



Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Abderrahmane Mira Bejaia

Faculté des sciences exactes

Département d'informatique

Cahier d'exercices, Génie Logiciel

Niveau : 3^{ème} année Licence

Spécialité : Systèmes Informatiques

Etabli par Dr HAMZA Lamia

2022/2023

Table des matières

Introduction	1
Guide de référence	3
Partie 1 : Introduction au Génie Logiciel	4
Partie 2 : Notation UML	9
Partie 3 : Méthodes de test logiciel	29
Série de TD N° 1	31
Corrigé de la série de TD N° 1	32
Série de TD N° 2	35
Corrigé de la série TD N° 2	38
Série de TD N° 3	46
Corrigé de la série TD N° 3	47
Série de TD N° 4	52
Corrigé de la série de TD N° 4	55
Série de TD N° 5	61
Corrigé de la série TD N° 5	63
Série de TD N° 6	67
Corrigé de la série TD N° 6	69
Série de TD N° 7	72
Corrigé de la série TD N° 7	74
Exercices de perfectionnement	77
Épreuve de Moyenne Durée 1	95
Corrigé de l'EMD 1	97

Épreuve de Moyenne Durée 2.....	101
Épreuve de Moyenne Durée 3.....	105
Corrigé de l'EMD 3	107
Examen de Rattrapage.....	110
Conclusion.....	114
Bibliographie.....	115

Introduction

Ce document a pour objectif d'offrir aux étudiants licence informatique un ensemble d'exercices lui permettant de concevoir une application en utilisant la pensée orientée objet. Il débute par un guide de référence, conçu par moi-même. Suivi d'un ensemble de séries d'exercices avec leurs corrections, ainsi des explications accompagnant quelques solutions. Enfin, un ensemble d'exercices de perfectionnement et de sujets d'examen sont mis volontairement pour les tests de connaissances. Le contenu du document se résume comme suit :

Guide de référence : Il englobe trois parties ; La première partie présente en sommaire les cycles de vie du logiciel. La deuxième partie présente en simplifié la notation des principaux diagrammes du langage UML (Unified Modeling Language). La troisième partie résume les méthodes de tests de logiciels.

Série de TD 1 : Couvre les principaux concepts de Génie Logiciel (Qualité d'un logiciel, cycles de vie d'un logiciel, prototypage, etc.)

Série de TD 2 : Apprend à l'étudiant la façon de modéliser les besoins d'un logiciel en utilisant les diagrammes de cas d'utilisation.

Série de TD 3 : Enseigne à l'étudiant la maîtrise de la description textuelle d'un cas d'utilisation.

Série de TD 4 : Inculque la manière de concevoir un diagramme de classes.

Série de TD 5 : Fait découvrir la conception dynamique en utilisant les diagrammes de séquence et de communication.

Série de TD 6 : Englobe les diagrammes d'états-transitions et d'activités qui font parties de la conception dynamique.

Série de TD 7 : Initie la manière d'utiliser les jeux de tests.

Exercices de perfectionnement : Un ensemble d'exercices est présenté dans cette partie, la plupart d'entre eux ont fait l'objet d'examens antérieurs.

Sujets d'examen : Deux sujets d'examen avec corrigés et deux autres sujets sans corrigés sont exposés dans cette partie.

Il est à noter que le guide de référence permet à tous les informaticiens (étudiants, chefs de projet, enseignants) de comprendre les exercices de ce document. De plus, les séries d'exercices de ce document sont présentés dans un ordre pédagogique très constructif afin d'inculquer aux lecteurs la maîtrise d'une conception orientée objet. En outre, la résolution des exercices de ce document permettra d'entamer un projet de développement.

Guide de référence

Le guide présenté dans ce document est divisé en trois parties :

La partie 1, intitulée « **Introduction au Génie Logiciel** », couvre les principaux cycles de vie d'un logiciel.

La partie 2, intitulée « **Notation des diagrammes UML** », a pour but de comprendre les principales notations du langage UML (Unified Modeling Language).

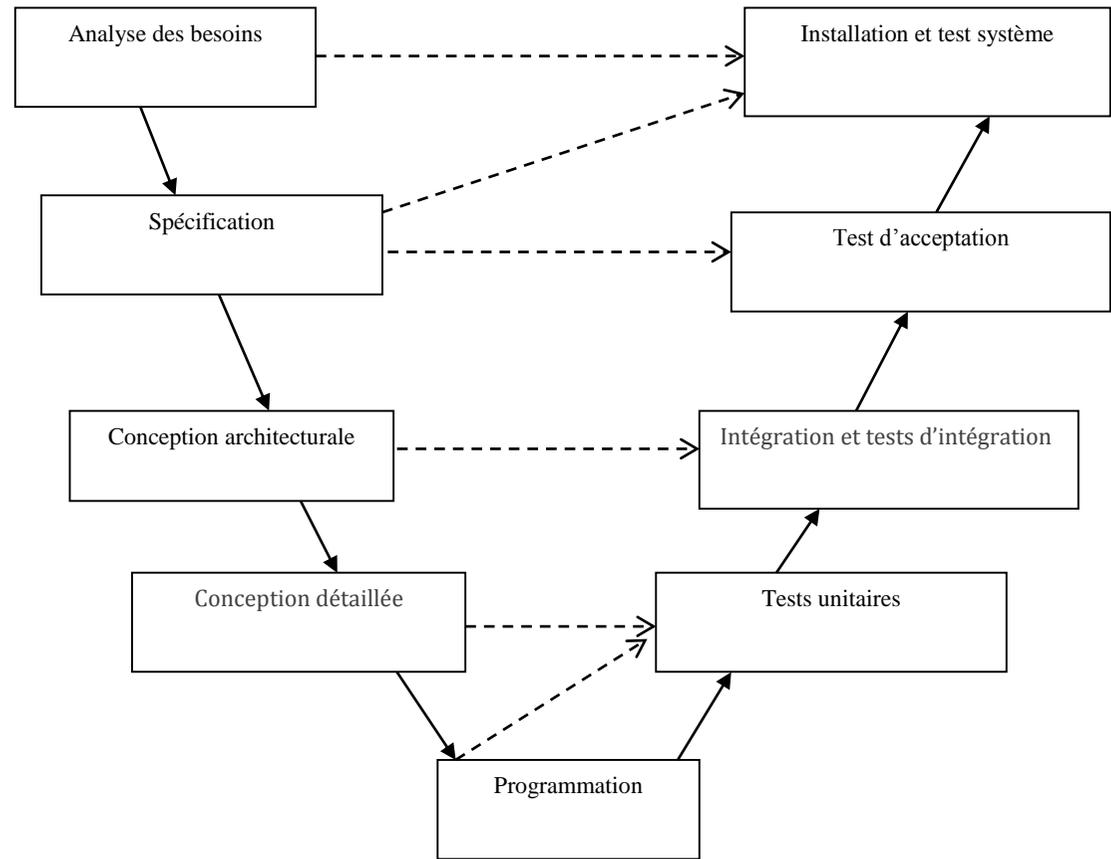
La partie 3, intitulée « **Test des logiciels** », permet de maîtriser la construction des jeux de tests.

Partie 1 : Introduction au Génie Logiciel

Définition du Génie Logiciel	Le génie logiciel est un domaine des « sciences de l'ingénieur » dont la finalité est la conception, la fabrication et la maintenance de systèmes logiciels complexes, sûrs et de qualité.	
Cycle de vie du logiciel	Le cycle de vie d'un logiciel est constitué de l'enchaînement des différentes activités nécessaires à son développement.	
Modèles de cycle de vie d'un logiciel	<p>Modèle en " Cascade " :</p> <p>Dans le modèle en cascade on convient d'avoir un certain nombre d'étapes.</p> <p>Une étape doit se terminer à une date donnée par la production de certains documents ou logiciels.</p>	<pre> graph TD A[Faisabilité] --> B[Analyse des besoins planification] B --> C[Conception du produit] C --> D[Conception détaillée] D --> E[Codage] E --> F[Intégration] F --> G[Installation] G --> H[Exploitation et Maintenance] B --> A C --> B D --> C E --> D F --> E G --> F H --> G </pre>

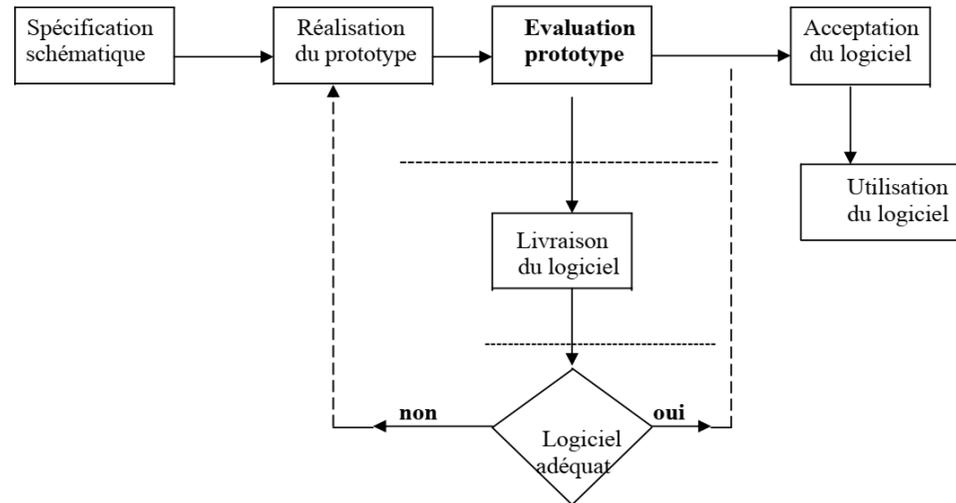
Modèle en " V " :

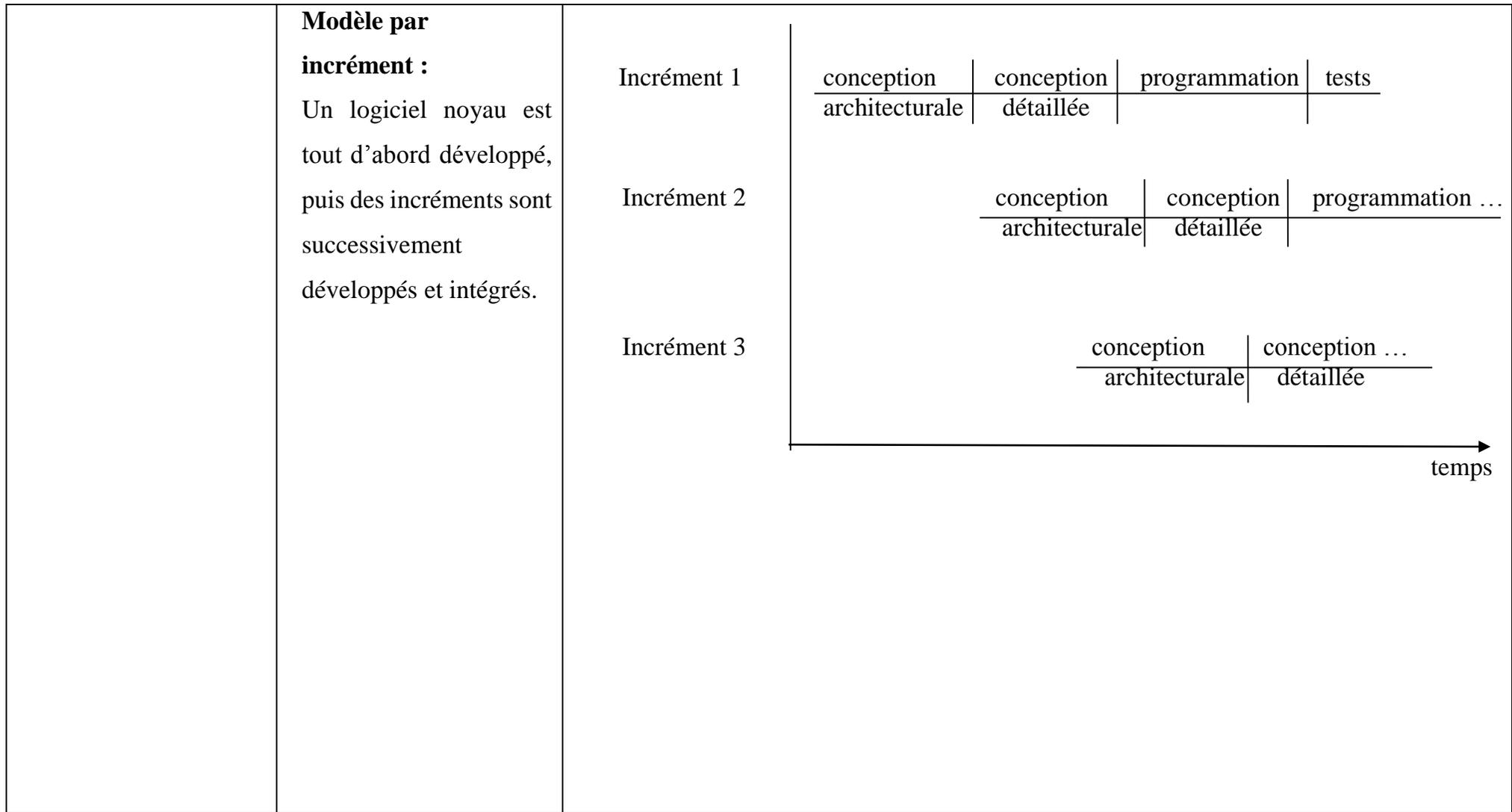
Le modèle en V montre non seulement l'enchaînement des phases successives, mais aussi les relations logiques entre phases plus éloignées à travers des liens **parallèles** exprimant les plans de tests de chaque étape.



**Modèle par
prototypage :**

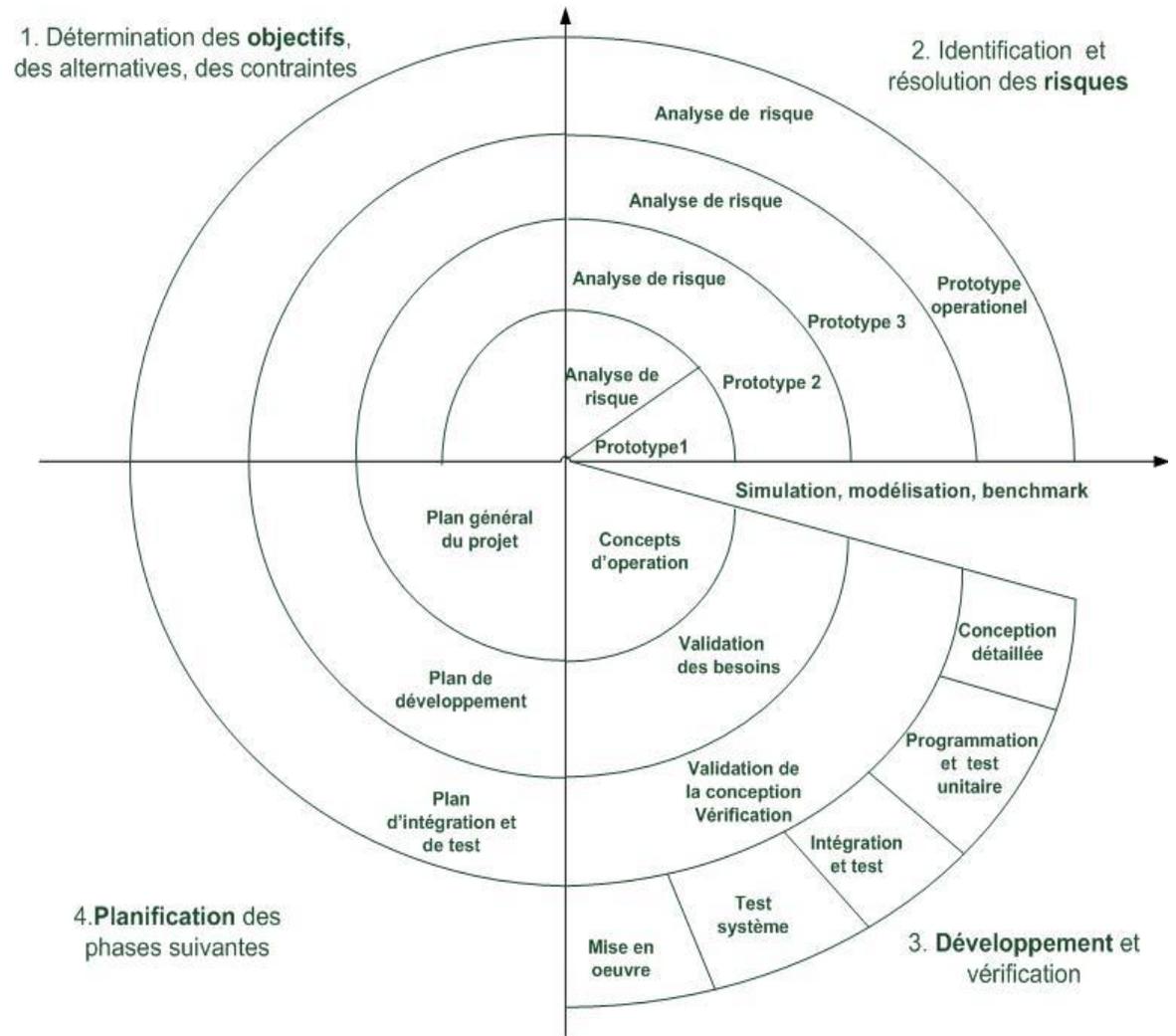
Il s'agit d'écrire une première spécification et de réaliser un sous-ensemble du produit logiciel final. Ce sous-ensemble est alors de plus en plus raffiné et évalué jusqu'à obtenir le produit final.



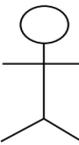
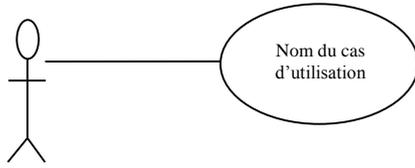


Modèle en spirale :

Chaque cycle de la spirale se déroule en quatre phases.



Partie 2 : Notation UML

Diagrammes	Elément de modélisation	Notation
Diagramme de cas d'utilisation	<p><i>Ce diagramme est destiné à représenter les besoins des utilisateurs par rapport au système. Sa représentation met en jeu trois concepts : l'acteur, le cas d'utilisation et l'interaction entre l'acteur et le cas d'utilisation.</i></p>	
	<p>L'Acteur représente un rôle joué par une entité externe qui interagit directement avec le système étudié.</p>	<div style="text-align: center;">  <p>Nom de l'acteur</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <p>« acteur » Nom de l'acteur</p> </div> </div>
	<p>Le Cas d'utilisation représente un certain nombre d'actions que le système devra exécuter en réponse à un besoin d'un acteur.</p>	<div style="text-align: center;">  <p>Nom du cas d'utilisation</p> </div>
	<p>L'interaction entre un acteur et un cas d'utilisation se représente par une ligne (trait plein).</p>	<div style="text-align: center;">  <p>Nom de l'acteur</p> </div>

Relations entre cas d'utilisation

□ *Relation d'inclusion*

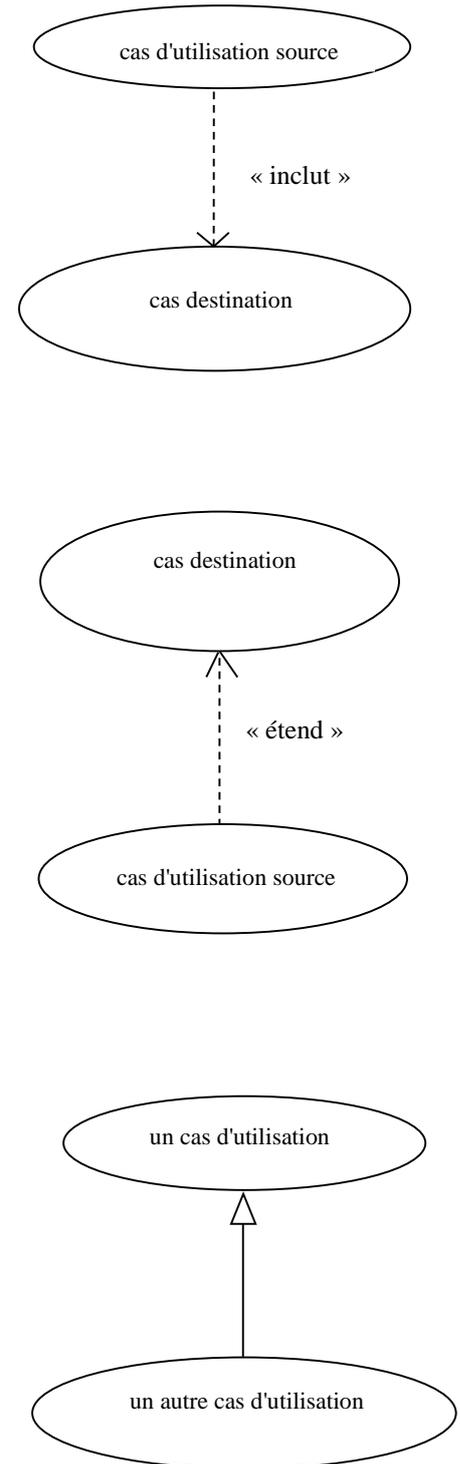
Dans une relation d'inclusion entre cas d'utilisation, une instance du cas d'utilisation source comprend (includ) le comportement décrit par le cas d'utilisation destination.

□ *Relation d'extension*

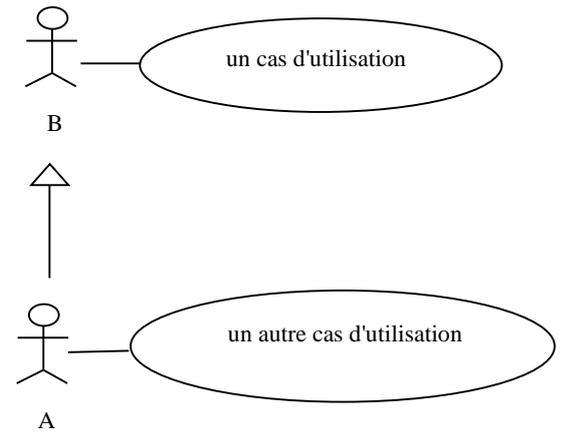
Une relation d'extension entre cas d'utilisation signifie que le cas d'utilisation source étend le comportement du cas d'utilisation destination.

□ *Relation de généralisation*

Une relation de généralisation de cas d'utilisation peut être définie conformément au principe de la spécialisation - généralisation de l'héritage.



Relations entre acteurs : La seule relation possible entre deux acteurs est la généralisation : un acteur A est une généralisation d'un acteur B si l'acteur A peut-être substitué par l'acteur B. Dans ce cas, tous les cas d'utilisation accessibles à A le sont aussi à B, mais l'inverse n'est pas vrai.



Description textuelle des cas d'utilisation	<p>La fiche de description textuelle d'un cas d'utilisation n'est pas normalisée. Cependant, on peut utiliser la structuration décrite dans la partie droite de ce tableau.</p>	<p>I. Identification</p> <ul style="list-style-type: none"> - le nom du cas ; - un résumé de son objectif ; - les acteurs impliqués (principaux et secondaires) ; - les dates de création et de mise à jour de la description courante ; - le nom des responsables ; - un numéro de version. <p>II. Séquencement</p> <p>L'événement qui déclenche le cas</p> <p>Pré-conditions</p> <p>Enchaînement nominal</p> <p>C'est le scénario principal qui doit se dérouler sans incident et qui permet d'aboutir au résultat souhaité.</p> <p>Enchaînements alternatifs</p> <p>C'est un embranchement dans un scénario nominal mais y revient toujours.</p> <p>Enchaînements d'exception</p> <p>Un scénario d'exception intervient quand une erreur se produit, le séquencement nominal s'interrompt, sans retour à l'enchaînement nominal.</p> <p>Post-conditions</p> <p>III. Rubriques optionnelles</p> <p>Contraintes non fonctionnelles, Contraintes liées à l'interface homme-machine.</p>
	<p>REMARQUE</p> <p><i>La série de TD 3 présente un exemple détaillé de la description textuelle des cas d'utilisation.</i></p>	

Diagramme de classes

Ce diagramme est utilisé pour présenter les classes et les interfaces des systèmes ainsi que leurs relations.

Classe : Une classe correspond à la description d'une famille d'objet ayant une même structure et un même comportement.

Visibilités d'attributs et méthodes :

- Public (+) : élément non encapsulé visible par tout.
- Privé (-) : élément encapsulé visible dans les sous-classes de la classe.
- Protégé (#) : élément encapsulé visible seulement dans la classe.

