DUT Informatique – S2

TP N°3 : Programmation Orientée Objet

Exercice 1

Cette exercice a pour objectif la gestion d'un récipient d'eau.
Un récipient est caractérisé par un nom, une capacité et sa contenance.
Les opérations applicables sur un récipient sont :

- affichage de l'état du récipient (valeur des variables d'instances)
- vérifier si le récipient est vide ou plein.
- prélever *n* litres du récipient, moins s'il ne contient pas *n* litres. La valeur effectivement prélevée est retournée.
- verser *n* litres dans le récipient (l'eau en trop déborde).
- remplir/vider totalement le récipient.

Sur le même principe que les exercices précédents vous devrez écrire la classe *Recipient* et une classe *TestRecipient* ne contenant qu'une méthode *main*.
1 - définir deux constructeurs : i) un constructeur qui initialise un récipient de nom "NoName", avec une capacité de 10 litres et contenant 0 litre à sa création, ii) un constructeur qui initialise un récipient de nom *nomR*, de capacité *capa* et contenant 0 litre.
2- des méthodes d'accès en lecture/écriture pour les caractéristiques du récipient.
3- les opérations décrites ci-dessus.
4- compléter le *main* de *TestRecipient* en créant 2 récipients *r1* et *r2* .
5- remplir *r2*, – verser 5 litres dans *r1*.
6- simuler le versement litre par litre et sans débordement du contenu de *r2* dans *r1*.
7- afficher l'état des deux récipients
8- créer un troisième récipient *r3* de capacité 20; simuler des versements/prélèvements entre les trois récipients puis afficher le récipient qui a la plus grande contenance.
9- générer la documentation (javadoc) de la classe *Recipient*. Se référer aux slides 32-35 du Cours N°1

Exercice 2 :

reprendre la classe *Personne* du TP précédent.
Partie I : ajoutez à la classe *Personne*
i) les attributs (variables d'instance) : *sexe*, *estMarie* et *conjoint* . Attention : on ne peut-être marié qu'à une seule personne ! (modifiez les constructeurs & *toString( )* pour tenir compte des nouveaux attributs).

ii) les services suivants :
1- Marier une personne à une autre. On vérifiera que les deux mariés ont plus de 18 ans, sinon le mariage n'aura pas lieu et on retournera *false*.
2 - Vérifier si deux personnes p1 et p2 sont mariés ensemble.
3 - Annuler le précédent mariage s'il y avait lieu (on vérifie si la personne n'était pas mariée).

iii) écrivez un autre constructeur qui créera une personne et la mariera en même temps à la personne donnée en argument/paramètre. Écrivez une méthode *Personne enfanter(String prenom, Char sexe)* qui créera une nouvelle personne de même nom que la mère, de sexe masculin ou féminin selon la lettre ('M' ou 'F') donnée en argument (attention: seule une femme peut enfanter!).

Partie II : définir une classe *Compte* en banque , en y intégrant deux méthodes, l'une déposant de l'argent, l'autre en retirant, et dont vous vous assurerez que l'attribut solde ne puisse jamais être négatif. Dans la même classe que celle de la question précédente, écrivez une méthode d'accès au solde, qui retourne ce dernier comme un entier, alors qu'il est stocké comme un réel.
On suppose maintenant que toute personne a un attribut compte de type *Compte*.
Simuler des mouvements sur le compte bancaire d'une personne. Exemple :
-- une personne qui crédite/débite son compte d'une somme donnée.
-- une personne qui donne de l'argent à une autre personne.
-- un compte commun pour deux personnes mariées avec des retraits/dépôts.
-- etc.