UD4.- Bucles y estructuras de decisión

Módulo: Programación
1.º DAM

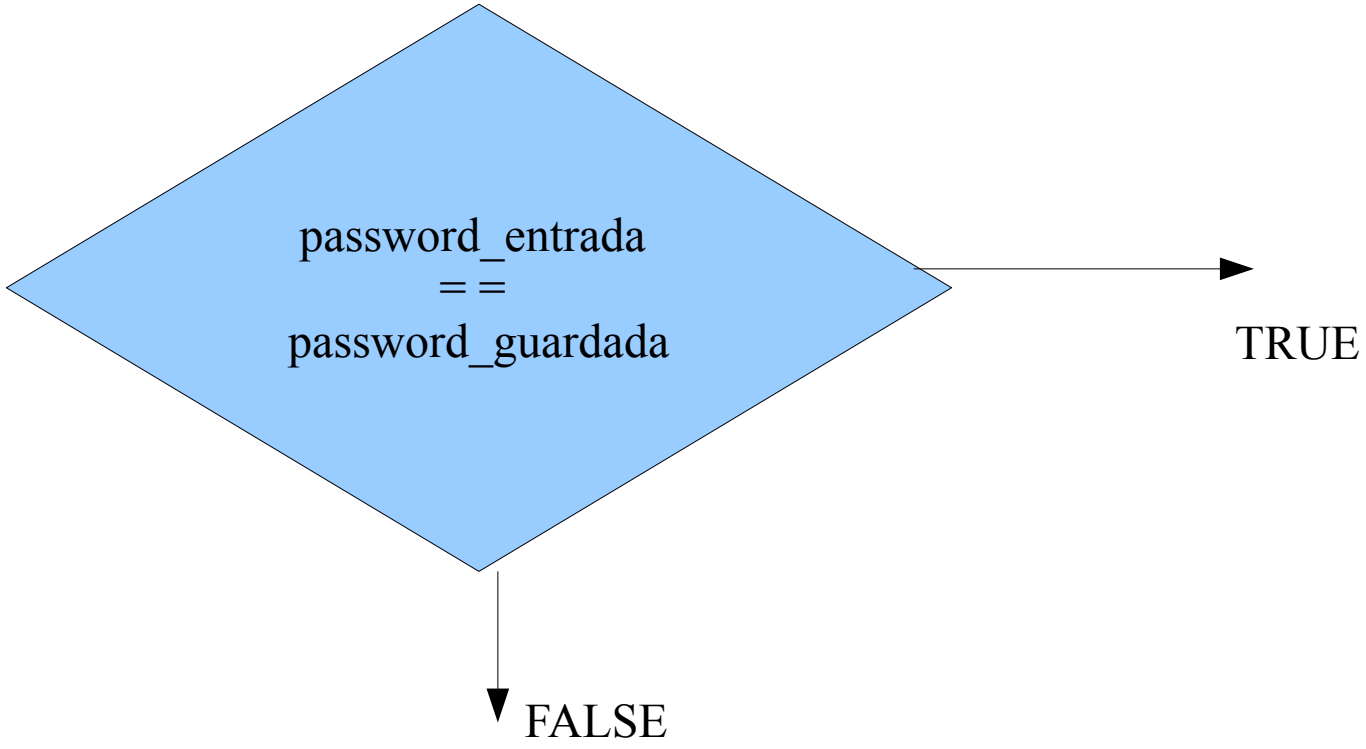
Unión Europea
Fondo Social Europeo
El FSE invierte en tu futuro

INDEX

- Estructuras de decisión (bifurcación)
  - if…
  - if…else
  - if…else if
  - switch … (case)
- Estructuras de repetición (bucles)
  - while…
  - do…while
  - for…

Expresiones condicionales

- Incluyen una condición a evaluar: true o false
- Incluyen un operador para especificar cuál es el resultado de la condición



