

Programmation avec Java: notions de base

N.KANAOU

Programme

- Introduction à la programmation informatique et aux différents langages de programmation.
- Présentation des concepts de base de la programmation de haut niveau. Aspect algorithmique. Introduction au langage Java.
- Syntaxe de base du langage Java, la sémantique et le concept de la compilation.
- Les variables, types et opérations mathématiques
- Opérateurs, affectation,
- conditionnelles et boucles.
- Tableaux (accès aux composants, ...).
- Procédures et Fonctions.
- Présentations de quelques classes clés.
- Des algorithmes fondamentaux.
- Des exemples applicatifs de l'ensemble des notions de Java présentées.

Aujourd'hui

conditionnelles et boucles

- Instruction conditionnelles.
- Instruction if
- Instruction if else
- Instruction if-else if
- Instruction switch
- exercices applicatifs

Instructions conditionnelles

Définition:

Les instructions conditionnelles, ou instructions de test , permettent de faire des choix dans un un programme. Elles permettent d'altérer le déroulement du programme en fonction de la valeur de conditions.

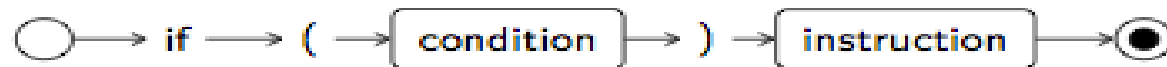
Il y a trois instructions conditionnelles en Java:

- 1.l'instruction if,
- 2.l'instruction if-else,
- 3.l'instruction switch.

INSTRUCTION IF (1/7)

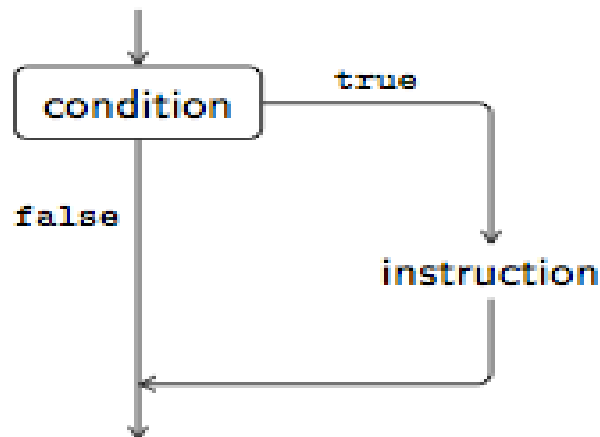
- L'instruction if se retrouve dans de nombreux langages de programmation.
- Elle permet d'exécuter une instruction si une condition est vérifiée.
- La condition est une expression booléenne et elle est dite vérifiée si sa valeur est true.
- l'instruction if consiste en le mot réservé if suivi d'une *condition entre* parenthèses, et enfin d'une instruction.

if (condition is true)
instruction exécutée



INSTRUCTION IF (2/7)

En ce qui concerne l'exécution de l'instruction if, la condition est d'abord évaluée, et si sa valeur est true, alors l'instruction est exécutée. Sinon, la valeur de la condition vaut false, et l'instruction est passée. Dans les cas, le programme continue à l'instruction suivante.



Exemple applicatif-1:

calculez a/b avec la condition que le dénominateur est différent de 0.

INSTRUCTION IF (3/7)

Exemple-applicatif-1:

```
package condit_boucles;

public class ClassePrincipales {

    public static void main(String[] args) {
        int a=6;
        int b = 2;
        float c = a/b;

        if (b == 0)
            System.out.println ("Division par 0 impossible !");
        System.out.print ("Résultat est =" +c);

    }
}
```

INSTRUCTION IF (4/7)

Cas de plusieurs instructions

Si on veut exécuter plus d'une instruction en fonction d'une condition, on utilise bloc de code .

```
if (condition is true)
{
    instruction-1 exécutée
    instruction-2 exécutée
    ...
    instruction-n exécutée
}
```

La condition est une expression booléenne et vous pouvez donc construire des conditions complexes en utilisant les opérateurs logiques ET, OU et NON.

INSTRUCTION IF (5/7)

Exemple- applicatif-2:

écrire un programme pour comparer la température T avec la valeur 37°. Si la température est supérieur à 37° , calculer le dépassement en degré par rapport au seuil 37° et afficher un message d'alerte.

Exemple-applicatif-3:

Écrire un programme qui affiche un message à l'écran si la valeur de la variable temperature est comprise entre 36 et 38.

