

62-31.3 XML

[XML-02-S06]
XSD – Dérivation

Solution

[Cédric Benoit]

Exercice 1 – Création de types simples personnalisés et de types complexes personnalisés dérivés de types existants

Elaborer pour chaque donnée et chaque type de structure ci-dessous un type simple ou un type complexe en utilisant (au mieux) les schémas XML.

Regrouper toutes vos définitions dans un fichier xsd.

Elaborer un fichier XML permettant de tester et de vérifier vos définitions.

1. Dans le club sportif de gymnastique, nous gérer plusieurs catégories de personnes, allant des enfants aux personnes âgées. Au niveau de l'âge nous aimerions pouvoir saisir ceci:

- Babygym: 0 à 2 ans
- Parent-Enfant: 3 à 4 ans
- Enfants: 5 à 8 ans
- Kids: 9 à 12 ans
- Jeunesse: 13 à 17 ans
- Adulte: 18 à 65 ans
- Senior 65 à 100 ans

Au niveau XML, vous avez des balises contenant des valeurs numériques entières comme: `<Age>6</Age>`, `<Age>25</Age>`

En vous basant sur la dérivation de types simples, quelles solutions proposez-vous pour que l'on saisisse le bon âge dans chacune des catégories.

```
<xs:simpleType name="TypeAge">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="125"/>
  </xs:restriction>
</xs:simpleType>
```

```
<xs:simpleType name="TypeAgeBabygym">
  <xs:restriction base="TypeAge">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="2"/>
  </xs:restriction>
</xs:simpleType>
```

```
<xs:simpleType name="TypeAgeParentEnfant">
  <xs:restriction base="TypeAge">
    <xs:minInclusive value="3"/>
    <xs:maxInclusive value="4"/>
  </xs:restriction>
</xs:simpleType>
```

```
<xs:simpleType name="TypeAgeEnfantine">
  <xs:restriction base="TypeAge">
    <xs:minInclusive value="5"/>
    <xs:maxInclusive value="8"/>
  </xs:restriction>
</xs:simpleType>
```

```

    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="TypeAgeKids">
  <xs:restriction base="TypeAge">
    <xs:minInclusive value="9"/>
    <xs:maxInclusive value="12"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="TypeAgeJeunesse">
  <xs:restriction base="TypeAge">
    <xs:minInclusive value="13"/>
    <xs:maxInclusive value="17"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="TypeAgeAdulte">
  <xs:restriction base="TypeAge">
    <xs:minInclusive value="18"/>
    <xs:maxInclusive value="65"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="TypeAgeSenior">
  <xs:restriction base="TypeAge">
    <xs:minInclusive value="65"/>
    <xs:maxInclusive value="100"/>
  </xs:restriction>
</xs:simpleType>

```

2. Nous disposons d'une structure adresse contenant les informations suivantes:

```

<Adresse>
  <Rue numMaison="32">Rue de la Gare</Rue>
  <Npa>2000</Npa>
  <Lieu>Neuchâtel</Lieu>
</Adresse>

```

Pour les adresse suisses, cette structure convient bien. Par contre, si on veut saisir des adresses étrangères, on aimerait pouvoir ajouté le pays comme ceci:

```

<Adresse>
  <Rue numMaison="128">Rue de Franche-Comté</Rue>
  <Npa>25300</Npa>
  <Lieu>Pontarlier</Lieu>
  <Pays>France</Pays>
</Adresse>

```

En utilisant la dérivation, quelles solutions de types complexes proposez-vous pour résoudre ce problème.

```

<xs:complexType name="TypeAdresse">
  <xs:sequence>
    <xs:element name="Rue">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:token">
            <xs:attribute name="numMaison"
type="xs:integer"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="Npa" type="xs:integer"/>
    <xs:element name="Lieu" type="xs:token"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="TypeAdressePays">
  <xs:complexContent>
    <xs:extension base="TypeAdresse">
      <xs:sequence>
        <xs:element name="Pays" type="xs:token"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

Exercice 2 – Adaptation des schémas XSD de l'exercice XML-02-E05

Reprendre les exercices de gestion des clients avec leurs commandes et produits, l'arbre généalogique avec les descendants, ainsi que l'exercice de gestion des voitures dans un garage réalisés dans l'exercice XML-02-E04 avec leurs document XML et schéma XML (XSD). Pour chacun de ces trois exercices, veuillez analyser et vérifier si un ou plusieurs types dérivés (types simples ou types complexes) pourraient être créés dans les schémas XML (XSD) correspondants, pour améliorer la gestion de vos données.

Par chaque document XML, veuillez « jouer » avec les données et structures XML (adresse, commande ...) en ajoutant, modifiant, et supprimant des données, afin de vérifier et de valider, si les types dérivés attribués sont corrects par rapport à ce que vous vouliez pouvoir saisir.

Pour le prochain cours

- Rassembler vos fichiers XML et XSD dans une archive (format zip ou 7z)
- Nommer votre archive selon le modèle : XML-02-E06_classe_nom_prenom
- Mettez l'archive dans le dossier "Devoirs" de Cyberlearn correspondant à votre classe (en cas de problème vous pouvez toujours envoyer l'archive par e-mail à l'adresse e-mail : cedric.benoit@he-arc.ch)
- Date limite : avant le début de la prochaine leçon
- Au début de la prochaine leçon un ou deux étudiants seront pris au hasard et présenteront leur travail en classe.