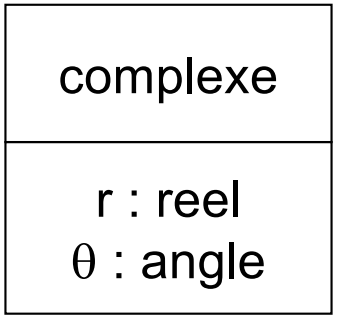
SysML/UML
Diagrammes de définition de blocs
Diagrammes de classes
Diagrammes d'objets

Architecture système et SysML

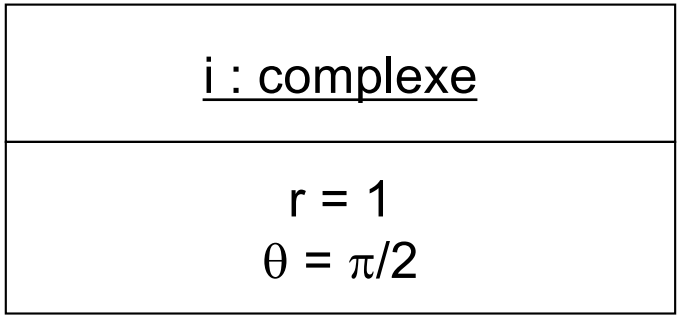
- SysML modélise l'architecture système en réutilisant (légèrement adaptés) trois diagrammes UML.
- Le diagramme de définition de blocs (très largement inspiré du diagramme de classes).
- Le diagramme de bloc interne (très largement inspiré du diagramme de structure composite avec ses notions d'interfaces de ports etc.).
- Le diagramme de package, reconduit à l'identique.
- Dans ce cours on décrit le diagramme de classes (tel qu'il était conçu pour développer les classes en langage objet) et les (petites) adaptations pour en faire le diagramme de définition de blocs.

Les classes

- Type abstrait d'objets partageant des caractéristiques communes
- Un objet est alors une instance (un représentant) de cette classe



