



Cours de Génie Logiciel



Sciences-U Lyon

Diagrammes UML

<http://www.rzo.free.fr>



Sommaire



- Les Diagrammes UML
 - **Niveaux de modèles**
 - Éléments communs aux diagrammes
 - Diagrammes de cas d'utilisation
 - Diagrammes de Classes
 - Diagrammes d'Objets
 - Diagrammes de Séquence

Niveaux de Modèles

- Les Niveaux de modèles

M3	MOF – Meta modèle de définition de langage
M2	Modèle de de langage
M1	Classes d'un programme
M0	Instanciacion



UML



Sommaire

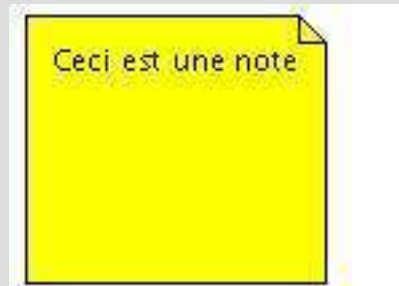


- Les Diagrammes UML
 - Niveaux de modèles
 - **Éléments communs aux diagrammes**
 - Diagrammes de cas d'utilisation
 - Diagrammes de Classes
 - Diagrammes d'Objets
 - Diagrammes de Séquence

Eléments communs

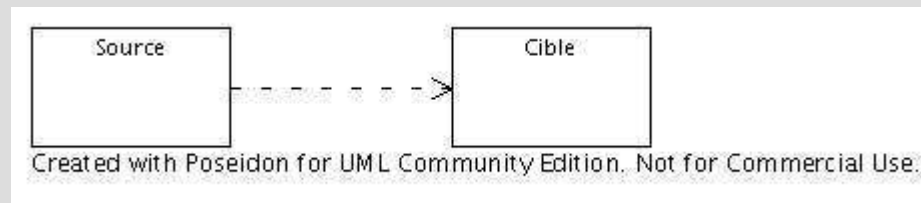
- Eléments communs aux diagrammes

- Note



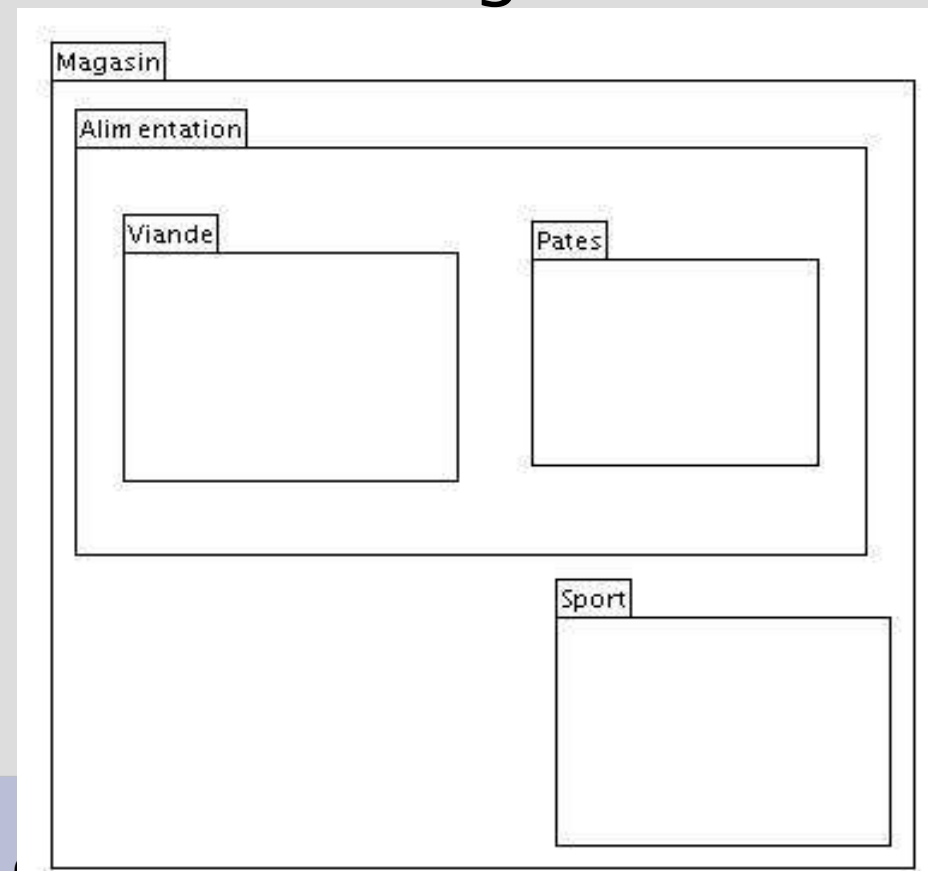
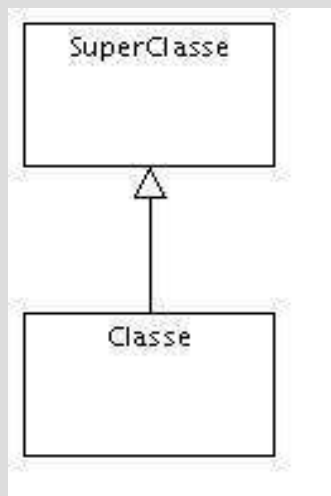
- Dépendance

- Modification dans la cible implique modification dans la source



Éléments communs

- Éléments communs aux diagrammes
 - Héritage
 - Packages





Éléments communs



- Éléments communs aux diagrammes
 - Stéréotypes
 - Classe prédéfinie
 - <<acteur>>



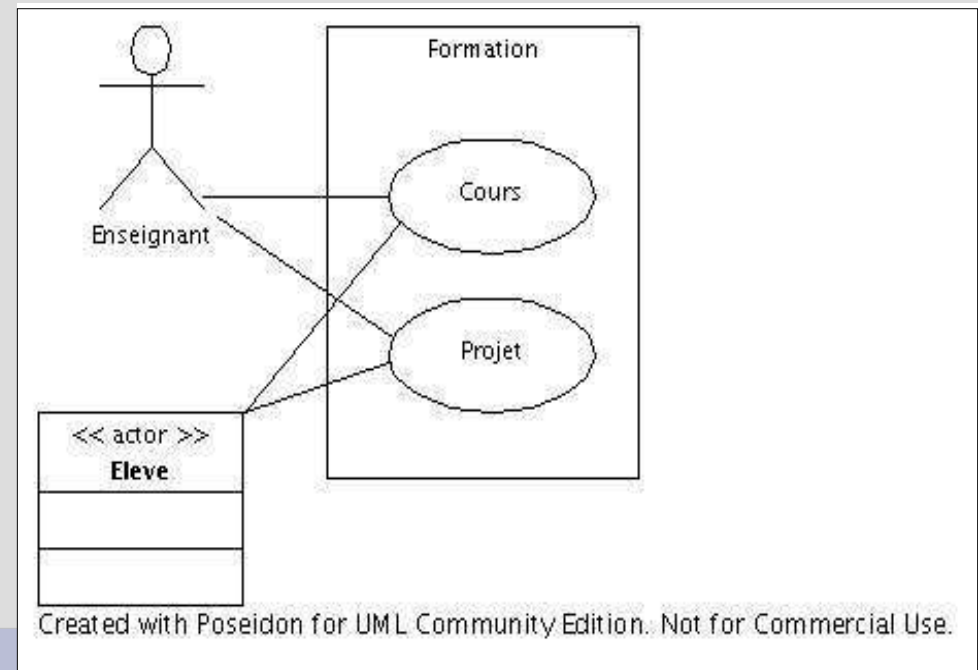
Sommaire



- Les Diagrammes UML
 - Niveaux de modèles
 - Elements communs aux diagrammes
 - **Diagrammes de cas d'utilisation**
 - Diagrammes de Classes
 - Diagrammes d'Objets
 - Diagrammes de Séquence

Cas d'utilisation

- Les Diagrammes de cas d'utilisation
 - Objectif : détermination des besoins
 - Fonctionnalités du système
 - Acteurs
 - Cas d'utilisation
 - Relations entre les 2
 - association
 - Système



