

DOCUMENTACIÓ DEBUG EXERCICIS

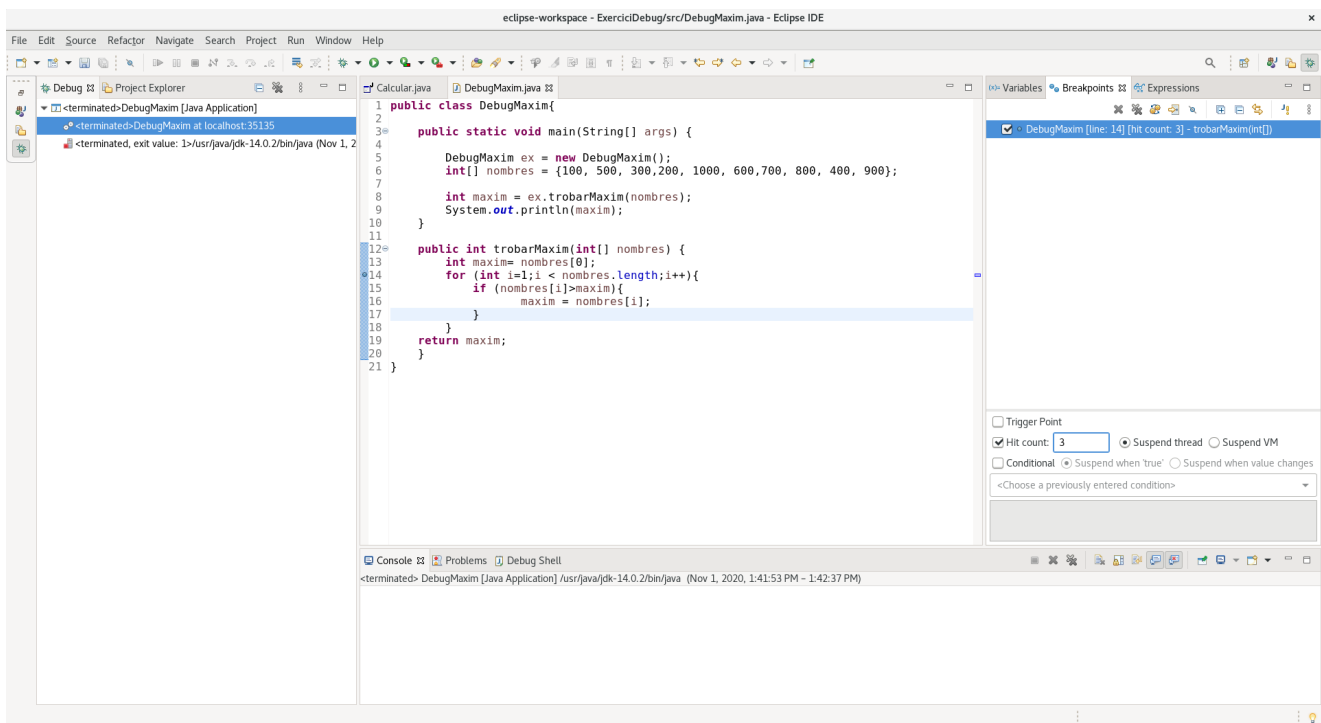
Sergi Muñoz Carmona
2n DAM

EXERCICI 1

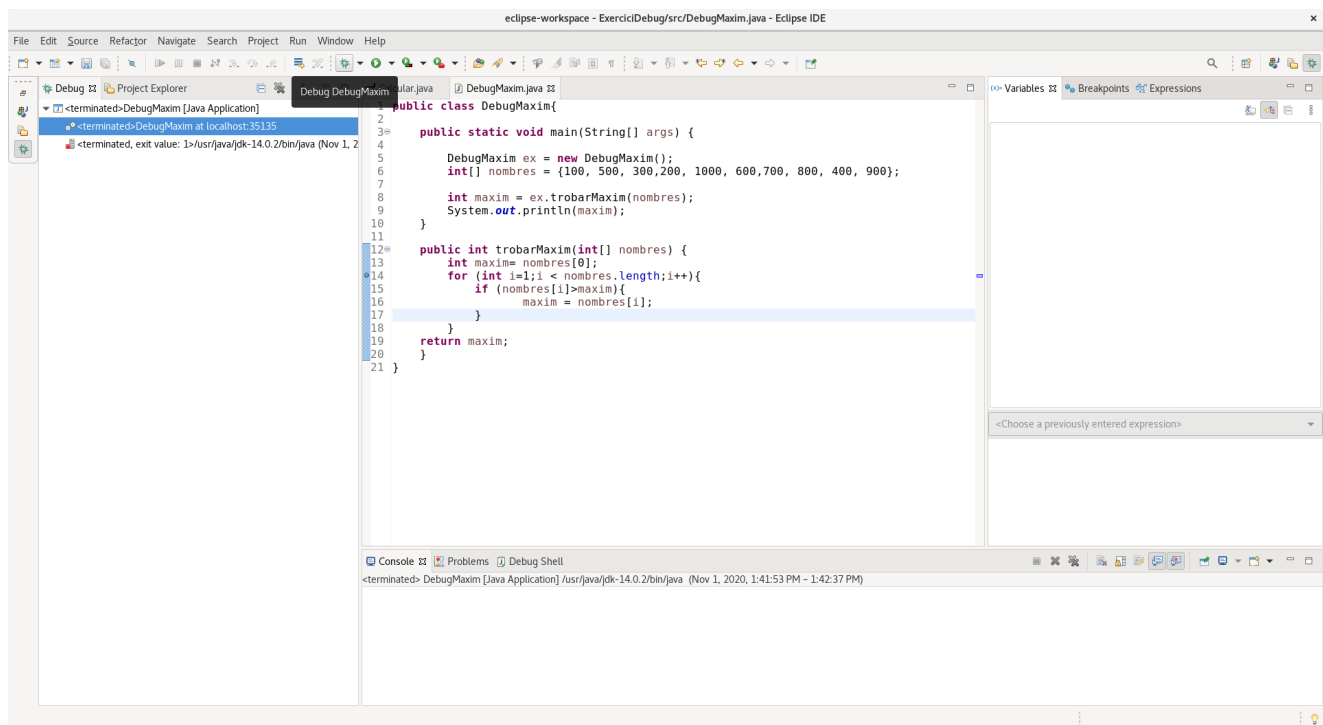
L'exercici ens demana introduir els punts de ruptura necessaris I mostrar els valors de “màxim” I “i” en la 3r interacció I el valor que conté el maxm abans del return.

