

Programació

UT2.1 Estructures de selecció

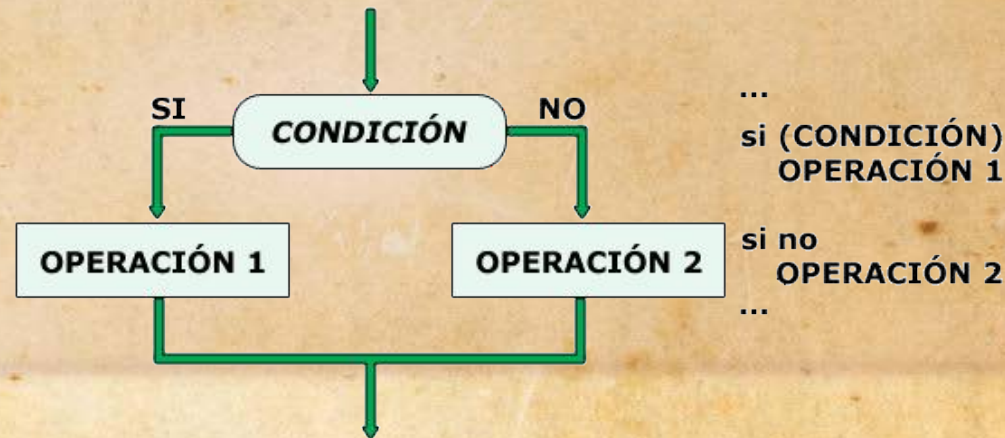
Introducció

- Primera idea errònia: Escriure directament el codi
- Es requereix un procés més general que comença per analitzar el problema i acaba en el desplegament de l'aplicació (cicle de vida)
- Cal esquematitzar els passos que s'han de seguir per a resoldre el problema (algoritme)



Introducció (II)

- Tot programa conté una seqüència de instruccions no lineal
- Cal controlar el flux de la informació
- Les estructures de control permeten fer este control



Cicle de vida

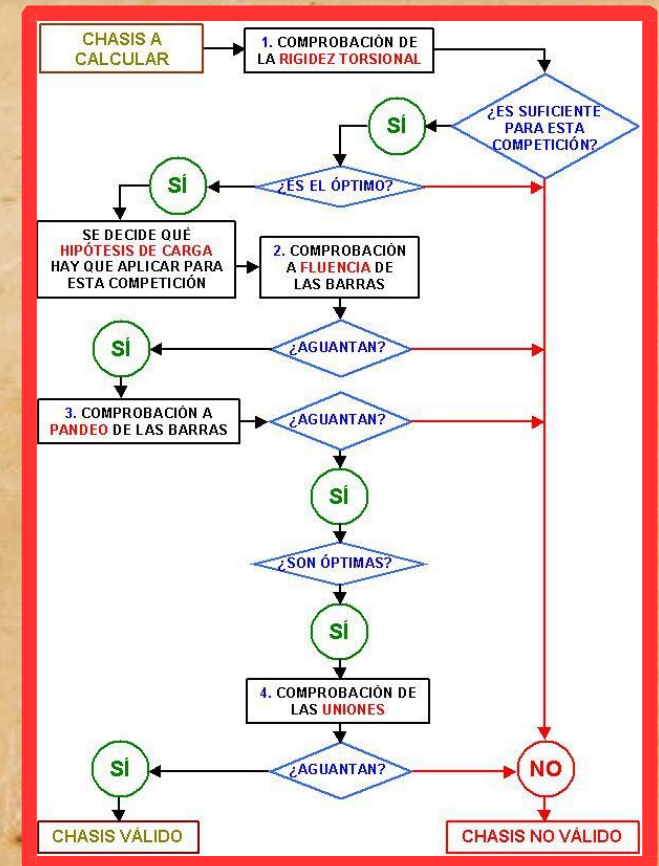
Fases que segueix el procés de creació d'una aplicació des que es proposa fins que finalitza la seua construcció

- ANÀLISI de el problema
- **DISSENY** de l'algoritme
- IMPLEMENTACIÓ del programa
- Verificació i PROVES
- Posada en marxa i MANTENIMENT



Disseny de l'algoritme

- Establir l'estratègia a seguir
- Optar per una metodologia
- Dividir el màxim possible la tasca
- Tenir en compte els recursos disponibles
- El resultat final d'esta fase és el **algoritme proposat**



Disseny de l' algoritme

- Metodologia: **programació estructurada**
- Els passos de resolució es divideixen en diferents **blocs d'instrucció** que segueixen les següents estructures.

