

Tableaux

Objectifs

- Appréhender la notion de tableaux
- Décrire les tableaux en pseudo code
- Apprendre à code des tableaux en C#

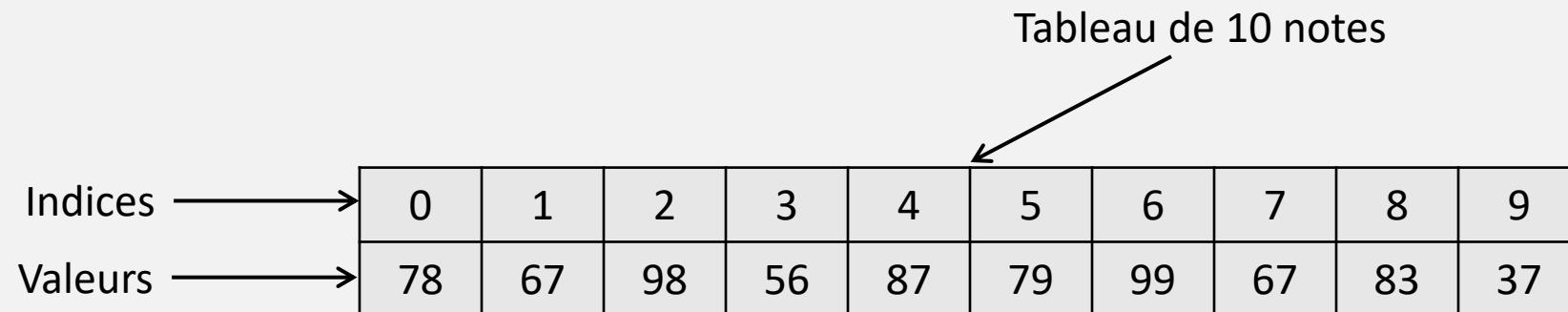
Problème

- Comment représenter une collection de noms ?
 - Une variable de type chaîne par nom ?
⇒ Beaucoup de variables, comment les déclarer, les nommer ?
- Comment représenter une collection de notes ?
 - Une variable de type réel par note ?
 - Beaucoup de variables, ...

Tableau

Un **tableau** est une **structure de données** représentant une **collection de valeurs du même type identifiées par un indice ou des indices**

Tableau 1D



- Un tableau est une collection de valeurs
- Le premier indice est 0
- Le dernier indice est donc la capacité du tableau – 1
⇒ un indice est donc un entier positif
- Un tableau a une capacité fixe

Tableau 1D – Déclaration et capacité

- Déclaration et initialisation :

`<type>[] <nomVariable> = créer <type>[<capacité>];`

ou

`<type>[] <nomVariable> = null;`

- Exemple : `réel[] notesEtudiants = créer réel[10];`

⇒ Déclaration d'une variable de type référence d'un tableau 1 dimension de réels

⇒ Initialisation avec un tableau d'une capacité de 10 réels indexés de 0 à 9

- Obtenir la capacité :

– `<nomVariable>.Capacité`

Tableau 1D - Accès

- Pour accéder à une valeur, on utilise l'opérateur « [] » en lui donnant l'indice dont on veut la valeur :
<variableTypeTableau>[<indice>] : agit comme une variable que vous avez l'habitude de manipuler
- Exemple :

```
réel[] notesEtudiants = créer réel[10];
notesEtudiants[0] = 78;
notesEtudiants[1] = 67;
[...]
écrireNL("La première note est : " + notesEtudiants[0].VersChaine());
```

Exemples d'algorithmes classiques

- Lecture :
 - Lire chaque valeur, une à une
- Affichage :
 - Afficher chaque valeur, une à une séparée par une virgule
- Recherche d'une valeur :
 - Chercher si une valeur est présente dans un tableau
 - Trouver l'indice de la première valeur égale à celle cherchée

Exemples d'algorithmes classiques

- Statistiques :
 - Minimum
 - Maximum
 - Moyenne
 - Médiane
 - Etc.
- Tri :
 - Trier les valeurs du tableau

Exemple – Saisie de notes

Écrire un algorithme qui :