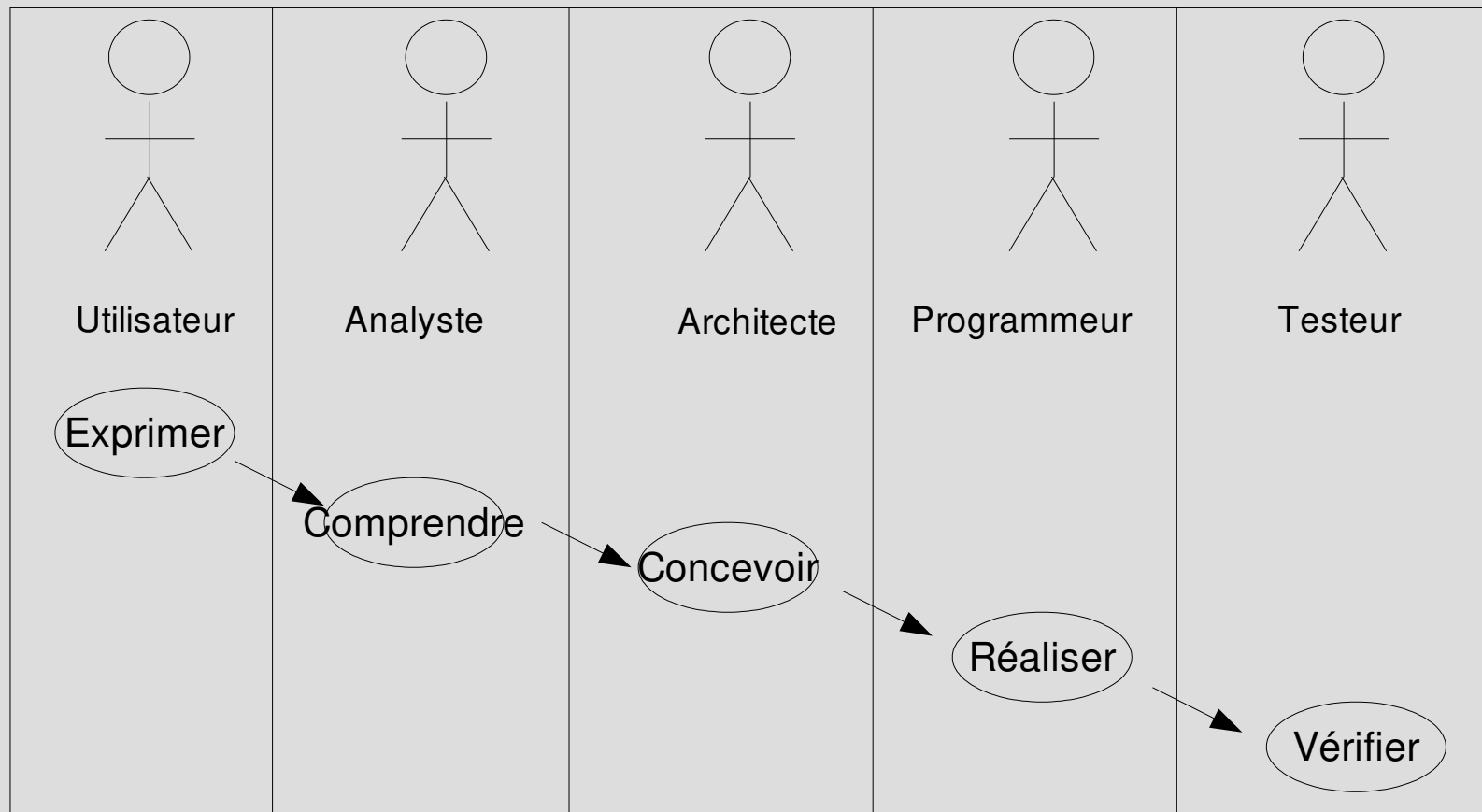


Cas d'utilisation

- Les Diagrammes de cas d'utilisation
 - Formalisation du cahier des charges
 - Evolutions du système
 - Centrés sur l'utilisateur
 - Expression simple
 - Permet le dialogue entre le client et le développeur
 - Point de départ du développement

Cas d'utilisation

- Les Diagrammes de cas d'utilisation





Cas d'utilisation



- Les acteurs
 - 1 acteur = plusieurs personnages
 - Ex : acheteurs
 - 1 personnage = plusieurs acteurs
 - Ex : boulanger, vendeur
 - Toute personne qui interagit avec le système



Cas d'utilisation



- Les acteurs
 - Acteurs principaux
 - Ex : client, dans le cas d'un distributeur de billets
 - Acteurs secondaires
 - Maintenance, tâches administratives
 - Matériel externe
 - Indispensable au domaine. Ex : imprimante pour le reçu
 - Autres systèmes
 - Ex : système bancaire



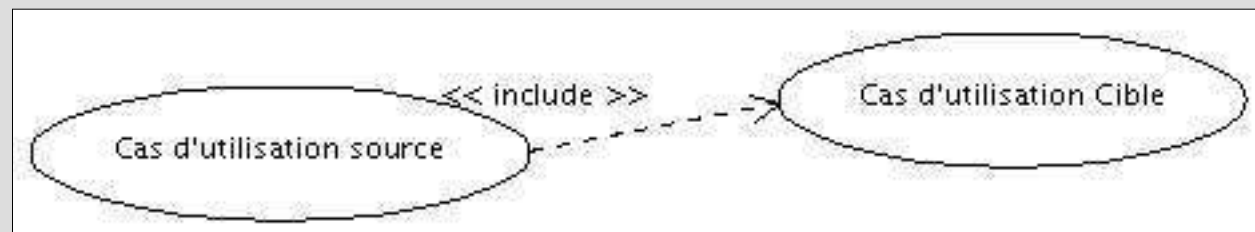
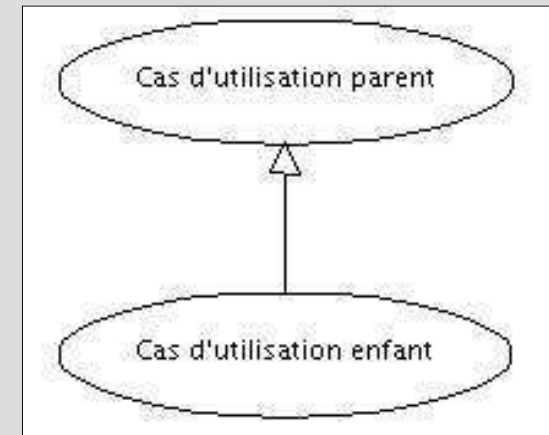
Cas d'utilisation



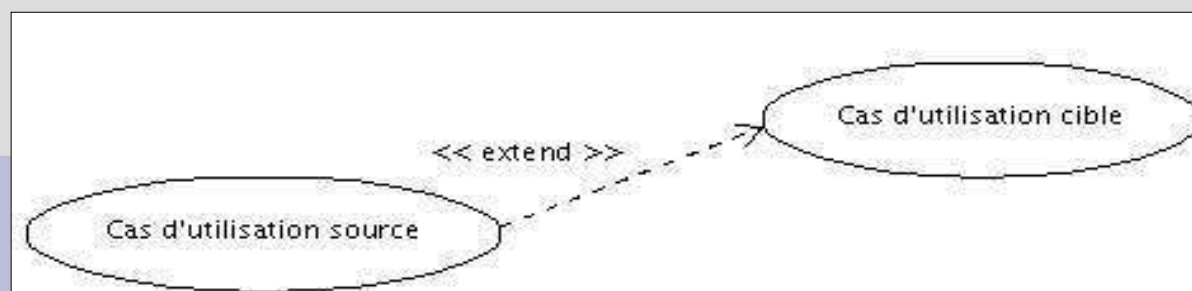
- Les cas d'utilisation
 - Stéréotype
 - <<acteur>>
 - Nom
 - Eleve
 - Propriété
 - Cours = “Genie Logiciel”

Cas d'utilisation

- Les cas d'utilisation
 - Relation de généralisation
 - Relation d'inclusion



- Relation d'extension





Cas d'utilisation



- Les cas d'utilisation
 - Représentent
 - Les tâches de l'acteur
 - Informations créées ou utilisées par l'acteur
 - Changements externes nécessaires au système
 - Doivent être simples
 - Possibilité de les fractionner
 - A partir de scénarii



Cas d'utilisation



- Les cas d'utilisation
 - Analyse
 - Pas conception
 - 'Build the right system'
 - Pas 'Build the system right'
 - Passage à une vue Objet indispensable
 - Pour la réalisation du système



Sommaire



- Les Diagrammes UML
 - Niveaux de modèles
 - Éléments communs aux diagrammes
 - Diagrammes de cas d'utilisation
 - **Diagrammes de Classes**
 - Diagrammes d'Objets
 - Diagrammes de Séquence



Sommaire



- **UML : Diagrammes de Classes**
 - **Présentation**
 - Classes
 - Associations
 - Autres types de classes
 - Exemples



Diagrammes de Classes



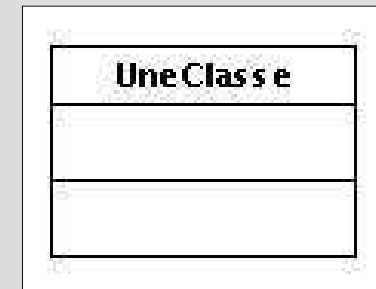
- Présentation
 - Structure statique des systèmes
 - Contient
 - Classes
 - Relations entre ces classes
 - Interfaces
 - Packages

Diagrammes de Classes

- Présentation

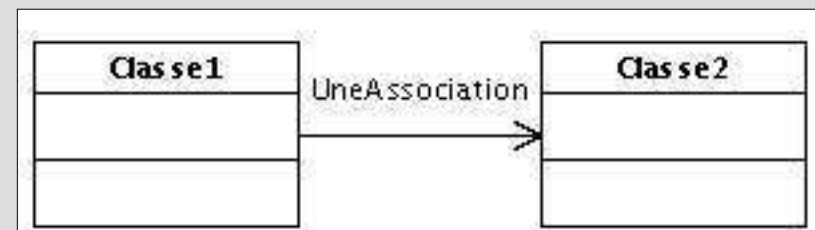
- Classe

- Instanciée par des objets
 - Unique dans un package
 - Syntaxe : 'nomPackage :: nomClasse'
 - Stéréotype, propriétés



