

# **BUCLE FOR**

## ESTRUCTURAS DE CONTROL

# ESTRUCTURAS DE CONTROL REPETITIVAS

- También conocidas como **bucles**.
- Nos permiten **repetir** un **bloque** de **sentencias** varias veces.
- Si el número de repeticiones (**iteraciones**) se conoce **a priori**, podemos usar el bucle **for**.

# BUCLE FOR

- **Sintaxis**

```
for (declaración/inicialización; condición; incremento/decremento) {  
    // acciones a ejecutar  
}
```

- **declaración/inicialización:**

- Se ejecuta una única vez, al inicio del bucle.
- Sirve para declarar y/o inicializar una (o más) variable que nos servirá para gestionar el bucle.
- Si se declara la variable aquí, no tiene vida más allá del bucle.

# BUCLE FOR

- **Sintaxis**

```
for (declaración/inicialización; condición; incremento/decremento) {  
    // acciones a ejecutar  
}
```

- **condición:**

- Es una **expresión booleana**.
- Sirve para indicar si debemos continuar en el bucle o salir de él.
- **Condición de continuación:** si es true, se vuelve a realizar una iteración más del bucle.

# BUCLE FOR

- **Sintaxis**

```
for (declaración/inicialización; condición; incremento/decremento) {  
    // acciones a ejecutar  
}
```

- **incremento/decremento:**

- Se ejecuta para cada iteración, después de ejecutar las acciones del cuerpo del bucle.
- Sirve para incrementar o decrementar las variables de control (declaradas o inicializadas en el propio bucle).

# BUCLE FOR

- Ejemplo: imprimir por pantalla los 5 primeros números enteros, comenzando en el 1.

```
for(int i = 1; i <= 5; i++) {  
    System.out.println(i);  
}
```

- Si después tratamos de acceder a la variable *i* tenemos un error, ya que dicha variable se destruye después de terminar el bucle.

# BUCLE FOR

- Si el cuerpo del bucle incluye una única sentencia, podemos ahorrarnos las llaves.

```
for(int i = 1; i <= 5; i++)  
    System.out.println(i);
```

# BUCLE FOR

- Se puede decrementar la variable de control, o incrementar en más de una unidad.

```
for(int i = 10; i >= 0; i--) {  
    System.out.println(i);  
}
```

```
for(int i = 0; i <= 100; i+=2)  
    System.out.println(i);
```



# BUCLES FOR

- Podemos utilizar variables declaradas fuera del bucle.
- Ejemplo: sumatorio de los 100 primeros números naturales.

```
int resultado = 0;
for(int i = 1; i <= 100; i++) {
    resultado += i;
}
System.out.println("El resultado es " + resultado);
```

# BUCLE INFINITO

- Un bucle del que no se sale nunca.
- Normalmente es un comportamiento erróneo.
- Revisar bien la cabecera del bucle: ¿Se incumple alguna vez la condición de continuación?

```
for(int i = 1; i > 0; i++)  
    System.out.println(i);
```

# BREAK Y CONTINUE

- Dos instrucciones que nos sirven para salir del bucle o saltar una iteración.
- Su uso no es muy recomendable.
  - Difícil de depurar errores.
- Es preferible articular el bucle o su organización de otra forma.

# BREAK

- Nos permite salir del bucle “en cualquier momento”.
- Suele ir asociado a un bloque if.
- *Ejemplo: identificar si un número es primo, buscando sus divisores.*

# CONTINUE

- Nos permite terminar una iteración sin finalizar todas sus sentencias, y continuar con la siguiente iteración.
- *Ejemplo: sumar todos los números del 1 al 10, menos el 7.*

```
int suma = 0;
for (int i = 1; i <= 10; i++) {
    if (i == 7)
        continue;
    suma += i;
}
```