét có miền giá trị là tập hợp các số x thỏa điều kiện x thuộc miền giá trị của kiểu cơ sở
x < Gia_tri_bien_tren

Ví dụ :

```
<xs:simpleType name="KY_SO">
        <xs:restriction base="xs:int">
            <xs:minInclusive value="0" />
            <xs:maxInclusive value="9" />
        </xs:restriction>
</xs:simpleType>

<xs:simpleType name="DIEM_SO">
        <xs:restriction base="xs:float">
            <xs:minInclusive value="0" />
            <xs:maxInclusive value="10" />
        </xs:restriction>
</xs:simpleType>
```

Giới hạn (ràng buộc) loại liệt kê trên kiểu cơ sở :

Cho phép xác định miền giá trị của kiểu đơn giản đang xét bằng cách liệt kê các giá trị của tập hợp này (tương tự như biểu thức liệt kế của DTD nhưng cho phép sử dụng với thuộc tính, thẻ thay vì chỉ dùng với thuộc tính)

Dạng khai báo các ràng buộc loại này như sau

```
<xs:simpleType name="Ten_kieu">
                <xs:restriction  base="Ten_kieu_co_so_loai_so">
                        <xs:enumeration value=”Gia_tri_1” />
                        <xs:enumeration value=”Gia_tri_2” />
                        ...
                        <xs:enumeration value=”Gia_tri_k” />
                </xs:restriction>
</xs:simpleType>
```

Ví dụ :

```
<xs:simpleType name="LOAI_KIEM_TRA">
     <xs:restriction >
          <xs:enumeration value="Kiểm tra 15 phút " />
          <xs:enumeration value="Kiểm tra 1 tiết " />
          <xs:enumeration value="Kiểm tra học kỳ " />
     </xs:restriction>
 </xs:simpleType>
```

```
<xs:simpleType name="LOAI_HOC_LUC" >
    <xs:restriction base="xs:string">
        <xs:enumeration value="Giỏi" />
        <xs:enumeration value="Khá" />
        <xs:enumeration value="Trung bình" />
        <xs:enumeration value="Yếu" />
    </xs:restriction>
</xs:simpleType>
```

*** Kiểu phức hợp**

Kiểu phức hợp (complexType)
Khái niệm :

Là các kiểu do người dùng tự định nghĩa cho phép mô tả nội dung và các thuộc tính của các thẻ được khai báo thuộc về kiểu đang xét

Ý nghĩa sử dụng :

Được sử dụng để mô tả kiểu của cáccác thẻ thỏa một trong 2 điều kiện :
Điều kiện 1 : Có thuộc tính
Điều kiện 2 : Có chứa thẻ khác

Các thẻ có thuộc tính không thể khai báo với kiểu cơ sở hay kiểu đơn giản vì các kiểu này không cho phép mô tả thông tin về thuộc tính

Các thẻ có chứa thẻ khác cũng không thể khai báo với kiểu cơ sở hay kiểu đơn giản vì các kiểu này không cho phép mô tả thông tin về các thành phần bên trong

Dạng khai báo chung các kiểu phức hợp như sau

```
<xs:complexType name="Ten_kieu">
    Dac_ta_cau_truc_noi_dung
    Dac_ta_thuoc_tinh
</xs:complexType>
```

Dac_ta_cau_truc_noi_dung :

Mô tả cách thức tổ chức, sắp xếp các thẻ con bên trong thẻ có kiểu là kiểu phức hợp đang xét.

Tương tự như DTD, XML Shema cũng cho phép nhiều dạng tổ chức sắp xếp (tuần tự, chọn, lặp) các thẻ con với các cú pháp riêng. Một trong các đặc tính mới của XML Shema là cho phép khai báo chi tiết hơn về số lần lặp của một thành phần

Dac_ta_thuoc_tinh :

Mô tả hệ thống các thuộc tính của thẻ có kiểu là kiểu phức hợp đang xét.

Việc mô tả các thuộc tính trong XML Shema cũng tương tự như mô tả thuộc tính trong DTD nhưng với mở rộng rất quan trọng : Cho phép định nghĩa và sử dụng các kiểu đơn giản để mô tả chi tiết về miền giá trị của một thuộc tính

- Đặc tả cấu trúc nội dung

Dac_ta_cau_truc_noi_dung :

XML Schema cho phép mô tả cách thức tổ chức, sắp xếp các thành phần bên trong thẻ qua 3 dạng cơ sở

Dạng tuần tự (tương tự như DTD):

Mô tả thứ tự xuất hiện tuần tự các thành phần

Dạng tùy chọn (hoàn toàn tương tự như DTD):

: Mô tả việc phải sử dụng một thành phần nào đó trong tập hợp các thành phần cho trước

Dạng lặp (bao hàm các dạng tùy chọn, chọn , lặp ít nhất 0 lần, lặp ít nhất 1 lần trong DTD) :

Mô tả việc cho phép lặp lại của các thành phần với các bản số

+ Tuần tự

Dạng tuần tự : Sử dụng thẻ/từ khóa sequence
Cú pháp :

```
<xs:complexType name="Ten_kieu">
      <xs:sequence>
         Thanh_phan_1
         Thanh_phan_2
         ....
         Thanh_phan_k
      </xs:sequence>
          ....
  </xs:complexType>
```

Ý nghĩa :

Các thành phần Thanh_phan_1, Thanh_phan_2, ... Thanh_phan_k phải xuất hiện duy nhất và đúng theo thứ tự trên trong thẻ tương ứng

Ví dụ :

```
  <xs:complexType name="DIEM">
     <xs:sequence>
        <xs:element name="x” type="xs:float" />
        <xs:element name="y” type="xs:float" />
     </xs:sequence>
  </xs:complexType>
```

+ Tùy chọn

Dạng tùy chọn : Sử dụng thẻ/từ khóa choice
Cú pháp :

```
<xs:complexType name="Ten_kieu">
      <xs:choice>
         Thanh_phan_1
         Thanh_phan_2
         ....
         Thanh_phan_k
      </xs:choice>
          ....
  </xs:complexType>
```

Ý nghĩa :

Thẻ có kiểu Ten_kieu phải sử dụng một thành phần trong số các thành phần Thanh_phan_1, Thanh_phan_2, ... Thanh_phan_k

Dạng này chỉ sử dụng trong một số trường hợp đặc thù và không thông dụng (vì sao ???)

Ví dụ :

```
<xs:complexType name="X">
   <xs:choice>
      <xs:element name="A” type="A" />
      <xs:element name="B” type="xs:string" />
   </xs:choice>
</xs:complexType>
```

Các thẻ có khai báo kiểu X phải bao hàm bên trong một trong 2 thẻ con sau

Thẻ có tên A và có kiểu A (cho phép tên kiểu và tên thẻ trùng nhau)

Thẻ có tên B và có kiểu là chuỗi

+ Lặp

Dạng lặp : Sử dụng thuộc tính/từ khóa minOccurs , maxOcuurs

Cú pháp (thông dụng)

```
<xs:complexType name="Ten_kieu">
   <xs:sequence>
      ...
      <xs:element name="Ten_the_con”
                  type="Kieu_the_con"
                  minOccurs=”So_lan_lap_toi_thieu”
                  manOccurs=”So_lan_lap_toi_da”    />
      ...

   </xs:sequence>
     ....
</xs:complexType>
```

Ý nghĩa :

Thẻ có kiểu Ten_kieu có chứa bên trong thẻ con có tên Ten_the_con với số lần lặp tối thiểu là So_lan_lap_toi_thieu và số lần lặp tối đa là So_lan_lap_toi_da.

Một số trường hợp thông dụng

Tùy chọn (có thẻ có hay không)

minOccurs=”0”

maxOccurs=”1”

Lặp lại ít nhất 0 lần (nhiều hoặc không có lần nào)

minOccurs=”0”

Lặp lại ít nhất 1 lần

minOccurs=”1”

Lặp lại ít nhất 1 lần và nhiều nhất 5 lần

minOccurs=”1”

maxOccurs=”5”

Lặp lại đúng 3 lần

minOccurs=”3”

maxOccurs=”3”

Ví dụ :

```
  <xs:complexType name="DA_THUC">
     <xs:sequence>
        <xs:element name="DON_THUC” type="DON_THUC"
                    minOccurs=”1” />
     </xs:sequence>
     <-- Mô tả các thuộc tính -->
      ...
  </xs:complexType>

 <xs:complexType name="DA_GIAC">
     <xs:sequence>
        <xs:element name="DIEM” type="DIEM"
                    minOccurs=”3”
                    maxOccurs=”3” />
     </xs:sequence>
     <-- Mô tả các thuộc tính -->
      ...
  </xs:complexType>

  <xs:complexType name="KHOI">
     <xs:sequence>
        <xs:element name="LOP” type="LOP"
                    minOccurs=”0”
                    maxOccurs=”12” />
     </xs:sequence>
     <-- Mô tả các thuộc tính -->
      ...
  </xs:complexType>

  <xs:complexType name="HOA_DON">
     <xs:sequence>
        <xs:element name="CT_HOA_DON” type="CT_HOA_DON"
                    minOccurs=”1”
                    maxOccurs=”10” />
     </xs:sequence>
     <-- Mô tả các thuộc tính -->
      ...
  </xs:complexType>
```

- Đặc tả thuộc tính

Dac_ta_thuoc_tinh

Cho phép mô tả hệ thống các thuộc tính của một thẻ

Cú pháp :

```
<xs:complexType  name="Ten_kieu">
Đặc tả cấu trúc nội dung
....
<xs:attribute name="Ten_thuoc_tinh" type="Kieu_thuoc_tinh"
            Tinh_chat_thuoc_tinh />
....
</xs:complexType>
```

Ten_thuoc_tinh : tên của thuộc tính của kiểu đang xét, không cho phép 2 thuộc tính có cùng tên

Kieu_thuoc_tinh : Tên của kiểu cơ sở hay kiểu đơn giản

Tinh_chat_thuoc_tinh : Mô tả một số tính chất của thuộc tính. XML Shema cho phép mô tả rất nhiều loại tính chất khác nhau, mỗi tính chất tương ứng với một từ khóa riêng

```
<xs:attribute name="Ten_thuoc_tinh" type="Kieu_thuoc_tinh"
                Tu_khoa_1=”Gia_tri_1”
                Tu_khoa_2=”Gia_tri_2”
                  ..
                Tu_khoa_k=”Gia_tri_k” />
```

Một số tính chất thông dụng như sau

Giá trị định sẵn : từ khóa default

Giá trị cố định : từ khóa fixed

Tùy chọn (có hay không có sử dụng : từ khóa use

Ví dụ :

```
<xs:attribute name="Tu_so" type="SO_NGUYEN_DUONG"
                default=”1” />

<xs:attribute name="Bien_so" type="xs:string"
                fixed=”x” />

<xs:attribute name="Ten_don_thuc" type="xs:string"
                use=”optional” />
```

2. Đặc tả thẻ

Với DTD, đặc tả cấu trúc tài liệu XML tập trung vào việc đặc tả các thẻ với rất nhiều dạng bố trí , sắp xếp các thành phần trong thẻ.

Với XML Schema, đặc tả cấu trúc tài liệu XML tập trung vào việc đặc tả các kiểu, đặc tả các thẻ trong XML Schema rất đơn giản và chỉ nhằm vào mục tiêu chính là xác định kiểu sẽ được sử dụng của thẻ.

Các thông tin cần mô tả khi đặc tả một thẻ trong XML bao gồm

- Tên thẻ
- Kiểu của thẻ
- Một số tính chất khác của thẻ

Dạng khai báo chung như sau

<xs:element name=”Ten_the” type=”Ten_kieu”
Thuoc_tinh_khac />

Ten_the :

Tên của thẻ đang xét và tuân theo cách đặt tên của định chuẩn XML

Ten_kieu :

Tên của kiểu tương ứng mô tả thông tin về thẻ. Thông thường tên kiểu và tên thẻ sẽ được đặt trùng nhau

Thuoc_tinh_khac :

Có nhiều loại thuộc tính khác nhau cho phép mô tả các tính chất của thẻ mà trong đó thông dụng nhất là 2 thuộc tính minOccurs, maxOccurs (đã trình bày).

Khi đặc tả các thẻ vấn đề quan trong nhất là xác định loại kiểu sẽ dùng trong thẻ. Tùy thuộc vào loại thẻ (theo cách phân loại của phần sau) loại kiểu tương ứng sẽ được dùng

*** Phân loại thẻ**

Hệ thống phân loại thẻ :

Có rất nhiều cách phân loại các thẻ, mỗi cách phục vụ cho một mục tiêu khác nhau. Với mục tiêu phân loại là nhằm xác định loại kiểu tương ứng được dùng, hệ thống các thẻ trong tài liệu XML có thể được phân loại như sau.

