



R1.01

Initiation au développement



Structure Conditionelle

Principe

SI une **condition** est VRAIE

ALORS on exécute CE bloc d'instructions (séquence/traitement n°1)

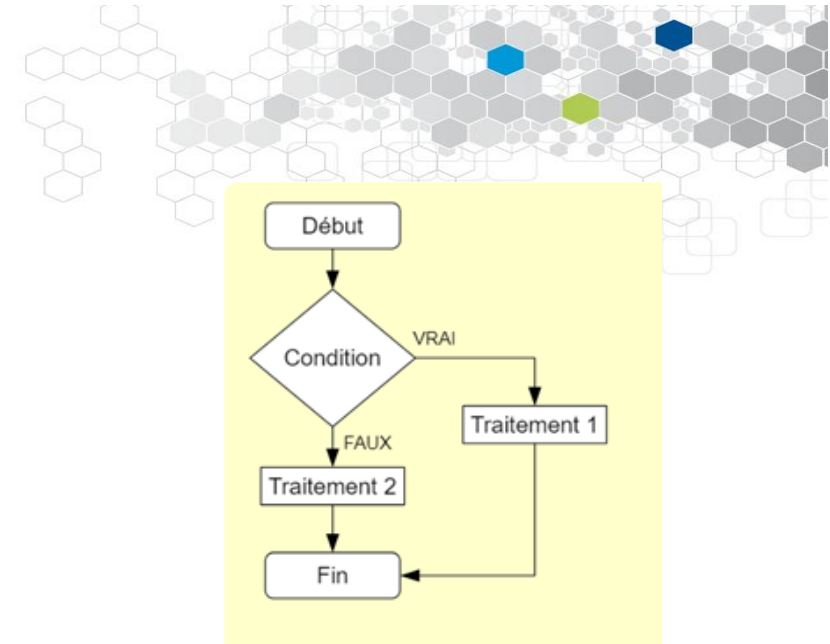
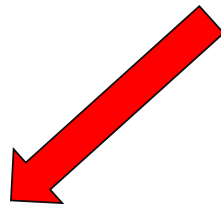
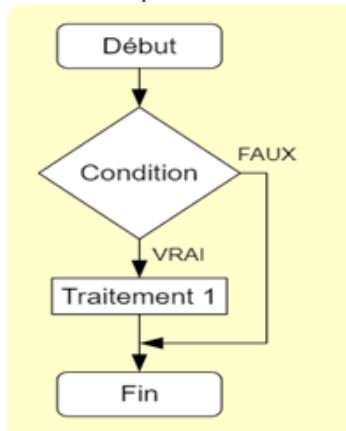
SINON

On exécute CE bloc d'instructions (séquence/traitement n°2)

FIN_SI

La **condition** est aussi appelée **proposition booléenne** .

•La partie "**SINON+bloc n°2**" peut être absente, selon les cas.



Structure conditionnelle: Bloc d'instructions conditionnel

Variables:

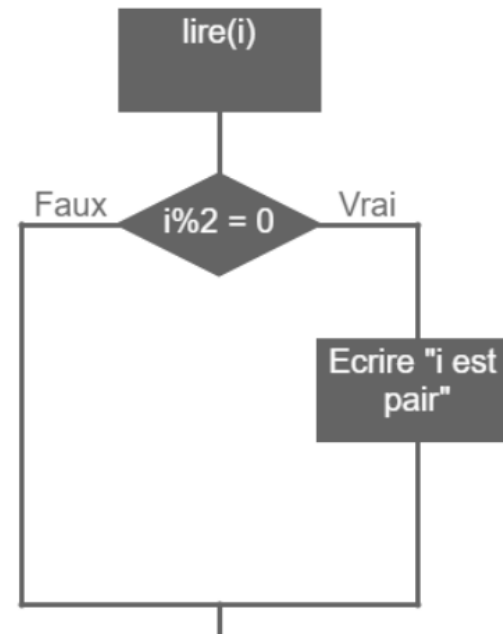
i entier

Début

Lire(i)