- Lineal
- Selecció o condicional
- Repetició o iterativa

**Estableixen el
flux de control
del programa**

Exemple quotidià. processador de textos



- **Procés lineal:** obertura d'un fitxer de text (no es pot fer res fins que acabe)
- **Procés selectiu:** les diferents opcions del processador de textos
- **Procés iteratiu:** imprimir diverses còpies d'un mateix document

Representació d'algoritmes

- Pseudocodi: Pseint
- Diagrames de flux o ordinograma
- Codi font del programa

algoritmo Sumar

variables

entero a, b, c

inicio

escribir("Introduzca el primer número (entero): ")

leer(a)

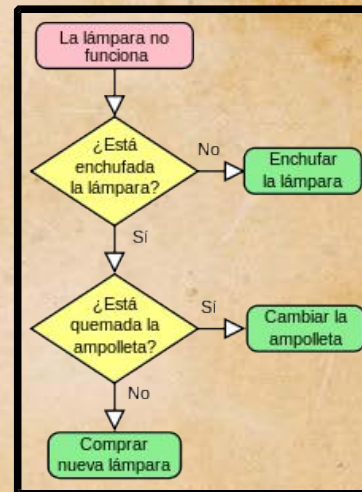
escribir("Introduzca el segundo número (entero): ")

leer(b)

$c \leftarrow a + b$

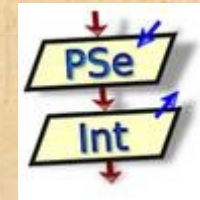
escribir("La suma es: ", c)

fin



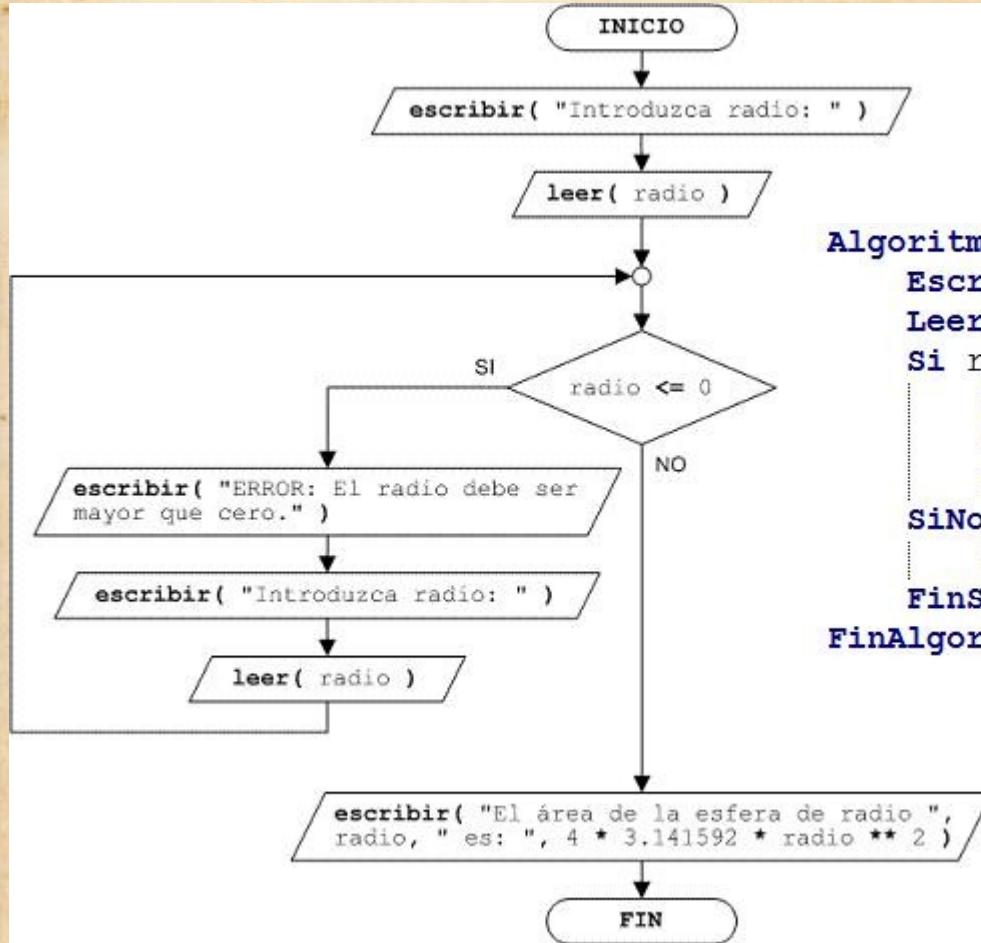
```
1 public class CalculoCuadrado1{
2     void mostrarMensaje(){
3         System.out.println("Has calculado el cuadrado de un numero");
4     }
5     int calcularCuadrado(int numero){
6         int cuadrado;
7         cuadrado=numero*numero;
8         return cuadrado;
9     }
10    public static void main(String args[]){
11        CalculoCuadrado1 cc=new CalculoCuadrado1();
12        int resultado=cc.calcularCuadrado(15);
13        System.out.println(resultado);
14        cc.mostrarMensaje();
15        System.out.println("FIN DEL PROGRAMA");
16    }
17 }
```


