Les structures itératives

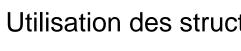


Pourquoi les boucles?

Supposons qu'on veuille afficher 20 fois le mot Bonjour

```
public class Programme {
public static void main(String[] args) {
 System.out.println("Bonjour");
 System.out.println("Bonjour");
```









Structures itératives Les types

- o Elles sont de 2 types:
 - 1. À bornes non définies: utilisées lorsque le nombre d'itérations est **inconnu**,
 - Boucle while
 - Boucle do...while

- 2. À bornes définies: utilisées lorsque le nombre d'itérations est **connu**,
 - Boucle for



Structures itératives La boucle while

Fonction: répéter une suite d'instructions tant qu'une condition est vraie

```
Syntaxe: while (conditionVraie){
    instructions;
}
```

Remarques:

- Si la condition est fausse dès le départ, les instructions ne sont jamais exécutées
- Si la condition est toujours vraie, le programme ne s'arrêtera jamais

Boucle while Exemple d'usage

```
public class Programme {
public static void main(String[] args) {
   System.out.println("Bonjour");
   System.out.println("Bonjour");
   System.out.println("Bonjour");
  System.out.println("Bonjour");
                                             public class Programme {
   System.out.println("Bonjour");
  System.out.println("Bonjour");
  System.out.println("Bonjour");
                                                 public static void main(String[] args) {
   System.out.println("Bonjour");
                                                    int nb=0;
   System.out.println("Bonjour");
                                                    while (nb<20) {
  System.out.println("Bonjour");
                                  Utilisation de la
                                                         System.out.println("Bonjour");
   System.out.println("Bonjour");
                                   boucle while
   System.out.println("Bonjour");
                                                         nb++;
  System.out.println("Bonjour");
   System.out.println("Bonjour");
   System.out.println("Bonjour");
  System.out.println("Bonjour");
   System.out.println("Bonjour");
   System.out.println("Bonjour");
   System.out.println("Bonjour");
   System.out.println("Bonjour");
```



Structure itérative boucle do... while

Fonction: exécuter une suite d'instructions au moins une fois et la répéter jusqu'à ce que la condition soit vraie

```
Syntaxe: do{
    instructions;
}while(conditionVraie);
```

Remarque: Les instructions sont exécutées au moins une fois



Boucle do...while Exemple d'usage

```
public class Programme {
 public static void main(String[] args) {
  System.out.println("Bonjour");
  System.out.println("Bonjour");
  System.out.println("Bonjour");
  System.out.println("Bonjour");
                                                        public class Programme {
  System.out.println("Bonjour");
  System.out.println("Bonjour");
                                                           public static void main(String[] args) {
  System.out.println("Bonjour");
                                                               int nb=0;
  System.out.println("Bonjour");
                                                               do{
  System.out.println("Bonjour");
                                                                  System.out.println("Bonjour");
  System.out.println("Bonjour");
                                         Utilisation de la
  System.out.println("Bonjour");
                                                                  nb++;
                                         boucle do..while
  System.out.println("Bonjour");
                                                               }while(nb<20);</pre>
  System.out.println("Bonjour");
  System.out.println("Bonjour");
  System.out.println("Bonjour");
  System.out.println("Bonjour");
  System.out.println("Bonjour");
  System.out.println("Bonjour");
  System.out.println("Bonjour");
  System.out.println("Bonjour");
```



Boucle à bornes définies

Dans le cas d'une boucle à **bornes définies**, le nombre d'itérations à effectuer est connu, grâce aux valeurs des bornes minimum et maximum fournies dans la définition de la boucle.

La boucle concernée: la boucle for.



Structure itérative boucle for

Fonction: répéter un bloc d'instructions un certain nombre de fois.

Syntaxe:

• Un for Incrémenté:

```
for(int i=valInitiale;i<valeurFinale;i++) {
   instructions;
}</pre>
```

• Un for décrémenté

```
for(int i=valeurFinale;i>=valInitiale;i--) {
   instructions;
}
```

Boucle for

Exemple d'usage

```
public class Programme {
   public static void main(String[] args) {
      System.out.println("Bonjour");
      System.out.println("Bonjour");
      System.out.println("Bonjour");
      System.out.println("Bonjour");
      System.out.println("Bonjour");
    }
}
```

Utilisation de la boucle **for**

```
public class Programme {
   public static void main(String[] args){
     for(int i=0;i<5;i++){
        System.out.println("Bonjour");
     }
   }
}</pre>
```



Exercice sur les structures itératives

Un jeu très amusant consiste à choisir un nombre défini et le faire deviner à l'adversaire en lui disant « plus grand » ou « plus petit ». Ecrire un programme qui devine un nombre choisi par l'utilisateur entre 1 et 100.

