

Leçon 3 : Contrôler des exécutions en JAVA

YAO JEAN YVES ARISTIDE : Enseignant -
Chercheur

AYIKPA KACOUTCHY JEAN : Enseignant -
Chercheur

Table des matières



I - 1- Les Structures Conditionnelles	3
II - Application 1 :	9
III - 2- Les Structures Itératives	14
IV - Application 2 :	17
V - Travaux Pratique	18

1- Les Structures Conditionnelles

I

1.1- La Structure Conditionnelle Simple

Syntaxe :

If(condition)

{

Bloc d'instructions ;

}

Exemple :

Écrire un programme qui vérifie si un nombre entré est égal à 2.

```
1 import java.util.Scanner; // classe de l'objet Scanner
2
3 public class exemple {
4
5     public static void main(String[] args) {
6         int val1; //Déclaration de la variable val1
7         System.out.println("Entrez un nombre entier svp!"); //affiche le message
8         //entre les doubles griffes
9         Scanner lire = new Scanner(System.in);
10        val1 = lire.nextInt(); //récupération de la valeur saisie par l'utilisateur
11        if(val1==2) //vérification si la valeur de la variable val1 est égale 2
12        {
13            System.out.println("la valeur saisie est correcte"); //exécutée si condition
14            //est vraie
15        }
16    }
17 }
```

Remarque

la condition doit être entre des parenthèses il est possible de définir plusieurs conditions à remplir avec les opérateurs ET et OU (&& et ||)

par exemple l'instruction suivante teste si les deux conditions sont vraies :

if ((condition1)&&(condition2))

{

Bloc d'instructions ;

}

L'instruction suivante exécutera les instructions si l'une ou l'autre des deux conditions est vraie

if ((condition1)|| (condition2))

{

Bloc d'instructions ;

}

1.2- La Structure Conditionnelle Alternative

Syntaxe :

If(condition)

{

Bloc d'instructions1 ;

}

else

{

Bloc d'instructions2;

}

Exemple :

Écrire un programme qui vérifie si un nombre entré est égal à 2 ou pas.

```
1 import java.util.Scanner; // classe de l'objet Scanner
2
3 class exemple {
4
5     public static void main(String[] args) {
6         int val1; //Déclaration de la variable val1
7         System.out.println("Entrez un nombre entier svp!"); //affiche le message
           entre les doubles griffes
8         Scanner lire = new Scanner(System.in);
9         val1 = lire.nextInt(); //récupération de la valeur saisie par l'utilisateur
10        if(val1==2) //vérification si la valeur de la variable val1 est égale 2
11        {
12            System.out.println("la valeur saisie est correcte"); //exécutée si condition
           est vraie
13        }
14        else //vérification si la valeur de la variable val1 est différente de 2
15        {
16            System.out.println("la saisie est incorrecte"); //exécutée dans le cas
           contraire (a valeur de la variable val1 est différente de 2) ;
17        }
18    }
19
20 }
21
```

1.3- La Structure Ternaire

Syntaxe :

$(Condition) ? action_si_vrai : action_si_faux;$

Action_si_vrai correspond à l'instruction à exécuter si la condition posée est vérifiée

Action_si_faux est l'instruction à réaliser dans le cas contraire

Exemple :

Écrire un programme qui affiche la valeur absolue d'un nombre.

```
1 import java.util.Scanner; // classe de l'objet Scanner
2
3 class exemple {
4
5     public static void main(String[] args) {
6         int val1, val2; //Déclaration de variables val1 et val2
7         Scanner lire = new Scanner(System.in);
8         System.out.println("Entrez un nombre entier svp!"); // affiche le message
9         // entre les doubles griffes
10        val1 = lire.nextInt(); //récupération de la valeur saisie par l'utilisateur
11        val2=val1; //affectation de la valeur de val1 à val2
12        val2=(val2>=0) ? val2 : -1 *val2; /* si la valeur de val2 est supérieure ou
13        égale à 0 alors val2 reçoit la valeur de val2 sinon val2 reçoit ( -1*val2) */
14        System.out.println("la valeur absolue de "+val1+" est "+val2); // affichera
15        // la valeur absolue de la valeur de val1 est la valeur de val2.
16    }
17 }
```

1.3- Les Structures Conditionnelles Imbriquées

Il n'existe pas de syntaxe figée pour les structures conditionnelles imbriquées, cependant nous pouvons présenter quelques-unes.

