

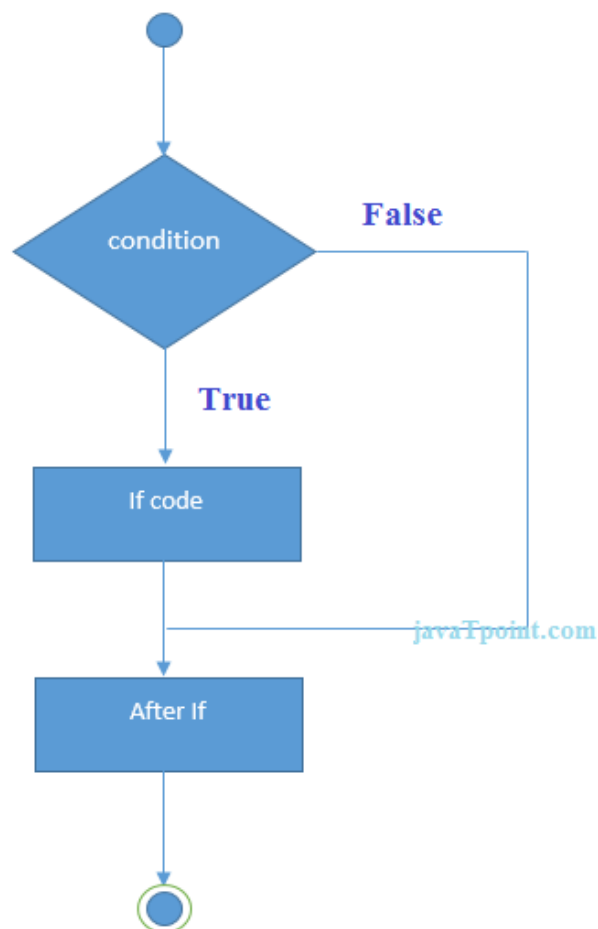
Estructuras de control: condicionales y bucles

Condicionales: if-else

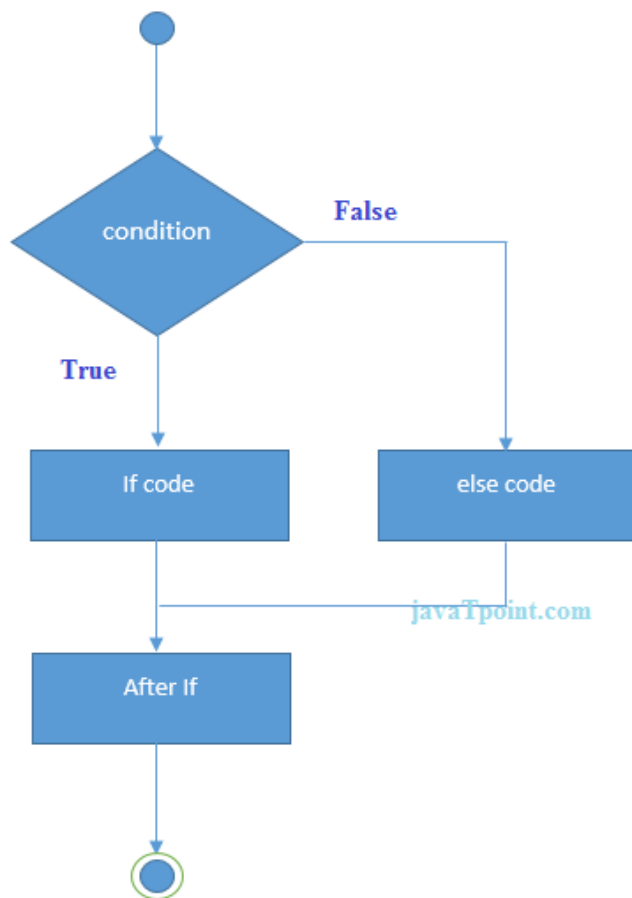
Un programa no tiene por qué ejecutar siempre la misma secuencia de instrucciones. Puede darse el caso de que, dependiendo del valor de alguna expresión o de alguna condición, interese ejecutar o evitar un conjunto de instrucciones.

Ya hemos visto los **operadores relacionales** (`==`, `!=`, `>`, `>=`, etc.) y los **lógicos** (`&&`, `||`, `!`), con ellos podremos establecer las condiciones para nuestras sentencias condicionales.