SI ($i \% 2 = 0$) **ALORS**
 Ecrire("i est pair")

FIN_SI

Fin

Structure conditionnelle: à deux branches



Exemple:

Variables:

i entier

Début

Lire(i)

SI $i \% 2 = 0$ **ALORS**

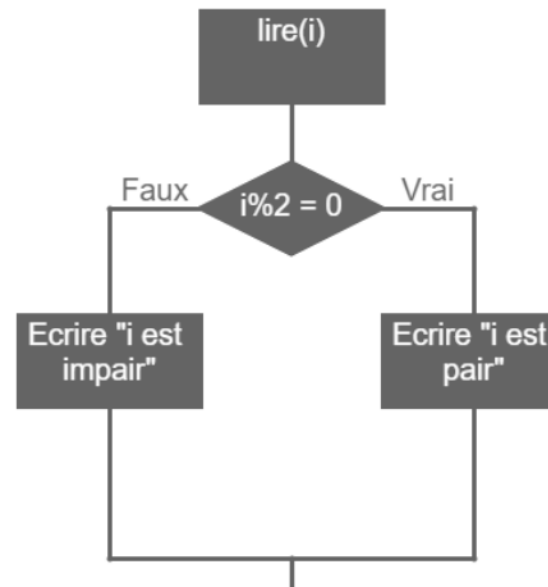
Ecrire("i est pair")

SINON

Ecrire("i est impair")

FIN_SI

Fin



Structure conditionnelle en Java



Structures conditionnelles imbriquées

```
if ( i < 256 )
    System.out.println("format 8 bits") ;
else if ( i < 65536)
    System.out.println("format 16 bits") ;
else
    System.out.println("format > 16 bits") ;
```



Les accolades delimitant les blocs d'instructions sont facultatives lorsqu'un bloc particulier ne comporte qu'une seule instruction.

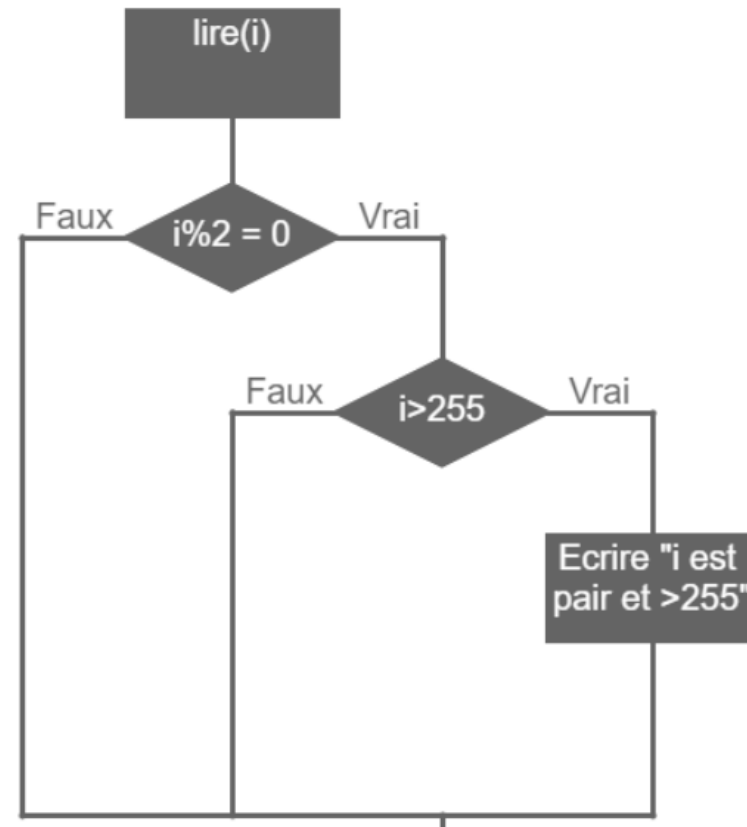
Structure conditionnelle avec ET/OU



```
Variables
    i entier
Début
    lire(i)
    SI i%2=0 ALORS
        SI i > 255 ALORS
            écrire("i est pair et >255")
        FIN_SI
    FIN_SI
Fin
```