- Associations

- Instanciées par des liens

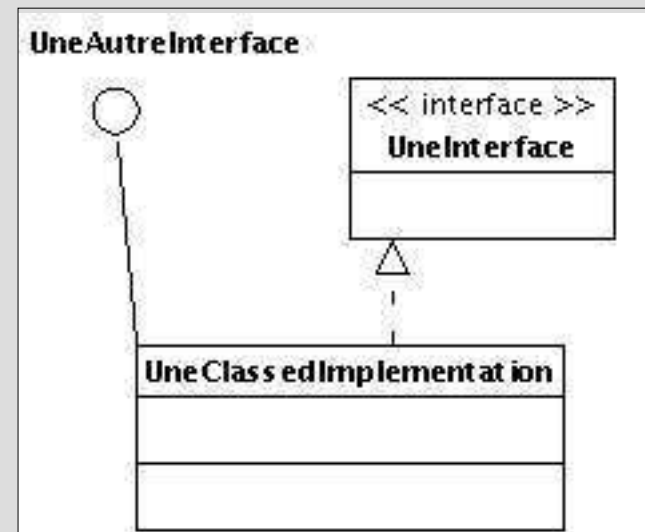


Diagrammes de Classes

- Présentation

- Interface

- 'Vue totale ou partielle sur un ensemble de services'
 - Descripteur des opérations
 - Sans code
 - Pas d'attribut
 - Pas d'association

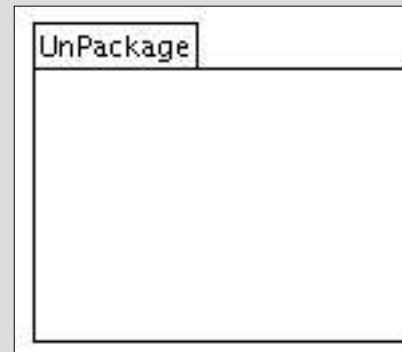




Diagrammes de Classes



- Présentation
 - Package





Sommaire



- **UML : Diagrammes de Classes**
 - Présentation
 - **Classes**
 - Associations
 - Autres types de classes
 - Exemples



Sommaire



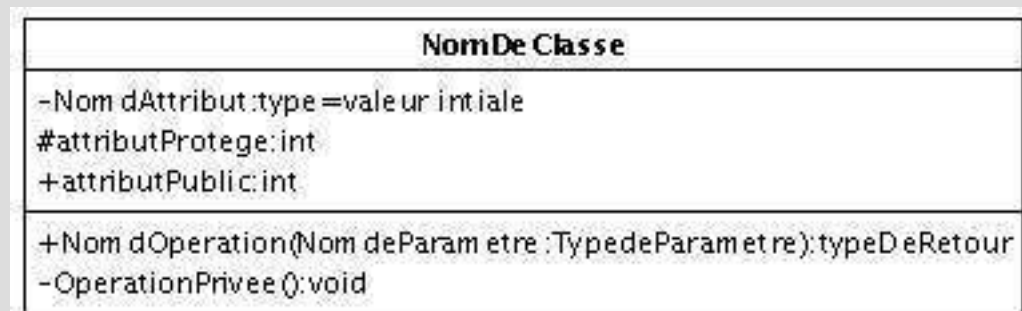
- **UML : Diagrammes de Classes : Classes**
 - **Attributs et Opérations**
 - **Classes paramétrables**
 - **Stéréotypes**
 - **Mots Clés**



Diagrammes de Classes



- Classes : Attributs et Opérations
 - Les compartiments
 - Nom de la classe
 - Attributs
 - Opérations





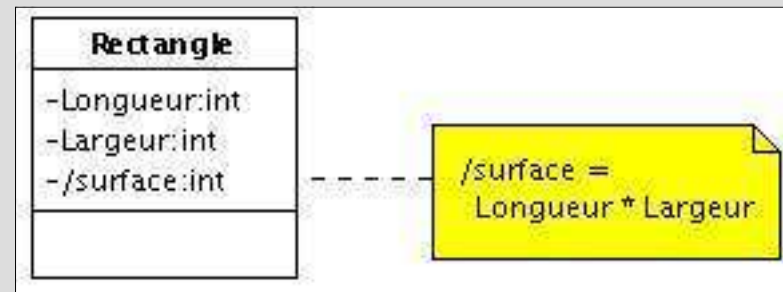
Diagrammes de Classes



- Classes : Attributs et Opérations
 - Attributs
 - *Ce qu'est la classe*
 - Nom : type = valeur initiale
 - Test : boolean = false
 - Peut être constant
 - 'const'
 - Représentation par la composition
 - Voir Associations
 - Représentation implicite

Diagrammes de Classes

- Classes : Attributs et Opérations
 - Attributs
 - Attributs dérivés
 - '/surface'





Diagrammes de Classes



- Classes : Attributs et Opérations
 - Opérations
 - *Ce que fait la classe*
 - Nom (nom de paramètre : type de paramètre) : type de Retour
 - multiplier(a : int, b : int) : int
 - Propriétés
 - Direction des arguments des opérations
 - Types d'opération



Diagrammes de Classes



- Classes : Attributs et Opérations
 - Opérations
 - Propriétés
 - Directions d'un argument d'une opération
 - In – argument en entrée, non modifié par l'opération
 - Out – argument en sortie, récupération d'information
 - Inout – argument passé à l'opération et modifiable



Diagrammes de Classes



- Classes : Attributs et Opérations
 - Opérations
 - Propriétés
 - Des Opérations
 - {request} – pas de modification de l'état de l'instance appelée
 - {abstract} – opération non implémentée
 - {final} – opération qui ne peut pas être redéfinie dans une sous-classe
 - {isRoot} – racine, opération définie pour la première fois dans la hiérarchie des classes



Diagrammes de Classes



- Classes : Attributs et Opérations
 - Visibilité
 - Pour attributs et opérations
 - + public
 - Accessible par tous les objets (dans et hors de la classe)
 - # protected
 - Accessible seulement par la classe et les sous-classes
 - - private
 - Accessible seulement par les objets de la classe



Diagrammes de Classes



- Classes : Attributs et Opérations
 - Compartiments supplémentaires
 - En plus du Nom de Classe, Attributs, Opérations
 - Responsabilité de la classe
 - Evènements générés
 - Représenté aussi par une relation de dépendance stéréotypée
<<envoi>>
 - Exceptions acceptées



Sommaire



- **UML : Diagrammes de Classes : Classes**
 - Attributs et Opérations
 - **Classes paramétrables**
 - Stéréotypes
 - Mots Clés



Diagrammes de Classes



- Classes : Classes paramétrables
 - = classes templates
 - Patron de classe : attributs, valeurs par défaut, opérations
 - Langages Eiffel, C++
 - Utilisé en conception détaillée

Paramètre
formel
Classe paramétrable

Elément de
la table
Table génétique



Sommaire



- **UML : Diagrammes de Classes : Classes**
 - **Attributs et Opérations**
 - **Classes paramétrables**
 - **Stéréotypes**
 - **Mots Clés**



Diagrammes de Classes



- Classes : Stéréotypes (1)
 - Définition
 - Extension du langage UML
 - Précision sur le type
 - De Classe
 - D'Association

'Création d'un nouvel élément de modélisation par extension de la sémantique d'un élément du méta-modèle'



Diagrammes de Classes



- Classes : Stéréotypes (2)
 - Différents stéréotypes
 - <<classe implémentation>>
 - Dans un langage de programmation donné
 - <<énumération>>
 - Domaine de valeur d'un type
 - <<méta-classe>>
 - Classe d'une classe
 - <<power type>>
 - Représente un type
 - Ses instances sont des sous-types de ce type



Diagrammes de Classes



- Classes : Stéréotypes (3)
 - Différents stéréotypes
 - <<processus>>
 - Flux de contrôle lourd
 - <<thread>>
 - Flux de contrôle léger
 - <<type>>
 - Domaine d'objets
 - Opération sur ce domaine
 - <<utilitaire>>
 - Regroupement d'opérations (Ex : Math)
 - Pas instanciables



Sommaire



- **UML : Diagrammes de Classes : Classes**
 - **Attributs et Opérations**
 - **Classes paramétrables**
 - **Stéréotypes**
 - **Mots Clés**



Diagrammes de Classes



- Classes : Mots clés (1)
 - Différents des stéréotypes
 - Propre à un métamodèle donné
 - Pas partie intégrante des spécifications UML



Diagrammes de Classes



- Classes : Mots clés (2)
 - Exemples fréquents
 - <<acteur>>
 - Ensemble de rôles joués par un acteur dans le système
 - <<interface>>
 - Descripteurs des opérations, sans code
 - <<signal>>, <<exception>>
 - Communication entre les classes
 - Hors appel de méthode



Sommaire



- **UML : Diagrammes de Classes**
 - **Présentation**
 - **Classes**
 - **Associations**
 - **Autres types de classes**
 - **Exemples**



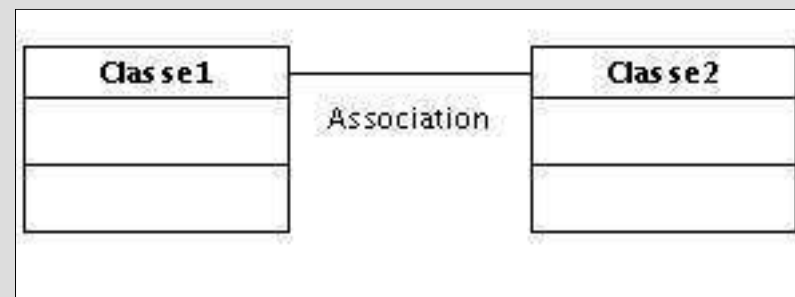
Diagrammes de Classes



- Associations
 - Relation structurelle entre deux classes d'objets
 - Durée de vie non négligeable
 - Par rapport aux objets qui instancient les classes concernées
 - Relie deux classificateurs
 - Classes, interfaces
 - Parfois plus : association représentée par une classe

