INF1 - TD 9 — Structures Simples

UNIVERSITÉ DE RENNES 1

Le but de ce TD est de créer et manipuler des structures (Cours 8). Les structures permettent de définir de nouveaux types, qu'on appelle *composés* : ils permettent de rassembler ou agréger des données de différents types, manipulées comme un tout.

On considère la gestion de la scolarité d'un lycée : les élèves (ex. Marie Dupont, Mohamed Ali) sont répartis en des classes (ex. la Terminale A, la Seconde 6), chaque élève obtient des notes dans différentes matières, et on doit calculer des statistiques sur leurs notes (moyennes, etc).

Exercice 1 : Structure `NotesMatiere` : notes obtenues pour une matière

On veut représenter, pour une matière donnée (Histoire, Informatique, Anglais...), toutes les notes obtenues par un élève. Exemple : pour la matière Informatique, une élève peut obtenir les notes 10.0, 15.0 et 15.5.

1. Définir la structure `NotesMatiere`, composée d'un champ `nom` de type `String`, et d'un champ `notes` de type tableau de `double`.
2. Définir deux exemples de notes, obtenues pour deux matières différentes. On définira pour cela deux variables de type `NotesMatiere`.
3. Écrire une fonction `afficheNotesMatiere`, qui affiche une structure `NotesMatiere` de manière lisible : le nom de la matière, puis les notes obtenues. Par exemple : `Informatique : 10.0, 15.0, 14.5`.
4. Indiquer comment afficher les deux exemples que vous avez définis précédemment, ainsi que l'affichage produit.
5. Écrire une fonction `moyenneNotes` qui calcule la moyenne des notes obtenue pour une matière. Pour l'exemple précédent, la fonction doit calculer la moyenne de 10.0, 15.0 et 14.5, c'est-à-dire 13.1666...
6. Écrire une fonction `afficheNotesEtMoyenne`, qui affiche une `NotesMatiere` de manière lisible, avec la moyenne des notes obtenues.

Exercice 2 : Structure `Eleve` : identité et notes d'un-e élève

On s'intéresse maintenant à représenter les informations utilisées pour afficher le bulletin de notes d'un ou une élève. Par exemple, une élève appelée Marie Dupont suit le cours d'informatique, de mathématiques, et d'arts plastiques, et a obtenu plusieurs notes dans chacune de ces matières.

1. On modélise un élève par son nom et les notes qu'il a obtenues dans différentes matières. Définir la structure `Eleve` correspondante. Elle sera composée d'un champ `nom`, d'un champ `prenom` chacun de type `String`, d'un champ `age` de type entier, et d'un tableau de `NotesMatiere`.
2. Définir deux variables de type `Eleve`, modélisant deux élèves : un élève étant inscrit dans 3 matières, et un élève étant inscrit dans 1 seule matière.
3. Écrire une fonction `afficheEleve`, qui prend en entrée un `Eleve`, et qui l'affiche de manière lisible. Par exemple :

```
Amélie Poirier (16 ans)
       Informatique : 11.0, 15.0, 17.5
       Mathématiques : 14.0, 13.0, 16.5, 15
```

4. Pour simplifier la création d'élèves, on décide de définir une fonction de construction dédiée. Écrire une fonction `construireEleve`, qui prend en entrée un nom, un prénom, et un âge, et renvoie l'`Eleve` nouveau correspondant. Les notes pour les matières peut-être géré ultérieurement : on pourra le définir comme vide initialement.

Exercice 3 : Structure `MatiereStat` : statistiques pour une matière

On considère maintenant la représentation des données de statistiques sur une matière donnée, pour l'ensemble des élèves d'une classe, sur l'année. On souhaite gérer : la moyenne la plus haute obtenue par un élève, la moyenne la plus basse obtenue par un élève, et la moyenne générale sur toute la classe. Par exemple : en Histoire, dans une classe de seconde, la moyenne la plus haute obtenue est 13.4, la moyenne la plus basse est 8.92, et la moyenne générale est de 11.5.

1. Définir la structure `MatiereStat` stockant ces trois informations statistiques pour une matière donnée. On stockera ces informations avec un champ pour le nom de la matière, de type `String`, et 3 champs de types `double`.

   Attention : on ne demande pas de calculer ces statistiques, il faut juste définir la structure.
2. Définir deux exemples de `MatiereStat`, pour deux matières distinctes.
3. Écrire une fonction `afficheMatiereStat`, qui prend en entrée une structure `MatiereStat` et qui affiche les informations contenues dans cette structure. Par exemple :

   ```
   Pour la matière Histoire
        Moyenne min. = 8.92
        Moyenne max. = 13.4
        Moyenne générale = 11.5
   ```

   Indiquez quel affichage est réalisé sur vos deux exemples définis à la question précédente.

Exercice 4 : Structure `Classe` : les classes du lycée

On souhaite maintenant modéliser les classes du lycée. Une classe de lycée possède un nom (Seconde 3, Terminale A, etc), un ensemble d'élèves, ainsi que les statistiques des matières enseignées à cette classe.

1. Définir la structure `Classe`. Définir également une variable de type `Classe`, comprenant 3 élèves, et dont les enseignements sont l'Histoire, l'Informatique, et les Arts Plastiques. On pourra prendre un exemple simple, où chaque matière ne donne lieu qu'à une seule note durant toute l'année, et où la classe ne comprend que 2 élèves.
2. Écrire une fonction `afficheMoyennes`, qui prend en entrée une `Classe` et qui, pour chaque matière, affiche la moyenne de chaque élève de la classe. On suppose que chaque élève a des notes dans les mêmes matières.
3. Écrire une fonction `calculeStatistiques`, qui prend en entrée une `Classe` et qui met à jour les statistiques pour toutes les matières enseignées à la classe. On supposera que tous les étudiants d'une même classe suivent les mêmes matières.
4. Écrire une fonction `meilleureMoyenne`, qui prend en entrée une `Classe` et le nom d'une matière, et renvoie l' `Eleve` de la classe qui a la meilleure moyenne dans cette matière.