He introduït un únic punt de ruptura assignat al bucle for, per tal de poder parar el bucle en la 3r interacció I mostrar els valors en aquell punt.

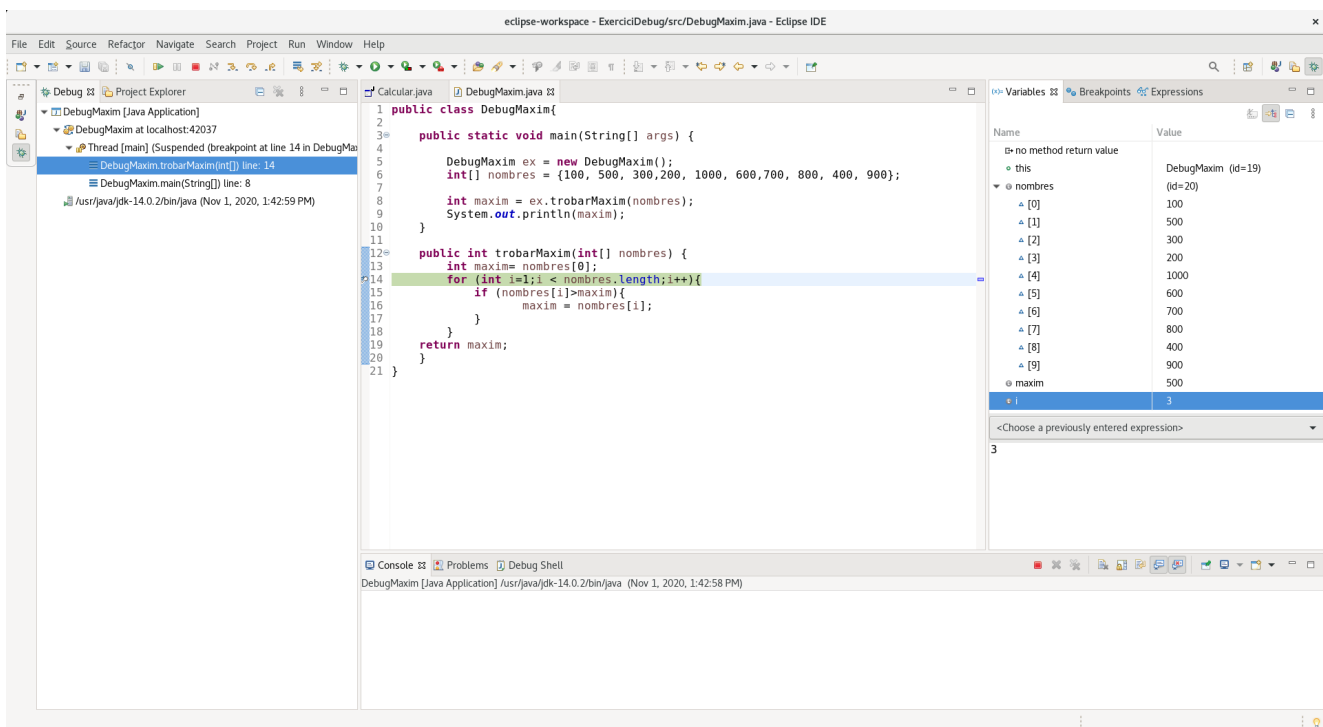
Per poder mostrar els valors en la 3r iteració, hem d'assignar l'opció “**Hit count**” I en el nostre cas posarem un **3**.