Équivaut à

```
Variables
    i entier
Début
    lire(i)
    SI i%2=0 ET i > 255 ALORS
        écrire("i est pair et >255")
    FIN_SI
Fin
```



Structure conditionnelle: *n* branches



Exemple d'une structure imbriquée : les jours de la semaine.

Variables

i entier

Début

lire(*i*)

SI *i*==1 ALORS

 écrire("lundi")

SINON SI *i*==2 ALORS

 écrire("mardi")

SINON SI *i*==3 ALORS

 écrire("mercredi")

SINON SI *i*==4 ALORS

 écrire("jeudi")

SINON SI *i*==5 ALORS

 écrire("vendredi")

SINON SI *i*==6 ALORS

 écrire("samedi")

SINON SI *i*==7 ALORS

 écrire("dimanche")

SINON

 écrire("numéro invalide")

FIN_SI

Fin

Alternative à n branches: Selon

Exemple d'une structure imbriquée : les jours de la semaine.

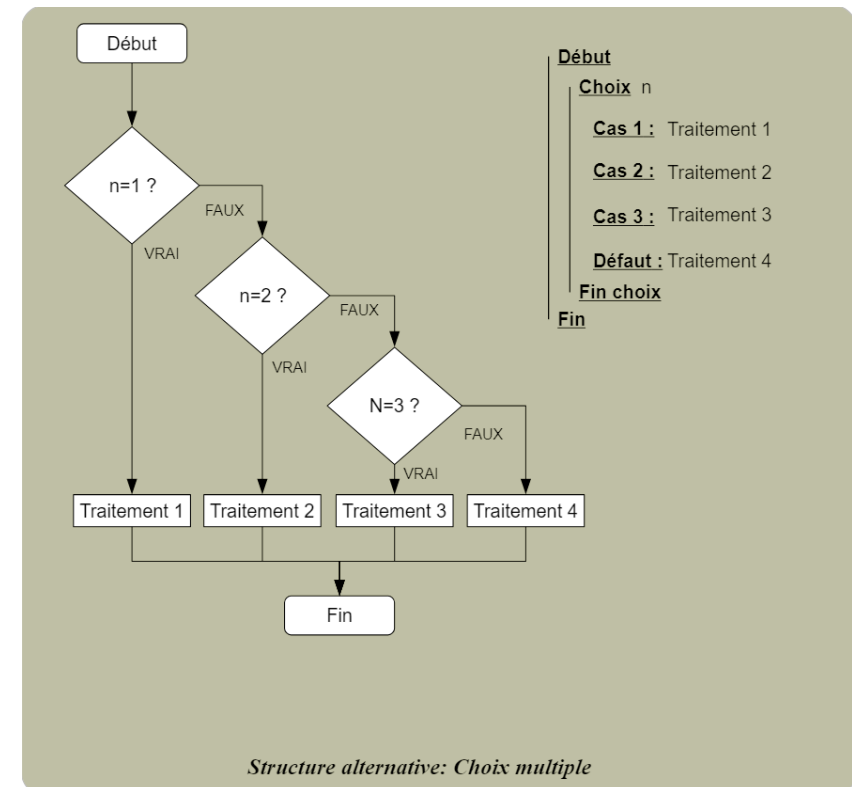
Dans le cas ou l'on compare i à des valeurs entières prédéterminées

```
Variables
i entier

Début
lire(i)
SELON i
  Cas 1 : écrire("lundi")
  Cas 2 : écrire("mardi")
  Cas 3 : écrire("mercredi")
  Cas 4 : écrire("jeudi")
  Cas 5 : écrire("vendredi")
  Cas 6 : écrire("samedi")
  Cas 7 : écrire("dimanche")
  défaut: écrire("numéro invalide")
FIN_SELON
Fin
```



- Le cas *défaut* est optionnel.
- La structure *SELON* ne s'applique pas lorsque l'on cherche à distinguer des intervalles de valeurs.



