

UD2 . Introducció a JAVA III. Estructures alternatives

Verónica Mascarós

Curso 23-24



```

e = m(b, " ");
-1 < e && b.splice(e, 1);
e = m(b, void 0);
-1 < e && b.splice(e, 1);
e = m(b, "");
-1 < e && b.splice(e, 1);
for (c = 0; c < d && c < b.length; c++) {
    a += b[c].b + ", ", n.push(b[c].b)
}
for (g = 0; g < f; g++) {
    e = Math.floor(b.length * Math.random());
    d.c + "</span></li>"), b[e] = b[e] + " "
}
for (; c < b.length; c++) {
    void 0 !== b[c] && ("parameter " + c + " is not defined")
}
function(b);
...ingle").h("mode_9...ent("

```

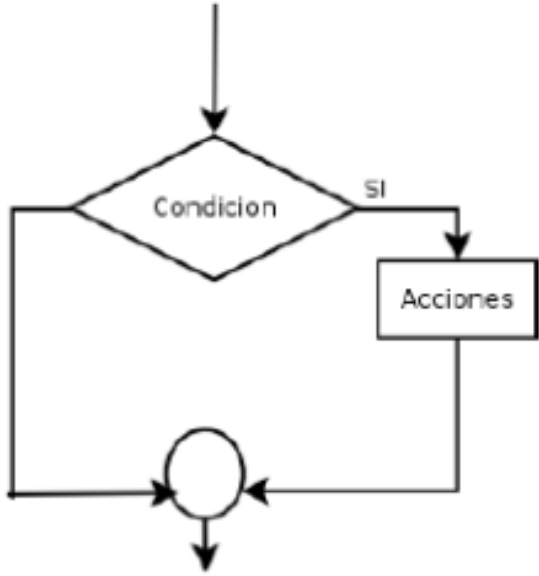
Estructuras alternativas (conditionals)

- Les estructures alternatives són construccions que permeten alterar el flux seqüencial d'un programa de manera que, en funció d'una condició o el valor d'una expressió, el mateix puga ser desviat en l'una o l'altra alternativa de codi.
- Les estructures alternatives disponibles a Java són:
 - Alternativa Simple (if)
 - Alternativa Doble (if-else)
 - Alternativa Múltiple (switch)

Alternativa simple: IF

- La sentència **if** permet l'execució d'una sèrie d'instruccions en funció del resultat d'una expressió lògica.
- El resultat d'avaluar una expressió lògica és o vertader (true) o fals (false).
- És molt simple, en llenguatge natural seria: "si aquesta condició és vertadera llavors realitza les següents accions"

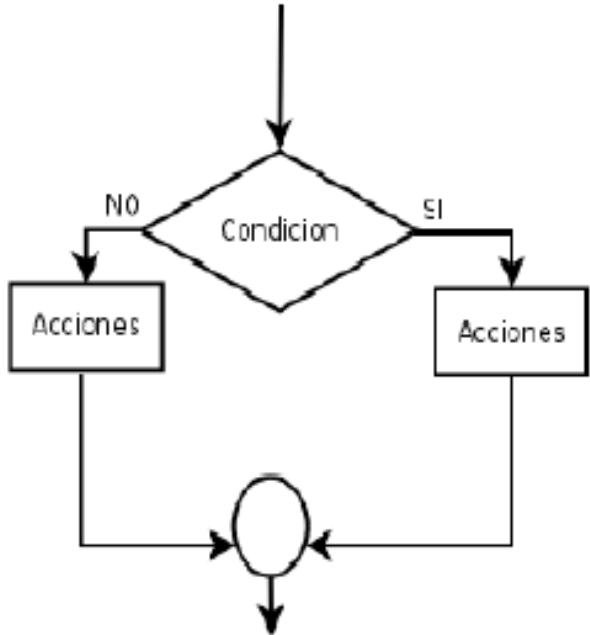
Alternativa simple: IF

CODI	ORDINOGRAMA
<pre>if (condició) { // Accions; }</pre> <p>El bloc Accions s'executa si la condició (expressió lògica) s'avalua a true (és vertadera).</p> <pre>int cont=0 ; if (cont == 0){ System.out.println("cont és 0"); // més instruccions... }</pre> <p>Si dins del if només hi ha una instrucció, no és necessari posar les claus.</p> <pre>int cont=0 ; if (cont == 0) System.out.println("cont és 0");</pre>	 <pre>graph TD; Entry(()) --> Condicion{Condicion}; Condicion -- SI --> Acciones[Acciones]; Acciones --> Merge(()); Merge --> Condicion; Merge --> Exit(())</pre>

