

Série d'exercices n° 2

Classes et Objets

Exercice 1

1. Donner l'implémentation d'une classe **Vecteur** d'entiers définie par deux attributs, une référence et une taille. La classe doit comporter un constructeur, une méthode *remplir*, une méthode *afficher* et une méthode *somme* qui calcule la somme de deux vecteurs.
2. Ecrire un programme qui crée deux vecteurs, calcule leur somme et affiche le résultat.

Exercice 2

On considère une entité **Pile** statique (sous forme d'un vecteur) d'entiers.

1. Implémenter une classe qui gère l'entité Pile comportant toutes les primitives de manipulation de la structure Pile.
2. Ecrire un programme qui crée une pile de n éléments et supprime toutes les répétitions d'une valeur val donnée.

Exercice 3

On considère une entité Ouvrage décrite par une référence, un titre, un nom d'auteur et une année d'édition.

1. Donner l'implémentation de la classe Ouvrage, contenant un constructeur avec paramètres, une méthode de saisie des attributs, une méthode pour afficher la description complète d'un ouvrage et une méthode *getAnnée* pour retourner l'année d'édition de l'ouvrage.
2. Ecrire un programme java permettant de stocker la description de n ouvrages (entrés au clavier) dans un vecteur T de type Ouvrage.
3. Ecrire un programme java qui affiche la description de tous les ouvrages du vecteur T édités entre deux années données A1 et A2.

Exercice 4

On considère l'entité **Intervalle** (fermé) dans l'ensemble des nombres réels. Un intervalle est défini par deux valeurs **BI** et **BS** correspondant respectivement, à la borne inférieure et la borne supérieure de l'intervalle.

1. Donner l'implémentation en java, de la classe Intervalle comportant les méthodes :

afficher : affiche l'objet intervalle [BI, BS]

milieu : calcule le point milieu de l'intervalle

avant : vérifie si un intervalle est avant un autre sur l'ensemble des réels

disjoint : vérifie si deux intervalles sont disjoints (aucune intersection entre eux)

contenu : vérifie si un intervalle est complètement inclus dans un autre

intersection : calcule l'intervalle intersection de deux autres intervalles

fusion : calcule l'intervalle fusion de deux autres intervalles, si possible.

2. Ecrire un programme permettant de :

- Créer un vecteur de n intervalles (n donné) de sorte que les intervalles soient *numérotés de manière séquentielle* (modifier la classe intervalle à cet effet).

- Détermine l'intersection de tous les intervalles et affiche le résultat.