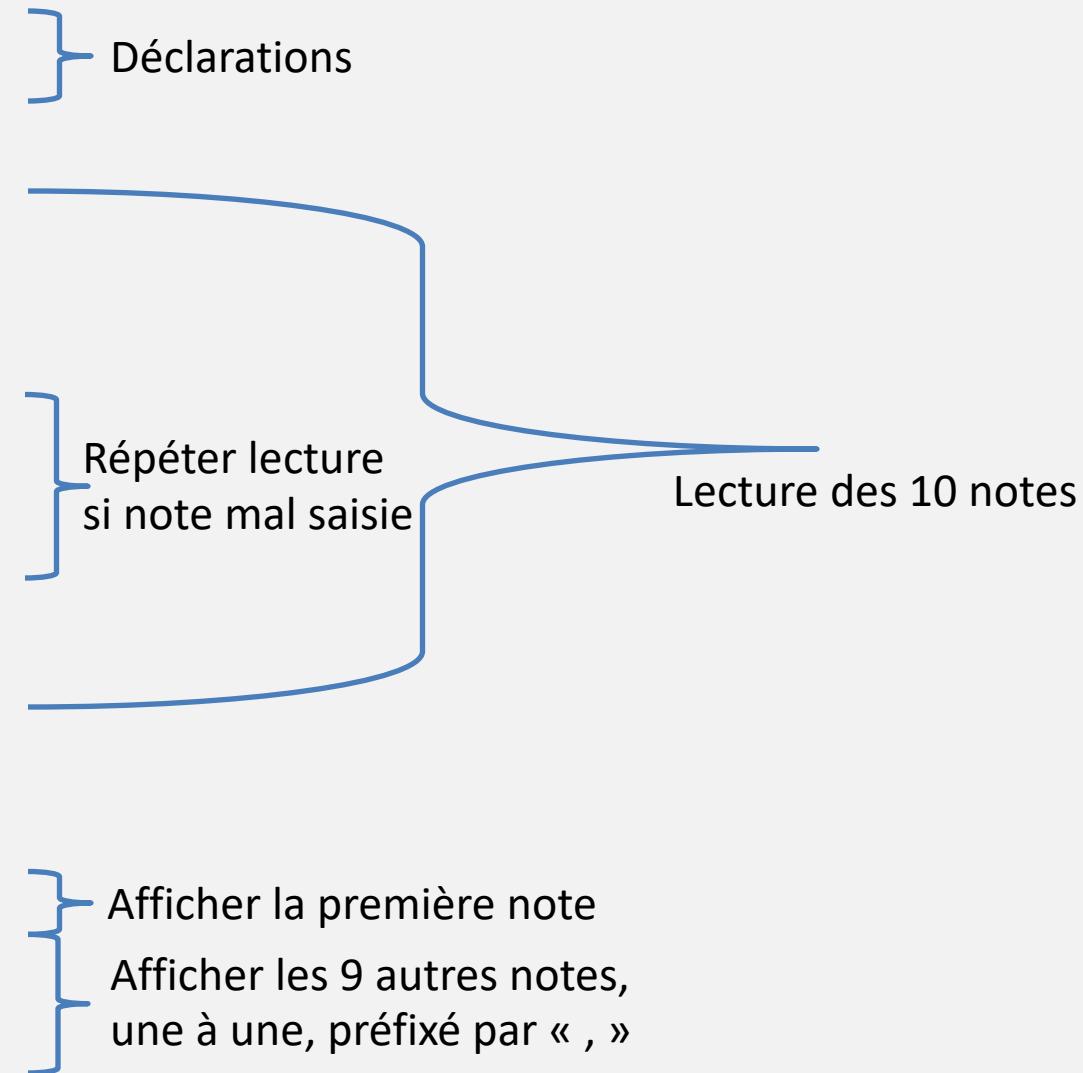
- lit 10 notes comprises entre 0 et 100
- affiche ces 10 notes

Exemple

```
réel[] notesEtudiants = créer réel[10];
réel noteCourante = 0.0;

écrireNL("Entrez 10 notes comprises entre 0 et 100");
pour entier numeroNote de 0 à notesEtudiants.Capacité - 1 {
    écrire("Entrez la note " + numeroNote.VersChaine() + " : ");
    noteCourante = lire()
    tant que noteCourante < 0 ou noteCourante > 100 faire {
        écrireNL("Note invalide !");
        noteCourante = lire();
    }
    notesEtudiants[numeroNote] = noteCourante;
}
écrireNL();

écrireNL("Voici les notes entrées : ");
écrire(notesEtudiants[0]);
pour entier numeroNote de 1 à notesEtudiants.Capacité - 1 {
    écrire(", " + notesEtudiants[numeroNote].VersChaine());
}
```



Et C# ?

- Déclaration et initialisation :

`<type>[] <nomVariable> = new <type>[<capacité>];`

ou

`<type>[] <nomVariable> = null;`

- Exemple : `decimal[] notesEtudiants = new decimal[10];`
- Obtenir la capacité :
 - `<nomVariable>.Length`

Et C# ?

- Comme en algorithmique, on utilise l'opérateur « [] »:
<variableTypeTableau>[<indice>] : agit comme une variable que vous avez l'habitude de manipuler
- Exemple :

```
decimal[] notesEtudiants = new decimal[10];
notesEtudiants[0] = 78m;
notesEtudiants[1] = 67m;
[...]
Console.Out.WriteLine("La première note est : " + notesEtudiants[0].ToString());
```

Exemple

```
réel[] notesEtudiants = créer réel[10];
réel noteCourante = 0.0;

écrireNL("Entrez 10 notes comprises entre 0 et 100");
pour entier numeroNote de 0 à notesEtudiants.Capacité - 1 {
    écrire("Entrez la note " + numeroNote.VersChaine() + " : ");
    noteCourante = lire()
    tant que noteCourante < 0 ou noteCourante > 100 faire {
        écrireNL("Note invalide !");
        noteCourante = lire();
    }
    notesEtudiants[numeroNote] = noteCourante;
}
écrireNL();

écrireNL("Voici les notes entrées : ");
écrire(notesEtudiants[0]);
pour entier numeroNote de 1 à notesEtudiants.Capacité - 1 {
    écrire(", " + notesEtudiants[numeroNote].VersChaine());
}
```

```
decimal[] notesEtudiants = new decimal[10];
decimal noteCourante = 0.0m;

Console.Out.WriteLine("Entrez 10 notes comprises entre 0 et 100");
for (int numeroNote = 0; numeroNote < notesEtudiants.Length; numeroNote++) {
    Console.Out.Write("Entrez la note " + numeroNote.ToString() + " : ");
    noteCourante = Console.In.ReadDecimal();
    while (noteCourante < 0 || noteCourante > 100) {
        Console.Out.WriteLine("Note invalide !");
        noteCourante = Console.In.ReadDecimal();
    }
    notesEtudiants[numeroNote] = noteCourante;
}
Console.Out.WriteLine();

Console.Out.WriteLine("Voici les notes entrées : ");
Console.Out.Write(notesEtudiants[0]);
for (int numeroNote = 1; numeroNote < notesEtudiants.Length; numeroNote++) {
    Console.Out.Write(", " + notesEtudiants[numeroNote].ToString());
}
```