En la siguiente figura se ve el flujo de control de un **condicional simple**, `if`:

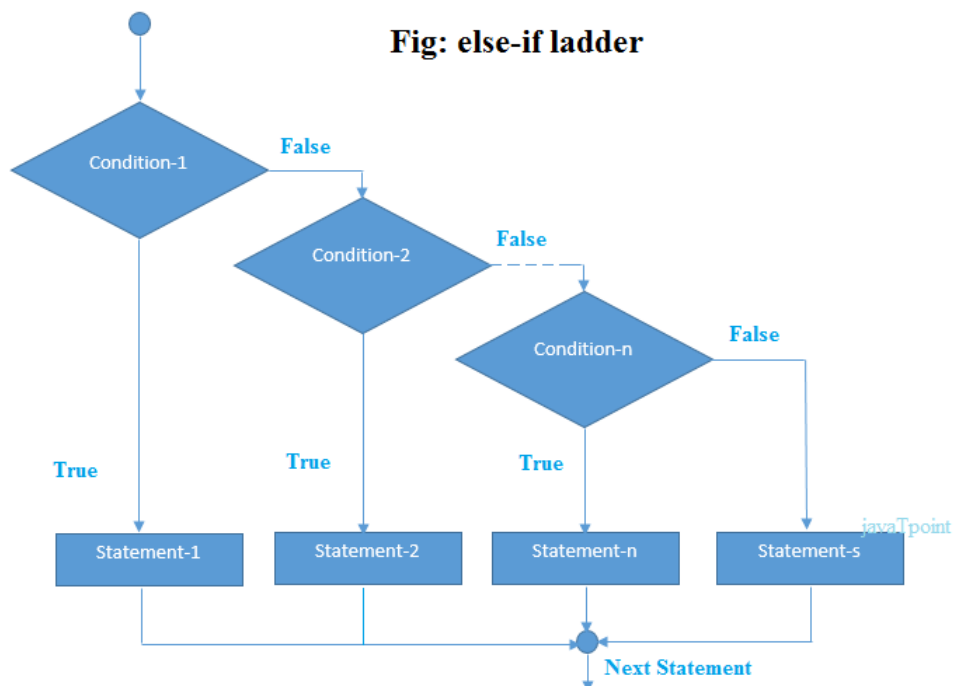


En la siguiente figura, el flujo de control de una **condicional doble**, `if-else`:



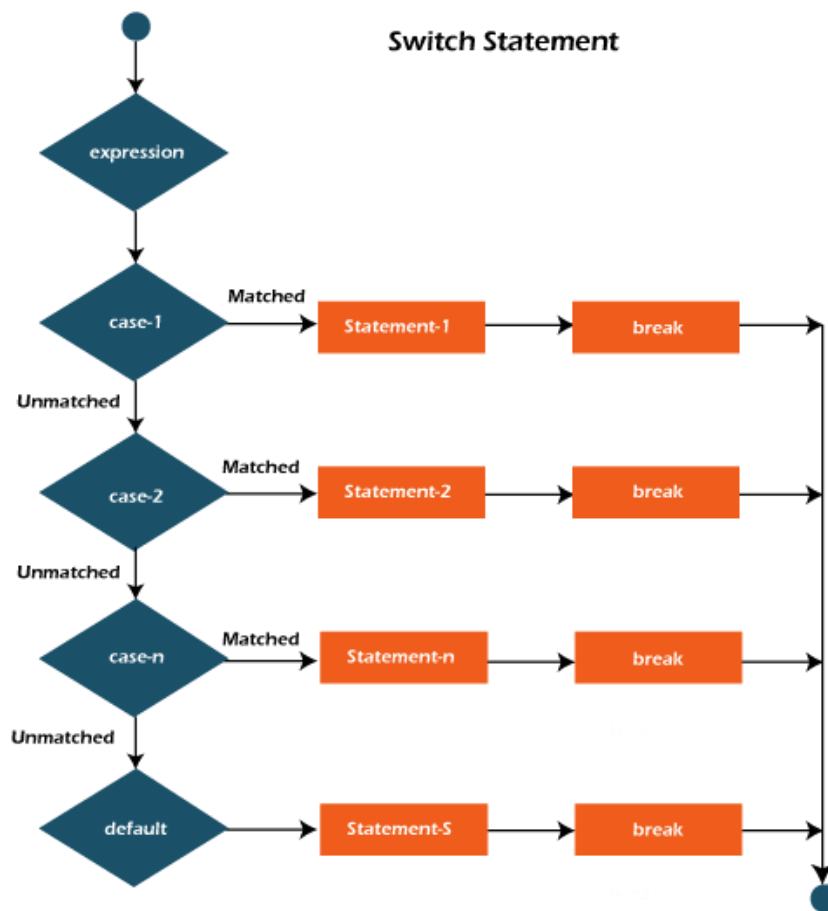
Destacar que las condicionales se pueden anidar unas dentro de otras, resultando en estructuras condicionales más complejas:

Fig: else-if ladder



Condicional múltiple: switch

En ocasiones, utilizar muchos `if-else` anidados puede dar como resultado un código poco legible. Para ello nos encontramos con la sentencia `switch`:



La cláusula `default` es opcional, se suele utilizar cuando no hay otra coincidencia. Recordemos la sentencia `break` nos asegurará que no se ejecuten todas las sentencias desde la primera coincidencia encontrada.

Bucles: estructuras de repetición

Un bucle es una estructura que contiene un bloque de instrucciones que se ejecutan repetidas veces. A cada repetición se le llama **iteración**.

Java dispone de tres tipos de bucles: `for`, `while`, `do-while`.