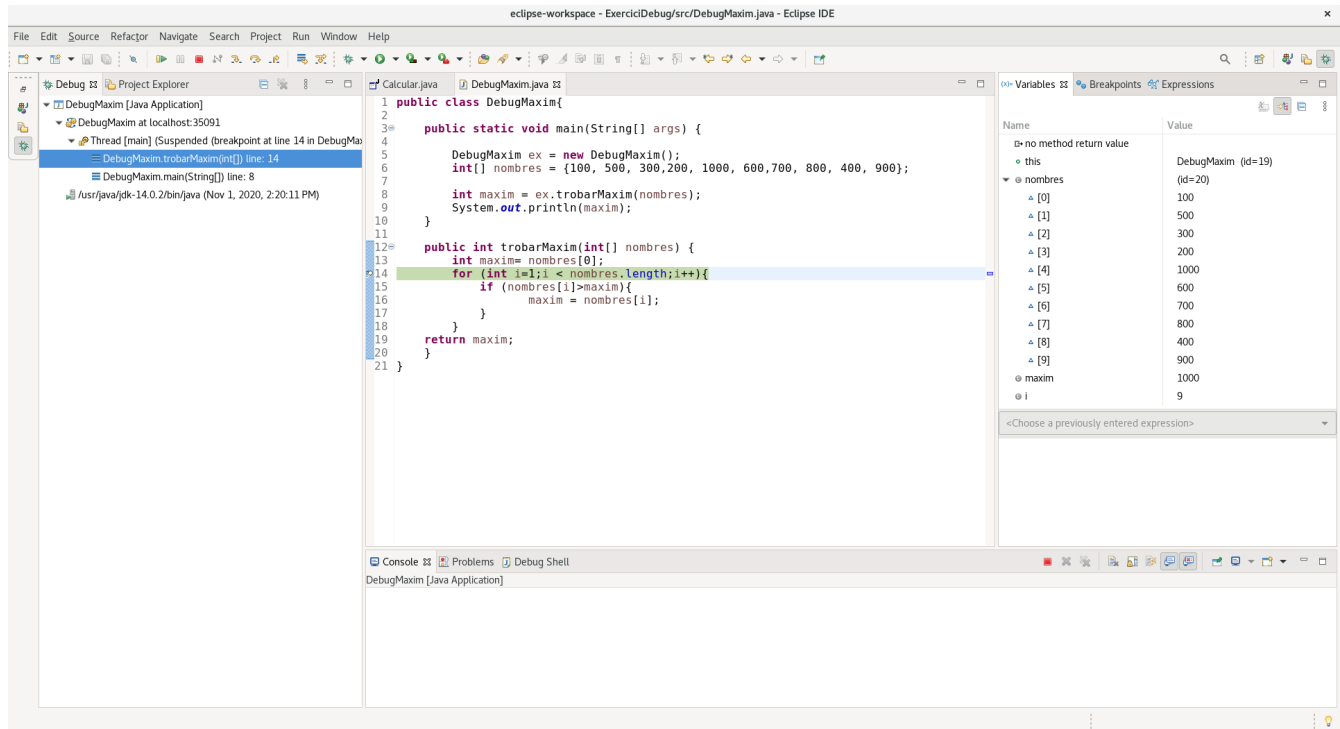
Inicialitzarem el procés de debug per veure els resultats I valors de “i” I “màxim”.



Veiem que els valors en la 3r iteració del bucle de “i” = 3 I de “màxim” = 500.