If(condition)

{

Bloc d'instructions1 ;

}

else if(condition2)

{

Bloc d'instructions2;

}

else

{

Bloc d'instructions n;

}

Autre syntaxe :

```

If(condition)
{
    If(condition)
    {
        Bloc d'instructions;
    }
    else
    {
        Bloc d'instructions;
    }
}

```

Exemple :

Écrire un programme qui permet de dire si un nombre est pair ou impair. Si le nombre est pair il faut aussi spécifier s'il est positif ou pas.

```

1 import java.util.Scanner; // classe de l'objet Scanner
2
3 class exemple {
4
5     public static void main(String[] args) {
6
7         int nb; // déclaration de la variable nb
8         Scanner lire = new Scanner(System.in);
9         System.out.println("saisissez un nombre svp!"); // affiche le message entre
10        les griffes
11        nb = lire.nextInt(); // récupère la valeur saisie dans la variable nb
12        if (nb % 2 == 0) // vérifie si le reste de la division de nb par 2 égal à 0 (%
13        = modulo).
14        {
15            // la condition est vérifiée (nb % 2 == 0)
16            if (nb >= 0) // vérifie encore si la valeur de nb est supérieure ou égale à 0
17            {
18                System.out.println(nb + " est un nombre pair positif"); // affiche la
19                valeur de la variable nb est un nombre pair positif.
20            }
21            else
22            {
23                System.out.println(nb + " est un nombre pair négatif"); // affiche la
24                valeur de la variable nb est un nombre pair négatif.
25            }
26            else
27            {
28                // la condition n'est pas vérifiée (nb % 2 <> 0)
29                System.out.println(nb + " est un nombre impair"); // affiche la valeur de la
30                variable nb est un nombre impair.
31            }
32        }
33    }
34}

```

```

30
31 }
32
33

```

1.4- La Structure de choix

Syntaxe :

Switch(variable)

```

{
case valeur1 : instruction1;
break;
case valeur2 : instruction 2;
break;
.
.
.
default: instruction_par_défaut;
break;
}

```

Exemple :

Écrire un programme qui dans un premier temps un menu de saisie pour afficher la salutation dans l'une des langues de choix.

- Il demande par la suite que l'utilisateur fasse son choix.
- Selon que la valeur saisie par ce dernier vaille 1, 2, 3 ou 4, une salutation est retournée.
- Si aucune de ces 4 valeurs n'est saisie, un message par défaut est retourné.

```

1 import java.util.Scanner; // classe de l'objet Scanner
2
3 class exemple {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         int choix; // déclaration de la variable choix
8         Scanner lire = new Scanner(System.in);
9         System.out.println("***** MENU SALUTATION *****"); // affichera ***** MENU
10        SALUTATION ***** et ira deux fois à la ligne
11        System.out.println("1 - FRANCAIS");
12        System.out.println("2 - ANGLAIS");
13        System.out.println("3 - ESPAGNOL");
14        System.out.println("4 - ALLEMAND");
15        /* affichera
16        1 - FRANÇAIS
17        2 - ANGLAIS
18        3 - ESPAGNOL

```

```
18      4 - ALLEMAND */
19
20      System.out.println("Faites votre choix svp!"); // affichera Faites votre
choix svp! Et ira à la ligne
21      choix = lire.nextInt(); //récupère la valeur saisie dans la variable choix
22      switch(choix) // selon la valeur de choix il affichera un message de
salutation si la valeur est comprise en 1 et 4 sinon il affichera La valeur saisie
est incorrecte
23      {
24          case 1: System.out.println("SALUT!");
25          break;
26          case 2: System.out.println("HELLO!");
27          break;
28          case 3: System.out.println("HOLA!");
29          break;
30          case 4: System.out.println("TAG!");
31          break;
32          default: System.out.println("La valeur saisie est incorrecte");
33      }
34
35  }
36 }
```


Application 1 :

II

Exercice

Énoncé 1 :

Écrire un programme permettant de donner le jour à partir d'un nombre saisi par l'utilisateur.

Solution :

```

#include <iostream>
using namespace std;

class exemple {
public:
    void main() {
        int n;
        cout << "Entrer un nombre compris de 1 à 7.";
        n = getint();
        switch (n)
        {
            case 1:
                cout << "Lundi";
                break;
            case 2:
                cout << "Mardi";
                break;
            case 3:
                cout << "Mercredi";
                break;
            case 4:
                cout << "Jeudi";
                break;
            case 5:
                cout << "Vendredi";
                break;
            case 6:
                cout << "Samedi";
                break;
            case 7:
                cout << "Dimanche";
                break;
        }
    }
}

```

```

    [ ]
    [ ] "Erreur la valeur saisie est incorrecte");
    [ ]
}
}
}
```

Exercice

Énoncé 2 :

Écrire un programme permettant de donner le grade à laquelle appartient une valeur donnée. La grille se présente comme suit :

de 0 à 40 : le grade est F ;

de 41 à 60 : le grade est C ;

de 61 à 70 : le grade est B ;

de 71 à 80 : le grade est A ;

plus de 81 : le grade est A+ ;

Solution :

```

int main() ;

class exemple {
public
    void main(int lr, int pts) {
        lr = (pts / 10);
        pts = lr * 10;
        if (pts > 81)
        {
            cout << "Grade = A+ ";
        }
        else if (pts > 71)
        {
            cout << "Grade = A ";
        }
        else if (pts > 61)
        {
            cout << "Grade = B ";
        }
        else if (pts > 41)
        {
            cout << "Grade = C ";
        }
        else if (pts > 0)
        {
            cout << "Grade = F ";
        }
        else
        {
            cout << "Grade = F- ";
        }
    }
}

