UD4.- Bucles i estructures de decisió

Mòdul: Programació
1r DAM
Curs 2018-2019

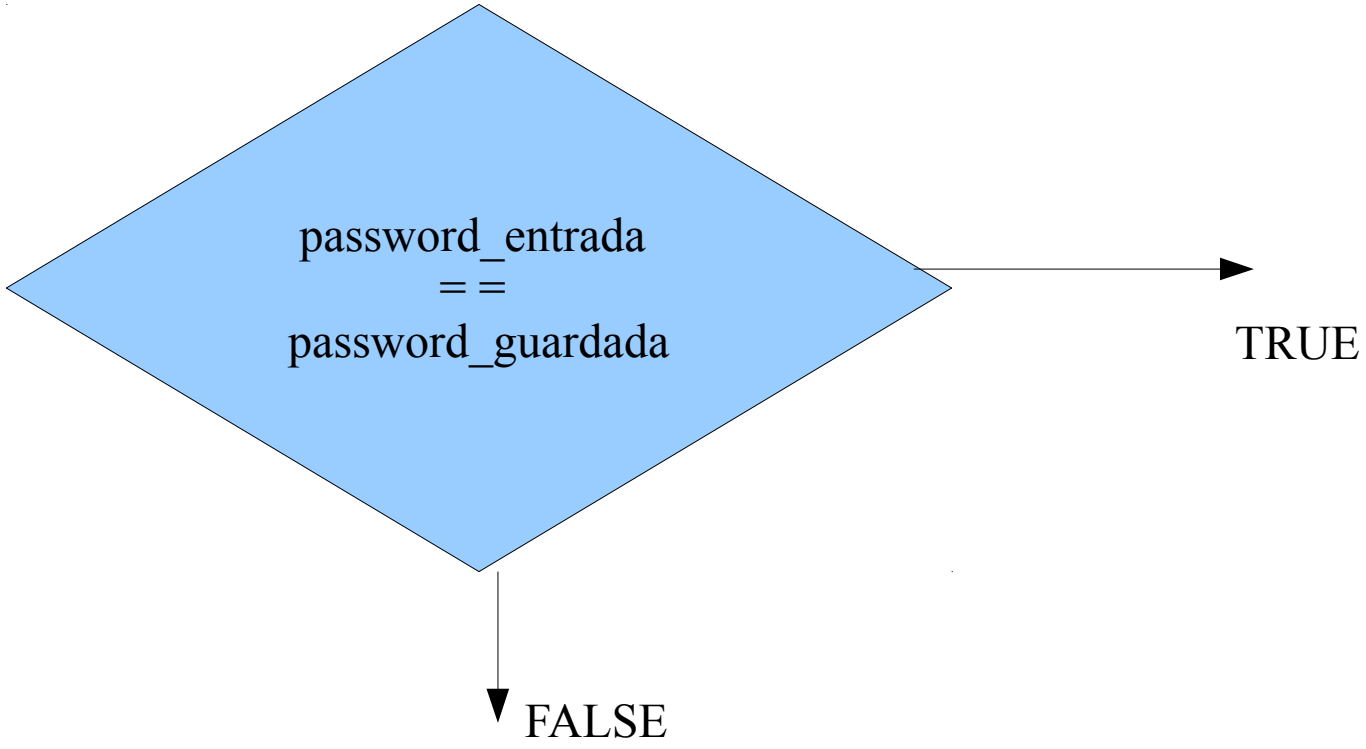
Unión Europea
Fondo Social Europeo
El FSE invierte en tu futuro

INDEX

- Estructures de decisió (bifurcaments)
  - if…
  - if…else
  - if…else if
  - switch … (case)
- Estructures de repetició (bucles)
  - while…
  - do…while
  - for…

Expressions condicionals

- Inclouen una condició a avaluar: true o false
- Inclouen un operador per especificar quin és el resultat de la condició



