

- ¿Qué es un error sintáctico?

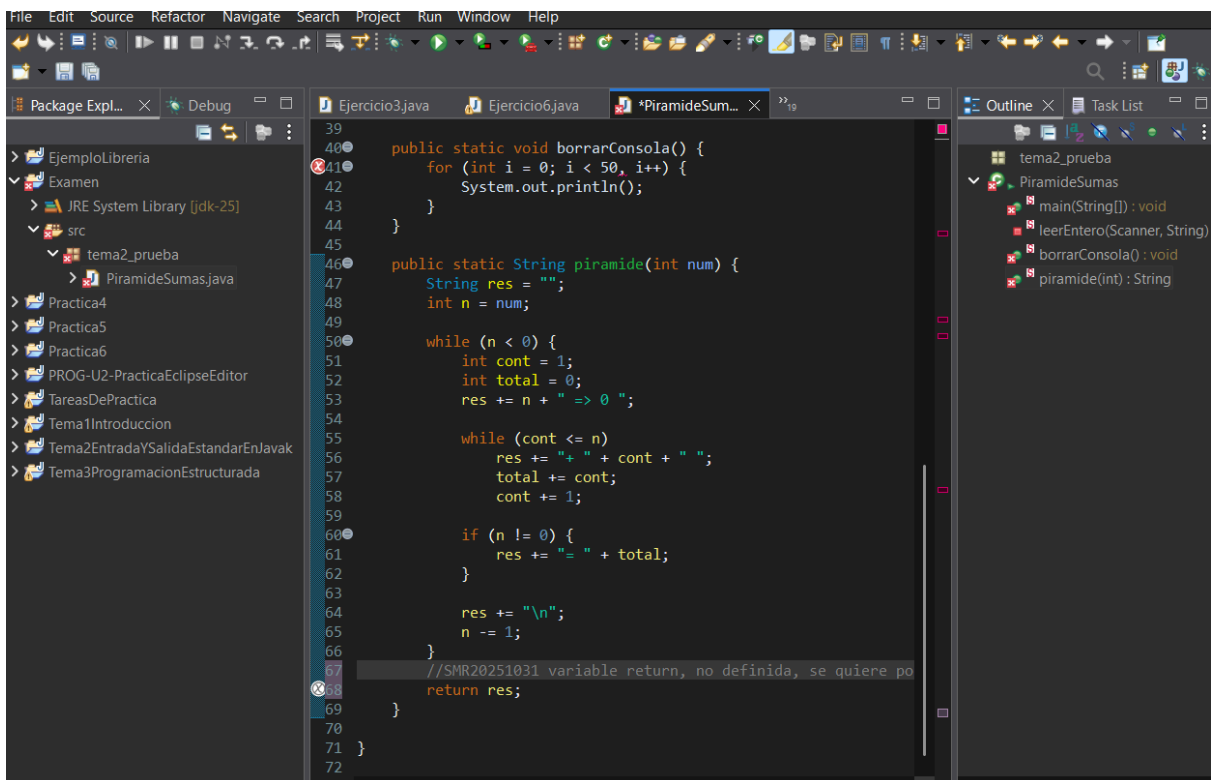
Un error sintáctico es cuando una función o variable no está bien definida o cerrada, causando errores que no dejan iniciar el programa adecuadamente, el intérprete no entiende lo que hemos escrito. Este tipo de errores suelen ser olvidar poner (;) , olvidar poner paréntesis y/o llaves o ponerlos de forma incorrecta.

