

EXAMEN FINAL DE GENIE LOGICIEL 2

Questions (5 Pts)

1. Qu'est-ce qu'une **ligne de produits logiciels** ? Donnez un **exemple de ligne de produits logiciels** sous forme de **diagramme FODA** (Feature Oriented Domain Analysis).
2. Citez les **nouvelles activités** du processus de développement, introduites par les lignes de produits logiciels.
3. Donnez le **principe de base** de la méthode **Rational Unified Process**.
4. Quel(s) énoncé (s) est (sont) vrai (s) à propos de la méthode **Two Tracks Unified Process (2TUP)** :
 - a) Le processus de développement de **2TUP** est basé sur un modèle en Y.
 - b) Le processus de développement de **2TUP** ne tient compte que des aspects techniques.
 - c) Aucun de ces énoncés.

Problème (15 Pts)

Une **banque** comporte plusieurs **agences** réparties sur un territoire donné. Une banque est caractérisée par le nom de son directeur général, sa capitale, 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. Les comptes rémunérés ont un taux d'intérêt et rapportent des intérêts versés annuellement.

1. Donnez la description complète de toutes les classes (remplissez tous les compartiments). Précisez 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 publiques.
2. Analysez les classes trouvées en (1) et modélisez-les en factorisant (par généralisation) 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 ? Dessinez le modèle de cette relation.
4. Donnez le diagramme de classes en n'utilisant que leur nom et ajoutez tous les détails possibles aux relations.
5. A partir du diagramme de classe élaboré en 4, déduire le **code java** correspondant (ne pas considérer les propriétés des classes).
6. Assurez l'unicité du **directeur général** de la banque en appliquant le design pattern **Singleton** (commenter le code).
7. Modélisez le fait qu'un employé peut créer un compte pour un client sans se préoccuper de son implémentation en appliquant le design pattern **Abstract Factory**.
8. Exprimez, en **OCL**, les contraintes suivantes :
 - Le solde d'un compte est toujours strictement supérieur à 0.
 - Le nom du client ne doit pas être nul.
9. On ne considère que la Banque et les agences correspondantes et on suppose que la banque communique avec ses agences au moyen du **protocole TCP/IP**. Donnez le diagramme de déploiement correspondant.

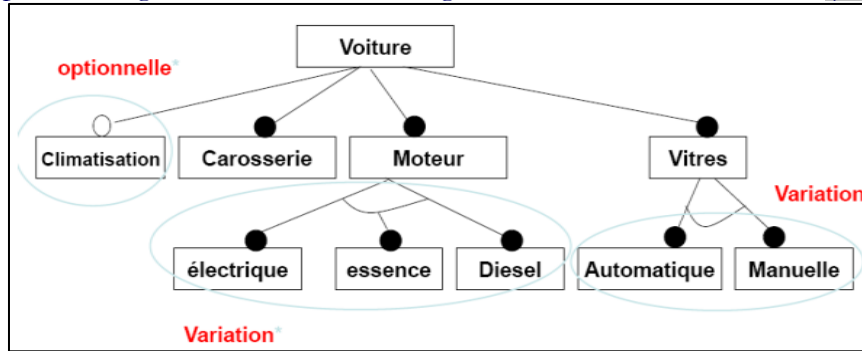
Bon courage

Corrigé type

Questions:

1) Définition d'une ligne de produits logiciels (LdP): Une LdP logiciels est un **ensemble de systèmes partageant** un ensemble de **propriétés communes** et satisfaisant des besoins spécifiques pour un **domaine particulier**. (0.5 Pt)

Exemple de ligne de produits logiciels sous forme de diagramme FODA : LdP Voiture (1 Pt)



2) Les **nouvelles activités** du processus de développement introduites par les lignes de produits logiciels: **La modélisation de la variabilité, la gestion des contraintes et la dérivation des produits**. (1.5Pts)

3) Le **principe de base** de la méthode **Rational Unified Process** : RUP est basée sur 4 phases (initialisation, élaboration, construction et transition), Chaque phase s'étale sur une ou plusieurs itérations et se termine par un Jalon. A chaque itération on procède par les 5 activités : expression des besoins, analyse, conception, codage, et test. (1.25 Pts)