- ✓ Supposons que le chiffre que vous cherchiez soit 20, un exemple d'exécution du programme est décrit ci-dessous :
 - Est-ce 45 ? moins, essaye encore
 - Est-ce 35 ? moins, essaye encore
 - Est-ce 15 ? plus, essaye encore
 - Est-ce 16? plus, essaye encore
 - Est-ce 20 ? Oui, Gagné



Les Tableaux



Pourquoi les tableaux?

```
public class Copies {
   public static void main(String[] args ){
      int val1, val2, val3, val4, val5, resultatSomme;
      java.util.Scanner entree =
         new java.util.Scanner(System.in);
      System.out.println("veuillez saisir val1");
      val1 = entree.nextInt();
      System.out.println("veuillez saisir val2");
      val2 = entree.nextInt();
      System.out.println("veuillez saisir val3");
      val3 = entree.nextInt();
      System.out.println("veuillez saisir val4");
      val4 = entree.nextInt();
      System.out.println("veuillez saisir val5");
      val5 = entree.nextInt();
      resultatSomme=val1+val2+val3+val4+val5;
      System.out.println("la somme des valeurs est de
  "+resultatSomme);
```

Programme java qui demande à l'utilisateur la saisie de 5 valeurs, et réalise leurs somme.



Nécessité d'une structure de données pour stocker un ensemble de données [-13-]

Structures de données Les tableaux

Un tableau est un ensemble **ordonné** et **homogène** de valeurs :

• Ordonné car les cases mémoires composants un tableau sont numérotés à partir de 0

• Homogène car un tableau possède des valeurs du même type.

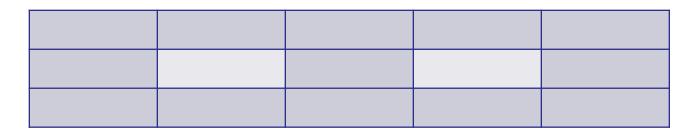


Les tableaux

- -Structure de données permettant d'effectuer un même traitement sur des données de même type.
- -Il y a deux types de tableaux:
- Tableau à **une** dimension



• Tableau à **deux** dimensions





Traitements sur les tableaux

- o Créer des tableaux
- o Trier des valeurs dans un tableau
- o Récupérer, consulter des valeurs dans un tableau
- O Rechercher si une valeur existe dans un tableau
- o Mettre à jour les valeurs d'un tableau
- Effectuer des opérations entre tableaux : comparaison de tableaux, multiplication,...



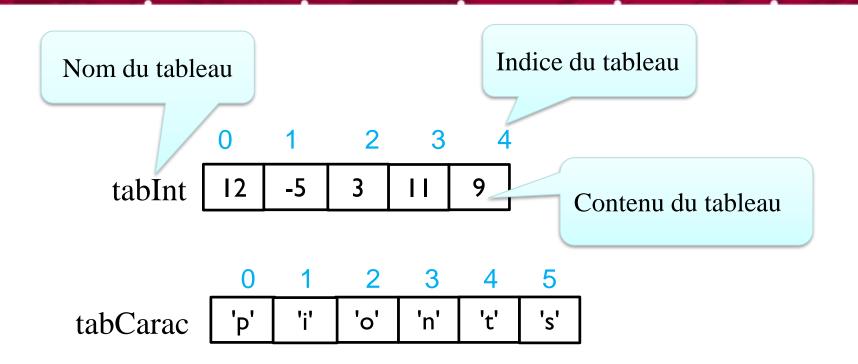
Définition d'un tableau

- On définit un tableau par:
 - Son nom
 - Ses dimensions 1 ou 2
 - Sa taille
 - Le type de données qu'il contient

O Un tableau doit être déclaré, initialisé, et rempli



Contenu d'un tableau à 1 dimension



Remarques:

L'indexation des tableaux commence toujours à 0



Manipulation d'un tableau à 1 dimension en java Exemple: Tableau d'entiers

O Déclaration d'un tableau d'entiers :

int[] tabValeur;

Création d'un tableau

tabValeur= **new int**[10];

On peut déclarer et créer un tableau dans une seul instruction



 La capacité du tableau est fournie par l'instruction nomDuTableau.length int longueur = tabValeur.length;



Manipulation d'un tableau à 1 dimension en java Exemple: Tableau de n'importe quel type

Déclaration et création d'un tableau de:

