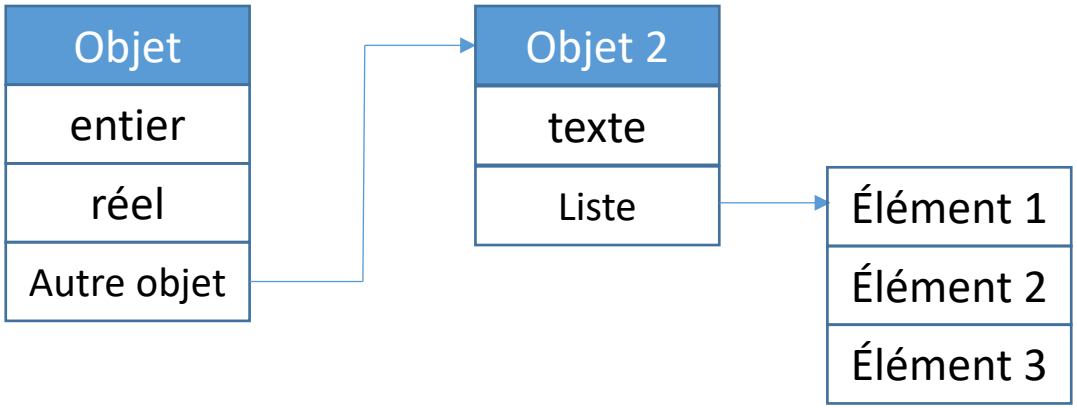
Bases de données (organisation générale)

Répétition 10

Les bases de données orientées-objet et XML

Les bases de données orientées-objet : pourquoi?

- Structures d'objets complexes:



- Contenu non-textuel : audio, vidéo, …

Rappel sur les BD-OO

- Une base de données orientée-objet pose d'abord l'existence des éléments suivants:
- D_i : un ensemble fini de domaines :

  D_{int}: L'ensemble des entiers

  D_{pays} : L'ensemble des pays

  ...
- A : un ensemble dénombrable d'attributs :

  NOMBRE ETUDIANTS, PAYS, ...
- ID : un ensemble dénombrable d'identificateurs :

  OBJ1, OBJ2, ...

Valeurs d'objets

- Valeurs de base

  nil

  $d \in D$

- Valeurs ensemble

  Sous-ensemble de ID

- Valeurs tuple

  Fonctions partielles de A vers ID

L'ensemble des objets

- Objet

  $(id, v), id \in ID, v$ valeur(s) de l'objet

- Existe sous 3 formes possibles:
  - Objet de base

    (id, v) où v est une valeur de base et de type D_i , avec $v \in D_i$
  - Objet ensemble

    (id, v) où v est une valeur d'ensemble et de type 2^T avec T union des types des objets $(id', v') \mid id' \in v$.
  - Objet tuple

    (id, v) où v est une valeur tuple et de type $\{(a_i, T_i)\} \mid v(a_i)$ est définie, T_i est le type de l'objet $(id', v') \mid id' = v(a_i)\}$.

Exemple d'objet (tuple)

- Une personne est un objet tuple de type

$$T_{pers} = \{(NOM, D_{string}), (PRENOM, D_{string}), (DATE\_NAISS, D_{date}), (ENFANTS, T_{enfants}), (ADRESSE, T_{adresse})\}.$$

Avec le type $T_{adresse} =$

$$\{(RUE, D_{string}), (NUM, D_{int}), (CP, D_{int}), (LOC, D_{string})\}$$

Et le type $T_{enfants} = 2^{D_{string}}$

Exercice 1

Un gérant de salons-lavoir désire conserver dans une base de données orientée-objet les informations concernant les différents modèles de machine a laver disponibles sur le marché.

Définissez une base de données orientée-objet satisfaisant les propriétés suivantes :

- On associe a chaque modèle de machine un nom de modèle, un fabricant, un lieu de fabrication et la liste des programmes de lavage disponible ainsi que le prix d'achat et le nombre de mois de garantie.
- Les fabricants sont caractérisés par un nom, un pays, le nombre d'employés et l'année de création
- Un programme de lavage contient au maximum 5 phases, chacune des phases étant décrite par sa durée, sa description, sa consommation d'eau et d'électricité ainsi que la vitesse de rotation du tambour durant la phase.