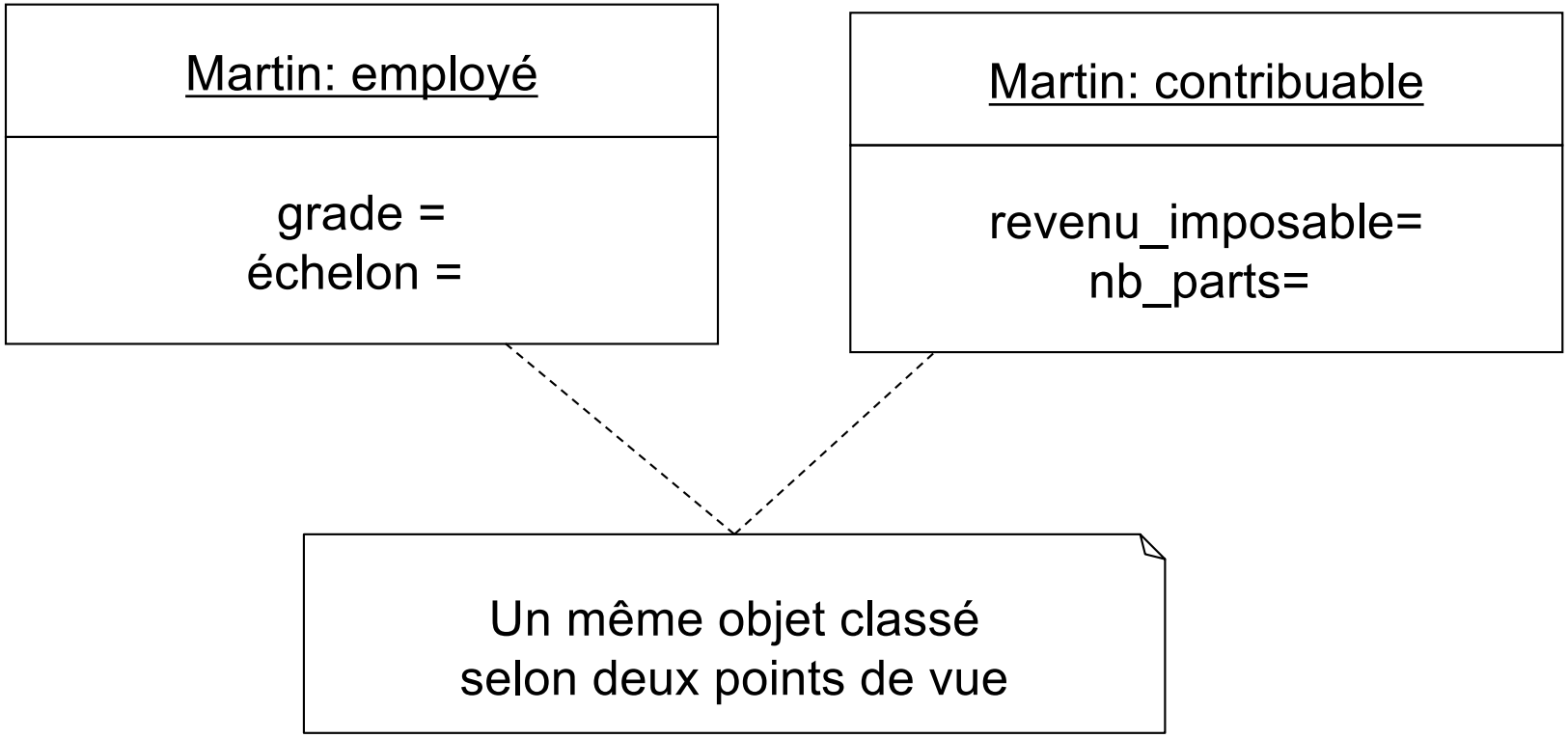
Une classe



Un objet instance de cette classe

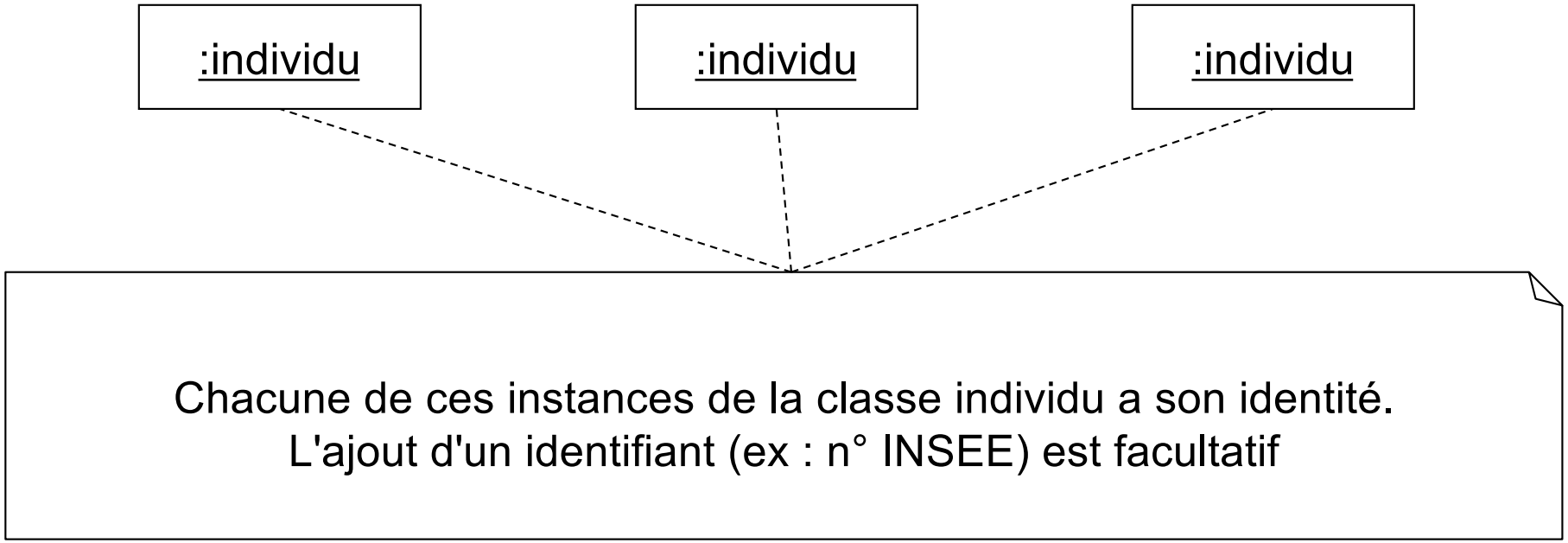
Démarche de classification

- La classification n'est pas unique (dépend du point de vue)