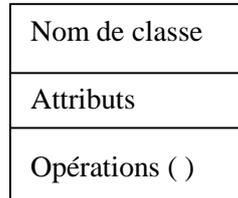
La syntaxe d'un attribut est la suivante :

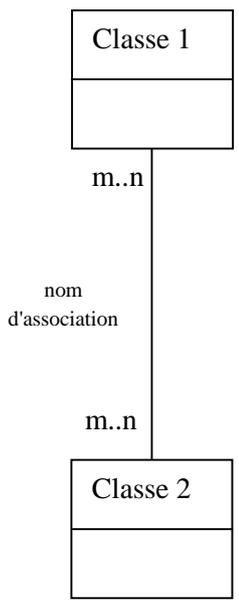
[Visibilité] nom [: type] [multiplicité]
[= valeurParDefaut]

Où, les crochets indiquent les clauses facultatives : seul nom est obligatoire.

La syntaxe d'une opération est la suivante :

[Visibilité] nom ([liste de paramètres])
[: type]



	<p><u>Association</u> : Une association entre classes représente les liens qui existent entre les instances de ces classes.</p> <p>Une association peut être identifiée par son nom.</p> <p>Il est possible d'exprimer les multiplicités (cardinalités) sur le lien d'association.</p> <p>Notation des multiplicités d'association :</p>	 <table border="1" data-bbox="957 840 1500 1108"> <tr> <td></td> <td>un et un seul</td> </tr> <tr> <td>0..1</td> <td>zéro ou un</td> </tr> <tr> <td>*</td> <td>de zéro à plusieurs</td> </tr> <tr> <td>0..*</td> <td>de zéro à plusieurs</td> </tr> <tr> <td>1..*</td> <td>de un à plusieurs</td> </tr> <tr> <td>N</td> <td>Exactement N</td> </tr> <tr> <td>M..N</td> <td>de M à N (entier)</td> </tr> </table>		un et un seul	0..1	zéro ou un	*	de zéro à plusieurs	0..*	de zéro à plusieurs	1..*	de un à plusieurs	N	Exactement N	M..N	de M à N (entier)
	un et un seul															
0..1	zéro ou un															
*	de zéro à plusieurs															
0..*	de zéro à plusieurs															
1..*	de un à plusieurs															
N	Exactement N															
M..N	de M à N (entier)															

REMARQUE

Il est possible de préciser le rôle d'une classe au sein d'une association. Le rôle est placé à une extrémité du lien d'association, il se distingue ainsi du nom de l'association situé au centre du lien. L'exercice 4 de la série de TD 4 présente un bon exemple d'utilisation des rôles.

	<p>□ <i>Association binaire :</i></p> <p>Une association binaire est représentée par un simple trait.</p>	
--	---	---

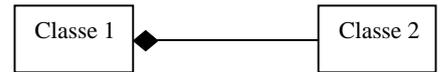
□ *Association n-aire*: Une association n-aire lie plus de deux classes. Elle est graphiquement représentée par un grand losange avec un chemin partant vers chaque classe participante. Le nom de l'association, le cas échéant, apparaît à proximité du losange.



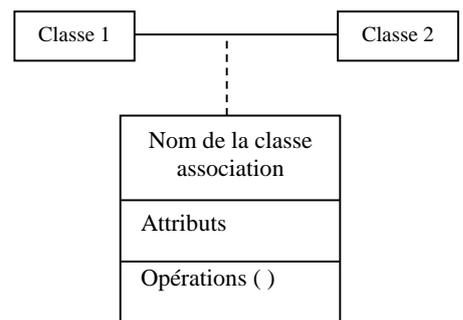
□ *Agrégation* : L'agrégation est une association qui décrit une relation d'inclusion entre une partie et un tout (l'agrégat). L'agrégation se représente par un petit losange blanc du côté de l'agrégat.



- ❑ *Composition* : Une composition est une forme forte d'agrégation. C'est-à-dire que la suppression de l'objet agrégat mène à la suppression des objets agrégés. La cardinalité du côté composite ne doit pas être supérieure à 1 (1 ou 0..1). La composition se représente par un petit losange de couleur noire.

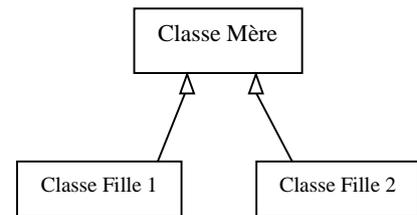


Classe-association : Une classe-association est nécessaire quand une association doit posséder ces propres propriétés.



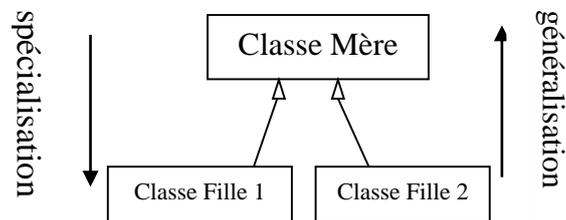
Héritage : L'héritage permet un partage hiérarchique de propriété (attributs et opérations).

La relation d'héritage entre deux classes est représentée par une flèche à la tête en forme de triangle blanc.



REMARQUE

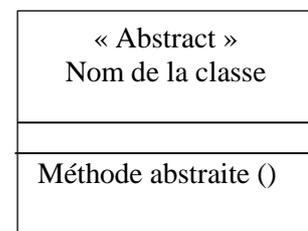
L'héritage est mis en œuvre grâce à deux propriétés qui sont : la généralisation et la spécialisation.



L'exercice 5 de la série de TD 4 présente un bon exemple de généralisation et spécialisation.

Classe abstraite :

Une classe abstraite est une classe qui n'a pas d'instance directe, mais dont les classes descendantes ont des instances.



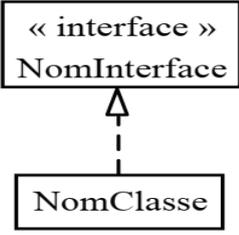
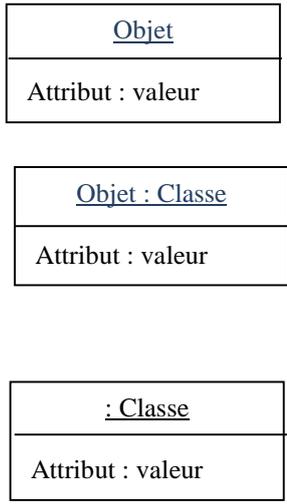
	<p><u>Interface</u> :</p> <p>Une interface n'a pas d'instances directes. Pour être utilisée, elle doit être réalisée (implémentée) par un classeur, qui est souvent une classe.</p>	
Diagramme d'objets	<p><i>Ce diagramme permet de représenter les instances des classes, c'est-à-dire des objets. Rappelant qu'un objet est une instantiation d'une classe.</i></p>	
	<p>Le nom d'un objet est souligné. Il peut être désigné sous trois formes :</p> <p>nom de l'objet, désignation directe et explicite d'un objet.</p> <p>nom de l'objet : nom de la classe, désignation incluant le nom de la classe.</p> <p>: nom de la classe, désignation anonyme d'un objet d'une classe donné.</p>	
	<p>REMARQUE</p>	
	<p><i>Dans un diagramme d'objet la relation entre les objets s'appelle un lien. Si le lien porte un nom (optionnel), il est souligné.</i></p>	

Diagramme de séquence

C'est est un diagramme d'interaction mettant l'accent sur la chronologie de l'envoi des messages.

Ligne de vie des objets : La ligne de vie des objets précise l'existence de l'objet concerné durant un certain laps du temps.

Barre d'activation: le début et la fin d'une bande correspondent respectivement au début et à la fin d'une période d'activité.

un objet



un objet



Message synchrone et asynchrone

❑ **Message synchrone** : Dans ce cas l'émetteur reste en attente de la réponse à son message avant de poursuivre ses actions. Le message de retour peut ne pas être représenté car il est inclus dans la fin d'exécution de l'opération de l'objet destinataire du message.

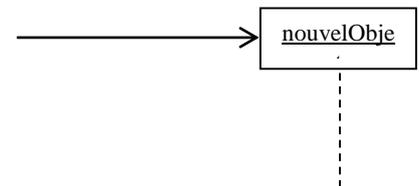
❑ **Message asynchrone** : Dans ce cas, l'émetteur n'attend pas la réponse à son message, il poursuit l'exécution de ses opérations.

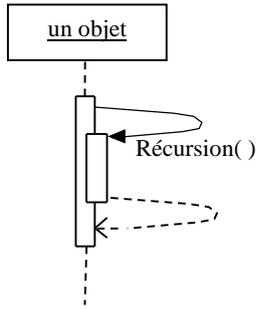
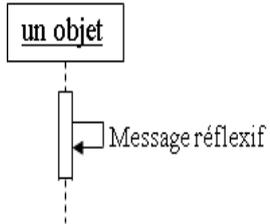
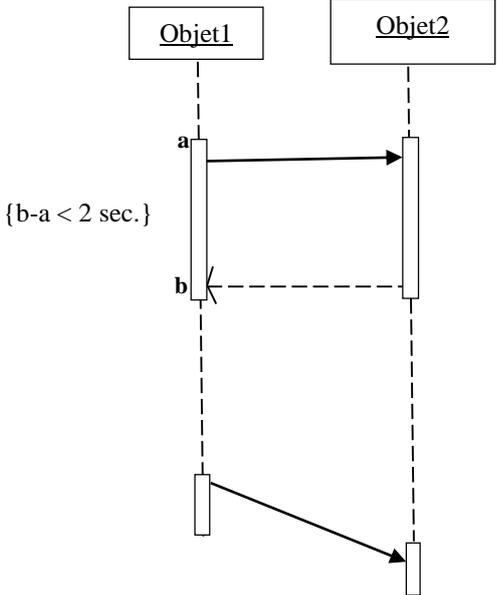
Messages	Notation graphique
Synchrone	
Asynchrone	
Retour	

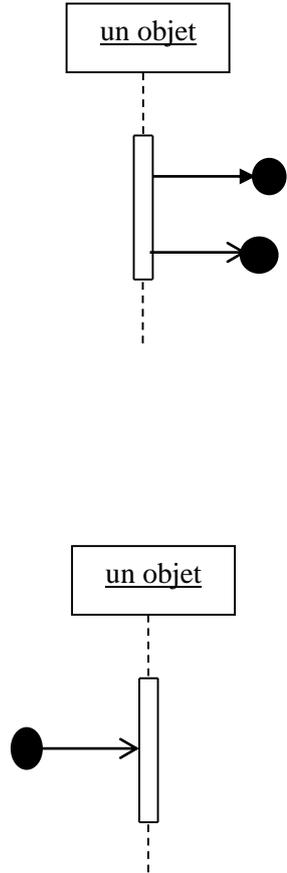
Création et destruction d'objets

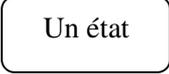
❑ La **création** d'un objet est matérialisée par une flèche qui pointe sur le sommet d'une ligne de vie.

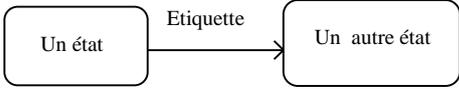
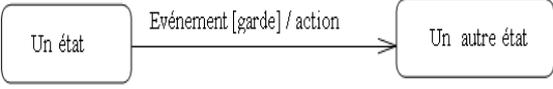
❑ La **destruction** d'un objet est matérialisée par une croix qui marque la fin de la ligne de vie de l'objet.

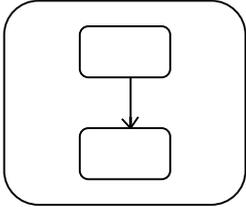
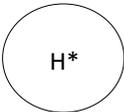


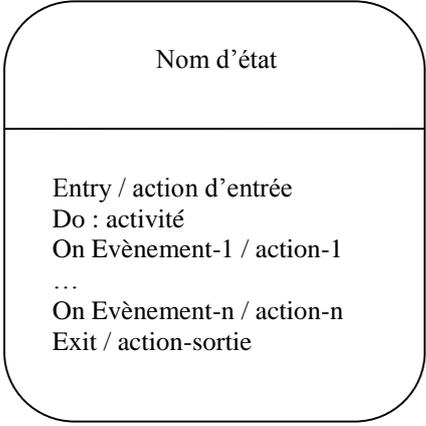
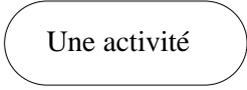
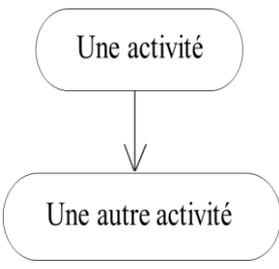
	<p><u>Message récursif :</u></p> <p>Le cas particulier des envois de messages récursifs se représente par un dédoublement de la bande rectangulaire. L'objet apparaît alors comme s'il était actif plusieurs fois.</p>	
	<p><u>Message réflexif :</u></p> <p>Un objet peut également s'envoyer un message. Cette situation se représente par une flèche qui revient en boucle sur la ligne de vie de l'objet.</p>	
	<p><u>Contraintes temporelles :</u></p> <p>Des contraintes temporelles entre les messages peuvent être spécifiées.</p> <p>Exemple : {b-a < 2 sec.}</p> <p>Lorsque l'émission d'un message requiert une certaine durée, il se représente sous la forme d'un trait oblique.</p>	

	<p><u>Messages perdus et trouvés :</u></p> <ul style="list-style-type: none"> ❑ Le message perdu est un message dont nous connaissons l'émetteur mais pas le récepteur. ❑ Le message trouvé est un message dont nous connaissons le destinataire mais pas l'émetteur. Ce message peut être utilisé pour modéliser le comportement d'un élément suite à la réception d'un message d'exception. 	
REMARQUE		
<p><i>On distingue deux types de diagrammes de séquence :</i></p> <p><i>1) Les diagrammes de séquence système qui sont utilisés pour la documentation d'un cas d'utilisation. Ils assurent la description des interactions dans des termes proches de l'utilisateur. La disposition des participants dans ce type de diagramme est :</i></p> <ul style="list-style-type: none"> ❑ <i>Acteur principal à gauche ;</i> ❑ <i>Un objet représentant le système en boîte noire ;</i> ❑ <i>Les éventuels acteurs secondaires sollicités durant le scénario à droite du système.</i> 		

	<p>2) <i>Les diagrammes de séquence représentant les interactions entre objets mettent en jeu :</i></p> <ul style="list-style-type: none"> ❑ <i>Un ensemble d'objets (instances de classes) ;</i> ❑ <i>La chronologie des échanges entre les objets ;</i> ❑ <i>Les contraintes de temps.</i> <p><i>Le premier exercice de la série de TD 5 concerne le diagramme de séquence objets, le deuxième exercice de la même série est un très bon exemple d'un diagramme de séquence système.</i></p>	
Diagramme de communication	<p><i>Le diagramme de communication est un diagramme d'interaction mettant l'accent sur l'organisation structurelle des objets qui envoient et reçoivent des messages.</i></p>	
	<p>Le lien entre les objets est représenté par un trait matérialisant le support des messages échangés. Les messages doivent être numérotés pour refléter l'ordre des échanges.</p>	 <p>La flèche représente le sens et l'identification du message</p>
Diagramme d'états-transitions	<p><i>Les diagrammes d'états-transitions décrivent le comportement interne d'un objet à l'aide d'un automate à états finis.</i></p>	
	<p><u>Etat</u> : Un état est une notion abstraite montrant la façon dont réagit un objet à un événement.</p> <p>Les états sont représentés par des rectangles aux coins arrondis.</p>	

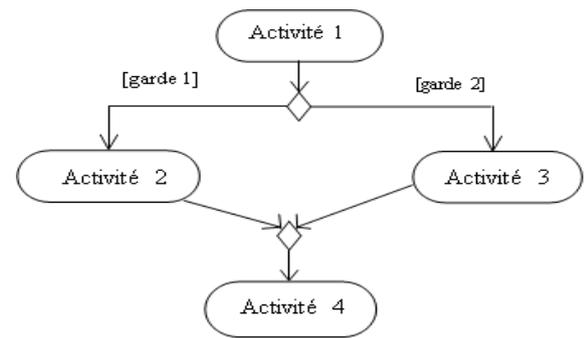
	<p><u>Transition</u> : Une transition représente le passage instantané d'un état vers un autre.</p> <p>Les transitions sont représentées par des arcs orientés liant les états entre eux.</p> <p><u>Etiquette</u> : Une transition peut être déclenchée par un événement, conditionné à l'aide de « gardes » (expression booléennes), et/ou être associée à une action.</p>	 
REMARQUE		
<i>Toutes les parties de l'étiquette de la transition sont facultatives. L'absence d'étiquette indique qu'une transition est automatique.</i>		
	<p><u>Etat initial</u> : L'état initial est représenté par un gros point noir.</p>	
	<p><u>Etat final</u> : L'état final est représenté par un gros point noir encerclé.</p>	
REMARQUE		
<i>En règle général l'état initial et l'état final sont facilement identifiable : on a pas besoin de leur donné un nom.</i>		

	<p><u>État composite</u> : C'est un état qui englobe d'autres états et transitions.</p> <p>Une transition vers un état composite est équivalente à une transition vers le premier sous-état de l'état composite.</p>	
	<p><u>Etat historique</u> : Par défaut on rentre dans un état composite via le premier sous-état. Si l'on souhaite rentrer dans l'état via le dernier sous-état actif avant de le quitter, on utilise un état historique, modélisé par un H encerclé.</p> <p>Il est possible de mémoriser le dernier sous-état actif, quel que soit son niveau de profondeur en utilisant un H* encerclé.</p>	 

	<p><u>L'action dans l'état</u> : Les états peuvent également contenir des actions.</p> <p>Les états peuvent être divisés en compartiments qui contiennent :</p> <ul style="list-style-type: none"> - Le nom de l'état ; - Les transitions internes, c'est-à-dire la liste des actions ou des activités internes effectuées pendant que l'élément est dans cet état. 	
Diagramme d'activités	<p><i>Les diagrammes d'activités décrivent le comportement d'une méthode, le déroulement d'un cas d'utilisation et/ou les enchainements d'activités.</i></p>	
	<p><u>Activité</u> : Une activité désigne une suite d'actions.</p>	
<p><u>Transition</u> : La transition est déclenchée par la fin d'une action et provoquent le début immédiat d'une autre action (elles sont automatiques).</p>		

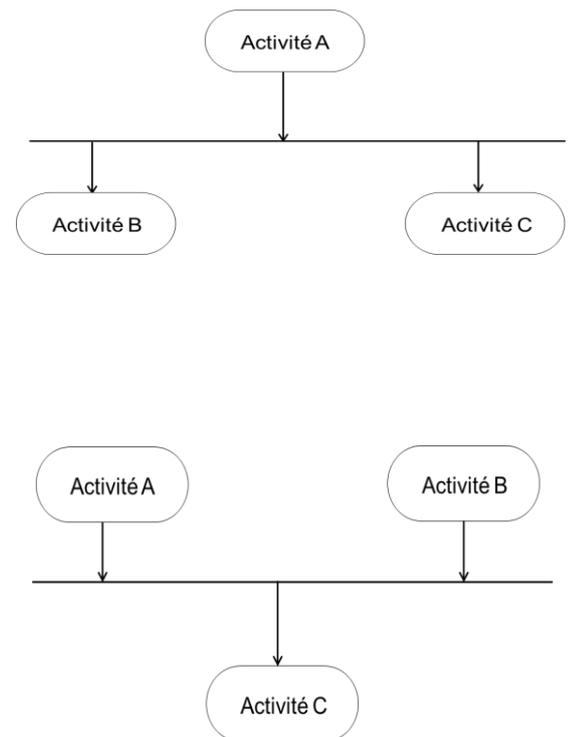
Le comportement conditionnel est décrit par des **décisions / fusions**.

- ❑ Une **décision** permet de représenter des transactions conditionnelles en utilisant des gardes.
- ❑ Une **fusion** marque la fin d'un comportement conditionnel.



Il est possible de synchroniser les transitions à l'aide des barres de synchronisation.

- ❑ **débranchement** : Les transitions qui partent d'un branchement ont lieu en même temps.
- ❑ **Jonction** : On ne franchit une jonction qu'après réalisation de toutes les transitions qui s'y rattachent.

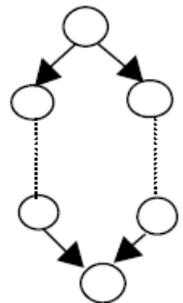
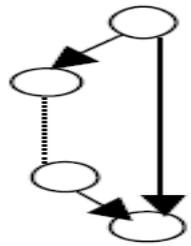
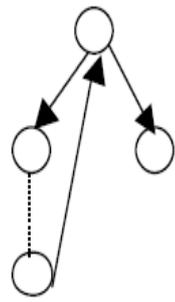


	<p><u>Couloir d'activités</u> : Le couloir d'activités sert à organiser un diagramme d'activités selon les acteurs ou responsables des activités représentées. Il est même possible d'identifier les objets principaux, qui sont manipulés d'activités en activités.</p>	<table border="1"><tr><td data-bbox="949 188 1292 757"><u>Objet : Classe</u></td><td data-bbox="1292 188 1560 757"><u>Objet : Classe</u></td></tr></table>	<u>Objet : Classe</u>	<u>Objet : Classe</u>
<u>Objet : Classe</u>	<u>Objet : Classe</u>			

Partie 3 : Méthodes de test logiciel

Nous distinguons deux méthodes de test : Statique et dynamique ; les méthodes statiques consistent en l'analyse textuelle du code du logiciel afin d'y détecter des erreurs, sans exécution du programme. Les méthodes de tests dynamiques consistent en l'exécution du programme à valider à l'aide d'un jeu de tests.

Méthodes de tests	Principe	Notation
REMARQUE		
<i>Pour la construction des jeux de tests, nous distinguons l'approche aléatoire, l'approche fonctionnelle ou « boîte noire » et l'approche structurelle ou « boîte blanche ». L'approche structurelle ou « boîte blanche » est la seule qui tient compte de la structure interne du code à tester.</i>		

Instructions de l'approche structurée	Affectation, Entrée/Sortie, Appel de sous-programme	
	If then else	
	If then	
	while	

Série de TD N° 1

Exercice 1 :

1. Citer les quatre principaux facteurs de qualité d'un logiciel. Proposer deux autres facteurs et dire dans quelles circonstances, ces deux derniers deviennent-ils plus importants que les autres ?
2. Quelles sont les activités principales du processus de développement d'un logiciel ?
3. Comment peut-on réduire l'écart entre les besoins réels d'un client et les besoins exprimés dans le cahier des charges ?
4. Montrer, pour chaque modèle de cycle de vie, les domaines d'application les plus appropriés.

Exercice 2 :

1. En quoi un modèle de cycle de vie divisé en phases aide-t-il à la gestion du développement d'un logiciel ?
2. Quels sont les inconvénients du modèle en cascade ?
3. Etablir un tableau montrant pour chacune des étapes du modèle en V les documents à délivrer en sortie.

Exercice 3 :

1. Dresser dans un tableau comparatif les types de projet correspondant à chaque modèle de cycle de vie.
2. Attribuer à chacun de ces projets, un modèle de cycle de vie qui lui est adéquat :
 - a) Développement d'un site web pour l'organisation du croissant rouge.
 - b) Développement d'un logiciel aéronautique (aviation) pour la NASA.
 - c) Développement d'un site web de vente de voitures en ligne.

Corrigé de la série de TD N° 1

Exercice 1 :

1. Les principaux facteurs de qualité d'un logiciel :

- La fiabilité (ou robustesse)
- La maintenabilité
- L'efficacité
- La facilité d'emploi

Deux autres facteurs :

- La portabilité (Pour des applications sur Internet)
- L'intégrité (Généralement pour des cas comme les banques)

2. Activités principales du processus de développement d'un logiciel :

Quel que soit l'approche adoptée pour développer un logiciel, on y retrouve un certain nombre d'activités de base :

- Analyse des besoins
- Spécification globale
- Conception architecturale et détaillée
- Programmation
- Gestion de configuration et intégration
- Validation et vérification
- Maquettage (prototypage)

Pour plus de détails le lecteur pourra se référer au document [7].

3. La réduction de l'écart entre les besoins réels d'un client et les besoins exprimés dans le cahier des charges se fait en utilisant le maquettage (prototypage rapide), on présente alors à l'utilisateur un système incomplet et on enrichit celui-ci à mesure que les besoins réels deviennent apparents.

4. Les domaines d'application les plus appropriés, pour chaque modèle de cycle de vie :

Le choix d'un modèle dépend fortement du domaine d'application, lorsque ce dernier est critique, les modèles adéquats sont le modèle de la spirale et à un degré moins le prototypage. Pour des applications classiques (gestion), le modèle en V ou le modèle en cascade peuvent être appropriés. Pour des domaines nécessitant les livraisons rapides mais non critiques le modèle à incrément pourra se justifier.

Exercice 2 :

1. Un cycle de vie décomposé en plusieurs phases permet d'avoir une meilleure visibilité du projet. Le cycle de vie permet ainsi de détecter les erreurs le plus tôt possible et ainsi de maîtriser la qualité du produit, les délais de sa réalisation et les coûts associés.

2. Inconvénients du modèle en cascade :

- Difficulté d'avoir toutes les spécifications du client ;
- Preuve tardive du bon fonctionnement ;
- Pas transparent au client lors du développement.