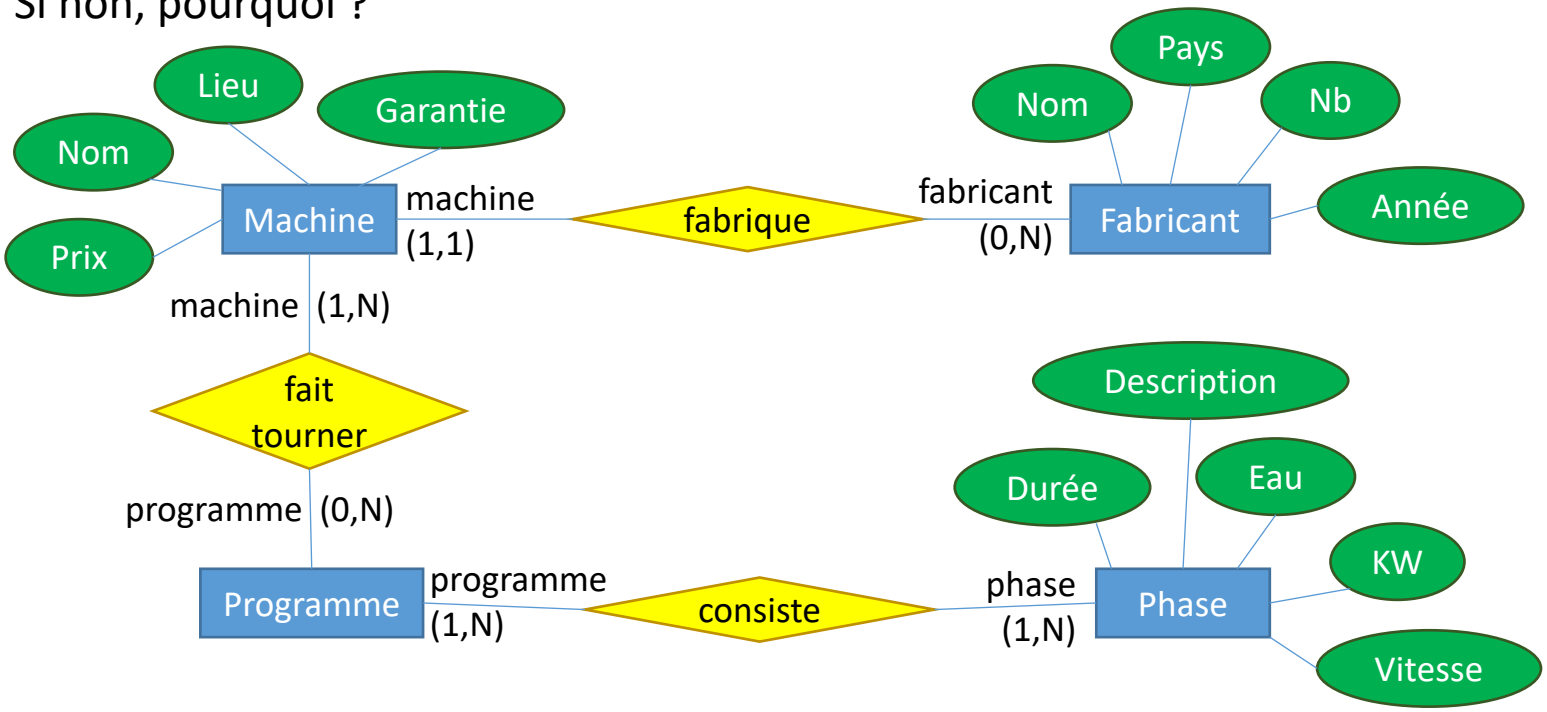
1. Ces informations peuvent-elle être représentées par un modèle entité-relation classique ?

   Si non, pourquoi ?
2. Donnez le modèle de la base de données orientée-objet correspondant à ce problème.