Bucles for

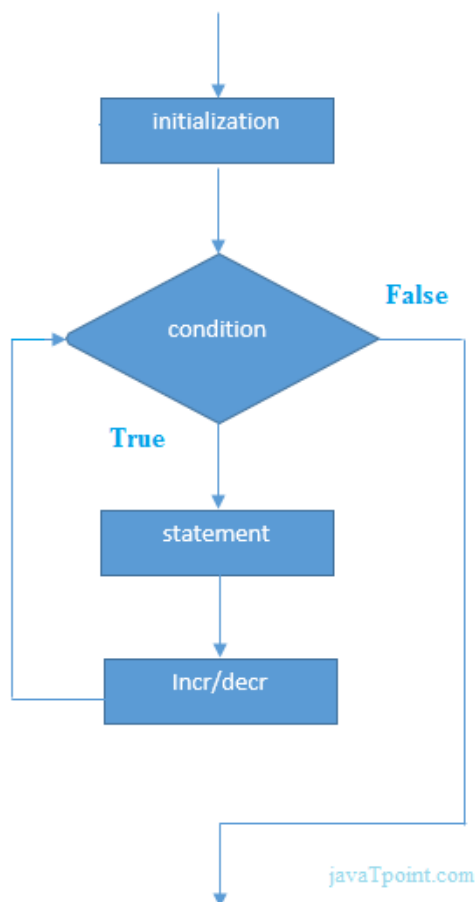
Los bucles `for` tienen la siguiente estructura:

```
for (inicialización; condición; incremento) {  
    Bloque de instrucciones  
    ...  
}
```

Inicialización: lista de instrucciones, separadas por comas, donde se inicializan las variables que van a controlar el bucle. Se ejecutan una sola vez antes de la primera iteración. Por convención, se suelen utilizar variables enteras llamadas, i, j, k, etc.

Condición: expresión booleana que controla las iteraciones del bucle. Se evalúa en cada iteración, y el bloque de instrucciones se ejecutará cuando esta condición es true.

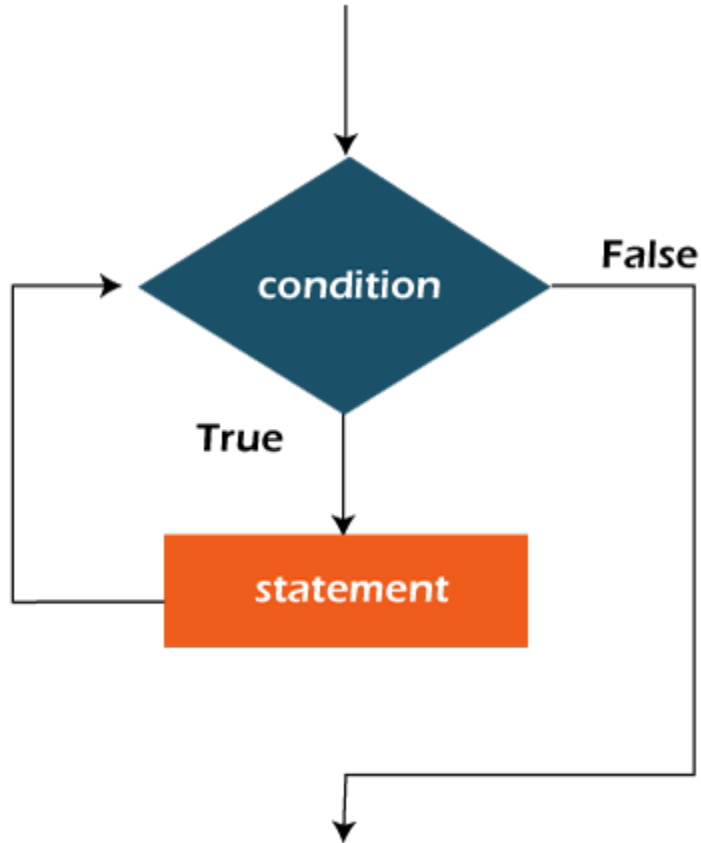
Incremento: lista de instrucciones, separadas por comas, donde se modifican las variables que controlan la condición. Se ejecuta al final de cada iteración. También pueden ser de decremento.



Bucles while

Semejante a la instrucción if, el bucle while realiza iteraciones mientras se cumpla la condición:

```
while (condición) {  
    Bloque de instrucciones  
    ...  
}
```



La diferencia con for, es que en general utilizaremos for para iterar un número determinado de veces, fijo. Mientras que con while no tenemos un número fijo de iteraciones.

Bucles do-while

Es similar al bucle while, con la diferencia de que primero se ejecuta el bloque de instrucciones y después se evalúa la condición:

```
do {  
    Bloque de instrucciones  
    ...  
} while (condición);
```

Sentencias break y continue: salida anticipada

La sentencia `break` sirve tanto para las estructuras de selección como para las de repetición. El programa al encontrar dicha sentencia saldrá del bloque que está ejecutando.

La sentencia `continue` solamente se utiliza en estructuras de repetición y lo que hace es terminar la iteración i y continuar por la iteración $i+1$.