

# XML валидация чрез XML Schema

## Цели на упражнението:

1. Създаване на XML схема
2. Валидация на XML документ с XML Schema
3. Дефиниране на прости и комплексни типове
4. Задаване на ограничения върху предефинираните типове
5. Използване на разширени типове и предефиниране на типове
6. Използване на анотации, нотации, регулярни изрази
7. Импортиране на XML схеми от други пространства от имена

## Средства за XML валидация чрез XML Schema:

За реализация на това упражнения могат да бъдат използвани някои от следните инструменти:

- ✓ [XML Validator](#)
- ✓ [XML Validator Online](#)
- ✓ [CoreFiling XML Schema Validator](#)

**Задача 1:** Превърнете дадения по-долу DTD документ в XML Schema. Създайте XML инстанция на тази схема и я валидирайте.

```
<!ELEMENT collection (description,recipe*)>
<!ELEMENT description ANY>
<!ELEMENT recipe (title,ingredient*,preparation,comment?,nutrition)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT ingredient (ingredient*,preparation)?>
<!ATTLIST ingredient name CDATA #REQUIRED
                    amount CDATA #IMPLIED
                    unit CDATA #IMPLIED>
<!ELEMENT preparation (step*)>
<!ELEMENT step (#PCDATA)>
<!ELEMENT comment (#PCDATA)>
<!ELEMENT nutrition EMPTY>
<!ATTLIST nutrition protein CDATA #REQUIRED
                    carbohydrates CDATA #REQUIRED
                    fat CDATA #REQUIRED
                    calories CDATA #REQUIRED
                    alcohol CDATA #IMPLIED>
```

## Решение

**Задача 2:** Дадения по-долу XML файл описва типовете сметки, които поддържа една примерна банка, нейните клиенти и сметките, които те имат. За този XML документ, създайте XML Schema, която изпълнява следните условия:

- Редът на срещане на под-елементите на bank (accounts, customers и customer\_accounts) и accounts (saving\_accounts и checking\_accounts) няма значение
- Всяка сметка има уникален идентификатор
- Всеки клиент има уникален идентификатор
- Баланса на сметката не може да бъде по-малък от -5000 и за това условие използвайте рестрикция на съществуващите предефинирани типове
- Атрибутът c\_id реферира към съответния клиент, а ac\_id към съответната му сметка

- Дефинирайте елементите customers и saving\_account като комплексен глобален тип, а елемента accounts като комплексен локален тип

```
<?xml version="1.0" encoding="UTF-8"?>
<bank>
  <accounts>
    <saving_accounts>
      <saving_account id="a1" interest="0.03">
        <balance>2500</balance>
      </saving_account>
      <saving_account id="a2" interest="0.03">
        <balance>15075</balance>
      </saving_account>
    </saving_accounts>
    <checking_accounts>
      <checking_account id="a3">
        <balance>4025</balance>
      </checking_account>
      <checking_account id="a4">
        <balance>-125</balance>
      </checking_account>
      <checking_account id="a5">
        <balance>325</balance>
      </checking_account>
    </checking_accounts>
  </accounts>
  <customers>
    <customer id="c1">
      <name>Ben Richerdson</name>
      <address>Park Drive 2</address>
    </customer>
    <customer id="c2">
      <name>Marc Wretcher</name>
```

```
<address>Mill Drive 75</address>
</customer>
<customer id="c3">
  <name>Angel Steady</name>
  <address>Lake Sight 15</address>
</customer>
</customers>
<customer_accounts>
  <customer_account c_id="c1" ac_id="a2"/>
  <customer_account c_id="c1" ac_id="a3"/>
  <customer_account c_id="c2" ac_id="a4"/>
  <customer_account c_id="c3" ac_id="a1"/>
  <customer_account c_id="c3" ac_id="a5"/>
</customer_accounts>
</bank>
```

## Решение

### Задача 3: В XML схемата от задача 2 направете следните промени:

- Дефинирайте два прости типа задаващи горна и долна граница на стойността на елемента balance. След това променете дефиницията на елемента balance като го представите като обединение на тези два прости типа
- Използвайки разширени типове на XML Schema, създайте нов тип customerExt, който разширява дефиницията на типа customer като добавя нов негов под-елемент contacts, който от своя страна се състои от 2 под-елемента - email и telephone. Използвайте новия тип customerExt вместо customer и запишете новата XML схема под името bank.xsd
- Създайте нова XML схема, която предефинира типа balance и customerExt от външната за нея XML схема - bank.xsd. Новият тип balance не трябва да бъде по-малък от -5200, а новият тип customerExt съдържа допълнително нов под-елемент image от тип base64Binary, който има един атрибут src от тип string. Запишете новата XML схема под името bankExt.xsd
- Добавете няколко анотации в по-горе създадената XML схема (bank.xsd) към избрани от вас комплексните типове, описващи тяхното предназначение

## Решение

**Задача 4:** Редактирайте XML схемата от задача 3 (bank.xsd) като добавите елемент postalCode към комплексния тип customer и включите регулярен израз за:

- пощенски код (четири цифрено число, например:1000)
- телефон (например в следния формат:+359-02-989-14-04)
- електронна поща

## Упътване

Синтаксис за използване на регулярен израз:

```
<xsd:restriction base="XXXX">  
  <xsd:pattern value="Regular_Expression"/>  
</xsd:restriction>
```

## Решение

**Задача 5:** Редактирайте XML схемата от задача 3 (bankExt.xsd) като добавите към елемента image атрибут type от тип нотация. Създайте XML инстанция на новата схема и валидирайте.

## Упътване

Синтаксис за дефиниране на нотации в XML схема: <notation id=ID name=NCName public=anyURI system=anyURI any attributes> (annotation?)</notation>

## Решение

Задача 6: Съставете XML Schema, която включва схеми от други пространства от имена. Създайте XML инстанция на тази схема и я валидирайте.

#### Упътване

Синтаксис за импортиране на външна схема: `<import id=ID namespace=anyURI schemaLocation=anyURI any attributes > (annotation?)</import>`

#### Решение



1

2

3

4

5

6

7

8