Thẻ bao gồm 2 nhóm chính

- Nhóm 1 : Nhóm các thẻ có thuộc tính
- Nhóm 2 : Nhóm các thẻ không có thuộc tính

Với các thẻ có thuộc tính, nhất thiết phải sử dụng kiểu phức hợp.
== > Khai báo kiểu phức hợp Y (có thể dùng tên thẻ đang xét)
== > Sử dụng Y là kiểu của thẻ đang xét

Với các thẻ không có thuộc tính việc sử dụng loại kiểu nào phụ thuộc vào việc phân loại chi tiết hơn các thẻ thuộc nhóm này

Các thẻ không có thuộc tính bao gồm 2 nhóm

- Nhóm 2.1 : Nhóm các thẻ không có thuộc tính và có chứa các thẻ con bên trong
- Nhóm 2.2 : Nhóm các thẻ không có thuộc tính và không chứa các thẻ con bên trong (nội dung là chuỗi văn bản)

Tương tự như nhóm 1, các thẻ thuộc nhóm 2.1 nhất thiết phải sử dụng kiểu phức hợp.
== > Khai báo kiểu phức hợp Y (có thể dùng tên thẻ đang xét)
== > Sử dụng Y là kiểu của thẻ đang xét

Các thẻ thuộc nhóm 2.2 có thể chọn sử dụng kiểu cơ sở hay kiểu đơn giản phụ thuộc vào miền giá trị MGT của chuỗi văn bản bên trong thẻ

Nếu miền giá trị MGT này tương ứng với miền giá trị của một kiểu cơ sở X nào đó
== > kiểu cơ sở X sẽ được dùng

Nếu miền giá trị MGT này chỉ tương ứng với một tập con của miền giá trị một kiểu cơ sở X nào đó
== > Khai báo kiểu đơn giản Y dựa trên kiểu cơ sở X
== > Sử dụng Y là kiểu của thẻ đang xét

*** Một thuật giải đặc tả thẻ**

Đặc tả thẻ gốc X với thông tin về kiểu tương ứng (giả sử là A)

Xét loại kiểu của A
A là kiểu phức hợp :
Đặc tả kiểu phức hợp A bao gồm
Đặc tả hệ thống các thẻ con của thẻ gốc X
Đặc tả thẻ X1 với thông tin về kiểu (giả sử là A1)
Đặc tả thẻ X2 với thông tin về kiểu (giả sử là A2)
...
Đặc tả thẻ XK với thông tin về kiểu (giả sử là Ak)
Đặc tả hệ thống các thuộc tính của thẻ gốc X
Đặc tả thuộc tính T1 với thông tin về kiểu (giả sử là B1)
Đặc tả thuộc tính T2 với thông tin về kiểu (giả sử là B2) ...
Đặc tả thuộc tính Tk với thông tin về kiểu (giả sử là Bk)
A là kiểu đơn giản :
Đặc tả kiểu đơn giản A bao gồm
Đặc tả kiểu cơ sở của A
Đặc tả các hạn chế trên kiểu cơ sở của A
A là kiểu cơ sở :
Không cần đặc tả thêm

Xét loại kiểu của A1
Xét loại kiểu của A2
...
Xét loại kiểu của Ak

Xét loại kiểu của B1
Xét loại kiểu của B2
...
Xét loại kiểu của Bk
Xét loại kiểu của T1
Xét loại kiểu của T2
...
Xét loại kiểu của Tk
.....
Xét loại kiểu của các kiểu phát sinh thêm khi đặc tả các kiểu phía trên
.....

V. Bài tập

1. Đặc tả

Yêu cầu chung
Đặc tả nội dung & cấu cấu trúc (với DTD hay Xml-schema) của tài liệu XML tương ứng các đối tượng trong thực tế

Hướng dẫn chung :
- Sử dụng thẻ gốc biểu diễn thông tin của đối tượng trong thực tế đang xét
- Sử dụng các thẻ con của thẻ gốc biểu diễn các "đối tượng con" của đối tượng thực tế đang xét (và tiếp tục nếu "đối tượng con" đang xét lại bao gồm bên trong các "đối tượng con" khác)

* Dãy số nguyên

Yêu cầu
Đặc tả nội dung & cấu cấu trúc (với DTD hay Xml-Schema) của tài liệu XML tương ứng dãy các số nguyên 1, 4, 5, -9, 10

* Ma trận các số nguyên

Yêu cầu
Đặc tả nội dung & cấu cấu trúc (với DTD hay Xml-schema) của tài liệu XML tương ứng ma trận các số nguyên

```
 1   4  12
-9 10  20
 0  4   44
```

* Đa giác

Yêu cầu
Đặc tả nội dung & cấu cấu trúc (với DTD hay Xml-schema) của tài liệu XML tương ứng đa giác ABCDE với
A(0,0) , B(1,6) , C(1,1) , D(7,7) , E(0,2)

* Biểu thức số nguyên

Yêu cầu
Đặc tả nội dung & cấu cấu trúc (với DTD hay Xml-schema) của tài liệu XML tương ứng biểu thức số học (chỉ bao gồm các số nguyên dương và 2 phép toán +, *)
P = 4*5 +10*2*6 + 15

* Biểu thức phân số

Yêu cầu
Đặc tả nội dung & cấu cấu trúc (với DTD hay Xml-schema) của tài liệu XML tương ứng biểu thức phân số (chỉ bao gồm các phân số và 2 phép toán +, *)
P = 4/5 +10/11*2/7 + 1/6*1/2*1/3 + 15/17

* Danh sách các khối lớp

Yêu cầu
Đặc tả nội dung & cấu cấu trúc (với DTD hay Xml-schema) của tài liệu XML tương ứng danh sách các khối lớp của trường cấp X. Biết rằng trường X có 3 khối lớp 10,11,12.
Khối 10 có 8 lớp: 10A1, 10A2,10A3, 10A4, 10A5,10A6,10A7,10A8
Khối 11 có 7 lớp : 11A1,11A2,11A4,11A5,11A6,11A7,11A8
Khối 12 có 5 lớp : 12A1, 12A2,12A4, 12A6,12A8

*** Bàn cờ gánh**

Yêu cầu
Đặc tả nội dung & cấu cấu trúc (với DTD hay Xml-schema) của tài liệu XML tương ứng trạng thái của một bàn cớ gánh

*** Phiếu điểm**

Yêu cầu
Đặc tả nội dung & cấu cấu trúc (với DTD hay Xml-schema) của tài liệu XML tương ứng phiếu điểm của một học sinh

Phiếu điểm
Họ và tên : ..... Giới tính :....
Ngày sinh :....
Địa chỉ
Môn học TBHK1 TBHK2 TBNK
....
.....

*** Hóa đơn bán hàng**

Yêu cầu
Đặc tả nội dung & cấu cấu trúc (với DTD hay Xml-schema) của tài liệu XML tương ứng hóa đơn bán hàng

Hóa đơn bán hàng
Khách hàng : .....
Ngày lập :....
Stt Mặt hàng Số lượng Đơn giá Thành tiến
....
.....

Tổng trị giá : .....

*** Bảng chấm công**

Yêu cầu
Đặc tả nội dung & cấu cấu trúc (với DTD hay Xml-schema) của tài liệu XML tương ứng bảng chấm công tháng của một đơn vị

Bảng chấm công tháng .... đơn vị .....
Nhân viên Số ngày công
....
.....

2. Xây dựng ứng dụng

Yêu cầu chung
Thiết kế và lập trình ứng dụng với các yêu cầu chức năng cho trước

Hướng dẫn chung
1. Thiết kế dữ liệu
   Sử dụng tập tin Xml biểu diễn thông tin các đối tượng trong thực tế
2. Thiết kế xử lý
   Sử dụng (n+1) các đơn thể với n là số lượng các loại thẻ có trong tập tin Xml
   (Đơn thể xử lý chính và n đơn thể xử lý trên n loại thẻ khác nhau)
   Đơn thể xử lý chính Bao gồm (m+1) hàm xử lý m hàm xử lý tương ứng m chức năng
      Hàm xử lý chính Main và
        Khai báo các biến
        Đọc dữ liệu từ tập tin Xml vào các biến liên quan
        Xuất thực đơn
        Chi_so= Chức năng chọn từ người dùng
        Gọi thực hiện hàm xử lý chức năng tương ứng với Chi_so

   Đơn thể xử lý trên kiểu dữ liệu X (thẻ loại X) XL_X - Kểu cấu trúc với các thành phần tương ứng thuộc tính của thẻ
   - Các hàm xử lý liên quan kiểu đang xét

<u>* Tính tiền thuê phòng</u>

Hệ thống thực tế
Khách sạng X có địa chỉ 123 ABC và điện thoại 333111 có bảng đơn giá thuê phòng như sau

Loại phòng	Đơn giá/Ngày
Loại A	250.000
Loại B	220.000
Loại C	180.000
Đặc biệt	340.000

Ghi chú :
  Nếu khách thuê quá 5 ngày được giảm 10%

Yêu cầu
Thiết kế và lập trình ứng dụng tính tiến thuê phòng với các yêu cầu chức năng như sau
1. Cập nhật thông tin về khách sạn
2. Bổ sung loại phòng mới
3. Cập nhật thông tin về loại phòng
4. Thanh lý loại phòng
5. Tính tiền thuê phòng

Hướng dẫn thiết kế
1. Thiết kế dữ liệu
   Sử dụng tập tin Khach_san.xml với
   Thẻ gốc : Biểu diễn khách sạn
   Các thẻ con của thẻ gốc : Biểu diễn các loại phòng
2. Thiết kế xử lý
   Sử dụng 3 đơn thể các hàm xử lý
   Đơn thể xử lý chính Tinh_tien_thue_phong

Bao gồm hàm xử lý chính Main và 5 hàm xử lý tương ứng 5 chức năng

Đơn thể XL_KHACH_SAN

- Kiểu cấu trúc KHACH_SAN
- Các hàm xử lý liên quan khách sạn

Đơn thể XL_LOAI_PHONG

- Kiểu cấu trúc LOAI_PHONG
- Các hàm xử lý liên quan loại phòng

- Dữ liệu

Đặc tả cấu trúc (với DTD)

```
<!DOCTYPE  KHACH_SAN [
    <!ELEMENT KHACH_SAN (LOAI_PHONG+) >
     <!ATTLIST KHACH_SAN    Ten   CDATA ,
                            Dien_thoai CDATA ,
                               Dia_chi CDATA ,
                            Muc_giam CDATA  ,
                            Ty_le_giam CDATA >
    <!ELEMENT LOAI_PHONG EMPTY >
    <!ATTLIST LOAI_PHONG   Ten  CDATA, Don_gia CDATA >
]>
```

Đặc tả cấu trúc (với Xml-Schema)

```
<xs:schema id="Khach_san"
                xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <xs:element name ="KHACH_SAN" type ="K_KHACH_SAN" />
 <xs:complexType name ="K_KHACH_SAN">
   <xs:sequence>
     <xs:element name ="LOAI_PHONG"            type="K_LOAI_PHONG" minOccurs="1" />
   </xs:sequence>

   <xs:attribute  name ="Ten"  type ="xs:string" />
   <xs:attribute  name ="Dien_thoai"  type ="xs:string" />
   <xs:attribute  name ="Dia_chi"  type ="xs:string" />
   <xs:attribute  name ="Muc_giam"  type ="xs:int" />
   <xs:attribute  name ="Ty_le_giam"  type ="xs:double" />
 </xs:complexType>
 <xs:complexType  name ="K_LOAI_PHONG">
   <xs:attribute  name ="Ten"  type ="xs:string" />
   <xs:attribute  name ="Don_gia"  type ="xs:int" />
 </xs:complexType>
</xs:schema>
```

Nội dung :

```
<KHACH_SAN  Ten="222"   Dien_thoai="2222"   Dia_chi="33333"
                    Muc_giam="7"   Ty_le_giam="10">
 <LOAI_PHONG  Ten="Loại   A"   Don_gia="250000" />
```

```
 <LOAI_PHONG  Ten="Loại   B"   Don_gia="220000" />
 <LOAI_PHONG  Ten="Loại   C"   Don_gia="180000" />
 <LOAI_PHONG  Ten="Đặc biệt"  Don_gia="380000" />
</KHACH_SAN>
```

- Đơn thể xử lý chính

```
Module Tinh_tien_thue_phong
    Dim Ks As KHACH_SAN
    Dim Duong_dan As String = "..\\..\\Du_lieu\\Khach_san.xml"
    Public Sub Main()
        Ks = XL_KHACH_SAN.Doc(Duong_dan)
        Dim Chuoi As String = "Chương trình tính tiền thuê phòng khách sạn" & vbCrLf
        Chuoi &= XL_KHACH_SAN.Chuoi(Ks) & vbCrLf
        Chuoi &= XL_KHACH_SAN.Chuoi_bang_don_gia(Ks) & vbCrLf
        Console.Write(Chuoi)
        Dim Thuc_don As String = ""
        Thuc_don &= "1. Cập nhật thông tin về khách sạn" & vbCrLf
        Thuc_don &= "2. Bổ sung loại phòng mới" & vbCrLf
        Thuc_don &= "3. Cập nhật thông tin về loại phòng" & vbCrLf
        Thuc_don &= "4. Thanh lý loại phòng" & vbCrLf
        Thuc_don &= "5. Tính tiền thuê phòng " & vbCrLf
        Thuc_don &= "6. Kết thúc " & vbCrLf

        Dim Chi_so As Integer
        Do
            Console.Write(Thuc_don)
            Chi_so = XL_SO_NGUYEN.Nhap("Chọn chức năng : ", 1, 6)
            If Chi_so = 1 Then
                Cap_nhat_thong_tin_khach_san()
            ElseIf Chi_so = 2 Then
                Bo_sung_loai_phong_moi()
            ElseIf Chi_so = 3 Then
                Cap_nhat_thong_tin_loai_phong()
            ElseIf Chi_so = 4 Then
                Thanh_ly_loai_phong()
            ElseIf Chi_so = 5 Then
                Tinh_tien_thue_phong()
            ElseIf Chi_so = 6 Then
                Ket_thuc()
            End If
        Loop While Chi_so <> 6
    End Sub
    Public Sub Cap_nhat_thong_tin_khach_san()
        ' Xuất thông tin hiện nay của khách sạn
        ' Nhập thông tin mới vào Ks
        ' Ghi Ks với Duong_dan
    End Sub
    Public Sub Bo_sung_loai_phong_moi()
```

```
        Dim Lp As LOAI_PHONG
        ' Nhập thông tin cho Lp
        ' Bổ sung Lp vào danh sách loại phòng của Ks
        ' Ghi Ks với duong_dan
    End Sub
    Public Sub Cap_nhat_thong_tin_loai_phong()
        Dim Lp As LOAI_PHONG
        '  Xuất thông tin bảng đơn giá ( danh sách loại phòng)
        '  Cho người dùng nhập Chi_so của loại phòng cần cập nhật
        '  Nhập thông tin cho Lp
        '  Cập nhật loại phòng thứ Chi_so với  Lp
        '  Ghi Ks với Duong_dan
    End Sub
    Public Sub Thanh_ly_loai_phong()
        '  Xuất thông tin bảng đơn giá ( danh sách loại phòng)
        '  Cho người dùng nhập Chi_so của loại phòng cần thanh lý
        '  Xóa loại phòng thứ Chi_so trong damnh sách loại phòng
        '  Ghi Ks với Duong_dan
    End Sub
    Public Sub Tinh_tien_thue_phong()
        Dim So_ngay As Integer
        Dim Lp As LOAI_PHONG

        Dim Tien As Integer

        '  Nhập So_ngay          ' Nhập Chi_so của loại phòng thuê
         ' Lp = Loại phòng thứ Chi_so trong danh sách loại phòng của Ks
         ' Tính tiền dựa trên qui tắc giảm giá
         ' Tạo và xuất chuỗi kết xuất Chuoi                    '
    End Sub
    Public Sub Ket_thuc()
        Console.WriteLine("Chào tạm biệt")
        Console.ReadLine()
    End Sub
End Module
```