3. Le tableau montrant pour chacune des étapes du modèle en V les documents à délivrer en sortie.

Étapes	Documents
analyse des besoins, spécification	cahier des charges plan de test d'acceptation
conception architecturale	document de conception plan de test d'intégration
conception détaillée	document de conception détaillée plan de test unitaire
réalisation	module (code source)
test unitaire	module de test
test d'intégration	système logiciel testé

Exercice 3 :

1. Tableau comparatif des types de projet correspondant à chaque modèle de cycle de vie.

MCVL (Modèle de Cycle de Vie d'un Logiciel)	Types de projet
Modèle en cascade	- Petits projets, - Projets dont les spécifications sont bien connues et fixes.
Modèle en V	Projets de taille et de complexité moyenne (les besoins sont bien connus, l'analyse et la conception sont claires).
Modèle par prototypage	Projets innovants.
Modèle par incrément	Grand projet et nécessité de livraisons rapide.
Modèle en spirale	Projets innovants, à risques, dont les enjeux sont importants.

2)

Projet	Type de projet	Modèle de cycle de vie
1	Simple, classique	Cascade
2	Critique, complexe, innovant	Prototypage, spirale
3	Complexe, non innovant	V/cascade (si l'équipe est expérimentée). Le prototypage et la spirale sont aussi possibles du fait que le projet est complexe.

Série de TD N° 2

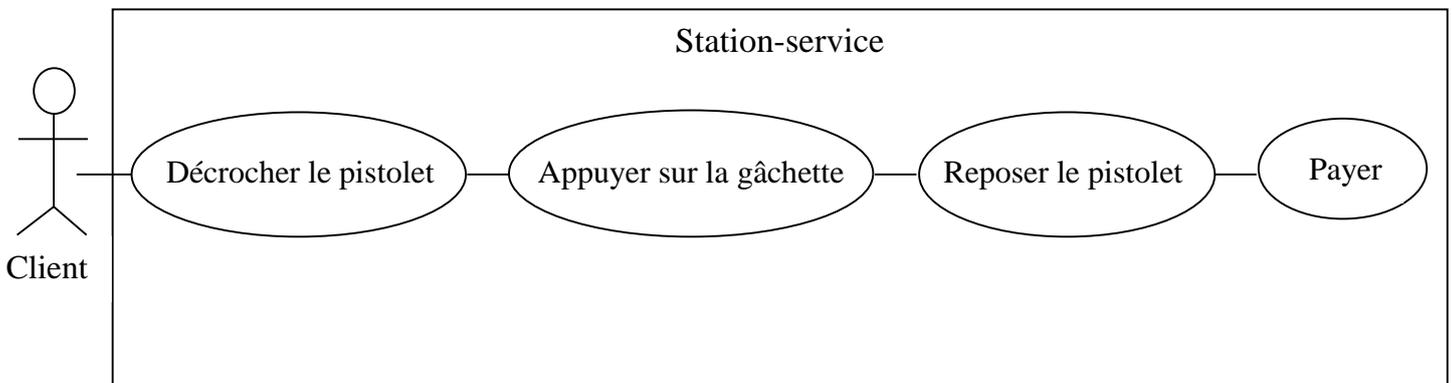
Exercice 1 : (station de service 1)

Considérons le système informatique qui gère une station-service de distribution d'essence. On s'intéresse à la modélisation de la prise d'essence par un client.

1. Le client se sert de l'essence de la façon suivante. Il prend un pistolet accroché à une pompe et appuie sur la gâchette pour prendre de l'essence. Qui est l'acteur du système ? Est-ce le client, le pistolet ou la gâchette ?
2. Le pompiste peut se servir de l'essence pour sa voiture. Est-ce un nouvel acteur ?
3. La station a un gérant qui utilise le système informatique pour des opérations de gestion. Est-ce un nouvel acteur ?
4. La station-service a un petit atelier d'entretien de véhicules dont s'occupe un mécanicien. Le gérant est remplacé par un chef d'atelier qui, en plus d'assurer la gestion, est aussi mécanicien. Comment modéliser cela ?

Exercice 2 : (station de service 2)

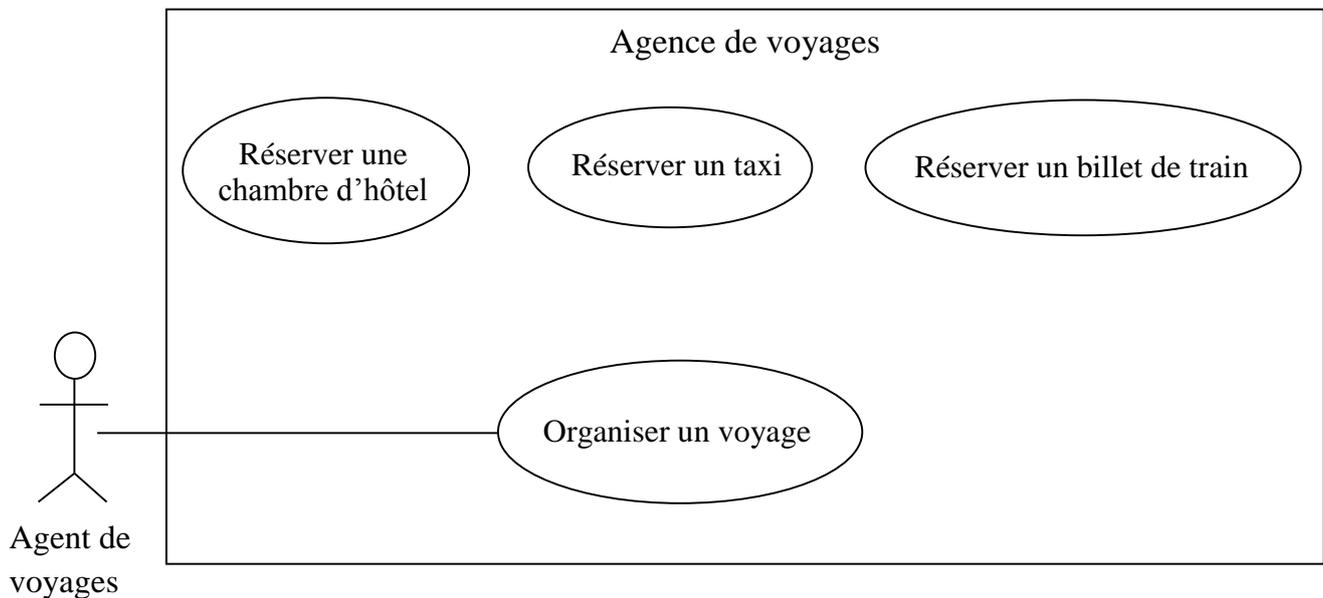
-Quel est le défaut du diagramme suivant ?



Exercice 3:

Choisissez et dessinez les relations entre les cas suivants :

1. Une agence de voyages organise des voyages où l'hébergement se fait en hôtel.
Le client doit disposer d'un taxi quand il arrive à la gare pour se rendre à l'hôtel.



2. Certains clients demandent à l'agent de voyages d'établir une facture détaillée. Cela donne lieu à un nouveau cas d'utilisation appelé « Établir une facture détaillée ». Comment mettre ce cas en relation avec les cas existants ?
3. Le voyage se fait soit par avion, soit par train. Comment modéliser cela ?

Exercice 4 :

Dans un magasin, un commerçant dispose d'un système de gestion de son stock d'articles dont les fonctionnalités sont les suivantes :

- Edition de la fiche d'un fournisseur.
- Possibilité d'ajouter un nouvel article (dans ce cas, la fiche fournisseur est automatiquement éditée. Si le fournisseur n'existe pas on peut alors le créer).
- Edition de l'inventaire : depuis cet écran, on a le choix d'imprimer l'inventaire, d'effacer un article ou d'éditer la fiche d'un article.

Modéliser cette situation par un diagramme de cas d'utilisation.

Exercice 5 :

Modéliser à l'aide d'un diagramme de cas d'utilisation une médiathèque dont le fonctionnement est décrit ci-après.

Une petite médiathèque n'a qu'une seule employée qui assume toutes les tâches :

- la gestion des œuvres de la médiathèque ;
- la gestion des adhérents.

Le prêt d'un exemplaire d'une œuvre donnée est limité à trois semaines. Si l'exemplaire n'est pas rapporté dans ce délai, cela génère un contentieux. Si l'exemplaire n'est toujours pas rendu au bout d'un an, une procédure judiciaire est déclenchée.

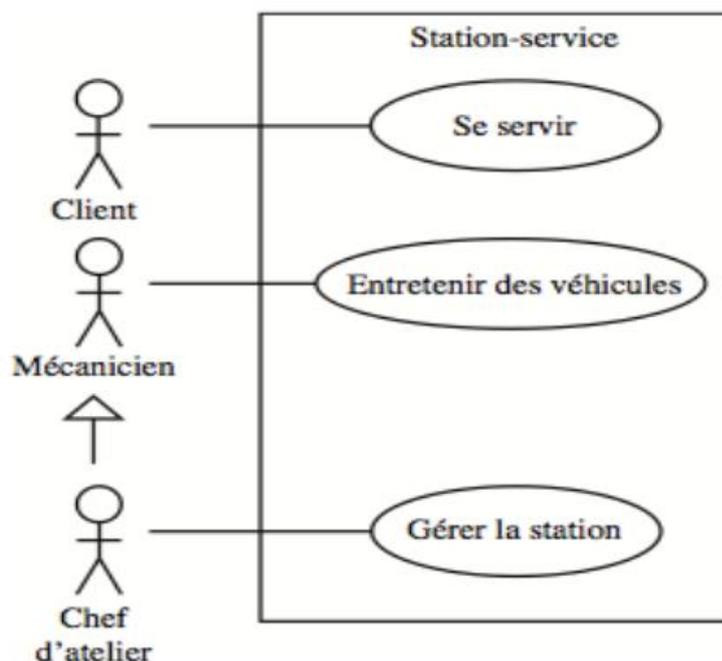
L'accès au système informatique est protégé par un mot de passe.

Corrigé de la série TD N° 2

Exercice 1 :

1. Le client est l'acteur du système. Le pistolet et la gâchette sont des ressources utilisées par le système.
2. Si le pompiste fait uniquement les tâches d'un client, il est inutile de créer un nouvel acteur. Il sera lui-même client.
3. Oui, le gérant est un nouvel acteur.
4. Un nouvel acteur, le chef d'atelier, est créé à la place du gérant. Il hérite d'un autre acteur « le mécanicien ».

La modélisation du système station-service est :

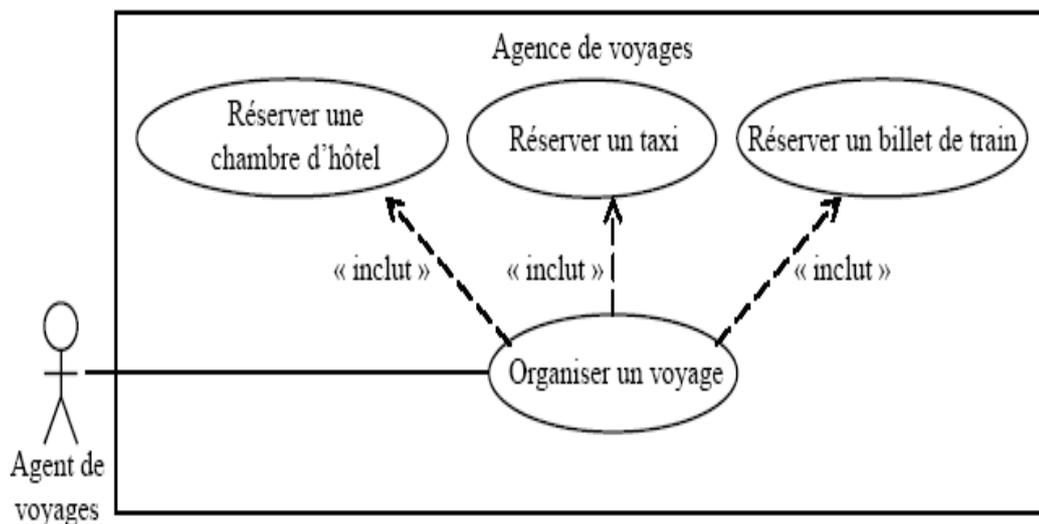


Exercice 2 :

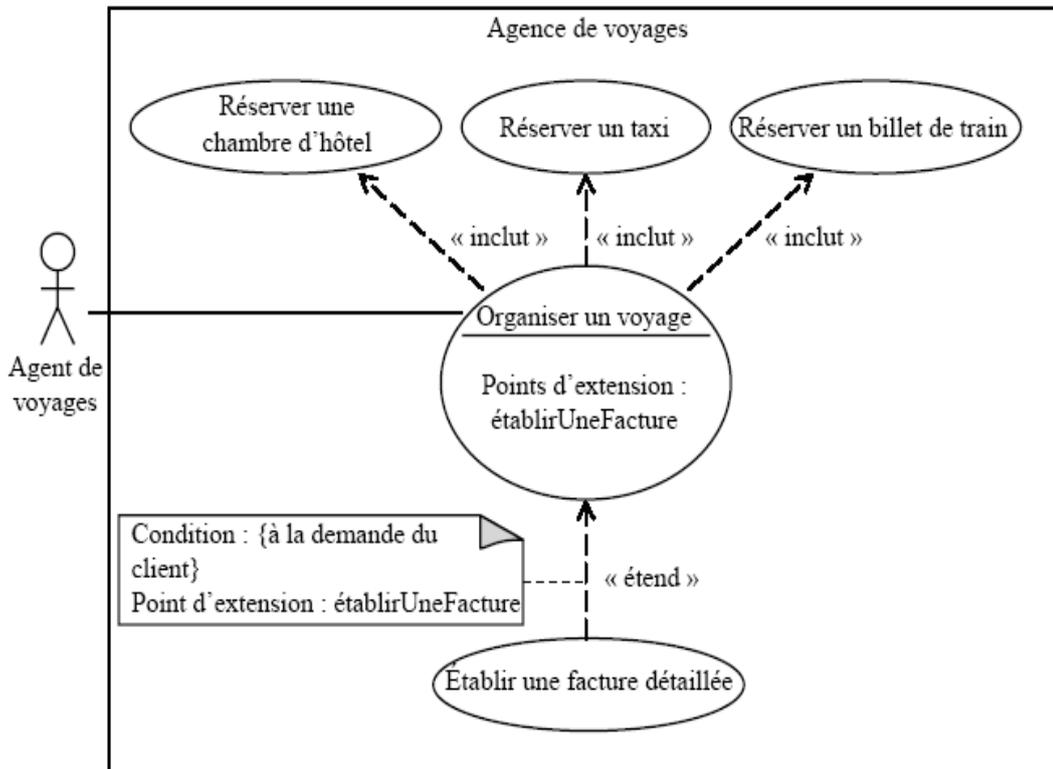
Il ne faut pas introduire de séquençement temporel entre des cas d'utilisation. De plus, il est incorrect d'utiliser un trait plein pour relier deux cas. Cette notation est réservée aux interactions entre les acteurs et les cas d'utilisation.

Exercice 3 :

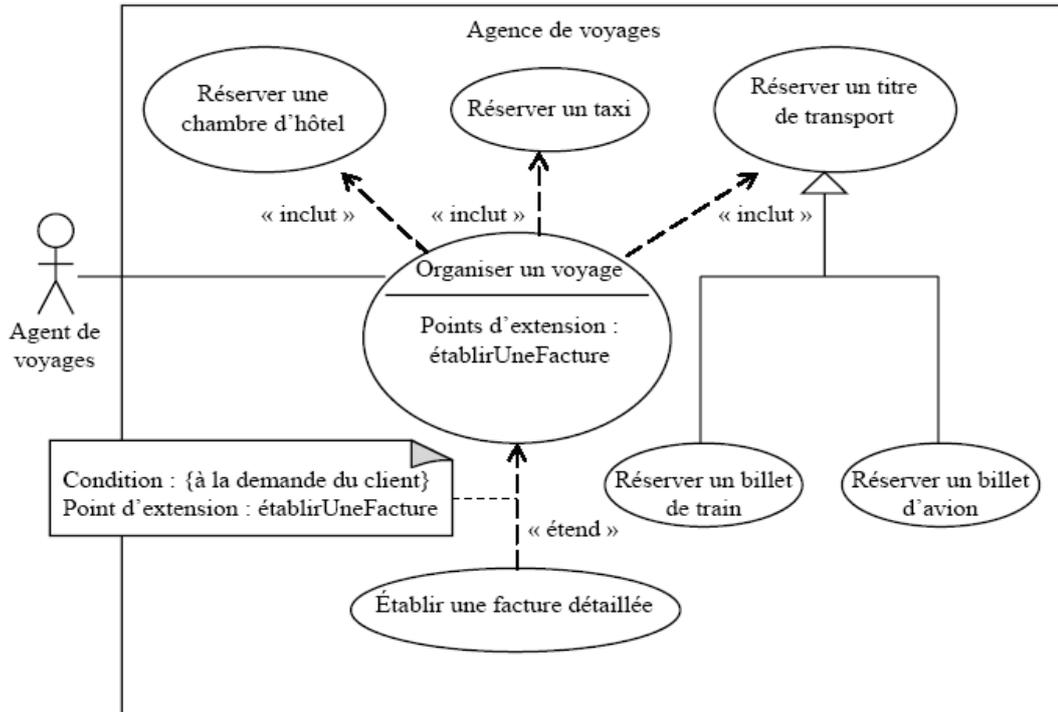
1. Le modélisateur a considéré que l'organisation d'un voyage est trop complexe pour être représentée par un seul cas d'utilisation. Il l'a donc décomposée en trois tâches modélisées par les trois cas d'utilisation « Réserver une chambre d'hôtel », « Réserver un taxi » et « Réserver un billet de train ». Ces trois tâches forment des transactions suffisamment isolées les unes des autres pour être des cas d'utilisation. De plus, ces cas sont mutuellement indépendants. Ils constituent des cas internes du système car ils ne sont pas reliés directement à un acteur.



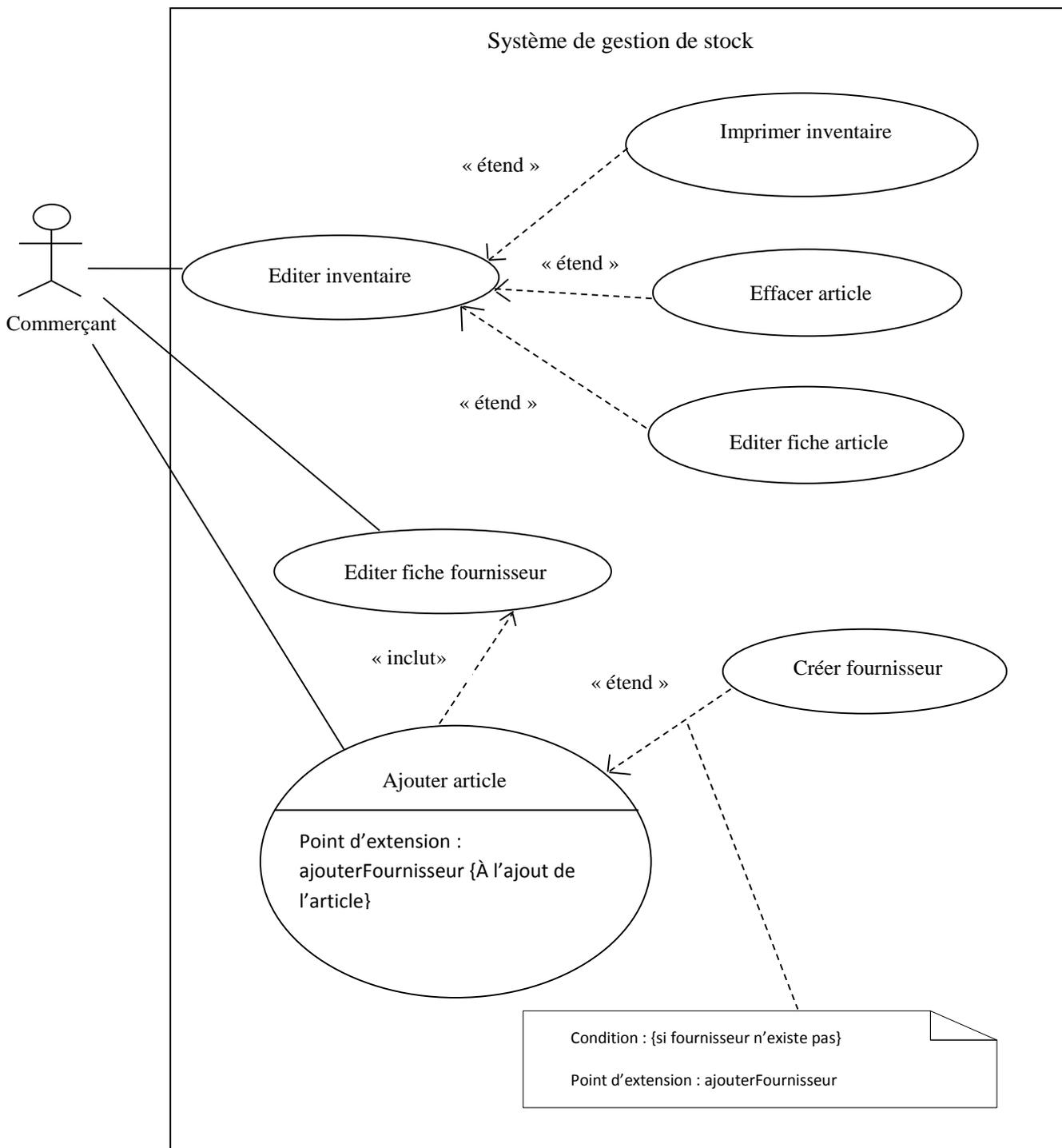
2. L'établissement d'une facture détaillée se fait uniquement sur demande du client. Ce caractère optionnel est modélisé par une relation d'extension entre les cas « Organiser un voyage » et « Établir une facture détaillée ». L'extension porte la condition « à la demande du client ».



3. Il y a maintenant deux cas particuliers : le voyage se fait en train ou en avion. Ces cas particuliers sont modélisés par les cas « Réserver un billet de train » et « Réserver un billet d’avion ». Ceux-ci sont liés à un cas plus général appelé « Réserver un titre de transport ».



Exercice 4 :



Exercice 5 :

La médiathèque n'emploie qu'une employée. Néanmoins, un acteur est déterminé par le rôle qu'il joue vis-à-vis du système à modéliser. Ici, l'employée a deux rôles essentiels :

- le rôle de bibliothécaire qui gère les œuvres ainsi que les adhérents ;
- le rôle de gestionnaire des contentieux ayant les connaissances juridiques suffisantes pour déclencher des procédures judiciaires.

Ainsi nous avons deux acteurs : **Bibliothécaire** et **Gestionnaire des contentieux**.

Un gestionnaire de contentieux est un bibliothécaire avec pouvoir.

Jusqu'à présent la médiathèque fonctionne avec une seule employée. Si, à l'avenir, plusieurs employés devenaient nécessaires, le système informatique pourrait fonctionner avec deux groupes d'utilisateurs : un premier groupe dont le rôle serait limité à celui des bibliothécaires et un deuxième groupe susceptible de gérer les contentieux en plus d'avoir un rôle de bibliothécaire.

L'authentification du groupe auquel appartient un utilisateur du système doit être contrôlé par un mot de passe. La gestion des mots de passe requiert la présence d'un administrateur du système. Ainsi, un nouveau rôle apparaît dans le système, ce qui justifie la nécessité d'un autre acteur Administrateur. Tous les cas d'utilisation liés aux acteurs incluent la procédure d'authentification matérialisée par le cas « S'authentifier ».

