

EXAMEN 2

APARTADO 2.2

- ¿Qué es un error sintáctico?

Error al escribir nuestro código fuente, es un tema de caracteres y no lógico.

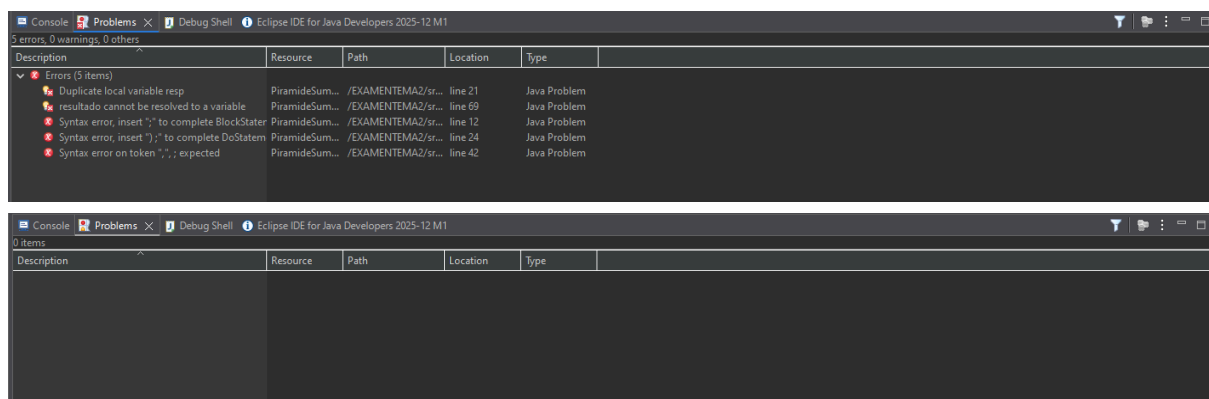
- ¿Cuándo los visualizamos?

Se ve en el propio IDE mientras escribimos.

- ¿Podemos depurar con errores de sintaxis?

Depurar no sirve de nada en estos casos ya que el error se ve antes de ejecutar.

AQUI DOS CAPTURAS CON LA VISTA PROBLEMS ANTES Y DESPUES



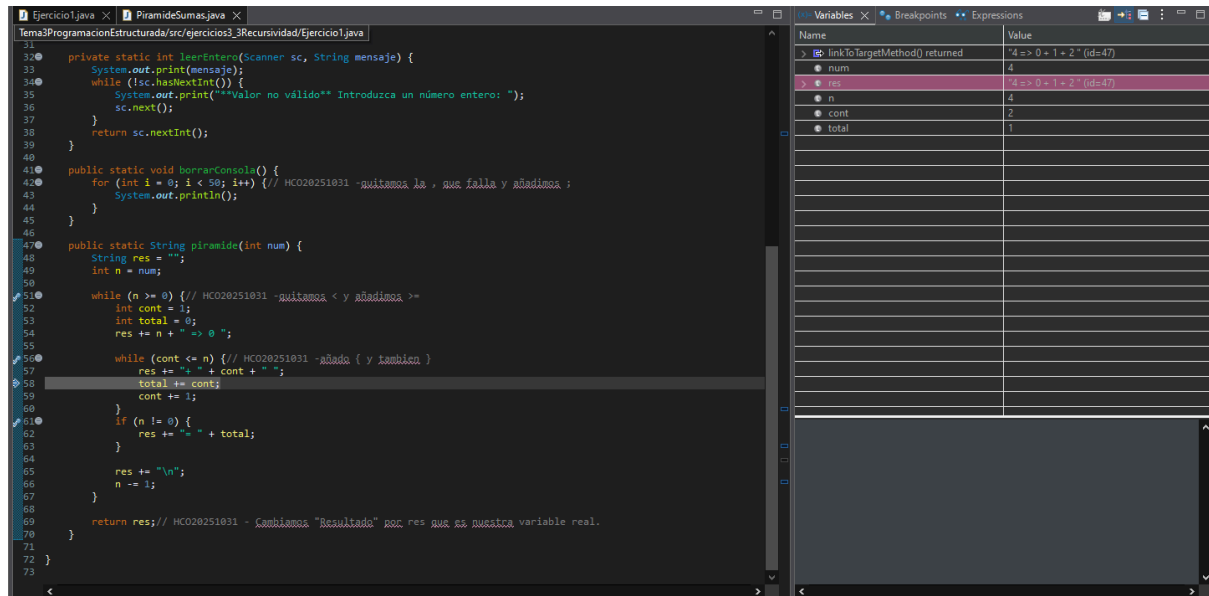
APARTADO 2.3

En cuanto a los errores que he corregido me doy cuenta al ejecutar que el rango que habíamos preestablecido no funciona y lo corrijo poniendo un breakpoint donde se establece el rango y dándome cuenta de que no entra nunca. Debíamos cambiar el AND por OR (no hay número que sea menor que 0 y mayor que 20).

