

# Tableau à une dimension

En JAVA, pour déclarer un tableau à une dimension, la syntaxe est

```
<type du tableau> <nom du tableau> [] = { <contenu du tableau>};
```

Pour exemple :

```
int tabEntier[] = {0,1,2,3,4,5,6,7,8,9};  
double tablDouble[] = {0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0};  
char tabCaractere[] = {'a','b','c','d','e','f','g'};  
String tabChaine[] = {"chaine1", "chaine2", "chaine3", "chaine4"};
```

On peut aussi déclarer un tableau vide, mais il devra impérativement contenir un nombre de cases bien défini. Par exemple :

```
int tabEntier[] = new int[6]; OU int[] tabEntier = new int[6];
```

**Exercice :** Écrire un algorithme permettant de saisir 30 notes et de les afficher après avoir multiplié toutes ces notes par un coefficient fourni par l'utilisateur.

## Rappel Exercice

```
Algorithme tableau_note  
//déclaration des variables  
Variable Note : tableau[1..30] de Réels  
Coef, i : entier  
Début  
    Afficher("Entrer le coefficient")  
    Saisir(Coef)  
    //Remplissage du tableau Note  
    Pour i Allant de 1 à 30 Faire  
        Afficher("Entrer la valeur de la note")  
        Saisie(Note[i])  
    FinPour  
    //Affichage des notes * Coef  
    Pour i Allant de 1 à 30 Faire  
        Affiche(Note[i] * Coef)  
    FinPour  
Fin
```

# Tableau multidimensionnel

En JAVA, pour déclarer un tableau à une dimension, la syntaxe est

```
<type tab> <nom tab> [][]... = { {val00,val01,val02,...}, {val10,val11,val12,...}, ...};
```

Exemple :

```
int premiersNb[][] = { {0,2,4,6,8},{1,3,5,7,9} };
```

Pour utiliser et rechercher dans un tableau, on a besoin :

- De la fonction <nom tab>.length qui retourne la taille d'un tableau,
- Et de la boucle for(int i = 0; i < <nom tab>.length ; i++)

Exemple :

```
for(int i = 0; i < tabCaractere.length; i++)
{ System.out.println("À l'emplacement " + i +" du tableau nous avons = " + tabCaractere[i]);}
```

Exercice :

Écrire un algorithme permettant la saisie des notes d'une classe de 30 étudiants en 5 matières.

Le Cnam Rhône-Alpes - 4 rue Dauier - 69007 Lyon - 04 78 58 30 99

## Rappel Exercice

Algorithme Notes

//déclaration des variables

Constante N = 30

M = 5

Variable note : tableau[1..N, 1..M] de Réels

i, j : entier

Début

//Remplissage du tableau note

Pour j Allant de 1 à N Faire

Pour i Allant de 1 à M Faire

Afficher("Entrer la note de l'étudiant", i, " dans la matière ", j)

Saisir(note[i, j])

FinPour

FinPour

Fin

# Exercices

## Exercice 4 : Nombre de moyennes >= 10

En utilisant les tableaux, écrire un algorithme qui permet la saisie d'une liste de n moyennes réelles et d'afficher le nombre des moyennes supérieures ou égales à 10. On suppose que n <= 100.

## Exercice 5 : Tableau à deux dimensions

Ecrire un algorithme qui permet :

La saisie des notes d'une classe de 5 étudiants en 4 matières

Calcul et affiche la moyenne de chaque étudiant

Calcul et affiche la moyenne de la classe dans chaque matière

Calcul et affiche la moyenne générale de la classe..

## Rappel Exercice 4

```
Algorithme Moy_sup_a_10
//M : indice supérieur du tableau
Constante M = 100
//Moy : tableau de M éléments réels
Variables Moy : tableau[1..M] des Réels
    i, n, k : Entiers
Début
    Afficher("Donner la valeur de n : ")
    Saisir(n)
    //Saisie des éléments du tableau
    Pour i Allant De 1 à n Faire
        Afficher("Donner l'élément ", i, " du tableau")
        Saisir(Moy[i])
    FinPour
    //k : compte le nombre de moyennes supérieurs ou égales à 10.
    k ← 0
    Pour i Allant de 1 à n Faire
        Si Moy[i] >= 10 Alors
            k ← k + 1
        FinSi
    FinPour
    //Affichage du résultat
    Afficher("Le nombre de moyennes supérieurs ou égales à 10 est ", k)
Fin
```



# Les structures / Objets

**Exemple :** Déclaration d'une structure nommée Etudiants

```
public class Etudiants{  
    String Nom;  
    String Prenom;  
    Int Age;  
    Float Moyenne;  
}
```

## Instanciation de l'objet

Après avoir défini un nouvel objet, on peut l'utiliser comme un objet normal. Or l'instance d'objet se fait grâce au mot clé **new**.