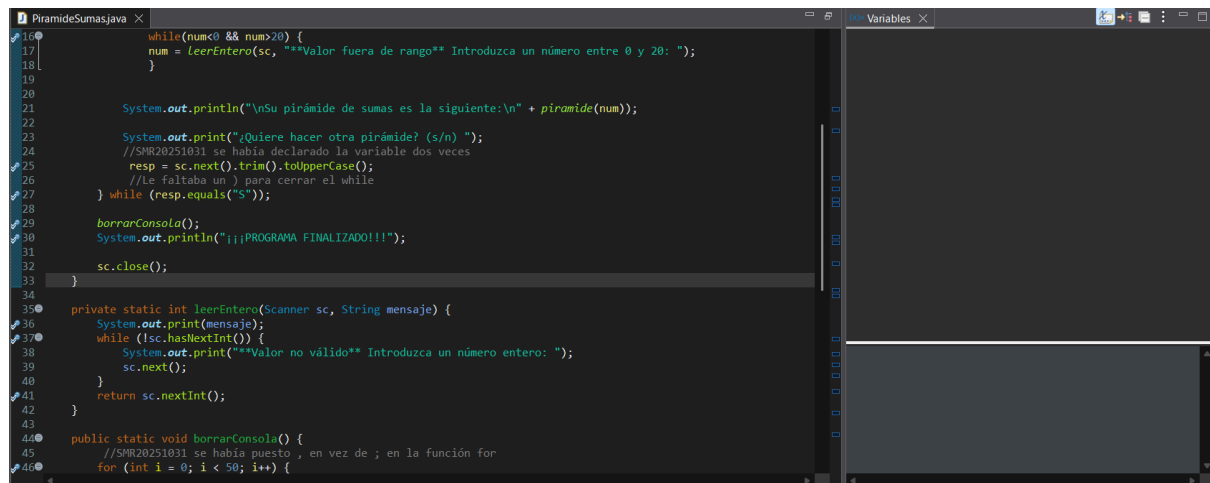
- ¿Cuándo los visualizamos?

Desde un IDE como eclipse nos muestra un mensaje de error antes de ejecutar el programa, indicando en qué línea y por qué da error.

- ¿Podemos depurar con errores de sintaxis?

No, cuando hay errores de sintaxis el programa no se ejecuta, por lo que la depuración al ser ejecutar el programa paso a paso, no nos deja depurar hasta solucionar esos errores.

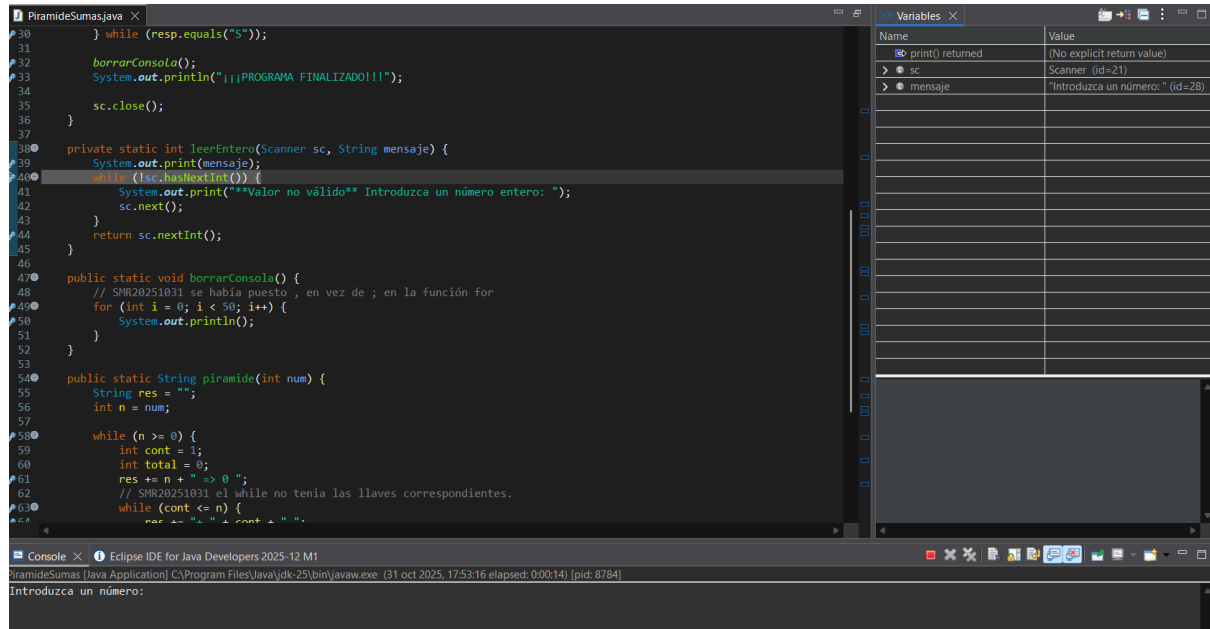




## Cómo depurar un programa:

Para depurar un programa necesitamos poner breakpoints en diferentes partes del código, las más importantes(condicionales, funciones, variables), ver el valor que van tomando las variables, si los condicionales se realizan correctamente, e ir probando el código de distintas formas.

