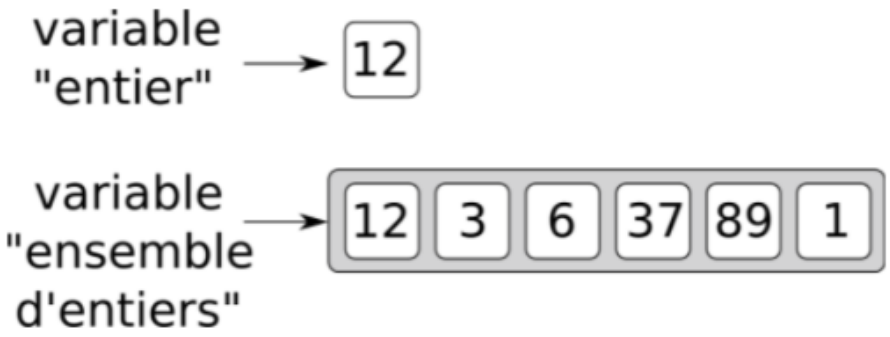
Besoin

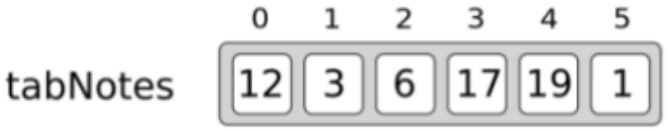
- Mémoriser un ensemble de données **du même type** et le plus souvent liées sémantiquement.
- Effectuer des opérations sur cet ensemble :
  - Mettre à jour / consulter
  - Rechercher
  - Compter



Structure de données : le tableau

- Le tableau est une structure de données **permettant de ranger, sous le même nom, plusieurs éléments de même type**.
- **Le nombre d'éléments d'un tableau est fixé.**
- Les éléments d'un tableau **sont repérés par leur position dans le tableau, appelée *indice***.
- L'élément d'indice **i** d'un tableau **T** est noté **T[i]**.
- **Les indices commencent le plus souvent à 0** (java, C, ...).

Exemple : les notes d'un étudiant.



Structure de données : le tableau

Déclaration

```
Variables
        tabNotes : tableau[6] d'entiers
```

Lecture dans un tableau

```
écrire(tabNotes[2])
a := tabNotes[3]
```

Écriture dans un tableau

```
tabNotes[2] := 11
tabNotes[3] := tabNotes[0]
tabNotes[4] := a*b
lire(tabNotes[1])
```

Initialisation d'un tableau

```
Variables
        tabNotes : tableau[6] d'entiers = {1,15,16,2,7,11}
        nbJoursDuMois  : tableau[] d'entiers = {31,28,31,30,31,30,31,31,30,31,30,31}
```

- Si la taille est spécifiée, le nombre de valeurs doit correspondre à cette taille.
- Si la taille 'est pas spécifiée, c'est le nombre de valeurs fournies qui la détermine.

Structure de données : le tableau

Tableau de constantes

```
Constantes
        NB_JOURS_DU_MOIS   : tableau[] d'entiers = {31,28,31,30,31,30,31,31,30,31,30,31}
```

- Le nombre de valeurs spécifiées détermine la taille.
- On peut toutefois préciser la taille si cela apporte de la clarté.

Structure de données : le tableau

Peuplement/Consultation d'un tableau

- Pour *peupler* un tableau avec des valeurs fournies par l'utilisateur, on doit lire les valeurs une à une (en provenance du clavier, d'un fichier, du réseau,...).
- Il en va de même pour la consultation des valeurs rangées dans un tableau.

⇒ Une boucle est donc nécessaire pour lire ou écrire un ensemble de valeurs dans un tableau.

Exemple

```
Constantes
        TAILLE entier = 6
Variables
        tabNotes : tableau[TAILLE] d'entiers
Début
        // "boucle de peuplement"
        POUR i de 0 à TAILLE-1 FAIRE
                lire(tabNotes[i])
        FINPOUR

        // "boucle de consultation"
        POUR i de 0 à TAILLE-1 FAIRE
                écrire(tabNotes[i])
        FINPOUR
Fin
```

Ne pas sortir des limites du tableau : indices de **0** à **TAILLE-1**

Le parcours peut être effectué dans le sens décroissant des indices

Tableaux en Java

Déclaration

- Forme générale:

```
<type>[] <nom>;
```

- Exemple:

```
int[] tMesValeurs;
```

- Avant de pouvoir utiliser un tableau, il faut le *créér* (allocation mémoire) :

```
int[] tMesValeurs = new int[8];
```