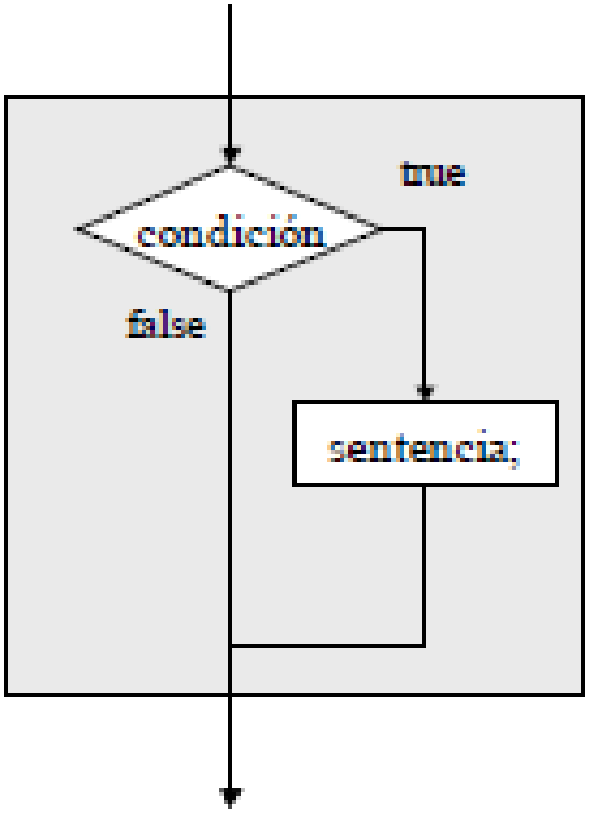
Expressions condicionals

- if …
- if … else
- if … else if …
- switch … case
- operador ?:

Com utilitzar expressions if...

```
if (condició)
    sentència;
```

```
if (condició) {
        sentència 1;
        sentència 2;
        ...
        sentència n;
}
```



Com utilitzar expressions if...

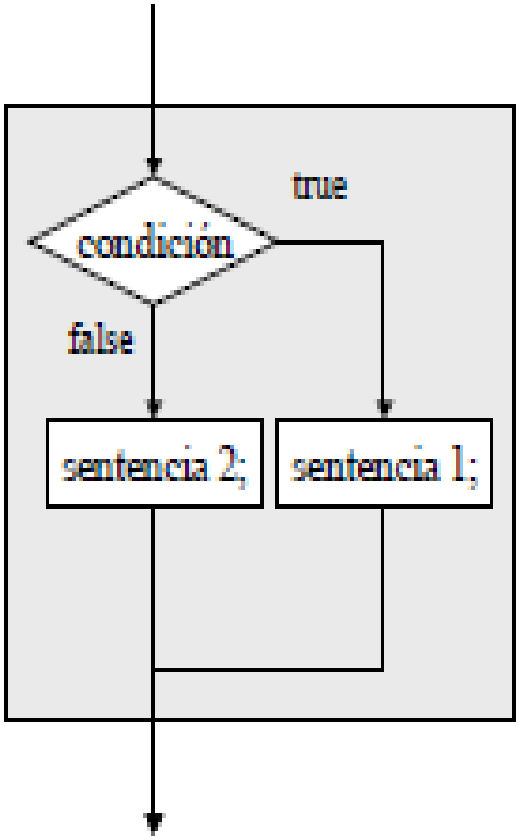
- Si condició == *true* → s'executen les instruccions dins de l'if.
- Si condició == *fals* → les instruccions dins de l'if no s'executen.

```
char caracter='a', caracter2='b';

if (caracter=='a'){
    System.out.println("S'ha llegit el caràcter a.");
}

if (caracter2=='a'){
    System.out.println("S'ha llegit el caràcter a.");
}
```

Com utilitzar expressions if … else

```
if (condició)
      sentència 1;
else
      sentència 2;
```

```
if (condició) {
      sentència 11;
      ..
      sentència 1n;
} else {
      sentència 21;
      ...
      sentència 2n;
}
```



Com utilitzar expressions if ... else

- Si condició == *true* → s'executen les instruccions dins de l'*if*.
- Si condició == *fals* → les instruccions dins de l'*if* no s'executen. S'executen les de l'*else*.

```
    int edad=15;
    if (edad < 18) {
        // codi a realitzar si la condició es complix
        System.out.println("no puedes salir a la hora del
patio");
    } else {
        // codi a realitzar si la condició no es complix
        System.out.println("puedes salir");
    }
```

Com utilitzar expressions if ... else

```
if ( condició )
        Bloc de codi a executar si la condició és certa
else
        Bloc de codi a executar si la condició és falsa
```

- La part de l'*else* és opcional
- El bloc { } s'utilitza per a més d'una instrucció.
- Un bloc de codi pot ser únicament la sentència buida ; → significa que no s'ha d'executar res.

