

Exercices POO – Classes – niveau 1

Exercice 1 : classe Etudiant

Écrire classe JAVA appelée « Etudiant » avec les membres privés suivant :

- nom : (de type char),
- note1, note2 : (de type float)
- calc_moy() : calcule la note moyenne.
- afficher () : affiche le nom et la note moyenne.

Le programme principal (main) demande à l'utilisateur d'entrer le nom et les notes d'un étudiant et affiche le nom et la note moyenne.

Exercice 2 : classe Compte

Écrire un programme en JAVA qui simule la gestion d'un simple compte bancaire. Le compte est créé avec un solde initial. Il est possible de déposer et de retirer des fonds, d'ajouter des intérêts et de connaître le solde actuel. Cela devrait être implémenté dans une classe nommée *Account* qui comprend :

- 1) Un constructeur par défaut qui met le solde initial à zéro.
- 2) Un constructeur qui accepte une balance initial comme paramètre.
- 3) Une fonction *getBalance* qui renvoie le solde actuel.
- 4) Une méthode *deposer* pour déposer un montant spécifié.
- 5) Une méthode *retirer* pour retirer un montant spécifié.
- 6) Une méthode *ajouter_Interet* pour ajouter de l'intérêt au compte.

La méthode *ajouter_Interet* prend le taux d'intérêt comme paramètre et modifie le solde du compte en solde * (1 + taux d'intérêt).

Exercice 3 : Classe rectangle

Écrire en JAVA un programme utilisant une classe **Rectangle** dont le constructeur prend deux paramètres, largeur et hauteur et qui offre les fonctions suivantes :

- 1) calcul du périmètre
- 2) calcul de la surface
- 3) affichage

ainsi que les accesseurs et mutateurs triviaux (lecture et modification de la largeur et de la hauteur).

Exercice 4 : classe Point

Réaliser en JAVA une classe Point permettant de manipuler un Point d'un plan.

On prévoira :

- 1) un Point est défini par ses coordonnées x et y (des membres privés)
- 2) les constructeurs

- 3) une fonction membre *deplace* effectuant une translation définie par ses deux arguments dx et dy (double)
- 4)une fonction membre *affiche* se contentant d'afficher les coordonnées cartésiennes du Point.
- 5)une fonction membre *saisir* se contentant de saisir les coordonnées cartésiennes du Point.
- 6)une fonction membre *distance* effectuant calculant la distance entre deux Points.
- 7)une fonction membre *milieu* donnant le milieu d'un segment.
- 8)un petit programme d'essai (*main*) gérant la classe Point.