Diagrammes de Classes

- Associations
 - Représentation



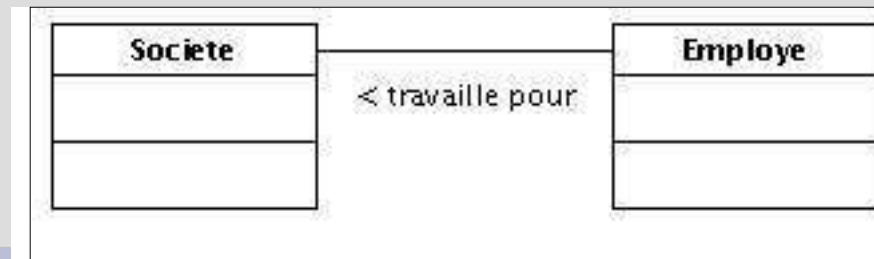
- Signification
 - On accède à la classe 1 depuis la classe 2
 - On accède à la classe 2 depuis la classe 1

Diagrammes de Classes

- Associations

- Nommage

- Souvent forme verbale
 - Active : travaille pour (Personne<->societe)
 - Passive : est employé par (Personne<->societe)
 - Lecture
 - Sens précisé par < ou >
 - De gauche à droite par défaut





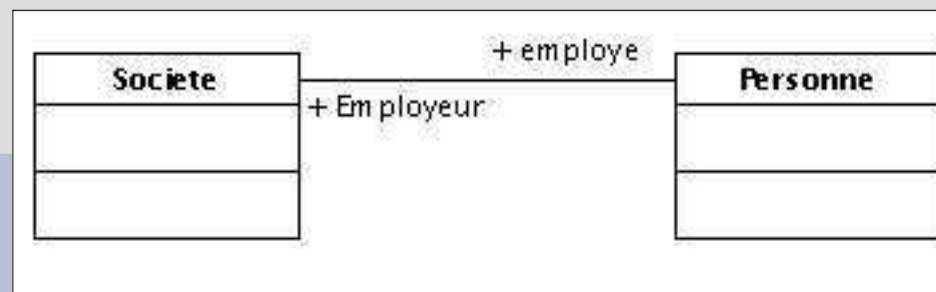
Diagrammes de Classes



- Associations

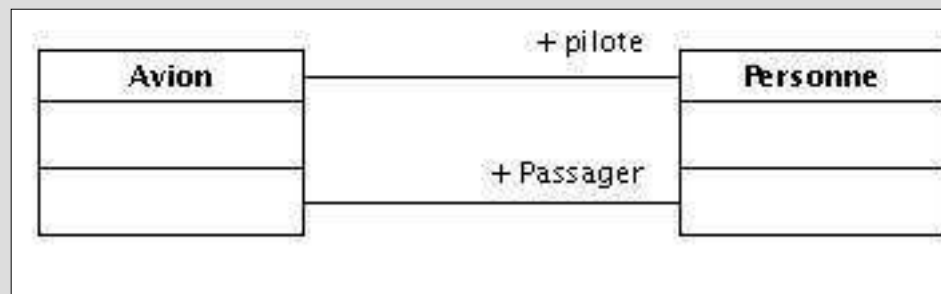
- Rôles

- Extrémité d'une association
 - Indication des rôles relatifs des deux classes reliées par association
 - Pseudo-attribut de la classe source
 - Ex : Employeur est un pseudo attribut de la classe Personne
 - Indication de visibilité
 - Public par défaut
 - Privé (-) ou protégé (#)



Diagrammes de Classes

- Associations
 - Associations multiples
 - Concepts distincts dans la classe source
 - Ici : dans Avion, attributs Pilote et Passagers
 - Pas les mêmes personnes
 - Sinon, erreur





Diagrammes de Classes



- Associations
 - Associations multiples

```
Public class Avion{  
  
    Personne pilote;  
    Personne passager1;  
    Personne passager2;  
  
    Personne getPilote(){  
        return pilote;  
    }  
  
    List getPassagers(){  
        return List.add(passager1, pas-  
sager2)  
    }  
}
```

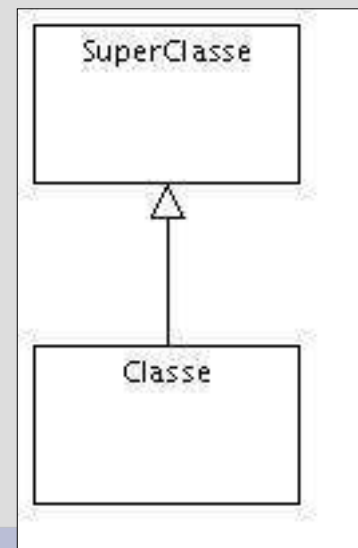


Diagrammes de Classes



- Associations
 - Types
 - Généralisation
 - Agrégation
 - Composition

- Associations
 - Généralisation
 - Une classe générique (super classe)
 - Une classe spécialisée (classe fille)





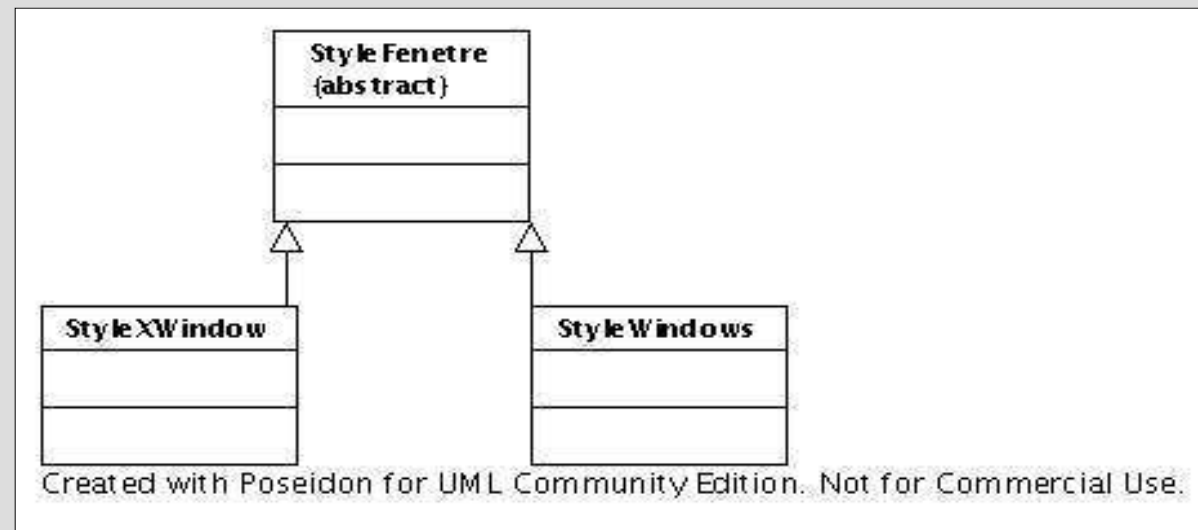
Diagrammes de Classes



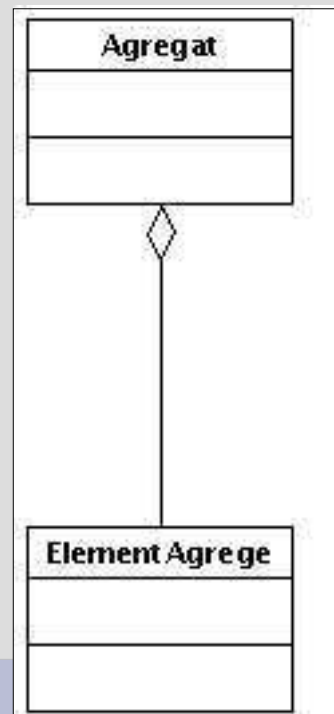
- Associations
 - Généralisation
 - Exemple

```
public class SuperClasse{  
}  
  
public class Classe extends SupperClasse{  
}
```

- Associations
 - Généralisation
 - Classes abstraites



- Associations
 - Agrégation
 - Les Elements existent indépendamment de l'ensemble



Diagrammes de Classes

- Associations

- Agrégation

- Modélisation d'une contrainte d'intégrité
 - Propagation de valeur d'une classe vers une autre
 - Propagation d'action d'une classe vers une autre
 - Exemple
 - personne propriétaire d'un immeuble
 - Propriétaire multiple: co-propriétaires