Alternativa doble: IF-ELSE

- La sentència **if-else** permet l'execució d'una sèrie d'instruccions en funció del resultat d'una expressió lògica.
- En llenguatge natural seria: "si aquesta condició és vertadera llavors fes això, sinó fes allò d'altre"

Alternativa simple: IF-ELSE

CODI	ORDINOGRAMA
<pre>if (condició) { // AccionsSI; } else { // AccionsNO; }</pre> <p>Si dins del if o del else només hi ha una instrucció, no és necessari posar les claus.</p> <p>El bloc Accions s'executa si la condició (expressió lògica) s'avalua a true (és vertadera). En cas contrari, s'executa el bloc de AccionsNO.</p> <pre>int cont=0 ; if (cont == 0){ System.out.println("cont és 0"); // més instruccions... } else { System.out.println("cont NO és 0"); // més instruccions... }</pre>	 <pre>graph TD Entry(()) --> Condicion{Condicion} Condicion -- NO --> Acciones1[Acciones] Condicion -- SI --> Acciones2[Acciones] Acciones1 --> Merge(()) Acciones2 --> Merge Merge --> Exit(())</pre> <p>NOTA: L'operador relacional per a comprovar si són iguals és ==, no un sol = que correspon amb l'operador d'assignació. Aquest error no el detecta el compilador i és difícil d'esbrinar.</p>

Alternativa doble: IF-ELSE

- En moltes ocasions, es "nien" estructures alternatives if-else, (una dins de l'altra) de manera que es pregunte per una condició si anteriorment no s'ha complit una altra successivament.
- Per exemple : suposem que realitzem un programa que mostra la nota d'un alumne en la forma (insuficient, suficient, bé, notable o excel·lent) en funció de la seua nota numèrica.

Alternativa doble

- Podria codificar-se de la següent forma:

- La clàusula else s'utilitza en l'últim cas = sinó

```
import java.util.Scanner;

/* En este exemple suposem que l'usuari introdueix el número correctament.
 * No es realitza comprovació. */

public class NotaAlumne {

    public static void main(String args[]) {

        // Declarem variables i el construcctor d'Scanner

        Scanner entrada = new Scanner(System.in);
        int nota;

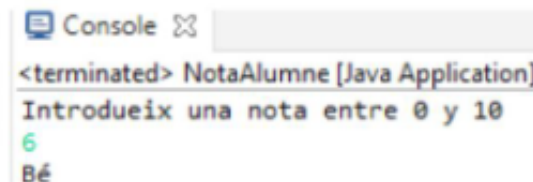
        // Llegim la nota des de teclat

        System.out.println("Introdueix una nota entre 0 y 10");
        nota = entrada.nextInt();

        if (nota < 5 ) {
            System.out.println("Insuficient");
        } else if (nota < 6) {
            System.out.println("Suficient");
        } else if (nota < 7) {
            System.out.println("Bé");
        } else if (nota < 9) {
            System.out.println("Notable");
        } else {
            System.out.println("Excel·lent");
        }

    } // Del main()

} // De la classe
```



Alternativa doble: IF-ELSE

- Si ho fem **només amb if**, podem tindre un resultat incoherent:

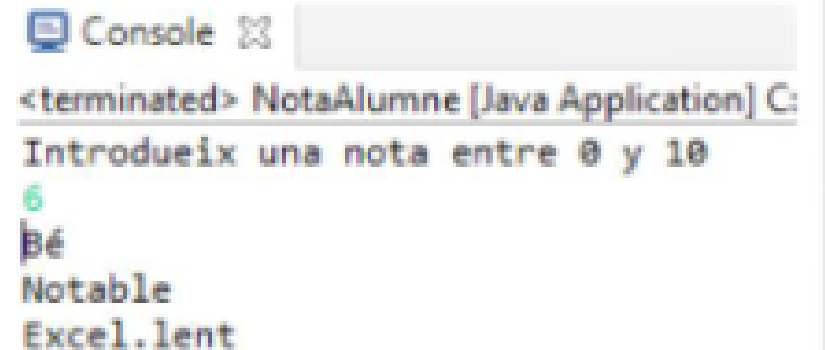
```
if (nota < 5 )
    System.out.println("Insuficient");

if (nota < 6)
    System.out.println("Suficient");

if (nota < 7)
    System.out.println("Bé");

if (nota < 9)
    System.out.println("Notable");

if (nota <=10)
    System.out.println("Excel.lent");
```