- On peut aussi combiner allocation et initialisation (variante) :

```
int[] tMesValeurs =  { 5, 9, -4, 2, 12, 17, 2, 7 };
```

- La taille du tableau peut également n'être définie que lors de l'exécution :

```
int[] tMesValeurs = new int[ taille ];
```

Tableaux en Java

Utilisations types

Éviter des répétitions

```
class MonthsTab {
    public static void main(String[] args){
        /*
         * on déclare un tableau constant
         */
        final String[] mois = { "", "janvier", "février", "mars", "avril",
                                "mai", "juin", "juillet", "août", "septembre",
                                "octobre", "novembre", "décembre"};
        int i ;
        for (i=1; i<mois.length; i++)
            System.out.println( mois[i] );
    }
}
```

Tableaux en Java

Utilisations types

Intervertir deux éléments

```
class SwapTabElem {
    public static void main(String[] args){ // args est un tableau de Strings
        if (args.length < 2)                // condition sur la dimension du tableau
            System.out.println("Il faut au moins 2 arguments");
        else {
            int i ;
            for (i=0; i<args.length ; i++)      // affichage des arguments
                System.out.print(args[i]+", ") ;//
            System.out.println();
            String temp = args[0] ;             //
            args[0] = args[1] ;                 // échange des deux premiers
            args[1] = temp ;                    //
            for (i=0; i<args.length ; i++)
                System.out.print(args[i]+", ") ;
            System.out.println();
        }
    }
}
```

Java language support for arrays

Basic support

operation	*typical code*
Declare an array	`double[] a;`
Create an array of a given length	`a = new double[1000];`
Refer to an array entry by index	`a[i] = b[j] + c[k];`
Refer to the length of an array	`a.length;`

Initialization options

no need to use a loop like

```
for (int i = 0; i < 1000; i++)
  a[i] = 0.0;
```

operation	*typical code*
Default initialization to 0 for numeric types	`a = new double[1000];`
Declare, create and initialize in one statement	`double[] a = new double[1000];`
Initialize to literal values	`double[] x = { 0.3, 0.6, 0.1 };`

BUT cost of creating an array is proportional to its length.

7

Programmation avec les tableaux : erreurs typiques

Array index out of bounds

```
double[] a = new double[10];
for (int i = 1; i <= 10; i++)
   a[i] = Math.random();
```

No a[10] (and a[0] unused)

Uninitialized array

```
double[] a;
for (int i = 0; i < 9; i++)
   a[i] = Math.random();
```

Never created the array

Undeclared variable

```
a = new double[10];
for (int i = 0; i < 10; i++)
   a[i] = Math.random();
```

What type of data does a refer to?

Ecrivez un programme java constituant un tableau, à partir de deux tableaux de même longueur préalablement saisis. **Le nouveau tableau sera la somme (une méthode) des éléments des deux tableaux de départ**.

Tableau 1 :

4	8	7	9	1	5	4	6

Tableau 2 :

7	6	5	2	1	3	7	4

Tableau à constituer :

11	14	12	11	2	8	11	10

```java
import java.util.*;
public class TestFonctionTables {

    static final Scanner input=new Scanner(System.in);
    static final int bornMax=10;


    static int[] generateTab( int tab[],int n) {
              for(int it=0;it<n;it++){
                tab[it]=(int)(Math.random()*bornMax);
                //-(bornMax/2);
                //System.out.println(tab[it]);
              }
            return tab;
      }

    static int[] additionTab( int tab1[],int tab2[],int n) {
            int sommeTab[]=new int[n];
            for(int it=0;it<n;it++){
                sommeTab[it]=(tab1[it]+tab2[it]);
                //System.out.println(tab[it]);
              }
            return sommeTab;
      }


    static void printTab( int tab[], int n) {
                //tab.length
              for(int it=0;it<n;it++)
                System.out.print(tab[it]+"\t");
            System.out.println("");
      }

    public static void main(String[] args) {

        int n;
        do{
        System.out.println("Entrer n:");
        n=input.nextInt();
        }while(n<1);

        int tab1[]=new int[n];
        int tab2[]=new int[n];
        int sommeTab[]=new int[n];


        generateTab(tab1,n);
        System.out.print("Table 1=\t");
        printTab(tab1,n);

        System.out.print("\n Table 2=\t");
        generateTab(tab2,n);
        printTab(tab2,n);
        System.out.print("\n Table1+ Table 2:");
        sommeTab=additionTab(tab1,tab2,n);
        printTab(sommeTab,n);

    }
}
```