El más obvio e importante es que la pirámide nunca salía por pantalla, así que busco el problema en la función dedicada a ello, me doy cuenta con los breakpoint de que los while no actúan. El primer while tiene mal puesta la condición, n no puede ser menor que 0. Al segundo while le añado los $\{ \}$, que limiten su actuación, y así que el total y el contador puedan desempeñar su función.

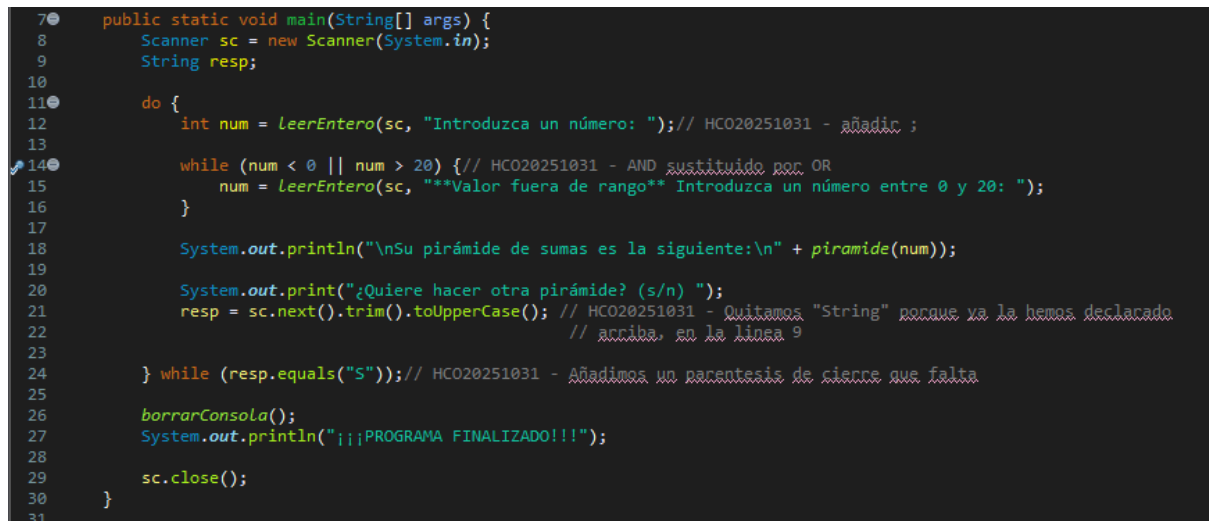
¿Cómo se depura un programa?

La depuración es una herramienta muy útil sobre todo cuando encontramos errores lógicos que no sabemos como solucionar con la ejecución de ordinaria del programa. ¿Cómo depuramos?, primero debemos intentar encontrar donde puede estar el fallo y colocar puntos de ruptura en ellos, estos harán que al depurar nuestro programa se pare y nos permita analizar y ver como van cambiando las variables y nuestras expresiones en directo. Asi nos será más fácil encontrar los errores que hacen que nuestro programa no funcione correctamente.



Puntos azules → breakpoints

Vista variables—> Nos permite ver su valor en ese momento exacto de la ejecución



Aquí coloque el primer breakpoint para ver el error de rango

```

47 public static String piramide(int num) {
48     String res = "";
49     int n = num;
50
51     while (n >= 0) { // HCO20251031 - quitamos < y añadimos >=
52         int cont = 1;
53         int total = 0;
54         res += n + " => 0 ";
55
56         while (cont <= n) { // HCO20251031 - añadido { y tambien }
57             res += " + " + cont + " ";
58             total += cont;
59             cont += 1;
60         }
61         if (n != 0) {
62             res += " = " + total;
63         }
64
65         res += "\n";
66         n -= 1;
67     }
68
69     return res; // HCO20251031 - Cambiamos "Resultado" por res que es nuestra variable real.
70 }
71
72 }
73

```

Aqui coloque los demas para ver el funcionamiento de los while

La vista mas importante me parece sin dudas la de las variables ya que suelen mostrarnos que esta no tiene el comportamiento que nosotros esperamos al pulsar F6/F8.

Los breakpoints por su parte los colocho en sitios clave como bucles o condicionales que suelen albergar los problemas de nuestro programa.