Selon en Java



Alternative à n branches

```
switch(jour){  
    case 1: System.out.println("lundi");    break ;  
    case 2: System.out.println("mardi");    break ;  
    case 3: System.out.println("mercredi"); break ;  
    case 4: System.out.println("jeudi");    break ;  
    case 5: System.out.println("vendredi"); break ;  
    case 6: System.out.println("samedi");   break ;  
    case 7: System.out.println("dimanche"); break ;  
    default: System.out.println("numéro erroné");  
}
```



- Le cas *default* est optionnel.
- Le *break* en fin de bloc est important ; il provoque la sortie de la structure et évite ainsi l'exécution du bloc suivant.
- Sans *break*, ce n'est plus véritablement une structure d'alternatives à n branches ; préférer des conditionnelles imbriquées.

Examples :

If and if-else statements.

<i>absolute value</i>	<code>if (x < 0) x = -x;</code>
<i>put the smaller value in x and the larger value in y</i>	<code>if (x > y) { int t = x; x = y; y = t; }</code>
<i>maximum of x and y</i>	<code>if (x > y) max = x; else max = y;</code>
<i>error check for division operation</i>	<code>if (den == 0) System.out.println("Division by zero"); else System.out.println("Quotient = " + num/den);</code>
<i>error check for quadratic formula</i>	<code>double discriminant = b*b - 4.0*c; if (discriminant < 0.0) { System.out.println("No real roots"); } else { System.out.println((-b + Math.sqrt(discriminant))/2.0); System.out.println((-b - Math.sqrt(discriminant))/2.0); }</code>



Nested if-else statement.

```
if      (income <      0) rate = 0.00;  
else if (income <  8925) rate = 0.10;  
else if (income < 36250) rate = 0.15;  
else if (income < 87850) rate = 0.23;  
else if (income <183250) rate = 0.28;  
else if (income <398350) rate = 0.33;  
else if (income <400000) rate = 0.35;  
else                                     rate = 0.396;
```

Question 1



```
Programme afficheur
Variables
    a, b, c : entier
Début
Lire (a, b, c)
SI a<b ALORS
/ SI b<c ALORS
/ / Écrire (Cas « 1 »)
/ SINON Écrire (Cas « 2 »)
/ FIN_SI
SINON
/ SI b=c ALORS
/ / Écrire (Cas « 3 »)
/ SINON Écrire (Cas « 4 »)
/ FIN_SI
FIN_SI
Fin
```

Pour a=8, b=8 et c=9 , l'agorithme affichera

- A. CAS « 1 »
- B. CAS « 2 »
- C. CAS « 3 »
- D. CAS « 4 »

Exercice n° 1: Dérouler un algorithme

Réalisez l'exécution de l'algorithme suivant en donnant successivement les valeurs 20, 150, 50, 51 et 400 à *Valeur*?

Algorithme déroulerAlgorithme

Variables:

valeur Entier

Début

Lire(*valeur*)

SI *valeur* > 100 ALORS

 Ecrire ("A")

 SI *valeur* > 200 ALORS

 Ecrire ("B")

 SINON

 Ecrire ("C")

 FIN_SI

SINON

 SI *valeur* > 50 ALORS

 Ecrire ("D")

 SINON

 Ecrire ("E")

 FIN_SI

FIN_SI

Fin

Correction



Réalisez l'exécution de l'algorithme suivant en donnant successivement les valeurs 20, 150, 50, 51 et 400 à *Valeur*?