Exercice 1

1. Ces informations peuvent-elle être représentées par un modèle entité-relation classique ?

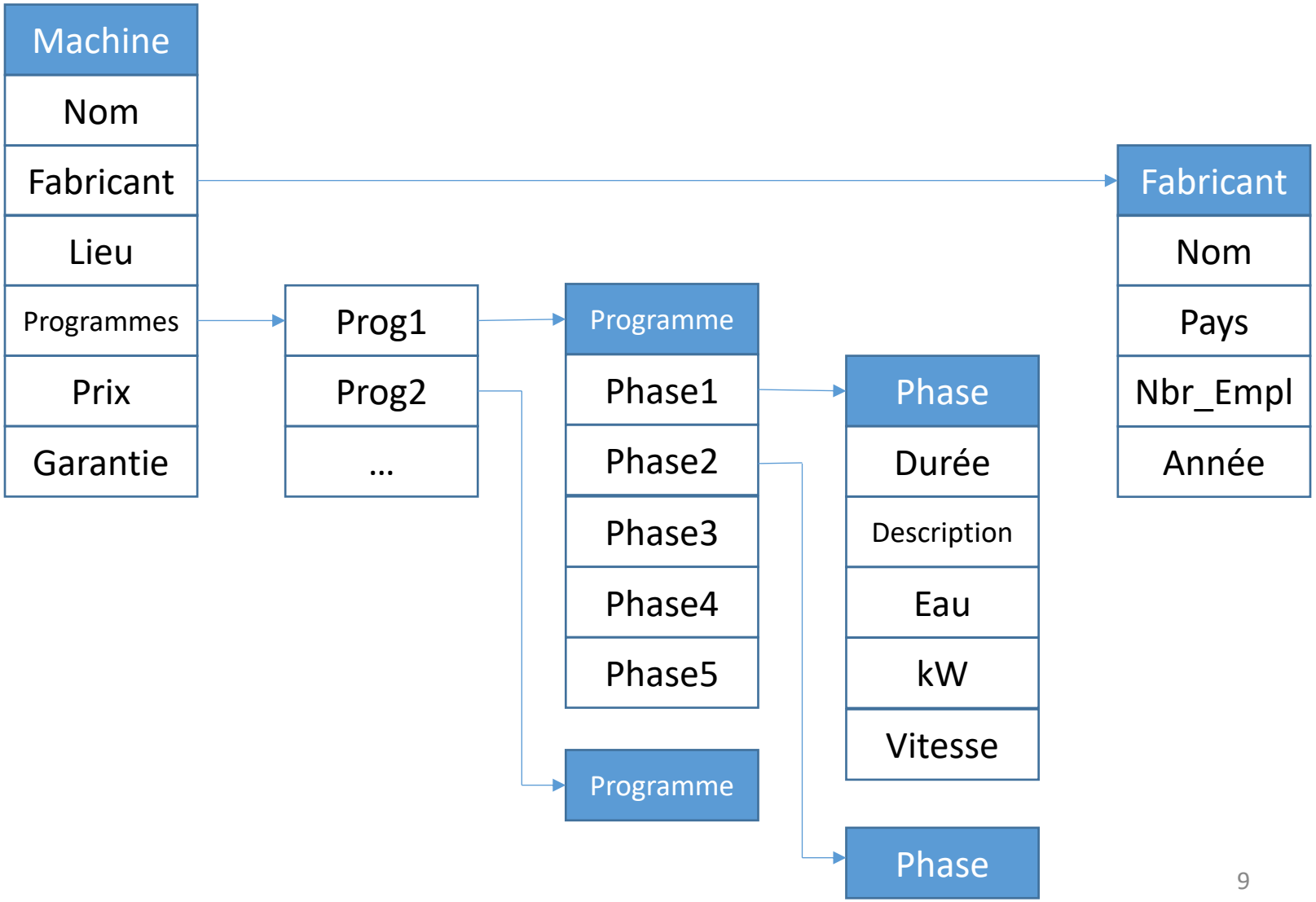
   Si non, pourquoi ?



- Nécessité d'attributs supplémentaires pour les clés (e.g. *Programme*)
- La contrainte suivante n'est pas représentable aisément : "Un programme a au maximum 5 phases".
  - On pourrait imaginer 5 relations entre *Programme* et *Phase*, mais que deviendrait mon modèle si j'avais une contrainte de type "Un programme a au maximum 150 phases"?