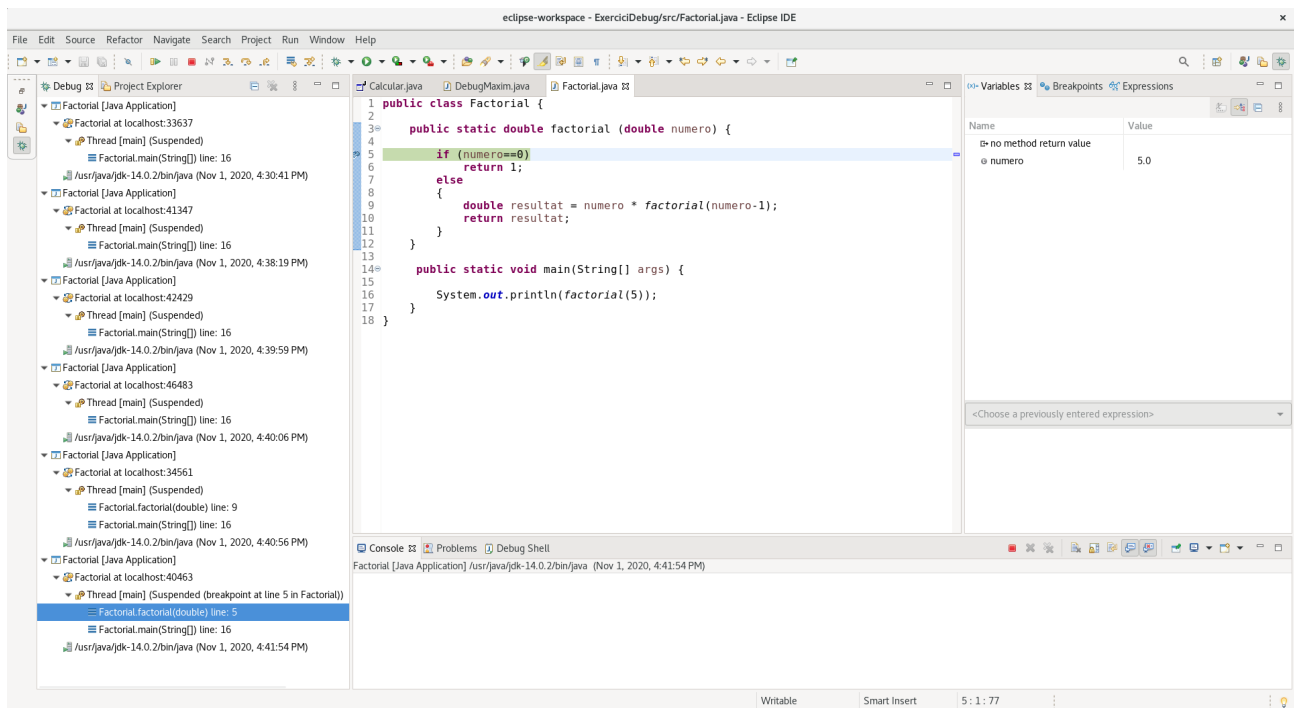
Per veure l'últim valor que té la variable “maxim” abans del return, anirem a l'última interacció del bucle for. L'última interacció és la 9 (**valor de I = 9**), on el valor de “**maxim**” és 1000.



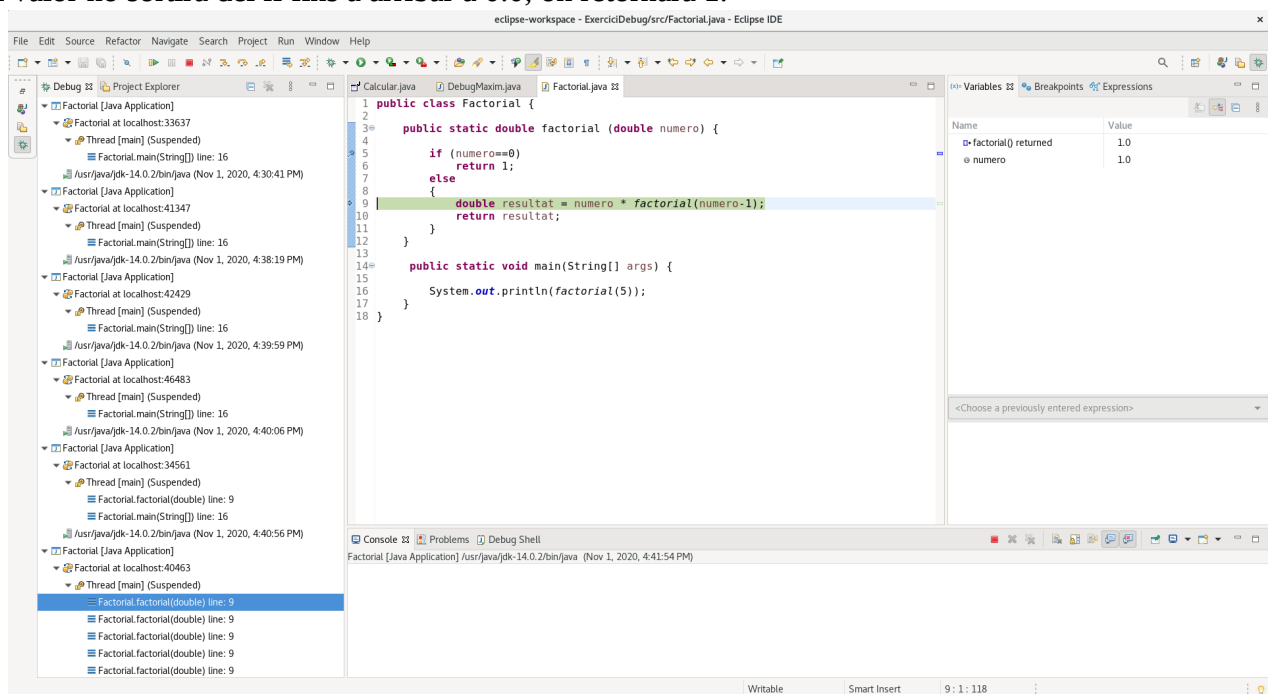
EXERCICI 2

·L'exercici ens demana depurar pas a pas el programa amb els breakpoints necessaris, veient el valor de les variables del factorial de 5.