Algorithme déroulé

Variables:

valeur Entier

Debut

SI *valeur* > 100 ALORS

 Ecrire ("A")

 SI *valeur* > 200 ALORS

 Ecrire ("B")

 SINON

 Ecrire ("C")

 FIN_SI

SINON

 SI *valeur* > 50 ALORS

 Ecrire ("D")

 SINON

 Ecrire ("E")

 FIN_SI

FIN_SI

Fin

Donnée	Sorties de l'algorithme (impression)
20	E
150	A C
50	E
51	D
400	A B

Exercice n° 2: Pizzas

- a) Concevoir un algorithme qui lit le diamètre de deux pizzas, leur numéro et leur prix puis qui imprime le numéro de celle qui a le meilleur rapport *taille/prix*.
- b) Traduire l'algorithme dans un programme Java complet.



Correction: Algorithme taillePrix



Algorithme taillePrix

Constantes:

PI réel = 3,141 592 653 59 constante mathématique

Variables:

<i>numéro1</i>	Entier	numéro de la première pizza
<i>numéro2</i>	Entier	numéro de la deuxième pizza
<i>diamètre1</i>	Réel	diamètre de la première pizza
<i>diamètre2</i>	Réel	diamètre de la deuxième pizza
<i>prix1</i>	Réel	prix de la première pizza
<i>prix2</i>	Réel	prix de la deuxième pizza
<i>rapport1</i>	Réel	Rapport taille/prix de la première pizza
<i>rapport2</i>	Réel	Rapport taille/prix de la deuxième pizza

Debut

Ecrire ("Entrer numéro, diamètre et prix de la première pizza")

Lire(*numéro1*, *diamètre1*, *prix1*)

Ecrire"Entrer numéro, diamètre et prix de la deuxième pizza"

Lire (*numéro2*, *diamètre2*, *prix2*)

$rapport1 := diamètre1 \times diamètre1 \times PI / (4 \times prix1)$

$rapport2 := diamètre2 \times diamètre2 \times PI / (4 \times prix2)$

SI $rapport1 > rapport2$ **ALORS**

Ecrire (" C'est la pizza" + *numéro1* + "qui a le meilleur rapport taille/prix.")

SINON SI $rapport1 < rapport2$ **ALORS**

Ecrire (" C'est la pizza", *numéro2*, "qui a le meilleur rapport taille/prix.")


SINON

Ecrire (" Les deux pizzas on le même rapport taille/prix.")

FIN_SI

Fin

taillePrix.java



```
import java.util.*;
class taillePrix{
    static Scanner input = new Scanner(System.in);
    public static void main(String args[]){
        int numero1 ;           // numero de la premiere pizza
        int numero2 ;           //numero de la deuxieme pizza
        double diametre1;       //diametre de la premiere pizza
        double diametre2 ;      //diametre de la deuxieme pizza
        double prix1;           //prix de la premiere pizza
        double prix2 ;          //prix de la deuxieme pizza
        double rapport1;        //Rapport taille/prix de la premiere pizza
        double rapport2 ;       //Rapport taille/prix de la deuxieme pizza

        System.out.println("Entrer numero, diametre et prix de la premiere pizza:");
        numero1=input.nextInt();
        diametre1=input.nextDouble();
        prix1= input.nextDouble();
        System.out.println("Entrer numero, diametre et prix de la deuxieme pizza:");
        numero2=input.nextInt();
        diametre2=input.nextDouble();
        prix2=input.nextDouble();

        rapport1= (diametre1 * diametre1 * Math.PI)/(4 * prix1);
        rapport2=(diametre2 * diametre2 * Math.PI)/(4 * prix2);

        if(rapport1 > rapport2)
            System.out.println("C'est la pizza numero "+ numero1+" qui a le meilleur rapport taille/prix.");
        else if (rapport1 < rapport2)
            System.out.println("C'est la pizza numero "+ numero2+ " qui a le meilleur rapport taille/prix.");
        else
            System.out.println("Les deux pizzas on le même rapport taille/prix.");
    }
}
```

Exercice n°3: Mois de l'année

Mois de l'année

Écrire un algorithme qui demande un numéro de mois à l'utilisateur et indique en retour son nom et le nombre de jours dans ce mois.