Com utilitzar expressions if ... else if ...

S'utilitzen per aniuar estructures de decisió

```
int mes=5;

if (mes == 1)
    System.out.print("gener");

else if (mes == 2)

    System.out.print("febrer");

else if (mes == 3)
    System.out.print("març");

else
    System.out.print("No sé...");
```

```
int mes=5;

if (mes == 1)
     System.out.print("gener");

else
     if (mes == 2)
          System.out.print("febrer");

     else
          if (mes == 3)
               System.out.print("març");

          else
               System.out.print("No sé...");
```

Com utilitzar expressions switch ... case

Construcció sintàctica molt compacta per seleccionar un bloc de codi a executar depenent d'un valor.

S'utilitza com alternativa a instruccions if ... else aniuaes.

Podem aplicar el switch sobre els tipus primitius **short**, **char**, **int**. També es pot utilitzar per a t**ipus enumerats** i **Strings** (desde la versió 7 de Java).

- Comprova que no n'hi ha duplicats
- Les condicions han de ser excloents.

```
int mes=2;
switch(mes) {

    case 1:
        System.out.println("Gener");
        break;

    case 2:
        System.out.println("Febrer");
        break;

    case 3:
        System.out.println("Març");
        break;

    default:
        System.out.println("No sé.");
        break;
}
```

Com utilitzar expressions switch ... case

La sentència ***break*** provoca l'acabament de la sentència condicional.

- Si no apareix, el codi següent continua executant-se.

El *default* és opcional

- Si no apareix, no s'executa res.

```
switch (mes) {
    case 1: case 3: case 5: case 7:
    case 8: case 10: case 12:
        dias = 31;
        break;

    case 4: case 6: case 9: case 11:
        dias = 30;
        break;

    case 2:
        if (bisiesto)
            dias = 29;
        else
            dias = 28;
        break;

    default:
        dias = 0;
}
```

Directrius per triar una estructura de decisió

- Les instruccions if... s'utilitzen per controlar l'execució d'un únic bloc de codi.

- Les instruccions if...else s'utilitzen per controlar l'execució de dos seccions de codi mutuament excloents.

- Les instruccions case... s'utilitzen quan es disposa d'una llista de valors possibles.

Operador ?:

És una forma compacta de decidir entre dos valors.

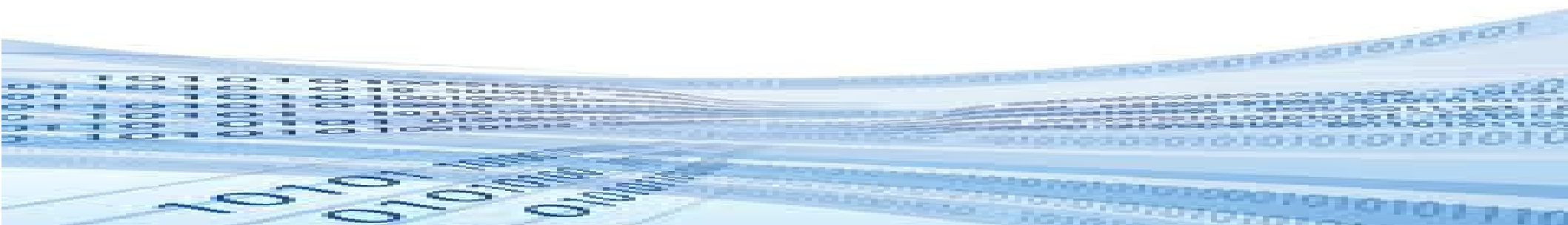
condició ? valor1 : valor2

- Si la condició és certa → es pren el valor1
- Si la condició és falsa → es pren el valor2