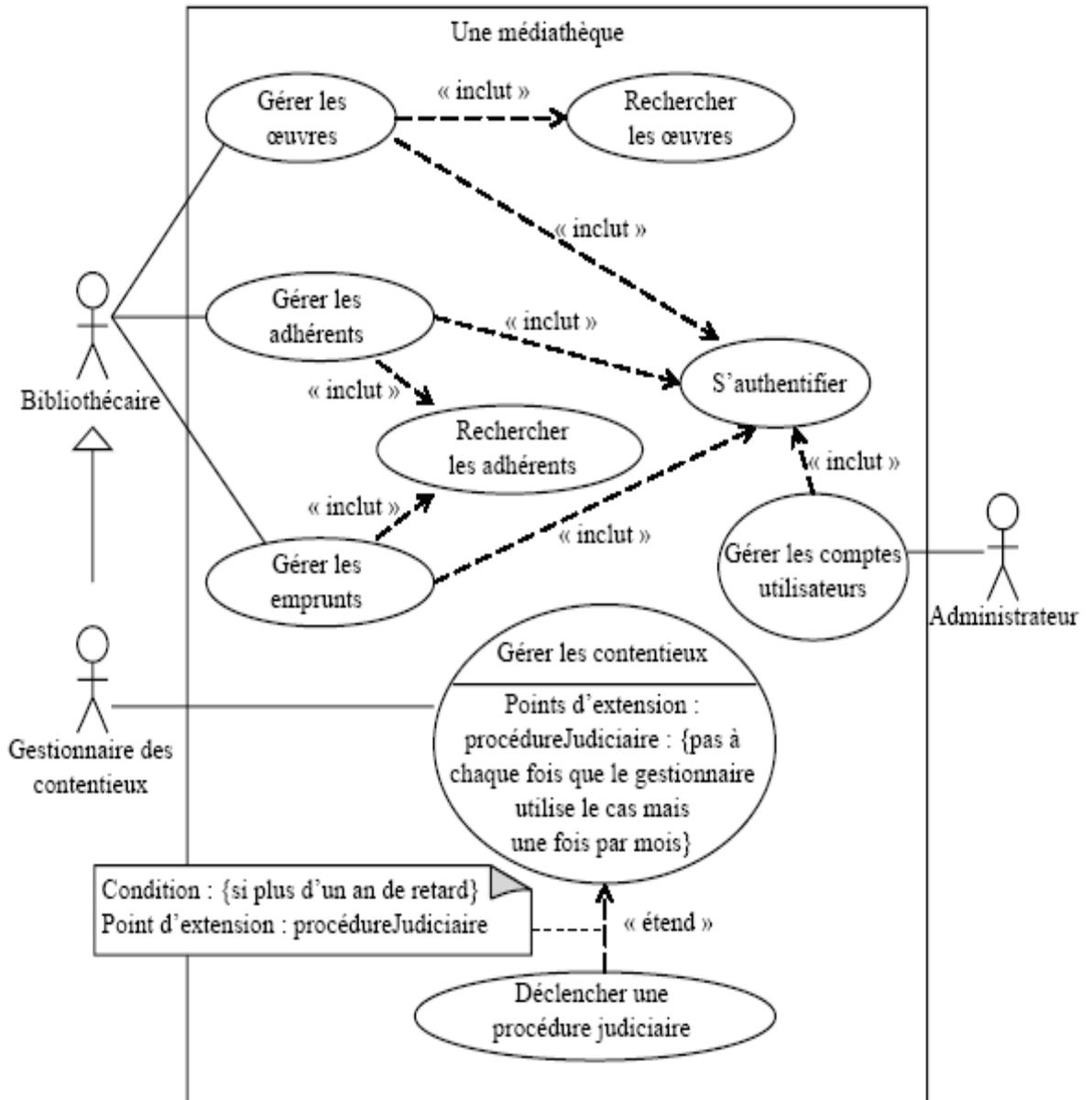
Dans le diagramme, la gestion des adhérents et la gestion des emprunts sont séparées : « Gérer les adhérents » consiste à ajouter, à supprimer ou à modifier l'enregistrement d'un adhérent dans la médiathèque, tandis que « Gérer les emprunts » consiste à prêter des exemplaires aux adhérents déjà inscrits.

La gestion des contentieux a deux degrés d'alerte :

- Un exemplaire n'a pas été rendu au bout de trois semaines.
- Un exemplaire n'a toujours pas été rapporté au bout d'un an.

Cela correspond à deux fonctionnalités distinctes puisque, dans le deuxième cas seulement, il faut déclencher une procédure judiciaire. Nous représentons cela par deux cas d'utilisation : « Gérer les contentieux » et « Déclencher une procédure judiciaire ». Ces deux cas sont liés par une relation d'extension soumise à la condition « si le retard dépasse un an ».

Le diagramme de cas d'utilisation est :



Remarque :

D'après le texte de l'exercice la gestion des contentieux s'applique par rapport à une œuvre donnée et sur un adhérent donné, ce que le gestionnaire des contentieux peut avoir grâce à la relation de généralisation qui le relie avec le bibliothécaire. Le bibliothécaire et le gestionnaire des contentieux peuvent avoir les œuvres grâce à la relation d'inclusion entre « Gérer les œuvres » et « Rechercher les œuvres » et peuvent avoir les adhérents grâce aux relations d'inclusion entre « Gérer les adhérents » et « Rechercher les adhérents » et entre « Gérer les emprunts » et « Rechercher les adhérents ».

Série de TD N° 3

Exercice 1 :

Modéliser avec un diagramme de cas d'utilisation le fonctionnement d'un distributeur automatique de cassettes vidéo dont la description est donnée ci-après.

Une personne souhaitant utiliser le distributeur doit avoir une carte magnétique spéciale. Les cartes sont disponibles au magasin qui gère le distributeur. Elles sont créditées d'un certain montant en euros et rechargeables au magasin. Le prix de la location est fixé par tranches de 6 heures (1 euro par tranche). Le fonctionnement du distributeur est le suivant : le client introduit sa carte ; si le crédit est supérieur ou égal à 1 euro, le client est autorisé à louer une cassette (il est invité à aller recharger sa carte au magasin sinon) ; le client choisit une cassette et part avec ; quand il la ramène, il l'introduit dans le distributeur puis insère sa carte ; celle-ci est alors débitée ; si le montant du débit excède le crédit de la carte, le client est invité à régulariser sa situation au magasin et le système mémorise le fait qu'il est débiteur ; la gestion des comptes débiteurs est prise en charge par le personnel du magasin. On ne s'intéresse ici qu'à la location des cassettes, et non à la gestion du distributeur par le personnel du magasin (ce qui exclut la gestion du stock des cassettes)

Exercice 2 :

Décrivez sous forme textuelle les cas d'utilisation « Emprunter une vidéo » et « Rechercher une vidéo » du diagramme présenté à l'exercice précédent. La recherche d'une vidéo peut se faire par genres ou par titres de film. Les différents genres sont action, aventure, comédie et drame. Quand une liste de films s'affiche, le client peut trier les films par titres ou par dates de sortie en salles.

Corrigé de la série TD N° 3

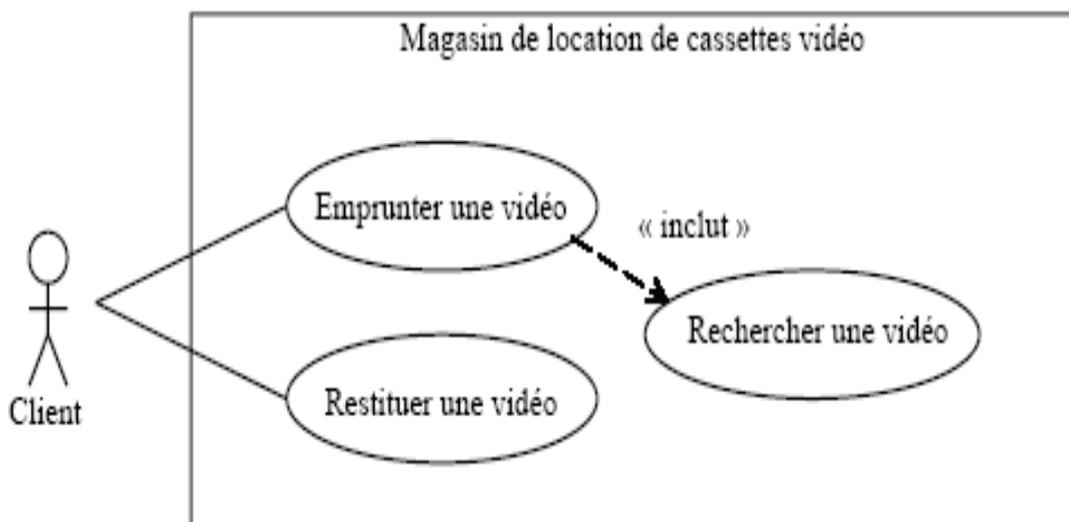
Exercice 1 :

Le seul acteur est le client.

L'acquisition d'une carte et sa recharge ne se font pas via le distributeur : il faut aller au magasin. Ces fonctions ne donnent pas lieu à des cas d'utilisation.

Il ne faut pas faire apparaître un séquençement temporel dans un diagramme de cas d'utilisation. On ne fait donc pas figurer les étapes successives telles que l'introduction de la carte puis le choix d'une cassette, etc. Ce niveau de détails apparaîtra quand on décrira les cas d'utilisation (sous forme textuelle par exemple). Dans un diagramme de cas d'utilisation, il faut rester au niveau des grandes fonctions et penser en termes de transactions.

Il y a donc les cas : « Emprunter une vidéo » et « Restituer une vidéo ». L'emprunt d'une vidéo inclut sa recherche. Une relation d'inclusion intervient donc entre les cas « Emprunter une vidéo » et « Rechercher une vidéo », comme le montre la figure suivante :



Exercice 2 :

Description du cas « Emprunter une vidéo »

Identification

Nom du cas : « Emprunter une vidéo ».

But : décrire les étapes permettant au client du magasin d'emprunter une cassette vidéo via le distributeur automatique.

Acteur principal : Client.

Acteur secondaire : néant.

Date de création : le 31/12/2014.

Date de mise à jour : le 1/1/2015.

Responsable : M. Dupont.

Version : 1.1.

Séquencement

Le cas d'utilisation commence lorsqu'un client introduit sa carte.

Pré-conditions

Le client possède une carte qu'il a achetée au magasin.

Le distributeur est alimenté en cassettes.

Enchaînement nominal

1. Le système vérifie la validité de la carte.
2. Le système vérifie que le crédit de la carte est supérieur ou égal à 1 euro.
3. Appel du cas « Rechercher une vidéo ».
4. Le client a choisi une vidéo.
5. Le système indique, d'après la valeur de la carte, pendant combien de temps (tranches de 6 heures) le client peut garder la cassette.
6. Le système délivre la cassette.
7. Le client prend la cassette.
8. Le système rend la carte au client.
9. Le client prend sa carte.

Enchaînements alternatifs

A1 : Le crédit de la carte est inférieur à 1 euro.

L'enchaînement démarre après le point 2 de la séquence nominale :

3. Le système indique que le crédit de la carte ne permet pas au client d'emprunter une vidéo.

4. Le système invite le client à aller recharger sa carte au magasin.

La séquence nominale reprend au point 8.

Enchaînements d'exception

E1 : La carte introduite n'est pas valide.

L'enchaînement démarre après le point 1 de la séquence nominale :

2. Le système indique que la carte n'est pas reconnue.

3. Le distributeur éjecte la carte.

E2 : La cassette n'est pas prise par le client.

L'enchaînement démarre après le point 6 de la séquence nominale :

7. Au bout de 15 secondes le distributeur avale la cassette.

8. Le système annule la transaction (toutes les opérations mémorisées par le système sont défaites).

9. Le distributeur éjecte la carte.

E3 : La carte n'est pas reprise par le client.

L'enchaînement démarre après le point 8 de la séquence nominale :

9. Au bout de 15 secondes le distributeur avale la carte.

10. Le système consigne cette erreur (date et heure de la transaction, identifiant du client, identifiant du film).

E4 : Le client a annulé la recherche (il n'a pas choisi de vidéo).

L'enchaînement démarre au point 4 de la séquence nominale :

5. Le distributeur éjecte la carte.

Post-conditions

Le système a enregistré les informations suivantes :

- La date et l'heure de la transaction, à la minute près : les tranches de 6 heures sont calculées à la minute près.

- L'identifiant du client.
- L'identifiant du film emprunté.

Rubriques optionnelles

Contraintes non fonctionnelles

Le distributeur doit fonctionner 24 heures sur 24 et 7 jours sur 7.

La vérification de la validité de la carte doit permettre la détection des contrefaçons.

Contrainte liée à l'interface homme-machine

Avant de délivrer la cassette, demander confirmation au client.

Description du cas « Rechercher une vidéo »

Identification

Nom du cas : « Rechercher une vidéo ».

But : décrire les étapes permettant au client de rechercher une vidéo via le distributeur automatique.

Acteur principal : néant (cas interne inclus dans le cas « Emprunter une vidéo »).

Acteur secondaire : néant.

Date de création : le 31/12/2014.

Responsable : M. Dupont.

Version : 1.0.

Séquencement

Le cas démarre au point 3 de la description du cas « Emprunter une vidéo ».

Enchaînement nominal (le choix du film se fait par genres)

1. Le système demande au client quels sont ses critères de recherche pour un film (les choix possibles sont : par titres ou par genres de film).
2. Le client choisit une recherche par genres.
3. Le système recherche les différents genres de film présents dans le distributeur.

4. Le système affiche une liste des genres (les choix possibles sont action, aventure, comédie et drame).
5. Le client choisit un genre de film.
6. Le système affiche la liste de tous les films du genre choisi présents dans le distributeur.
7. Le client sélectionne un film.

Enchaînements alternatifs

A1 : Le client choisit une recherche par titres.

L'enchaînement démarre après le point 1 de la séquence nominale :

2. Le client choisit une recherche par titres.
3. Le système affiche la liste de tous les films classés par ordre alphabétique des titres.

La séquence nominale reprend au point 7.

Enchaînements d'exception

E1 : Le client annule la recherche.

L'enchaînement peut démarrer aux points 2, 5 et 7 de la séquence nominale :

Appel de l'exception E4 du cas « Emprunter une vidéo ».

Post-conditions

Le système a mémorisé le film choisi par le client.

Rubriques optionnelles

Contraintes non fonctionnelles

Contraintes liées à l'interface homme-machine

Quand une liste de films s'affiche, le client peut trier la liste par titres ou par dates de sortie en salles.

Le client peut se déplacer dans la liste et la parcourir de haut en bas et de bas en haut.

Ne pas afficher plus de 10 films à la fois dans la liste.

Série de TD N° 4

Exercice 1 :

Dessiner les diagrammes de classes correspondant aux situations suivantes :

- 1) Un répertoire contient des fichiers.
- 2) Une pièce contient des murs.

Exercice 2 :

On souhaite modéliser le fonctionnement pédagogique d'une université. Étudiants et enseignants sont des personnes caractérisées par un numéro, un nom, un prénom et une adresse. Chaque enseignant possède un grade et on souhaite mémoriser pour chaque étudiant son année d'étude et le diplôme préparé. Les cours sont organisés en modules caractérisés par un code et un intitulé. Plusieurs enseignants peuvent intervenir dans un module, à une date, une heure et dans une salle données. Un enseignant intervient habituellement dans plusieurs modules. Les étudiants s'inscrivent dans les modules à une date et à une heure donnée, mais seuls les étudiants qui suivent effectivement le module obtiennent une note moyenne en fin d'année. Des contrôles sont organisés pour chaque module. Ils sont caractérisés par un numéro de contrôle et une date. Les étudiants ayant effectué le contrôle possèdent une note pour ce contrôle.

- Proposer un diagramme de classes modélisant les spécifications ci-dessus.

Exercice 3 :

Dans une bibliothèque universitaire, on dispose d'un fond documentaire varié : des livres, des périodiques, des proceedings et des thèses.

Ces documents peuvent être consultés sur site sans restriction. Par contre, les emprunts ne peuvent être effectués que pour les livres et les thèses. Tout document est identifié par une référence, un titre, un résumé et un ensemble de mots clés.

La fiche de chaque livre comporte les informations supplémentaires suivantes : les noms des auteurs et de l'éditeur, la date d'édition, le nombre de volumes et le nombre d'exemplaires.

Quant à la fiche de périodique, elle comporte les informations suivantes : le nom de l'éditeur, le numéro de l'édition, la date de parution et le nombre d'exemplaires.

La fiche de thèse comporte les informations suivantes : le nom de l'auteur, la date et lieu de soutenance, le nombre d'exemplaires.

La fiche de chaque proceedings de séminaire comporte les informations suivantes : la désignation et la date du séminaire, le nom de l'institution organisatrice, le nombre d'exemplaires.

En plus des opérations d'emprunt, de consultation et de restitution, le nombre d'exemplaires d'un document peut être modifié suite à l'acquisition ou la perte d'un exemplaire.

- Proposer un diagramme de classes modélisant les spécifications ci-dessus.

Exercice 4 :

Un hôtel est composé d'au moins deux chambres. Chaque chambre dispose d'une salle d'eau qui peut être une douche ou une salle de bain. L'hôtel héberge des personnes. Il peut employer du personnel et est dirigé par un des employés. L'hôtel a les caractéristiques suivantes : une adresse, le nombre de pièces, la catégorie. Une chambre est caractérisée par le nombre et le type de lits, le prix et le numéro. On peut calculer le chiffre d'affaires, le loyer en fonction des occupants.

- Donner le diagramme de classes.

Exercice 5 :

Une carte géographique est caractérisée par une échelle, la longitude et la latitude de son coin inférieur gauche, la hauteur et la largeur de la zone couverte par la carte.

La carte comporte un ensemble de données géographiques de natures diverses :

Les villes et les montagnes sont repérées par un point unique. Chaque point a 2 coordonnées x et y calculées par rapport au coin inférieur gauche de la carte.

Un nom est associé à chaque donnée géographique repérée par un point.

Les routes et les rivières sont repérées par des lignes brisées, c'est à dire par un ensemble de points correspondant aux extrémités de ses segments de droite. Les routes et les rivières ont des noms et des épaisseurs de trait.

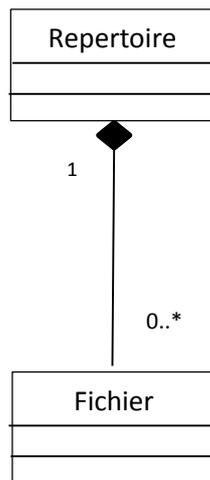
Les lacs, mers et forêts sont représentés par des régions caractérisées par un nom et une couleur de remplissage. Une région est une ligne brisée refermée sur elle-même.

- Proposer un diagramme de classes modélisant les spécifications ci-dessus.

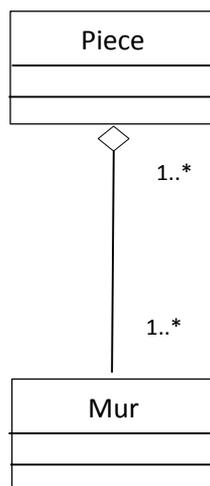
Corrigé de la série de TD N° 4

Exercice 1 :

- 1) Un répertoire contient des fichiers (composition)



- 2) Une pièce contient des murs (agrégation)



Remarque:

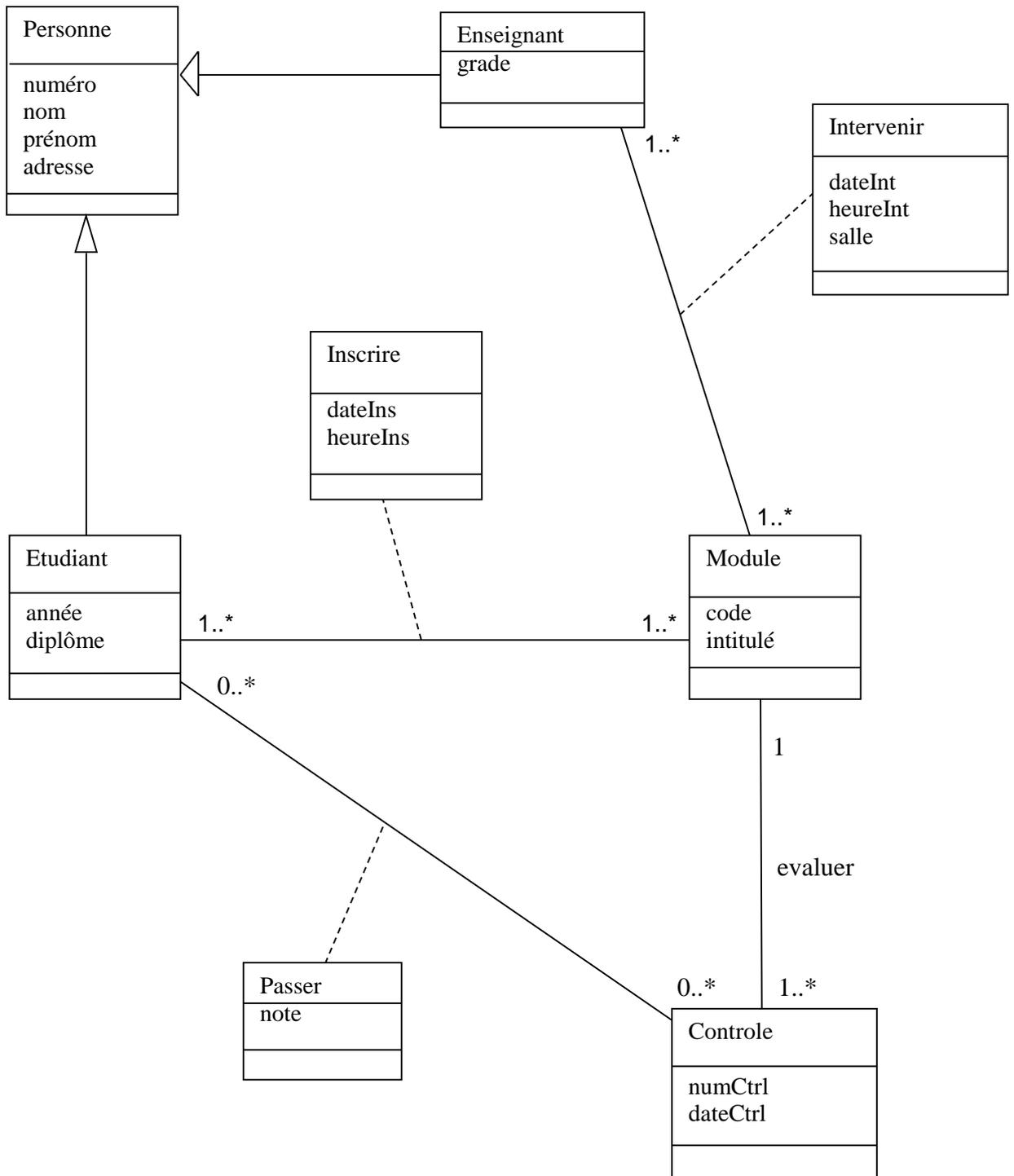
Les phrases 1 et 2 illustrent ce qui différencie l'agrégation de la composition.

« Un répertoire contient des fichiers » : il s'agit d'une composition.

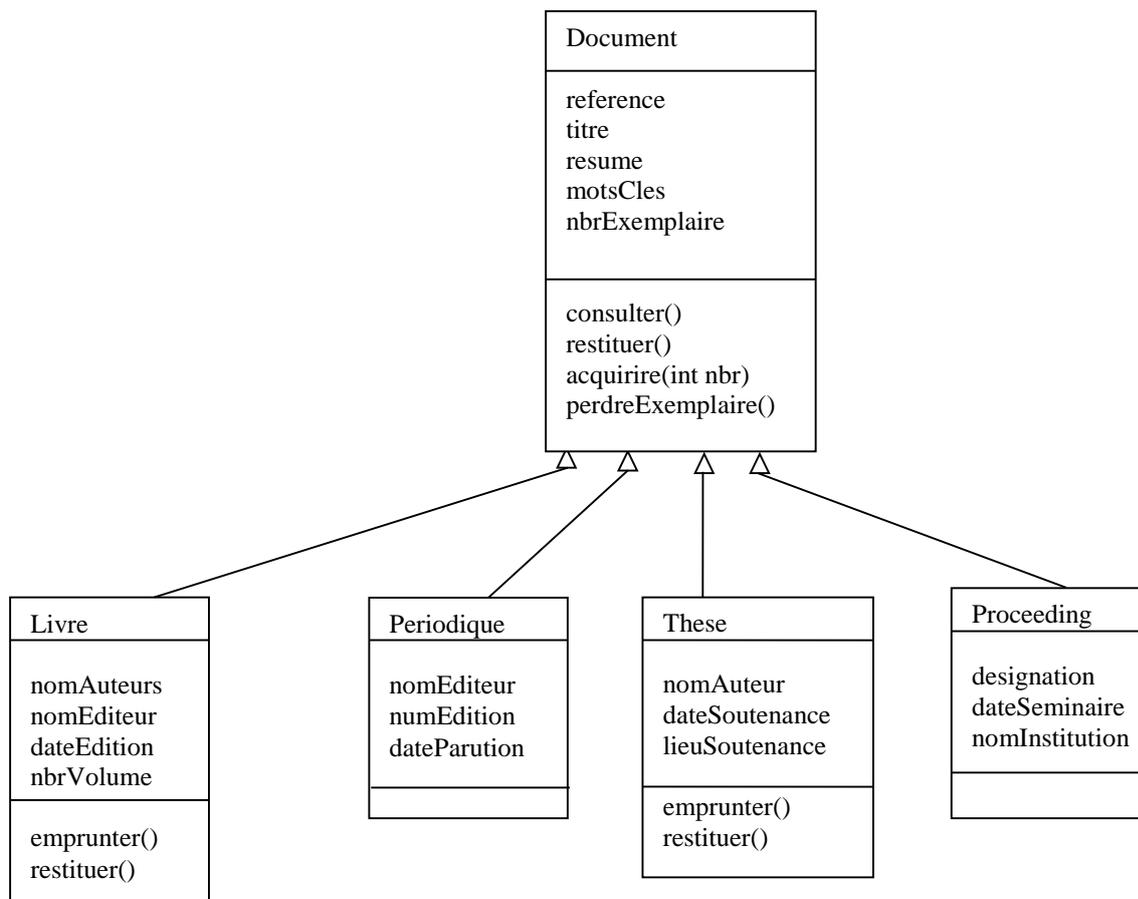
Premier critère à vérifier : la multiplicité ne doit pas être supérieure à un du côté du composite. C'est bien le cas dans la première phrase, puisqu'un fichier appartient à un et un seul répertoire. Second critère : la suppression d'un répertoire entraîne la suppression de tous les fichiers qu'il contient.