Descarrega i instal·la PSeInt



- ☐ Pseint és una aplicació molt interessant per a iniciar-se en el món de la programació.
- ☐ Descarrega a la teua màquina l'executable per a poder fer proves amb ell i utilitzar-lo en les teues primeres tasques de programació
- ☐ <http://pseint.sourceforge.net/index.php>

Exemple de pseudocodi i el seu ordinograma



Algoritmo AreaCirculo

Escribir 'Introduzca radio:'

Leer radio

Si radio <= 0 Entonces

Escribir 'Error: El radio debe ser mayor que cero'

Escribir 'Introduzca radio:'

Leer radio

SiNo

Escribir 'El área de la esfera de radio ', radio, ' es ', 3.1416 * radio^2

FinSi

FinAlgoritmo

És interessant que proves el codi directament en PSEint

Implementació d'un programa

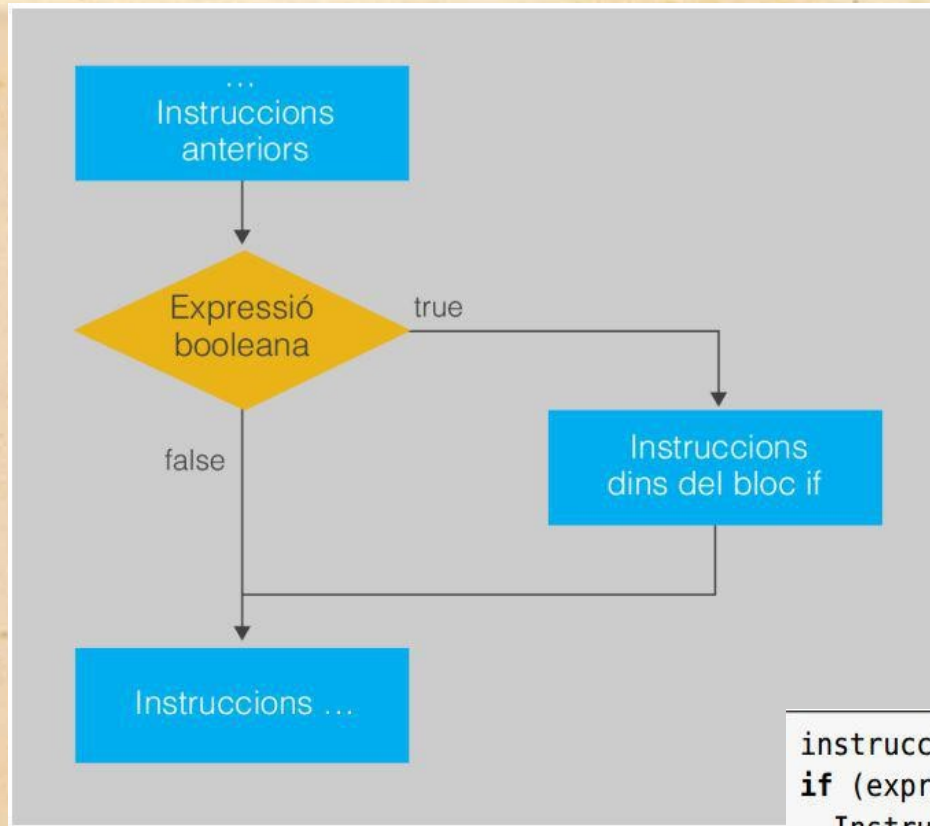
- Transformar el disseny a codi font
- Fase fàcil o complexa depenent del disseny
(Un bon disseny farà que la implementació siga un procediment fàcil)
- **Important: documentar en esta fase**



Estructures de selecció

- Permeten prendre decisions sobre quin conjunt d'instruccions s'han d'executar en un punt concret del programa
- Es basen en una expressió que retorna un resultat booleà (**condició lògica**)
- La majoria de llenguatges de programació comparteixen esta estructura