<u>Mois</u>	<u>Numéro</u>	<u>Nombre jours</u>
Janvier	1	31
Février	2	28
Mars	3	31
Avril	4	30
Mai	5	31
Juin	6	30
Juillet	7	31
Août	8	31
Septembre	9	30
Octobre	10	31
Novembre	11	30
Décembre	12	31

Correction



Algorithme moisJours

Variables:

mois Entier numéro de mois

Debut

Ecrire ("Entrer un numéro de mois: ")

Lire (*mois*)

SELON *mois*

- Cas 1** : Ecrire("Janvier, 31 jours")
- Cas 2** : Ecrire("Février, 28 ou 29 jours")
- Cas 3** : Ecrire("Mars, 31 jours")
- Cas 4** : Ecrire("Avril, 30 jours")
- Cas 5** : Ecrire("Mai, 31 jours")
- Cas 6** : Ecrire("Juin, 30 jours")
- Cas 7** : Ecrire("Juillet, 31 jours")
- Cas 8** : Ecrire("Août, 31 jours ")
- Cas 9** : Ecrire("Septembre, 30 jours")
- Cas 10** : Ecrire("Octobre, 31 jours")
- Cas 11** : Ecrire("Novembre, 30 jours")
- Cas 12** : Ecrire("Décembre, 31 jours")
- défaut** : Ecrire("Numéro invalide")

FIN_SELON

Fin

<u>Mois</u>	<u>Numéro</u>	<u>Nombre</u> <u>jours</u>
Janvier	1	31
Février	2	28
Mars	3	31
Avril	4	30
Mai	5	31
Juin	6	30
Juillet	7	31
Août	8	31
Septembre	9	30
Octobre	10	31
Novembre	11	30
Décembre	12	31

moisJours.java



```
import java.util.*;

class moisJours{
    static Scanner input = new Scanner(System.in);

    public static void main(String args[]){
        int mois; //          numéro de mois
        System.out.println ("Entrer un numéro de mois");
        mois=input.nextInt();

        switch(mois){
            case 1 : System.out.println("Janvier, 31 jours");break;
            case 2 : System.out.println("Février, 28 ou 29 jours");break;
            case 3 : System.out.println("Mars, 31 jours");break;
            case 4 : System.out.println("Avril, 30 jours");break;
            case 5 : System.out.println("Mai, 31 jours");break;
            case 6 : System.out.println("Juin, 30 jours");break;
            case 7 : System.out.println("Juillet, 31 jours");break;
            case 8 : System.out.println("Août, 31 jours ");break;
            case 9 : System.out.println("Septembre, 30 jours");break;
            case 10 : System.out.println("Octobre, 31 jours");break;
            case 11 : System.out.println("Novembre, 30 jours");break;
            case 12 : System.out.println("Décembre, 31 jours");break;
            default : System.out.println("Numéro invalide");
        }
    }
}
```

Exercice n° 4: Années bissextiles



- Écrire un programme Java complet qui demande une année à l'utilisateur, et qui affiche si l'année entrée est bissextile ou non.
 - Une année est bissextile si elle est divisible par 4, mais pas par 100.
 - Les années divisibles par 400 sont cependant bissextiles.

Correction



```
import java.util.Scanner;
class IsBissextiles {

    private static Scanner scanner = new Scanner(System.in);

    public static void main(String args[]) {
        System.out.print("Entrez une année? ");
        int annee = scanner.nextInt();

        if((annee%4==0) &&(annee%100!=0) || (annee%400==0))
            System.out.println("L'année " + annee + " est bissextile.");
        else
            System.out.println("L'année " + annee + " n'est pas bissextile.");
    }
}
```