UD3. Estructuras condicionales y Bucles

Módulo: Programación

Unión Europea
Fondo Social Europeo
El FSE invierte en tu futuro

Germán Gascón Grau
g.gascongrau@edu.gva.es

Contenidos

- Estructuras de decisión (bifurcación)
  - if
  - if … else
  - if … else if
  - switch …. case
  - ?: (operador ternario)
- Estructuras de repetición (bucles)
  - while …
  - do … while
  - for ...

Expresiones condicionales

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

Expresiones condicionales

- Las expresiones condicionales contienen una o varias condiciones que son evaluadas a un valor boolean: true o false.
- Estas expresiones están formadas por operadores condicionales y operadores lógicos.
- Por ejemplo:

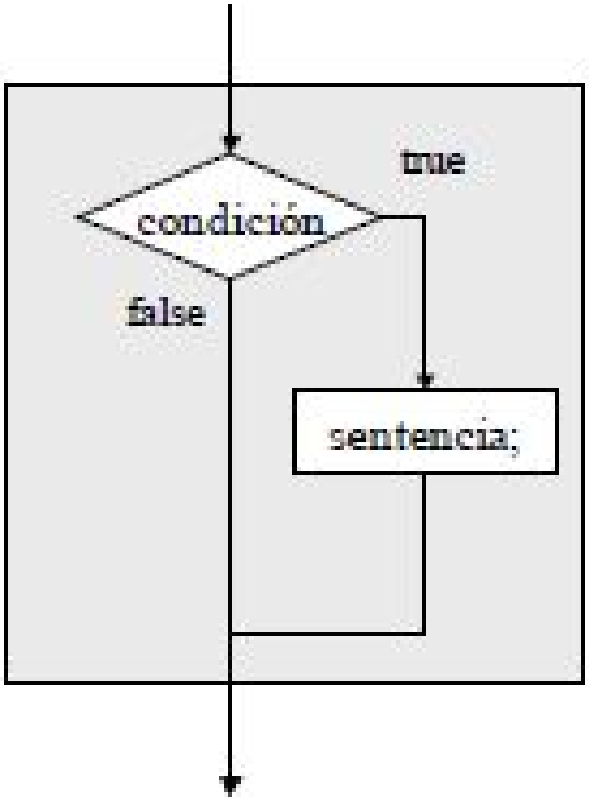
```
int x = 8;
boolean resultado;
resultado = (x > 5 && x < 10);
```

Toma de decisiones con if

- **if** permite tomar decisiones en función del valor de una expresión condicional.

```
if (condición)
    sentencia;
```

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



Cómo utilizar expresiones if ...

- Si condición == true → se ejecutan las instrucciones dentro del bloque if.
- Si condición == false → las instrucciones dentro del bloque if no se ejecutan.

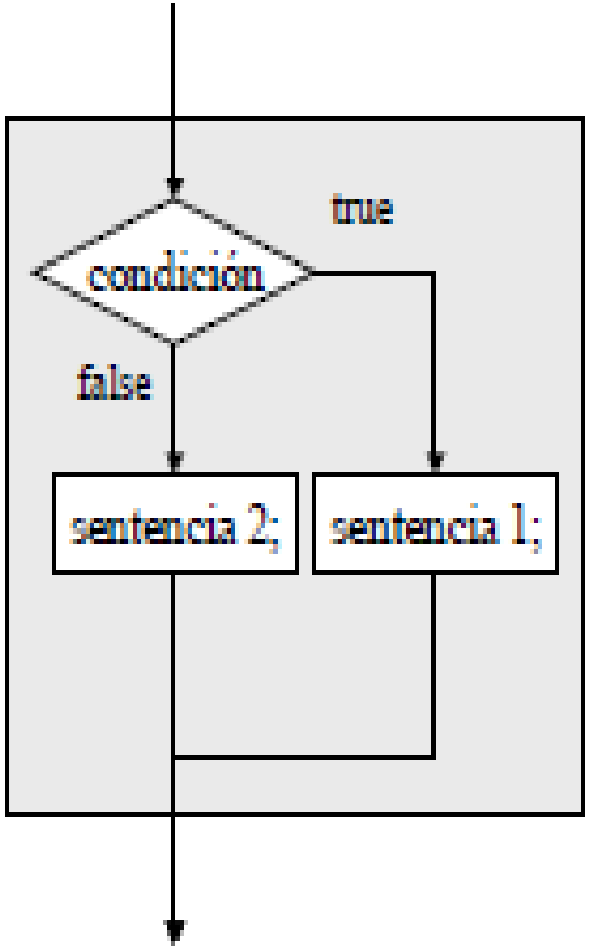
```
char caracter1 = 'a';
char caracter2 = 'b';
if (caracter1 == 'a'){
    System.out.println("caracter1 es a");
}

if (caracter2 == 'a'){
    System.out.println("caracter2 es a");
}
```