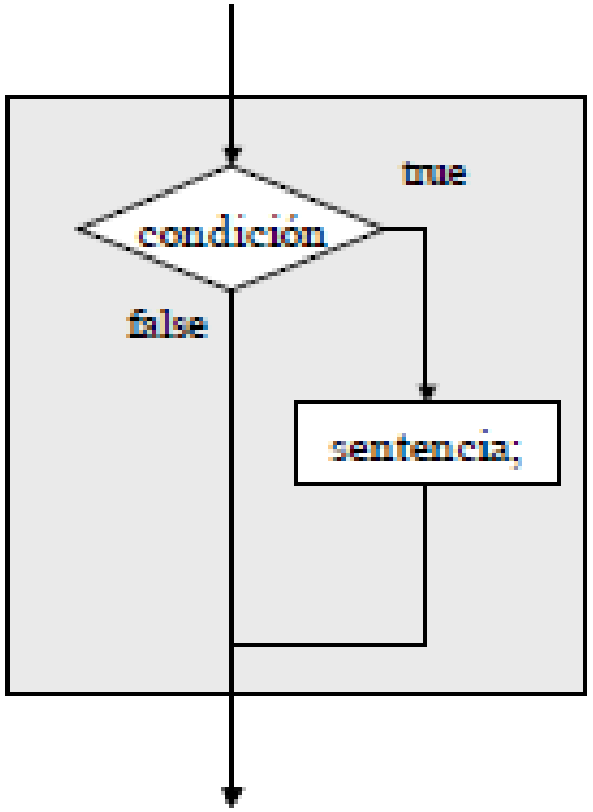
Expresiones condicionales

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

Como utilizar expresiones if...

```
if (condición)
    sentencia;
```

```
if (condición) {
        sentencia 1;
        sentencia 2;
        ...
        sentencia n;
}
```



Como utilizar expresiones if...

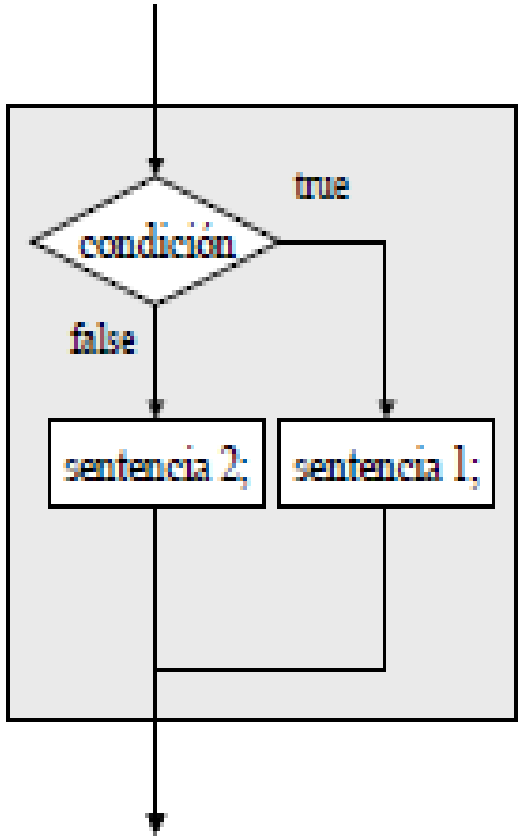
- Si condición == *true* → se ejecutan las instrucciones dentro de la if.
- Si condición == *falso* → las instrucciones dentro de la if no se ejecutan.

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

if (caracter=='a'){
    System.out.println("Se ha leído el carácter a.");
}

if (caracter2=='a'){
    System.out.println("Se ha leído el carácter a.");
}
```

Como utilizar expresiones if … else

```
if (condición)
      sentencia 1;
else
      sentencia 2;
```

```
if (condición) {
        sentencia 11;
        ..
        sentencia 1n;
} else {
        sentencia 21;
        ...
        sentencia 2n;
}
```



Como utilizar expresiones if ... else

- Si condición == *true* → se ejecutan las instrucciones dentro del *if*.
- Si condición == *falso* → las instrucciones dentro del *if* no se ejecutan. Se ejecutan las del *else*.

```
    int edad=15;
    if (edad < 18) {
        // código a realizar si la condición se cumple
        System.out.println("no puedes salir a la hora del
patio");
    } else {
        // código a realizar si la condición no se cumple
        System.out.println("puedes salir");
    }
```

Como utilizar expresiones if … else

```
if ( condición )
        Bloque de código a ejecutar si la condición es cierta
else
        Bloque de código a ejecutar si la condición es falsa
```

- La parte del *else* es opcional
- El bloque { } se utiliza para más de una instrucción.
- Un bloque de código puede ser únicamente la sentencia vacía ; → significa que no se tiene que ejecutar nada.

Como utilizar expresiones if ... else if ...

Se utilizan para anidar estructuras de decisión

```
int mes=5;

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

else if (mes == 2)

    System.out.print("febrero");

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

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

```
int mes=5;

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

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

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

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

Como utilizar expresiones switch ... case

Construcción sintáctica muy compacta para seleccionar un bloque de código a ejecutar dependiente de un valor.

Se utiliza como alternativa a instrucciones if ... else anidadas.

Podemos aplicar el switch sobre los tipos primitivos **short**, **char**, **int**. También se puede utilizar para t**ipos enumerados** y Strings (desde la versión 7 de Java).

- Comprueba que no hay duplicados
- Las condiciones tienen que ser excluyentes.

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

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

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

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

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

Como utilizar expresiones switch ... case

La sentencia ***break*** provoca el acabado de la sentencia condicional.

- Si no aparece, el código siguiente continúa ejecutándose.

El *default* es opcional

- Si no aparece, no se ejecuta nada.