Procédons à la même analyse pour la seconde phrase, « Une pièce contient des murs ». Cette fois-ci, après vérification du premier critère, nous devons abandonner la composition. En effet, un mur peut appartenir à deux pièces contiguës (voire plus). La relation n'est donc qu'une agrégation. Afin de compléter les multiplicités, nous considérons qu'une pièce contient au moins un mur (circulaire).

Exercice 2 :



Exercice 3 :

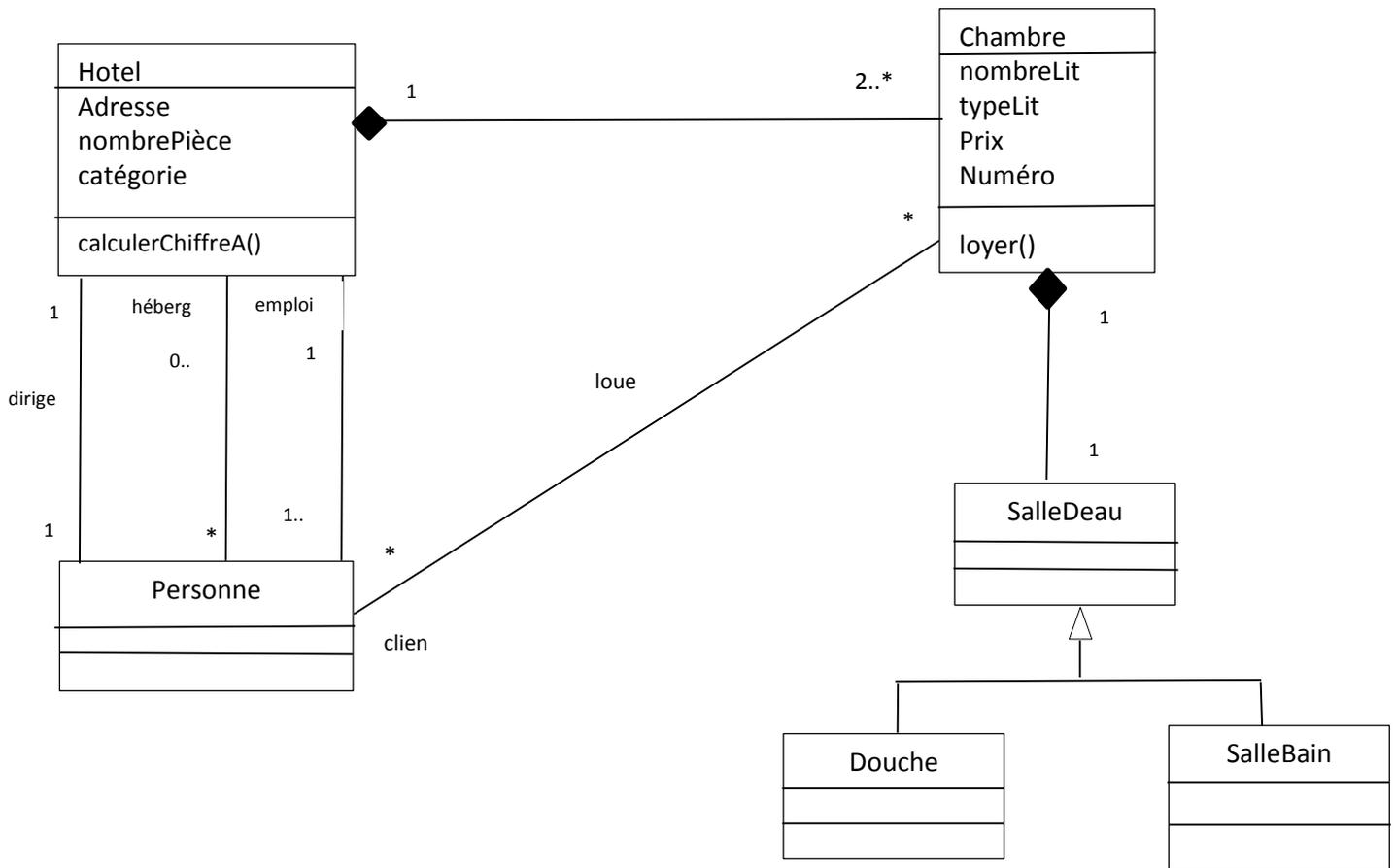


Remarque:

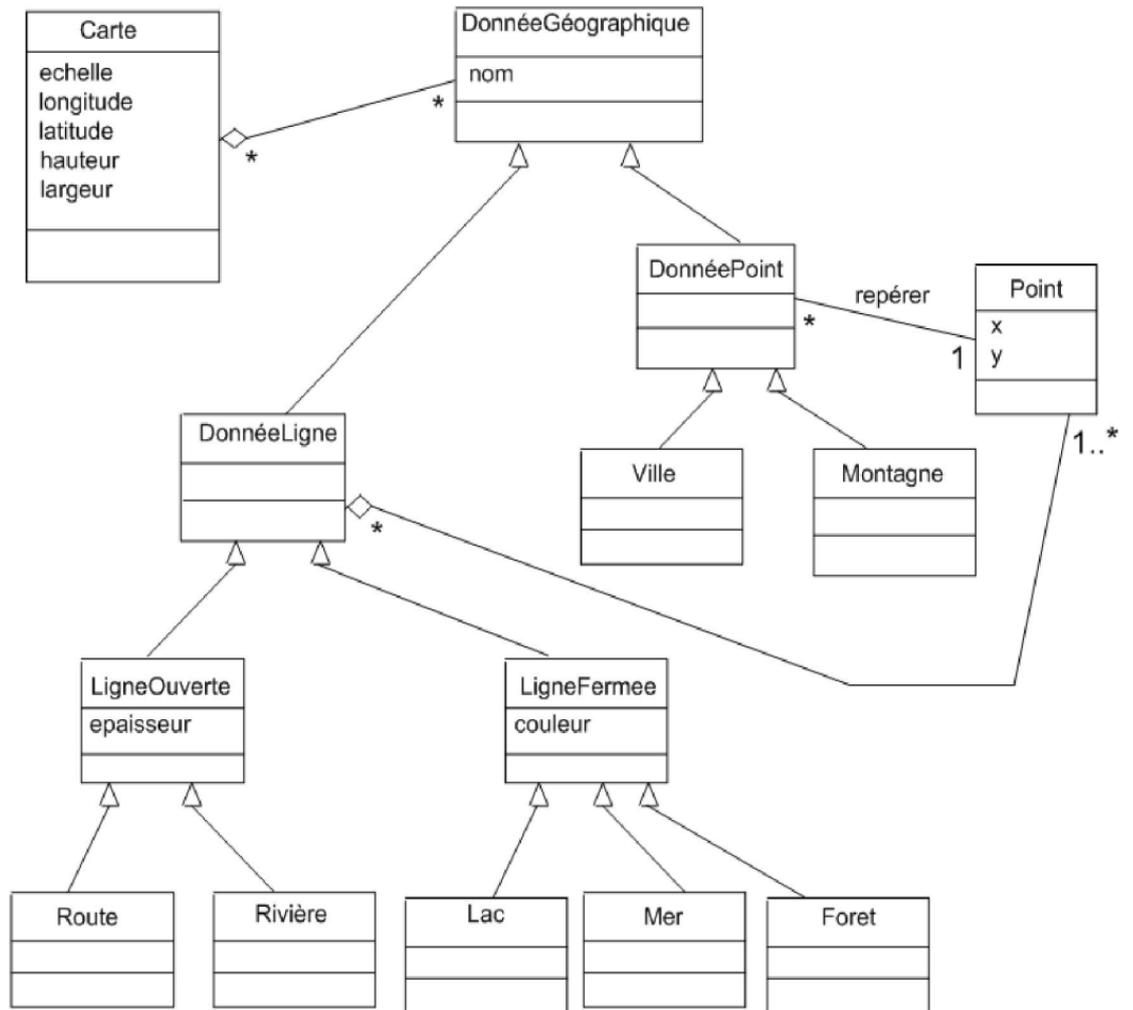
L'opération `restituer ()` est répétée dans la classe mère `Document` et les deux classes filles `Livre` et `These` pour représenter le polymorphisme. Sachant que, la restitution après consultation et celle après emprunt est différente.

La phrase « le nombre d'exemplaires d'un document peut être modifié suite à l'acquisition ou la perte d'un exemplaire » fait apparaître les deux méthodes `acquérir ()` et `perdreExemplaire ()` dans la classe mère `Document`. Pour représenter la modification de nombre d'exemplaires suite à l'acquisition ou la perte il suffit d'ajouter dans la méthode `acquérir ()` ou bien `perdreExemplaire ()` la variable `nbr` qui est un entier et qui sera utilisée dans l'implémentation pour calculer le nombre d'exemplaires modifié.

Exercice 4 :



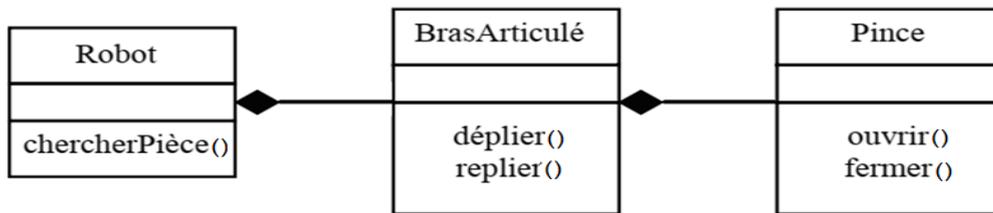
Exercice 5 :



Série de TD N° 5

Exercice 1 :

Le diagramme de classes présenté à la figure suivante modélise un robot qui dispose d'un bras articulé se terminant par une pince. Le fonctionnement du robot est le suivant : le robot déploie son bras, attrape la pièce avec sa pince, replie son bras puis relâche la pièce.



1. Représentez à l'aide d'un diagramme de séquence les interactions entre les objets robot, brasArticulé et pince.
2. Transformez le diagramme de séquence en un diagramme de communication.

Exercice 2 :

Les stagiaires d'un établissement de formation désirent développer une application Web genre réseau social appelée netStagiaire. L'application permet aux membres inscrits de créer un réseau d'amis et d'échanger des contenus.

La validation des nouvelles inscriptions et la suppression des comptes existant se font par l'administration. Il est à noter que la tâche de suppression nécessite la connexion au compte à supprimer. Ainsi, pour devenir membre de netStagiaire, un internaute doit s'inscrire lors de sa visite du site web. Lorsque l'administrateur valide l'inscription, l'internaute devient membre ; il peut alors se connecter pour

réaliser les opérations suivantes : chercher un membre, inviter un membre d'être son ami, accepter un autre membre comme ami, refuser une invitation d'amitié, retirer un ami de sa liste d'amis. Un membre peut également publier des contenus. L'application netStagiaire assure une vérification des contenus publiés par les membres et peut avertir, par un message, un membre qui publie du contenu non conforme au règlement de netStagiaire. Un membre du réseau netStagiaire est identifié par une adresse e-mail, un nom et un prénom. Un membre peut avoir plusieurs amis qui sont aussi membres de netStagiaire. Un membre peut effectuer une publication de contenu à une date donnée. Une publication concerne une ressource à publier ; une ressource possède un nom et peut être soit un message, soit une photo/vidéo soit un lien vers une page web. Une photo/vidéo a en plus du nom, une taille en Mo¹. Un lien vers une page web a une adresse (URL). Un message a un texte.

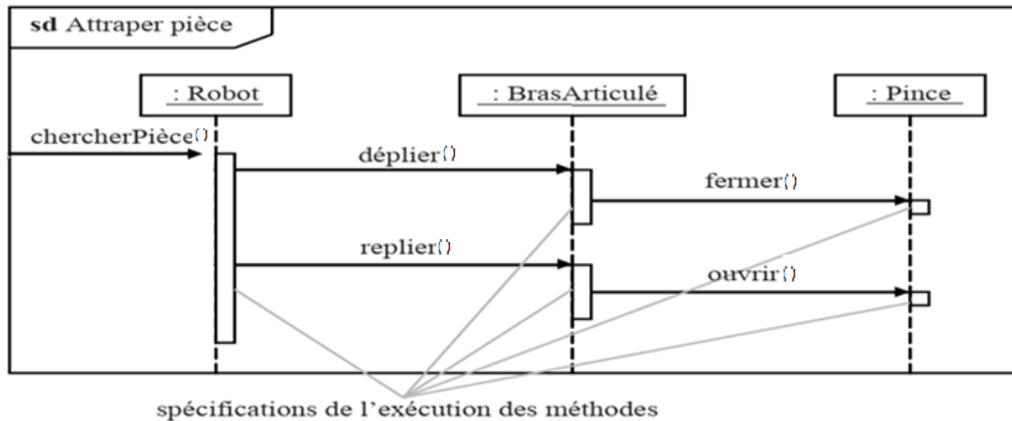
- 1) Etablir le diagramme de cas d'utilisation.
- 2) Etablir le diagramme de séquence du cas d'utilisation « Publier contenu ».
- 3) Etablir le diagramme de classes.

¹Mo : Méga-octet

Corrigé de la série TD N° 5

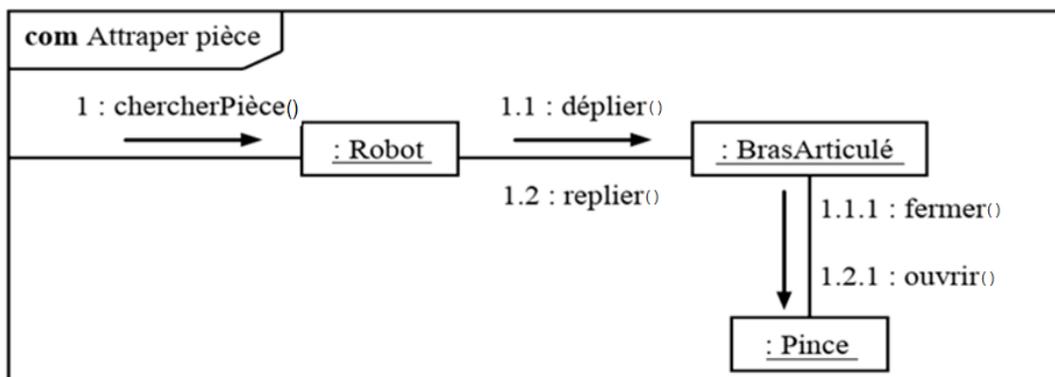
Exercice 1 :

1)



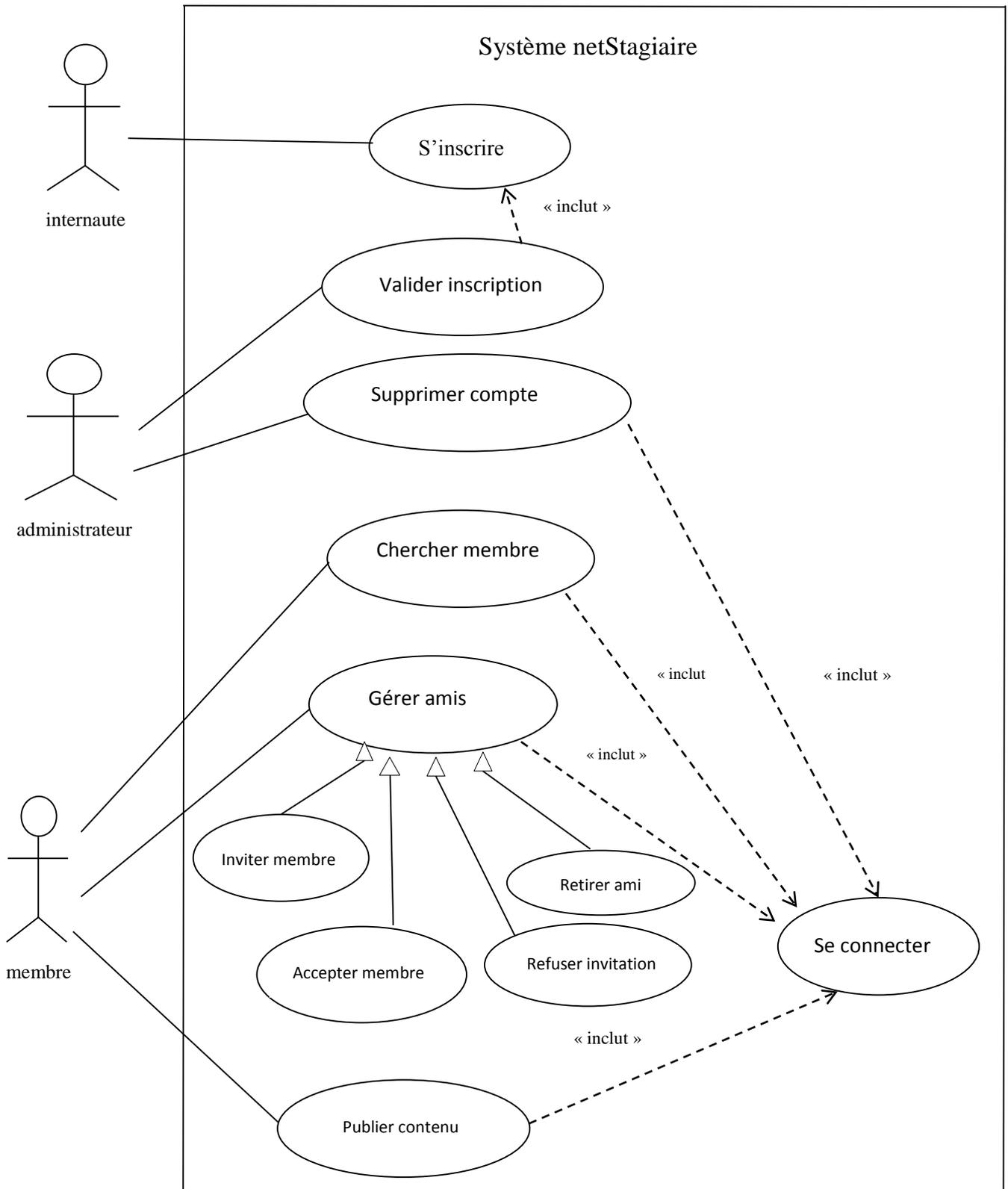
L'émetteur du message est supposé apparaître dans un diagramme non représenté ici.

2)

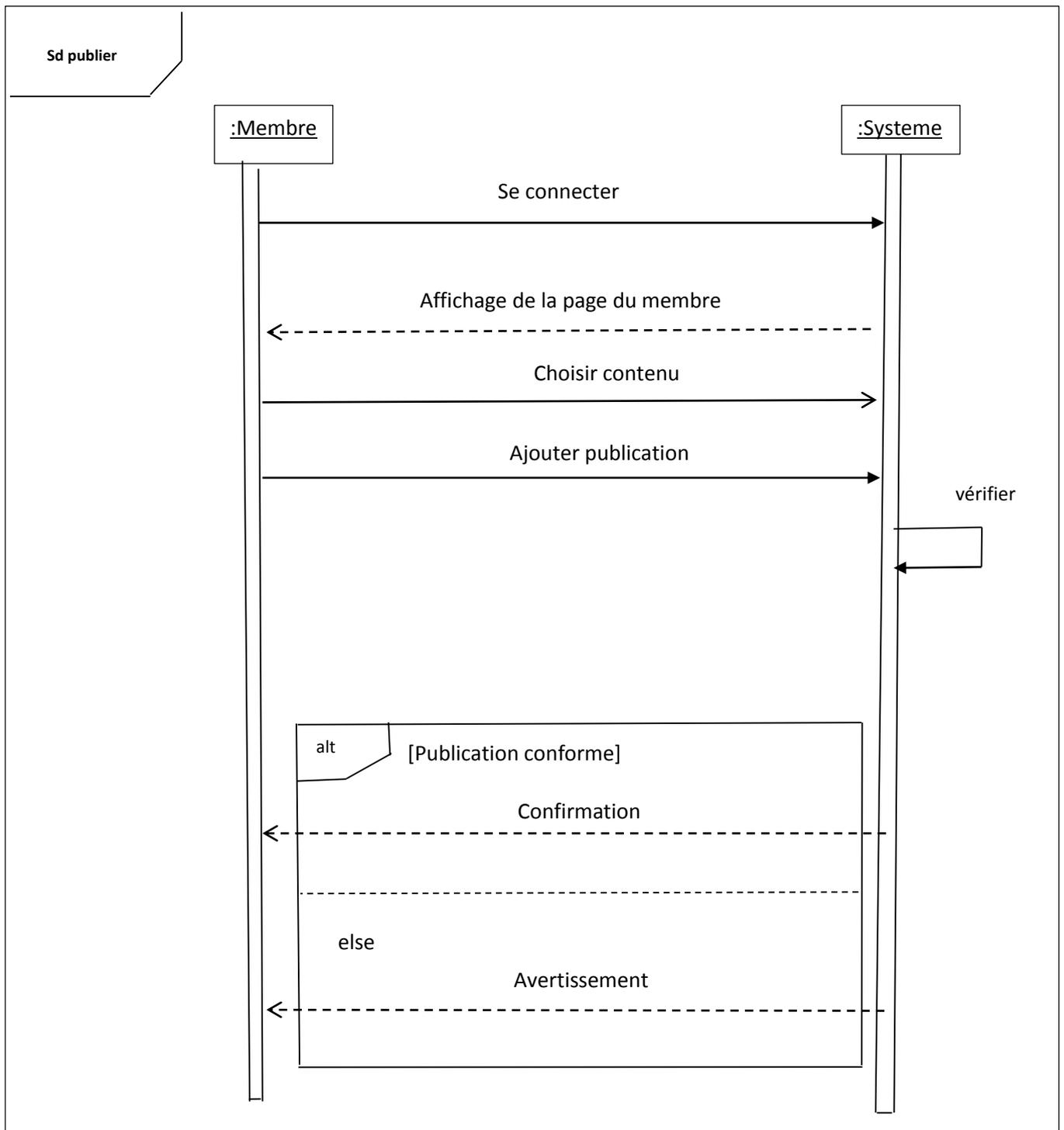


Le premier message, *chercherPièce*, porte le numéro 1 ; le message suivant, *déplier*, est emboîté dans le message *chercherPièce* et porte en conséquence le numéro 1.1 ; le message *fermer* (numéro 1.1.1) est emboîté dans le message *déplier*.

Exercice 2 :



2)

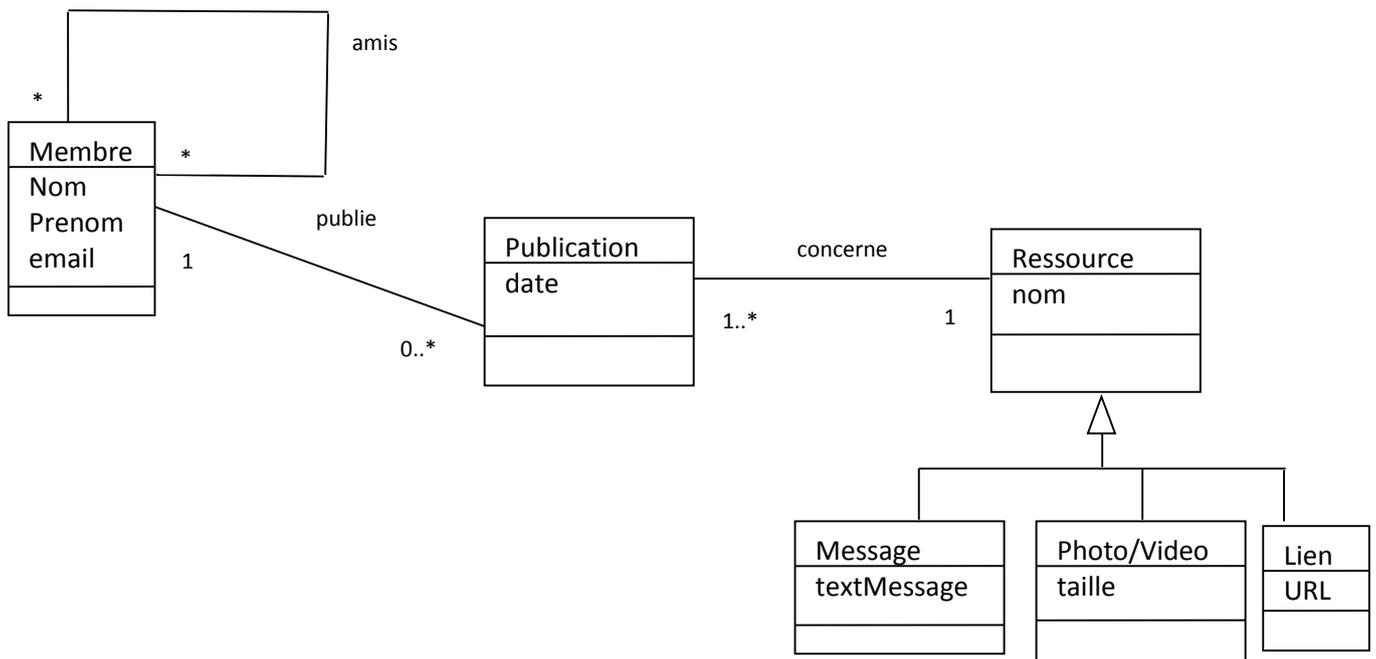


Remarque :

En UML Plusieurs représentation des scénarios alternatifs (ou d'exception) sont possibles :

- Grâce aux fragments d'interactions, il est possible de documenter un scénario alternatif (ou d'exception) dans le diagramme de séquence du scénario nominale.
- Des notes (optionnelles) permettent d'indiquer les scénarii alternatifs et d'exceptions.
- Chaque scénario nominal ou d'exception peut-être documenté par son propre diagramme de séquence.
- Le message « Se connecter » peut être suivi par un fragment « alt » dans le cas où l'énoncé de l'exercice précise que la connexion demande un login/mot de passe.

