Estructuras de repetición (Bucles)

SENTENCIAS REPETITIVAS O BUCLES

Los bucles, iteraciones o sentencias repetitivas modifican el flujo secuencial de un programa permitiendo la ejecución reiterada de una sentencia o bloque de sentencias. En Java hay tres tipos diferentes de bucles: for, while y do-while.

Todos ellos tienen algo en común, **se ejecutan reiteradamente mientras la condición sea TRUE**, y **dejan de ejecutarse cuando la condición pasa a ser FALSE**.

SENTENCIA FOR

La sentencia for tiene la característica de que tiene bien definido el inicio del bloque, la evaluación de la expresión, el incremento de valor y el bloque de sentencias.

La estructura del bucle for es:

for (inicialización; condición; incremento)

sentencia;

O si se desean repetir varias sentencias irán englobadas entre llaves. Las llaves sólo son necesarias si se quieren repetir varias sentencias, aunque se recomienda su uso porque facilita la lectura del código fuente y ayuda a evitar errores al modificarlo:

for (inicialización; condición; incremento) {

bloque\_sentencias;

}

Tanto la inicialización, como incremento son opcionales y pueden estar o no, aunque normalmente aparecerán en la estructura.

Esta estructura la podríamos reproducir mediante una sentencia while de la siguiente forma:

inicialización;

while (condición) {

bloque\_sentencias;

incremento;

}

Las funcionalidades en las que utilizaremos la sentencia for serán las mismas que las sentencias while y do-while, que serán contadores, recorrer estructuras,…

Por ejemplo, si queremos definir un contador de 1 a 10 mediante una sentencia for utilizaremos el siguiente código:

for (int x=1;x<=10;x++=) {

System.out.println("Valor del contador: " + x);

}

En pantalla obtendremos el siguiente resultado:

Valor del contador: 1

Valor del contador: 2

Valor del contador: 3

Valor del contador: 4

Valor del contador: 5

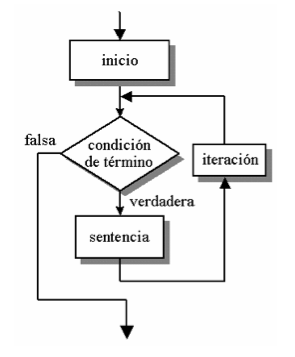
Valor del contador: 6

Valor del contador: 7

Valor del contador: 8

Valor del contador: 9

Valor del contador: 10



Es completamente legal en Java declarar una variable dentro de la cabecera de un bucle for. De esta forma la variable (local) sólo tiene ámbito dentro del bucle.

SENTENCIA WHILE

La estructura repetitiva while realiza una primera evaluación antes de ejecutar el bloque de sentencias. Si la expresión es true pasa a ejecutar el bloque de sentencias.

Cada vez que termina de ejecutar el bloque de sentencias vuelve a evaluar la expresión. Si la expresión sigue siendo true vuelve a ejecutar el bloque. En el caso de que la expresión sea false se saldrá del bucle.

Es por ello que dentro del bloque de sentencias deberán de existir sentencias que modifiquen la evaluación de la expresión, ya que de no hacerse se podría entrar en un bucle infinito.

La estructura de la sentencia while es la siguiente:

while (expresion) {

bloque\_sentencias;

}

Por ejemplo podemos realizar un contador de 1 a 10 de la siguiente forma:

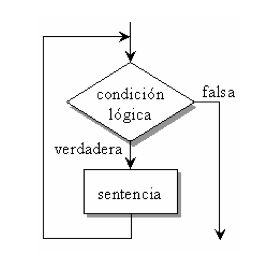
int contador = 1;

while (contador <= 10) {

System.out.println(contador);

contador++;

}



Cualquier bucle while puede transformarse en un bucle for, y viceversa.

SENTENCIA DO-WHILE

En el caso de la estructura repetitiva do-while el funcionamiento es el mismo que el de while. Pero con una diferencia, primero se ejecuta el bloque de sentencias y luego se evalúa la expresión. Por lo tanto siempre se ejecutará, al menos una vez, el bloque de sentencias.

La estructura de la sentencia do-while es:

do {

bloque\_sentencias;

} while (expresion);

Al igual que anteriormente, en el bloque de sentencias deberemos de modificar alguna de las condiciones de la expresión para poder salir del bucle.

Un ejemplo del bucle do-while sería el ejemplo en el que le pedimos al usuario que introduzca números por teclado, los cuales mostraremos en forma de eco por pantalla, hasta que introduzca el cero. En ese caso saldremos del bucle.

Utilizaremos la estructura do-while en vez de la while ya que al menos vamos a pedirle al usuario el primer número.

El código sería el siguiente:

Scanner reader = new Scanner(System.in);

int iNumero = 0;

do {

System.out.println("Introduce carácter por consola");

iNumero = reader.nextInt();

System.out.println(iNumero);

} while (iNumero <> 0);

En el caso de haberlo realizado con un bucle while tendríamos que cambiar la forma de codificarlo, por ejemplo inicializando iNumero a un número que no sea 0, o bien repetiendo la captura y salida de datos (ejemplo a continuación):

Scanner reader = new Scanner(System.in);

int iNumero = 0;

System.out.println("Introduce carácter por consola");

iNumero = reader.nextInt();

System.out.println(iNumero);

while (iNumero <> 0) {

System.out.println("Introduce carácter por consola");

iNumero = reader.nextInt();

System.out.println(iNumero);

}