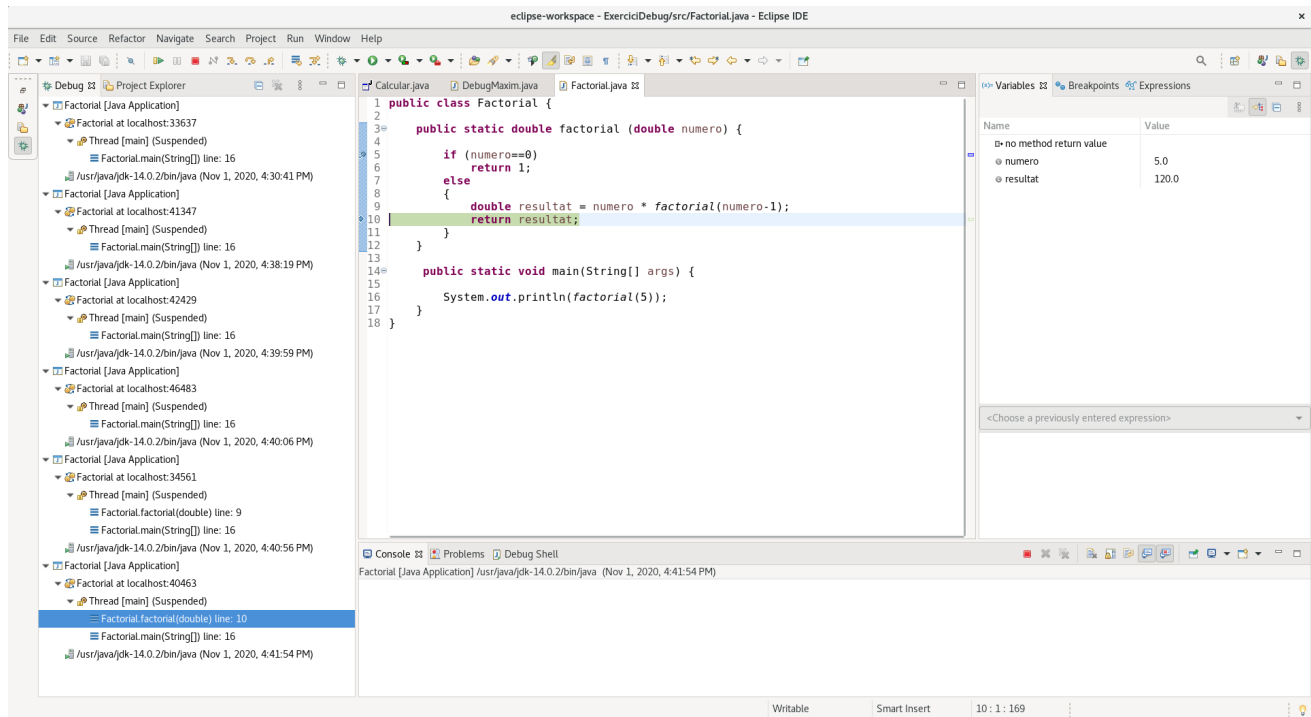
He determinat que únicament podem posar un breakpoint en la condició if, per tal de veure els valors canviants de les variables número I resultat. Primer comença amb el valor de 5.0.



