

TP n°1 en Atelier Génie logiciel

Notions abordées : Cycle de vie du logiciel, les modèles de développement

Exercice 1: Les modèles de développement du logiciel permettent d'organiser les différentes phases du cycle de vie du logiciel pour l'obtention d'un logiciel fiable, adaptable et efficace.

a) Décrire les principaux modèles classiques de développement du logiciel, montrer leurs principales forces et faiblesses.

b) Indiquer la ou les phases du cycle de vie d'un logiciel où est produit chacun des documents suivants :

Manuel utilisateur final, conception architecturale, spécifications des modules, code source, cahier des charges, plan de test, conception détaillé, estimation des coûts, rapport des tests, documentation.

Exercice 2: Classer les tâches suivantes selon le modèle en cascade :

tests d'acceptation, organisation du projet, test unitaires, synthèse des exigences, estimation des coûts, conception de haut niveau, étude de marché, conception de bas niveau, test système, synthèse sur la conception, implémentation, spécification des exigences.

Exercice 3: Un concepteur souhaite décrire l'architecture des codes source, des bibliothèques, des différents fichiers exécutables ainsi que les liens entre eux, lors du développement d'un logiciel, alors il doit utiliser un diagramme de : séquence, package, composant, déploiement.

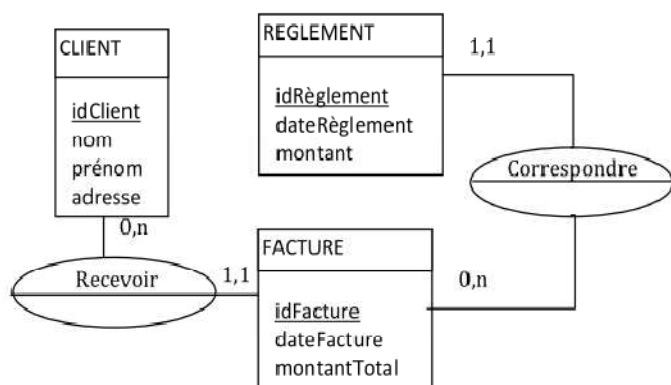
Exercice 4: Quel énoncé est faux concernant la méthode de développement logiciel UP ?

- a) C'est une méthode itérative et incrémentale
- b) C'est une méthode pilotée par les risques
- c) C'est une méthode conduite par les cas d'utilisation
- d) Aucune de ces réponses

Exercice 5: Décrire en quelques lignes les phases que vous suivez lorsque vous êtes seul à développer un logiciel (une application informatique).

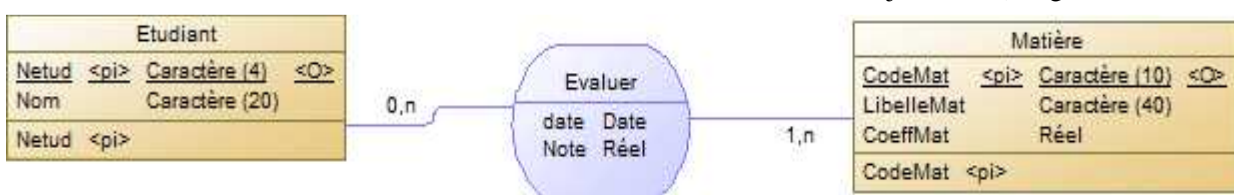
- Est ce que vous retrouvez certains éléments du Processus Unifié ?
- Quels tests effectuez-vous lorsque vous développez seul ?

Exercice 6: Soit le modèle relationnel suivant :



Utiliser un outil d'AGL, pour traduire ce modèle relationnel en modèle orienté objet UML (Diagramme de classe).

Exercice 7: Traduire le modèle relationnel ci-dessous en modèle orienté objet UML (Diagramme de classe).