Toma de decisiones con if … else

```
if (condición)
    sentencia1;
else
     sentencia2;
```

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



Cómo utilizar expresiones if … else

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

```
int edad = 18;
if (edad < 18) {
    // código a realizar si la condición se cumple
    System.out.println("Sólo los mayores de edad pueden salir a la calle a la hora del patio");
} else {
    // código a realizar si la condición no se cumple
    System.out.println("Puedes salir a la calle a la hora del patio");
}
```

Cómo 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.

Cómo 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...");
```

Cómo utilizar expresiones switch … case

- Construcción sintáctica 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 tipos enumerados y Strings.
- Comprueba que no hay duplicados.
- No comprueba que se traten todos los casos.
- Las condiciones tienen que ser excluyentes.

Cómo utilizar expresiones switch … case

- Si no se incluye una sentencia **break**, todos los siguientes case serán ejecutados (efecto fallthrough). Desde Java 12 existe una versión de switch que no tiene fallthrough.

- El ***default*** es opcional. Si no aparece, no se ejecuta nada.

Sintaxis

```
switch (condición) {
    case valor1:
        sentencias;
        break;
    case valor2:
        sentencias;
        break;
    default:
        sentencias;
        break;
}
```

Ejemplos switch ... case

```
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;
}
```

```
int mes = 3;
int dias;
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 (esBisiesto(mes))
         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 **if...else if...** se utilizan para controlar la ejecución de **dos o más secciones de código** mutuamente excluyentes.
- Las instrucciones **switch...case...** se utilizan cuando se dispone de una **lista de valores posibles**.

Operador ?:

- Es una forma compacta de decidir entre dos valores (if...else…)
- Devuelve el valor por lo que puede ser utilizado en asignaciones.

Sintaxis

```
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).

Ejemplo de uso del operador ?:

```
// forma clásica
if (edad >= 65)
    descuento = 20;
else
    descuento = 10;
```