INSTRUCTION IF (6/7)

Exemple applicatif-2

```
{  
    int temperature = 39;  
    int diff = temperature-37;  
    if (temperature > 38)  
        System.out.print ("Attention, température trop élevée :");  
        System.out.println (temperature);  
        System.out.println ("la différence est: " + diff);  
}
```

INSTRUCTION IF (7/7)

//exemple applicatif-3

```
{  
    int temperaturee = 38;  
    if ((temperaturee > 36)& (temperaturee < 39))  
        System.out.print ("la température est ambiante :");  
        System.out.println (temperaturee);  
}
```

INSTRUCTION IF-ELSE (1/5)

Parfois, on aimerait exécuter une instruction si une condition est vérifiée et une autre si la condition n'est pas vérifiée. Une solution pour ce faire est d'utiliser if-else.

La syntaxe de l'instruction if-else est:

```
if (condition is true)
    instruction exécutée
else
    instruction exécutée
```

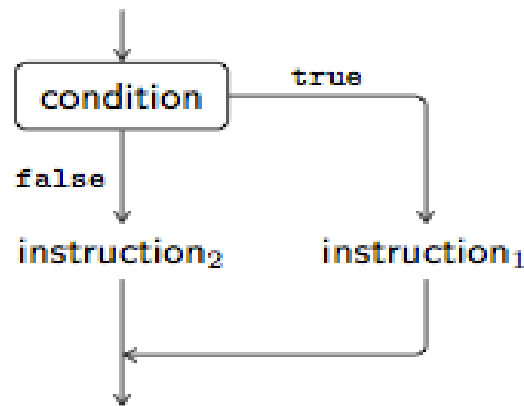
INSTRUCTION IF-ELSE (2/5)



La première partie est exactement la même que pour l'instruction if; vient ensuite la seconde partie constituée du mot réservé else suivi d'une instruction. Lorsque cette instruction est un bloc, il est appelé *bloc* else. Tout comme pour l'instruction if, il est recommandé de toujours en utiliser un.

INSTRUCTION IF-ELSE (3/5)

En ce qui concerne l'exécution, l'instruction if-else représente un aiguillage avec deux sorties possibles. La condition est tout d'abord évaluée et si sa valeur vaut true, instruction-1 est exécutée, sinon c'est instruction-2 qui est exécutée. Le programme continue ensuite son exécution.



INSTRUCTION IF-ELSE (4/5)

Exemple-applicatif-4:

Écrire un programme qui permet d'afficher la moyenne d'un élève et précise si l'élève a réussi ou il a échoué son examen.

INSTRUCTION IF-ELSE (5/5)

//exemple applicatif-4

```
{  
    int points = 14;  
  
    if (points >= 10)  
    {  
        System.out.println ("réussi");  
    }  
    else  
    {  
        System.out.println ("raté");  
    }  
}
```


INSTRUCTION IF-ELSE IF (1/3)

Si on a plusieurs instructions, que chacune de celles-ci ne peut être exécutée que si une certaine condition est vérifiée et qu'on ne veut en exécuter qu'une seule, on va devoir utiliser plusieurs instructions if-else imbriquées les unes dans les autres.

```
if (condition is true)
    instruction exécutée
else
    if (condition is true)
        instruction exécutée
    else
        If ....
```

Exemple applicatif : affichez une lettre correspondance à des points sur 20 en suivant la notation suivante : A (18-20), B (16-18), C (14-16), D (12-14) et F (0-12):

Exemple explicatif : le programme pour afficher une lettre correspondance à des points sur 20 en suivant la notation suivante : A (18–20), B (16–18), C (14–16), D (12–14) et F (0–12):

```
int points=14;
char grade;
if (18 <= points && points <= 20)
{
    grade = 'A';
    System.out.println ("le grade est: " +
grade);
}
else
{
    if (16 <= points && points <18)
    {
        grade = 'B';
        System.out.println ("le grade est:
" + grade);
    }
    else
    {
        if (14 <= points && points < 16)
        {
            grade = 'C';
            System.out.println ("le grade
est: " + grade);
        }
        else
        {
            grade = 'D';
            System.out.println ("le grade est: "
+ grade);
        }
        else
        {
            grade = 'F';
            System.out.println ("le grade est:
" + grade);}}}}}
```

INSTRUCTION IF-ELSE IF (3/3)