Exercice 1

- Penser en termes d'objets



Exercice 1

- Définition des domaines de base

D_{string}	:	L'ensemble des chaînes de caractères
D_{int}	:	L'ensemble des entiers
D_{float}	:	L'ensemble des nombres en virgule flottante
D_{pays}	:	L'ensemble des pays
D_{annee}	:	L'ensemble des années
$D_{vitesse}$	:	L'ensemble des vitesses de rotation

Exercice 1

- Fabricants : objets tuples de type

$$T_{fab} = \{(NOM, D_{string}),$$
$$(PAYS, D_{pays}),$$
$$(NBR\_EMPL, D_{int}),$$
$$(ANNEE, D_{annee})\}$$

Exercice 1

- Phases de lavage : objets tuples de type

$$T_{phase} = \{(DUREE, D_{string}),$$
$$(DESCRIPTION, D_{string}),$$
$$(EAU, D_{float}),$$
$$(KW, D_{float}),$$
$$(VITESSE, D_{vitesse})\}$$

Exercice 1

- Programmes de lavage : objets tuples de type

$$
\begin{aligned}
T_{prog} \;=\; \{&\big(PHASE1, T_{phase}\big),\\
&\big(PHASE2, T_{phase}\big),\\
&\big(PHASE3, T_{phase}\big),\\
&\big(PHASE4, T_{phase}\big),\\
&(PHASE5, T_{phase})\}
\end{aligned}
$$

Exercice 1

- Listes de programmes: objets ensembles de type

$$T_{LIST\_PROG} = 2^{T_{prog}}$$

Exercice 1

- Machines : objets tuples de type

$$T_{machine} = \{(NOM, D_{string}),$$
$$(FABRICANT, T_{fab}),$$
$$(LIEU, D_{pays}),$$
$$(PROGS, T_{LIST\_PROG}),$$
$$(PRIX, D_{float}),$$
$$(GARANTIE, D_{int})\}$$

Exercice 2

Une ville conserve des informations sur son réseau routier dans une base de données orientée-objet. Les objets représentés dans la base de données sont les suivants :

- Des points géographiques caractérisés par leurs coordonnées (latitude, longitude).
- Des segments de route caracterisés par un point début et un point fin, et les sens de circulation autorises (debut-fin (*df)*, fin-debut (*fd*) ou bi-directionnel (*bi*)).

On demande de définir les types des objets figurant dans la base de données orientée-objet.

Exercice 2

$D_{latitude}$: L'ensemble des latitudes

$D_{longitude}$: L'ensemble des longitudes

D_{sens} : L'ensemble des sens de circulation

$$T_{coord} = \{(LATITUDE, D_{latitude}), (LONGITUDE, D_{longitude})\}$$

$$T_{segment} = \{(DEBUT, T_{coord}), (FIN, T_{coord}), (SENS, D_{sens})\}$$

$$T_{route} = 2^{T_{segment}}$$

Le langage XML : pourquoi?

e**X**tensible **M**arkup **L**anguage.

Permet de conserver les données avec leur structure.

L'interprétation de cette structure n'est pas définie dans le document (possibilité d'utiliser un fichier *.css* pour ce faire).

```
<voiture>
        <marque>Opel</marque>
        <modele>Astra</marque>
        <moteur>
                <marque>Isuzu</marque>
                <puissance>110cv</puissance>
                <carburant>Diesel</carburant>
        </moteur>
</voiture>
```

Les cousins de XML : HTML et XHTML

- HTML : Hyper-Text Markup Langage
  - Similaire au XML, mais le contenu des balises est prédéfini.
  - Le langage est permissif :
    - Le type (DOCTYPE) et l'encodage (e.g. UTF-8) sont optionnels.
    - pas besoin de fermer certaines balises.
    - Certaines balises sont optionnelles
  - Exemple:

```
<HTML>
    <HEAD>
        <TITLE>Ceci est mon titre</TITLE>
    </HEAD>
    <BODY>
        Voici le corps de la page HTML.
    </BODY>
</HTML>
```

- XHTML : version plus "stricte" de HTML.
  - Utilise un DTD fourni par le W3C.
  - Fermeture des balises obligatoires (ou / en fin de balise).
  - DOCTYPE et encodage obligatoires
  - Balises écrites en minuscule uniquement.