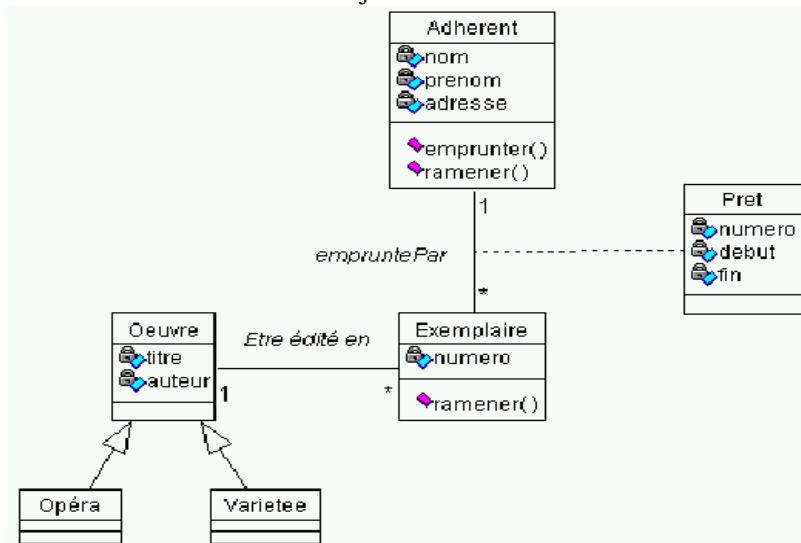
Exercice 8 : Effectuer un reverse engineering (passage du code source java au diagramme de classe UML) :

```

Class ClasseA {
    private String nom ;
    private ClasseC C1;
    ClasseA (String nom) { this.nom=nom ; this.C1= new ClasseC(nom); C1.afficher(); }
}
class ClasseB extends ClasseA { void test() ; }
class ClasseC {
    private String nom;
    ClasseC (String n) { nom = n; }
    void afficher () { System.out.println("le nom de la classe C est " + nom); }
}

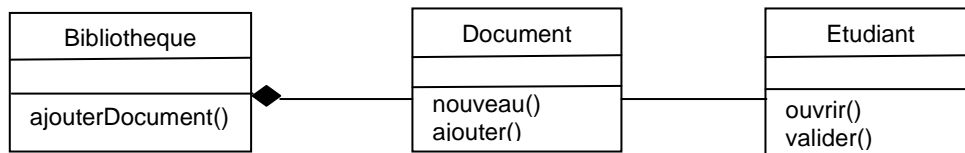
```

Exercice 9 : Soit le modèle objet UML ci-dessous



1. Générer le code source java
2. Traduire le modèle ci-contre en modèle relationnel.

Exercice 10: Le diagramme de classes ci-dessus décrit des bibliothèques composés de documents, eux-mêmes composés d'Etudiant. L'implémentation des classes en question est la suivante :



<pre> class Biblio{ private Document document ; public void ajouterDocument() { document.nouveau() ; document.ajouter() } } </pre>	<pre> class Document { private Etudiant exempl ; public void nouveau() { ... exempl.ouvrir() ; } public void ajouter() { ... exempl.valider() ; } } </pre>
<pre> class Etudiant { public void ouvrir() { ... } public void valider() { ... } } </pre>	<p>Début programme principal</p> <p>Biblio biblio ;</p> <p>biblio.ajouterDocument() ;</p> <p>Fin programme principal</p>

Question : Utiliser l'outil d'AGL, pour représenter l'échange de messages résultant de l'exécution du programme principal, par un diagramme de séquence.

Exercice 11: Phase de spécifications des besoins. Soient les cas d'utilisation suivants :

- Passer une commande; Passer une commande urgente ; Suivre une commande; Valider l'utilisateur; Passer une commande; Expédier commande totale ou partielle

Le suivi de la commande désigne le processus complet, du passage à l'expédition. Il peut toutefois arriver qu'une commande passée ne soit pas envoyée. Passer une commande urgente est un cas particulier de passer une commande. Pour passer une commande, il faut nécessairement valider l'utilisateur.

Question: Utiliser l'outil d'AGL, pour coder les uses cases ci-dessus en diagramme de cas d'utilisation sans représenter les acteurs