

## TP 2 XML

**Objectif :** Documents XML et DTD/XML schéma

### Exercice 1:

- a. Construisez un fichier XML bien formé contenant un carnet d'adresses ou agenda téléphonique.  
Chaque enregistrement, nommé «contact» spécifie:
  - Indication si le contact est une personne ou une entreprise, etc.
  - Le nom,
  - Le prénom (seulement pour les personnes),
  - Le numéro de téléphone.
  - Une adresse qui contient les informations suivantes: Rue (une chaîne); Numéro(un numéro entier) ; **Ville**; **Code** postal (éventuellement **CEDEX**); **Pays** (une chaîne)
- b. Donnez un Schéma XML pour le fichier construit et validez le fichier XML par rapport à cette définition. Utiliser le validateur XML schéma de votre outils XML préféré (editix par exemple)

### Exercice 2:

Soit le fichier bank.xml suivant :

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

```
<bank xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="bank.xsd">
  <accounts>
    <savings_accounts>
      <savings_account id="a1" interest="0.03">
        <balance>2500</balance>
      </savings_account>
      <savings_account id="a2" interest="0.03">
        <balance>15075</balance>
      </savings_account>
    </savings_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>
</bank>
```

```

<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>

```

Ecrire un XML schéma pour le fichier bank.xml conformément aux règles suivantes :

- Il existe deux types de comptes : checking accounts et savings accounts
- "id" unique pour les comptes dans "accounts"
- "id" unique pour les clients dans "customers"
- "c\_id" référence vers les clients et "ac\_id" référence les comptes
- Le solde (account balance) doit être en dessus de -5000

- Utiliser l'héritage pour définir "checking accounts" et "saving accounts" en se basant sur un type en commun "account"

- Réfléchissez sur la cohérence de votre solution en termes de structure de fichier XML, des balises ou attributs XML. Validez votre XSD ainsi que votre XML avec votre outil XML.

### Exercice 3:

On souhaite représenter des expressions mathématiques simples dans un format XML proche de celui de MathML (standard du W3C). Les principes du format proposé sont les suivants :

- On représente une valeur numérique comme le contenu d'une balise `<cn>`. Par exemple `<cn>15</cn>`
- Une opération est représentée par une balise `<apply>`. En général la balise contient trois sous-balises : La première sous-balise est **vide** (la balise `<plus/>` dans l'exemple suivant) et représente l'opération à réaliser. Les deux autres sont les opérandes de l'opération. Par exemple,  $2+3$  s'écrit :

```

<apply>
  <plus/>
  <cn>2</cn>
  <cn>3</cn>
</apply>".

```

- On propose les opérateurs suivants :
  - plus : addition
  - times : multiplication
  - divide : division
  - minus : soustraction
  - quotient : division entière
- Tous les opérateurs sont binaires sauf la soustraction qui peut aussi être unaire et représente alors le changement de signe.
- un élément `<apply>` peut contenir un ou deux éléments `<apply>` comme opérande(s). Cette inclusion joue le rôle de parenthèses. Par exemple, pour représenter  $(2 + 3) * 4$ , on écrit :

```

<apply>
  <times/>
  <apply>
    <plus/>
    <cn>2</cn>
    <cn>3</cn>
  </apply>
  <cn>4</cn>
</apply>

```

Ecrire un schéma pour ce langage XML proposé.

**NB :** Pour définir une balise vide en xml schéma, on utilise l'attribut "nillable" avec la valeur "true" (par exemple : `<xsd:element name="toto" nillable="true"/>`). On utilise le type complexe "xsd:choice" pour définir le choix d'une balise parmi un ensemble de balises. La syntaxe de ce type se définit de la même façon que le type "xsd:sequence"