- Đơn thể XL_KHACH_SAN

```
Imports System.Xml
Structure KHACH_SAN
    Public Ten As String
    Public Dien_thoai As String
    Public Dia_chi As String
    Public Muc_giam As Integer
    Public Ty_le_giam As Double
    Public Danh_sach_loai_phong As ArrayList
End Structure
Module XL_KHACH_SAN

    Public Function Doc(ByVal Duong_dan As String) As KHACH_SAN
```

```
    Dim Kq As KHACH_SAN
    Dim Tai_lieu As New XmlDocument
    Tai_lieu.Load(Duong_dan) ' Chưa xử lý lỗi
    Dim Goc As XmlElement = Tai_lieu.DocumentElement
    Kq.Ten = Goc.GetAttribute("Ten")
    Kq.Dien_thoai = Goc.GetAttribute("Dien_thoai")
    Kq.Dia_chi = Goc.GetAttribute("Dia_chi")
    Kq.Muc_giam = Goc.GetAttribute("Muc_giam")
    Kq.Ty_le_giam = Goc.GetAttribute("Ty_le_giam")
    Kq.Danh_sach_loai_phong = New ArrayList
    For Each Nut_loai_phong As XmlElement In Goc.ChildNodes
        Dim Loai_phong As LOAI_PHONG = XL_LOAI_PHONG.Khoi_tao(Nut_loai_phong)
        Kq.Danh_sach_loai_phong.Add(Loai_phong)
    Next
    Return Kq
End Function

Public Function Ghi(ByVal Ks As KHACH_SAN, ByVal Duong_dan As String) As Boolean
    Dim Kq As Boolean = True
    Dim Tai_lieu As New XmlDocument
    Dim Goc As XmlElement = Tai_lieu.CreateElement("KHACH_SAN")
    Goc.SetAttribute("Ten", Ks.Ten)
    Goc.SetAttribute("Dien_thoai", Ks.Dien_thoai)
    Goc.SetAttribute("Dia_chi", Ks.Dia_chi)
    Goc.SetAttribute("Muc_giam", Ks.Muc_giam)
    Goc.SetAttribute("Ty_le_giam", Ks.Ty_le_giam)
    Tai_lieu.AppendChild(Goc)
    For Each Lp As LOAI_PHONG In Ks.Danh_sach_loai_phong
        Dim Nut_Lp As XmlElement = XL_LOAI_PHONG.Nut(Lp, Tai_lieu)
        Goc.AppendChild(Nut_Lp)
    Next
    Tai_lieu.Save(Duong_dan)
    Return Kq
End Function

Public Function Chuoi(ByVal Ks As KHACH_SAN) As String
    Dim Kq As String = ""
    Kq &= "Khách sạn:" & Ks.Ten & vbCrLf
    Kq &= "Điện thoại " & Ks.Dien_thoai & vbCrLf
    Kq &= "Địa chỉ " & Ks.Dia_chi & vbCrLf
    Kq &= "Mức giảm " & Ks.Muc_giam & vbCrLf
    Kq &= "Tỷ lệ giảm " & Ks.Ty_le_giam
    Return Kq
End Function

Public Function Chuoi_bang_don_gia(ByVal Ks As KHACH_SAN) As String
    Dim Kq As String = ""
    Kq &= "Stt" & vbTab & "Loại phòng" & vbTab & "Đơn giá " & vbCrLf
    For i As Integer = 0 To Ks.Danh_sach_loai_phong.Count - 1
        Kq &= i & vbTab
```

```
            Kq &= XL_LOAI_PHONG.Chuoi(Ks.Danh_sach_loai_phong(i)) & vbCrLf
        Next
        Return Kq
    End Function

End Module
```

- Đơn thể XL_LOAI_PHONG

```
Imports System.Xml
Public Structure LOAI_PHONG
    Public Ten As String
    Public Don_gia As Integer
End Structure
Module XL_LOAI_PHONG

    Public Function Khoi_tao(ByVal Nut As XmlElement) As LOAI_PHONG
        Dim Kq As LOAI_PHONG
        Kq.Ten = Nut.GetAttribute("Ten")
        Kq.Don_gia = Nut.GetAttribute("Don_gia")
        Return Kq
    End Function

    Public Function Nut(ByVal Lp As LOAI_PHONG, ByVal Tai_lieu As XmlDocument) As
XmlElement
        Dim Kq As XmlElement = Tai_lieu.CreateElement("LOAI_PHONG")
        Kq.SetAttribute("Ten", Lp.Ten)
        Kq.SetAttribute("Don_gia", Lp.Don_gia)
        Return Kq
    End Function

    Public Function Chuoi(ByVal Lp As LOAI_PHONG) As String
        Dim Kq As String = ""
        Kq &= Lp.Ten & vbTab
        Kq &= Lp.Don_gia
        Return Kq
    End Function

End Module
```

*** Đổi ngoại tệ**

Hệ thống thực tế
Cửa hàng đổi tiền X có địa chỉ 124 ABC và điện thoại 333112 có bảng tỷ giá như sau

Ngoại tệ	Tỷ giá
....	
....	

Yêu cầu
Thiết kế và lập trình ứng dụng đổi ngoại tệ với các yêu cầu chức năng như sau

1. Cập nhật thông tin về của hàng đổi tiền
2. Bổ sung ngoại tệ mới
3. Cập nhật tỷ giá của ngoại tệ
4. Thanh lý ngoại tệ
5. Tính tiền khi đổi ngoại tệ

Hướng dẫn thiết kế
1. Thiết kế dữ liệu
   Sử dụng tập tin Cua_hang.xml với
   Thẻ gốc : Biểu diễn thông tin về cửa hàng
   Các thẻ con của thẻ gốc : Biểu diễn các ngoại tệ
2. Thiết kế xử lý
   Sử dụng 3 đơn thể các hàm xử lý
   Đơn thể xử lý chính Doi_ngoai_te
      Bao gồm hàm xử lý chính Main và 5 hàm xử lý tương ứng 5 chức năng
   Đơn thể XL_CUA_HANG
     - Kiểu cấu trúc CUA_HANG
     - Các hàm xử lý liên quan cửa hàng đổi tiền
   Đơn thể XL_NGOAI_TE
     - Kiểu cấu trúc NGOAI_TE
     - Các hàm xử lý liên quan ngoại tệ

*** Trắc nghiệm**

Yêu cầu
Thiết kế và lập trình ứng dụng trắc nghiệm với các yêu cầu sau
1. Biên soạn câu hỏi mới
2. Cập nhật câu hỏi đã soạn
3. Xóa câu hỏi đã soạn
4. Tự rèn luyện với các câu hỏi đã soạn

Ghi chú :
- Câu hỏi trắc nghiệm chỉ bao gồm văn bản, có nhiều chọn lựa khác nhau và chỉ có duy nhất một đáp án đúng
- Với chức năng tự rèn luyện, ứng dụng phát sinh ngẫu nhiên 1 câu hỏi

Hướng dẫn thiết kế
1. Thiết kế dữ liệu
   Sử dụng tập tin Trac_nghiem.xml với
   Thẻ gốc (DE_BAI) : Biểu diễn thông tin chung của các câu hỏi (nếu cần thiết)
   Các thẻ con của thẻ gốc (CAU_HOI) : Biểu diễn các câu hỏi
   Các thẻ con của thẻ CAU_HOI : Biểu diễn các chọn lựa

2. Thiết kế xử lý
   Sử dụng 4 đơn thể các hàm xử lý
   Đơn thể xử lý chính Trac_nghiem
     Bao gồm hàm xử lý chính Main và 4 hàm xử lý tương ứng 4 chức năng
   Đơn thể XL_DE_BAI
     - Kiểu cấu trúc DE_BAI
     - Các hàm xử lý liên quan danh sách các câu hỏi trong đề bài

Đơn thể XL_CAU_HOI
- Kiểu cấu trúc CAU_HOI
- Các hàm xử lý liên quan câu hỏi

Đơn thể XL_CHON_LUA
- Kiểu cấu trúc CHON_LUA
- Các hàm xử lý liên quan chọn lựa

*** Xếp hạng bóng đá**

Yêu cầu :
Thiết kế và lập trình ứng dụng xếp hạng bóng đá của giải vô địch bóng đá VN với các yêu cầu sau
1. Ghi nhận kết quả các trận đấu
2. Lập bảng xếp hạng

Ghi chú :
Đây là bài tập dành cho các sinh viên tự xếp mình vào loại khá/giỏi
=== > Không có hướng dẫn thêm

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

*** 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)

*** 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

*** Mô hình đối tượng dữ liệu**

Đối tượng dữ liệu
Môt 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 Stuio.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 Microsof Office
Word
Excel

*** Mô hình đối tượng thể hiện**

Đối tượng thể hiện
Môt 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 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)
Mô hình đối tượng thể hiện thứ 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 của 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
1. 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 câu trúc theo các cột của bảng
- Cột khóa chính là cột cho phép xác định duy nhất dòng trrong 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 câu 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 trrong 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 câu 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 trrong 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

1. 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 thể 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 XmlDocumen
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 vần Xpath	
selectNodes	Trả về một tập hợp nút (NodeList) theo dựa trên chuỗi truy vần 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	
reaplaceChild	Thay thế một nút con này bằng nút con khác của nút đang xét	
cloneNode	Tão bản sao của nút đang xét	Sử dụng tham số để quyết định tạo “bản sau cạn” (không xét các nút con) hay “bản sau sâu” (ngược lại)

*** XmlDocument**

Đối tượngXmlDocument

Hàm	Ý nghĩa	Ghi chú
createElement	Trả về nút (Element) với tên nút	
creatAttribute	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)	Sử dụng cho mọi loại tài liệu có cấu trúc

*** XmlElement**

ĐốitượngXmlElement

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 họp các nút con (NodeList) theo tên trong tham số	

2. Một số kỹ thuật lập trình với DOM

*** 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
Thuật giải
Khai báo đối tượng XmlDocument Kq
Kq = Dữ liệu đọc từ tập tin Xml với Duong_dan
(Xử lý thông báo lỗi nếu đường dẫn sai hay tập tin tương ứng không có cấu trúc tập tin XML)
Trả Kq

Ghi chú :
Để có thẻ tái sử dụng hàm trên có thể tạo hàm đọc tài liệu
- Bên trong một đơn thể (ví dụ LT_XML)
- Bên trong một lớp đối tượng (với khai báo là hàm mức lớp)

Lập trình với VB.NET

```
Public Function Doc_tai_lieu(ByVal Duong_dan As String) As XmlDocument
    Dim Kq As New XmlDocument
```

```
        Try
            Kq.Load(Duong_dan)
        Catch Loi As Exception
            Dim Thong_bao As String = "Lỗi khi đọc tập tin " & Duong_dan & vbCrLf
            Thong_bao &= Loi.Message
            Console.WriteLine(Thong_bao)
            ' Hay MessageBox.Show(Thong_bao)

        End Try
        Return Kq
    End Function
```

- Bài tập 1

Mục tiêu :

- Tạo lập và sử dụng hàm đọc tài liệu === > Tái sử dụng cho các ứng dụng khác
- Tìm hiểu về các lỗi có thể có khi đọc

Yêu cầu :
Viết chương trình nhập vào đường dẫn của một tập tin Xml và sau đó đọc, xuất nội dung dưới dạng chuỗi của các thẻ bên trong tập tin Xml

Ghi chú :
Cần thử nghiệm với các trường hợp lỗi khác nhau
- Đường dẫn sai
- Tập tin không có cấu trúc theo định chuẩn XML

- Bài tập 2

Mục tiêu :
- Tìm hiểu và sử dụng một số hàm của DOM
- Mở đầu cho việc trình bày ngôn ngữ truy vấn XPath

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

*** 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
Thuật giải
Khai báo biến logic Kq
Kq = Kết quả khi ghi
(Xử lý thông báo lỗi khi ghi)
Trả Kq

Ghi chú :
Để có thể tái sử dụng hàm trên có thể tạo hàm đọc tài liệu
- Bên trong một đơn thể (ví dụ LT_XML)
- Bên trong một lớp đối tượng (với khai báo là hàm mức lớp)

Lập trình với Vb.NET

```
Public Function Ghi_tai_lieu(ByVal Tai_lieu As XmlDocument, ByVal Duong_dan As String) As
Boolean
    Dim Kq As Boolean = True
    Try
        Tai_lieu.Save(Duong_dan)
    Catch Loi As Exception
        Kq = False
        Dim Thong_bao As String = "Lỗi khi ghi tập tin " & Duong_dan & vbCrLf
        Thong_bao &= Loi.Message
         Console.WriteLine(Thong_bao)
        ' Hay MessageBox.Show(Thong_bao)
    End Try

    Return Kq
End Function
```

- Bài tập 1

Mục tiêu :

- Tạo lập và sử dụng hàm ghi tài liệu === > Tái sử dụng cho các ứng dụng khác
- Tìm hiểu về các lỗi có thể có khi ghi
- Tìm hiểu và sử dụng một số hàm của DOM

Yêu cầu :
Viết chương trình nhập vào 2 đường dẫn tương ứng 2 tập tin Xml. Đọc 2 tập tin trên vào 2 đối tượng XmlDocument Tai_lieu_1,Tai_lieu_2 và sau đó tạo đối tượng Tai_lieu bao gồm tất cả các đối tượng của Tai_lieu_1,Tai_lieu_2. Ghi Tai_lieu vào tập tin Kq.xml

- Bài tập 2

Mục tiêu :
- Tìm hiểu và sử dụng một số hàm của DOM
- Mở đầu cho việc trình bày về Xpath

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

*** 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

*** 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_l;eu
Gán các thuộc tính của Kq tương ứng các biến thành phần của đối tượng
Trả Kq

*** Thể 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 :
Thể hiện cây Công ty - Đơn vị
Thể hiện cây Trường - Khối - Lớp
Thể hiện cây các số nguyên

Hướng giải quyết chung :
Sử dụng đối tượng thể hiện cây trong thư viện đối tượng giao diện người dùng

Hướng giải quyết cụ thể với Visual Studio.NET
Sử dụng đối tượng thể hiện TreeView

3. Ngôn ngữ XPath

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

*** 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 họp con thông tin của một tập họp thông tin nào đó

Tập họp thông tin

Khái niệm chung mô tả các thông tin được biểu diễn theo một dạng nào đó. Đặc điểm của tập họp thông tin đang xét là các thông tin trong tập họp này phải bao gồm bên trong các thành phần con Tp1,Tp2, ... theo một cấu trúc nào đó.