## Breakpoints y vistas más importantes a la hora de depurar



Los breakpoints más importantes son los que se ponen donde las variables consiguen nuevos valores y en las funciones e inicios de bucles, además de poner breakpoints condicionales para que se paren solo cuando se cumpla la condición. Las vistas más importantes son la de variables y expressions, la primera para ver los valores que van adquiriendo las variables y la segunda para poder llevar otros cálculos además de los puestos en el código.

## ¿Dónde colocaste los breakpoints y qué valores viste que confirmaron el fallo?

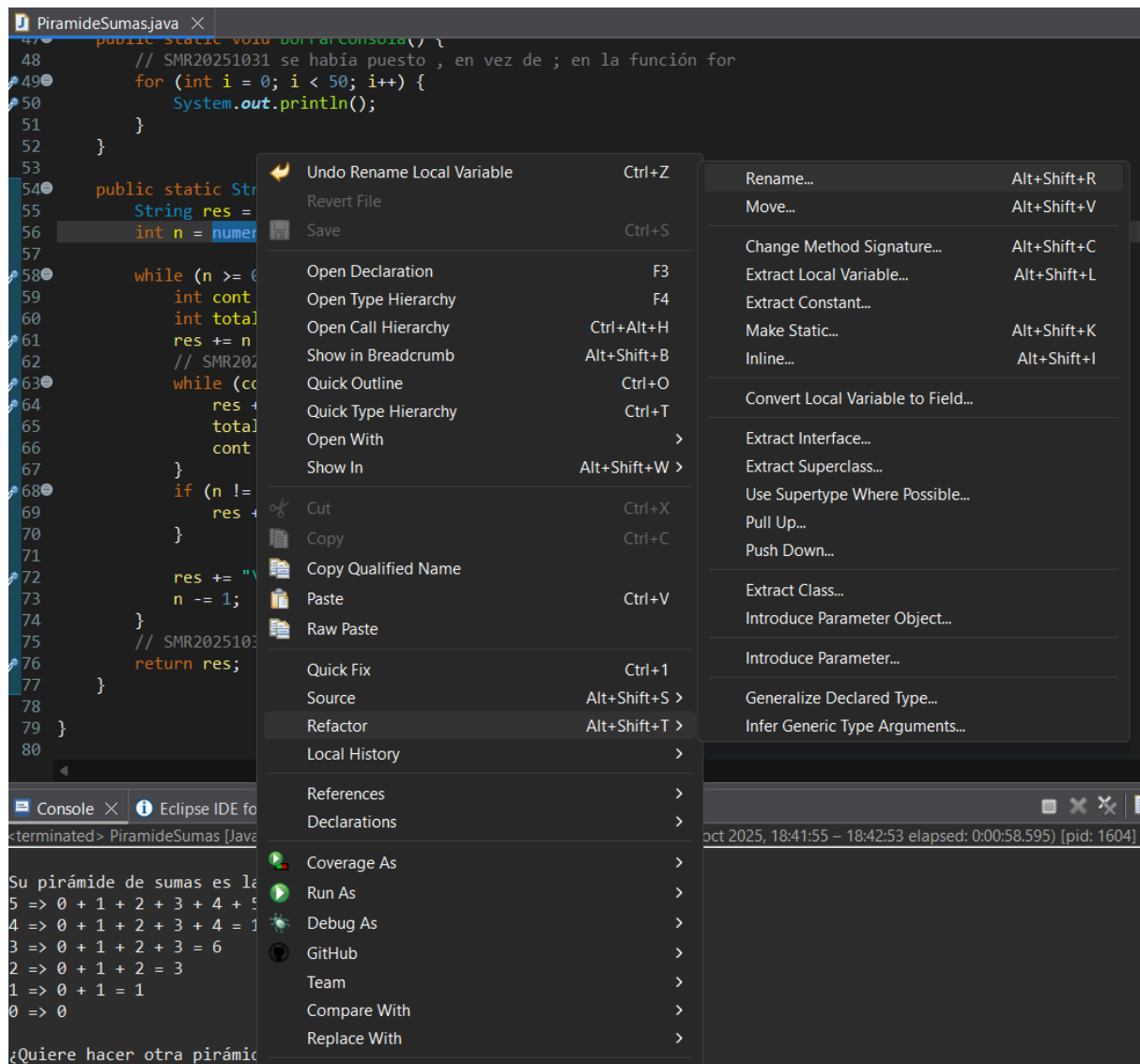
1. Breakpoint en el while del inicio, de valores fuera de rango, con el que al poner un valor fuera del rango 0-20 se inicia igualmente y te intenta hacer la pirámide.
2. Breakpoints en los while y en los if de la función piramide, que al no estar cerrado el while no paraba de sumar 1 hasta el infinito