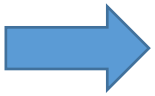
DTD : Data Type Definition

- Définit la structure du document XML
- Pour être valide, un contenu de fichier XML doit respecter le DTD.
- Exemple de DTD:

$$T_{pers} = \{(NOM, D_{string}), (PRENOM, D_{string}), (DATE\_NAISS, D_{date}), (ENFANTS, T_{enfants}), (ADRESSE, T_{adresse})\}.$$

$$T_{adresse} = \{(RUE, D_{string}), (NUM, D_{int}), (CP, D_{int}), (LOC, D_{string})\}.$$

$$T_{enfants} = 2^{D_{string}}$$

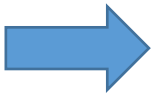


```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE pers [
<!ELEMENT pers (nom,prenom,
        date_naiss,enfants, adresse)>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT prenom (#PCDATA)>
<!ELEMENT date_naiss (#PCDATA)>
<!ELEMENT enfants (enfant*)>
<!ELEMENT enfant (#PCDATA)>
<!ELEMENT adresse (rue, num, cp, loc)>
<!ELEMENT rue (#PCDATA)>
<!ELEMENT num (#PCDATA)>
<!ELEMENT cp (#PCDATA)>
<!ELEMENT loc (#PCDATA)>
]>
```

Exemple de contenu valide

- Pour le DTD donné, un contenu valide est par exemple (vérifiable sur http://www.xmlvalidation.com/) :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE pers [
<!ELEMENT pers (nom,prenom,
        date_naiss,enfants, adresse)>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT prenom (#PCDATA)>
<!ELEMENT date_naiss (#PCDATA)>
<!ELEMENT enfants (enfant*)>
<!ELEMENT enfant (#PCDATA)>
<!ELEMENT adresse (rue, num, cp, loc)>
<!ELEMENT rue (#PCDATA)>
<!ELEMENT num (#PCDATA)>
<!ELEMENT cp (#PCDATA)>
<!ELEMENT loc (#PCDATA)>
]>
```



