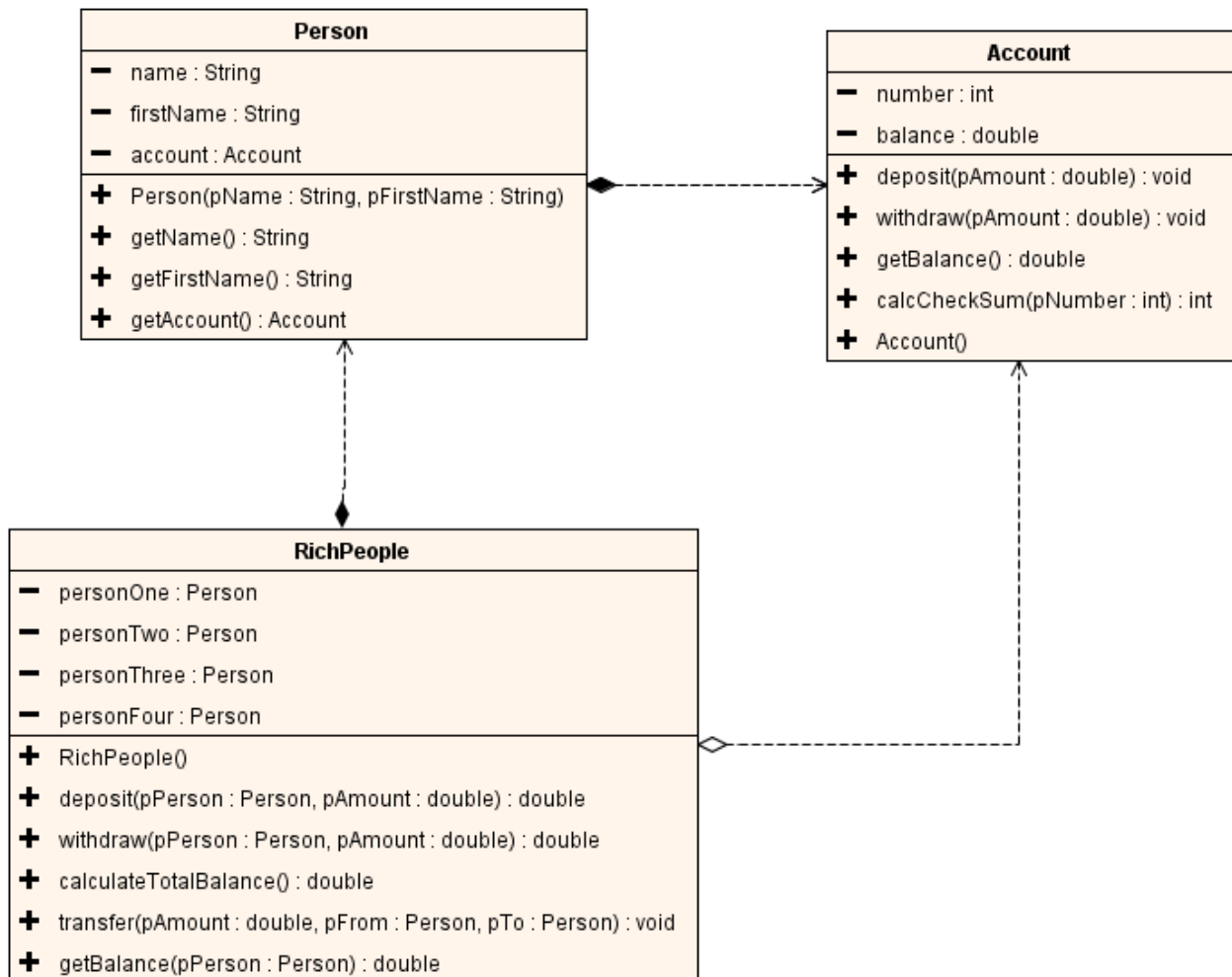


Rich People

Vous devez implémenter le projet **Rich People**. Il s'agit d'un groupe de quatre personnes riches qui possèdent chacune un compte en banque. Chaque membre de ce groupe peut transférer une certaine somme d'argent sur le compte d'un autre membre afin de pouvoir réaliser des investissements. De même, il est possible d'ajouter ou de retirer un certain montant sur un autre compte. Il est aussi possible d'extraire le solde actuel de chaque membre du groupe ou le solde total du groupe.

Il est important de respecter le diagramme UML suivant :



Classe Account

1. Attributs et un manipulateur : le compte possède un solde **balance** (nombre réel) et un numéro bancaire **number** composé de sept chiffres (nombre entier). On peut demander le solde actuel (accesseur).
2. `deposit(...)` permet d'ajouter un certain montant.
3. `withdraw(...)` permet de retirer un certain montant (sans vérification quelconque).

4. La méthode `calcChecksum(...)` retourne la somme de tous les chiffres de `pNumber`. Le nombre de chiffres est inconnue. Utilisez la boucle `while`.

Exemple : Si `pNumber` est égal à 51243, `calcChecksum(...)` retourne la valeur 15.

5. Le constructeur programme le solde à zéro. Ensuite, il crée un numéro bancaire à l'aide d'une valeur aléatoire comprise entre [50000;99999] et ajoute un nombre de vérification (utilisez `calcChecksum()`) derrière la valeur aléatoire.

Exemple : Pour la valeur aléatoire 65341, le numéro de vérification est égal à 19. En conséquence, `number` sera égal à 6534119.

Classe *Person*

1. Attributs : les informations sur la personne comportent
 - a. Le nom `name`
 - b. Le prénom `firstName`
 - c. Une référence vers une instance de la classe (objet) `Account`
2. Programmez des accesseurs pour chaque attribut.
3. Le constructeur initialise le nom et le prénom et crée un nouveau compte bancaire.

Classe *RichPeople*

1. Attributs : Les quatre attributs contiennent des références vers quatre instances de la classe `Person`.
2. Le constructeur crée quatre instances de la classe `Person` et enregistre les références vers celles-ci dans les quatre attributs correspondants. Les noms de ces quatre personnes sont :
 1. *Jeff Bezos*
 2. *Bernard Arnault*
 3. *Bill Gates*
 4. *Warren Buffet*
3. Dans cette classe, il est aussi possible de d'ajouter et de retirer de l'argent. Programmez les deux méthodes `deposit(...)` et `withdraw(...)`. Consultez le diagramme UML pour la définition exacte de celles-ci.
4. `calculateTotalBalance()` calcule le solde total des comptes des quatres personnes.
5. `getBalance()` permet de demander le solde actuel du compte d'une personne (bonus+2).
6. `transfer(...)` permet de transférer un montant d'un compte d'une personne vers le compte d'une autre personne. Veillez à ce que le compte source est débité et que le compte destinataire est crédité.