TP n° 5

Ce TP est évalué.
Le travail se fait seul.
Il est à rendre sur eprel à la fin du TP.
Vous devez soumettre une archive tar.gz contenant un répertoire par exercice comme dans le stub fourni.
Pensez à vérifier que votre travail est soumis correctement.
Le dernier exercice est optionnel.
Un fichier Stub est disponible sur eprel.

Exercice 3 (patron)**.** Vous devez mettre en oeuvre des classes java utilisant le patron de conception *composition* pour résoudre le problème de la fête. Plus de détails concernant ce patron sont donnés dans le cours, voir sur internet. Plus de détails concernant java sont disponibles à la fin du sujet.

Le problème de la fête est le suivant. Une organisation comporte différentes personnes qui ne sont pas toutes très marrantes. Chaque personne a un *fun factor* qui est un entier positif ou nul.

La hiérarchie de l'entreprise est un arbre. On ne peut pas s'amuser à une fête si on est en présence de son supérieur direct (n+1) ou bien si on est en présence de son subalterne direct (on est le n+1 de cette personne).

Une fête consiste donc à choisir des personnes de sorte qu'aucun participant ne soit en présence de son supérieur direct. La qualité d'une fête est mesuré par la somme du *fun factor* des participants.

Étant donné une hiérarchie de personnes, on veut calculer la qualité de la meilleure fête possible.

Indication. Il faut procéder de bas en haut et calculer pour une personne x deux meilleures fêtes possibles pour lui et ses subalternes (pas forcément directs) : celle où x n'est pas invité, celle où x peut être invité.

Détails concernant Java. Vous devez avoir une classe Travailleur et une classe Chef avec des constructeurs prenant en argument un entier positif qui est le *fun factor*. On ajoute un subalterne à un chef avec la méthode `addSubalterne`. On calcule la meilleure fête pour une organisation donnée par son chef en appelant la méthode `bestParty`.

Votre réalisation doit fonctionner avec tous les fichiers `Exemple?.java` fournis.

Exercice 4 (patron)**.** Si un sommet ne mémorise pas les résultats dans un attribut, vous parcourez inutilement l'arbre plusieurs fois. Reprenez l'exercice précédent en sauvant les résultats. Ainsi le calcul est fait lors du premier appel mais pas pour les suivants qui se contentent de retourner la valeur de ce premier calcul.

Exercice 5 (patron)**.** Vérifiez les calculs des exercices précédents en utilisant une méthode plus brutale.

Indication. Soit vous procédez de manière exhaustive en regardant tous les sous-ensembles de personnes (méthode *perebor*), soit pour éviter de faire trop de calcul vous faites un *bactrack*.