Ví dụ :
- Mảng 1 chiều các số nguyên là tập họp các thông tin với các thành phần con là các số nguyên
- Mảng 2 chiều các phân số là tập họp 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 họp 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 phân mềm quản lý nhân sự là tập họp 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 họp thông tin với các thành phần con là các nút. Các nút này được sắp xếp theo dạng cây

Tập họp con thông tin

Khái niệm cho phép mô tả kết quả của việc truy vấn thông tin trên tập họp thông tin gốc. Tập họp 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 họp gốc nhưng không nhất thiết.

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

*** 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 giãn 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 sỡ 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 vần 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 thướng cần sử dụng thành phân lọc để mô tả chi tiết hơn thông tin cần truy vấn

Mỗi ngôn ngữ truy vần 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 mẩu tin cần truy vấn

- Ngôn ngữ Xpath cho phép định theo đuờng 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 , thự 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 chó phép lọc các nút cần truy vấn

*** 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_ví_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 họp 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 họp các lớp có sĩ 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 họp 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

Chuỗi định vị

Xpath cho phép định vị theo 2 cách

Cách 1 : Dùng trục định vị với tên cụ thể

Cách 2 : Dùng dạng tốc ký với các từ khóa thay thế tên trục 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 vi 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
peceding	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 vi 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 (kết quả trung gian) theo 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

<u>Chuỗi lọc</u>

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)

*** Minh họa trực quan XPath**

Đoạn videoClip minh họa trực quan việc sử dụng Xpath

IV. Bài tập

1. Rèn luyện kỹ năng

*** Đọc tập tin Xml**

Mục tiêu :

- Tạo lập và sử dụng hàm đọc tài liệu === > Tái sử dụng cho các ứng dụng khác
- Tìm hiểu về các lỗi có thể có khi đọc

Yêu cầu :
Viết chương trình nhập vào đường dẫn của một tập tin Xml và sau đó đọc, xuất nội dung dưới dạng chuỗi của các thẻ bên trong tập tin Xml

Ghi chú :
Cần thử nghiệm với các trường hợp lỗi khác nhau

- Đường dẫn sai
- Tập tin không có cấu trúc theo định chuẩn XML

*** Đọc tập tin Xml và truy vấn**

Mục tiêu :

- Tìm hiểu và sử dụng một số hàm của DOM
- Sử dụng ngôn ngữ truy vấn XPath

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

*** Ghi tập tin Xml**

Mục tiêu :

- Tạo lập và sử dụng hàm ghi tài liệu === > Tái sử dụng cho các ứng dụng khác
- Tìm hiểu về các lỗi có thể có khi ghi
- Tìm hiểu và sử dụng một số hàm của DOM

Yêu cầu :
Viết chương trình nhập vào 2 đường dẫn tương ứng 2 tập tin Xml. Đọc 2 tập tin trên vào 2 đối tượng XmlDocument Tai_lieu_1,Tai_lieu_2 và sau đó tạo đối tượng Tai_lieu bao gồm tất cả các đối tượng của Tai_lieu_1,Tai_lieu_2. Ghi Tai_lieu vào tập tin Kq.xml

*** Truy vấn và Ghi tập tin Xml**

Mục tiêu :
- Tìm hiểu và sử dụng một số hàm của DOM
- Sử dụng Xpath

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

*** Thể hiện cây tổ chức trường**

Mục tiêu :
Rèn luyện kỹ năng trình bày nội dung tài liệu Xml với Windows Form

Yêu cầu :
Tạo tập tin Xml biểu diễn thông tin về tổ chức của một trường (thông tin về trường, các khối, các lớp , các học sinh)
Viết chương trình đọc và xuất các thông tin trên dưới dạng cây

*** Thể hiện cây các số nguyên**

Mục tiêu :
Rèn luyện kỹ năng trình bày nội dung tài liệu Xml với Windows Form

Yêu cầu :
Tạo tập tin Xml biểu diễn thông tin về cây các số nguyên
Viết chương trình đọc và xuất các thông tin trên dưới dạng cây

2. Xây dựng ứng dụng

*** Tính tiền thuê phòng**

Hệ thống thực tế
Khách sạng X có địa chỉ 123 ABC và điện thoại 333111 có bảng đơn giá thuê phòng như sau

Loại phòng	Đơn giá/Ngày
Loại A	250.000
Loại B	220.000
Loại C	180.000
Đặc biệt	340.000

Ghi chú :
Nếu khách thuê quá 5 ngày được giảm 10%

Yêu cầu
Thiết kế và lập trình ứng dụng tính tiến thuê phòng với các yêu cầu chức năng như sau
1. Cập nhật thông tin về khách sạn
2. Bổ sung loại phòng mới
3. Cập nhật thông tin về loại phòng
4. Thanh lý loại phòng
5. Tính tiền thuê phòng

*** Đổi ngoại tệ**

Hệ thống thực tế
Cửa hàng đổi tiền X có địa chỉ 124 ABC và điện thoại 333112 có bảng tỷ giá như sau

Ngoại tệ	Tỷ giá
....	
....	

Yêu cầu
Thiết kế và lập trình ứng dụng đổi ngoại tệ với các yêu cầu chức năng như sau
1. Cập nhật thông tin về của hàng đổi tiền
2. Bổ sung ngoại tệ mới
3. Cập nhật tỷ giá của ngoại tệ
4. Thanh lý ngoại tệ
5. Tính tiền khi đổi ngoại tệ

*** Trắc nghiệm**

Yêu cầu
Thiết kế và lập trình ứng dụng trắc nghiệm với các yêu cầu sau
1. Xem đề bài
2. Biên soạn câu hỏi mới
3. Cập nhật câu hỏi đã soạn
4. Xóa câu hỏi đã soạn
5. Tự rèn luyện với các câu hỏi đã soạn

Ghi chú :
- Câu hỏi trắc nghiệm chỉ bao gồm văn bản, có nhiều chọn lựa khác nhau và chỉ có duy nhất một đáp án đúng
- Với chức năng tự rèn luyện, ứng dụng phát sinh ngẫu nhiên 1 câu hỏi

*** Xếp hạng bóng đá**

Yêu cầu :Thiết kế và lập trình ứng dụng xếp hạng bóng đá của giải vô địch bóng đá VN với các yêu cầu sau

1. Ghi nhận kết quả các trận đấu
2. Lập bảng xếp hạng

Ghi chú :

Đây là bài tập dành cho các sinh viên tự xếp mình vào loại khá/giỏi

Chương 4 : Biến đổi tài liệu XML với XSLT

I. Mở đầu về XSLT

Chương trình XSLT :
Khái niệm :

- Một loại tài liệu XML đặc biệt bao gồm các thẻ xử lý cho phép biến đổi một tài liệu XML thành một tài liệu văn bản bất ký
- Một loại chương trình thông dịch đặc biệt với
  + Dữ liệu nguồn : Tài liệu XML
  + Kết xuất : Tài liệu dạng văn bản

Tài liệu XML ---- > Chương trình XSLT ---- > Tài liệu văn bản

Các ứng dụng chính :

XSLT có 2 ứng dụng chính hiện nay
1. Thực hiện biến đổi từ tập tin XMl vào trang Web với ngôn ngữ HTML
2. Thực hiện biến đổi từ tập tin XML vào tập tin XML khác

Xml --- > Html
Chp phép thể hiện nội dung tập tin Xml trên trang Web
Ví dụ :
Tập tin Xml Don_thuc.xml
<DON_THUC He_so="4" So_mu="6" />
thông qua xử lý của chương trình Xuat_don_thuc.Xslt sẽ được thể hiện trên trang Web
4x6

Xml -- > Xml
Cho phép tạo tập tin Xml mới từ tập tin Xml hiện có để có thể trích rút thông tin, tái cấu trúc các thẻ, v.v...
Ví dụ :
Tập tin Xml Don_thuc.xml
<DON_THUC He_so="4" So_mu="6" />
thông qua xử lý của một chương trình Xslt sẽ tạo ra tập tin Don_thuc_1.xml như sau

```
<DON_THUC>
        <He_so> 4 </He_so>
        <So_mu> 6 </So_mu>
</DON_THUC>
```

1. Cấu trúc chương trình XSLT

Cấu trúc chương trình XSLT
Cấu trúc chương trình XSLT đơn giãn

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

```
    <xsl:template  match="/" >
          Các lệnh ( thẻ )  xử lý cho phép trích rút thông tin từ Tập tin Xml nguồn và kết xuất vào tập tin kết quả
    </xsl:template>
</xsl:stylesheet>
```

Ví dụ 1 :

Chương trình sau cho phép biến đối tập tin Nguoi_dung.xml

```
<NGUOI_DUNG Ho_ten="Trần văn Long" />
```

để tạo tập tin văn bản với nội dung

Xin chào Trần văn Long. Đây là chương trình XSLT đầu tiên của tôi

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:output method="text"/>
  <xsl:template match="/" >
   Xin chào <xsl:value-of select="/NGUOI_DUNG/@Ho_ten"/>
   .Đây là chương trình XSLT đầu tiên của tôi
  </xsl:template>

</xsl:stylesheet>
```

Ví dụ 2 : Chương trình sau cho phép biến đối tập tin xml

```
<GOC>
   <SO Gia_tri="5" />
   <SO Gia_tri="7" />
 </GOC>
```

để tạo tập tin văn bản với nội dung

5+7 =12