## ¿En qué situación debemos usar Step Into (F5), Step Over (F6), Step Return (F7) y Resume (F8)?

- Step Into: Cuando queramos ver el código de una función que se llama en el método actual.
- Step Over: Cuando quieres avanzar a la siguiente línea o breakpoint
- Step Return: Cuando estás en una función y quieres salir de ella sin ir paso por paso.
- Resume: Cuando quieres que el programa siga hasta el siguiente breakpoint.

## Refactorización: renombrar `num` → `numero`

Hice clic derecho en la variable, usé la herramienta refactor, me fuí a su apartado de rename, y entonces pude cambiar el nombre de la variable en todo el código sin ir uno por uno, lo considero útil porque te ahorra tiempo y posibles errores.



Crear una plantilla (template) “borrar”

The screenshot shows an IDE window titled "PiramideSumas.java". The code includes a `Scanner` object, a `leerEntero` method for reading integers with validation, and a `borrarConsola` method. The `borrarConsola` method contains a `for` loop that prints empty lines. A popup menu is visible over the `println` call in the `for` loop, showing a template named "borrar" with the description "Crea saltos de línea simulando un borrado de consola, con n saltos de línea en numLineas". The popup also shows the method signature `borrar(): void - Method stub` and the code snippet `for (int i = 0; i < numLineas; i++) { System.out.println(); }`. At the bottom of the popup, there are instructions: "Press 'Ctrl+Space' to show Template Proposals" and "Press 'Tab' from proposal table or click/hover for focus".

```
34
35     sc.close();
36 }
37
38 private static int leerEntero(Scanner sc, String mensaje) {
39     System.out.print(mensaje);
40     while (!sc.hasNextInt()) {
41         System.out.print("**Valor no válido** Introduzca un número entero: ");
42         sc.next();
43     }
44     return sc.nextInt();
45 }
46
47 public static void borrarConsola() {
48     // SMR20251031 se había puesto , en vez de ; en la función for
49     for (int i = 0; i < 50; i++) {
50         System.out.println();
51     }
52 } borrar
53
54 public st
55 String
56 int n
57
58 while
59 i
60 i
61 r
62 /
63 w
64
65
66
67
```

borrar - Crea saltos de línea simulando un borrado de consola, con n saltos de línea en numLineas | borrarConsola() {  
for (int i = 0; i < numLineas; i++) {  
 System.out.println();  
}  
}

borrar(): void - Method stub

Press 'Ctrl+Space' to show Template Proposals

Press 'Tab' from proposal table or click/hover for focus

La plantilla sirve para ahorrar tiempo en hacer acciones repetitivas, por ejemplo, `System.out.println()`, lo vas a poner muchas veces en distintas ocasiones, por lo que al hacer una plantilla y llamarla `sysout`, reduces el tiempo que tardas en ponerlo y te puedes dedicar a partes más importantes.