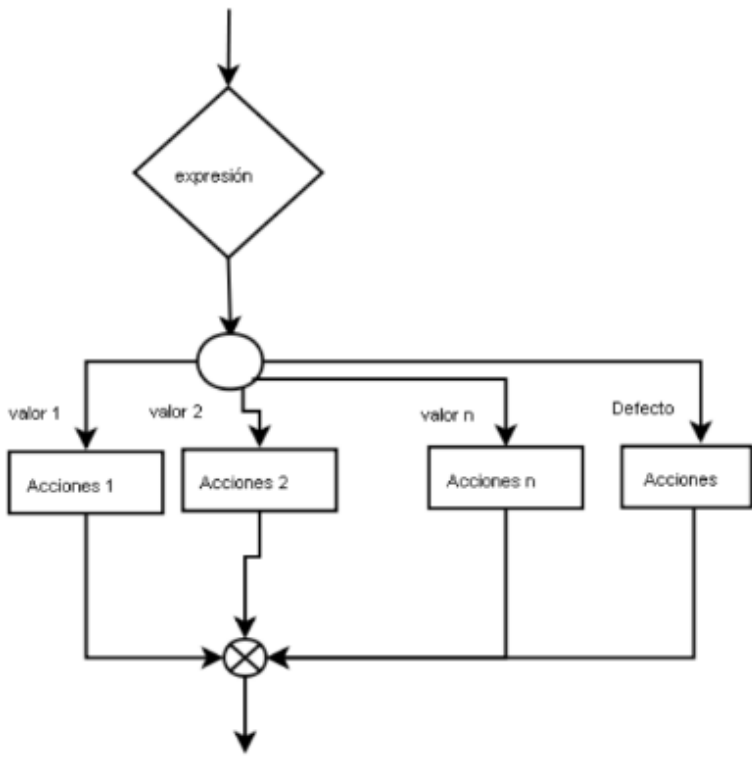
Console

<terminated> NotaAlumne [Java Application] C:
Introdueix una nota entre 0 y 10
6
Bé
Notable
Excel.lent

Alternativa múltiple: SWITCH

- A vegades és necessari comparar el valor d'una variable amb una sèrie de valors concrets.
- La selecció múltiple és molt semblant (encara que no és exactament igual) a una seqüència de diverses sentències if com les del exemple anterior.
- En llenguatge natural seria alguna cosa així com: "Si variable val valor, llavors entra per case valor1; si variable val valor2, llavors entra per case valor2,... si variable no val cap dels valors que hi ha en els diferents case llavors entra per default.

Alternativa multiple: SWITCH

CODI	ORDINOGRAMA
<pre>switch (expressió) { case valor1: // Accions1; break; case valor2: // Accions2; break; case valorN: // Accions3; break; default: // Accions per defecte; }</pre>	 <pre>graph TD; Entry(()) --> Decision{expresión}; Decision --> Merge(()); Merge -- "valor 1" --> Acc1[Acciones 1]; Merge -- "valor 2" --> Acc2[Acciones 2]; Merge -- "valor n" --> Accn[Acciones n]; Merge -- "Defecto" --> AccDef[Acciones]; Acc1 --> Join((X)); Acc2 --> Join; Accn --> Join; AccDef --> Join; Join --> Exit(())</pre> <p>The flowchart illustrates the execution of a switch statement. It begins with an entry point leading to a decision diamond labeled 'expresión'. If the expression matches one of the cases (valor 1, valor 2, valor n, or Defecto), the flow proceeds to the corresponding action box (Acciones 1, Acciones 2, Acciones n, or Acciones). All paths converge at a join node (a circle with an 'X'), which then leads to the final exit point.</p>

És molt important entendre que **en el switch s'avalua una expressió** (un valor concret com 0, 5, 1...) no una condició (true o false) com en el if i el if-else.

Alternativa múltiple: SWITCH

- El programa comprova el valor de expressió i saltarà al case que corresponga amb aquest valor (valor1 o valor2 o ...) executant el codi de dit case (Accions1 o Accions2 o ...).
- Si no coincideix cap valor, saltarà al default i executarà les Accions per defecte.
- És important afegir la sentència **break;** al final de cada case, ja que en cas contrari el programa continuarà executant el codi de les altres accions i normalment no voldrem que faci això (encara que Java permet fer-ho, és confús i per això està desaconsellat).

Alternativa múltiple

- Exemple: Programa que demana un número (i) per teclat, si el número és 0 mostra el missatge "i és zero.", si el número és 1 mostra el missatge "i és u.", si el número és 2 mostra el missatge "i és dos.", en qualsevol altre cas mostra el missatge "el número és diferent a 0,1 o 2").

```
import java.util.Scanner;

public class ExempleSwitch {

    public static void main(String args[]) {

        // Declarem la variable entera i, i el constructor d'Scanner

        int i;
        Scanner entrada = new Scanner(System.in);

        // Llegim un número per teclat i l'assignem a la variable "i"

        System.out.println("Introdueix un número");
        i = entrada.nextInt();

        switch (i)
        {
            case 0:
                System.out.println("i és zero.");
                break;

            case 1:
                System.out.println("i és u.");
                break;

            case 2:
                System.out.println("i és dos.");
                break;

            default:
                System.out.println("el número és diferent a 0,1 o 2");

        } // Del Switch

    } // Del main()
} // De la classe
```

Console ✕

<terminated> ExempleSwitch
Introdueix un número
0
i és zero.