```
<?xmlversion="1.0"encoding="UTF-8" ?>
<xsl:stylesheetversion="1.0"xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:outputmethod ="text"/>
 <xsl:template match="/" >
   <xsl:value-of select="/GOC/SO[1]/@Gia_tri"/> +
   <xsl:value-of select="/GOC/SO[2]/@Gia_tri"/> =
   <xsl:value-of select="/GOC/SO[1]/@Gia_tri + /GOC/SO[2]/@Gia_tri" />
 </xsl:template>
</xsl:stylesheet>
```

2. Cho thực hiện chương trình XSLT

Cho thực hiện chương trình XSLT

Quá trình thực hiện

Bao gồm 3 bước

Bước 1 : Chuẩn bị dữ liệu nguồn là tập tin XML

Bước 2 : Soạn thảo chương trình XSLT

Bước 3 : Cho thực hiện chương trình

Bước 1 : Dữ liệu nguồn có thể được chuẩn bị thông qua một trong các cách sau

- Cách 1 : Sử dụng trình soạn thảo văn bản bất kỳ (vì tài liệu XML chỉ là một văn bản)
- Cách 2 : Sử dụng trình soạn thảo XML Editor

Bước 2 : Chương trình XSLT có thể được chuẩn bị thông qua một trong các cách sau
- Cách 1 : Sử dụng trình soạn thảo văn bản bất kỳ (vì tài liệu XML chỉ là một văn bản)
- Cách 2 : Sử dụng trình soạn thảo XML Editor
- Cách 3 : Sử dụng trình soạn thảo chương trình XSLT (XSLT Editor)

Bước 3 :
Tùy theo mục tiêu của việc thực hiện có thể tiến hành một trong 3 cách sau
Cách 1 : Sử dụng môi trường lập trình
Cho thực hiện trực tiếp bên trong môi trường lập trình
Cách này thích hợp cho việc học tập và thử nghiệm chương trình XSLT
Cách 2 : Sử dụng trình duyệt Web
Cho thực hiện trực tiếp với sự hổ trợ của trình duyệt Web
Cách này cho phép ứng dụng trực tiếp XSLT trong việc thể hiện hto6ng tin trên Web
Cách 3 : Tự viết chương trình
Cho thực hiện thông qu việc viết một ứng dụng trong ngôn ngữ lập trình khác (ví dụ C#). Ứng dụng này sẽ
- Nạp/Đọc chương trình XSLT vào bộ nhớ
- Chuẩn bị dữ liệu nguồn (nếu cần thiết)
- Cho thực hiện
- Xử lý kết xuất được tạo ra (nếu cần thiết)

Cách này thích hợp khi cần "nhúng" chương trình XSLT vào một ứng dụng để có thể thực hiện nhanh, dễ bảo trì, chuẩn một số xử lý biến đổi nào đó liên quan tài liệu XML

*** Sử dụng môi trường lập trình**

Với môi trường lập trình Visual Studio.NET
Bươc 1 : Tạo tập tin Xml nguồn
Chọn Project - Add New Item với loại tập tin là Xml
=== > Cửa sổ cho phép soạn thảo tập tin Xml

Bước 2 : Tạo chương trính XSLT
Chọn Project - Add New Item với loại tập tin Xslt
=== > Cửa sổ cho phép soạn thảo chương trình XSLT

Bước 3 : Cho thực hiện
3.1 Bước 3.1 :
Chọn của sổ Properties để xác định tập tin Xml nguồn và tập tin kết xuất
3.2 Bước 3.2 :
Quay về cửa sổ soạn thảo chương trình XSLT (Click và cửa sổ) và sau đó chọn chức năng Xml ---> Debug XSLT

Lưu ý :
Bước 3.1 Chỉ cần thực hiện một lần nếu không thay đổi tập tin nguồn
Có thể đánh dấu điểm ngắt bên trong chương trình XSLT tương tự khi Debug ứng dụng với ngôn ngữ lập trình khác

*** Sử dụng trình duyệt Web**

Bươc 1 : Tạo tập tin Xml nguồn
với chỉ thị yêu cầu thực hiện chương trình XSL
<?xml-stylesheet type="text/xsl" href=Chuỗi đường dẫn đến tập tin chương trình XSLT ?>

Ví dụ :
<?xmlversion="1.0"encoding="utf-8" ?>
<?xml-stylesheettype="text/xsl" href="Xuat_loi_chao.xslt" ?>
<NGUOI_DUNGHo_ten="Trần văn Long" />

Bước 2 : Tạo chương trính XSLT
Chọn Project - Add New Item với loại tập tin Xslt
=== > Cửa sổ cho phép soạn thảo chương trình XSLT

Bước 3 : Cho thực hiện
Mở trình duyệt Web và sau đó chọn URL là đường dẫn đến tập tin Xml

*** Tự viết chương trình**

Bươc 1 : Tạo tập tin Xml nguồn Bước 2 : Tạo chương trính XSLT
Chọn Project - Add New Item với loại tập tin Xslt
=== > Cửa sổ cho phép soạn thảo chương trình XSLT

Bước 3 : Cho thực hiện
....
Khai báo đối tượng Bo_thuc_hien
Đọc tập tin chương trình XSL vào Bo_thuc_hien
Yêu cầu Bo_thuc_hien thực hiện chương trình XSLT với dữ liệu nguồn và kết xuất
....

Ví dụ : với Visual Studio.NET 2005 VB.NET
Đoạn chương trình sau sẽ cho thực hiện chương trình Xuat_loi_chao.Xslt
- Dữ liệu nguồn là tập tin Nguoi_dung.xml
- Kết xuất là tập tin văn bản Loi_chao.txt
(Tật cả các tập tin đều đặt trong thư mục của Project)

```
Imports System.Xml
Imports System.Xml.Xsl
Module Thuc_hien_XSLT
   Public Sub Main()
      Dim Duong_dan_Xml As String = "..\..\Nguoi_dung.xml"
      Dim Duong_dan_Xslt As String = "..\..\Xuat_loi_chao.xslt"
      Dim Duong_dan_Kq As String = "..\..\Loi_chao.txt"

      Dim Thuc_hien As New XslCompiledTransform(True)
      Thuc_hien.Load(Duong_dan_Xslt)
```

```
        Thuc_hien.Transform(Duong_dan_Xml, Duong_dan_Kq)

    End Sub
End Module
```

3. Các ví dụ minh họa

Các ví dụ minh họa
Mục tiêu :
Minh họa trực quan một số chương trình XSLT.
Các chương trình này sẽ được diễn giải chi tiết về sau trong các mục khác
=== > Chưa yêu cầu hiểu ý nghĩa các lệnh
=== > Sử dụng để rèn luyện cách cho thực hiện chương trình XSLT

* Xuất cây trường - khối - lớp

Với tập tin Truong.xml có nội dung như sau

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

<TRUONGTen="Trướng cấp 3 XXX">
 <KHOITen="Khối 10" >
   <LOPTen="Lớp 10A" />
   <LOPTen="Lớp 10B" />
   <LOPTen="Lớp 10C" />
   <LOPTen="Lớp 10D" />
 </KHOI>
 <KHOITen="Khối 11" >
   <LOPTen="Lớp 11A" />
   <LOPTen="Lớp 11B" />
   <LOPTen="Lớp 11C" />
 </KHOI>
 <KHOITen="Khối 12" >
   <LOPTen="Lớp 12A" />
   <LOPTen="Lớp 12B" />
   <LOPTen="Lớp 12C" />
 </KHOI>
</TRUONG>
```

Chương trình Xuat_truong.xslt sau sẽ kết xuất (dạng Html) các thông tin về trường (bao gồm thông tin khối, lớp)

```
<?xmlversion="1.0"encoding="UTF-8" ?>
<xsl:stylesheetversion="1.0"xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:outputmethod ="html" />

 <xsl:template match="/" >
   <xsl:apply-templates select="TRUONG" />
 </xsl:template>

 <xsl:template match="TRUONG">
```

```
   <xsl:value-of select="@Ten"/>
   <br />
   Danh sách các khối lớp <br />
   <xsl:apply-templates select="KHOI" />
 </xsl:template>
 <xsl:template match="KHOI">
   <xsl:value-of select="@Ten"/>
   <br />
   <xsl:apply-templates select="LOP" />
 </xsl:template>
  <xsl:template match="LOP">
    <xsl:value-of select="@Ten"/>
    <br />
 </xsl:template>

</xsl:stylesheet>
```

Ghi chú :

Thuộc tính select trong thẻ xsl:apply-templates có thể được bỏ qua và khi đó sẽ được hiểu là select="*" (cho lượng giá là các nút con của nút ngữ cảnh)
=== > Một trong các cách đơn giãn tổ chức chương trình Xslt là tổ chức chương trình theo các loại thẻ có trong tập tin Xml và gọi thực hiện (so khớp) không cần tham số
Gọi thực hiện : <xsl:apply-templates />
Khai báo hàm/mẩu so khớp :

```
      <xsl:template match="tên loại thẻ" >
         Các thẻ xử lý
      </xsl:template>
<xsl:stylesheetversion="1.0"xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:outputmethod ="html" />


 <xsl:templatematch="/" >
   <xsl:apply-templates   />
 </xsl:template>

 <xsl:templatematch="TRUONG">
   <xsl:value-ofselect="@Ten"/>
   <br />
   Danh sách các khối lớp <br />
   <xsl:apply-templates   />
 </xsl:template>
 <xsl:templatematch="KHOI">
   <xsl:value-ofselect="@Ten"/>
   <br />
   <xsl:apply-templates   />
 </xsl:template>

 <xsl:templatematch="LOP">
```

```
   <xsl:value-ofselect="@Ten"/>
   <br />
 </xsl:template>

</xsl:stylesheet>
```

*** Xuất danh sách chọn**

Chương trình Xslt sau sẽ xuất danh sách chọn các đơn vị từ tập tin Cong_ty.xml

```
<?xmlversion="1.0"encoding="UTF-8" ?>
<xsl:stylesheetversion="1.0"xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:outputmethod ="html" />
 <xsl:templatematch="/" >
   <html>
     <body>
       <xsl:apply-templates />
     </body>
   </html>
 </xsl:template>
 <xsl:templatematch="CONG_TY" >
   Danh sách đơn vị :
   <select>
     <xsl:apply-templates />
   </select>
 </xsl:template>
 <xsl:templatematch="DON_VI" >
   <option>
     <xsl:value-ofselect ="@Ten"/>
   </option>
 </xsl:template>
</xsl:stylesheet>
```

*** Sắp xếp kết quả thi đấu Olympic**

Với tập tin Xml Ket_qua_Olympic.xml

```
<?xmlversion="1.0"encoding="utf-8" ?>
<KET_QUA>
 <QUOC_GIATen="AAA"So_vang="10"So_bac="7"So_dong="2" />
 <QUOC_GIATen="XXX"So_vang="6"So_bac="0"So_dong="12" />
 <QUOC_GIATen="BBB"So_vang="10"So_bac="8"So_dong="13" />
 <QUOC_GIATen="DDD"So_vang="4"So_bac="17"So_dong="0" />
 <QUOC_GIATen="MMM"So_vang="6"So_bac="1"So_dong="0" />
 <QUOC_GIATen="KKK"So_vang="6"So_bac="0"So_dong="2" />
 <QUOC_GIATen="LLL"So_vang="10"So_bac="4"So_dong="23" />
 <QUOC_GIATen="PPP"So_vang="3"So_bac="27"So_dong="100" />
</KET_QUA>
```

Đoạn chương trình XSL sau sắp xếp các quốc gia giảm dần theo thứ tự ưu tiên

- Ưu tiên 1 : Số huy chương vàng
- Ưu tiên 2 : Số huy chương bạc
- Ưu tiên 3 : Số huy chương đồng

```
<?xmlversion="1.0"encoding="UTF-8" ?>
<xsl:stylesheetversion="1.0"xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:outputmethod ="xml"indent="yes" />
 <xsl:templatematch="/" >
   <xsl:apply-templates />
 </xsl:template>
 <xsl:templatematch="KET_QUA"   >
   <xsl:copy >
     <xsl:apply-templates   select ="QUOC_GIA">
       <xsl:sort    order="descending"data-type="number"select="@So_vang" />
       <xsl:sort    order="descending"data-type="number"select="@So_bac" />
       <xsl:sort    order="descending"data-type="number"select="@So_dong" />

     </xsl:apply-templates>
   </xsl:copy>
 </xsl:template>
 <xsl:templatematch="QUOC_GIA" >
   <!--<xsl:copy-of select="."/>-->
   <xsl:copy >
     <xsl:copy-ofselect ="@*"/>
   </xsl:copy>

 </xsl:template>

</xsl:stylesheet>
```

II. Các thao tác cơ bản

Các thao tác cơ bản
Mục tiêu :
Trình bày các kỹ thuật xử lý cơ bản khi xây dựng chương trình XSLT

Nội dung :
1. Trích rút thông tin và kết xuất với thẻ xử lý xsl:value-of , xsl:variable
2. Xử lý rẽ nhánh với xsl:if , xsl:choose
3. Xử lý lặp với xsl:for-each. Xử lý so khớp với xsl:apply-templates , xsl:template

1. Trích rút và kết xuất thông tin

Vấn đề :
Cần trích một số thông tin trong tập tin Xml nguồn và đưa vào tập tin kết xuất

Hướng giải quyết :

Cách 1 : Trích rút thông tin từ tập tin Xml và sau đó kết xuất trực tiếp với thẻ xử lý

xsl:value -of
Cách 2 : Trích rút thông tin vào biến với thẻ xử lý xsl:variable và sau đó sử dụng biến này trong thẻ xử lý xsl:value-of

Thẻ xsl:value-of

Ý nghĩa :
Cho phép trích rút thông tin từ tập tin Xml hay từ giá trị của biến và sau đó đưa vào tập tin kết quả
Cú pháp :
Nếu trích rút thông tin từ tập tin Xml nguồn
<xsl:value-of select="Biểu thức Xpath" />
Nếu trích rút thông tin từ biến
<xsl:value-of select="$Ten_bien" />

Thẻ xsl:variable

Ý nghĩa :
Cho phép trích rút thông tin từ tập tin Xml và đưa vào một biến (đúng ra là hằng ví nội dung biến này không thể thay đổi được)
Cú pháp :
<xsl:variable name="Ten_bien" select="Biểu thức Xpath" />

Ví dụ : Chương trình tính tổng 2 số nguyên có thể thực hiện theo 2 cách

Cách 1 : Trích rút thông tin trực tiếp

```
<?xmlversion="1.0"encoding="UTF-8" ?>
<xsl:stylesheetversion="1.0"xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:outputmethod ="text"/>
 <xsl:template match="/" >
   <xsl:value-of select="/GOC/SO[1]/@Gia_tri"/> +
   <xsl:value-of select="/GOC/SO[2]/@Gia_tri"/> =
   <xsl:value-of select="/GOC/SO[1]/@Gia_tri + /GOC/SO[2]/@Gia_tri" />
 </xsl:template>
 </xsl:template>
</xsl:stylesheet>
```

Cách 2 : Thông qua các biến

```
<?xmlversion="1.0"encoding="UTF-8" ?>
<xsl:stylesheetversion="1.0"xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:outputmethod ="text"/>

 <xsl:templatematch="/" >
   <xsl:variablename="So_1"select="/GOC/SO[1]/@Gia_tri" />
   <xsl:variablename="So_2"select="/GOC/SO[2]/@Gia_tri" />

   <xsl:value-ofselect="$So_1" /> +
   <xsl:value-ofselect="$So_2" /> =
   <xsl:value-ofselect="$So_1 + $So_2"/>

 </xsl:template>
```

```
</xsl:stylesheet>
```
Lưu ý :

- Chỉ số các thẻ của tập tin Xml bắt đầu từ 1

- Biểu thức bên trong thuộc tính select có thể
  + Một biểu thức Xpath duy nhất
  + Một biến duy nhất
  + Biểu thức số học với thành phần là biểu thức Xpath hay biến

Điều này cho phép thực hiện một số xử lý trên thông tin nguồn trước khi kết xuất, tuy nhiên các xử lý này khá giới hạn ví Xslt được thiết kế và sử dụng vào mục tiếu chính là biến đổi