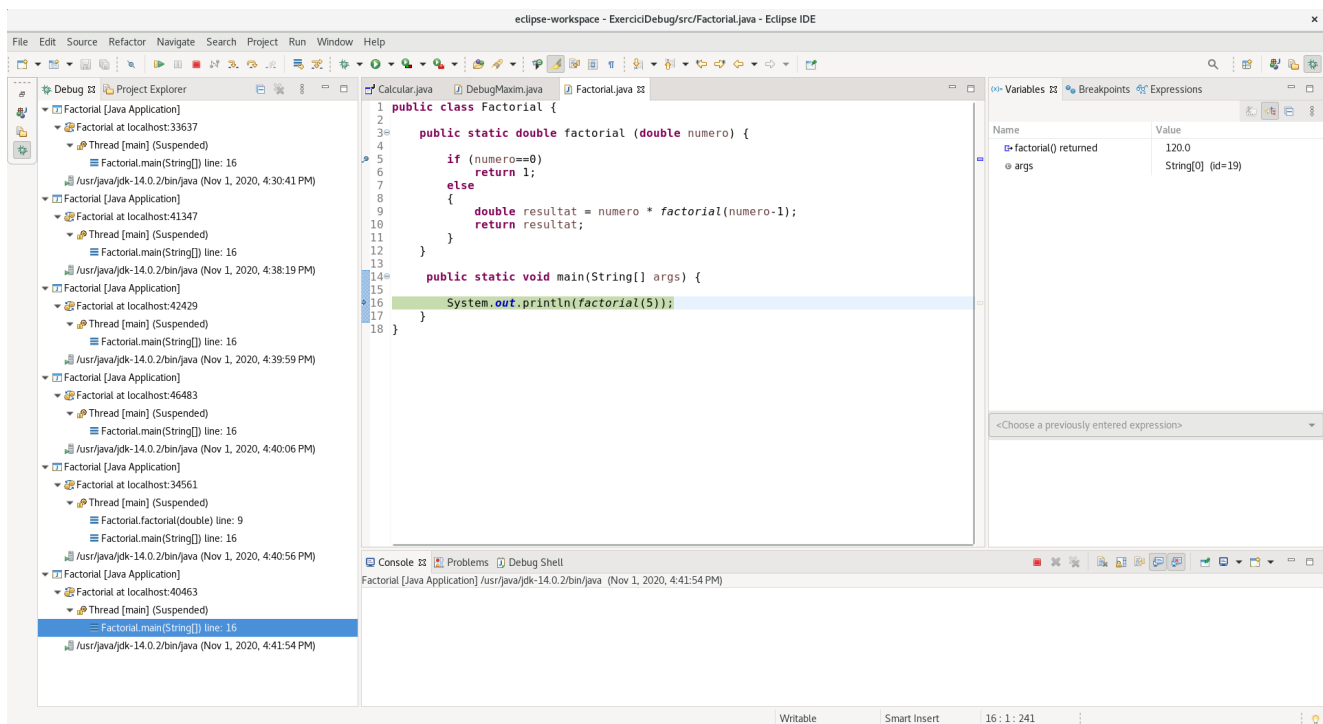
El valor no sortirà del if fins a arribar a 0.0, on retornarà 1.



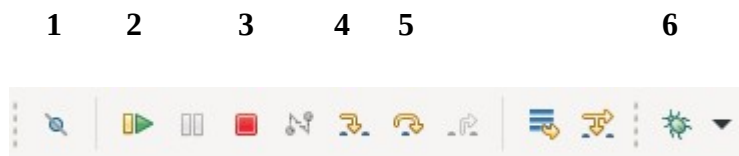
Aconseguit el valor del return, 1, entrarà a la part de resultat on començarà a emmagatzemar el valor del factorial 5, anant del valor 1 al 5.



Finalment, ja arribat al número 5 i amb el valor del factorial, es printarà el resultat final.



·L'exercici ens demana la definició I utilitat de les abreviatures del teclat per a la debugació.



1.Skip all breakpoints (Control + Alt + B). Deshabilita tots els breakpoints temporalment I els ignora en el procés d'execució.

2.Resume (F8). S'utilitza per donar continuïtat fins als breakpoints.

3.Terminate (Control + F2). Finalitza l'execució del programa.

4.Step Into (F5). S'utilitza per entrar a la trucada d'un mètode I es parará en la primera línia.

5.Step Over (F6). S'utilitza per anar pas a pas per el programa sense entrar a les trucades dels mètodes.

6.Debug Factorial. S'utilitza per inicialitzar el debug d'un programa.

·**Si el punt de Breakpoint està a la línia 16 (línies DEL CODI ANTERIOR, NO DEL TEU): té cap sentit? Per què?**

No té cap sentit, ja que no entrem al programa I hem anat directament a la sortida I resultat del programa.

·**Com podem fer per veure com funciona a partir d'aquest punt la funció factorial? Quin és la abreviatura de teclat?**

Si volem executar el programa, utilitzarem la tecla **F8 (Resume)**.