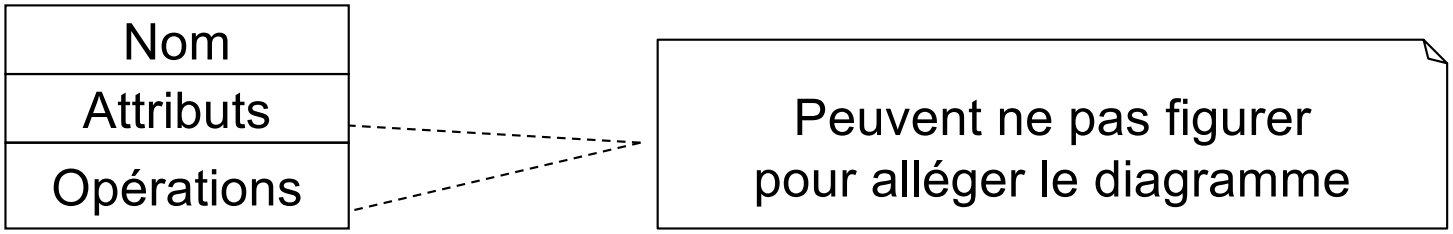


Identité des objets

- Chaque objet a sa propre identité indépendamment de son nom (qui peut ne pas exister)



Représentation des classes



- **Syntaxe des attributs :**
  - visibilité nom[multiplicité]:type=valeur_initiale{propriété}
- **Syntaxe des opérations**
  - visibilité nom(arguments):type{propriété}
  - multiplicité, valeur initiale et propriété sont optionnelles
  - visibilité type et arguments existent toujours mais peuvent ne pas figurer pour alléger le diagramme