3)



Série de TD N° 6

Exercice 1 :

Le changement de saisons constitue une boucle continue. On considère un objet de la classe « Saison » de durée de vie infinie. En utilisant des événements de type « change », donner le diagramme des états-transitions de la classe « Saison » correspondant aux états d'une année climatique en l'Algérie. Exemple en 2018, l'Été commence le 21 Juin, l'automne le 23 Septembre, l'hiver le 23 Décembre et le printemps le 20 Mars.

Exercice 2 :

Dessiner un diagramme d'état/transition résumant les états possibles d'un objet « contrat » tel que décrit ci-dessous.

Un ensemble de personnes décident d'établir un contrat. Pour ce faire elles rédigent un projet par itération successive. Le contrat est ensuite informellement accepté par les parties, et devient ce que l'on appelle un préaccord. A ce stade il peut toujours être l'objet de modification et revenir à l'état de projet. Une fois le préaccord définitivement établi, le contrat est signé par les parties. Dès ce moment les partenaires sont liés. Une fois signé, le contrat peut être rendu exécutoire par une décision d'une des parties. Un contrat en exécution peut faire l'objet de discussions qui sont réglées par un arbitre désigné à cet effet. Le contrat une fois exécuté prend fin.

Exercice 3 :

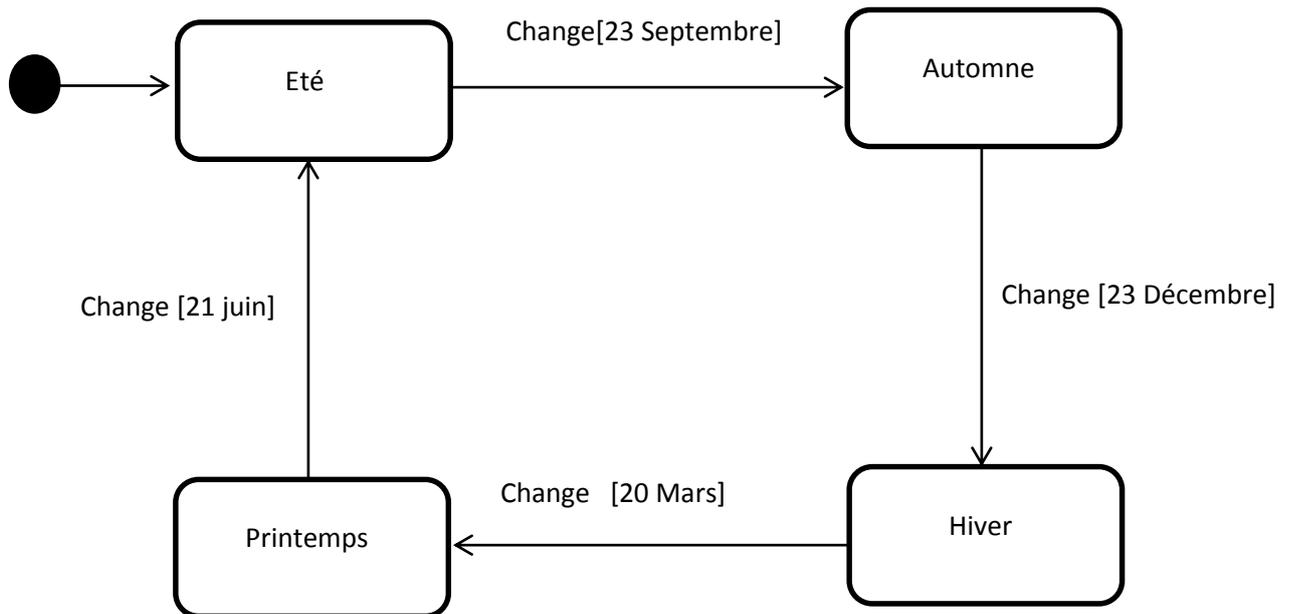
Lors de l'utilisation d'un Distributeur de billets, le client introduit sa carte dont la validité est immédiatement vérifiée. Il est ensuite invité à saisir le code de la carte. Après trois tentatives infructueuses, la carte est avalée. Sinon le client peut indiquer le montant qu'il désire retirer, le solde de son compte bancaire est alors

consulté pour s'assurer que le retrait est possible. En cas de solde insuffisant, le client en est informé et peut alors saisir un montant inférieur. Si le solde du compte est suffisant, le distributeur restitue la carte et délivre alors les billets accompagnés d'un reçu.

- *Décrire le fonctionnement de ce distributeur de billets via un diagramme d'activités.*

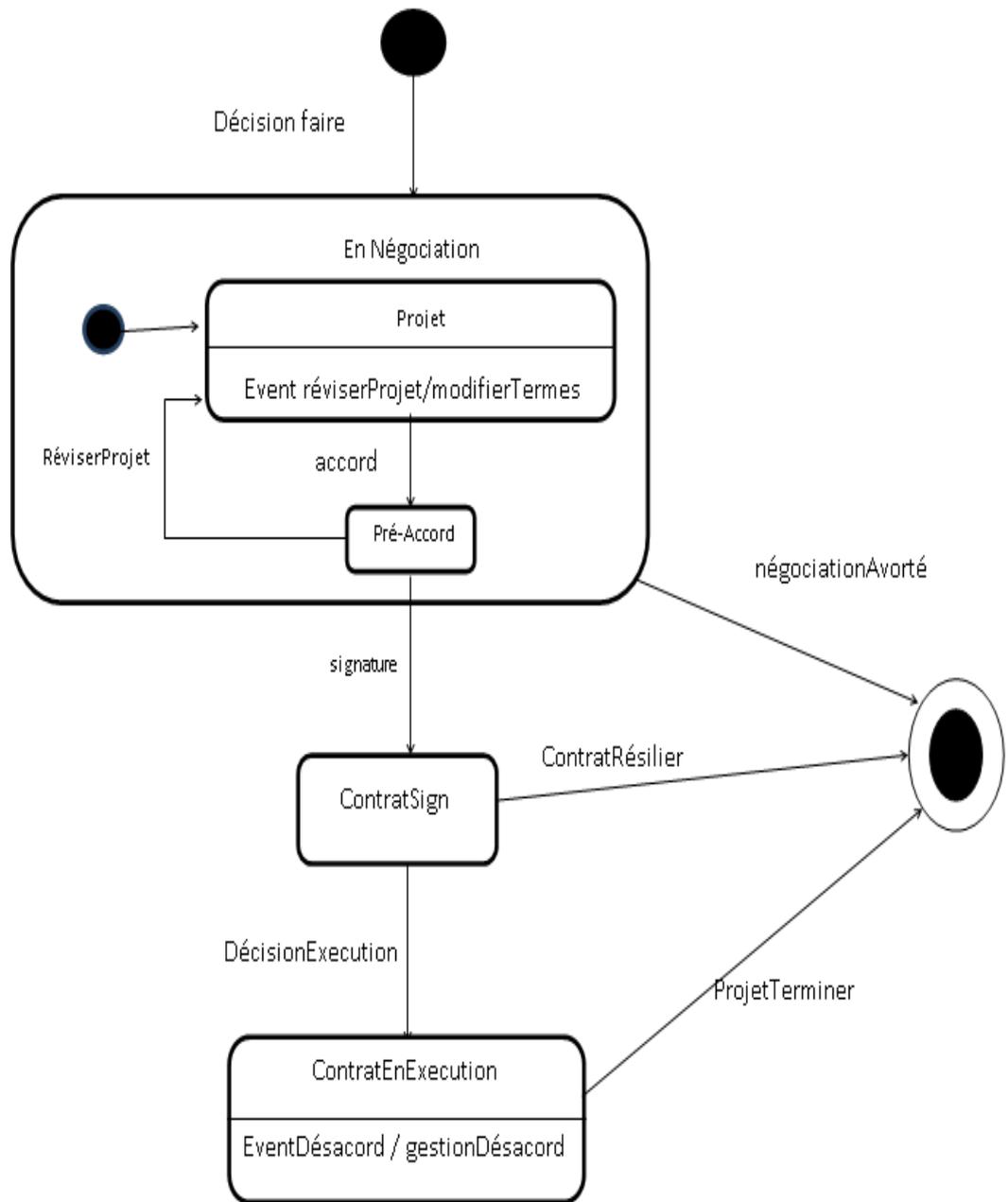
Corrigé de la série TD N° 6

Exercice 1 :

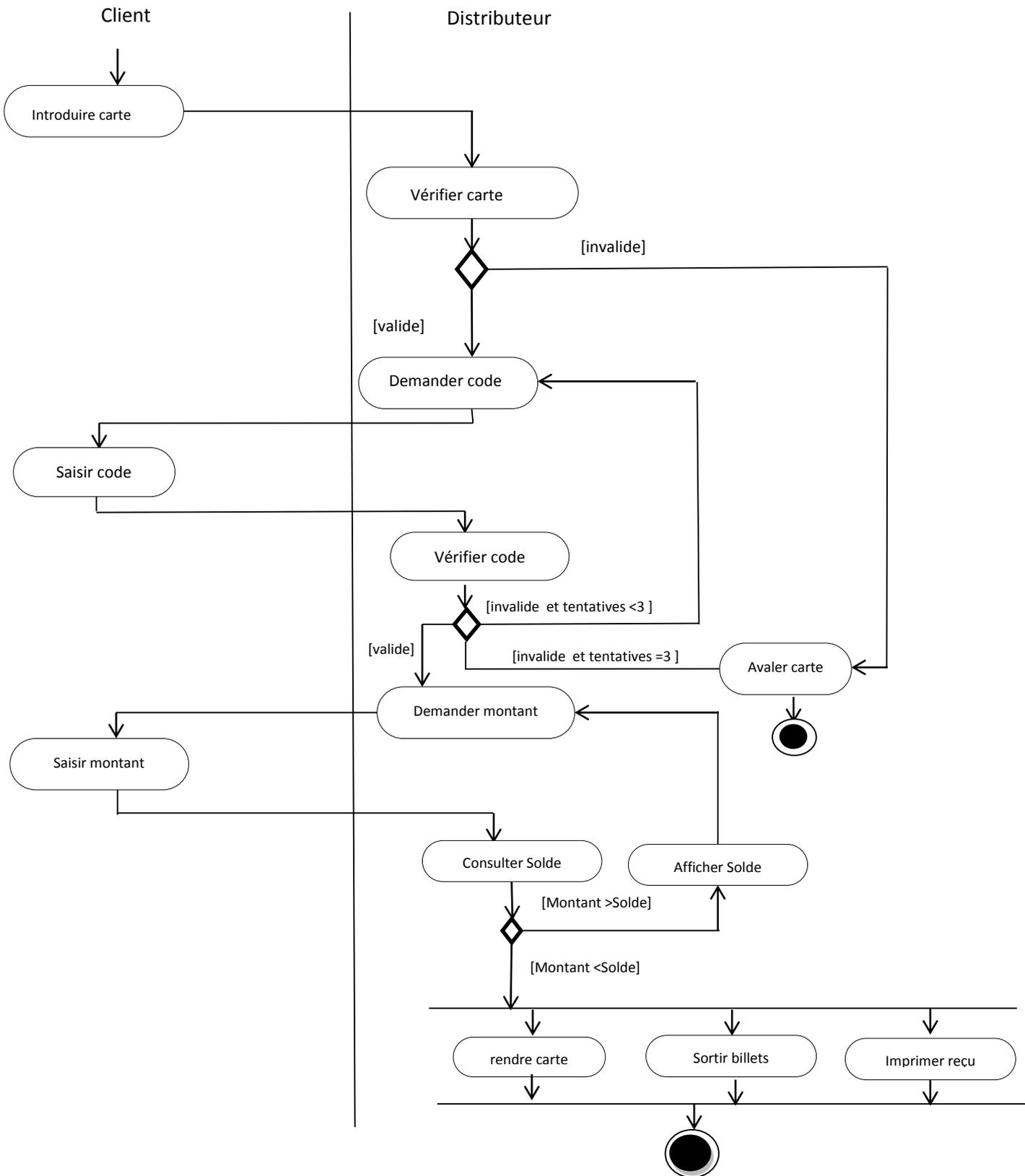


Remarque : Dans cet exercice, l'état initial peut être lié à n'importe quel autre état (l'essentiel est qu'il y est un seul état initial). Les crochets désignent les conditions de changement de saison.

Exercice 2 :



Exercice 3 :



Série de TD N° 7

Exercice 1 :

Donner 3 jeux d'essai satisfaisant les critères de couverture des instructions, des arcs et des chemins pour l'extrait de code suivant :

```
if (x > 10) then a = a + 1; endif  
if (x % 2 = 0) then b = b + 1; endif (où x % 2 donne le  
reste de la division entière de x par 2).
```

Exercice 2:

Soit le programme suivant :

```
1. string1, string2 : deux chaînes de caractères  
2. lire(string1, string2)  
3. si (string1.length = string2.length) alors  
// string1.length donne la longueur de la chaîne string1  
4. i ← 1  
5. tant que i <= string1.length et string1.character[i] =  
string2.character[i]  
// string1.character[i] donne le ième caractère de la chaîne  
string1  
6. i ← i + 1  
7. fin tant que  
8. fin si
```

Donner un jeu d'essai couvrant tous les chemins possibles ; on suppose que la plus petite chaîne à une taille ≤ 1 (0 ou 1). Pour y parvenir, dessiner le graphe de contrôle en numérotant les nœuds du graphe puis lister tous les chemins avec la suite des numéros.

Exercice 3 :

Soit le code suivant : $table(i)$ dénote le i ème élément du tableau $table$, les indices variant de 1 au nombre d'éléments du tableau :

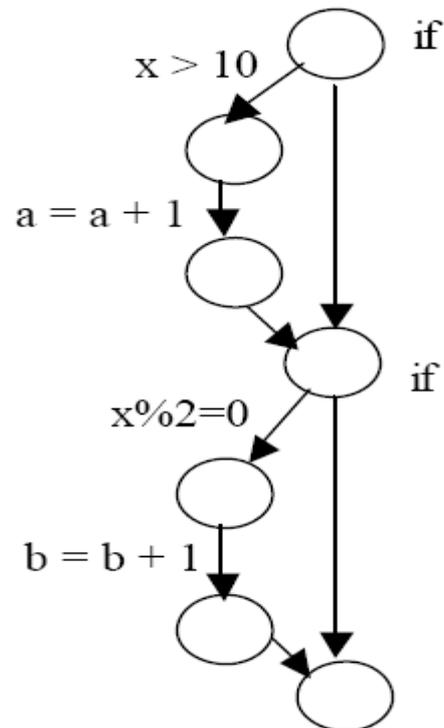
```
found := false;
if number_of_items  $\neq$  0 then counter := 1;
while ((not found) and (counter < number_of_items)) loop
if table(counter) = desired_element then
found := true;
end if;
counter := counter + 1;
end loop;
end if;
if found then
write("the desired element exists");
else
write("the desired element does not exist");
end if;
```

- 1) Ce programme est incorrect. Pourquoi ?
- 2) Construire le graphe de contrôle.
- 3) Montrer que le jeu d'essai suivant satisfait le critère de couverture des arcs du graphe de contrôle sans détecter pour autant l'erreur. $\{(number_of_items = 0), (number_of_items = 3, desired_element \text{ au rang } 2 \text{ de } table)\}$
- 4) Montrer que si l'on impose au jeu d'essai de tester toutes les combinaisons valides des termes des conditions composées (*couverture des conditions*) on peut trouver l'erreur.

Corrigé de la série TD N° 7

Exercice 1:

Graphe de contrôle :

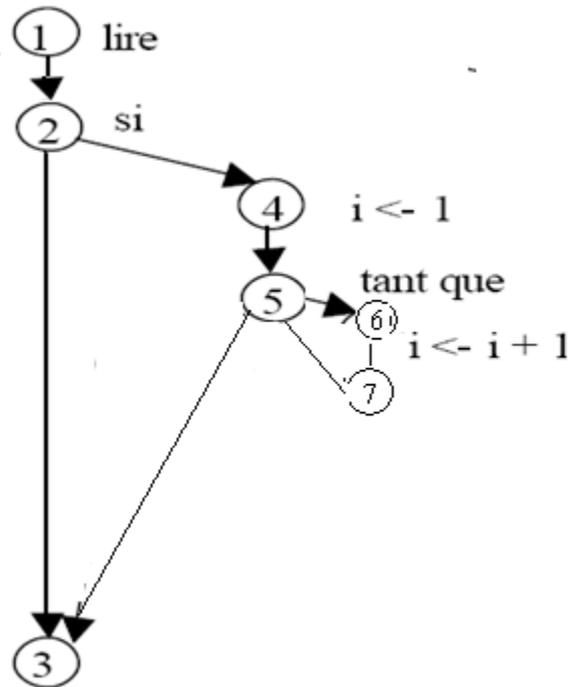


Couverture des instructions $\{x=12\}$.

Couverture des arcs $\{x=5, x=12\}$.

Couverture des chemins $\{x=5, x=6, x=11, x=12\}$.

Exercice 2 :



3 chemins possibles :

1 2 3, 1 2 4 5 3, 1 2 4 5 6 7 5 3

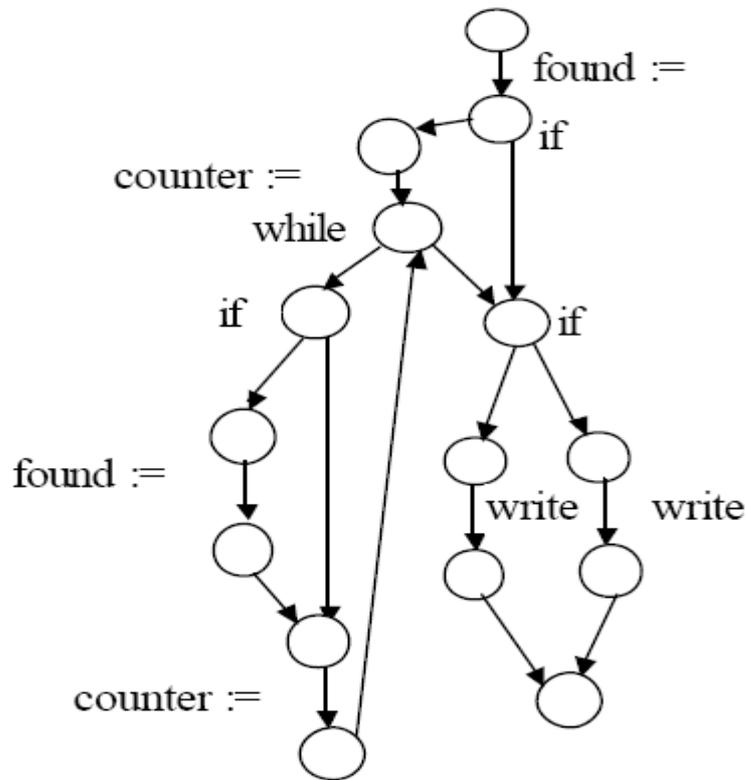
jeu d'essai correspondant :

vide a, a b, a a (aa ab ou bien aa aa impossible car max longueur = 1),

Exercice 3 :

1)

C'est une erreur classique de fin de parcours de tableau. Quand les indices commencent à 1, la boucle doit se terminer avec un \leq nb éléments du tableau. Il faudrait avoir : `while ((not found) and (counter \leq number_of_items)) loop ...`



2)

avec `number_of_items = 0` on passe par les deux branches **else** des **if** en dehors du **while**; avec `number_of_items = 3` et `desired_element` au rang 2 de la table on boucle deux fois en passant une fois dans la branche **then** et une fois dans la branche **else** du **if** dans le **while** et on termine par la branche **then** du **if** final. On couvre donc bien tous les arcs.

3)

Le problème est que seule la partie **not found** de la condition du **while** est testée (la première fois avec la valeur **true** et la seconde fois avec la valeur **false**). La partie `counter < number_of_items` est seulement testée à **true**.

Pour tester à faux cette deuxième partie il faut un test comme (`number_of_items = 3`, `desired_element` au rang 3 de la table). Avec ce test, on s'aperçoit que le programme ne trouve pas l'élément à cause de l'erreur de programmation.

Exercices de perfectionnement

Exercice 1 :

On souhaite qu'un utilisateur se connecte à un serveur (le système étudié) par des protocoles comme FTP ou telnet

- Le protocole FTP permet le transfert des fichiers, nécessite une identification.
- Le protocole telnet sert pour exécuter les commandes, nécessite une identification.
- HTTP pour le transfert des données (pages html).
- Mail permet de transférer les fichiers, nécessite une identification.

➤ *Modéliser ce système par un diagramme de cas d'utilisation.*

Exercice 2 :

Un nouveau Système de Gestion de Stock (GTS) doit être mis en place. Celui-ci sera utilisé par le service stock et doit couvrir les activités de gestion des stocks et de réapprovisionnement.

Dans le service stock, se côtoient les chefs d'équipes et les magasiniers. Les magasiniers sont responsables par la réception des marchandises et leur expédition aux autres magasins de la société. Les chefs d'équipe gèrent le réapprovisionnement : ils soumettent aux fournisseurs les demandes de devis pour les marchandises et élaborent les commandes. Par contre, tous les employés, qu'ils soient chef d'équipe ou magasinier, sont responsables de la gestion du catalogue (inclusion des nouveaux produits, mise à jour des produits, etc.). Toutes ces tâches seront gérées par le système de gestion GTS. Le processus de réapprovisionnement se déroule de la manière suivante : les magasins transmettent au service de stock leurs besoins. Le chef d'équipe vérifie si les marchandises demandées sont disponibles. Si c'est le cas, un ordre d'expédition est transmis aux magasiniers, qui vont conditionner

les produits et les expédier au magasin qui les a demandés. Si le stock est insuffisant, le chef d'équipe soumet aux fournisseurs une demande de devis. Ceux-ci répondent par un devis concernant les marchandises en question. Le chef d'équipe choisit alors le meilleur devis et élabore une commande qui est transmise au fournisseur. Lorsque la commande est honorée, les magasiniers réceptionnent les marchandises, actualisent le stock et procèdent à leur expédition aux magasins.

➤ *Modéliser ce système par un diagramme de cas d'utilisation.*

Exercice 3 :

Une entreprise souhaite modéliser avec UML le processus de formation de ses employés afin d'informatiser certaines tâches.

Le processus de formation est initialisé quand le responsable formation reçoit une demande de formation d'un employé. Cet employé peut éventuellement consulter le catalogue des formations offertes par les organismes agréés par l'entreprise. Cette demande est instruite par le responsable qui transmet son accord ou son refus à l'employé.

En cas d'accord, le responsable cherche la formation adéquate dans le catalogue des formations agréées qu'il tient à jour. Il informe l'employé du contenu de la formation et lui soumet la liste des prochaines sessions prévues. Lorsque l'employé fait son choix il inscrit l'employé à la session retenue auprès de l'organisme de formation concerné.

En cas d'empêchement l'employé doit avertir au plus vite le responsable formation pour que celui-ci demande l'annulation de l'inscription.

A la fin de la formation l'employé transmet une appréciation sur le stage suivi et un document attestant sa présence.

Le responsable formation contrôle la facture envoyée par l'organisme de formation.