Diagrammes de Classes

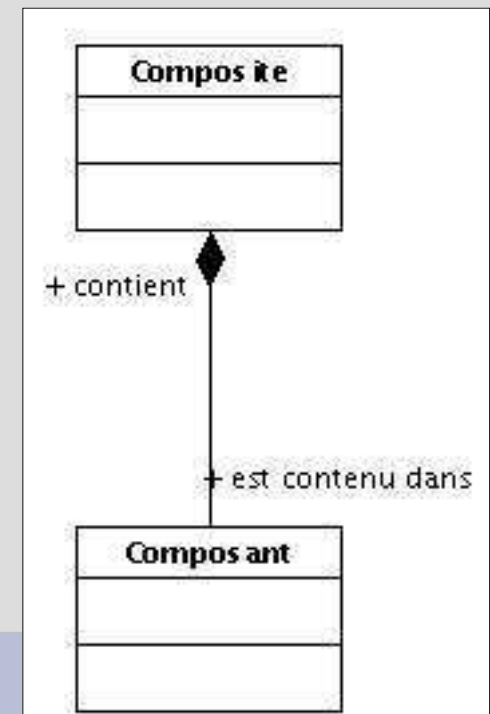


- Associations

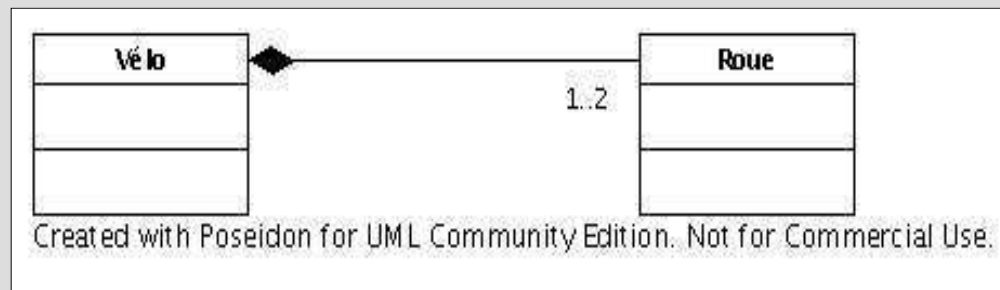
```
public class Immeuble {  
    /**  
     * @poseidon-type Personne  
     */  
    public java.util.Collection Proprietaire = new java.util.TreeSet();  
}
```

```
public class Personne {  
    /**  
     * @poseidon-type Immeuble  
     */  
    public java.util.Collection immeuble = new java.util.TreeSet();  
}
```


- Associations
 - Composition
 - Les éléments sont spécifiques à l'ensemble : ne peuvent pas être partagés (ex : roue de vélo)
 - L'ensemble n'existe que par la présence des éléments
 - Et disparaît avec le dernier élément



- Associations
 - Composition
 - Cas particulier d'agrégation
 - Couplage plus fort
 - Durée de vie identique pour le composant et le composite





Diagrammes de Classes



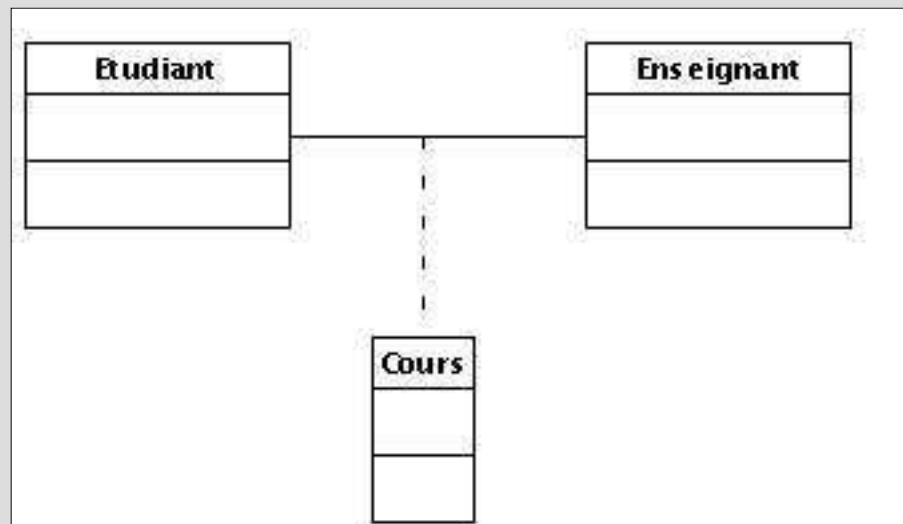
- Associations
 - Composition

```
public class Velo {  
    /**  
     * @poseidon-type Roue  
     */  
    public java.util.Collection roue = new java.util.TreeSet();  
}
```

```
public class Roue {  
    /**  
     * <p></p>  
     */  
    public Vélo vélo;  
}
```

Diagrammes de Classes

- Associations
 - Classe d'association
 - L'association peut être manipulée

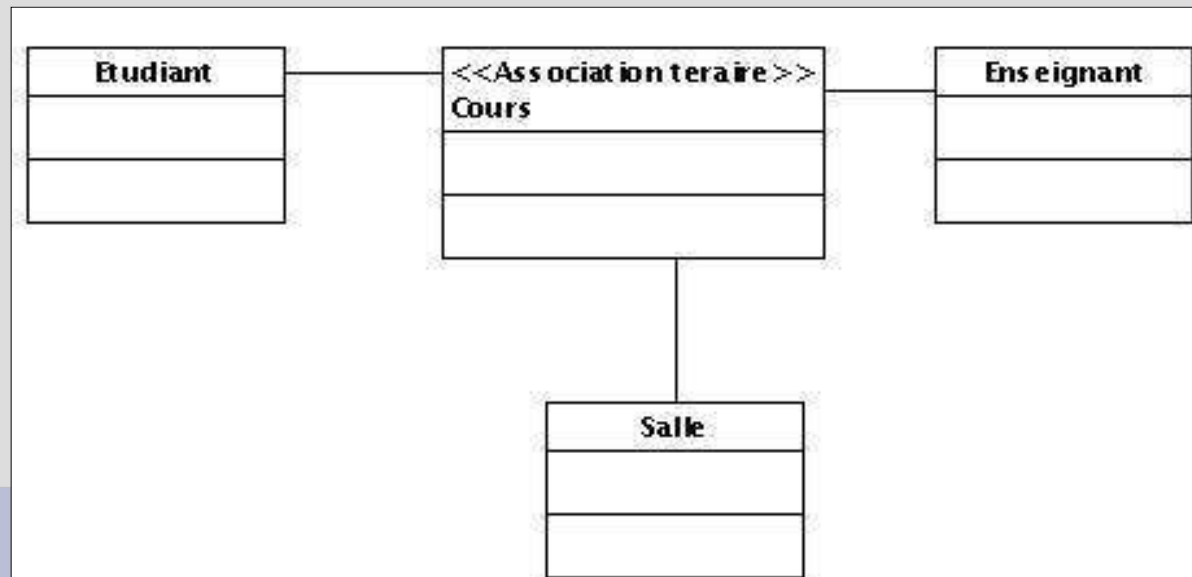


Diagrammes de Classes

- Associations

- Arité

- Associations binaires - classiques
 - Associations n-aires
 - Parfois représentées par un losange

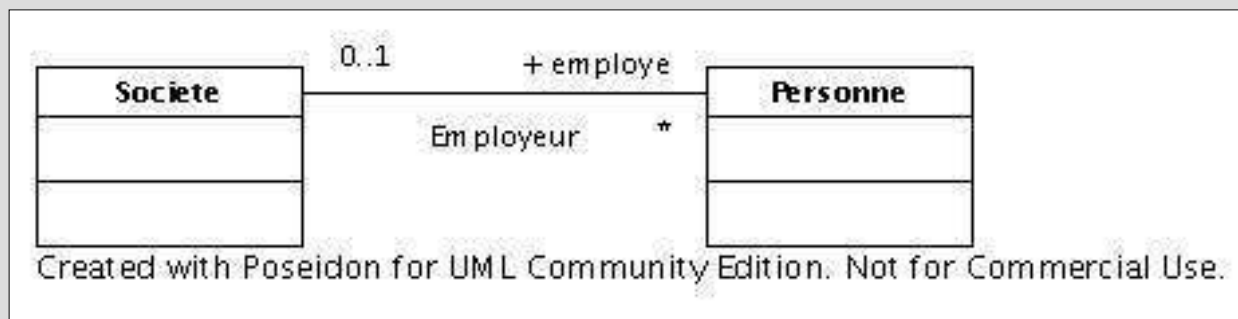


Diagrammes de Classes

- Associations

- Multiplicité

- Contraintes liées au domaine d'application
 - Valable pendant toute la vie de l'objet
 - Pas d'influence sur l'ordre de création des objets (associations simples)