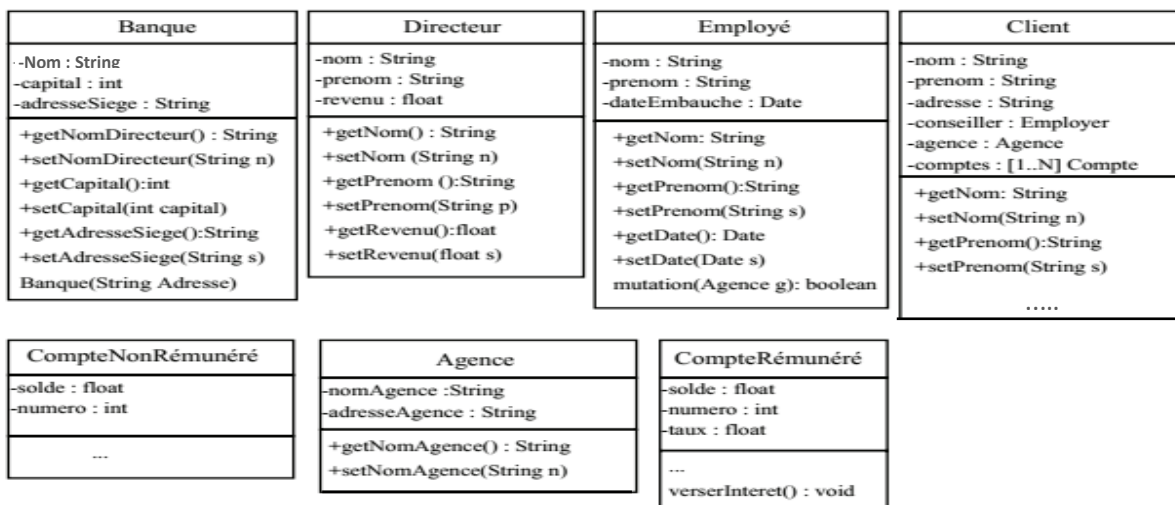
Ou bien, citer le principe de base d'UP : Utilise le langage UML, processus piloté par les cas d'utilisation, processus centré sur l'architecture, processus itératif et incrémental, et orienté risques.

4) Quel(s) énoncé (s) est (sont) vrai (s) à propos de la méthode **Two Tracks Unified Process (2TUP)** :

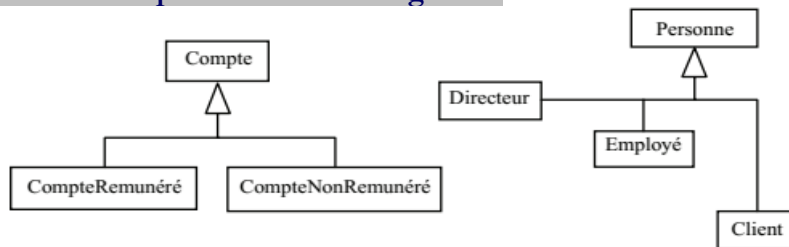
a) Le processus de développement de **2TUP** est basé sur un modèle en Y. (0.75 Pts)

Problème:

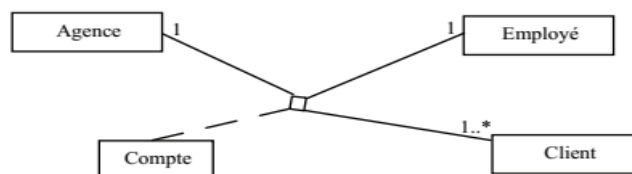
1. Les classes (2.75 pts)