```
<pers>
    <nom>Hiard</nom>
    <prenom>Samuel</prenom>
    <date_naiss>09/09/1980</date_naiss>
    <enfants>
    </enfants>
    <adresse>
        <rue>Grande traverse</rue>
        <num>10</num>
        <cp>4000</cp>
        <loc>Angleur</loc>
    </adresse>
</pers>
```

Exercice 3

- Donnez le DTD d'un fichier XML représentant un ou plusieurs segments de route.
- Donnez le DTD d'un fichier XML représentant une machine à laver.

Exercice 3

Rappel de la structure des objets:

$D_{latitude}$: L'ensemble des latitudes

$D_{longitude}$: L'ensemble des longitudes

D_{sens} : L'ensemble des sens de circulation

$$T_{coord} = \{(LATITUDE, D_{latitude}), (LONGITUDE, D_{longitude})\}$$

$$T_{segment} = \{(DEBUT, T_{coord}), (FIN, T_{coord}), (SENS, D_{sens})\}$$

$$T_{route} = 2^{T_{segment}}$$

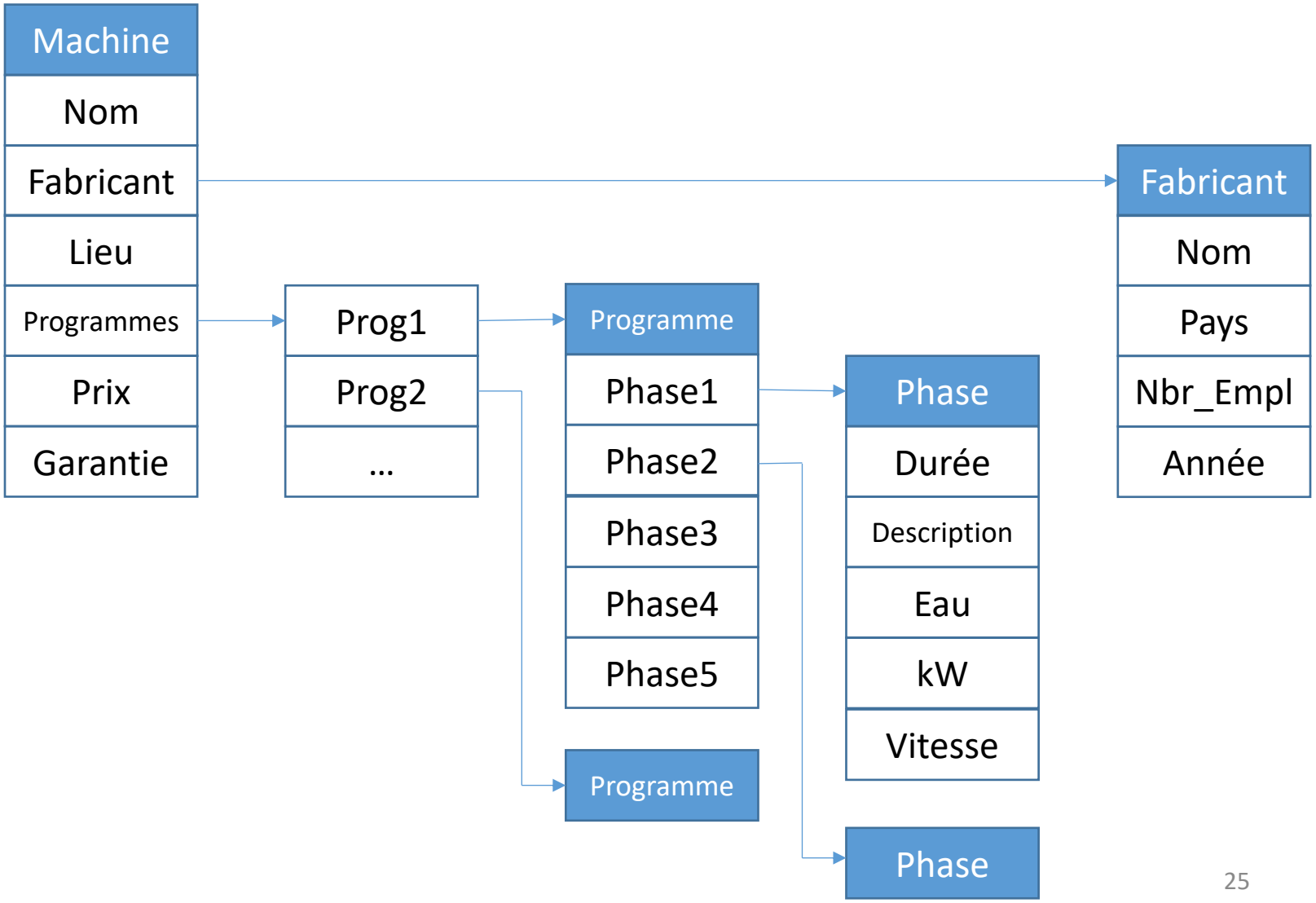
Exercice 3

- DTD correspondant aux informations de segments de route:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE segments [
<!ELEMENT segments (segment+)>
<!ELEMENT segment (debut, fin, sens)>
<!ELEMENT debut (coord)>
<!ELEMENT fin (coord)>
<!ELEMENT sens (#PCDATA)>
<!ELEMENT coord (latitude, longitude)>
<!ELEMENT latitude (#PCDATA)>
<!ELEMENT longitude (#PCDATA)>
]>
```

Exercice 3

- Rappel de la structure des objets :