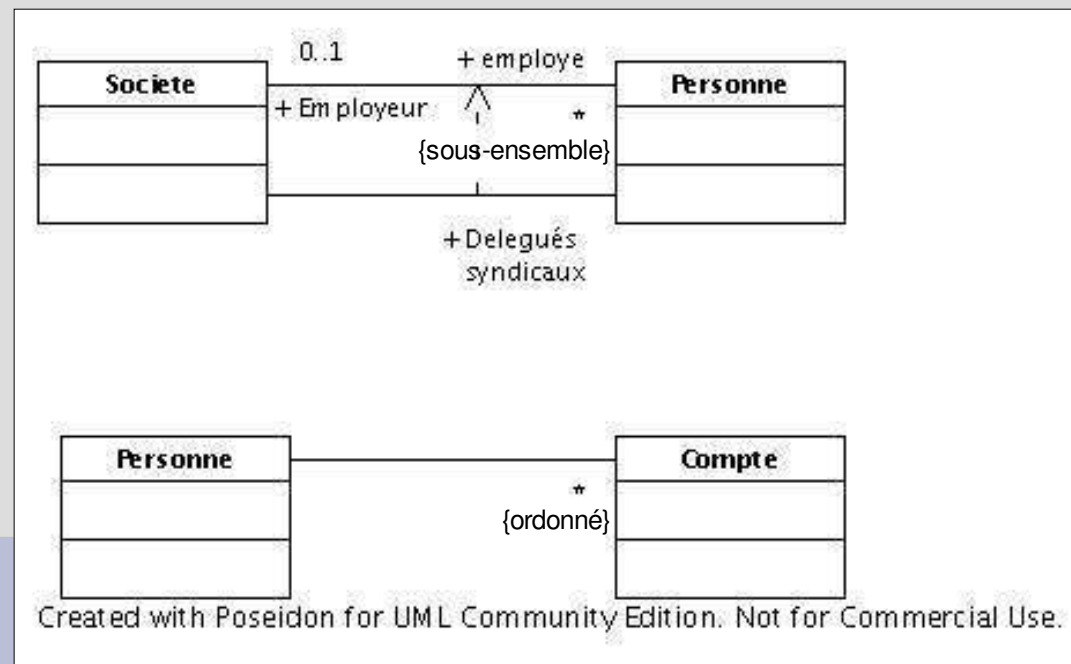
Diagrammes de Classes



- Associations
 - Multiplicité
 - possibilités

1	Un seul
0..1	Zéro ou un
N	N (entier naturel)
M..N	De M à N (entiers naturels)
*	De zéro à plusieurs
0..*	De zéro à plusieurs
1..*	D'un à plusieurs

- Associations
 - Contraintes : exemples
 - Peuvent être également exprimer par le langage OCL
 - Object Constraint Language

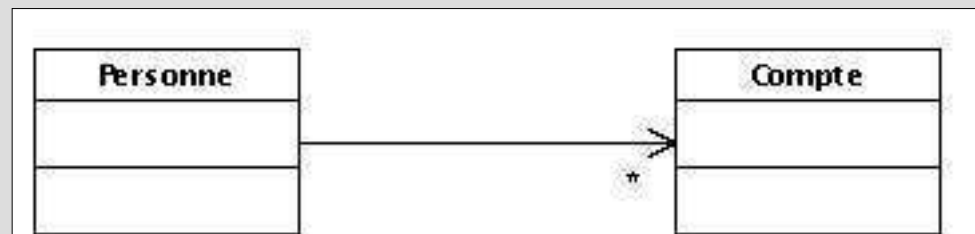


Diagrammes de Classes

- Associations

- Navigabilité

- Possibilité d'accès d'une classe à l'autre
 - Association orientée
 - Compte est un attribut de Personne



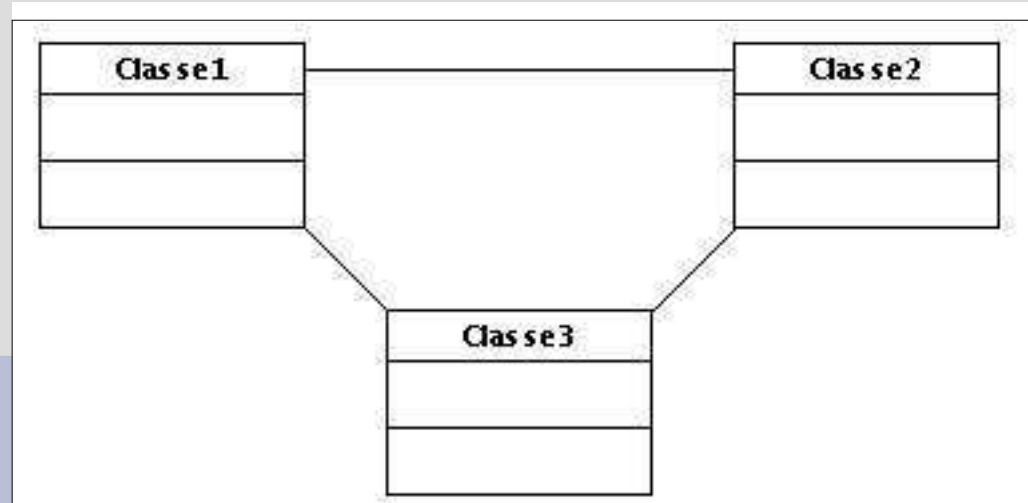
Diagrammes de Classes

- Associations

- Navigabilité

- Association bidirectionnelle

- Une modification dans une classe est répercutée dans la classe associée
 - Complexe si association en chaîne
 - Erreur si boucle d'associations



- Associations
 - Dépendances
 - 'Associations légères'



- Liaison
 - <<lie>>
- Permission
 - <<ami>>
- Utilisation
 - <<utilise>>
 - <<instancie>>
 - <<appelle>>
- Abstraction
 - <<crée>>
 - <<dérive>>
 - <<raffine>>
 - <<réalise>>
 - <<trace>>



Sommaire



- **UML : Diagrammes de Classes**
 - Présentation
 - Classes
 - Associations
 - **Autres types de classes**
 - Exemples



Sommaire



- Autres Classes
 - Classes actives
 - Réification de flots de contrôle
 - = classe stéréotypée
 - <<thread>> ou
 - <<processus>>
 - Émission et réception de signaux, d'évènements

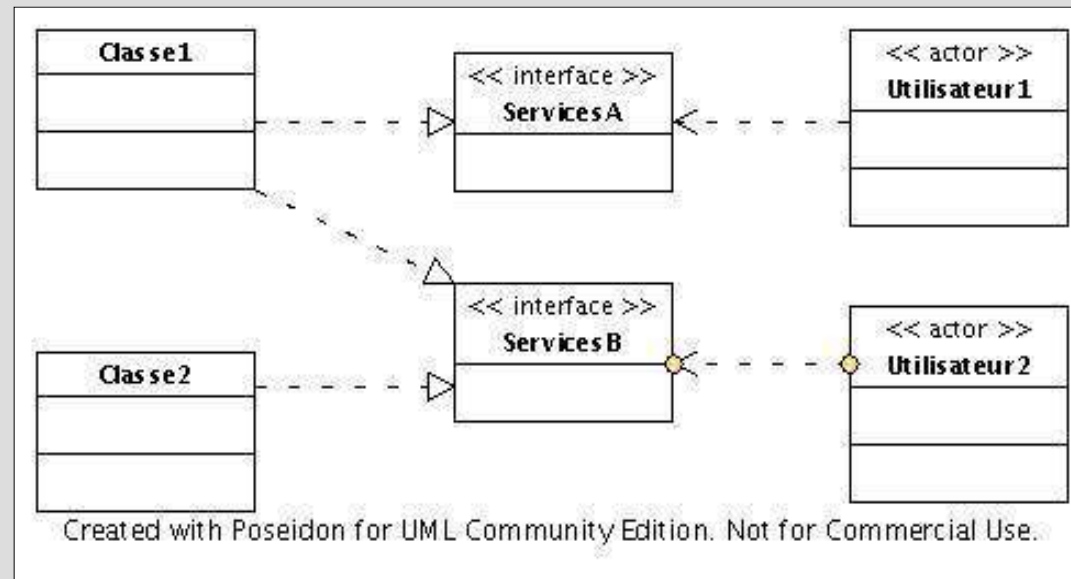


Sommaire



- **UML : Diagrammes de Classes**
 - Présentation
 - Classes
 - Associations
 - Autres types de classes
 - **Exemples**

- Exemples
 - Interfaces





Sommaire



- Les Diagrammes UML
 - Niveaux de modèles
 - Éléments communs aux diagrammes
 - Diagrammes de cas d'utilisation
 - Diagrammes de Classes
 - **Diagrammes d'Objets**
 - Diagrammes de séquence



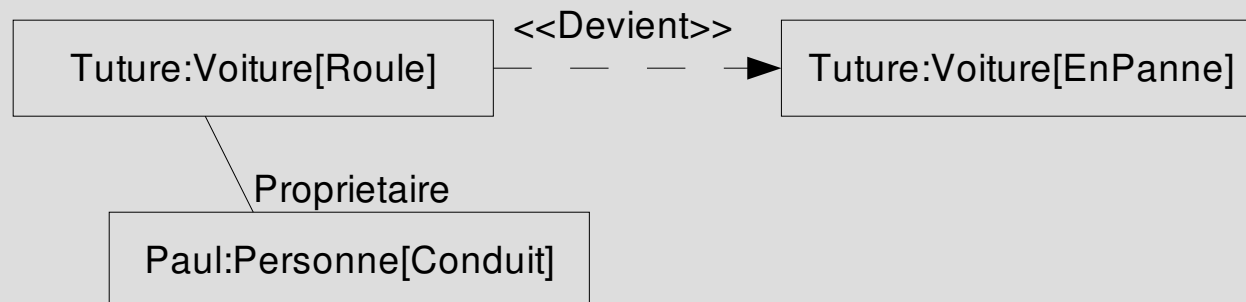
Sommaire

- **UML : Diagrammes d'Objet**
 - = Diagrammes d'instances
 - Objets et liens
 - Structure statique
 - Permet d'identifier les objets présents
 - Avant de réaliser le diagramme de classe

Nom de l'objet:Classe[Etat]

Sommaire

- UML : Diagrammes d'Objet
 - Exemple





Sommaire



- Les Diagrammes UML
 - Niveaux de modèles
 - Éléments communs aux diagrammes
 - Diagrammes de cas d'utilisation
 - Diagrammes de Classes
 - Diagrammes d'Objets
 - **Diagrammes de Séquence**