Estructura de selecció simple



Pseint

```
Algoritmo EstructuraSeleccion  
    Si expresion_logica Entonces  
        acciones_por_verdadero  
    Fin Si  
FinAlgoritmo
```

Java

```
instruccions del programa  
if (expressió booleana) {  
    Instruccions per executar si l'expressió avalua a true (cert) – Bloc if  
}  
resta d'instruccions del programa
```


A tindre en compte en Java

- La **condició lògica** ha d'estar **sempre entre parèntesi**.
Després del parèntesi, no hi ha cap ';'.
- Les instruccions a executar si la condició és certa han d'estar **sempre englobades entre dues claus** {}, excepte quan el bloc està format per una sola línia (opcional). Això forma un **bloc d'instruccions**.
- És recomanable que les **instruccions d'un bloc** estiguen **tabulades**.

Pràctica 1

Realitza el disseny amb Pseint de el següent programa de càlcul d'un descompte i després escriu-lo en Java:

El valor mínim per a optar a un descompte és de 100 € i el descompte serà d'un 10%.

- 1) Demana que s'introduisca un preu en €
- 2) Llig el preu introduït
- 3) Comprova que el preu introduït siga més gran o igual al valor mínim per a optar al descompte
 - En cas afirmatiu, aplica el descompte al preu inicial
- 4) Mostra el preu

Solució pràctica 1

Algoritmo calculpreu

MINIM \leftarrow 100

DESCOMPTE \leftarrow 10

Escribir "Introdueix el preu: "

Leer preu

Si preu \geq MINIM **Entonces**

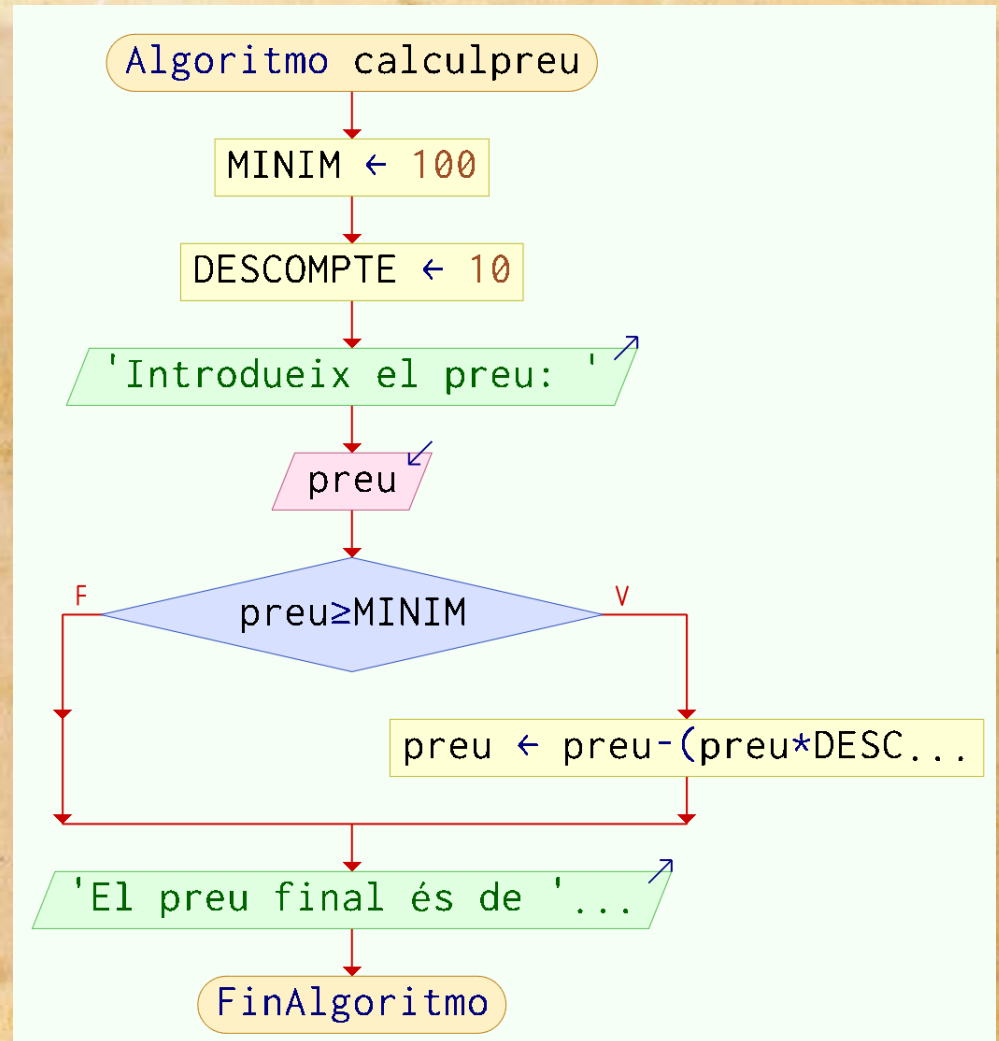
... preu \leftarrow preu - (preu * DESCOMPTE / 100)

