

## Travaux dirigés 3 Programmation orientée objet (Java) SMA5

### Exercice 1

1. Créez une classe *Ensemble* pour représenter des sous-ensembles de l'ensemble des entiers compris entre 0 et N - 1, où N est une constante donnée, avec les méthodes suivantes :

- int **cardinal**() : nombre d'éléments de l'ensemble,
- boolean **contient**(int i) : test d'appartenance,
- void **ajout**(int i) : ajout d'un élément à l'ensemble,
- String **toString**() : transformation de l'ensemble en une chaîne de caractères de la forme {5; 6; 10; 15; 20; 21; 22; 30}

Les éléments d'un ensemble seront mémorisés dans une variable d'instance de type tableau d'entiers.

Pour essayer la classe *Ensemble*, écrivez un programme déterminant les nombres différents contenus dans une suite de nombres lue sur l'entrée standard.

2. Modifiez la classe précédente afin de permettre la représentation d'ensembles d'entiers quelconques, c'est-à-dire non nécessairement compris entre 0 et une constante N connue à l'avance.

### Exercice 2

Un étudiant est défini par 7 informations : un nom, 4 notes, la moyenne de ses notes et un résultat qui vaut "non admis" si (moyenne < 10) sinon "admis". Définir la classe *Etudiant* en la dotant des méthodes nécessaires pour construire les objets et leur donnant la possibilité de calculer leurs moyennes et leurs résultats.

### Exercice 3

Définissez une classe *Equation* pour créer des objets représentant une équation du second degré et donnez à chaque objet la possibilité de calculer les racines de l'équation correspondante

$$ax^2 + bx + c = 0$$

$$\text{discriminant} = b^2 - 4ac$$

$$\text{racines} = (-b \pm \sqrt{\text{discriminant}}) / 2a$$

### Exercice 4

1. Définir la classe *Complexe* pour représenter les nombres complexes. Donner les méthodes correspondantes aux opérations arithmétiques sur 2 variables complexes
2. Définir une classe *TestComplexe* qui déclare des variables complexes et demande au choix de réaliser une opération arithmétique

## Exercice 5

Définir une classe **TD** (String nom-td, int nb-etud, int maxeffectif) maxeffectif est une variable de classe pour indiquer l'effectif maximal par TD et une classe **Admin** (String nom). Cette classe est chargée de gérer les effectifs des objets de la classe **TD**. Elle doit être en mesure d'effectuer les traitements suivants :

- ajouter x personnes à l'effectif d'un groupe de TD
- transférer x personnes d'un groupe vers un autre groupe

## Exercice 6

Définir une classe **Mairie** (String nom-mairie), une classe **Personne** (String nom, adresse ; Personne epouxde ; Acte act) une classe **Acte** (int no-enreg ; String lieu-mariage ; String date) pour simuler le déroulement suivant :

- Création de la mairie m1 et deux personnes p1, p2 qui sont célibataires.
- La mairie m1 unit les personnes p1, p2 de la façon suivante :
- Elle procède à l'union de deux personnes
- Puis elle crée un objet de la classe Acte pour établir un acte de mariage
- Enfin, elle remet cet acte à p1 et p2

## Exercice 7

Définir les classes **Guichet** (String station ; Ticket tt), **Ticket** (String destination ; double prix), **Personne** (String nom ; Ticket tick) pour simuler l'achat d'un ticket par une personne dans un guichet :

- Création d'un objet g de la classe Guichet : ce guichet n'a pas de ticket fabriqué
- Création d'un objet p de la classe Personne : cette personne n'a pas de ticket
- La personne p demande un ticket au guichet g ; celui-ci fabrique un ticket qui reste en possession du guichet
- Le guichet g donne le ticket à la personne p ; p a son ticket et le guichet ne l'a plus.