Chapitre 4

Faire des choix

1. L'instruction `if-else`

Similaire à C++ :

```
if (condition)
{
     Ensemble d'instructions;
}
```

Comme en C ou C++, s'il n'y a qu'une seule instruction, alors les accolades ne sont pas obligatoires.

Avec un bloc « else » :

```
if (condition)     // Si la condition est vraie
{
    instructions;
}
else               // sinon (la condition est fausse)
{
    instructions;
}
```

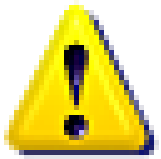
1.1 Comment écrire une condition

1.1.1 Les opérateurs relationnels

La condition doit être booléenne (vraie ou fausse).

Cette condition peut être composée d'expressions arithmétiques ou de valeurs numériques que l'on compare entre elles avec les mêmes opérateurs qu'en C++ :

== , < , <= , > , >= , !=



Attention

Comme en C++, l'opérateur de **test d'égalité** est "==".

Le signe "=" correspond à une **affectation**.

1.1.2 Les opérateurs logiques

Ils permettent d'associer plusieurs conditions simples. Ce sont les mêmes qu'en C++.

Opérateur	Signification
!	**NON** logique
&&	**ET** logique
| |	**OU** logique

Exemples :

```
int x=3, y=5, z=2, r=6;
```

- La condition : `(x<y) && (z<r)` est vraie
- La condition : `(x>y) || (z<r)` est vraie
- La condition : `!(z<r)` est fausse

1.2 Des `if-else` imbriqués.

Il est possible de placer des structures **`if-else`** à l'intérieur d'**`if-else`**. On dit alors que les structures **`if-else`** sont imbriquées.
Deux notations possibles :

```
if (condition1)
   Instruction1;
else
   if (condition2)
      Instruction2;
   else
      if (condition3)
         Instruction3;
      else
         Instruction4;
```

```
if (condition1)
   Instruction1;
else if (condition2)
   Instruction2;
else if (condition3)
   Instruction3;
else
   Instruction4;
```

2. L'instruction `switch`, ou comment faire des choix multiples

Lorsque le nombre de choix devient plus grand que deux, l'utilisation de **`if-else`** devient rapidement fastidieuse.

→ Utilisation de **`switch`** comme en C++.

2.1 Construction du `switch`

```
switch (valeur)
{
  case etiquette1 :
    // une ou plusieurs instructions
     break;

  case etiquette2 :
  case etiquette3 :
     // Une ou plusieurs instructions
     break;

  default :
     // Une ou plusieurs instructions
}
```

La variable **`valeur`** est évaluée. Suivant sa valeur, le programme recherche l'étiquette correspondante :

- Si le programme trouve une étiquette correspondant au contenu de la variable **`valeur`**, il exécute la ou les instructions qui suivent l'étiquette, jusqu'à rencontrer le mot-clé **`break`**.

- S'il n'existe pas d'étiquette correspondant à **`valeur`**, alors le programme exécute les instructions de l'étiquette **`default`**.

Remarques :

- Le type de variable **`valeur`** ne peut être que **`char`**, **`int`**, **`byte`**, **`short`**, **`long`**, **`String`**. Il n'est pas possible de tester des valeurs réelles (**`float`**, **`double`**).

- Une étiquette peut contenir 0, 1 ou plusieurs instructions.

- L'instruction **`break`** permet de sortir du bloc **`switch`**. S'il n'y a pas de **`break`** pour une étiquette donnée, le programme exécute les instructions de l'étiquette suivante.

2.2 Exemple

Écriture d'un programme qui simule une machine à calculer dont les opérations sont l'addition (+), la soustraction (-), la multiplication (*) et la division (/).

Le programme demande à l'utilisateur d'entrer deux valeurs numériques, puis le caractère correspondant à l'opération à effectuer. Suivant le caractère entré ('+', '-', '*', '/') le programme affiche l'opération effectuée, ainsi que le résultat.

```
// A taper dans un fichier Calculette.java

import java.util.*

public class Calculette
{
    public static void main(String[] argument)
    {
        int     a, b;
        char    opérateur;
        double  calcul = 0;
        boolean OK = true;
        Scanner clavier = new Scanner(System.in);
```

```
System.out.print("Entrer la première valeur : ");
a = clavier.nextInt();

System.out.print("Entrer la seconde valeur : ");
b = clavier.nextInt();

System.out.print("Type de l'opération(+,-,*,/) : ");
opérateur = clavier.next().charAt(0);
```

```
switch (opérateur)
{
  case '+' : calcul = a + b;
             break;

  case '-' : calcul = a - b;
             break;

  case '/' : if ( b != 0 )
               calcul = a / b;
             else
               OK = false;
             break;

  case '*' : calcul = a * b ;
             break;

  default  : OK = false ;
}
```

```
    if (OK == true)
    {
      System.out.print("Resultat de l'operation : ");
      System.out.println(a+ opérateur +b+ "=" + calcul);
    }
    else
      System.out.println("Operation non conforme !");
  }
}
```

Résultats d'exécutions :

```
C:\Java>java Calculette
Entrer la première valeur : 2
Entrer la seconde valeur : 3
Type de l'opération : (+, -, *, /) : *
Cette operation a pour résultat : 2*3 =  6.0
```

```
C:\Java>java Calculette
Entrer la première valeur : 2
Entrer la seconde valeur : 0
Type de l'opération : (+, -, *, /) : /
Operation non conforme !
```