

4.C. Estructuras de repetición.

✓ Hecho

1. Estructuras de repetición.

1.1. Estructura for.

Hemos indicado anteriormente que el bucle `for` es un bucle controlado por contador. Este tipo de bucle tiene las siguientes características:

- Se ejecuta un número determinado de veces.
- Utiliza una variable contadora que controla las iteraciones del bucle.



Imagen extraída de curso Programación del MECD.

En general, existen tres operaciones que se llevan a cabo en este tipo de bucles:

- Se inicializa la variable contadora.
- Se evalúa el valor de la variable contador, por medio de una comparación de su valor con el número de iteraciones especificado.
- Se modifica o actualiza el valor del contador a través de incrementos o decrementos de éste, en cada una de las iteraciones.

Recomendación

La inicialización de la variable contadora debe realizarse correctamente para garantizar que el bucle se lleve a cabo, al menos, la primera repetición de su código interno.

La condición de terminación del bucle debe variar en el interior del mismo, de no ser así, podemos caer en la creación de un bucle infinito.

Cuestión que se debe evitar por todos los medios.

Es necesario estudiar el número de veces que se repite el bucle, pues debe ajustarse al número de veces estipulado.

En la siguiente tabla, podemos ver la especificación de la estructura `for`:

Estructura repetitiva <code>for</code>	
Sintaxis:	<p>Donde inicialización es una expresión en la que se inicializa una variable de control, que será la encargada de controlar el final del bucle.</p> <p>Donde condición es una expresión que evaluará la variable de control. Mientras la condición sea falsa, el cuerpo del bucle estará repitiéndose. Cuando la condición se cumpla, terminará la ejecución del bucle.</p> <p>Donde iteración indica la manera en la que la variable de control va cambiando en cada iteración del bucle. Podrá ser mediante incremento o decremento, y no solo de uno en uno.</p>
<code>for (inicialización;</code>	
<code>condición; iteración)</code>	
<code></code>	
<code>sentencia;</code>	
<code></code>	
<code>(estructura <code>for</code> con una única</code>	
<code>sentencia)</code>	
Sintaxis:	
<code>for (inicialización;</code>	
<code>condición; iteración)</code>	
<code>{</code>	
<code></code>	
<code>sentencia1;</code>	
<code></code>	
<code>sentencia2;</code>	
<code></code>	
<code>...</code>	
<code></code>	
<code>sentenciaN;</code>	
<code>}</code>	
<code>(estructura <code>for</code> con un bloque</code>	
<code>de sentencias)</code>	

Debes conocer

Como venimos haciendo para el resto de estructuras, [visualiza el siguiente programa Java](#) y podrás analizar un ejemplo de utilización del bucle `for` para la impresión por pantalla de la tabla de multiplicar del siete. Lee atentamente los comentarios incluidos en el código, pues aclaran algunas cuestiones interesantes sobre este bucle.

```
public class repetitiva_for {  
  
    /* En este ejemplo se utiliza la estructura repetitiva for  
  
    * para representar en pantalla la tabla de multiplicar del siete  
  
    */  
  
    public static void main(String[] args) {  
  
        // Declaración e inicialización de variables  
  
        int numero = 7;  
  
        int contador;  
  
        int resultado=0;  
  
  
        //Salida de información  
  
        System.out.println ("Tabla de multiplicar del " + numero);  
  
        System.out.println ("..... ");  
    }  
}
```

```

//Utilizamos ahora el bucle for
for (contador=1; contador<=10;contador++)

/* La cabecera del bucle incorpora la inicialización de la variable
 * de control, la condición de multiplicación hasta el 10 y el
 * incremento de dicha variable de uno en uno en cada iteración del
 * bucle.
 * En este caso contador++ incrementará en una unidad el valor de
 * dicha variable.
 */

{
    resultado = contador * numero;

    System.out.println(numero + " x " + contador + " = " + resultado);

    /* A través del operador + aplicado a cadenas de caracteres,
     * concatenamos los valores de las variables con las cadenas de
     * caracteres que necesitamos para representar correctamente la
     * salida de cada multiplicación.
     */
}
}
}

```

Autoevaluación

Cuando construimos la cabecera de un bucle for, podemos prescindir de alguno de los tres elementos que la forman e incluso, podemos utilizar más de una variable contadora separando éstas por comas. Pero, ¿Qué conseguiremos si construimos un bucle de la siguiente forma?

```
for (;;) { //instrucciones }
```

- ☒ Un bucle infinito.
- ☐ Nada, dará un error.
- ☐ Un bucle que se ejecutaría una única vez.

◀ 4.B. Estructuras de selección.

Ir a...

4.D. Estructuras de salto. ▶