Sentencia if en Java: Controla la lógica en nuestro programa

Palabras clave : *sentencia if, sentencias condicionales, booleano, diagrama de flujo*

Este artículo abordará la sentencia if en Java. La sentencia if es una sentencia condicional que nos permite controlar la lógica de nuestros programas. Dado que toda la programación se basa en condiciones, la sentencia if puede considerarse uno de los pilares de la programación.

¿Qué es la declaración if en Java?

Una sentencia if en Java determina si el programa debe realizar una o varias operaciones, dependiendo de si se cumplen o no una o más condiciones predefinidas. La condición especificada es una expresión lógica que puede ser verdadera o falsa. En otras palabras, usamos una expresión lógica para decidir si nuestros programas deben ejecutar cierto código o no. Es decir, mediante la sentencia if, se puede especificar qué debe suceder si se cumple o no una condición. Esto crea la programación condicional, que es la base de toda la lógica de un programa.

En resumen, la declaración if en Java es:https://code-knowledge.com/wp-content/uploads/2021/02/If-statements-Java-programming.svg

- Una declaración condicional que determina lo que sucederá en el programa dependiendo de sus condiciones.
- Tiene una condición que siempre da como resultado una variable booleana (verdadero o falso).
- Constituye la base de la lógica de nuestros programas.

La declaración if en Java determina si se debe realizar una operación o no, dependiendo de la condición

¿Cómo funciona la declaración if en Java?

Intentemos ilustrar una sentencia if con un diagrama de flujo. Si no sabes qué es un diagrama de flujo o necesitas refrescar la memoria, puedes leer más sobre diagramas de flujo en el artículo sobre algoritmos .

La sentencia
`for` es una estructura de control de programación que repite un bloque de código un número específico de veces. Se usa comúnmente para iterar sobre colecciones como listas o para ejecutar código un número determinado de veces, y su sintaxis básica incluye una inicialización, una condición y un incremento/decremento, separados por punto y coma.
This video explains what a for loop is and how to use it:

```
for (let i = 0; i < 5; i++) {
  console.log("Iteración número: " + i);
}
frutas = ["manzana", "banana", "cereza"]
```

```
for fruta in frutas:
  print(fruta)
```

Sentencia FOR

Utilice la sentencia FOR para crear un bucle de programa FOR...NEXT. Un bucle de programa es una serie de sentencias que se ejecutan repetidamente hasta que se ha llevado a cabo el número de repeticiones especificado o hasta que se cumplen las condiciones especificadas.

Se asigna a *variable* el valor de *inicio*, que es el valor inicial del contador. En *fin* se especifica el valor final del contador.

Las *sentencias.bucle* que siguen a la cláusula FOR se ejecutan hasta que se encuentra la sentencia NEXT. El contador se ajusta conforme al valor especificado por la cláusula STEP.

Llegado este punto, se realiza una comprobación del valor del contador. Si es igual o menor que *fin*, la ejecución del programa se ramifica a la sentencia que sigue a la cláusula FOR y el proceso se repite. Si es mayor que *fin*, la ejecución continúa con la sentencia que figura a continuación de la sentencia NEXT.

La condición WHILE indica que mientras la expresión WHILE se evalúe como verdadera, seguirá ejecutándose el bucle. Cuando la expresión WHILE se evalúe como falsa, el bucle finalizará y la ejecución del programa continuará con la sentencia que figura tras la sentencia NEXT. Si una expresión WHILE o UNTIL se evalúa como valor nulo, la condición es falsa.

Las sentencias

`SELECT`, `CASE` y `SWITCH` (o `SEGUN`) son estructuras de control que permiten ejecutar diferentes bloques de código basándose en el valor de una variable o expresión, funcionando como alternativas a múltiples sentencias `if-else` anidadas. `SELECT CASE` y `SWITCH` son más comunes en la programación y se utilizan para evaluar un valor y ejecutar el código de un "caso" coincidente, mientras que `SELECT` con `CASE` es una sentencia SQL que se usa para crear nuevas columnas o categorizar datos en una consulta

Las sentencias
SELECT, CASE y SWITCH (o SEGUN) son estructuras de control que permiten ejecutar diferentes bloques de código basándose en el valor de una variable o expresión, funcionando como alternativas a múltiples sentencias if-else anidadas. SELECT CASE y SWITCH son más comunes en la programación y se utilizan para evaluar un valor y ejecutar el código de un "caso" coincidente, mientras que SELECT con CASE es una sentencia SQL que se usa para crear nuevas columnas o categorizar datos en una consulta.

```
' https://excelmacromastery.com/
Sub público  Select_Case_Example()

    ' Leer el valor de la celda A1
en la hoja1
    Dim airportCode As  String
    airportCode = Sheet1.Range( "A1"
).Value

    ' Imprimir el nombre del
aeropuerto en la ventana inmediata
(Ctrl + G)
    Seleccionar  Caso airportCode
        Caso  "LHR"
```

```
            Debug.Print  "London
Heathrow"
        Caso  "JFK"
            Debug.Print  "John F
Kennedy"
        Caso  "SIN"
            Debug.Print  "Singapore"
    Fin  Seleccionar

Fin de  subtítulo
' https://excelmacromastery.com/
Sub público  If_Example()

    ' Leer el valor de la celda A1
en la hoja1
    Dim airportCode As  String
    airportCode = Sheet1.Range( "A1"
).Value

    ' Imprimir el nombre del
aeropuerto en la ventana inmediata
(Ctrl + G)
    Si airportCode = "LHR"  Entonces
            Debug.Print  "London
Heathrow"
    De lo contrario Si airportCode =
"JFK"  Entonces
            Debug.Print  "John F
Kennedy"
```

```
    De lo contrario Si airportCode =
"SIN"  Entonces
            Debug.Print  "Singapore"
    Fin  Si

Fin de  subtítulo
```

La sentencia `do-while` (o `repetir-mientras`) es una estructura de control de flujo que repite un bloque de código al menos una vez antes de evaluar una condición. Se ejecuta el código dentro del bloque `do` y luego se revisa la condición `while`; si es verdadera, se repite; si es falsa, el bucle termina. Esta estructura es útil cuando es necesario que una acción se realice como mínimo una vez, independientemente de si la condición inicial se cumple o no.

No hay que confundir los rombos de las estructuras condicionales con los de las estructuras repetitivas do while.

En este problema por lo menos se carga un valor. Si se carga un valor mayor o igual a 100 se trata de un número de tres cifras, si es mayor o igual a 10 se trata de un valor de dos dígitos, en caso contrario se trata de un valor de un dígito. Este bloque se repite hasta que se ingresa en la variable valor el número 0 con lo

que la condición de la estructura do while retorna falso y sale del bloque repetitivo finalizando el programa.

La sentencia `while` (mientras) es una estructura de control de programación que repite un bloque de código mientras una condición sea verdadera. Evalúa la condición al principio de cada iteración y, si es falsa desde el inicio, el bloque de código nunca se ejecuta. Es útil cuando el número de repeticiones no es fijo, pero se conoce la condición que debe cumplirse para que el bucle termine.

Funcionamiento

- **Evaluación de la condición:** Se evalúa una expresión booleana (verdadera o falsa) al principio de cada repetición.
- **Ejecución del código:** Si la condición es verdadera, se ejecuta el bloque de código dentro del bucle.
- **Repetición:** El proceso se repite, volviendo a evaluar la condición.
- **Terminación:** El bucle finaliza cuando la condición se vuelve falsa.