


Open · Opened 2 weeks ago by  **Rubén Montero**

La sentencia return es como un hacha

Resumen

- Veremos ejemplos de *Unreachable statement*
- Repasaremos brevemente la sintaxis del bucle `for` simple
- Escribiremos un método en `SimpleMathDemo` que tiene un `return` dentro de un bucle
- Veremos cómo se comporta invocándolo desde `Main`

Descripción

En los métodos que hemos escrito hasta ahora, la sentencia `return` devuelve un valor.

Todavía no hemos comentado que, la sentencia `return` devuelve un valor **y** termina la ejecución del método.

Por ejemplo:

```
private String greeting() {  
    String string1 = "hola, ";  
    String string2 = "amigo";  
    return string1 + string2;  
    // La siguiente línea produce un error Unreachable Statement  
    System.out.println("Este print va despues");  
}
```

Se conoce como *Dead code*, *Unreachable code*, código muerto o inalcanzable a aquel que **nunca puede ejecutarse**.

¿Y si ponemos `return` en un bucle?

Pequeño repaso: Bucles `for`

Recordemos que un bucle `for` **sencillo** (también hay bucles *for-each*) tiene la siguiente sintaxis:

```
for (int i = 0; i < 10; i++) {  
  
}
```

- `i` es la variable *índice* que se va actualizando dentro del bucle
- `i < 10` es la *condición de terminación*. Si es verdadera, el bucle se ejecuta. En el momento en que pasa a ser **falsa**, el bucle no se repite
- `i++` es el modo de incremento del índice. Podría ser `i+=2`, por ejemplo

Completando el repaso

Se pide que:

- Añadas un nuevo método a `SimpleMathDemo`
- Tendrá este código:

```
public void printMultiplication() {  
    System.out.println("Los múltiplos de 7 menores que 70 son:");  
    for (int i = 0; i < 70; i+=7) {  
        System.out.println(i);  
    }  
}
```

A lo nuestro

Continuemos.

- ¿Y si ponemos `return` en un bucle?

Vamos a añadir este **otro** método a `SimpleMathDemo` :

```
public void loop50Times() {  
    for (int i = 0; i < 50; i++) {  
        System.out.println("Imprimiendo... " + i);  
        if (i == 38) {  
            return; // Se puede hacer return si el método devuelve  
                    // void. Tiene el mismo efecto...  
                    // ¡Pero no devuelve nada!  
        }  
    }  
}
```

Invócalo también desde `Main.java` .

¿Qué se imprime por consola?

Por último

Una vez verifiques que el *test* funciona correctamente, la tarea ha sido completada.

Haz `commit` y `push` para subir los cambios al repositorio.



Rubén Montero @ruben.montero changed milestone to [%Sprint 1](#) 2 weeks ago



Ania Blanco @ania.blanco mentioned in commit [13020cb6](#) 2 weeks ago