```
// forma compacta

descuento = edad >= 65 ? 20 : 10;
```

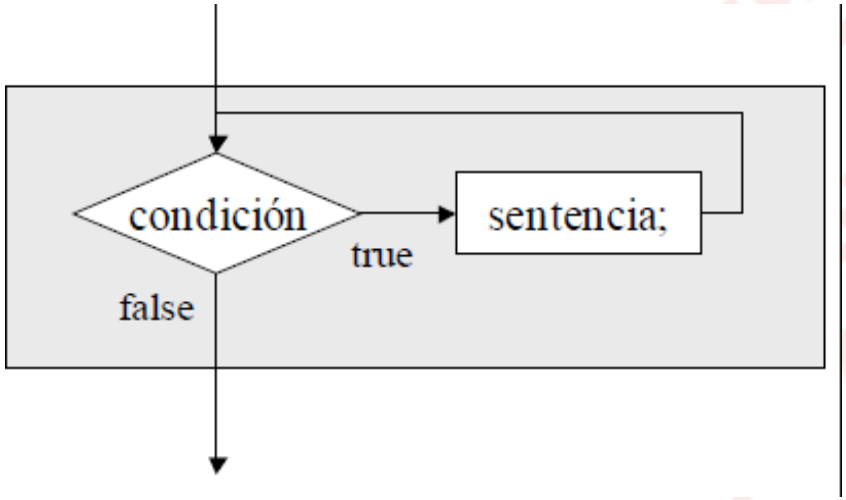
Estructuras repetitivas (bucles)

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

Cómo utilizar expresiones while...

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

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



Cómo 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.

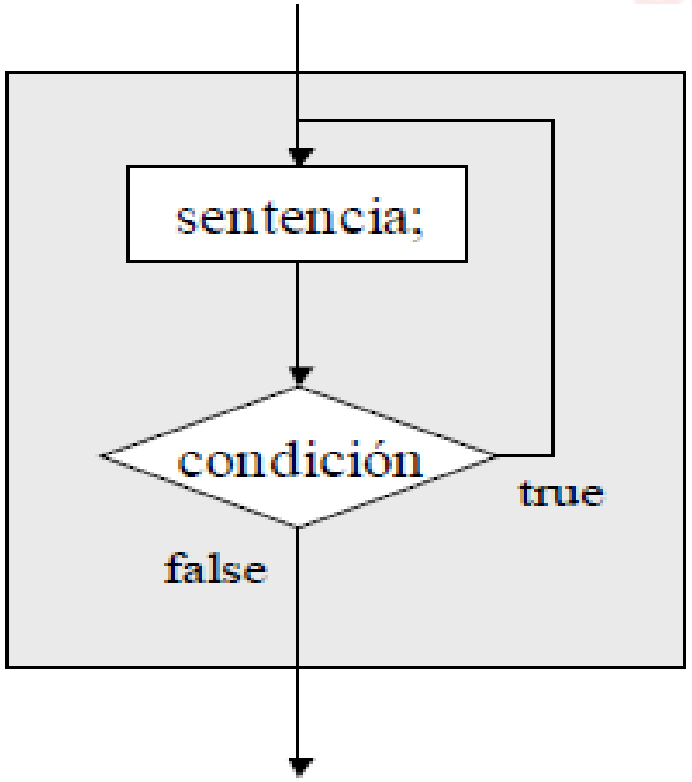
```
// cálculo del factorial de n
int fact = 1;

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

Cómo utilizar expresiones do...while

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

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



Cómo 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.

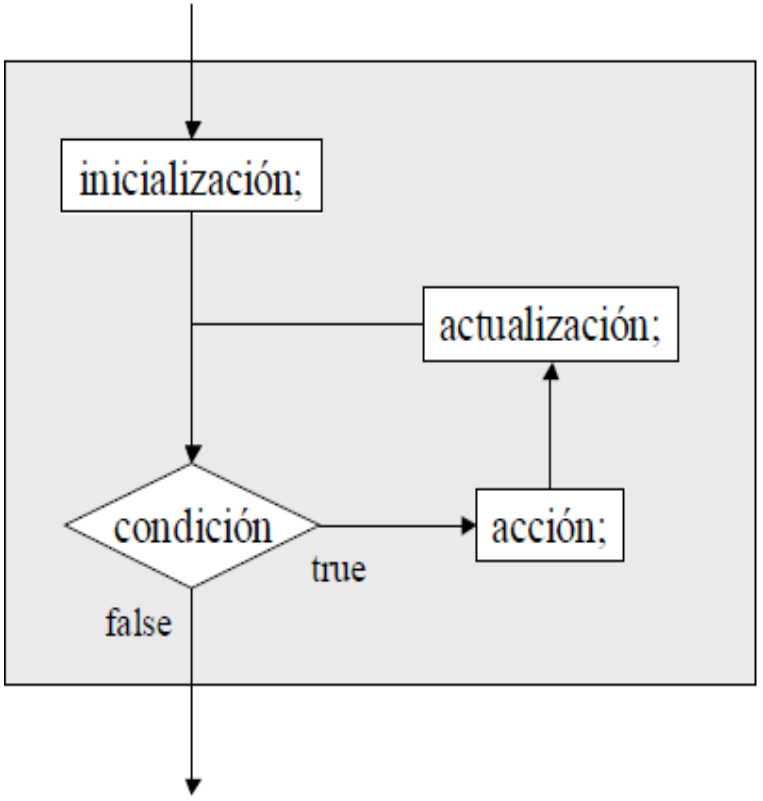
```
// pedir una nota hasta que sea mayor o igual que 5
double nota;

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

Cómo utilizar expresiones for...

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

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



Cómo utilizar expresiones for...

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

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

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

- ¿Cómo sería la versión ascendente del bucle?