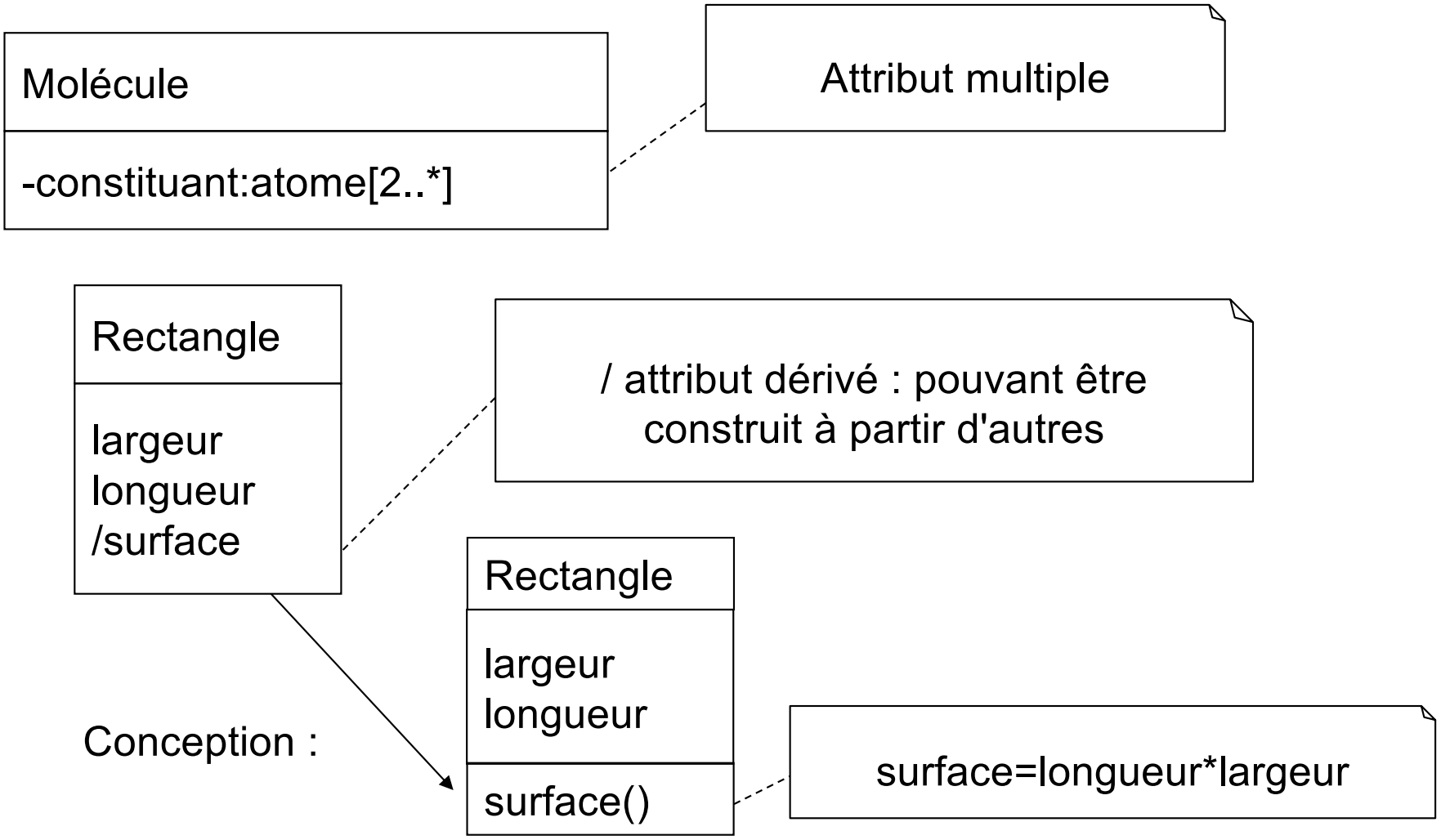
Visibilité des attributs ou opérations

- + : public : élément visible de l'extérieur
- - : protégé : élément visible seulement des sous-classes
- # : privé : élément invisible de l'extérieur
- Syntaxe :
  - +Opération_publique(arguments):type
  - -Attribut_protégé:type
  - #Attribut_privé:type

Types et propriétés

- Types :
  - Types "primitifs" (booléen, entier, réel, chaîne de caractère...) : -longueur : entier
  - Types énumérés : -couleur:enum{noir,blanc}
  - Classes : -emprunteur:individu
- Propriétés :
  - Permettent d'exprimer des contraintes comme l'aspect non modifiable d'un attribut etc.