```
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;
}
```

Directrices para elegir una estructura de decisión

- Las instrucciones if… se utilizan para controlar la ejecución de un único bloque de código.

- Las instrucciones if…else se utilizan para controlar la ejecución de dos secciones de código mutuamente excluyentes.

- Las instrucciones case… se utilizan cuando se dispone de una lista de valores posibles.

Operador ?:

Es una forma compacta de decidir entre dos valores.

condición ? valor1 : valor2

- Si la condición es cierta → se toma el valor1
- Si la condición es falsa → se toma el valor2

Los dos valores tienen que ser del mismo tipo o tipos compatibles (*casting*).

```
//forma compacta

variable = condición ? valor1:valor2;
```

```
//forma clásica
if (condición)
    variable = valor1;
else
    variable = valor2;
```

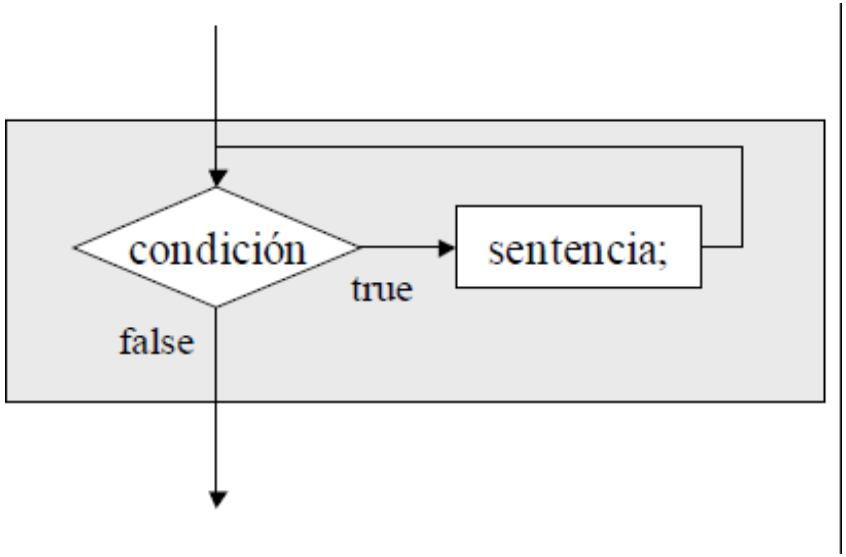
Estructuras repetitivas (bucles)

- while…
- do … while
- for…
- Sentencias break y continue

Como utilizar expresiones while...

```
while (condición)
    sentencia;
```

```
while (condición) {
    sentencia 1;
    sentencia 2;
    ...
    sentencia n;
}
```



Como utilizar expresiones while...

Ejecuta el código del bucle únicamente si la condición se evalúa a true y se repite hasta que la expresión sea falsa.

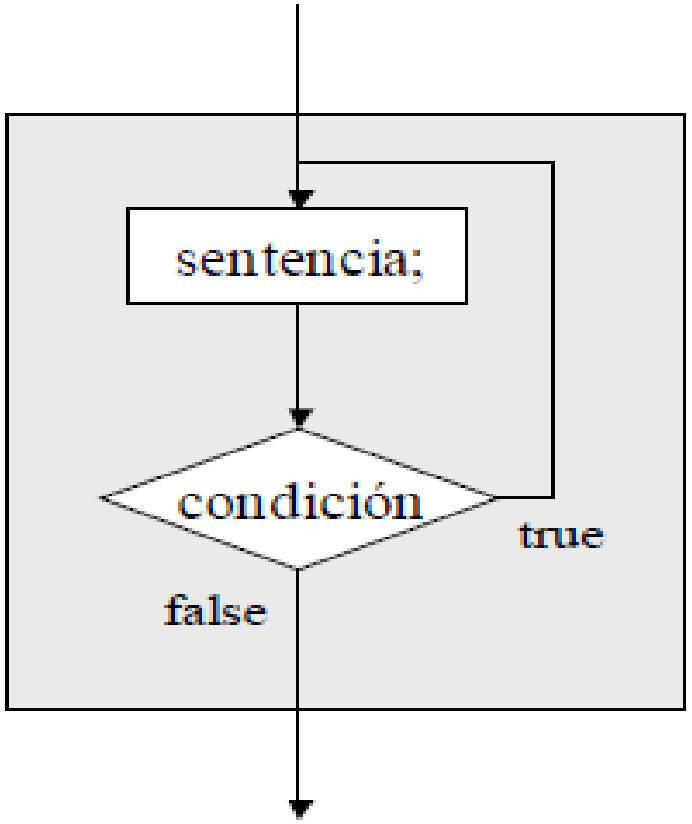
- Se ejecutará 0 o más veces
- La condición de finalización se comprueba al principio del bucle.

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

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

Como utilizar expresiones do...while