Diagrammes de Séquence



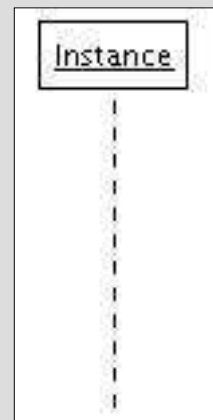
- Les Diagrammes de séquence
 - Interactions entre objets
 - Voir diagrammes de collaboration
 - Représentation des interactions dans le temps
 - Scénarii complexes
 - Avec un nombre limité d'objets

Diagrammes de Séquence

- Les Diagrammes de séquence

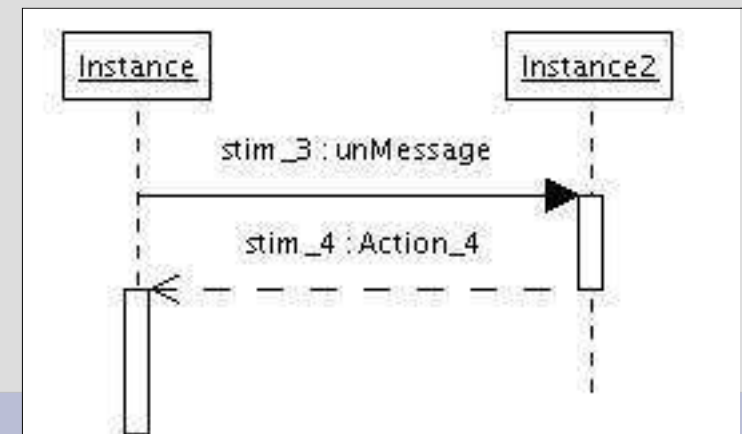
- Instances

- Objet
- Ligne de vie



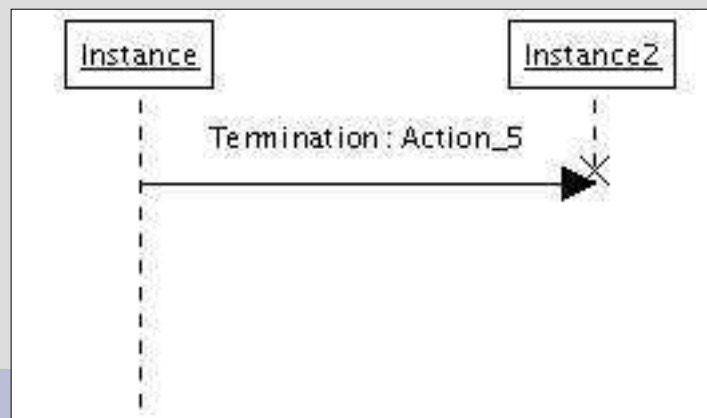
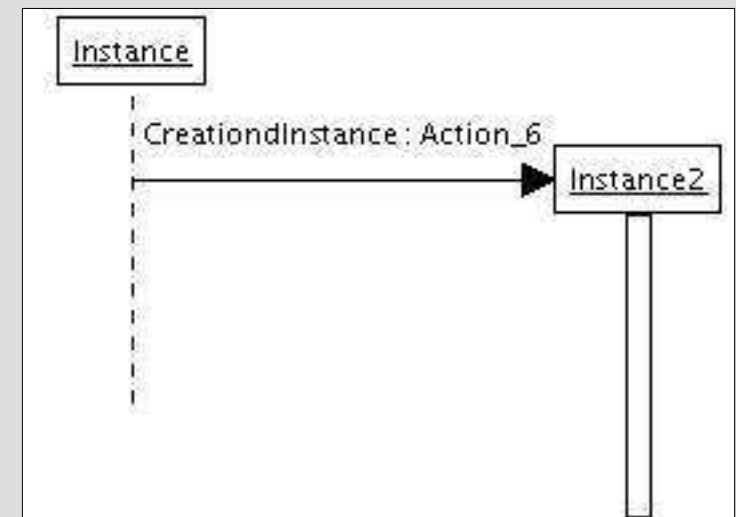
- Message

- Activation de l'instance émetrice
- Message
 - Activation de l'instance receptrice
- Réponse
 - Activation de l'instance émetrice



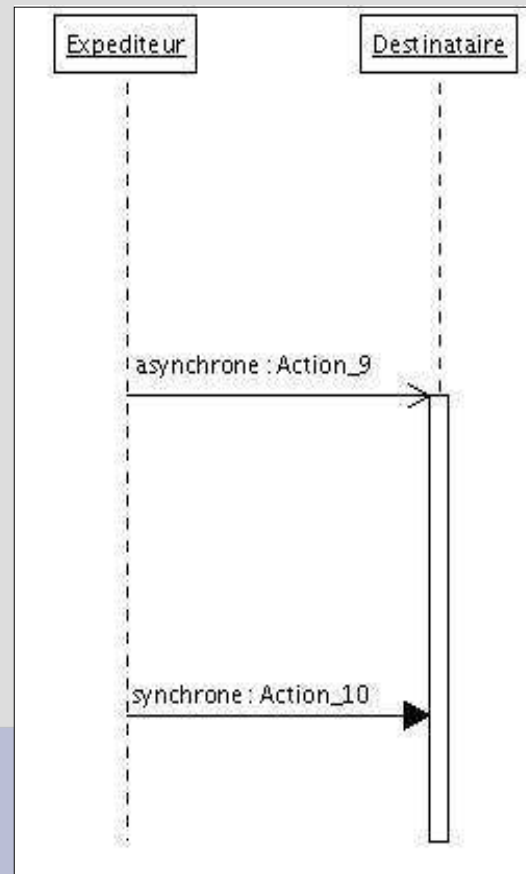
Diagrammes de Séquence

- Les Diagrammes de séquence
 - Message de Creation
 - Message de Termination



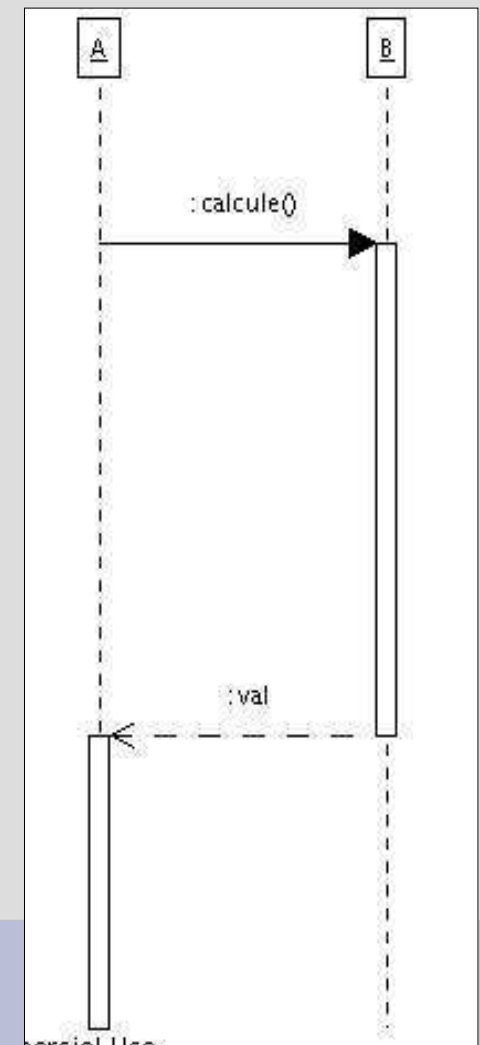
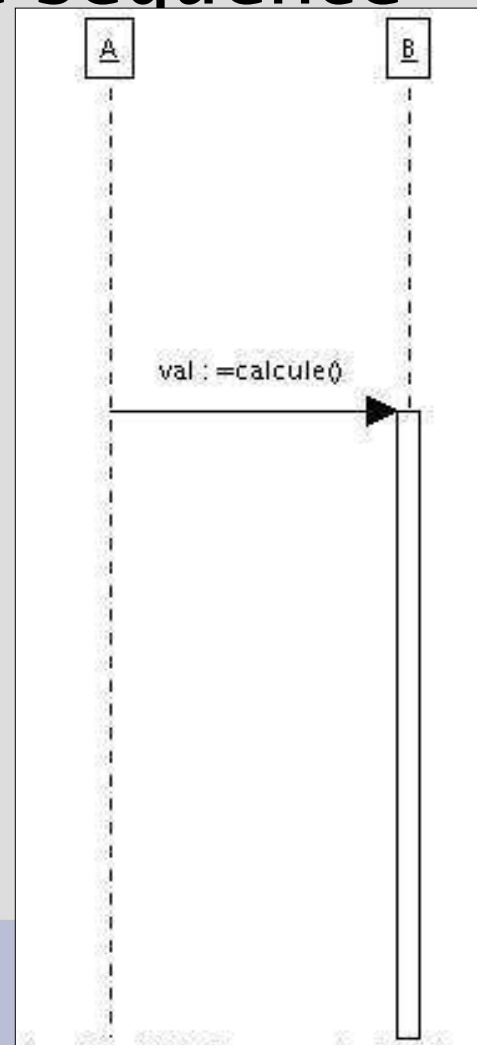
Diagrammes de Séquence