2. Rẻ nhánh

Vấn đề :
Cần rè nhánh xử lý kết xuất tùy vào điều kiện của tập tin Xml nguồn

Hướng giải quyết :

Cách 1 : Sử dụng thẻ xử lý xsl:if . Cách này cho phép chỉ kết xuất trong trường hợp một điều kiện nào đó được thỏa (và nếu không thỏa thì không kết xuất gì cả)
Cách 2 : Sử dụng thẻ xử lý xsl:choose . Cách này cho phép kết xuất tùy theo nhiều điều kiện với các trường hợp khác nhau

Thẻ xsl:if

Ý nghĩa :
Cho phép chỉ thực hiện một số thẻ xử lý khi điều kiện được thỏa
Cú pháp :

```
<xsl:if  test="Biểu thức logic "  >
  Các thẻ xử lý
</xsl:if>
```

Ghi chú : Biểu thức logic bao gồm các biểu thức tính toán (trên chuỗi Xpath hay giá trị biến) cùng với các phép toán quan hệ >, >= , <,<= , = , != và các phép toán logic not , and , or

Thẻ xsl:choose

Ý nghĩa :
Tương tự như thẻ xsl:if nhưng cho phép sử dụng nhiều điều kiện khác nhau

Cú pháp :

```
<xsl:choose>
  <xsl:when test="Biểu thức logic 1 " >
     Các thẻ xử lý khi biểu thức logic 1 thỏa
  </xsl:when>
  <xsl:when test="Biểu thức logic 2 " >
     Các thẻ xử lý khi biểu thức logic 2 thỏa
  </xsl:when>
```

```
      <xsl:otherwise >
          Các thẻ xử lý khi tất cả các biểu thức logic trên đều không thỏa
       </xsl:when>
     </xsl:choose>
```

Ghi chú :

Thẻ xử lý trên có tác dụng tương tự cấu trúc

```
 if ( Diều kiện 1 )
    {  ....
    }
 else if ( Điều kiện 2 )
   { ...     }
  .....
  else
    { ....
    }
```

Ví dụ :

Chương trình xác định số nguyên lớn nhất có thể thực hiện theo 2 cách

Cách 1 : Sử dụng xsl:if

```
<?xmlversion="1.0"encoding="UTF-8" ?>
<xsl:stylesheetversion="1.0"xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:outputmethod ="text"     />
 <xsl:template match="/" >
   <xsl:variable name="So_1" select="/GOC/SO[1]/@Gia_tri" />
   <xsl:variable name="So_2" select="/GOC/SO[2]/@Gia_tri" />
   Số lớn nhất trong 2 số
   <xsl:value-of select="$So_1"/> và
   <xsl:value-of select="$So_2"/> là
   <xsl:if test="$So_1> $So_2" >
     <xsl:value-of select ="$So_1"/>
   </xsl:if>
   <xsl:if test="$So_1 &lt;=$So_2" >
      <xsl:value-of select ="$So_2"/>
   </xsl:if>

</xsl:stylesheet>
```

Cách 2 : Sử dụng xsl:choose

```
<?xmlversion="1.0"encoding="UTF-8" ?>
<xsl:stylesheetversion="1.0"xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:outputmethod ="text"      />

 <xsl:templatematch="/" >
   <xsl:variablename="So_1"select="/GOC/SO[1]/@Gia_tri" />
   <xsl:variablename="So_2"select="/GOC/SO[2]/@Gia_tri" />
   Số lớn nhất trong 2 số
```

```
    <xsl:value-of select="$So_1"/> và
    <xsl:value-of select="$So_2"/> là
    <xsl:choose>
      <xsl:whentest="$So_1 > $So_2">
        <xsl:value-ofselect ="$So_1"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-ofselect ="$So_2"/>
      </xsl:otherwise>
    </xsl:choose>

 </xsl:template>
</xsl:stylesheet>
```

3. Vòng lặp

Vấn đề :
Cần lặp lại các xử lý kết xuất trên một tập họp các nút của tãi liệu Xml nguồn

Hướng giải quyết :

Sử dụng thẻ xsl:for-each với biểu thức Xpath tương ứng tập họp nút cần lặp

Thẻ xsl:for-each

Ý nghĩa :
Cho phép lặp lại việc thức hiện các thẻ xử lý trên tập họp các nút là kết quả của một chuỗi truy vấn Xpath

Cú pháp :

```
<xsl:for-each   select="Biểu thức Xpath "  >
  Các thẻ xử lý
</xsl:for-each>
```

Ghi chú : Các thẻ xử lý bên trong vòng lặp có thể sử dụng biểu thức Xpath theo cách định vị tương đối từ nút ngữ cảnh (nút hiện hành) thay cho sử dụng định vị tuyệt đối

Ví dụ :
Với tập tin nguồn Cong_ty.xml :

```
<?xml version="1.0" encoding="utf-8" ?>

<CONG_TY Ten="Công ty X">
 <DON_VI Ten="Đơn vị A" />
 <DON_VI Ten="Đơn vị B" />
 <DON_VI Ten="Đơn vị C" />
 <DON_VI Ten="Đơn vị D" />
</CONG_TY>
```

Chương trình Xuat_cong_ty.xslt sau sẽ kết xuất thông tin về công ty cùng với các đơn vị (theo dạng kết xuất Html)
Công ty X

Danh sách các đơn vị
Đơn vị A
Đơn vị B
Đơn vị C
Đơn vị D

```
<?xmlversion="1.0"encoding="UTF-8" ?>
<xsl:stylesheetversion="1.0"xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:outputmethod ="html" />
 <xsl:template match="/" >
    <xsl:value-of select="/CONG_TY/@Ten"/>
    <br/>
    Danh sách các đơn vị <br />
    <xsl:for-each select="/CONG_TY/DON_VI" >
      <xsl:value-of select="@Ten"/>
      <br />
    </xsl:for-each>
 </xsl:template>


</xsl:stylesheet>
```

4. Hàm

Vấn đề :

Với tập tin Xml có cấu trúc phức tạp hay xử lý kết xuất phức tạp. Việc tổ chức chương trình Xslt chỉ với một thẻ xsl:template duy nhật (tương tự hàm Main duy nhất trong C#)

=== > Chương trình khó viết
=== > Chương trình khó đọc
====> Chương trình khó bảo trì
====> Các đoạn lệnh không tái sử dụng được

Hướng giải quyết :

Tổ chức chương trình thành các phần nhỏ với thẻ xử lý xsl:template (tương tự các hàm tự định nghĩa. Mỗi phần như thế có tên gọi là tập mẩu và đóng vai trò tương tự như hàm trong ngôn ngữ lập trình khác

Thẻ xsl:template

Ý nghĩa :

Cho phép tổ chức chương trình Xslt với các thành phần nhỏ
== > Dễ viết
==> Dễ đọc
== > Dễ bảo trì
===> Tái sử dụng

Cú pháp khai báo

```
<xsl:template match="Biểu thức Xpath">
  Các thẻ xử lý
```

</xsl:template>

Cú pháp "gọi thực hiện"

<xsl:apply-templates select="Biểu thức Xpath" />

Cơ chế "gọi thực hiện" (cơ chế so khớp)

Quá trình "gọi thực hiện" (so khớp) của thẻ xử lý xsl:apply-templates như sau

Bước 1 : Lượng giá biểu thức Xpath của thẻ xử lý xsl:apply-templates

Bước 2 : Tìm khai báo xsl:template có thuộc tính match so khớp đúng

Bước 3 : "Gọi thực hiện " nhiều lần các thẻ xử lý bên trong, mỗi lần với một nút ngữ cảnh thuộc danh sách ước lượng của bước 1

Ví dụ 1 : Chương trình xuất thông tin về công ty có thể viết lại như sau

```
<?xmlversion="1.0"encoding="UTF-8" ?>
<xsl:stylesheetversion="1.0"xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:outputmethod ="html" />


 <xsl:templatematch="/" >
   <xsl:value-ofselect="/CONG_TY/@Ten"/>
   <br/>
   Danh sách các đơn vị <br />
   <xsl:apply-templatesselect="/CONG_TY/DON_VI" />
 </xsl:template>

 <xsl:templatematch="DON_VI" >
   <xsl:value-ofselect="@Ten"/>
   <br />
 </xsl:template>
</xsl:stylesheet>
```

Ví dụ 2 :

Với tập tin Truong.xml có nội dung như sau

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

<TRUONGTen="Trướng cấp 3 XXX">
 <KHOITen="Khối 10" >
   <LOPTen="Lớp 10A" />
   <LOPTen="Lớp 10B" />
   <LOPTen="Lớp 10C" />
   <LOPTen="Lớp 10D" />
 </KHOI>
 <KHOITen="Khối 11" >
   <LOPTen="Lớp 11A" />
   <LOPTen="Lớp 11B" />
   <LOPTen="Lớp 11C" />
 </KHOI>
 <KHOITen="Khối 12" >
   <LOPTen="Lớp 12A" />
   <LOPTen="Lớp 12B" />
   <LOPTen="Lớp 12C" />
```

```
 </KHOI>
</TRUONG>
```

Chương trình Xuat_truong.xslt sau sẽ kết xuất (dạng Html) các thông tin về trường (bao gồm thông tin khối, lớp)

```
<?xmlversion="1.0"encoding="UTF-8" ?>
<xsl:stylesheetversion="1.0"xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:outputmethod ="html" />

 <xsl:template match="/" >
   <xsl:apply-templates select="TRUONG" />
 </xsl:template>

 <xsl:template match="TRUONG">
   <xsl:value-of select="@Ten"/>
   <br />
   Danh sách các khối lớp <br />
   <xsl:apply-templates select="KHOI" />
 </xsl:template>
 <xsl:template match="KHOI">
   <xsl:value-of select="@Ten"/>
   <br />
   <xsl:apply-templates select="LOP" />
 </xsl:template>
  <xsl:template match="LOP">
    <xsl:value-of select="@Ten"/>
    <br />
 </xsl:template>

</xsl:stylesheet>
```

Ghi chú :

Thuộc tính select trong thẻ xsl:apply-templates có thể được bỏ qua và khi đó sẽ được hiểu là select="*" (cho lượng giá là các nút con của nút ngữ cảnh)

=== > Một trong các cách đơn giãn tổ chức chương trình Xslt là tổ chức chương trình theo các loại thẻ có trong tập tin Xml và gọi thực hiện (so khớp) không cần tham số

Gọi thực hiện : <xsl:apply-templates />

Khai báo hàm/mẩu so khớp :

```
    <xsl:template match="tên loại thẻ" >
       Các thẻ xử lý
    </xsl:template>
<xsl:stylesheetversion="1.0"xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:outputmethod ="html" />


 <xsl:templatematch="/" >
   <xsl:apply-templates   />
 </xsl:template>
```

```
<xsl:templatematch="TRUONG">
  <xsl:value-ofselect="@Ten"/>
  <br />
  Danh sách các khối lớp <br />
  <xsl:apply-templates  />
</xsl:template>
<xsl:templatematch="KHOI">
  <xsl:value-ofselect="@Ten"/>
  <br />
  <xsl:apply-templates  />
</xsl:template>

<xsl:templatematch="LOP">
  <xsl:value-ofselect="@Ten"/>
  <br />
</xsl:template>

</xsl:stylesheet>
```

III. Một số kỹ thuật xử lý

Một số kỹ thuật xử lý
Mục tiêu :
Trình bày 2 kỹ thuật cơ bản tương ứng 2 ứng dụng chính của XML
- Biến đổi XML ---- > HTML
- Biến đổi XML ---- > XML

1. XML -- > HTML

Mục tiêu :
Trình bày một số kỹ thuật cơ bản cho phép thể hiện nội dung tập tin Xml trên trang Web
Ví dụ :
với tập tin Xml Don_thuc.xml

```
<DON_THUC  He_so="4"  So_mu="6" />
```

chương trình Xslt sau sẽ cho phép thể hiện đơn thức dưới dạng trình bày trên Web

4x6

Chương trình Xuat_don_thuc.xslt

```
<?xmlversion="1.0"encoding="UTF-8" ?>
<xsl:stylesheetversion="1.0"xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:outputmethod ="html" />
<xsl:templatematch="/" >
  <html>
    <body>
      <xsl:apply-templates />
    </body>
```

```
    </html>
 </xsl:template>
 <xsl:templatematch="DON_THUC" >
    P(x)=
    <xsl:value-ofselect ="@He_so" /> x
    <sup>
      <xsl:value-ofselect ="@So_mu"/>
    </sup>
 </xsl:template>
</xsl:stylesheet>
```

*** XML -- > Thẻ select**

Vấn đề :

Cần xuất danh sách chọn trên trang Web từ một danh sách các nút của tập tin Xml trong một ứng dụng Web

Ví dụ :

Xuất danh sách các mặt hàng
Xuất danh sách các đơn vị
Xuất danh sách các khối
....

Hướng giải quyết :

Sử dụng thẻ select , option của ngôn ngữ Html

```
....
<select>
  <xsl:apply-templates select="Biểu thức Xpath tương ứng danh sách" />
</select>
....