Fin Si

Escribir "El preu final és de ", preu, " euros"

FinAlgoritmo

PSeInt



Solució pràctica 1

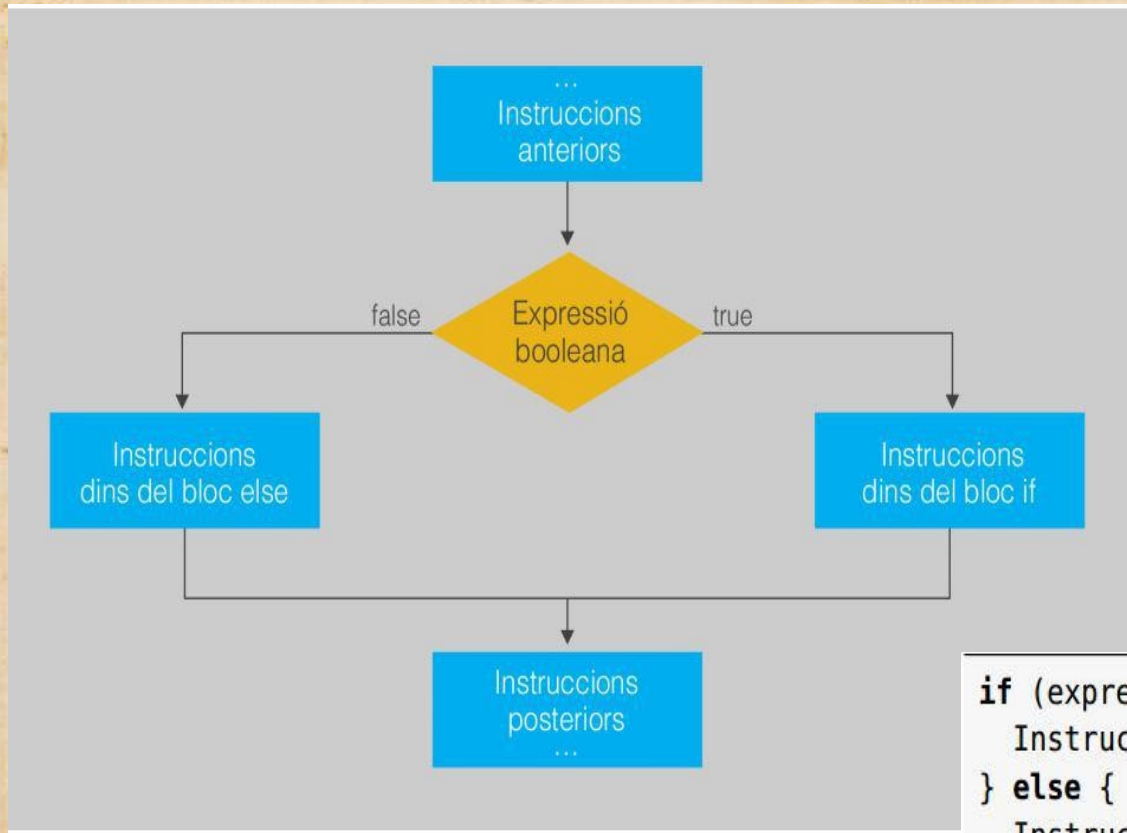
Java

```
public class CalculPreu {  
    public static void main(String[] args) {  
        Scanner teclat = new Scanner(System.in);  
        final int MINIM = 100;  
        final int DESCOMPTE = 10;  
        float preu;  
  
        System.out.print("Introdueix el preu: ");  
        preu = teclat.nextFloat();  
        if(preu >= MINIM){  
            preu -= preu * DESCOMPTE / 100;  
        }  
        // System.out.println("El preu final és de " + preu + " euros");  
        System.out.printf("El preu final és de %.2f euros\n", preu);  
  
        teclat.close();  
    }  
}
```


Pràctica 2

Modifica el programa anterior (primer en Pseint i després en Java) perquè a més d'aplicar el descompte del 10% als productes de 100€ o més, aplique també una penalització de 2€ si el preu és menor a 30€