**Syntaxe :**      NomObjet NObj1 = new NomObjet();

**Exemple :**      Etudiants Etud1 = new Etudiants();

# Les structures / Objets

## Comment mettre des données dans un objet ?

Le plus simple est de créer une méthode qui récupère ces paramètres et initialise les variables de notre objet, ce qui achèvera notre constructeur d'initialisation. Pour cela, il faut ajouter dans la déclaration de la classe NomObjet, la méthode suivante :

```
public class Nomobjet{  
    .....  
    public NomObjet(<Type> pnchp1, <Type> pnchp2)  
    {  
        nomchp1 = pnchp1;  
        nomchp2 = pnchp2;  
    }  
}
```

Par exemple, dans ce cas, l'objet Etudiants peut-être déclarer ainsi

```
Etudiants Etud1 = new Etudiants("SAUNIER", "Remi", 18, 14.1);
```

# Les structures / Objets

## L'accès au champ d'un objet

Pour cela, en JAVA, on utilise un accesseur qui est une méthode qui va nous permettre d'accéder aux variables de nos objets en lecture. On ajoute dans la déclaration de la classe NomObjet, la méthode suivante :

```
public class Nomobjet{  
    .....  
    public <type> getNomChp1() {  
        return NomChp1;  
    }
```

Par exemple, pour l'objet Etudiants on peut-être déclarer :

```
public String getNom() {  
    return Nom;  
}
```

# Les structures / Objets

## Remplir le champ d'un objet

Pour cela, en JAVA, on utilise un mutateur qui est une méthode qui va nous permettre d'accéder aux variables de nos objets en écriture. On ajoute dans la déclaration de la classe NomObjet, la méthode suivante :

```
public class Nomobjet{  
    .....  
    public void setNomChp1(<type> pnchp1) {  
        NomChp1=pnChp1;  
    }
```

Par exemple, pour l'objet Etudiants on peut-être déclarer :

```
public void setNom(String pNom) {  
    Nom=pNom;  
}
```

# Les structures / Objets

## En Résumé

On peut ajouter plusieurs types de méthodes dans un objet, qui se définissent ainsi :

- les constructeurs -> méthodes servant à créer des objets ;
- les accesseurs -> méthodes servant à accéder aux données des objets ;
- les méthodes d'instance → méthodes servant à la gestion des objets.

**Exercice :** Écrire un algorithme permettant de remplir la fiche de tous les étudiants de la classe.

## Rappel Exercice

Algorithme fiche

//Définition de la structure classe

Type Structure Classe

    nom : chaîne

    prénom : chaîne

    age : entier

    moyenne : réel

FinStructure

//M : taille maximale du tableau

Constante M = 100

//Définition des variables

Variable étudiant : tableau[1 .. M] de classe

    i, n : entier

Début

    Afficher("Entrer nb élève de la classe")

    Saisir(n)

//Remplissage du tableau étudiant

    Pour i Allant de 1 à n Faire

        Afficher("Entrer le nom ", i)

        Saisir(étudiant[i].nom)

        Afficher("Entrer le prénom ", i)

        Saisir(étudiant[i].prénom)

        Afficher("Entrer l'age ", i)

        Saisir(étudiant[i].age)

        Afficher("Entrer la moyenne ", i)

        Saisir(étudiant[i].moyenne)

    FinPour

Fin

# Exercice

## Exercice 6 : Structure imbriquée

Etudiant est une structure composée de trois champs : nom, prénom et date\_de\_naissance. Nom et prénom sont de type chaîne. Date\_de\_naissance est de type N\_date. N\_date est une structure composée de trois champs : jour, mois et année.

Jour et année de type entiers, mois est de type chaîne.

Créer l'objet Etudiant et les méthodes permettant d'accéder facilement à tous les champs de l'objet étudiant.

Ecrire un algorithme qui permet de saisir et d'afficher l'année de naissance d'un étudiant.

## Rappel Exercice 6

```
Algorithme Année_de_naissance
//Déclaration de la structure N_Date
Type Structure N_Date
    Jour : entier
    Mois : chaîne
    Année : entier
FinStruct
//Déclaration de la structure Etudiant
Type Structure Etudiant
    Nom : chaîne
    Prénom : chaîne
    Date_naissance : N_date
FinStruct
//Déclaration de la variable etudiant1 et type Etudiant
Variable etudiant1 : Etudiant
//Algorithme principal
Début
    Afficher("Entrer l'année de naissance de l'étudiant1")
    Saisir(etudiant1.date_naissance.année)
    Afficher("L'année de naissance de l'étudiant1 est :", etudiant1.date_naissance.année)
Fin
```