- Les Diagrammes de séquence
 - Catégories de messages
 - Synchrone
 - Asynchrone



Diagrammes de Séquence

- Les Diagrammes de séquence
 - Appel de procédure
 - Plusieurs représentations





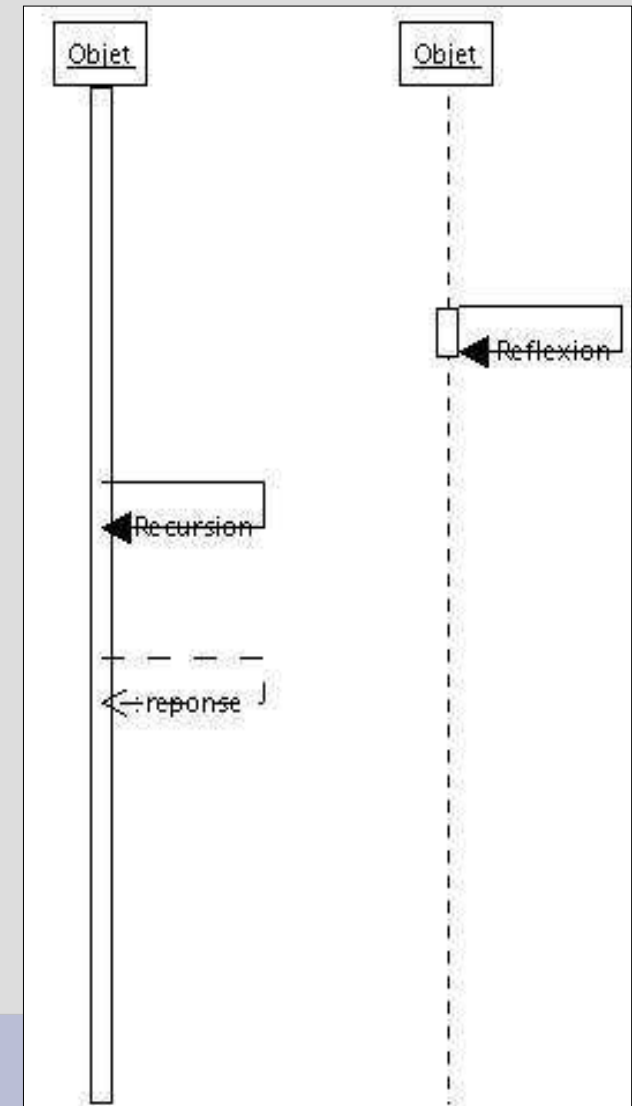
Diagrammes de Séquence



- Les Diagrammes de séquence
 - Catégories d'envoi de message
 - Appel d'un objet à lui-même
 - Flot de contrôle à plat
 - Appel de procédure
 - = flot de contrôle emboité
 - Retour d'appel de procédure

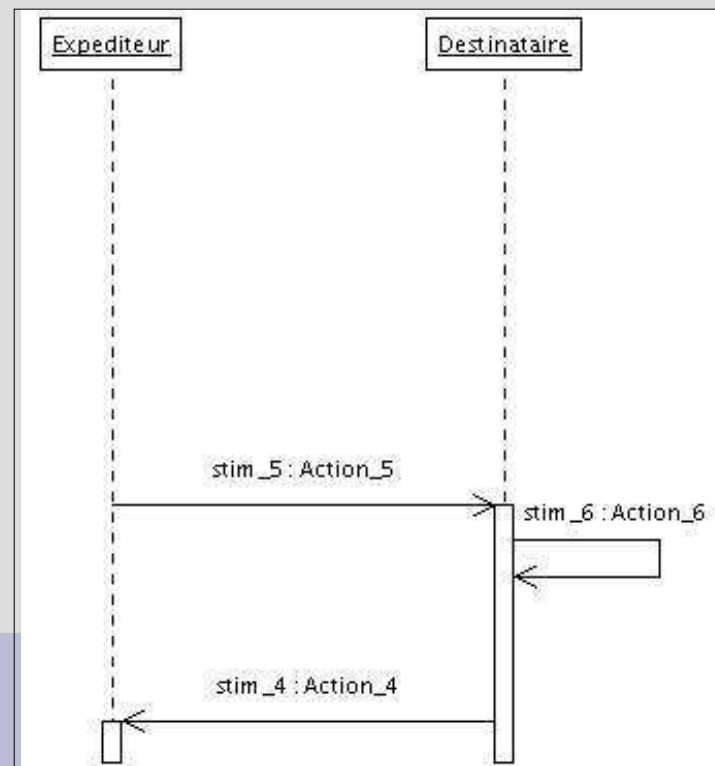
Diagrammes de Séquence

- Les Diagrammes de séquence
 - Catégories d'envoi de message
 - Appel d'un objet à lui-même



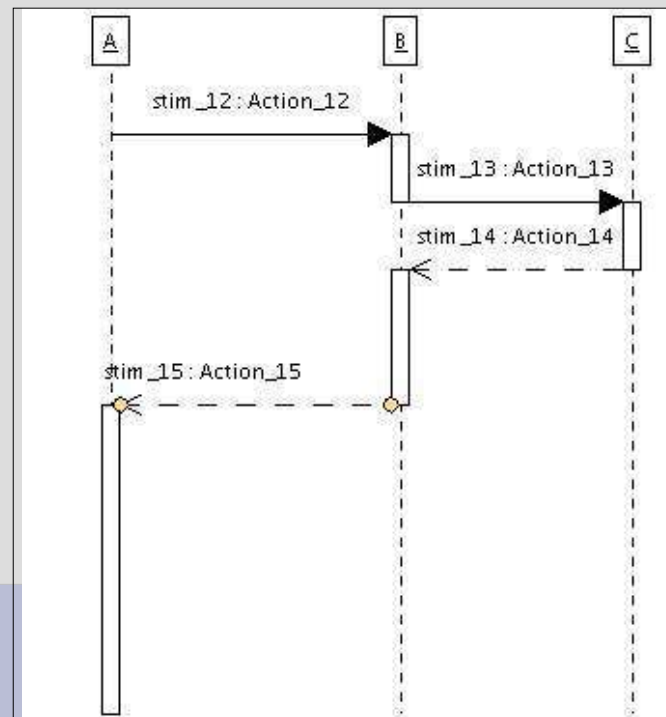
Diagrammes de Séquence

- Les Diagrammes de séquence
 - Catégories d'envoi de message
 - Flot de contrôle à plat



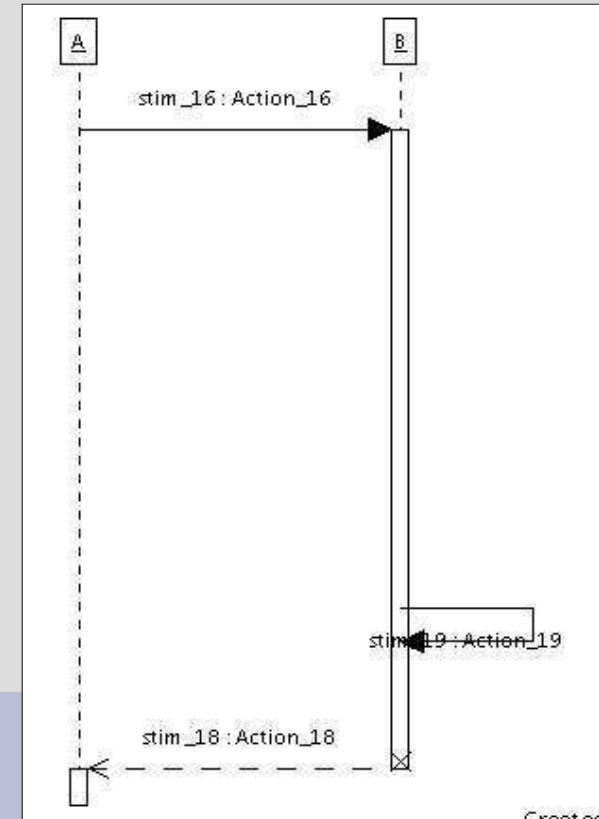
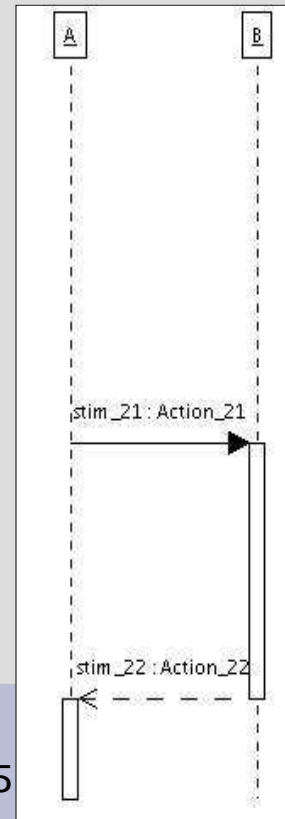
Diagrammes de Séquence

- Les Diagrammes de séquence
 - Catégories d'envoi de message
 - Appel de procédure
 - = flot de contrôle emboîté



Diagrammes de Séquence

- Les Diagrammes de séquence
 - Catégories d'envoi de message
 - Retour d'appel de procédure
 - Fin d'activation d'objet
 - != fin de vie d'objet





Diagrammes UML



- Bilan
 - Analyse : diagrammes de cas d'usage
 - Conception : diagramme de classes
 - Implémentation : classes et séquences