➤ *Modéliser ce système par un diagramme de cas d'utilisation.*

Exercice 4 :

Le client d'un distributeur automatique de produits courants peut y trouver des produits alimentaires (pains, conserves, boissons, etc.) ainsi que d'autres types de produits courants (lessives, savons, etc.). Une fois qu'il a choisi les produits qu'il désire acheter, il doit ensuite payer ses achats. Il existe deux façons de payer les produits : soit en espèces soit par carte de crédit. Lors de l'achat d'un produit alimentaire et uniquement dans ce cas, le client vérifie la date de limite de consommation du produit.

➤ *Proposer un diagramme de cas d'utilisation pour la description ci-dessus.*

Exercice 5 :

Cet exercice concerne un système simplifié de caisse enregistreuse de supermarché. Le déroulement normal d'utilisation de la caisse est le suivant :

- Un client arrive à la caisse avec des articles à payer
- Le caissier enregistre le numéro d'identification (CPU) de chaque article, ainsi que la quantité si elle est supérieure à 1.
- La caisse affiche le prix de chaque article et son libellé.
- Lorsque tous les achats sont enregistrés, le caissier signale la fin de la vente.
- La caisse affiche le total des achats.
- Le client choisit son mode de paiement :
 - numéraire : le caissier encaisse l'argent reçu, la caisse indique l'argent à rendre au client.
 - chèque : le caissier vérifie la solvabilité du client en transmettant une requête à un centre d'autorisation via la caisse.
 - carte de crédit : un terminal bancaire fait partie de la caisse. Il transmet une demande d'autorisation à un centre d'autorisation en fonction du type de la carte.

- La caisse enregistre la vente et imprime un ticket.
- Le caissier donne le ticket de caisse au client.

Après la saisie des articles, le client peut présenter au caissier des coupons de réduction pour certains articles. Lorsque le paiement est terminé, la caisse transmet les informations sur le nombre d'articles vendus au système de gestion des stocks. Tous les matins, le responsable du magasin initialise les caisses pour la journée.

➤ *Proposer un diagramme de cas d'utilisation pour la description ci-dessus.*

Exercice 6 :

Indiquer les instanciations, généralisations, spécialisations, agrégations, associations présentes dans les bouts de phrases suivants :

- a) Ahmed est un restaurateur.
- b) il tient le restaurant « Mondial » qui est un restaurant.
- c) un restaurateur est une personne,
- d) donc Ahmed est une personne.
- e) un repas complet est composé d'une entrée, d'un milieu et d'un dessert.
- f) un plat est une entrée ou un milieu ou un dessert.
- g) la table 3 est une table du restaurant ;
- i) un jus est une boisson ;
- j) une boisson est soit du jus, soit de l'eau, soit du café.

Exercice 7 :

Il s'agit d'établir le schéma des données pour la gestion des formations d'un institut privé. Un cours est caractérisé par un numéro de cours (NOCOURS), un libellé (LIBELLE), une durée en heures (DUREE) et un type (TYPE). Un cours peut faire l'objet dans l'année de plusieurs sessions. Une session est caractérisée par un numéro (NOSES), une date de début (DATE) et un prix (PRIX). Une session est

le plus souvent assurée par plusieurs animateurs et est placée sous la responsabilité d'un animateur principal. Un animateur peut intervenir dans plusieurs sessions au cours de l'année. On désire mémoriser le nombre d'heures (NBH) effectué par un animateur pour chaque session.

Un animateur est caractérisé par un numéro (NOANI), un nom (NOMA) et une adresse (ADRA).

Chaque session est suivie par un certain nombre de participants. Un participant est une personne indépendante ou un employé d'une entreprise cliente. Un participant est caractérisé par un numéro (NO-PAR), un nom (NOMP) et une adresse (ADRP). Dans le cas d'un employé, on enregistre le nom (NOM-EN) et l'adresse de l'entreprise (ADREN).

- *Établir un schéma conceptuel (diagramme de classes UML) pour cette base de données.*

Exercice 8 :

On souhaite implanter une base de données pour gérer le fonctionnement des enseignements et des examens d'une université.

Un cours est caractérisé par son titre, l'amphithéâtre dans lequel il se déroule, le jour et l'heure auxquels il se déroule et le nombre total d'heures de cours.

Chaque cours est assuré par un enseignant et un seul. Un enseignant peut par contre assurer plusieurs cours. Un enseignant est caractérisé par un numéro de poste, son nom, son prénom et son statut.

Un cours donné est associé à un certain nombre de TD caractérisés par un numéro, la salle dans laquelle ils se déroulent, le jour et l'heure auxquels ils se déroulent et le nombre total d'heures de TD. Un TD dépend d'un cours et un seul. Un TD est susceptible d'être assuré par plusieurs enseignants. Un enseignant peut assurer plusieurs TD.

Chaque cours fait partie d'une UE et une seule. Une UE regroupe plusieurs cours. Une UE est caractérisée par un code et un intitulé.

Chaque UE est associée à un diplôme et un seul. Un diplôme comporte plusieurs UE. Un diplôme est caractérisé par un numéro et un intitulé.

Les étudiants sont caractérisés par leur numéro d'étudiant, leur nom, leur prénom et leur date de naissance et leur adresse. Chaque étudiant est inscrit dans un diplôme et un seul. Un diplôme accueille plusieurs étudiants.

Les étudiants font également partie d'un bloc et d'un seul, sauf les dispensés de contrôle continu. Un bloc accueille plusieurs étudiants. Un bloc n'est caractérisé que par un numéro. Un bloc peut être réparti sur plusieurs TD (demi-groupes, par exemple), mais un TD donné n'est associé qu'à un bloc d'étudiants.

Les étudiants passent dans l'année plusieurs examens. Ils reçoivent une note pour chaque examen. Chaque examen est passé par plusieurs étudiants. Un examen est caractérisé par une année, un numéro de session, un lieu, une date, une heure et un coefficient.

Un examen est associé à un cours et un seul, un cours peut être associé à plusieurs examens.

➤ *Proposer un diagramme de classe modélisant les spécifications ci-dessus.*

Exercice 9 :

Une entreprise de fabrication et de distribution de matériels possède une usine et plusieurs lieux de stockage /expédition.

Un produit est caractérisé par un numéro (NOP), un libellé (LIB) et un prix unitaire (PU).

Chaque produit peut être stocké dans un ou plusieurs dépôts. Un dépôt est caractérisé par un numéro (NOD). Dans chaque dépôt, on connaît la quantité en stock de chaque produit (QTS) et la quantité disponible (QTD) (la différence

représente la quantité réservée pour des commandes déjà validées mais non livrées).

Un client est déterminé par son numéro (NOCLI), son nom (NOM), son adresse (ADR), le total de son chiffre d'affaire (CA) et le taux de réduction (RED). Chaque client est livré à partir d'un dépôt privilégié, ou à partir d'un dépôt de secours en cas de défaillance du premier.

A un client peuvent être associées une ou plusieurs commandes, chacune étant caractérisée par un numéro (NOCOM) et une date (DAC). Une ligne comporte une quantité commandée (QTC), un délai de livraison (DEL) et un code de livraison (CL) indiquant si la livraison est intervenue.

A chaque commande peuvent être associées une ou plusieurs factures, une facture étant élaborée dès qu'une livraison est intervenue. Une facture est caractérisée par un numéro de facture (NOF), une date (DAF), un montant(MOF). Une facture peut concerner plusieurs produits. Chaque ligne comprend la quantité facturée(QTF) et le montant correspondant (MOP)

➤ *Proposer un diagramme de classes modélisant les spécifications ci-dessus.*

Exercice 10 :

Le gestionnaire d'un complexe sportif associatif souhaite informatiser la gestion des réservations de terrains et d'équipements sportifs.

Un terrain est caractérisé par un numéro de terrain, un nom, un type (salle ou extérieur) et une superficie. Un terrain peut permettre la pratique de plusieurs sports (par exemple, hand-ball, basket-ball et volley-ball dans un gymnase), éventuellement simultanément. Afin de gérer cela, on souhaite savoir pour chaque terrain, combien de « sous-terrains » il contient pour un sport donné (par exemple, il peut y avoir 3 terrains de tennis dans un gymnase).

Un sport est identifié par un code ainsi que l'intitulé complet de la discipline en question.

Divers équipements sont également à la disposition des adhérents de l'association (ballons, raquettes, dossards², etc.). Ces équipements sont répartis en lots numérotés. Chaque lot est caractérisé par le type d'équipement (ballons, haies³, etc.) et le nombre d'unités qui le composent.

Pour des raisons de stockage, un lot d'équipement donné est lié à un terrain et un seul. Il ne peut pas être utilisé ailleurs. Finalement, un lot d'équipement peut être dédié à un seul sport (raquette de tennis), mais pas nécessairement (dossards).

Les adhérents sont identifiés par un numéro d'adhérent, un nom, un prénom, une adresse et un numéro de téléphone. Lorsqu'ils passent une réservation, on note le jour désiré, l'heure de début et l'heure de fin. Une réservation concerne un terrain, un sport précis et éventuellement divers équipements.

➤ *Proposer un diagramme de classes modélisant les spécifications ci-dessus.*

Exercice 11 :

Le propriétaire d'un cirque souhaite informatiser une partie de la gestion de ses spectacles. *Proposer un diagramme de classes qui réponde aux spécifications, fournies ci-dessous.*

Les membres du personnel du cirque sont caractérisés par un numéro (en général leur numéro INSEE), leur nom, leur prénom, leur date de naissance et leur salaire.

On souhaite de surcroît stocker les pseudonymes des artistes et le numéro du permis de conduire des chauffeurs de poids lourds.

Les artistes sont susceptibles d'assurer plusieurs numéros, chaque numéro étant caractérisé par un code, son nom, le nombre d'artistes présents sur scène et sa durée. De plus, on souhaite savoir l'instrument utilisé pour les numéros musicaux,

² Dossard : pièce de tissu que les participants à une épreuve sportive portent en général sur le dos, et qui indique leur numéro d'ordre.

³ Haie : barrière, en bois et en métal, qui constitue un obstacle dans certaines courses à pied.

l'animal concerné par les numéros de dressage et le type des acrobaties (contorsionnisme, équilibrisme, trapèze volant, etc.).

Par ailleurs, chaque numéro peut nécessiter un certain nombre d'accessoires caractérisés par un numéro de série, une désignation, une couleur et un volume.

On souhaite également savoir, individuellement, quels artistes utilisent quels accessoires.

Enfin, les accessoires sont rangés après chaque spectacle dans des camions caractérisés par leur numéro d'immatriculation, leur marque, leur modèle et leur capacité (en volume). Selon la taille du camion, une équipe plus ou moins nombreuse de chauffeurs lui est assigné (de un à trois chauffeurs).

Exercice 12 :

Une académie souhaite gérer les cours dispensés dans plusieurs collèges. Pour cela, on dispose des renseignements suivants :

- Chaque collège possède d'un site Internet
- Chaque collège est structuré en départements, qui regroupent chacun des enseignants spécifiques. Parmi ces enseignants, l'un d'eux est responsable du département.
- Un enseignant se définit par son nom, prénom, tél, mail, date de prise de fonction et son indice.
- Chaque enseignant ne dispense qu'une seule matière.
- Les étudiants suivent quant à eux plusieurs matières et reçoivent une note pour chacune d'elle.
- Pour chaque étudiant, on veut gérer son nom, prénom, tél, mail, ainsi que son année d'entrée au collège.

- Une matière peut être enseignée par plusieurs enseignants mais a toujours lieu dans la même salle de cours (chacune ayant un nombre de places déterminé).
- On désire pouvoir calculer la moyenne par matière ainsi que par département
- On veut également calculer la moyenne générale d'un élève et pouvoir afficher les matières dans lesquelles il n'a pas été noté.
- Enfin, on doit pouvoir imprimer la fiche signalétique (nom, prénom, tél, mail) d'un enseignant ou d'un élève.

➤ *Élaborez le diagramme de classes correspondant.*

Pour simplifier l'exercice, on limitera le diagramme à une seule année d'étude.

Exercice 13 :

On souhaite modéliser le fonctionnement d'un SGBDOO (Système de Gestion de Base de Données Orienté Objets). Dans un tel système, une base de données est caractérisée par un nom de base de données, un propriétaire et une date de création. Plusieurs classes composent cette base de données. Chaque classe possède un nom de classe. Les classes sont liées entre elles par un lien d'héritage, chaque classe pouvant hériter de plusieurs classes mères. Chaque classe est composée d'objets caractérisés par un oid (object identifier) et une taille. Un objet appartient à une et une seule classe. Les classes sont stockées dans des pages du disque dur à une date donnée. Une classe peut être stockée sur plusieurs pages et une page peut accueillir plusieurs classes. Chaque page est caractérisée par un numéro de page et l'espace libre dans cette page. Finalement, chaque page appartient à un segment du disque dur. Un segment est caractérisé par un identifiant segment, un numéro de disque et un numéro de cylindre. Un segment contient plusieurs pages.

➤ *Proposer un diagramme de classes modélisant les spécifications ci-dessus.*

Exercice 14 :

Un vol est un parcours aérien caractérisé par un numéroVol, une ville-départ, une ville-d'arrivée, une heure-départ, une heure-d'arrivée, une distance et une fréquence.

Lorsqu'un vol est programmé pour une date déterminée il constitue un départ. Un vol n'est programmé qu'une seule fois dans une journée à l'heure prévue. Un certain nombre de passagers peut être enregistré pour un départ. Un passager est caractérisé par son nom, son adresse et son n°-téléphone. Un avion est affecté à chaque départ. Un avion est caractérisé par un numéro, un type et une capacité. Un avion utilise une certaine quantité de carburant pour accomplir le trajet. Cette dernière dépend des conditions atmosphériques, donc de la date. Un certain nombre de personnels est affecté à chaque départ. On distingue les personnels non-navigants des personnels navigants. Parmi ces derniers, on distingue le(s) pilote(s). Un membre du personnel est caractérisé par son nom, son adresse et son n°-téléphone.

➤ *Proposer un diagramme de classes modélisant les spécifications ci-dessus.*

Exercice 15 :

On désire automatiser la gestion d'une petite bibliothèque municipale. Pour cela, on a analysé son fonctionnement pour obtenir la liste suivante de règles et d'affirmations :

- Les adhérents ont un prénom et un nom.
- La bibliothèque comprend un ensemble de documents et un ensemble d'adhérents.
- Les adhérents sont inscrits ou désinscrits sur une simple demande.
- De nouveaux documents sont ajoutés régulièrement à la bibliothèque.
- Ces documents sont soit des journaux, soit des volumes.
- Les volumes sont soit des dictionnaires, soit des livres, soit des Bd (Bande dessinée).

- Les documents sont caractérisés par un titre.
- Les volumes ont en plus un auteur. Les Bd ont en plus un nom de destinataire.
- Les journaux ont, outre les caractéristiques des documents, une date de parution.
- Seuls les livres sont empruntables.
- Un adhérent peut emprunter ou restituer un livre.
- Les adhérents peuvent emprunter des livres (et uniquement des livres) et on doit pouvoir savoir à tout moment quels sont les livres empruntés par un adhérent.
- Un adhérent peut emprunter au plus 3 livres.
- La date de restitution d'un livre emprunté est fixée au moment du prêt. Cette date peut être prolongée sur demande.

- 1) *Réaliser le diagramme de classes permettant d'automatiser la bibliothèque municipale.*
- 2) *Définissez les attributs et les méthodes de chaque classe de ce diagramme, ainsi que le type et les multiplicités (cardinalités) des associations entre les classes.*

Exercice 16 :

Une chaîne de restaurants d'entreprise souhaite informatiser divers aspects de son activité. Proposer un diagramme de classes modélisant les spécifications ci-dessous.

Les clients des restaurants sont caractérisés par leur numéro de carte de restauration, leur nom, leur prénom et par un code de tarification dépendant de leur rémunération. Chaque client appartient également à une entreprise caractérisée par son numéro de SIRET⁴ et sa raison sociale (nom de l'entreprise). Les clients consomment des repas à une date donnée. Un repas donné n'est consommé que

⁴ SIRET (Système Informatique pour le Répertoire des Entreprises sur le Territoire): numéro qui identifie l'établissement de l'entreprise.

par un seul client. Un repas occupe une table, mais une table donnée peut accueillir plusieurs repas. On souhaite également stocker l'heure d'arrivée et l'heure de départ des clients d'un repas sur la table correspondante. Une table est caractérisée par sa capacité (en sièges). Finalement, chaque repas est payé à une caisse donnée. Un restaurant est caractérisé par un code, un nom et une adresse (rue et ville). Il est organisé en îlots⁵ dédiés chacun à une spécialité (grill, poisson, pizzas,...) et caractérisés par un numéro unique dans l'entreprise de restauration et un emplacement. Divers plats sont servis sur les îlots. Chaque plat est caractérisé par un code, un nom, un apport énergétique et un prix de base. Chaque repas est constitué d'au moins un plat. Les plats sont finalement classés en catégories (entrée, dessert, ...) caractérisées par un code et un libellé.

Exercice 17 :

L'université comporte des personnels administratifs et techniques, des enseignants, des étudiants et des chercheurs (qui sont tous des personnes). Certains étudiants peuvent être des chercheurs (les doctorants) ou des enseignants (les assistants enseignants). Certaines personnes (étudiants ou non) peuvent être à la fois chercheurs et enseignants.

➤ *Proposer un diagramme de classes qui répond à la description fournie ci-dessus.*

Exercice 18 :

Une bibliothèque compte les titres suivants parmi les livres dont elle dispose : «History of World War 2», «The Adventures of Robin Hood», et deux exemplaires de « Harry Potter and the Philosopher's Stone». Samy et Nihal sont des utilisateurs abonnés. Nihal a emprunté «The Adventures of Robin Hood» tandis que Samy a emprunté deux livres : «History of World War 2 » et une copie de

⁵ Ilot: petit comptoir entouré de plancher.

«Harry Potter and the Philosopher's Stone».

➤ *Dessiner un diagramme d'objets représentant les objets (instances de classe) et les liens (instances d'association) de l'énoncé ci-dessus.*

Exercice 19 :

Une banque compte plusieurs agences réparties sur le territoire d'un pays. Une banque est caractérisée par le nom de son directeur général, son capital global, son propre nom et de l'adresse de son siège social. Le directeur général est identifié par son nom, son prénom et son revenu.

Une agence a un numéro d'agence et une adresse. Chaque agence emploie plusieurs employés, qui se caractérisent par leur nom, prénom et date d'embauche. Les employés peuvent demander leur mutation d'une agence à une autre, mais un employé ne peut travailler que dans une seule agence. Les employés d'une agence ne font que gérer des clients.

Un client ne peut avoir des comptes que dans une seule agence de la banque. Chaque nouveau client se voit systématiquement attribuer un employé de l'agence (conseiller). Les clients ont un nom, un prénom et une adresse. Les comptes sont de nature différente selon qu'ils soient rémunérés ou non (comptes courants). Les comptes rémunérés ont un taux d'intérêt et rapportent des intérêts versés annuellement.

1. *Donner la description complète de toutes les classes (remplissez tous les compartiments). Préciser les types des attributs et les types de retour des fonctions. Les attributs sont tous privés. Chaque attribut possède deux méthodes publiques (getAttribut renvoie la valeur d'un attribut et setAttribut affecte une nouvelle valeur à un attribut). Toutes les autres méthodes sont accessibles uniquement dans le package de la classe.*
2. *Analyser les classes trouvées en (1) et modélisez-les en factorisant (par généralisation ou autre) au mieux la description des propriétés.*
3. *Une relation particulière lie l'agence, le client, l'employé et le compte. De*

quelle relation s'agit-il ? Dessiner le modèle de cette relation.

4. *Donner le diagramme de classes en n'utilisant que leur nom et ajoutez tous les ornements possibles aux relations.*

Exercice 20 :

Le processus de réservation d'une chambre d'hôtel est décrit comme suit :

- L'hôtelier peut consulter la disponibilité des chambres selon certains critères (date, nombre de personnes).
- A partir de là il peut sélectionner la ou les chambres réservées et enregistrer les dates de réservation.
- Ensuite, il vérifie si le client est déjà client de son hôtel. Si ce n'est pas le cas, il enregistre les informations concernant le client.

L'hôtelier peut réserver une ou plusieurs chambres pour un client, modifier et annuler une réservation. La modification ou l'annulation d'une réservation se fait suite à la demande du client. La trace des annulations doit être gardée.

L'hôtelier peut aussi facturer un séjour. Le prix de l'occupation des chambres est calculé en fonction du type de chambre, de la durée et du nombre d'occupants.

L'hôtelier peut consulter la fiche d'un client, c'est-à-dire l'état de sa ou ses réservations. Réaliser les diagrammes suivants :

- *Le diagramme de cas d'utilisation*
- *Le diagramme de séquence du scénario nominal du cas d'utilisation « réserver une chambre »*
- *Le diagramme d'activités décrivant l'ensemble des scénarios du cas d'utilisation « réserver une chambre »*
- *Le diagramme de séquence du scénario nominal du cas d'utilisation « facturer un séjour »*

- *Le diagramme d'activités décrivant l'ensemble des scénarios du cas d'utilisation « facturer un séjour »*

Exercice 21 :

On dispose d'une machine à café qui propose des expressos à partir d'eau et de capsules pré-remplie de café. La machine à café possède 3 boutons :

- Un bouton ON pour mettre la machine à café sous tension.
- Un bouton OFF pour éteindre la machine à café à tout moment.
- Un bouton lumineux :
 - Au démarrage, il est clignotant tant que l'eau n'est pas sous pression. Si l'utilisateur appuie dessus, il ne se passe rien. Il faut attendre 1 minute pour que la machine soit prête.
 - Il est lumineux et non clignotant quand la machine est finalement sous-pression. Si l'utilisateur appuie dessus, le café coule.
 - Il est clignotant aussi au moment de servir le café. Si l'utilisateur appuie dessus, le café s'arrête de couler et on peut le déguster.

On admet que la mise sous pression peut commencer seulement si le réservoir d'eau est suffisamment plein. Quand le réservoir d'eau se vide, la machine n'a plus de pression et restera sans pression tant que le réservoir n'a pas été re-rempli.

- *Proposer un diagramme d'états-transitions pour la classe Machine à café.*

Exercice 22 :

Voici la recette pour faire une bonne mousse au chocolat :

- Commencer par casser le chocolat en morceaux, puis le faire fondre.
- En parallèle, casser les œufs en séparant les blancs des jaunes.
- Quand le chocolat est fondu, ajouter les jaunes d'œuf.
- Battre les blancs en neige jusqu'à ce qu'ils soient bien fermes.
- Les incorporer délicatement à la préparation chocolat sans les briser.
- Verser dans des ramequins individuels.

- Mettre au frais au moins 3 heures au réfrigérateur avant de servir.

➤ *Etablir le diagramme d'activités pour modéliser la recette.*

Exercice 23 :

L'entreprise MegaKebab regroupe de nombreux restaurants appelés "Points Kebab". Elle est spécialisée dans la livraison à domicile de Kebabs et autres spécialités. La direction de MegaKebab souhaite informatiser le processus de commande/fabrication/livraison via un logiciel baptisé CyberKebab. Grâce à ce logiciel, MegaKebab pourra gérer à distance et de manière centralisée toutes les commandes, les Points Kebab et les employés appelés "Collaborateurs". Tous les plats pourront être commandés sur Internet. Chaque plat est décrit par un nom, une photo et un prix. Il existe des plats chauds et des plats froids. Une durée est associée à chaque plat chaud : si le temps écoulé entre la fin de préparation et la livraison est supérieur à cette durée, le client peut se faire rembourser sa commande. Le client doit alors remplir une demande sur papier et l'envoyer au gérant de MegaKebab. Pour passer commande par Internet le client doit disposer d'une carte de crédit qui l'identifie de manière unique. Lors d'une première commande il lui est également demandé de saisir son nom et son adresse. Une même commande peut comporter plusieurs plats en quantités différentes. Après avoir passé sa commande, le client peut à tout moment consulter l'état de sa commande. Tant que la commande n'est pas partie du Point Kebab, il peut l'annuler. Pour assurer un service 24h/24 dans toute la ville, MegaKebab fait appel à un grand nombre de collaborateurs, souvent étudiants, qui ont des horaires très flexibles. Lors de leur embauche, un appareil appelé "Kepad" leur est remis. Cet appareil contient un téléphone et un GPS. Il suffit d'appuyer sur un bouton pour faire part de leur disponibilité ou de leur non disponibilité auprès de MegaKebab. A tout moment le gérant peut consulter l'état du système global et affecter un collaborateur soit à un Point Kebab soit à la livraison. C'est aussi le gérant qui affecte une commande à un Point Kebab et à un livreur. En cas de surcharge d'un Point Kebab, une affectation peut être modifiée. Un livreur consulte les commandes qui lui ont été

affectées sur son Kepad. Il peut consulter la carte et se situer par rapport aux points Kebab et aux clients à livrer. Le livreur utilise également le Kepad pour indiquer quand il récupère une commande auprès du Point Kebab et quand il livre la commande au client. Dans chaque Point Kebab un collaborateur joue le rôle de "coordinateur". C'est le seul du restaurant à agir directement avec CyberKebab : les autres collaborateurs préparent les plats. Le coordinateur consulte les commandes à réaliser et indique pour chaque commande quand sa préparation débute, quand elle se termine et quand elle est remise au livreur. Réaliser les diagrammes suivants, dans l'ordre que vous souhaitez :

- 1) *Diagramme de cas d'utilisation*
- 2) *Diagramme de classes*
- 3) *Diagramme de séquences du passage d'une commande par un nouveau client.*

Épreuve de Moyenne Durée 1

Exercice 1

1. Quels sont les avantages du modèle en cascade ?
2. Quelles sont les limites de l'approche fonctionnelle ?
3. Donner une définition des termes suivants : Objet, Classe.
4. Quel est le principal objectif de l'héritage ?
5. Citer les trois méthodes de base qui ont influencé UML.
6. A quoi correspond la notion d'acteur dans les diagrammes de cas d'utilisation ?