```
do
      sentencia;
while (condición);
```

```
do {
      sentencia 1;
      sentencia 2;
      ...
      sentencia n;
} while (condición);
```



Como utilizar expresiones do...while

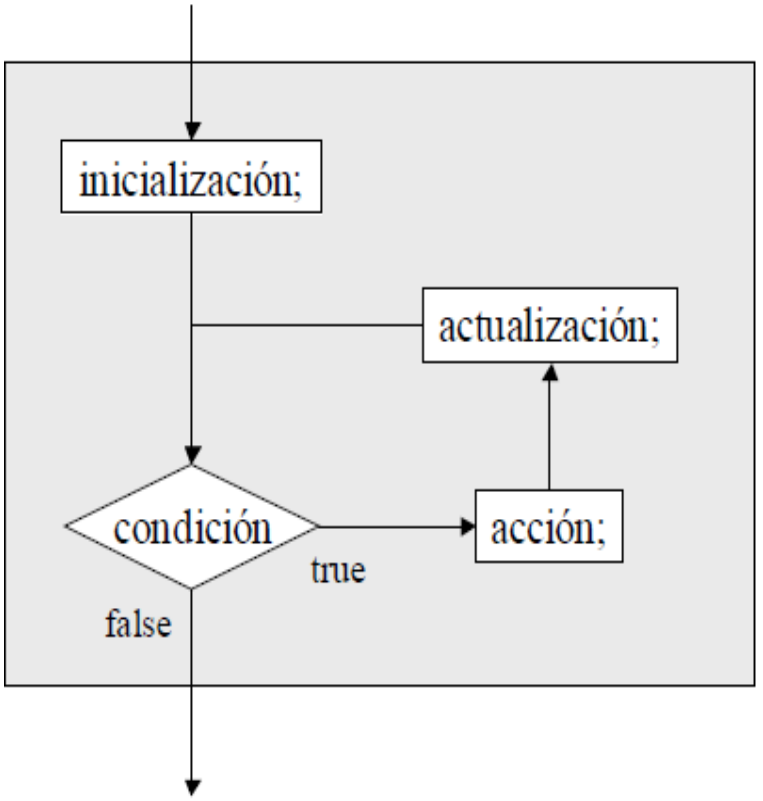
- Ejecuta el código del bucle y después evalúa la condición. Repite hasta que la condición sea falsa.
  - Se ejecutará 1 o más veces
  - La condición de acabado se comprueba al final del bucle.

```
//pedir una nota hasta que sea mayor o igual que 5
double nota;

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

Como utilizar expresiones for...

```
for(inicialización;condición;actualización)
    sentencia;
```

```
for(inicialización;condición;actualización) {
    sentencia 1;
    sentencia 2;
    ...
    sentencia n;
}
```



Como utilizar expresiones for...

Se utilizan cuando conocemos el número de veces que deseamos que se repita la ejecución de un código.

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

```
// queremos saber el factorial de m
int fact = 1;
for (n = m; n > 0; n--)
   fact*= n;
```

Como utilizar expresiones for...

Los bucles “for” tradicionalmente utilizan un contador local al bucle.

- Local al bucle quiere decir que su ámbito es el del bucle y no se puede emplear fuera del bucle.
- Nombres de variables comunes para los contadores son: i, j, k, cont.

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

Sentencias *break* y continue.

- Cuando un bucle está en ejecución (lanzado), Java dispone de dos sentencias para forzar su comportamiento:
  - **break**: provoca el acabado del bucle. Aborta el bucle.
  - **continue**: provoca el comienzo de una nueva repetición. Aborta la iteración del bucle actual.

- Se recomienda **NO** utilizar sentencias break y continue a no ser que sea evidente su utilización.