Exercice 3

- DTD correspondant aux informations de machines à laver:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE machine [
<!ELEMENT machine (nom, fabricant, lieu,
programmes, prix, garantie)>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT lieu (#PCDATA)>
<!ELEMENT prix (#PCDATA)>
<!ELEMENT garantie (#PCDATA)>
<!ELEMENT fabricant (nom, pays,
nbr_empl, annee)>
<!ELEMENT pays (#PCDATA)>
<!ELEMENT nbr_empl (#PCDATA)>
<!ELEMENT annee (#PCDATA)>
<!ELEMENT programmes (programme+)>
<!ELEMENT programme (phase, phase?,
        phase?, phase?, phase?)>
<!ELEMENT phase (duree, description,
        eau, kw, vitesse)>
<!ELEMENT duree (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT eau (#PCDATA)>
<!ELEMENT kw (#PCDATA)>
<!ELEMENT vitesse (#PCDATA)>
]>
```

Exercice 4

- Donnez un exemple de contenu valide d'un fichier XML représentant au moins 2 segments de route.
- Donnez un exemple de contenu valide d'un fichier XML représentant au moins 1 machine à laver.

Exercice 4 (Rappel)

- DTD correspondant aux informations de segments de route:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE segments [
<!ELEMENT segments (segment+)>
<!ELEMENT segment (debut, fin, sens)>
<!ELEMENT debut (coord)>
<!ELEMENT fin (coord)>
<!ELEMENT sens (#PCDATA)>
<!ELEMENT coord (latitude, longitude)>
<!ELEMENT latitude (#PCDATA)>
<!ELEMENT longitude (#PCDATA)>
]>
```

Exercice 4

- Représentation possible de deux segments de route

```
<segments>
        <segment>
                <debut>
                        <coord>
                                <latitude>50.712461N</latitude>
                                <longitude>5.081593E</longitude>
                        </coord>
                </debut>
                <fin>
                        <coord>
                                <latitude>50.677339N</latitude>
                                <longitude>5.460450E</longitude>
                        </coord>
                </fin>
                <sens>bi</sens>
        </segment>
```

Exercice 4 (suite)

```
        <segment>
                <debut>
                        <coord>
                                <latitude>48.870184N</latitude>
                                <longitude>2.779932E</longitude>
                        </coord>
                </debut>
                <fin>
                        <coord>
                                <latitude>48.872584N</latitude>
                                <longitude>2.776767E</longitude>
                        </coord>
                </fin>
                <sens>df</sens>
        </segment>
</segments>
```

Exercice 4 (Rappel)

- DTD correspondant aux informations de machines à laver:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE machine [
<!ELEMENT machine (nom, fabricant, lieu,
programmes, prix, garantie)>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT lieu (#PCDATA)>
<!ELEMENT prix (#PCDATA)>
<!ELEMENT garantie (#PCDATA)>
<!ELEMENT fabricant (nom, pays,
nbr_empl, annee)>
<!ELEMENT pays (#PCDATA)>
<!ELEMENT nbr_empl (#PCDATA)>
<!ELEMENT annee (#PCDATA)>
<!ELEMENT programmes (programme+)>
<!ELEMENT programme (phase, phase?,
        phase?, phase?, phase?)>
<!ELEMENT phase (duree, description,
        eau, kw, vitesse)>
<!ELEMENT duree (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT eau (#PCDATA)>
<!ELEMENT kw (#PCDATA)>
<!ELEMENT vitesse (#PCDATA)>
]>
```

Exercice 4

- Représentation possible d'une machine à laver:

```
<machine>
        <nom>WKR 770 WPS</nom>
        <fabricant>
                <nom>Miele</nom>
                <pays>Allemagne</pays>
                <nbr_empl>16600</nbr_empl>
                <annee>1899</annee>
        </fabricant>
        <lieu>Republique tcheque</lieu>
        <programmes>
                (voir page suivante)
        </programmes>
        <prix> 2000 euros</prix>
        <garantie>15 ans</garantie>
</machine>
```

Exercice 4 (suite)

- Liste des programmes (ici, un seul):

```
<programme>
        <phase>
                <duree>30 minutes</duree>
                <description>Lavage</description>
                <eau>50 litres</eau>
                <kw>0.1</kw>
                <vitesse>600 Tr/min</vitesse>
        </phase>
        <phase>
                <duree>5 minutes</duree>
                <description>Essorage</description>
                <eau>0</eau>
                <kw>0.5</kw>
                <vitesse>1600 Tr/min</vitesse>
        </phase>
</programme>
```