Selecció simple amb camí alternatiu "else"



```
Algoritmo EstructuraSeleccion
  Si expresion_logica Entonces
    acciones_por_verdadero
  SiNo
    acciones_por_falso
  Fin Si
FinAlgoritmo
```

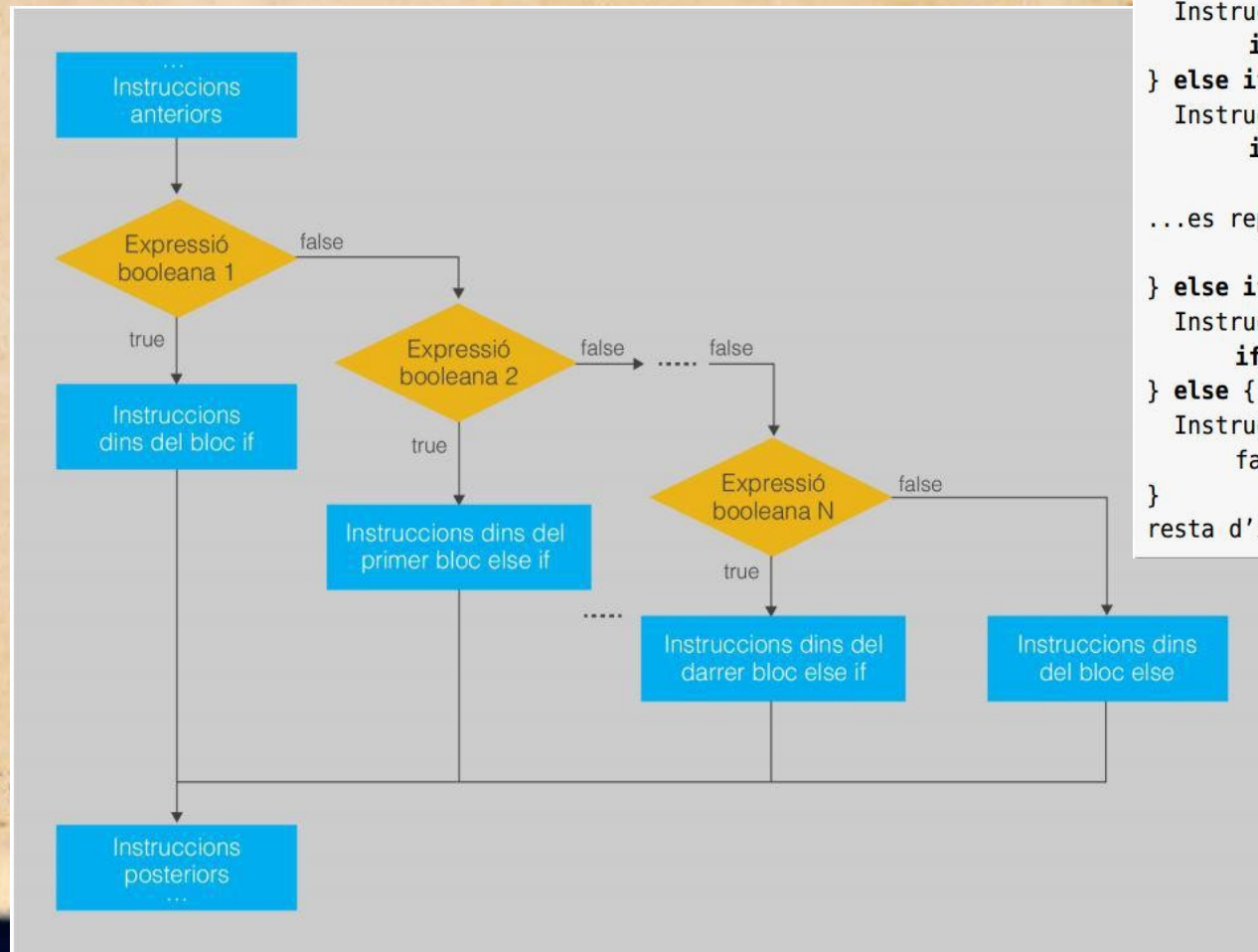
```
if (expressió booleana) {
  Instruccions per executar si l'expressió l'avalua a true (cert) – Bloc if
} else {
  Instruccions alternatives per executar si l'expressió l'avalua a false (fals)
  – Bloc else
}
```


Pràctica 3

Realitza el disseny en Pseint i posterior implementació en Java del següent programa "endevina el número"

- 1) El número secret serà el 241
- 2) Demana un número per teclat
- 3) Llig el número
- 4) Comprova si el número introduït és igual al número secret
 - Si és igual s'informa de l'encert
 - Si no és igual s'informa que s'ha fallat.