```

```
    "Grade = C ");  
}  
  
{  
    "Grade = F ");  
}  
}  
}
```

2- Les Structures Itératives



2.1- La Structure while

Syntaxe :

Initialisation

while(condition)

{

Bloc d'instructions ;

variation ;

}

Exemple :

Écrire un programme qui affiche les 10 premiers nombres.

```
1
2 class exemple {
3
4     public static void main(String[] args) {
5
6         int val1; //Déclaration de la variable val1
7         val1=1; //initialisation de la variable ( Initialisation )
8         while (val1<=10) //vérification de la condition
9         {
10            System.out.println(val1); //affichera de 1 à 10
11            val1++; //incrémentation de la variable ( la variation )
12        }
13
14    }
15 }
16
```

2.2- La Structure do .. while

Syntaxe :

do

{

Bloc d'instructions1 ;

```
}
```

```
while( condition );
```

Exemple :

Écrire un programme qui affiche les 10 premiers nombres.

```
1 class exemple {
2
3     public static void main(String[] args) {
4
5         int val1; //Déclaration de la variable val1
6         val1=1; ///initialisation de la variable ( Initialisation )
7         do
8         {
9             System.out.println(val1); //affichera de 1 à 10
10            val1++; //incrémentatation de la variable
11        }
12        while(val1<=10); //vérification de la condition
13
14
15    }
16 }
17
```

2.3- La Structure for

Syntaxe :

```
for(initialisation; condition; compteur)
```

```
{
```

```
Bloc d'instructions;
```

```
}
```

- *Initialisation* : ce paramètre est exploité une seule fois (juste avant la première itération) car il permet d'initialiser la variable qui sera vérifiée dans le paramètre suivant (à savoir la condition) avant l'exécution du bloc d'instructions.
- *Condition* : ce paramètre vérifie la condition d'arrêt de la boucle. Avant la première itération, cette condition est vérifiée juste après l'initialisation de la variable. Après cela, la condition est toujours vérifiée après l'exécution du compteur. Ainsi la condition d'arrêt précède toujours l'exécution du bloc d'instructions.
- *Compteur* : ce paramètre est toujours sollicité après l'exécution du bloc d'instructions et juste avant que la condition ne soit vérifiée.

Exemple :

Écrire un programme qui affiche les 10 premiers nombres.

```
1 class exemple {
2
3     public static void main(String[] args) {
4         // TODO Auto-generated method stub
5
```

```
6   int val1; //Déclaration de la variable val1
7   for(val1=1;val1<=10;val1++) //initialisation de val1, condition et
   incrémentation
8   {
9       System.out.println(val1); //affichera de 1 à 10
10  }
11
12  }
13 }
14 /**
15 Explication : La variable val1 est initialisée à 1 dans un premier temps, la
   condition est ensuite vérifiée avant que l'instruction ne soit exécutée. Juste après,
   val1 est incrémentée et la condition est à nouveau vérifiée avant l'exécution de
   l'instruction et le cycle se poursuit jusqu'à ce que val1 soit supérieur à 10 avant
   de sortir de la boucle. */
16
```


Application 2 :

IV

Exercice

Énoncé 1 :

Écrire un programme qui affiche la table de multiplication d'un nombre saisi par l'utilisateur pour des multiplicateurs de 1 à 10.

De la forme (par exemple) si on saisit 7 alors :

$$1 \times 7 = 7$$

$$2 \times 7 = 14$$

:

$$10 \times 7 = 70$$

NB : Le nombre doit être obligatoirement positif.

Solution :

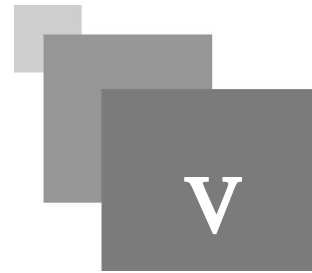
```

#include <iostream>
using namespace std;

class exemple {
public:
    void main() {
        int i;
        int lr = 0;
        do
        {
            cout << "Enter le nombre";
            nbre = lr;
        }
        while (nbre <= 0);
        {
            i = 1;
            while (i <= 10)
            {
                cout << i << " * " << nbre << " = " << nbre*i;
                cout << endl;
                i++;
            }
        }
    }
}

```

Travaux Pratique



Énoncé :

Écrire un programme en JAVA qui contraint l'utilisateur à saisir un nombre pair avant l'affichage du triple de ce nombre.

```
1 import java.util.Scanner; // classe de l'objet Scanner
2
3 class exemple {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7
8         int res,nb; // déclaration des variables nb et res
9         Scanner lire = new Scanner(System.in);
10        do
11        {
12            System.out.println("Saisissez un nombre pair svp!"); // affichera : Saisissez
un nombre pair svp!
13            nb = lire.nextInt(); // récupère la valeur saisie
14        }
15        while (nb%2!=0); // vérifie si le nombre est impair
16        res =nb*3;
17        System.out.println("Le triple de "+nb+" est: "+res);
18
19
20    }
21 }
22
```