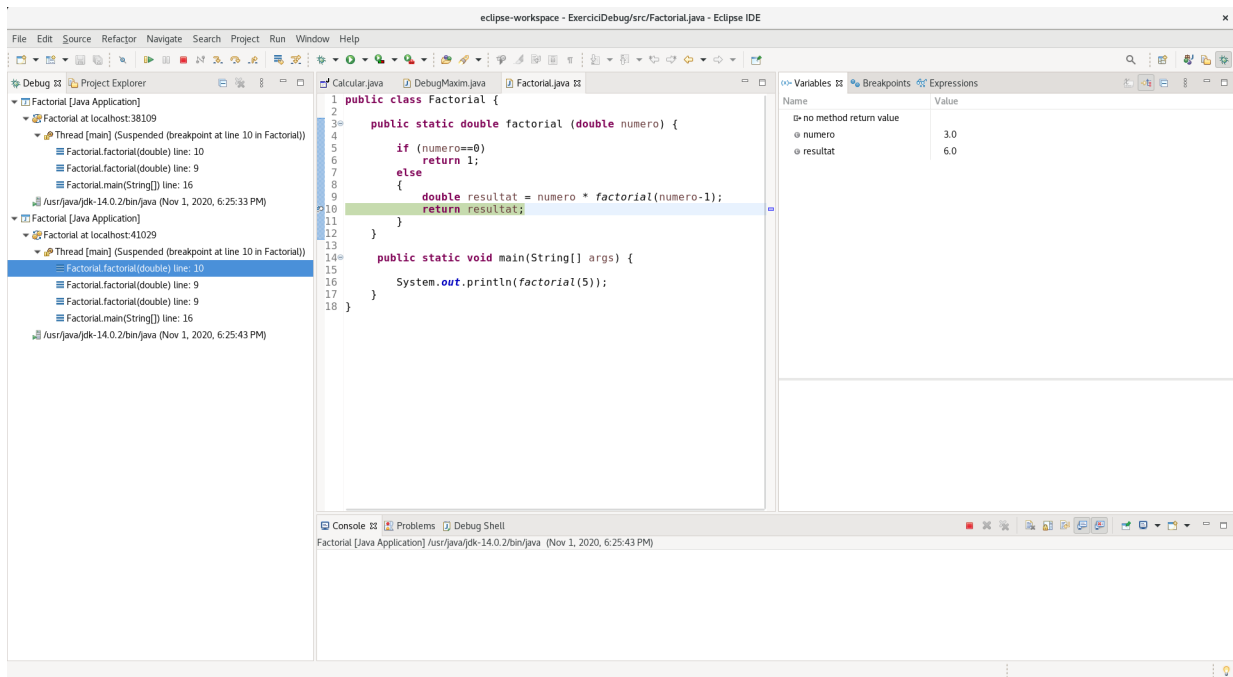
·**Si el punt de Breakpoint es troba a la línia 6, quantes vegades parará l'execució abans d'acabar el programa?**

Es parará fins que arribi al número factorial que hem insertat més l'execució de printar l'execució.

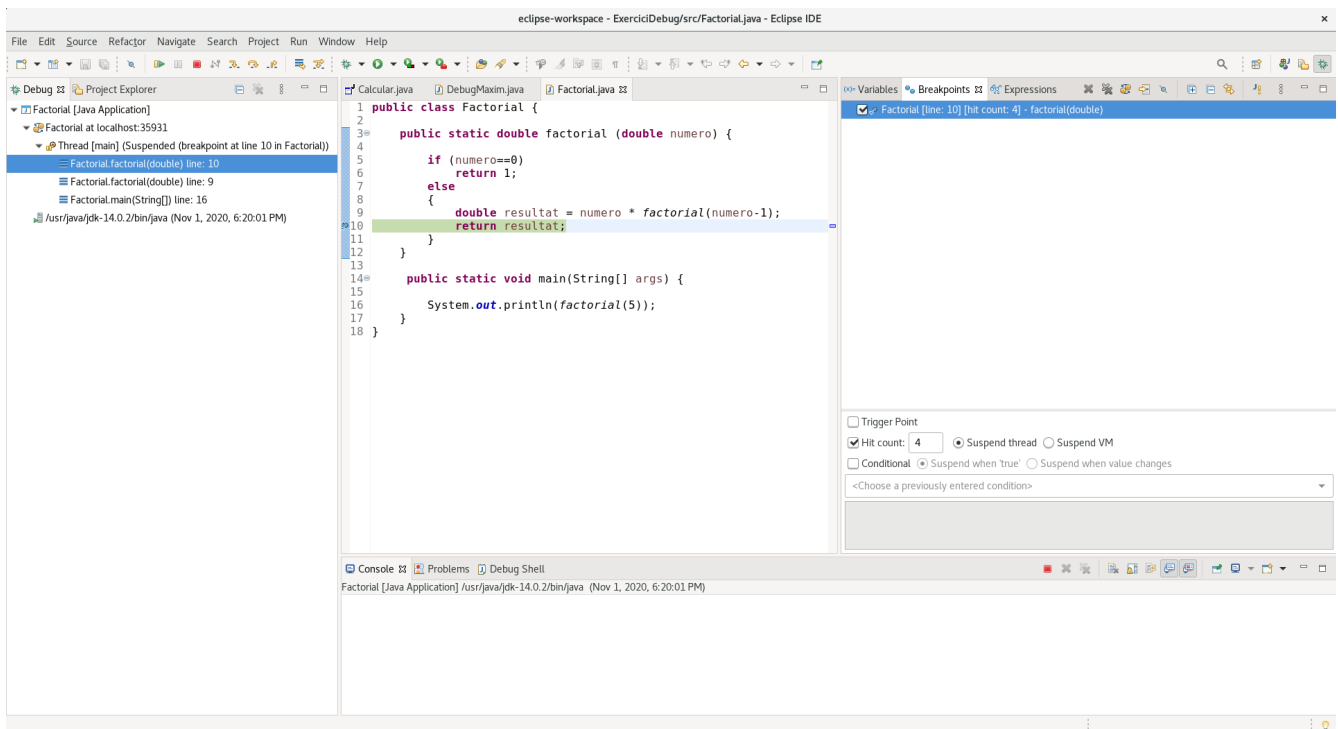
Exemple: en la línia 6, arribarà fins a 5 execucions I una més.

