

TD1 : Classes, instances, encapsulation

Exercice 1.-

La méthode main prend comme paramètre un tableau de chaînes de caractères. Ceci permet d'exécuter le programme en tapant une entrée en ligne de commande.

- a) Ecrire un programme `exo1-a.java` composé d'une seule classe qui permet d'afficher deux chaînes : la première est donnée en ligne de commande, la seconde est un paramètre de la méthode `println`.
- b) La méthode `Integer.parseInt (chaîne)` convertit le paramètre chaîne en un entier. Ecrire un programme `exo1-b.java` qui lit en ligne de commande deux nombres et les affiche puis affiche leur somme.
- c) Ecrire un programme `exo1-c.java` qui affiche la somme de n entiers lus en ligne de commande.

Exercice 2.- Ecrire les programmes suivants :

- a) calculer et afficher la somme de 2 nombres entiers donnés.
- b) calculer et afficher la somme de n nombres entiers quelconques donnés.
- c) Soit T un tableau de n entiers, ($n \geq 100$), calculer et afficher la somme des éléments de T .

Exercice 3.- Un tableau d'entiers est caractérisé par sa taille effective et le tableau de ses composants (au plus 100 éléments). La classe possède les méthodes suivantes :

- a) initialisation ;
- b) affichage des éléments ;
- c) insertion d'une valeur donnée ;
- d) suppression d'une valeur donnée ;
- e) vérification si le tableau est trié en ordre croissant ou non ;
- f) tri des éléments dans l'ordre croissant ;
- g) recherche du maximum des composants ;
- h) recherche du minimum des composants ;
- i) recherche séquentielle d'une valeur donnée en paramètre ;
- j) recherche dichotomique d'une valeur donnée en paramètre ;

Définir la classe `Tableau` qui répond à la description donnée et écrire une classe `ProgPrincipale` qui permet de tester ces méthodes en affichant un menu où l'on peut choisir une des opérations citées ci-dessus appliquées à un tableau d'entiers donné.

Exercice 4.-

Ecrire un programme java qui décrit une classe COMPTE caractérisée par les attributs suivants :

NuméroCpte : Entier

NomTitulaire : Chaîne

Solde : réel

Définir les méthodes de cette classe sachant que l'on veut pouvoir débiter, créditer le compte et afficher le solde de chaque compte créé.

Exercice 5.- Un article en stock est un objet caractérisé par les attributs suivants :

- sa désignation : une chaîne de caractères ;

- son prix unitaire : un nombre réel ;

- sa quantité : un entier ;

Il est traité par les méthodes suivantes :

- initArticle pour initialiser un article avec des paramètres en entrée ;

- afficherArticle permet d'afficher les caractéristiques d'un article ;

- prixTTC permet de calculer le prix TTC d'un article donné ;

- reduireStock permet de diminuer le stock d'articles d'un nombre donné ;

- augmenterStock permet d'augmenter la quantité en stock d'une valeur donnée ;

Ecrire une classe Article et une autre classe PrincipaleArticle pour la tester.

Exercice 6.- Un point est caractérisé par les attributs privés x et y. Les méthodes qui peuvent traiter un point sont :

- Point (x,y) : initialise un point par des valeurs données en paramètres ;

- Point () : initialise un point à l'origine (0,0) ;

- afficherPoint () : affiche un point sous la forme : (x, y) ;

- deplacerPoint(dx, dy) déplace le point de sa position (x, y) à la position (x+dx, y+dy)

a) Définir une classe Point qui représente l'objet point en respectant la description ci-dessus ;

b) Définir une classe ProgPrincipalPoint qui permet de tester les méthodes précédentes.

c) Définir une nouvelle classe TableauPoints, caractérisée par un attribut qui définit un tableau de points et possède une méthode distance qui retourne la distance entre deux points.

- d) Modifier la classe ProgPrincipalPoint pour calculer la distance minimale entre les points d'un tableau de n points, n entier lu en ligne de commande.

Exercice 7.- Les nombres complexes

Un objet complexe est caractérisé par deux nombres réels : sa partie réelle (pReel) et sa partie imaginaire (pImag). Les calculs sur les nombres complexes sont les suivants :

- module : retourne $\sqrt{pReel^2 + pImag^2}$;
- addition et soustraction : retournent respectivement la somme et la différence de deux nombres complexes ; elles sont static ;
- afficherComplexe permet d'afficher un nombre complexe ;

Ecrire une autre classe qui contient la méthode main et permet de tester les traitements prévus.