Selecció simple niada



```
instruccions del programa
if (expressió booleana 1) {
    Instruccions per executar si l'expressió 1 l'avalua a true (cert) – Bloc if
} else if (expressió booleana 2) {
    Instruccions per executar si l'expressió 2 l'avalua a true (cert) – Bloc else if
} else if (expressió booleana 3) {
    Instruccions per executar si l'expressió 3 l'avalua a true (cert) – Bloc else if
...es repeteix tant cops com calgui...
} else if (expressió booleana N) {
    Instruccions per executar si l'expressió N avalua a true (cert) – Bloc else if
} else {
    Instruccions alternatives si totes les expressions 1...N avaluen a false (fals) – Bloc else
}
resta d'instruccions del programa
```

```
Algoritmo EstructuraSeleccion
Si expresion_logica Entonces
    acciones_por_verdadero
SiNo Si expresion_logica2
    acciones_por_verdadero
SiNo
    acciones_por_falso
FinSi
    acciones_por_falso
Fin Si
FinAlgoritmo
```


Pràctica 4

Dissenya i implementa el següent programa:

- 1) Introduir una nota
- 2) Llegir la nota
- 3) Mostra informació sobre els valors possibles de notes (0 - 10)
 - Si és més gran o igual que 9 o igual a 10 la nota és "Excel·lent"
 - Entre 6.5 (inclòs) i 9, la nota és "Notable"
 - Entre 5 (inclòs) i 6.5 la nota és "Aprovat"
 - En qualsevol altre cas "Suspès"

Pràctica 5



Dissenya i implementa el següent programa:

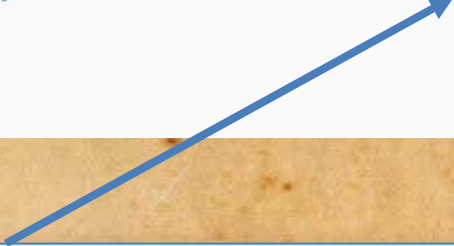
Decidir (fer servir constants):

- Quin és el valor mínim per a optar al descompte
- Quant es descomptarà
- Quin és el valor descomptat màxim possible
- Demanar el preu inicial, en euros
- Llegir-lo
- Comprovar si és positiu o no:
 - En cas positiu, vore si el preu introduït és més gran o igual al valor mínim per optar a l' descompte
- En eixe cas, calcular el descompte.
- Comprovar si el descompte supera el màxim permisible
 - Si és així, el descompte es redueix al màxim permisible
- Aplicar el descompte sobre el preu inicial
- Mostrar el preu final
- Si el preu és negatiu, mostra un missatge d'error

Àmbit de variables

En el moment en que es defineixen diversos blocs ({ }), cada variable declarada té un àmbit definit

