

**Rapport**

**TP UML**

Thomas NICOLAS

**Sommaire**

1. **Objectif du rapport**
2. **Objectif des TP**
   1. *TD cas d’utilisation / cas d’utilisation avancé*
   2. *TD Modélisation du domaine*
   3. *TD Des classes au code*
   4. *TD Diagrammes de séquence*
   5. *TD Relation entre classes et code*
   6. *TD Associations de classes*

**TD 1**

**Cas d’utilisation**

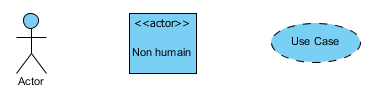
1. **Objectifs du TP**

Les diagrammes de cas d’utilisation servent à représenter les acteurs et leur rôle dans le système.

Dans ce TP les exercices avaient pour but d’identifier, en lisant le sujet, les mots clé qui représenteraient un Acteur ou un Use Case.

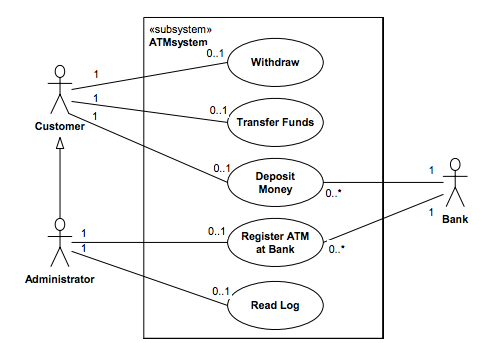
Les pronoms ou noms qui reviennent régulièrement donneront lieu à des acteurs, et les verbes d’action précédés du nom d’un acteur seront des use case, reliés avec cet acteur.

Dans le diagramme un acteur peut être représenté de deux manières :



Et un use case par un rond : 

Un diagramme de use case complet ressemble au final à ça :



**TD 2**

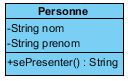
**Modélisation du domaine**

1. **Objectifs du TP**

Un diagramme de classe permet de modéliser tous les concepts dont est fait le système. Il permet également de prévoir la structure du code Java du programme.

Une classe représentée correspond à un concept identifié dans le Cahier des charges. La difficulté est de savoir ce qui sera représenté par une classe ou par un attribut de classe.

La représentation d’une classe est simple :



En haut du cadre se trouve le nom de classe. La partie du milieu contient les attributs de la classe et la dernière partie les méthodes de la classe.

**TD 3**

**Des classes au code**

**1.Objectifs du TP**

Dans le diagramme de classe, il y a des associations entre les différentes classes.

On en différencie 4 :

* L’héritage, ou toute classe héritée est une spécialisation de la classe mère
* La composition, ou une classe est faite d’une ou plusieurs classes, comme attributs.
* L’agrégation, comme pour la composition, une classe possède des objets d’une autre classe comme attributs, mais leur cycle de vie est lié. Si l’un disparait, les autres aussi.
* L’association, c’est le lien le plus faible, chacune des classes liée possède un attribut du type de l’autre. Le lien peut également être à sens unique.

**TD 4**

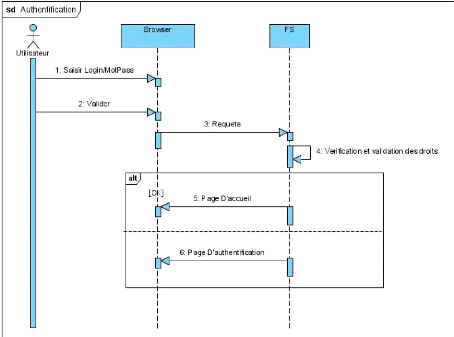
**Diagramme de séquence**

1. **Objectifs du TP**

Le but d’un diagramme de séquence est de représenter les actions nécessaires pour arriver au but voulu (au use case associé).

Dans le diagramme, chaque acteur intervenant au cours de l’action est représenté.

Voici un exemple de diagramme de séquence :



Les « personnages » en haut représentent les acteurs.

La ligne verticale sous les acteurs représente l’évolution du temps et chaque bloc sur la ligne correspond à une action de l’acteur et représente la durée de cette action.

Une flèche entre deux acteurs correspond à une action et une flèche pointillée est un « feedback », une réponse.

**TD 5**

**Passer du code aux classes**

**1.Objectifs du TP**

Certains éléments du code Java donnent des indications sur le modèle de classe associée.

Voici un exemple de classe Java :



Dans la déclaration de la classe, le mot clé « extends » correspond à un héritage. Pour les autres associations, elles sont nécessaires dans le cas ou un attribut d’une classe est un objet (ou tableau d’objets) d’une autre classe.

La différence entre association et agrégation ou composition dépend de l’interprétation du programmeur. Il devra décider si oui ou non les cycles de vie des deux classes sont confondus.