Els dos valors han de ser del mateix tipus o tipus compatibles (*casting*).

```
//forma compacta

variable = condició ? valor1:valor2;
```

```
//forma clàssica
if (condició)
   variable = valor1;
else
   variable = valor2;
```



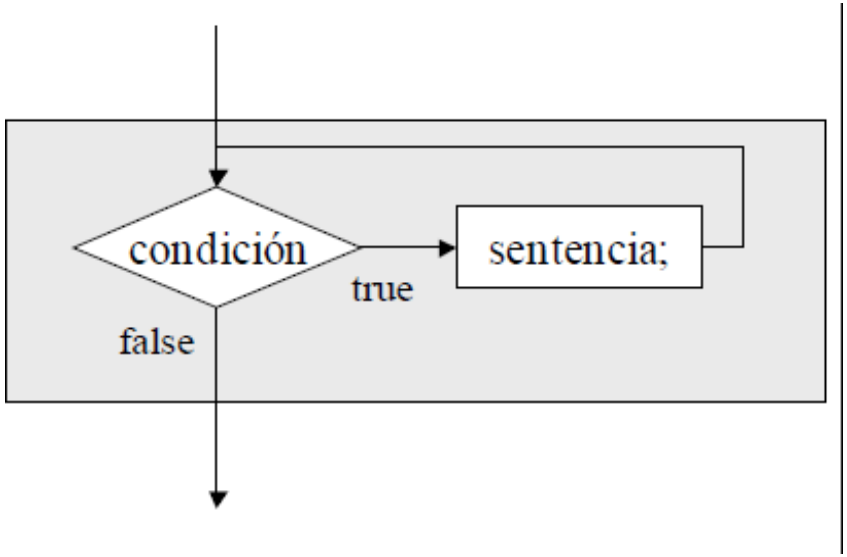
Estructures repetitives (bucles)

- while…
- do … while
- for…
- Sentències break i continue

Com utilitzar expressions while...

```
while (condició)
    sentència;
```

```
while (condició) {
    sentència 1;
    sentència 2;
    ...
    sentència n;
}
```



Com utilitzar expressions while...

Executa el codi del bucle únicament si la condició s'avalua a true i es repetix fins que l'expressió siga falsa.

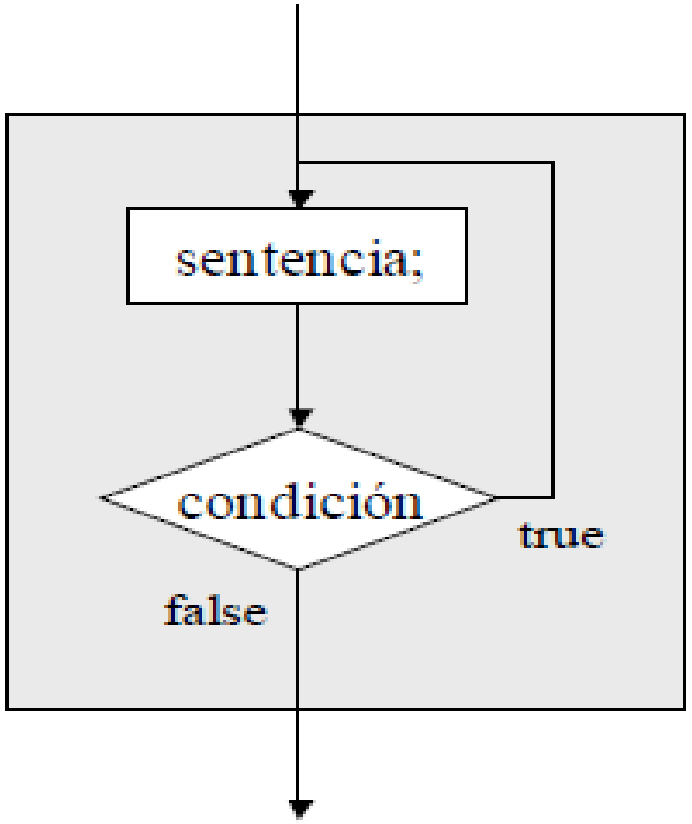
- S'executarà 0 o més vegades
- La condició d'acabament es comprova al principi del bucle.

```
//càlcul del factorial de n
int fact = 1;

while (n > 0) {
   fact = fact*n;
   n--;
}
```