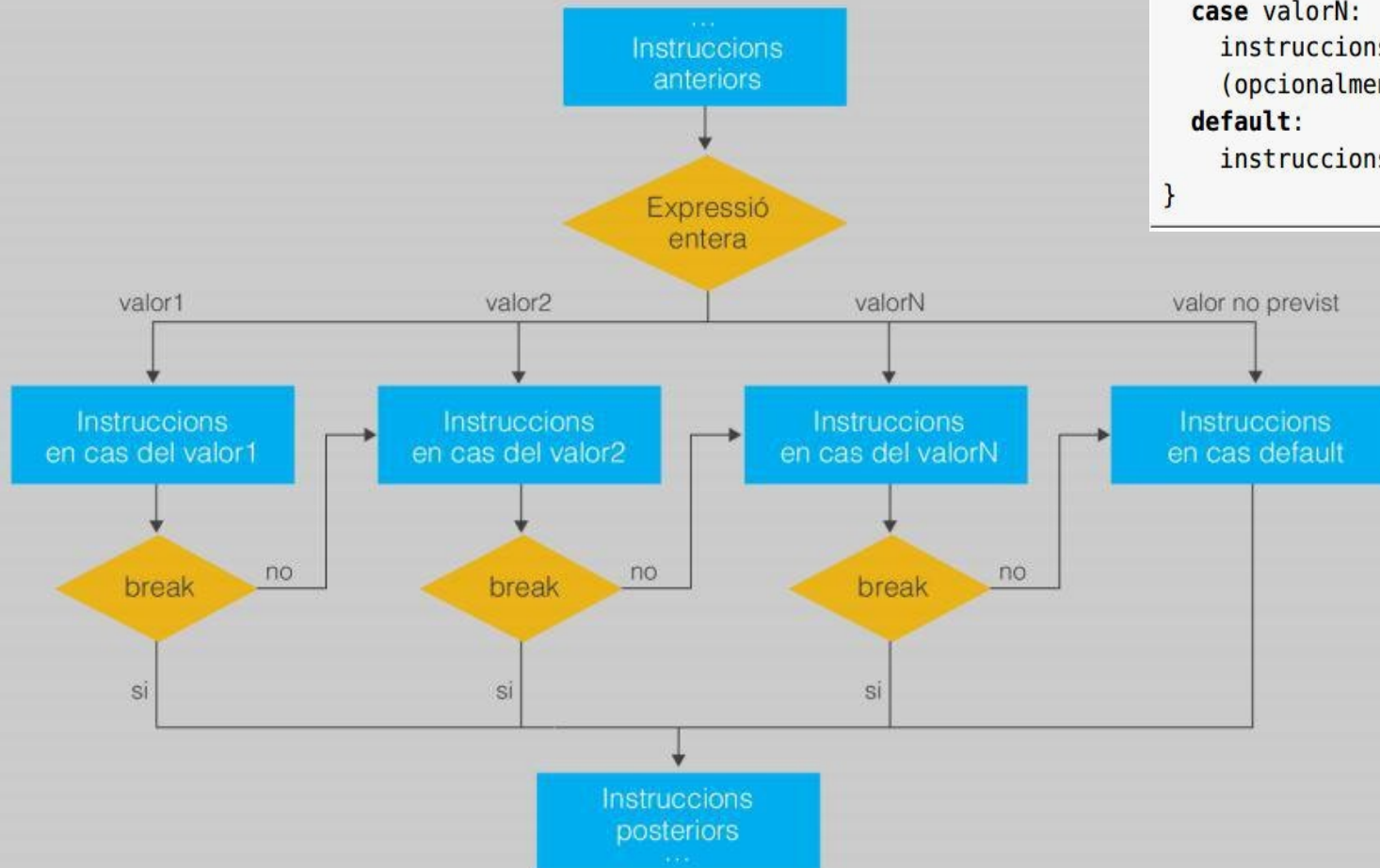
```
public class AmbitVariables {  
    static int a = 1;  
  
    public static void main(String[] args) {  
        int b = 2;  
        System.out.println(a + b);  
        {  
            int c = 3;  
            System.out.println(a + b + c);  
        }  
        System.out.println(a + b + c);  
    }  
}
```



En este punt no existeix la variable "c" ja que està fora del seu àmbit o "scope"

Selecció (Switch)

```
switch(expressió de tipus enter) {  
  case valor1:  
    instruccions si l'expressió avalua a valor1  
    (opcionalment) break;  
  case valor2:  
    instruccions si l'expressió avalua a valor2  
    (opcionalment) break;  
  ...  
  case valorN:  
    instruccions si l'expressió avalua a valorN  
    (opcionalment) break;  
  default:  
    instruccions si l'expressió avalua a algun valor que no és valor1...valorN  
}
```



Esta estructura té una sintaxi una mica diferent a les anteriors.

És interessant que intentes comprendre-la practicant, ja que el seu ús és molt estès.

A tindre en compte

- Es poden **agrupar casos**.
- **No oblides** definir la sentència "**break**" al final de cada "case", si no va agrupat amb el següent cas.
- **No oblides** la sentència "**default**" (en este cas, al ser l'última, no necessita en el seu bloc definir la sentència "break")
- Un bloc "switch" **sempre té el seu equivalent** mitjançant blocs "if - else "

Pràctica 6

- Dissenya i implementa el següent programa:
 - Demana dos nombres enters
 - Mostra un menú que done a triar entre 4 opcions possibles
 - Llig l'opció seleccionada
 - Si és 1, els valors es sumen
 - Si és 2, es resten
 - Si és 3, es multipliquen
 - Si és 4, es divideixen
 - Qualsevol altra opció és incorrecta
 - Mostra el resultat per pantalla

Selecció (Switch)

- Analitza i comprova el funcionament del següent programa

```
public static void main(String[] args) {  
  
    int nota = 5;  
  
    switch (nota) {  
        case 0: case 1: case 2:  
            System.out.println("Fatal");  
            break;  
        case 3: case 4:  
            System.out.println("Mal");  
            break;  
        case 5: case 6: case 7:  
            System.out.println("Bé");  
            break;  
        case 8: case 9: case 10:  
            System.out.println("Molt bé");  
            break;  
        default:  
            System.out.println("Nota incorrecta");  
    }  
}
```