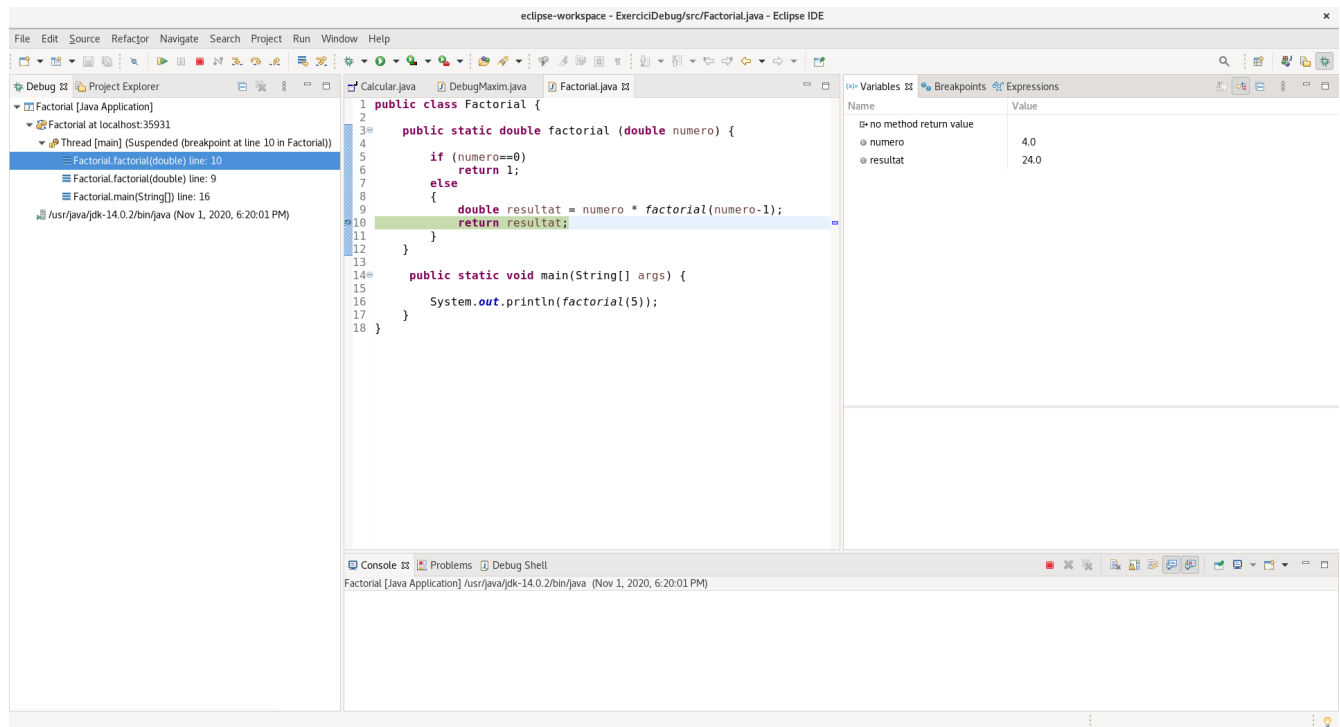
•Poseu un breakpoint a la linia 10 que només s’aturi la tercera vegada que passa.

En l’apartat breakpoints, afegirem la condició de hit count = 3.



•Canvieu les propietats del breakpoint pq s’aturi quan `numero == 4`.





•Extra: Creus que aquest codi pot tenir cap tipus de problema? Explica quin.

El problema que veig és que necessita un tractament d'excepcions en cas de no introduir un número.

•Extra: Creus que aquest codi pot tenir cap tipus d'optimització? Tan si la resposta es sí com si es no, explica per què.

Si, podem optimitzar afegint que l'usuari sigui el que introdueixi el valor del factorial. També algun cas de control d'errors.

Afegiu el tractament d'alguna excepció. Per exemple demanant a l'usuari el valor per calcular el factorial i afegiu un breakpoint que aturi el programa quan salti l'excepció. A la vista Breakpoints, seleccioneu la icona J!(Java Exception Breakpoint).