Console ✕

<terminated> ExempleSwitch
Introdueix un número
1
i és u.

Console ✕

<terminated> ExempleSwitch
Introdueix un número
2
i és dos.

Console ✕

<terminated> ExempleSwitch [Java Appli
Introdueix un número
3
el número és diferent a 0,1 o 2

Alternativa múltiple

- Exemple: Programa que mostra un menú d'opcions per calcular l'àrea d'un quadrat, o un rectangle o un triangle segons trie l'usuari. En cada cas el programa demanarà les dades necessàries.

```
import java.util.Scanner;  
  
public class MenuOpcionsAmbSwitch {  
    public static void main(String[] args) {
```

```
        Scanner entrada = new Scanner(System.in);  
        int opcio;  
        double costat;  
        double base;  
        double altura;  
        double area;  
  
        System.out.println(" CÀLCUL DE ÀREES");  
        System.out.println(" -----");  
        System.out.println(" 1. Quadrat");  
        System.out.println(" 2. Rectàngle");  
        System.out.println(" 3. Triàngle");  
        System.out.print("\n Tria una opció (1-3): ");
```

```
        opcio= entrada.nextInt();
```

```
        switch (opcio) {  
            case 1:  
                System.out.print("\nHas triat calcular l'àrea d'un QUADRAT: ");  
                System.out.print("\nIntrodueix el costat del quadrat en cm: ");  
                costat = entrada.nextDouble();  
                area = costat*costat;  
                System.out.println("\nL'àrea del quadrat és " + area + " cm2");  
                break;  
  
            case 2:  
                System.out.print("\nHas triat calcular l'àrea d'un RECTANGLE: ");  
                System.out.print("\nIntrodueix la base del rectangle en cm: ");  
                base = entrada.nextDouble();  
                System.out.print("Introdueix l'altura del rectangle en cm: ");  
                altura = entrada.nextDouble();  
                area = base*altura;  
                System.out.println("El àrea del rectangle es " + area + " cm2");  
                break;  
  
            case 3:  
                System.out.print("\nHas triat calcular l'àrea d'un TRIANGLE: ");  
                System.out.print("\nIntrodueix la base del triangle en cm: ");  
                base = entrada.nextDouble();  
                System.out.print("Introdueix l'altura del triangle en cm: ");  
                altura = entrada.nextDouble();  
                area = (base*altura)/2;  
                System.out.println("El àrea del triangle es " + area + " cm2");  
                break;
```

```
            default:  
                System.out.print("\nOPCIÓ INCORRECTA !!!!.");
```

```
        } //Del switch  
    } //Del main()  
} //De la classe
```

```
Console  
<terminated> MenuOpcionsAmbSwitch [Java Application]  
CÀLCUL DE ÀREES  
-----  
1. Quadrat  
2. Rectàngle  
3. Triàngle  
  
Tria una opció (1-3): 1  
  
Has triat calcular l'àrea d'un QUADRAT:  
Introdueix el costat del quadrat en cm: 3  
  
L'àrea del quadrat és 9.0 cm2
```

```
Console  
<terminated> MenuOpcionsAmbSwitch [Java Application]  
CÀLCUL DE ÀREES  
-----  
1. Quadrat  
2. Rectàngle  
3. Triàngle  
  
Tria una opció (1-3): 2  
  
Has triat calcular l'àrea d'un RECTANGLE:  
Introdueix la base del rectangle en cm: 3  
Introdueix l'altura del rectangle en cm: 5  
  
L'àrea del rectangle es 15.0 cm2
```

```
Console  
<terminated> MenuOpcionsAmbSwitch [Java Appli  
CÀLCUL DE ÀREES  
-----  
1. Quadrat  
2. Rectàngle  
3. Triàngle  
  
Tria una opció (1-3): 3  
  
Has triat calcular l'àrea d'un TRIANGLE:  
Introdueix la base del triangle en cm: 3  
Introdueix l'altura del triangle en cm: 5  
  
L'àrea del triangle es 7.5 cm2
```

```
Console  
<terminated> MenuOpcionsAmbSwitch  
CÀLCUL DE ÀREES  
-----  
1. Quadrat  
2. Rectàngle  
3. Triàngle  
  
Tria una opció (1-3): 4  
  
OPCIÓ INCORRECTA !!!!.
```