<xsl:template match="Biểu thức Xpath tương ứng một phần tử trong danh sách" >
   <option>
      Thẻ xử lý kết xuất giá trị
   </option>
</xsl:template>
```

Ví dụ :

Chương trình Xslt sau sẽ xuất danh sách chọn các đơn vị từ tập tin Cong_ty.xml

```
<?xmlversion="1.0"encoding="UTF-8" ?>
<xsl:stylesheetversion="1.0"xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:outputmethod ="html" />
 <xsl:templatematch="/" >
    <html>
      <body>
        <xsl:apply-templates />
      </body>
    </html>
 </xsl:template>
 <xsl:templatematch="CONG_TY" >
    Danh sách đơn vị :
```

```
    <select>
      <xsl:apply-templates />
    </select>
 </xsl:template>
 <xsl:templatematch="DON_VI" >
    <option>
      <xsl:value-ofselect ="@Ten"/>
    </option>
 </xsl:template>
</xsl:stylesheet>
```

*** XML --- > thẻ ul,ol**

Vấn đề :

Cần xuất danh sách trình bày nội dung và các mục trên trên trang Web từ một nút với nhiều nút con của tập tin Xml trong một ứng dụng Web theo dạng

Nội dung 1

  Mục con 11

  Mục con 12

  ....

Ví dụ :

  Xuất nội dung câu hỏi trắc nghiệm

  Xuất tổ chức trường -khối - lớp theo dạng cây

    ....

Hướng giải quyết :

  Sử dụng thẻ ul kết hợp li với các mục không có thứ tự

  Hay sử dụng thẻ ol kết hợp li với các mục có thứ tự

```
  ....
  <ul>
    <xsl:apply-templates select="Biểu thức Xpath tương ứng danh sách" />
  </ul>
   ....
```

hay

```
  ....
  <ol>
    <xsl:apply-templates select="Biểu thức Xpath tương ứng danh sách" />
  </ol>
   ....
```

và

```
   <xsl:template match="Biểu thức Xpath tương ứng một phần tử trong danh sách" >
     <li>
        Thẻ xử lý kết xuất giá trị
      </li>
   </xsl:template>
```

Ví dụ 1:

  Giả sử đã có tập tin Cau_hoi.xml với nội dung như sau

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

```
<CAU_HOINoi_dung="Nước Việt Nam hình chử gì ?">
 <CHON_LUANoi_dung="Chử M" />
 <CHON_LUANoi_dung="Chử S" />
 <CHON_LUANoi_dung="Chử K" />
 <CHON_LUANoi_dung="Chử R" />
</CAU_HOI>
```

Chương trình Xslt sau sẽ trình bày câu hỏi trên Web

Câu hỏi trắc nghiệm
Nước Việt Nam hình chử gì ?
Chử M
Chử S
Chử K
Chử R

```
<?xmlversion="1.0"encoding="UTF-8" ?>
<xsl:stylesheetversion="1.0"xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:outputmethod ="html" />
 <xsl:templatematch="/" >
   <html>
     <body>
       Câu hỏi trắc nghiệm <br />
       <xsl:apply-templates />
     </body>
   </html>
 </xsl:template>
 <xsl:templatematch="CAU_HOI" >
   <ol >
     <xsl:value-ofselect ="@Noi_dung" />
     <xsl:apply-templates />
   </ol>
 </xsl:template>
 <xsl:templatematch="CHON_LUA" >
   <li>
     <xsl:value-ofselect ="@Noi_dung" />
   </li>
 </xsl:template>
</xsl:stylesheet>
```

Ví dụ 2 :

Chương trình Xslt sau sẽ trình bày tổ chức các khối lớp của một trường theo dạng cây phân cấp

```
<?xmlversion="1.0"encoding="UTF-8" ?>
<xsl:stylesheetversion="1.0"xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:outputmethod ="html" />
 <xsl:templatematch="/" >
   <html>
     <body>
       <xsl:apply-templates />
```

```
    </body>
  </html>
</xsl:template>
<xsl:templatematch="TRUONG" >
  <ul >
    <xsl:value-ofselect ="@Ten" />
    <xsl:apply-templates />
  </ul>
</xsl:template>
<xsl:templatematch="KHOI" >
  <ul >
    <xsl:value-ofselect ="@Ten" />
    <xsl:apply-templates />
  </ul>
</xsl:template>
<xsl:templatematch="LOP" >
  <ul >
    <xsl:value-ofselect ="@Ten" />
    <xsl:apply-templates />
  </ul>
</xsl:template>
</xsl:stylesheet>
```

*** XML -- > Thẻ Table**

Vấn đề :

Cần xuất danh sách dạng lưới trên trang Web từ một danh sách các nút của tập tin Xml trong một ứng dụng Web

Ví dụ :

Xuất danh sách các mặt hàng : tên , Đơn giá

Xuất danh sách các nhân viên : Họ tên , Ngày sinh , Giới tính

Xuất danh sách các môn học : tên môn , Số tiết LT, Số tiết thực hành

....

Hướng giải quyết :

Sử dụng thẻ table , tr, td của ngôn ngữ Html

```
....
<table>
  <xsl:apply-templates select="Biểu thức Xpath tương ứng danh sách" />
</table>
....

<xsl:template match="Biểu thức Xpath tương ứng một phần tử trong danh sách" >
  <tr>
    <td>
      Thẻ xử lý kết xuất giá trị tại cột thứ 1
    </td>
    <td>
      Thẻ xử lý kết xuất giá trị tại cột thứ 12
    </td>
```

```
        .....

      </tr>
   </xsl:template>
```

Ví dụ :

Chương trình Xslt sau sẽ xuất bảng đơn giá thuê phòng từ tập tin Bang_don_gia.xml

```
<?xmlversion="1.0"encoding="UTF-8" ?>
<xsl:stylesheetversion="1.0"xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:outputmethod ="html" />
 <xsl:templatematch="/" >
   <html>
     <body>
       <divalign="center">
         Bảng đơn giá thuê phòng <br />
         <xsl:apply-templates />
       </div>
     </body>
   </html>
 </xsl:template>
 <xsl:templatematch="BANG_DON_GIA" >
   <tableborder="2">
     <tr>
       <td>
         Loại phòng
       </td>
       <td>
         Đơn giá
       </td>
     </tr>
     <xsl:apply-templates />
   </table>
 </xsl:template>
 <xsl:templatematch="LOAI_PHONG" >
   <tr>
     <td>
       <xsl:value-ofselect ="@Ten"/>
     </td>
     <td>
       <xsl:value-ofselect ="@Don_gia"/>
     </td>
   </tr>
 </xsl:template>
</xsl:stylesheet>
```

2. XML --- > XML

Mục tiêu :

Trình bày một số kỹ thuật cơ bản cho phép tạo tài liệu Xml mới dựa trên một tài liệu Xml đã có

- Trích rút thông tin
- Tái cấu trúc

*** Tạo nút và thuộc tính**

Vấn đề :
Cần tạo thẻ mới X cùng với các thuộc tính trong tập tin xml kết xuất

Hướng giải quyết :

Cách 1 : Tạo lập trực tiếp thẻ mới X trong chương trình Xslt (tương tự như sao5n thảo tập tin XML
Cách 2 : Sử dụng các thẻ xử lý xsl:element , xsl:attribute

Thẻ xsl:element

Ý nghĩa :
Cho phép tạo thẻ mới trong tập tin Xml kết xuất
Cú pháp :

```
<xsl:element name="Ten_the" >
        Các thẻ xử lý tạo thuộc tính ( nếu có )
        Các thẻ khác
</xsl:element>
```

Thẻ xsl:attribute

Ý nghĩa :
Cho phép tạo thuộc tính của một thẻ trong tập tin Xml kết xuất
Cú pháp :

```
<xsl:element name="Ten_the" >
        <xsl:attribute name="ten_thuoc_tinh" >
              Thẻ xử lý kết xuất giá trị của thuộc tính
         </xsl:attribute>
        Các thẻ khác
</xsl:element>
```

Vi dụ :
Chương trình Xslt sau đây biến đổi tập tin Phieu_thu.xml với tập tin Xml kết xuất có các nút con tương ứng các thuộc tính

```
<?xmlversion="1.0"encoding="utf-8" ?>
<PHIEU_THUKhach_hang="Trần văn Long"
        Ngay_thu="11/2/2007"
        So_tien="40000" >

</PHIEU_THU>
```

Chương trình Xslt

```
<?xmlversion="1.0"encoding="UTF-8" ?>
<xsl:stylesheetversion="1.0"xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:outputmethod ="xml"indent="yes" />
 <xsl:templatematch="/" >
```

```
   <xsl:apply-templates />
 </xsl:template>

 <xsl:templatematch="PHIEU_THU" >
   <xsl:variablename="Ngay_thu"select ="@Ngay_thu" />
   <xsl:variablename="So_tien"select ="@So_tien" />
   <xsl:variablename ="Ten"select ="@Khach_hang" />
   <PHIEU_THUNgay_thu="{$Ngay_thu}"
           So_tien="{$So_tien}">
     <KHACH_HANGTen="{$Ten}" />
   </PHIEU_THU>
 </xsl:template>

</xsl:stylesheet>
```

*** Sao chép nút**

Vấn đề :
Cần tạo thẻ kết xuất trong tập tin xml kết xuất có cùng tên và các thuộc tính với thẻ trong tập tin nguồn

Hướng giải quyết :
Cách 1 : Sử dụng các thẻ xử lý xsl:element , xsl:attribute
Cách 2 : Sử dụng các thẻ xử lý xsl:copy , xsl:attribute

Thẻ xsl:copy

Ý nghĩa :
Cho phép sao chép thẻ từ tập tin xml nguồn (với nút ngữ cảnh tương ứng thẻ) sang tập tin xml kết xuất

Cú pháp :

```
<xsl:copy >
      Các thẻ xử lý tạo thuộc tính ( nếu có )
      Các thẻ khác
</xsl:copy>
```

Kết hợp với xsl:attribute để sao chép thẻ - thuộc tính

```
<xsl:copy >
    <xsl:for-each select="@*" >
        <xsl:attribute name="{name()}" >
            <xsl:value-of select="." />
        </xsl:attribute>
    </xsl:for-each>
  </xsl:copy>
```

Ví dụ 1:

Chương trình sau trích danh sách các khối của tập tin truong.xml

```
<?xmlversion="1.0"encoding="UTF-8" ?>
<xsl:stylesheetversion="1.0"xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:outputmethod ="xml"indent="yes" />
```

```
<xsl:templatematch="/" >
      <xsl:apply-templates />
</xsl:template>
<xsl:templatematch="TRUONG" >
  <xsl:copy >
    <xsl:attributename="Ten" >
      <xsl:value-ofselect="@Ten"/>
    </xsl:attribute>
    <xsl:apply-templates />
  </xsl:copy>
</xsl:template>
<xsl:templatematch="KHOI" >
  <xsl:copy >
    <xsl:attributename="Ten" >
      <xsl:value-ofselect="@Ten"/>
    </xsl:attribute>
  </xsl:copy>

</xsl:template>

</xsl:stylesheet>
```

Vi dụ 2 :

Chương trình Xslt sau cho phép biến đổi tập tin Xml bất kỳ theo qui tắc : Tất cả thuộc tính sẽ biến thành thẻ con

```
<?xmlversion="1.0"encoding="UTF-8" ?>
<xsl:stylesheetversion="1.0"xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:outputmethod ="xml"indent="yes" />
 <xsl:templatematch="/" >
   <xsl:apply-templates />
 </xsl:template>
 <xsl:templatematch="*" >
   <xsl:copy >
     <xsl:for-eachselect="@*" >
       <xsl:elementname="{name()}" >
         <xsl:value-ofselect="."/>
       </xsl:element>
     </xsl:for-each>
     <xsl:apply-templates />
   </xsl:copy>
 </xsl:template>
</xsl:stylesheet>
```

*** Sao chép nút - thuộc tính - nút con**

Vấn đề :

Cần sao chép toàn bộ thẻ X , tất cả thuộc tính của X, tất các thẻ con mọi cấp của X trong tập tin xml nguồn vào tập tin Xml kết xuất

Hướng giải quyết :

Cách 1 : Sử dụng các thẻ xử lý xsl:copy , xsl:attribute

Cách 2 : Sử dụng thẻ xử lý xsl:copy-of

Thẻ xsl:copy-of

Ý nghĩa :

Cho phép sao chép toàn bộ thẻ X , tất cả thuộc tính của X, tất các thẻ con mọi cấp của X trong tập tin xml nguồn vào tập tin Xml kết xuất

Cú pháp :

<xsl:copy-of select="Biểu thức Xpath" />

Vi dụ :

Cho tập tin xml Bang_phan_cong.xml

```
<?xmlversion="1.0"encoding="utf-8" ?>
<TRUONGTen="Trường X" >
 <MONTen="Toán">
   <GIAO_VIENTen="Trần văn Minh" >
     <LOPTen="10A1" />
     <LOPTen="11A1" />
     <LOPTen="12A1"   />
   </GIAO_VIEN>
   <GIAO_VIENTen="Trần văn Lộc" >
     <LOPTen="10B1"   />
     <LOPTen="11B1"   />
     <LOPTen="12B1"   />
   </GIAO_VIEN>
   <GIAO_VIENTen="Trần văn Hùng" >
     <LOPTen="10B2"   />
     <LOPTen="11A3"   />
     <LOPTen="12B2"   />
   </GIAO_VIEN>
 </MON>
 <MONTen="Lý">
   <GIAO_VIENTen="Lê văn Sơn" >
     <LOPTen="10A1"   />
     <LOPTen="11B1"   />
     <LOPTen="12C1"   />
   </GIAO_VIEN>
   <GIAO_VIENTen="Nguyễn thị bé Nhỏ" >
     <LOPTen="10B1"   />
     <LOPTen="11B2"   />
     <LOPTen="12B1"   />
   </GIAO_VIEN>
 </MON>
 <MONTen="Hóa">
   <GIAO_VIENTen="Ngô thị Bích" >
     <LOPTen="10B1"   />
     <LOPTen="11B1"   />
     <LOPTen="12C1"   />
   </GIAO_VIEN>
   <GIAO_VIENTen="Lê văn Lớn" >
```