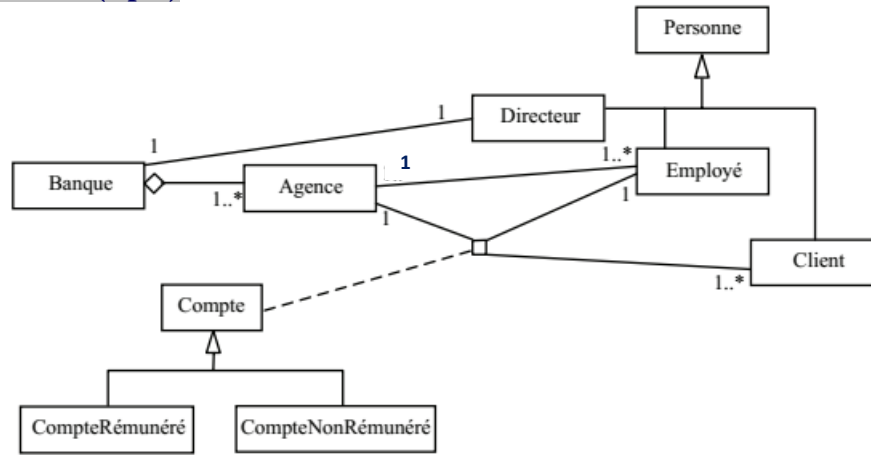
2. (0.75 pt) : Deux généralisations peuvent être distinguées :



3. (0.75 pt) : une solution qui préserve la relation entre l'agence, le client, l'employé et le compte, est de modéliser « Compte » comme étant une classe-association.



4. Diagramme de classes (3 pts)



5. Code JAVA (1.5 pts)

```

Public Class Compte {private Agence A; private Employe E; private Client C; ...}
Public Class CompteRemunire extends Compte {...}
Public Class CompteNonRemunire extends Compte {...}
Public Class Personne {...}
Public Class Client extends Personne {private Employe E; ... }
Public Class Employe extends Personne { private Agence A; ...}
Public Class Directeur extends Personne { private Banque B; ...}
Public Class Banque{ private Agence [] AG;}
Public Class Agence { private Banque B; private Client[] C; }
... reste relative aux propriétés et au diagramme de classe élaboré.

```

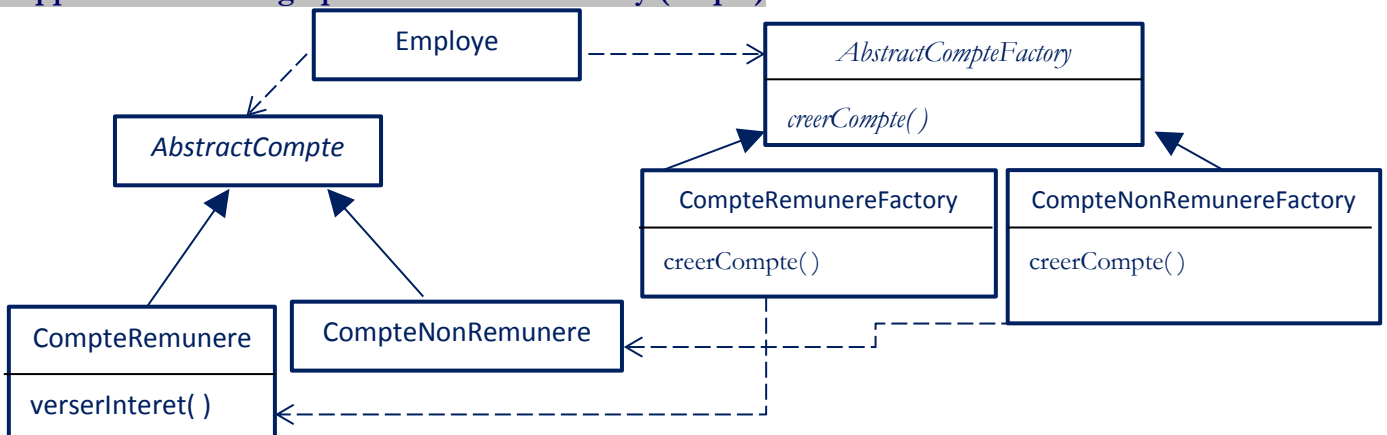
6. Application du design pattern Singleton sur la classe directeur général (1.5 pts)

```

Class DirecteurGeneral {
private static DirecteurGeneral instance = null ;
private DirecteurGeneral() {} // constructeur privé
public static DirecteurGeneral getInstance() // ne crée une instance que s'il n'en existe pas.
if (instance==null)
instance=new DirecteurGeneral ( ) ;
return instance ;} }

```

7. Application du design pattern Abstract Factory (1.5 pts)



8. Contraintes OCL : (2 Pts)

- Context Client
inv : solde > 0
- Cotext Client
inv : nom <> null

9. Diagramme de déploiement : (1.25 Pts)

