

Examen de Java

Session : ING1 - Juin 2008 - session 2 - CERGY / PAU

Note : la matière Eclipse sera évaluée sur le rendu de projet 2° semestre.

Conditions : Examen sur machines EISTI. Vous disposez d'un accès à la Javadoc 1.5 en interne, à l'URL : <http://www.eisti.fr/javadoc/>

En fin d'examen, vous devez rendre :

- L'ensemble des classes sources, dans une arborescence "src", inscrite dans un (des) paquetage(s)
- Les classes compilées (même si les écritures ne sont pas complètes, l'état de votre source DOIT pouvoir compiler), dans un répertoire "bin".
- La génération de la javadoc.
- Le tout dans une archive ZIP, ou TGZ à la rigueur (aucun autre format)
- Vous pouvez aussi poster une archive JAR

Commande de livraison : `renduprog2` sur votre compte deathstar.

Thème : les réseaux sociaux

L'actualité du Web est très proluxe en ce moment sur les thèmes des réseaux de personne. Nous utilisons ce prétexte conjoncturel pour vous faire réfléchir sur les nécessités "programmatiques" de ce type de réseaux.

Le but du devoir est de modéliser un petit réseau de personnes dans lequel on peut retrouver le "circuit" permettant de négocier entre deux personnes "par relation". Vous construirez un réseau simple, définirez une façon de qualifier les relations (guidée par le sujet) puis élaborerez un [algorithme](#) de recherche d'un chemin qui permette de passer par des "amis" et des "bonnes recommandations".

La programmation doit être claire.

La programmation doit faire usage des collections, et ne pas réimplémenter des [algorithmes](#) triviaux.

Vous devez mettre en oeuvre toutes vos connaissances Java pour proposer des solutions aux questions posées.

COMMENTEZ ET JUSTIFIEZ VOS INTENTIONS.

Questions de cours POO (4 points)

A quoi sert une classe finale ?

Pourquoi l'encapsulation très privée ralentit-elle le travail de développement ? Pouvez-vous mettre ceci en regard avec les avantages que cette pratique procure ?

Quel est le bénéfice de la programmation objet ?

Toutes les classes d'une application Java ne sont pas nécessairement toujours "montées" en mémoire vive. Expliquez pourquoi.

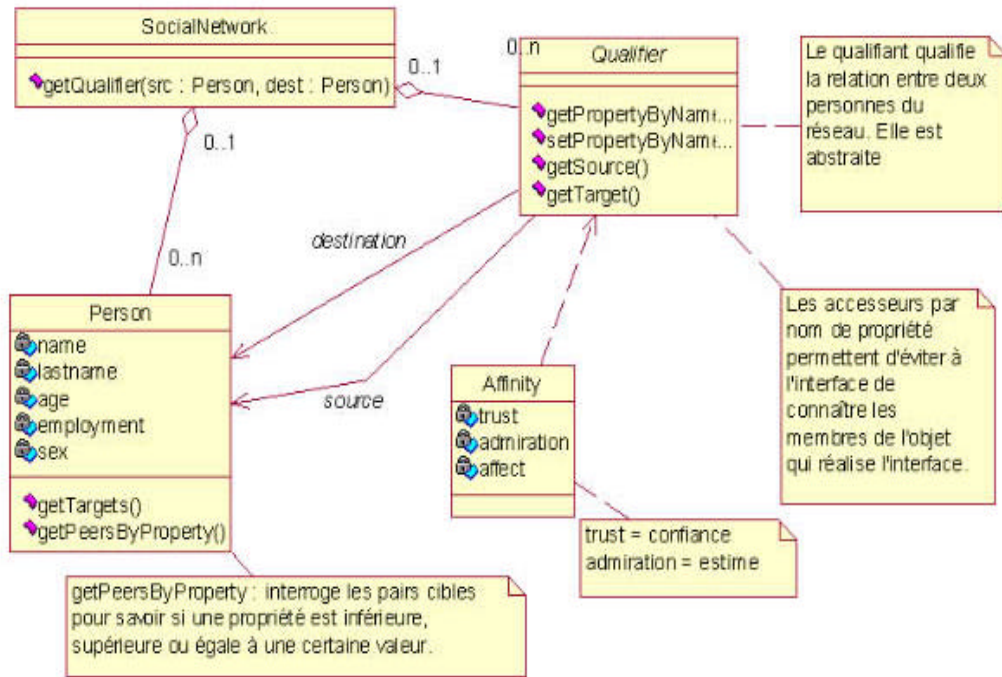
Conception de classes (6 points)

Un réseau social est composé d'individus en relation.

1. Déterminez la classe `Personne` représentant un individu du réseau. Fournissez entre trois et cinq champs descriptifs. La classe `Personne` doit être totalement mutable.
2. Déterminez une façon de constituer un réseau de personnes. Identifiez (la ou) les collections nécessaires pour que :
 - Chaque personne puisse être liée à un nombre quelconque d'autres personnes (si une personne est liée à elle-même, on pourrait y coder une "estime de soi"!!).
 - Le réseau et ses relations ne CONNAISSENT PAS à priori quels sont les qualifications de ces relations.
 - Le réseau a une existence propre en tant qu'objet. Les mécanismes de collection doivent être intégrés PAR COMPOSITION.
 - Malgré le point précédent, nous savons que ces relations seront qualifiées par quelque chose.
 - Nous savons également que les qualifiants sont des propriétés, assimilables à des

- paire nom => valeur, les deux étant exprimables en texte. Vous devrez trouver quel objet pourrait répondre à ce besoin.
- o Utilisez une interface pour stocker l'instance qui qualifie la relation.
3. Constituez les classes sous forme de carcasses SANS ECRIRE LE CODE DES METHODES. Votre objectif est de bien décider des membres et des méthodes utiles pour manipuler ce modèle et de savoir le proposer en JAVA.
 4. Le réseau doit être manipulable, c'est-à-dire que toutes les méthodes qui permette de le construire à partir d'un réseau vide doivent être présentes.
 5. On désire qualifier les relations par une mesure d'affinité. Cette classe très simple stocke quelques qualificatifs quantifiés pour mesurer l'affinité entre deux personnes (Attention, penser que ces relations n'ont pas forcément la même valeur dans les deux sens !!) :
 - o L'amitié affective (0 à 10)
 - o L'estime professionnelle (0 à 10)
 - o Le degré de confiance (0 à 10)

Pour vous aider, une transcription UML partielle a été faite de ce problème :



Vous devez évidemment trouver comment enregistrer les "couples" de Affinity pour chaque "connexion" entre des personnes du réseau.

Barème :

- Explications et analyse du problème
- carcasse de la classe Personne
- carcasse de la classe Réseau
- carcasse de l'interface Qualifiant
- carcasse d'une classe Affinité implémentant l'interface précédente

Programmation de méthodes (4 points)

Ecrivez une méthode qui à partir de deux personnes données du réseau, détermine s'il existe un chemin qui passe de la première à la deuxième par un "chemin de confiance". (On appellera chemin de confiance un chemin dans le réseau où la confiance ne descend pas au dessous de 7 inclus).

Programmation d'application (6 points)

1. Ecrivez les **constructeurs** des classes ci-dessus et
2. un programme principal qui construit un jeu de test et imprime la construction.

Barème :

- 1 point par **constructeur**
- 1 point pour la classe principale portant le main().