Com utilitzar expressions do...while

```
do
      sentència;
while (condició);
```

```
do {
      sentència 1;
      sentència 2;
      ...
      sentència n;
} while (condició);
```



Com utilitzar expressions do...while

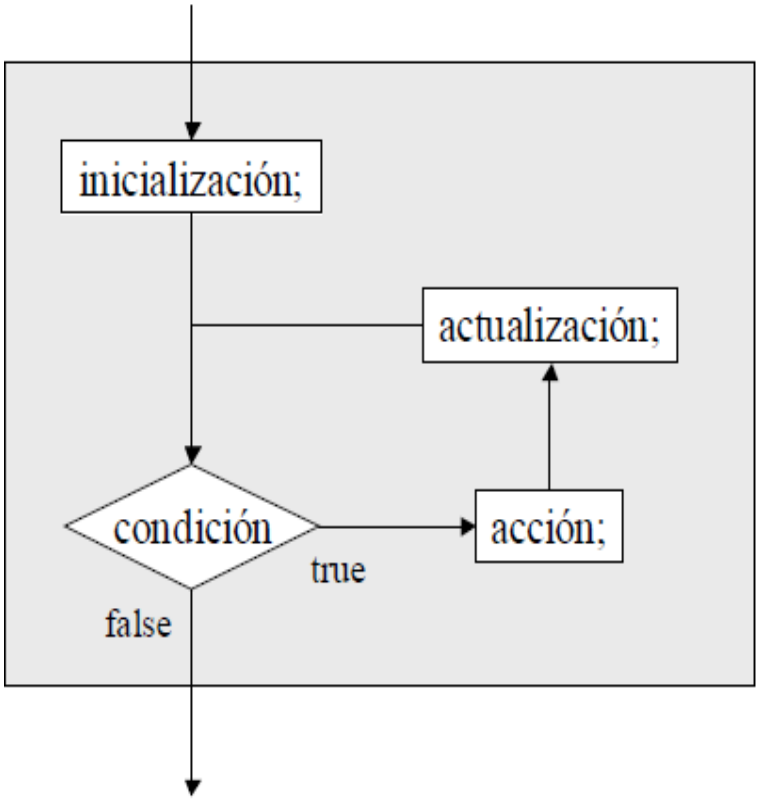
- Executa el codi del bucle i després avalua la condició. Repetix fins que la condició siga falsa.
  - S'executarà 1 o més vegades
  - La condició d'acabament es comprova al final del bucle.

```
//demanar una nota fins que siga major o igual que 5
double nota;

do {
   nota = lector.nextDouble();
} while (nota < 5.0);
```

Com utilitzar expressions for...

```
for(inicialització;condició;actualització)
    sentència;
```

```
for(inicialització;condició;actualització) {
    sentència 1;
    sentència 2;
    ...
    sentència n;
}
```



Com utilitzar expressions for...

S'utilitzen quan coneguem el nombre de vegades que desitgem que es repetisca l'execució d'un codi.

```
for (i = 0; i < 10; i++)
   System.out.print(i);
```

```
// m és el número del que volem saber el factorial
int fact = 1;
for (n = m; n > 0; n--)
   fact*= n;
```

Com utilitzar expressions for...

Els bucles “for” tradicionalment utilitzen un comptador local al bucle.

- Local al bucle vol dir que el seu àmbit és el del bucle i no es pot emprar fora del bucle.
- Noms de variables comuns per als comptadors són: i, j, k, cont.

```
for (int i = 0; i < 10; i++)
   System.out.print(i);
```

Sentències *break* i *continue*

- Quan un bucle està en execució (llançat), Java disposa de dos sentències per forçar el seu comportament:
  - **break**: provoca l'acabament del bucle. Avorta el bucle.
  - **continue**: provoca el començament d'una nova repetició. Avorta la passada pel bucle actual.

- Es recomana **NO** utilitzar sentències break i continue a no ser que siga evident la seua utilització.