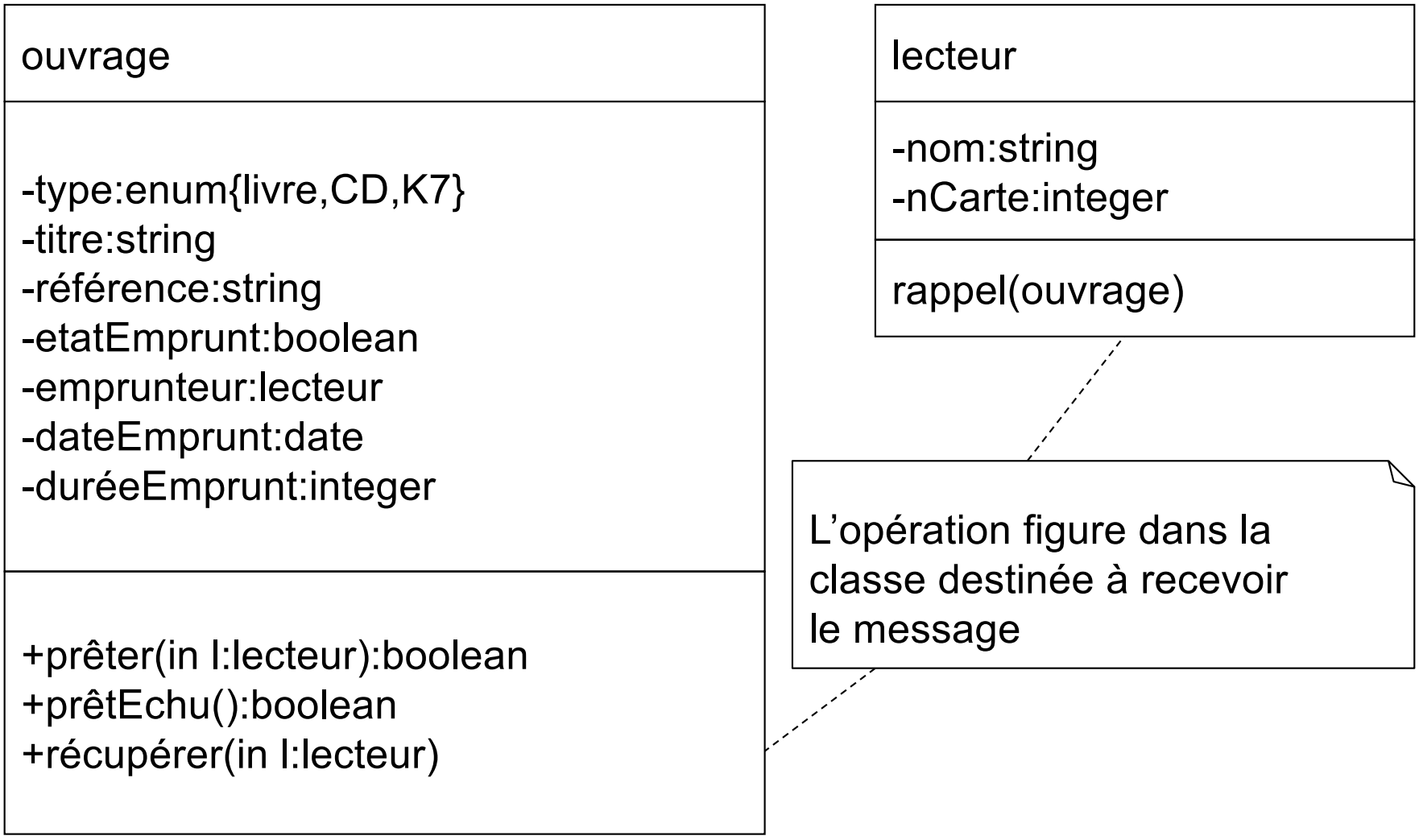
Attributs multiples, attributs dérivés



Arguments des opérations

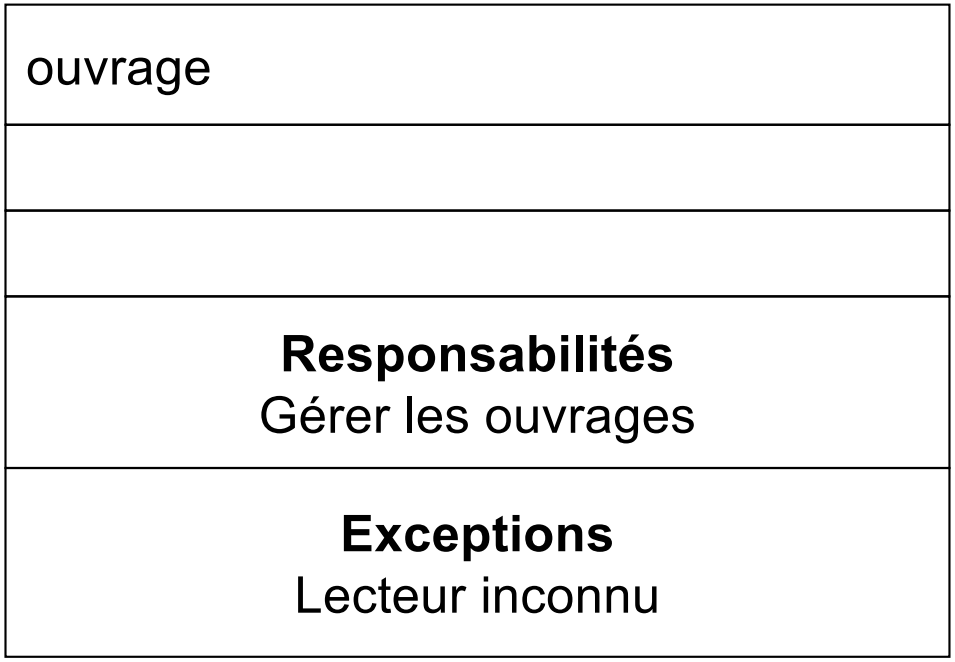
- Un argument :
  - direction nom : type=valeur_par_défaut
- Direction :
  - in : paramètre d'entrée ne pouvant être modifié (par défaut si non précisée)
  - out : paramètre de sortie
  - inout : paramètre d'entrée pouvant être modifié

Exemple



Responsabilités, exceptions

- Des compartiments supplémentaires peuvent être ajoutés pour préciser les responsabilités d'une classe et les exceptions qu'elle doit traiter. UML n'impose pour cela aucune syntaxe particulière :



Adaptations SysML

- L'architecture système ne comporte pas de classe à proprement parler, mais des blocs (peuvent être des systèmes, sous-systèmes, composants) sachant que ces blocs pourront être décomposés (diagramme de bloc interne)
- Le bloc est vu comme une spécialisation (stéréotypage) de la notion de classe (stéréotype représenté par le mot clé <<block>> entre <<>> comme il est d'usage en UML)
- La notion d'attributs est remplacée par celle de values (avec une signification très voisine)
- Il est possible d'exprimer qu'un bloc est composé en indiquant ses parties dans un compartiment parts

Adaptations SysML

Exemple

<<Block>> Ligne de métro
Parts Trains : train[1..*] Voie : type voie
Values Longueur : integer Vitesse_comm : integer

<<Block>> Train
Parts Bogies : type_bogie[2..*] Caisse : type_caisse
Values Masse: integer Vitesse_max : integer

- Un bloc peut aussi posséder des opérations (démarrer(), arrêter() pour le bloc train par exemple)
- SysML utilise dans le Block Definition Diagram (BDD) les notions d'association, d'agrégation, de composition, de généralisation / spécialisation etc. exactement comme UML (voir cours suivants)