Đặc tả cấu trúc 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 các 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

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.

1.1. Kiểu định nghĩa sẵn

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

1.2. 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

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) 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>
```

Trong đó:

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 khoá minInclusive (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 hợp 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 hợp 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 khoá maxInclusive (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 kê 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>
```

1.3. **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:**

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.

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 , maxOccurs

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"
                     maxOccurs="So_lan_lap_toi_da"  />
        ...
     </xs:sequenc
          e>
        ....
</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**:

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=”G
            ia_tri_k” />
```

Một số tính chất thông dụng như sau

Giá trị định sẳn : 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_DUO
            NG" 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 thẻ 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 thẻ 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

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)

*** 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

*** 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

Họvà 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 tiền

....

.....

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 Sốngày 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

Sửdụng tập tin Xml biểu diễn thông tin các đối tượng trong thực tế

2. Thiết kế xử lý

*** 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

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 kếxử lý

- Dữ liệu đặc tảcấu 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>