Exercice 2

Pour chacun des énoncés suivants, donner un diagramme de classes :

- a) Une librairie vend des livres, caractérisés par leur auteur et leur nombre de pages; certains livres possèdent également d'autres caractéristiques : une fourchette des âges pour les livres pour enfants, et la discipline et le niveau pour les livres scolaires.
- b) Un polygone est constitué de points. Un point possède une abscisse et une ordonnée.

Exercice 3

1. Donner un exemple de diagramme de cas d'utilisation avec une relation de généralisation entre deux acteurs. Expliquer en quelques phrases le diagramme que vous avez proposé.

2. Est-il possible, uniquement avec un diagramme de cas d'utilisation, de spécifier qu'un cas d'utilisation doit s'exécuter obligatoirement avant un autre cas d'utilisation ? Justifier votre réponse.

Exercice 4

Répondre par vrai ou faux aux affirmations suivantes et expliquer votre réponse :

- a) Une composition est un type d'agrégation.
- b) Un diagramme de séquence fait intervenir des classes.
- c) Un diagramme de séquence est basé sur une représentation temporelle.

Corrigé de l'EMD 1

Exercice 1

1. Avantages du modèle en cascade

- Simple à mettre en œuvre ;
- La documentation est produite à chaque étape.

2. Les limites de l'approche fonctionnelle

- Un programme est conçu comme un ensemble de modules fonctionnels (procédures ou fonctions) qui manipulent des données ;
- Communication entre fonctions par passage de paramètres ou par variables globales ;
- Accès libre aux données par n'importe quelle fonction ;
- Difficulté de réutiliser du code déjà écrit et testé ;
- etc.

3. Donner une définition des termes suivants : Objet, Classe.

Objet :

Un objet est une entité du monde réel représentée par un état et un ensemble d'opérations qui manipulent cet état.

Un objet est caractérisé par :

- un identificateur (nom de l'objet) ;
- un état (sous forme d'un ensemble d'attributs) ;
- un comportement (un ensemble d'opérations).

Classe :

Une classe correspond à la description d'une famille d'objet ayant une même structure et un même comportement.

Une classe définit, une partie statique et une partie dynamique :

- La partie statique est représentée par un ensemble d'attributs pouvant posséder une valeur (ces attributs caractérisent l'état des objets durant leurs exécutions).
- La partie dynamique est représentée par l'ensemble des opérations qui définissent le comportement commun aux objets de la même classe.

4. Quel est le principal objectif de l'héritage ?

L'objectif de l'héritage est la réutilisation d'un concept pour la spécification d'un nouveau concept. Ainsi, il permet un partage hiérarchique des propriétés.

5. Citer les trois méthodes de base qui ont influencé UML.

OMT (Object Modeling Technique)

BOOCH (de Grady Booch)

OOSE(Object Oriented Software Engineering)

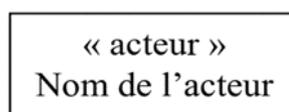
6. A quoi correspond la notion d'acteur dans les diagrammes de cas d'utilisation ?

Un acteur représente un rôle, joué par une entité externe (utilisateur humain, dispositif matériel, ou autre système), qui interagit directement avec le système étudié.

Un acteur se représente par un petit bonhomme ayant son nom inscrit dessous.

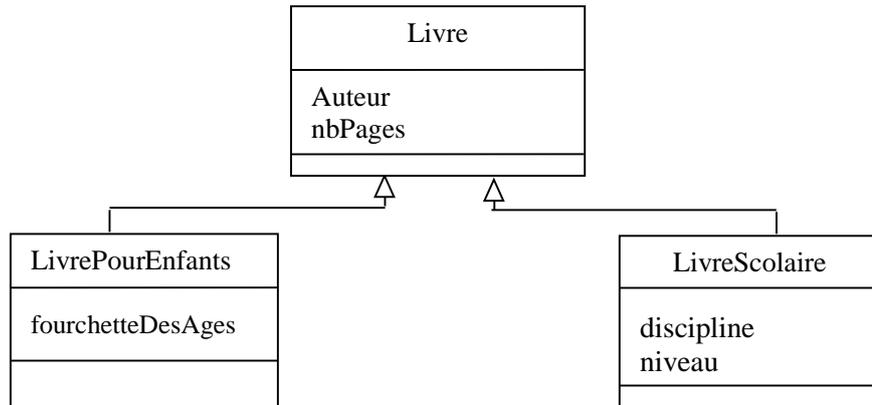


Un acteur peut aussi être formalisé par un rectangle contenant le mot clé « acteur » avec son nom juste en dessous.

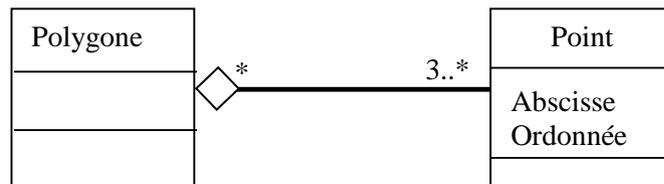


Exercice 2

a)



b)



Exercice 3

1. La bonne réponse consiste à rappeler le principe de généralisation entre acteur.
Elle englobe : Le diagramme correct et une explication textuelle correcte
2. Non, les cas d'utilisation ne s'enchaînent pas, car il n'y a aucune représentation temporelle dans un diagramme de cas d'utilisation.

Exercice 4

Répondre par vrai ou faux aux affirmations suivantes :

a) Une composition est un type d'agrégation. **Vrai**

Explication :

L'agrégation et la composition sont deux types d'association particulière qui exprime une relation d'inclusion entre une partie et un tout. Une composition est

une forme forte d'agrégation. C'est-à-dire que la suppression de l'objet agrégat mène à la suppression des objets agrégés. La cardinalité du côté composite ne doit pas être supérieure à 1 (1 ou 0..1). La composition se représente par un petit losange de couleur noire.

b) Un diagramme de séquence fait intervenir des classes. **Faux**

Explication :

Un diagramme de séquence fait intervenir des objets, parce que les interactions se produisent entre objets ou instances de classe.

c) Un diagramme de séquence est basé sur une représentation temporelle. **Vrai**

Explication :

Un diagramme de séquence montre un séquençage temporel des messages. L'ordre d'envoi d'un message est déterminé par sa position sur l'axe vertical (**la ligne de vie**) du diagramme ; le temps s'écoule « **de haut en bas** » de cet axe.

Université de Bejaia
Faculté des sciences exactes
Département d'Informatique
Module : GL
Tous documents et appareils électroniques interdits

Nom : Prénom : Groupe :

Épreuve de Moyenne Durée 2

Exercice 1

1. Citer les symptômes les plus caractéristiques de la crise du logiciel.

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

2. Quels sont les avantages du modèle en V ?

.....
.....
.....
.....
.....
.....
.....
.....
.....

3. Quelle différence y a-t-il entre un objet et un attribut ?

.....
.....
.....
.....
.....
.....
.....
.....
.....

4. Pourquoi dit-on que le polymorphisme découle directement de la notion d'héritage ?

.....
.....
.....
.....
.....
.....
.....
.....
.....

Exercice 2

Répondre par vrai ou faux aux affirmations suivantes :

- a) Un objet est une instance de classe.
- b) L'abstraction est une forme de spécialisation.
- c) L'agrégation est un type d'association.
- d) Le polymorphisme est une forme d'encapsulation.
- e) Il est possible de lier une classe à elle-même par association.
- f) Le diagramme d'objets permet la représentation d'instances des classes et des liens entre instances.
- g) Dans un diagramme de cas d'utilisation, il y a au moins un acteur et un cas d'utilisation.

h) Dans un diagramme de séquence, un objet peut envoyer un message à lui-même.

.....

i) Un diagramme de communication utilise une représentation temporelle.

j) La dernière partie de la description textuelle d'un cas d'utilisation est une rubrique optionnelle. Elle contient généralement des spécifications non fonctionnelles.

Exercice 3

Un message électronique comporte un titre ainsi que l'adresse du destinataire. Il est composé d'un en-tête et d'un corps. Il peut contenir éventuellement une ou plusieurs pièces jointes. Tracer le diagramme de classes correspondant à cette description.

Exercice 4

Modéliser avec un diagramme de cas d'utilisation le fonctionnement d'une banque qui interagit avec ses clients. Les guichetiers créent les comptes, déposent l'argent des clients dans les comptes, et peuvent aussi fermer le compte, le guichetier chef peut en plus de ceci annuler ce compte. L'opération de dépôt d'argent peut se faire de deux manières différentes : en numéraire ou par voie de chèques.

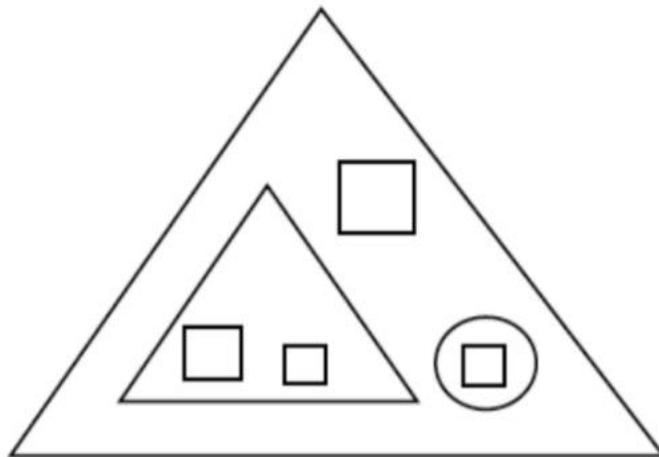
Épreuve de Moyenne Durée 3

Exercice 1

1. Donner une définition du terme « Génie Logiciel ».
2. Quelles sont les différences entre l'approche fonctionnelle et l'approche orientée objet ?
3. Est-il possible de lier une classe à elle-même par association ? Justifier.
4. Que représente la multiplicité au sein d'une association ?
5. Que décrit un diagramme de séquence ?

Exercice 2

Le dessin ci-dessous représente des figures (triangles, carrés ou cercles) emboîtés. Les triangles contiennent une ou plusieurs figures. Les carrés ne contiennent rien. Les cercles contiennent exactement une figure. Les figures possèdent des côtés. On dira que les cercles ont un seul côté, les triangles trois côtés et les carrés quatre côtés.



- Donner le diagramme de classes.

Exercice 3

Un système informatique doit permettre à des acheteurs potentiels de préparer l'achat de chevaux (mais pas l'achat proprement dit). L'achat d'un cheval concerne soit une jument soit un étalon. Dans le premier cas, on doit impérativement examiner l'état de maternité du cheval, et éventuellement vérifier que la jument n'a pas un jeune poulain en ce moment. Que l'on souhaite acheter un étalon ou une jument, on doit effectuer un examen des vaccinations. En outre l'acheteur peut souhaiter, lors de la préparation de son achat, consulter le caractère du cheval ou bien en connaître la robe. Toutes les informations en rapport à la filiation d'un cheval sont obtenues en consultant la base de données externe des haras nationaux.

- Donner un diagramme de cas d'utilisation modélisant le système de préparation d'achat.

Corrigé de l'EMD 3

Exercice 1

1. Donner une définition du terme « Génie Logiciel ».

Le terme Génie Logiciel (en anglais software engineering) désigne l'ensemble des méthodes, des techniques et outils contribuant à la production d'un logiciel de qualité avec maîtrise des coûts et délais.

2. Quelles sont les différences entre l'approche fonctionnelle et l'approche orientée objet ?

L'approche fonctionnelle

- Raisonnement en termes de fonctions du système ;
- Séparation des données et du code de traitement ;
- Décomposition fonctionnelle descendante.

L'approche orientée objet

- Regroupement données-traitements ;
- Diminution de l'écart entre le monde réel et sa représentation informatique ;
- Décomposition par identification des relations entre objets.

3. Est-il possible de lier une classe à elle-même par association ? Justifier.

Oui il est possible de lier une classe à elle-même par association. Lorsque le lien existe entre des objets de la même classe, on parle d'association réflexive.

4. Que représente la multiplicité au sein d'une association ?

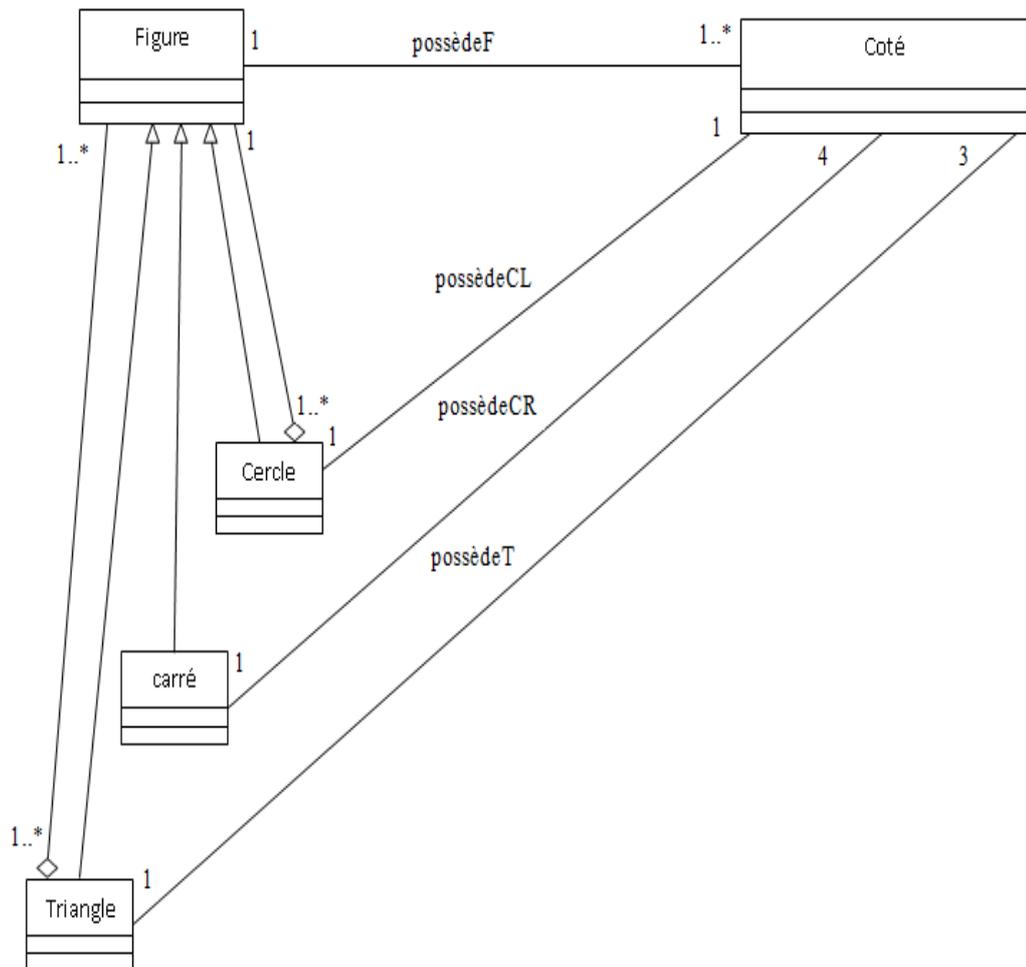
La multiplicité indique un domaine de valeurs pour préciser le nombre d'instance d'une classe vis-à-vis d'une autre classe pour une association donnée.

5. Que décrit un diagramme de séquence ?

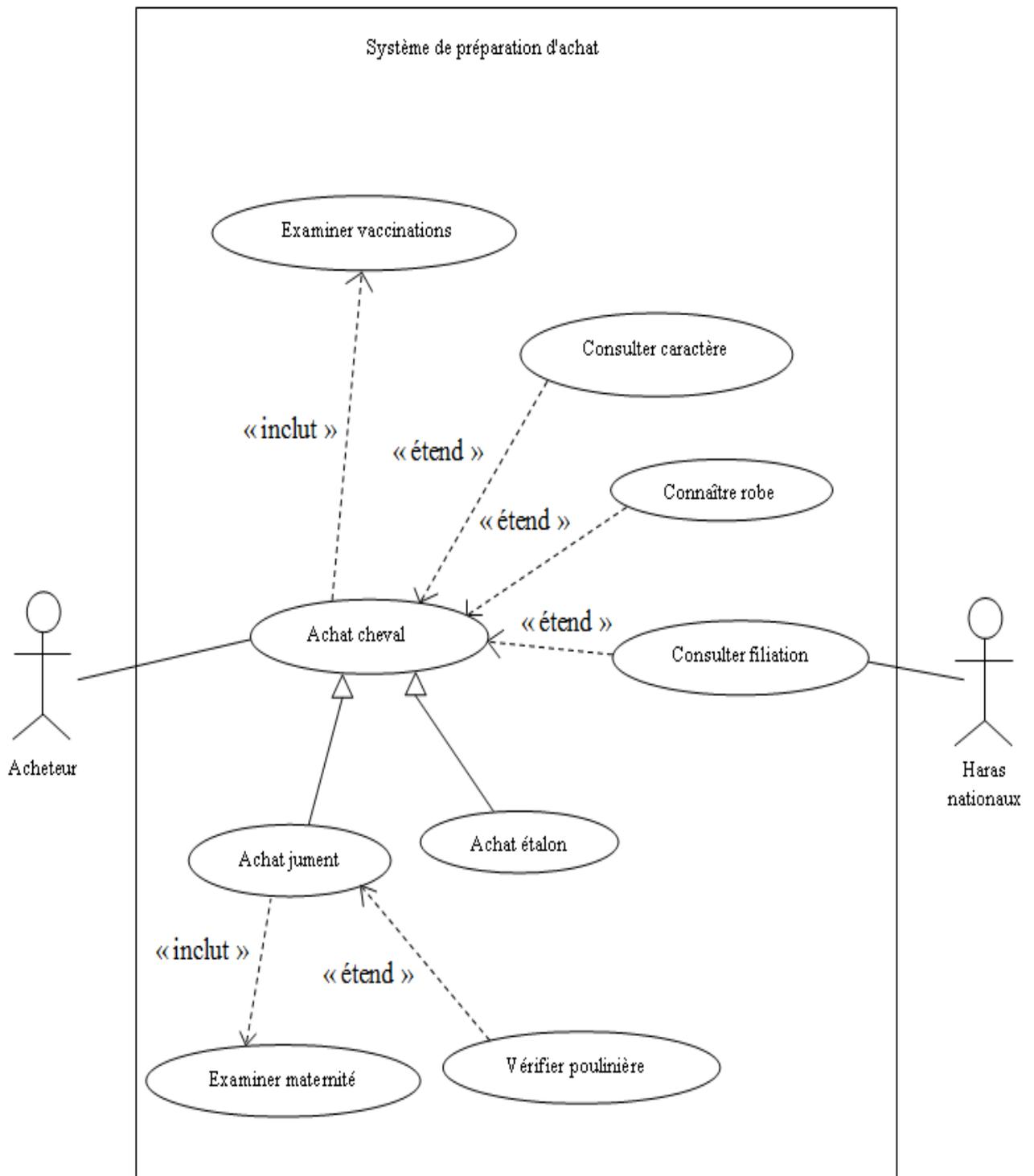
Le diagramme de séquence représente les interactions entre objets en indiquant la chronologie des échanges.

Le diagramme de séquence peut être aussi utilisé pour documenter un cas d'utilisation.

Exercice 2



Exercice 3



Université de Bejaia
Faculté des sciences exactes
Département d'Informatique
Module : GL
Documents personnels non autorisés

Nom : Prénom : Groupe :

Examen de Rattrapage

Exercice 1

1. Expliquer le rôle de cycle de vie d'un logiciel.

.....
.....
.....
.....
.....
.....
.....
.....
.....

2. Quelle est l'importance de l'analyse des besoins dans la modélisation d'un système

.....
.....
.....
.....
.....
.....
.....
.....
.....

3. Quels sont les avantages du modèle en V ?

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

Exercice 2

1. Quelles sont les différences entre l'approche fonctionnelle et l'approche orientée Objets ?

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

2. Pourquoi dit-on que le polymorphisme découle directement de la notion d'héritage ?

.....
.....
.....
.....
.....
.....
.....
.....
.....

.....
.....
.....
.....
.....
.....

Exercice 3

Répondre par vrai ou faux aux affirmations suivantes :

- a) Un objet est une instance de classe.
- b) L'abstraction est une forme de spécialisation.
- c) Le diagramme d'objets permet la représentation d'instances des classes et des liens entre instances.
- d) La dernière partie de la description textuelle d'un cas d'utilisation est une rubrique optionnelle. Elle contient généralement des spécifications non fonctionnelles.
- e) Pour effectuer des tests unitaires, il faut absolument avoir une parfaite connaissance du langage de programmation utilisé pour implémenter l'unité à tester.
- f) Les méthodes de tests statiques consistent en l'analyse textuelle du code du logiciel afin d'y détecter des erreurs, sans exécution du programme.

Exercice 4

Un document comprend une table de matières, un index et plusieurs chapitres. Chaque chapitre comprend plusieurs paragraphes. Le document peut utiliser jusqu'à 3 polices de caractères.

Si un document est supprimé, les chapitres, les paragraphes, l'index et la table des matières le sont aussi. Les images de chaque paragraphe illustrent le contenu et sans ce dernier, elles n'ont pas de sens.

Le chapitre est identifié par un numéro, l'index par un nom, la table des matières par un nom, le paragraphe par un numéro qui débute à 1 et se termine avec le dernier paragraphe du chapitre.

La police est identifiée par un numéro de police de caractère.

- Proposer un diagramme de classes modélisant les spécifications ci-dessus.

Conclusion

Ce cahier d'exercices offre aux étudiants un accompagnement pédagogique important en vue d'assembler leurs connaissances dans le domaine du génie logiciel.

Le guide de référence présenté dans ce document permet de rappeler les concepts de base du module génie logiciel. Il offre une explication concise des cycles de vie logiciel, des principales notations UML et des jeux de test logiciel.

La première série de TD présente des exercices liés au cycle de vie de logiciel. Les cinq séries d'après (2, 3, 4, 5, 6) présentent des exercices permettant à l'étudiant de se familiariser avec la modélisation orientée objet en utilisant le langage UML. La série de TD 7 est dédiée aux tests de logiciel. De plus, un ensemble d'exercices de perfectionnement et sujets d'examen sont mis pour donner l'occasion à l'étudiant de tester les connaissances acquises.

Ce document peut être utilisé, en plus des étudiants licence informatique, par les étudiants de fin de cycle préparant un projet logiciel et aussi les enseignants voulant découvrir ou enseigner le module Génie Logiciel.

Bibliographie

- [1] G. Booch, I. Jacobson, J. Rumbaugh, *Le guide de l'utilisateur UML*, éditions Eyrolles, 2001.
- [2] B. Charroux, A. Osmani, Y. Thierry-Mieg, *UML 2, pratique de la modélisation*, éditions Synthex, 2010.
- [3] L. Djakhdjakha, *Outils de modélisation des Systèmes d'Information (SI)*, Support de cours, Université de Guelma, Algérie, pp. 71, 2018.
- [4] J. Gabay, D. Gabay, *UML 2 Analyse et conception, Mise en œuvre guidée avec études de cas*, éditions Dunod, 2008.
- [5] P. Gérard, *UML*, Cours en ligne, Université Paris 13, France, 2013. <https://lipn.univ-paris13.fr/~gerard/uml-s2/>
- [6] D. Gustafson, *Génie Logiciel*, éditions Dunod, 2003.
- [7] L. HAMZA, *Génie Logiciel*, Support de cours, Université de Bejaia, 2019. <https://elearning.univbejaia.dz/course/view.php?id=5958>
- [8] J. Longchamp, *Génie Logiciel*, cours de CNAM, 2006.
- [9] P-A. Muller, N. Gaertner, *Modélisation objet avec UML*, éditions Eyrolles, 2003.
- [10] P. Roques, *UML par la pratique : études de cas et exercices corrigés*, éditions Eyrolles, 2009.
- [11] F. Van der Heyde, L- Debrauwer, *UML 2 - Initiation, exemples et exercices corrigés*, 3^e édition ENI, 2012.
- [12] <https://www.omg.org/>