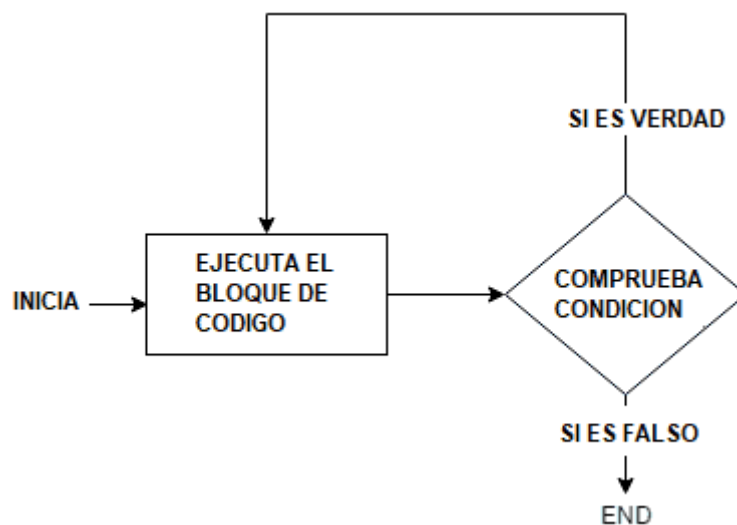


CICLO DO WHILE

Esquema while





- El bucle do while comienza con la ejecución de la(s) declaración(es). No hay verificación de ninguna condición la primera vez.
- Después de la ejecución de los enunciados, y la actualización del valor de la variable, la condición se verifica para el valor verdadero o falso. Si se evalúa como verdadero, comienza la siguiente iteración del ciclo.
- Cuando la condición se vuelve falsa, el ciclo finaliza y marca el final de su ciclo de vida.
- Es importante tener en cuenta que el bucle do-while ejecutará sus declaraciones al menos una vez antes de que se verifique cualquier condición, y por lo tanto es un ejemplo de bucle de control de salida.

Sintaxis

do

sentencia

while (condición);

sentencia

Una sentencia que se ejecuta al menos una vez y es re ejecutada cada vez que la condición se evalúa a verdadera. Para ejecutar múltiples sentencias dentro de un bucle, utilice la sentencia block ({ ... }) para agrupar aquellas sentencias.

condición

Una expresión se evalúa después de cada pase del bucle. Si condición se evalúa como verdadera, la sentencia se re-ejecuta. Cuando condición se evalúa como falsa, el control pasa a la siguiente sentencia hacer mientras.

Ejemplo:

```
// Programa Java para ilustrar el bucle do-while
```

```
class dowhileloopDemo
```

```
{
```

```
    public static void main(String args[])
```

```
    {
```

```
        int x = 21;
```

```
do
{
    //El código dentro del do se imprime incluso
    //si la condición es falsa
    System.out.println("Valor de x :" + x);
    x++;
}
while (x < 20);
}
```

Salida:

Valor de x: 21

Uso del valor false explícito o de variables finales

En el bucle while no puedes utilizar como condición booleana:

- El valor false explícito, ya que eso significa que el código que contenga el bucle es inalcanzable (unreachable code).
- Una variable booleana, o un objeto de tipo Boolean, marcado como final y cuyo valor sea false. Lo importante en este caso es que la variable o el objeto Boolean están marcados como final. Si no lo están, el compilador no ve el valor que contienen (o al que apuntan, en caso de ser un objeto). Una variable u objeto final es inmutable, y el compilador sí tiene en cuenta su valor al realizar su tarea. En este caso, pues, si la variable es final y su valor es false, el código que contiene el bucle es, de nuevo, inalcanzable.