

Contrôle de Rattrapage de Génie Logiciel 2

Problème

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 le **diagramme de classes** de ce système.
2. Dédurre le **code java** correspondant (ne pas considérer les propriétés des classes).
3. Répondre à deux questions parmi les questions suivantes :
 - a) Assurez l'unicité du **directeur général** de la banque en appliquant le design pattern **Singleton** (commenter le code).
 - b) 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**.
 - c) Exprimez, en **OCL**, les contraintes suivantes :
 - a. Le solde d'un compte est toujours strictement supérieur à 0.
 - b. Le nom du client ne doit pas être nul.
 - d) 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.

Questions de Cours

1. 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.
2. Quel énoncé est vrai à propos des **lignes de produit logiciels** ?

L'ingénierie des lignes de produits logiciels passe nécessairement par deux phases complémentaires, à savoir :

 - a) L'ingénierie du domaine et la gestion de la variabilité.
 - b) La gestion de la Commonalité et l'ingénierie d'application.
 - c) L'ingénierie du domaine et l'ingénierie d'application.
 - d) Aucun de ces énoncés.
3. Répondre par vrai ou faux, justifier votre réponse ?
 - a) **FODA** (Feature Oriented Domain Analysis) est un **design pattern de création**.
 - b) **Abstract Factory** est un type de ligne de produits logiciels.

Bon courage

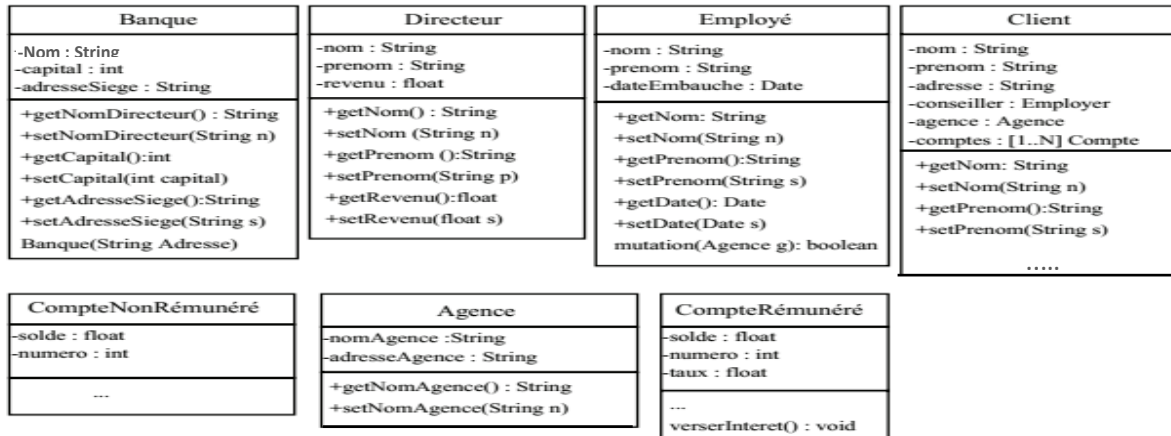
Contrôle de Rattrapage- Génie Logiciel 2

Corrige type

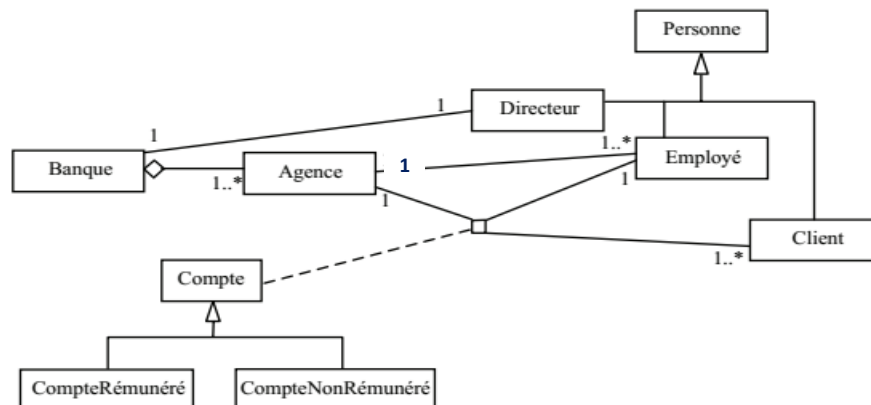
Problème

1. (4 pts)

- Les classes



- Diagramme de classes



2. Code JAVA (2 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.

3. (Choix de deux questions)

a) Application du design pattern **Singleton** sur la classe **directeur général** (2 pts)

```
Class DirecteurGeneral {
```

```
private static DirecteurGeneral instance = null ;
```

```
private DirecteurGeneral() { } // constructeur privé
```

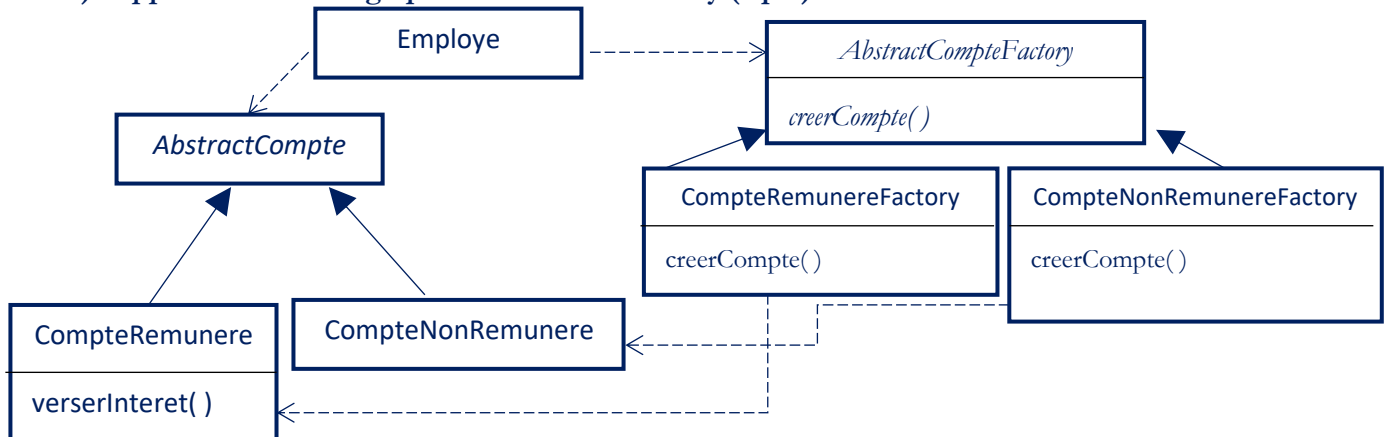
```
public static DirecteurGeneral getInstance() { // ne crée une instance que s'il n'en existe pas.
```

```
if (instance==null)
```

```
instance=new DirecteurGeneral ( ) ;
```

```
return instance ; }
```

b) Application du design pattern Abstract Factory (2 pts)



c) Contraintes OCL :(2 Pts)

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

d) Diagramme de déploiement : (2 Pts)



Questions de Cours

- 1) Le **principe de base** de la méthode **Rational Unified Process (RUP)**: 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 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.
- 2) a) (1.25Pt)
- 3) c) (1.25 Pt)
- 4)
 - a) Faux, FODA est un modèle de description de ligne de produits logiciels (1.5 Pt).
 - b) Faux, Abstract Factory est un design pattern (1.5 Pt).