

## TD 5 : structures (éléments de correction)

### Exercice 3 : Étudiants

Soit structure **Etudiant**

```
{  
    champ nom : chaîne de caractères  
    champ prenom : chaîne de caractères  
    champ groupe : entier  
    champ notes : tableau de réels  
    champ nbNotes : entier  
    champ moyenne : réel  
}
```

#### Algorithme 1 : CreerEtudiant

**Principe** : Initialiser les champs renseignés ; déclarer et initialiser le tableau des notes avec des 0.

**Entrée** :  
    xNom : entier  
    xPrenom : entier  
    xGroupe : entier  
    xNombreNotes : entier

**Local** :  
    i : entier

**Sortie** :  
    e : **Etudiant**

**Début**

```
    Soit e : structure Etudiant  
    e.nom = xNom  
    e.prenom = xPrenom  
    e.groupe = xGroupe  
    Soit e.notes = tableau de xNombreNotes réels  
    Pour i de 0 à xNombreNotes - 1 avec un pas de +1 Faire  
    |        e.notes[i] = 0.0  
    Fin Pour  
    e.nbNotes = 0  
    e.moyenne = 0.0
```

**Fin**

#### Algorithme 2 : AfficherEtudiant

**Principe** : Afficher les champs simples et parcourir le tableau des notes pour afficher les notes

**Préconditions** : xEtudiant ≠ null

**Entrée** :  
    xEtudiant : structure **Etudiant**  
    xNombreNotes : entier

**Local** :  
    i : entier

**Sortie** :  
    e : structure **Etudiant**

**Début**

```
    Afficher("Etudiant : {xEtudiant.prenom} {xEtudiant.nom} groupe {xEtudiant.groupe}")  
    sauter une ligne  
    Pour i de 0 à xNombreNotes - 1 avec un pas de +1 Faire  
    |        Afficher("{xEtudiant.notes[i]} ")  
    Fin Pour  
    Afficher("moyenne = {xEtudiant.moyenne} sur {xEtudiant.nbNotes}")
```

**Fin**

### Algorithme 3 : InsérerProchaineNoteEtudiant

**Principe :** Insérer la note à la première position libre (champs nbNotes) si nbNotes < xNombreNotes et parcourir le tableau pour mettre à jour la moyenne sur les nbNotes+1 valeurs déjà renseignées.

**Préconditions :** xEtudiant ≠ null

**Entrée :** xEtudiant : structure **Etudiant**

xNombreNotes : entier

xEntier : entier

**Local :** i : entier

**Sortie :** void

**Début**

```
Si xEtudiant.nbNotes < xNombreNotes Alors
    xEtudiant.notes[xEtudiant.nbNotes] = xEntier
    xEtudiant.nbNotes = xEtudiant.nbNotes + 1
    xEtudiant.moyenne = 0.0
    Pour i de 0 à xEtudiant.nbNotes avec un pas de +1 Faire
        xEtudiant.moyenne = xEtudiant.moyenne + xEtudiant.notes[i]
    Fin Pour
    xEtudiant.moyenne = xEtudiant.moyenne / xEtudiant.nbNotes
Sinon
    Afficher("insertion impossible !")
Fin Si
```

**Fin**

### Algorithme 4 : AfficherEtudiantsPromo

**Principe :** parcourir le tableau d'étudiants et les afficher si ≠ null

**Entrée :** xPromo: tableau de xTaillePromo structure **Etudiant**

xTaillePromo : entier

xNombreNotes : entier

**Local :** i : entier

j : entier

**Sortie :** void

**Début**

```
Pour i de 0 à xTaillePromo - 1 avec un pas de +1 Faire
    Si xPromo[i] ≠ null Alors
        Afficher("Etudiant {xPromo[i].prenom} {xPromo[i].nom} gr. {xPromo[i].groupe}")
        sauter une ligne
        Pour j de 0 à xNombreNotes - 1 avec un pas de +1 Faire
            Afficher("{xPromo[i].notes[i]} ")
        Fin Pour
        Afficher("moyenne = {xPromo[i].moyenne} sur {xPromo[i].nbNotes}")
    Fin Si
Fin Pour
```

**Fin**

### Algorithme 5 : CalculerMoyenneGeneralePromo1

**Préconditions :** toutes les notes de tous les étudiants ont été renseignées (et donc la moyenne de chacun est à jour), les étudiants sont ≠ null

**Principe :** Parcourir le tableau des étudiants et sommer les moyennes puis diviser par le nombre d'étudiants

**Entrée :** xPromo: tableau de xTaillePromo structure **Etudiant**

xTaillePromo : entier

**Local :** i: entier

**Sortie :** moyennePromo : réel

**Début**

```
Soit moyennePromo = 0.0
Pour i de 0 à xTaillePromo - 1 avec un pas de +1 Faire
    moyennePromo = moyennePromo + xEtudiants[i].moyenne
Fin Pour
moyennePromo = moyennePromo / xTaillePromo
```

**Fin**

### Algorithme 5 : CalculerMoyenneGeneralePromo2

**Préconditions :** toutes les notes de tous les étudiants ont été renseignées (et donc la moyenne de chacun est à jour)

**Principe :** Parcourir le tableau des étudiants et sommer les moyennes puis diviser par le nombre d'étudiants

**Entrée :** xPromo: **tableau de** xTaillePromo structure **Etudiant**

xTaillePromo : **entier**

**Local :** i: **entier**

nbEtudiants = **entier**

**Sortie :** moyennePromo : **réel**

**Début**

**Soit** moyennePromo = 0.0

**Soit** nbEtudiants = 0

**Pour** i **de** 0 **à** xTaillePromo - 1 **avec un pas de** +1 **Faire**

**Si** xEtudiants[i] **≠ null** **Alors**

        moyennePromo = moyennePromo + xEtudiants[i].moyenne

        nbEtudiants = nbEtudiants + 1

**Fin Si**

**Fin Pour**

moyennePromo = moyennePromo / nbEtudiants

**Fin**

### Algorithme 6 : CalculerMoyenneGeneraleGroupe

**Préconditions :** toutes les notes de tous les étudiants ont été renseignées (et donc la moyenne de chacun est à jour)

**Principe :** Parcourir le tableau des étudiants et sommer les moyennes si l'étudiant appartient au groupe donné puis diviser par le nombre d'étudiants du groupe (gardé à jour lors du parcours).

**Entrée :** xPromo: **tableau de** xTaillePromo structure **Etudiant**

xTaillePromo : **entier**

xGroupe : **entier**

**Local :** indiceEtudiant : **entier**

nbEtudiantsGroupe = **entier**

**Sortie :** moyenneGroupe : **réel**

**Début**

**Soit** moyenneGroupe = 0.0

**Soit** nbEtudiantsGroupe = 0

**Pour** i **de** 0 **à** xTaillePromo - 1 **avec un pas de** +1 **Faire**

**Si** xEtudiants[i] **≠ null** && xEtudiants[i].groupe == xGroupe **Alors**

        moyenneGroupe = moyenneGroupe + xEtudiants[i].moyenne

        nbEtudiantsGroupe = nbEtudiantsGroupe + 1

**Fin Si**

**Fin Pour**

**Si** nbEtudiantsGroupe > 0 **Alors**

    moyenneGroupe = moyenneGroupe / nbEtudiantsGroupe

    Afficher("Moyenne du groupe {xGroupe} = {moyenneGroupe}")

**Sinon**

    Afficher("aucun étudiant dans le groupe {xGroupe} !")

**Fin Si**

**Fin**