O Chaines de caractères:
String[] tabValeur= new String[10];

o Booléens:

```
boolean[] tabValeur= new boolean[10];
```

O Caractères :

```
char[] tabValeur= new char[10];
```

o Réels:

```
double[] tabValeur= new double[10];
```

On dimensionne le tableau selon le besoin

Utilisation d'un tableau : par les indices

o Accès en lecture

```
System.out.println(tableau[4]); le contenu du tableau à l'indice 4 est affiché à l'écran
```

o Accès en Ecriture :

```
tableau[3] =18; la valeur 18 est placée dans le tableau à l'indice 3
java.util.Scanner entree =new Scanner(System.in);
tableau[5] =entree.nextInt();
```

la valeur entrée par l'utilisateur est enregistrée dans le tableau à l'indice 5

o attention! tableau=18



Affichage et affectation

```
public class Exemple{
  public static void main(String[] args ){
     int[] tabVal = new int[8];
  java.util.Scanner entree = new java.util.Scanner Scanner(System.in);
     for (int i = 0; i < 8; i++) {
      System.out.println("veuillez saisir une valeur à l'indice "+i);
         tabVal[i] = entree.nextInt();
         System.out.println("la valeur saisie est"+tabVal[i]);
```

Exemple 1 avec un tableau Valeurs saisies par l'utilisateur

```
public class Exemple{
public static void main(String[] args ){
int val1,val2,val3, val4, val5, resultatSomme;
java.util.Scanner entree =new java.util.Scanner
Scanner(System.in);
System.out.println("veuillez saisir la valeur de val1");
val1 = entree.nextInt();
System.out.println("veuillez saisir la valeur de val2");
val2 = entree.nextInt();
System.out.println("veuillez saisir la valeur de val3");
val3 = entree.nextInt();
System.out.println("veuillez saisir la valeur de val4");
val4 = entree.nextInt();
System.out.println("veuillez saisir la valeur de val5");
val5 = entree.nextInt();
resultatSomme=val1+val2+val3+val4+val5;
System.out.println("La somme est de : "+resultatSomme);
```

```
public class Exemple{
public static void main(String[] args ){
int resultatSomme=0;
int[] tabVal = new int[5];
java.util.Scanner entree = new java.util.Scanner(System.in);
for (int i = 0; i < 5; i++) {
     System.out.println("veuillez saisir la valeur "+i );
     tabVal[i] = entree.nextInt();
resultatSomme=resultatSomme+tabVal[i];
System.out.println("La somme est de:"+resultatSomme);
```



Remplissage du tableau Valeurs aléatoires

```
public class Exemple{
public static void main(String[] args ){
int[] tabVal = new int[5];
java.util.Random generateur = new java.util.Random(System.currentTimeMillis());
for (int i = 0; i < 5; i++) {
    tabVal[i] = generateur.nextInt(20);
    System.out.println("la valeur remlpie est:"+ tabVal[i]);
```



Tableau à deux dimensions

matriceDEntiers

	0	1	2	3	4	5	6
0	10	3	25	14	2	1	8
1	9	20	7	12	2	4	7

Tableau à 2 lignes et 7 colonnes

Nombre de lignes



int[][] matriceDEntiers= new int[2][7];



Nombre de colonnes



Tableau à deux dimensions

O Accès en lecture :

```
System.out.println(matriceDEntiers[1][6]);
```

la valeur contenue en ligne 1 colonne 6 est affichée à l'écran

O Accès en écriture :

```
matriceDEntiers[1][4]= 36
```

la valeur fournie est enregistrée en ligne 1, colonne 4

```
java.util.Scanner entree = new Scanner(System.in);
matriceDEntiers[0][4]=entree.nextInt();
```



Manipulation d'un tableau à 2 dim. en java Exemple: Tableau de n'importe quel type

Déclaration et création d'un tableau de:

O Chaines de caractères:

```
String[][] tabValeur= new String[10][10];
```

o Booléens:

```
boolean[][] tabValeur= new boolean[10][10];
```

o Caractères:

```
char[][] tabValeur= new char[10][10];
```

o Réels:

```
double[][] tabValeur= new double[10][10];
```

On dimensionne le tableau selon le besoin

Saisie dans un tab. à 2 dim.

```
public class Exemple{
   public static void main(String[] args ){
      int[][] matriceDEntiers= new int[2][7];
     java.util.Scanner entree = new java.util.Scanner(System.in);
      for (int i = 0; i < 2; i++) {//parcourir lignes</pre>
         for (int j = 0; j < 7; j++) {//parcours colonnes
            System.out.println("Saisie pour ["+i +"]["+j+ "]");
            matriceDEntiers[i][j] = entree.nextInt();
```