```
    <LOPTen="10A1"   />
    <LOPTen="12B1"   />
   </GIAO_VIEN>
   <GIAO_VIENTen="Đổ thị Tuyết" >
    <LOPTen="10B2"   />
    <LOPTen="11A3"   />
    <LOPTen="12B2" />
   </GIAO_VIEN>
 </MON>
 <MONTen="Sinh">
   <GIAO_VIENTen="Nguyễn hùng Cường">
    <LOPTen="10A1"   />
    <LOPTen="11A1"   />
    <LOPTen="12A1"   />
   </GIAO_VIEN>
   <GIAO_VIENTen="Lê văn Tùng" >
    <LOPTen="11A2"   />
    <LOPTen="12B1"   />
   </GIAO_VIEN>
   <GIAO_VIENTen="Nguyễn thị Đẹp" >
    <LOPTen="11B2"   />
    <LOPTen="11A3"   />
    <LOPTen="11B3" />
   </GIAO_VIEN>
 </MON>
</TRUONG>
```

Đoạn chương trình sau cho phép tái cấu trúc tập tin Bang_phan_cong.xml với thẻ MON chuyển thành thuộc tính Bo_mon của thẻ GIAO_VIEN

```
<?xmlversion="1.0"encoding="UTF-8" ?>
<xsl:stylesheetversion="1.0"xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:outputmethod ="xml"indent="yes" />
 <xsl:templatematch="/" >
   <xsl:apply-templates />
 </xsl:template>
 <xsl:templatematch="TRUONG" >
   <xsl:copy >
    <xsl:attributename="Ten" >
     <xsl:value-ofselect="@Ten"/>
    </xsl:attribute>
    <xsl:apply-templates />
   </xsl:copy>
 </xsl:template>
 <xsl:templatematch="GIAO_VIEN" >
   <xsl:copy >
    <xsl:attributename="Ten" >
     <xsl:value-ofselect="@Ten"/>
```

```
    </xsl:attribute>
    <xsl:attributename="Bo_mon" >
      <xsl:value-ofselect="../@Ten"/>
    </xsl:attribute>
    <xsl:copy-ofselect="LOP"/>
  </xsl:copy>
</xsl:template>

</xsl:stylesheet>
```

*** Sắp thứ tự các nút**

Vấn đề :
Cần sắp thứ tự danh sách các thẻ X của tập tin xml kết xuất

Hướng giải quyết :
Sử dụng thẻ xử lý xsl:sort kết hợp với xsl:apply-templates

Thẻ xsl:sort

Ý nghĩa :
Cho phép sắp thứ tự danh sách các thẻ X của tập tin xml kết xuất
Cú pháp :
Sắp tăng

```
<xsl:sort order="accending" select="Thuộc tính" />
```

Sắp giảm

```
<xsl:sort order="descending" select="Thuộc tính" />
```

Kết hợp với xsl:apply-templates để tiến hành sắp thứ tự các kết quả sau khi thực hiện so khớp các hàm/mẩu

```
<xsl:apply-templates select="Biểu thức Xpath" >
  <xsl:sort order="...." select="...." />
  <xslsort order="...." select="...." />
   .....
</xsl:aplly-templates>
```

Ví dụ : Với tập tin Xml Ket_qua_Olympic.xml

```
<?xmlversion="1.0"encoding="utf-8" ?>
<KET_QUA>
 <QUOC_GIATen="AAA"So_vang="10"So_bac="7"So_dong="2" />
 <QUOC_GIATen="XXX"So_vang="6"So_bac="0"So_dong="12" />
 <QUOC_GIATen="BBB"So_vang="10"So_bac="8"So_dong="13" />
 <QUOC_GIATen="DDD"So_vang="4"So_bac="17"So_dong="0" />
 <QUOC_GIATen="MMM"So_vang="6"So_bac="1"So_dong="0" />
 <QUOC_GIATen="KKK"So_vang="6"So_bac="0"So_dong="2" />
 <QUOC_GIATen="LLL"So_vang="10"So_bac="4"So_dong="23" />
 <QUOC_GIATen="PPP"So_vang="3"So_bac="27"So_dong="100" />
</KET_QUA>
```

Đoạn chương trình XSL sau sắp xếp các quốc gia giảm dần theo thứ tự ưu tiên
- Ưu tiên 1 : Số huy chương vàng
- Ưu tiên 2 : Số huy chương bạc
- Ưu tiên 3 : Số huy chương đồng

```
<?xmlversion="1.0"encoding="UTF-8" ?>
<xsl:stylesheetversion="1.0"xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:outputmethod ="xml"indent="yes" />
 <xsl:templatematch="/" >
   <xsl:apply-templates />
 </xsl:template>
 <xsl:templatematch="KET_QUA"   >
   <xsl:copy >
     <xsl:apply-templates   select ="QUOC_GIA">
       <xsl:sort    order="descending"data-type="number"select="@So_vang" />
       <xsl:sort    order="descending"data-type="number"select="@So_bac" />
       <xsl:sort    order="descending"data-type="number"select="@So_dong" />

     </xsl:apply-templates>
   </xsl:copy>
 </xsl:template>
 <xsl:templatematch="QUOC_GIA" >
   <!--<xsl:copy-of select="."/>-->
   <xsl:copy >
     <xsl:copy-ofselect ="@*"/>
   </xsl:copy>

 </xsl:template>

</xsl:stylesheet>
```

IV. Bài tập

1. XML --- > HTML

* Tích 2 phân số

Yêu cầu :

Viết chương trình XSLT cho phép tạo kết xuất từ dữ liệu nguồn
- Dữ liệu nguồn : Tập tin xml biểu diễn thông tin về 2 phân số
- Kết xuất : Trang Web thể hiện kết quả nhân 2 phân số

Ví dụ : vời phân số 4/7, 5/11
Kết xuất sẽ là
Kết quả tính tích 2 phân số 1/7 và 5/11
4/7*5/11=20/77

*** Phương trình đường thẳng**

Yêu cầu :

Viết chương trình XSLT cho phép tạo kết xuất từ dữ liệu nguồn
- Dữ liệu nguồn : Tập tin xml biểu diễn thông tin về các hệ số của phương trình đường thẳng trong mặt phẳng
- Kết xuất : Trang Web thể hiện kết quả là phương trình đường thẳng

Ví dụ :
Vời giá trị các hệ số 2,3,4
Kết xuất sẽ là : Phương trình đường thẳng 2x + 3y + 4=0

Vời giá trị các hệ số 7,-3
Kết xuất sẽ là : Phương trình đường thẳng 7x - 3y =0

*** Phương trình đường tròn**

Yêu cầu :

Viết chương trình XSLT cho phép tạo kết xuất từ dữ liệu nguồn
- Dữ liệu nguồn : Tập tin xml biểu diễn thông tin về tọa độ tâm và bán kính của một đường tròn
- Kết xuất : Trang Web thể hiện kết quả là phương trình đường tròn

Ví dụ :
Vời giá trị các tọa độ tâm 4,5 và bán kính 3
Kết xuất sẽ là : Phương trình đường tròn (x-4)2 + (y-5) 2 = 9

Vời giá trị các tọa độ tâm -2,3 và bán kính 7
Kết xuất sẽ là : Phương trình đường tròn (x + 2)2 + (y-3) 2 = 49

*** Đề trắc nghiệm**

Yêu cầu :

Viết chương trình XSLT cho phép tạo kết xuất từ dữ liệu nguồn
- Dữ liệu nguồn : Tập tin xml biểu diễn thông tin về đề trắc nghiệm
- Kết xuất : Trang Web thể hiện đề trắc nghiệm

*** Hồ sơ học sinh**

Yêu cầu :

Viết chương trình XSLT cho phép tạo kết xuất từ dữ liệu nguồn
- Dữ liệu nguồn : Tập tin xml biểu diễn thông tin về hồ sơ học sinh bao gồm : Họ và tên , giới tính, ngày sinh , địa chỉ
- Kết xuất :

a) Trang Web thể hiện hồ sơ học sinh dạng xem
b) Trang Web thể hiện hồ sơ học sinh dạng nhập liệu (cập nhật)

*** Bảng xếp hạng Olympic**

Yêu cầu :

Viết chương trình XSLT cho phép tạo kết xuất từ dữ liệu nguồn
- Dữ liệu nguồn : Tập tin xml biểu diễn thông tin kết quả thi đấu Olympic các quốc gia
- Kết xuất :
a) Trang Web thể hiện bảng kết quả thi đấu
b) Trang Web cho phép cập nhật số huy chương vàng, bạc, đồng

2. XML - XML

*** Hồ sơ nhân viên**

Yêu cầu :

Viết chương trình XSLT cho phép tạo kết xuất từ dữ liệu nguồn
- Dữ liệu nguồn : Tập tin xml biểu diễn thông tin về hồ sơ nhân viên với
  + Thông tin bao gồm : Họ và tên, Giới tính , Ngày sinh, Địa chỉ , Đơn vị
  + Tất cả các thông tin đều biểu diễn dưới dạng thẻ con
- Kết xuất : Tập tin Xml
a) Tất cả các thông tin đều biểu diễn dạng thuộc tính
b) Tất cả các thông ngoại trừ đơn vị đều biểu diễn dạng thuộc tính

*** Trường - khối - lớp**

Yêu cầu :

Viết chương trình XSLT cho phép tạo kết xuất từ dữ liệu nguồn
- Dữ liệu nguồn : Tập tin xml biểu diễn thông tin tổ chức trường , các khối của trường, các lớp của khối
- Kết xuất :
a) Tập tin Xml chỉ bao gồm các lớp có sỉ số trên 30
b) Tập tin Xml chỉ bao gồm các khối có hơn 5 lớp

*** Bảng phân công giáo viên**

Yêu cầu :

Viết chương trình XSLT cho phép tạo kết xuất từ dữ liệu nguồn
- Dữ liệu nguồn : Tập tin xml biểu diễn thông tin bảng phân công các giáo viên của một trường
- Kết xuất :
a) Tập tin Xml chỉ bao gồm danh sách các bộ môn cùng với số lượng các giáo viên
b) Tập tin Xml chỉ bao gồm danh sách các giáo viên được phân công dạy trên 2 lớp

*** Cây số nguyên**

Yêu cầu :

Viết chương trình XSLT cho phép tạo kết xuất từ dữ liệu nguồn
- Dữ liệu nguồn : Tập tin xml biểu diễn thông tin cây các số nguyên
- Kết xuất :
a) Tập tin Xml chỉ bao gồm danh sách các nút lá
b) Tập tin Xml chỉ bao gồm danh sách các nút có giá trị từ 1 đến 10
c) Tập tin Xml là tập tin nguồn sau khi bỏ đi các nút lá
d) Tập tin Xml là tập tin nguồn sau khi bỏ đi tất cả các nút con cấp 1 (con của gốc)
e) Tập tin Xml là tập tin nguồn sau khi sắp thứ tự tăng các nút cùng cha

Đồ án

Mục tiêu :
Giúp sinh viên
- Có cơ hội ôn tập, rèn luyện các kỹ thuật liên quan XML
- Có cơ hội tìm hiểu và sử dụng tốt ngôn ngữ lập trình
- Rút các kinh nghiệm khi xây dựng ứng dụng thực tế

Điểm số :
Đồ án là tùy chọn, sinh viên không bắt buộc phải đăng ký.
Với các sinh viên đã đăng ký điểm số của đồ án sẽ được cộng vào điểm thi cuối môn
=== > Điểm thi cuối môn sẽ là 10 + n
với -2<=n <=2
Đăng ký và không nộp bài n =-2
=== > Cần suy nhỉ trước khi đăng ký

Yêu cầu chung :
Viết một chương trình ứng dụng với các chức năng được yêu cầu

Kết quả nộp :
1. Báo cáo viết về hồ sơ thiết kế (tập tin Word)
2. Chương trình nguồn
3 . Chương trình đã biên dịch cùng với các dữ liệu thử nghiệm

Thời hạn :
- Hạn chót đăng ký : Ngày 1/10/2007. Sinh viên đăng ký đồ án nào sẽ chọn mục thảo luận và cho ý kiến về việc đăng ký
- Hạn chót nộp kết quả : Sẽ được thông báo sau (dự kiến trong tuần lễ thi)

Lứu ý :
- 1 sinh viên trên 1 đồ án
- Có thể sẽ có vấn đáp trực tiếp với giáo viên khi nộp đồ án nhằm xác định chính xác người thực hiện

1. Dò mìn

Các yêu cầu chức năng :
1. Đánh cờ (chức năng chính tương ứng việc xử lý việc chọn vị trí của người dùng)
2. Ghi bàn cờ vào tập tin
3. Đọc bàn cờ đã ghi

2. Bán vé tàu hỏa

Các yêu cầu chức năng :
1. Tính tiền bán vé
2. Quản lý (thêm,xóa,sửa) các ga
3. Cập nhật bảng đơn giá

3. Đánh cơ caro

Các yêu cầu chức năng :
1. Đánh cờ (chức năng chính tương ứng việc xử lý việc đi cờ của người dùng)
2. Ghi bàn cờ vào tập tin
3. Đọc bàn cờ đã ghi

4. Đánh cơ tướng

Các yêu cầu chức năng :
1. Đánh cờ (chức năng chính tương ứng việc xử lý việc đi cờ của người dùng)
2. Ghi bàn cờ vào tập tin
3. Đọc bàn cờ đã ghi

5. Trò chơi ô chử

Các yêu cầu chức năng :
1. Đọc và đoán ô chử đả được lưu trữ trên 1 tập tin
2. Soạn thảo và ghi ô chử vào tập tin

6. Trò chơi Sudoku

Các yêu cầu chức năng :
1. Đọc và giải ma trận Sudoku đả được lưu trữ trên 1 tập tin
2. Soạn thảo ma trận Sudoku và ghi vào tập tin

7. Rồng vàng

Các yêu cầu chức năng :
1. Sắp xếp các từ (vòng 1 của trò chơi)
2. Trả lời các câu đố (vòng 2)
2. Soạn thảo câu đố (vòng 1 , 2)

The End