Dans l'exemple précédent, le niveau d'indentation augmente au fur et à mesure qu'on avance dans le code, Pour rendre le code plus clair dans de tels cas, il est conseillé de ne pas définir un bloc de code pour les else dont l'unique instruction est une instruction if ou if-else. Donc:

```
int pointss=18;
char grades;
if (18 <= pointss && pointss <= 20)
{
    grades = 'A';
    System.out.println ("le grade est: " +
grades);
}
else if (16 <= pointss && pointss < 18)
{
    grades = 'B';
    System.out.println ("le grade est: " +
grades);
}
else if (14 <= pointss && pointss < 16)
{
    grades = 'C';
    System.out.println ("le grade est: " +
grades);
}
else if (12 <= pointss && pointss < 14)
{
    grades = 'D';
    System.out.println ("le grade est: "
+ grades);
}
else
{
    grades = 'F';
    System.out.println ("le grade est: "
+ grades);}
```

INSTRUCTION SWITCH (1/5)

Cette instruction représente un aiguillage multiple : le choix est fait en fonction de la valeur d'une variable; chaque chemin possible correspond à une valeur différente pour la variable.

La valeur de l'expression du switch est tout d'abord évaluée et le programme se rend directement au case correspondant à cette valeur.

L'instruction switch n'est ni plus ni moins qu'un raccourci d'écriture pour l'instruction if-else if: facilités au niveau de l'écriture du code et de sa lisibilité . Une petite différence: avec switch, la valeur de l'expression est évaluée une seule fois, mais avec l'équivalent if-else if, elle l'est à chaque fois).

INSTRUCTION SWITCH (2/5)

La syntaxe de l'instruction switch est:

```
switch (donnée)
{
    case '1':
        instruction ;
        break;

    case '2':
        instruction ;
        break;

    default:
        instruction ;
}
```

Le cas default correspond au dernier else et n'est pas obligatoire. Si on omet ce cas et qu'aucun case ne correspond à la valeur de l'expression, aucune instruction n'est exécutée.

INSTRUCTION SWITCH (3/5)

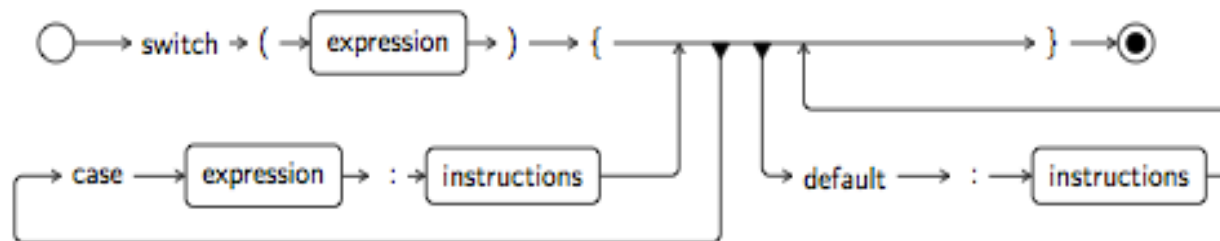
En ce qui concerne l'exécution:

- on commence avec le mot réservé switch suivi d'une expression entre parenthèses.

Vient ensuite un bloc de code qui contient zéro, un ou plusieurs case et éventuellement un default.

- chaque cas est défini par le mot réservé case suivi d'une valeur et de deux-points (:), suivi d'une séquence d'instructions.

- Le cas par défaut est construit avec le mot réservé default suivi de deux-points et d'une séquence d'instructions.



INSTRUCTION SWITCH (4/5)

Exemple d'application: Affichage d'une phrase à l'écran en fonction de la valeur de la variable sexe. Réécrivez le code suivant en utilisant l'instruction switch:

```
char sexe = 'F';

if (sexe == 'M')
{
    System.out.println ("Vous êtes un homme.");
}
else if (sexe == 'F')
{
    System.out.println ("Vous êtes une femme.");
}
else
{
    System.out.println ("Vous êtes de sexe indéterminé.");
}
```

INSTRUCTION SWITCH (5/5)

Solution:

```
char sexe = 'F';

switch (sexe)
{
    case 'M':
        System.out.println ("Vous êtes un homme.");
        break;

    case 'F':
        System.out.println ("Vous êtes une femme.");
        break;

    default:
        System.out.println ("Vous êtes de sexe indéterminé.");
}
```


Exos applicatifs

Exemple applicatif: écrire un programme qui attribue des lettres selon la taille de la chemise en respectant les informations suivantes:

- [34-36[= Taille XS
- [36-38[= Taille S
- [38-40[= Taille SM
- [40-42[= Taille M
- [42-44[= Taille L

Le switch est une solution qui n'est pas adaptable à des valeurs sous forme d'intervalle mais plutôt à des valeurs exactes.