Affichage d'un tab à 2 dim

```
public class Exemple{
 public static void main(String[] args ){
    int[][] matriceDEntiers= new int[2][7];
    java.util.Scanner entree = new java.util.Scanner(System.in);
      //Pour parcourir chaque ligne
      for (int i = 0; i < 2; i++) {
        //Pour parcourir chaque colonne
         for (int j = 0; j < 7; j++) {
        matriceDEntiers[i][j]= entree.nextInt(); //remplissage de la matrice
        //Affichage du contenu de la matrice
             System.out.println(" matriceDEntiers ["+i +"]["+ j + "] vaut
                         "+matriceDEntiers [i][j] );
                                                                         29
```

Recherche dans un tableau

 Utilisation d'une variable booléenne: technique du flag

Un peu d'Histoire!!!

Le flag, en anglais, est un petit drapeau, qui va rester baissé aussi longtemps que l'événement attendu ne se produit pas. Et, aussitôt que cet événement a lieu, le petit drapeau se lève (la variable booléenne change de valeur). Ainsi, la valeur finale de la variable booléenne permet au programmeur de savoir si l'événement a eu lieu ou non.



Technique du flag

```
public class Flag{
                                                                                     Qu'est ce qui
   public static void main(String[] args ){
        int[] tabVal = new int[35];
                                                                                     cloche??
        java.util.Scanner entree = new java.util.Scanner(System.in);
        java.util.Random generateur = new java.util.Random(System.currentTimeMillis());
        for (int i = 0; i < 35; i++) {
                                                                                       🖳 Declaration 📃 Console 🖂
                                                                                       <terminated> Flag [Java Application
            tabVal[i]= generateur.nextInt(25);
                                                                                       veuillez saisir la valeur à
                                                                                       Valeur non trouvée
                                                                                       Valeur non trouvée
                                                                                       Valeur non trouvée
        System.out.println("veuillez saisir la valeur à chercher");
                                                                                       Valeur non trouvée
                                                                                       Valeur non trouvée
        int x=entree.nextInt();
                                                                                       Valeur non trouvée
                                                                                       Valeur non trouvée
                                                                                       Valeur non trouvée
        for (int i = 0; i < 35; i++) {
                                                                                       Valeur non trouvée
                                                                                       Valeur non trouvée
            if (tabVal[i]==x) {
                                                                                       Valeur non trouvée
                                                                                       Valeur non trouvée
                                                                                       Valeur non trouvée
                 System.out.println("Valeur trouvée !");
                                                                                       Valeur trouvée !
                                                                                       Valeur non trouvée
           }else {
                                                                                       Valeur non trouvée
                                                                                       Valeur non trouvée
                                                                                       Valeur non trouvée
                 System.out.println("Valeur non trouvée");
                                                                                       Valeur non trouvée
                                                                                       Valeur non trouvée
                                                                                       Valeur non trouvée
                                                                                       Valeur trouvée !
                                                                                       Valeur non trouvée
                                                                                       Valeur non trouvée
                                                                                       Valeur non trouvée
}}
                                                                                       Valeur non trouvée
                                                                                       Valeur trouvée !
                                                                                       Valeur non trouvée
                                                                                       Valeur non trouvée
                                                                                       Valeur non trouvée
                                                                                       Valeur non trouvée
```

Technique du flag



```
public class Flag {
   public static void main(String[] args ){
      boolean trouve=false;
      int[] tabVal = new int[35];
      java.util.Scanner entree = new java.util.Scanner(System.in);
      java.util.Random generateur = new java.util.Random(System.currentTimeMillis());
    for (int i = 0; i < 35; i++) {
       tabVal[i]= generateur.nextInt(25);}
       System.out.println("veuillez saisir la valeur à chercher");
       int x=entree.nextInt();
       int i = 0;
                                                 Declaration ☐ Console ☒ \ ■ Properties
     while ((trouve == false) && (i<35)) {
                                                 <terminated> Flag [Java Application] C:\Program F
         if (tabVal[i]==x) {
                                                 veuillez saisir la valeur à chercher
           trouve=true;
                                                 10
         } else { i++; }
                                                 la valeur est bien dans le tableau
      if(trouve == false){
      System.out.println("Désolée, la valeur n'est pas dans le tableau");
      }else{ System.out.println("la valeur est bien dans le tableau");}
                                                                                    32
  }}
```

Exercice

• Ecrire un programme java dont lequel 10 valeurs aléatoires sont générées et stockées dans un tableau à 1 dimension. Ensuite, le programme affiche: la somme, le minimum, le maximum et la moyenne de ces valeurs.

