Pràctica depuració de codi

L'objectiu d'aquesta activitat és familiaritzar-se amb l'ús de l'eina Debug per depurar el codi. Conèixer el concepte de punt d'interrupció (o breakpoint) i la seva aplicació.

Es demana la col·locació de punts d'interrupció per poder anar executant, pas a pas, el codi, validant els valors que van adquirint les variables.

S'ha de crear un nou projecte de Java que permeti calcular el factorial. S'ofereix, a continuació, un codi de programació que soluciona el factorial.

Haureu d'entregar un document amb el nom: **M05-PT1-Debug-Cognoms-Nom.*** amb les captures necessàries per documentar la pràctica.

Exercici 1

```
public class ExerciciMaxim{

	public static void main(String[] args) {

		ExerciciMaxim ex = new ExerciciMaxim();
		int[] nombres = {100, 500, 300,200, 1000, 600,700, 800,
400, 900};

		int maxim = ex.trobarMaxim(nombres);
		System.out.println(maxim);
	}

	public int trobarMaxim(int[] nombres) {
		int maxim= nombres[0];
		for (int i=1;i < nombres.length;i++){
			if (nombres[i]>maxim){
				maxim = nombres[i];
			}
		}
	return maxim;
	}
}
```

Es demana introduir els punts de ruptura (breakpoints) necessaris, per tal de poder executar el codi pas a pas i anar validant els valors que prenen les variables en cada moment.

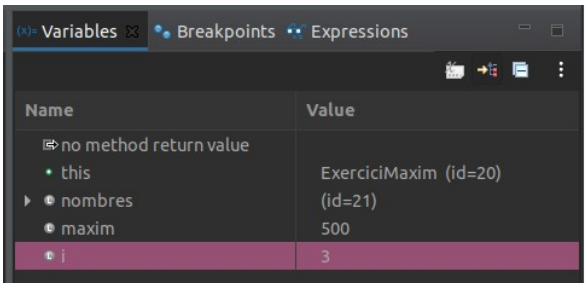
Digues en quina/es línia/es poses el/s punt/s de ruptura.

```
int maxim = ex.trobarMaxim(nombres);
```

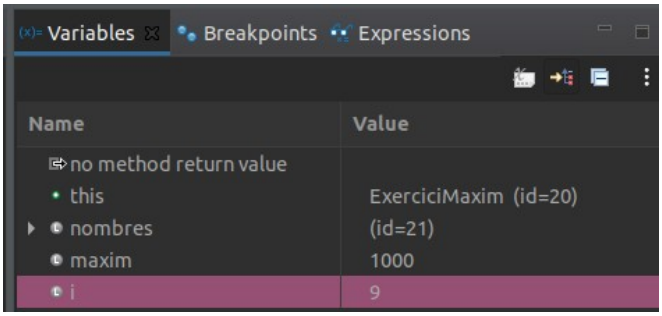
```
if (nombres[i]>maxim){
```

```
return maxim;
```

Enganxa una captura de pantalla on es vegi els valors que tenen les variables i i màxim després de la 3a iteració del bucle for.



Enganxa una captura de pantalla on es vegi el valor que té la variable màxim abans de fer el return.



Exercici 2

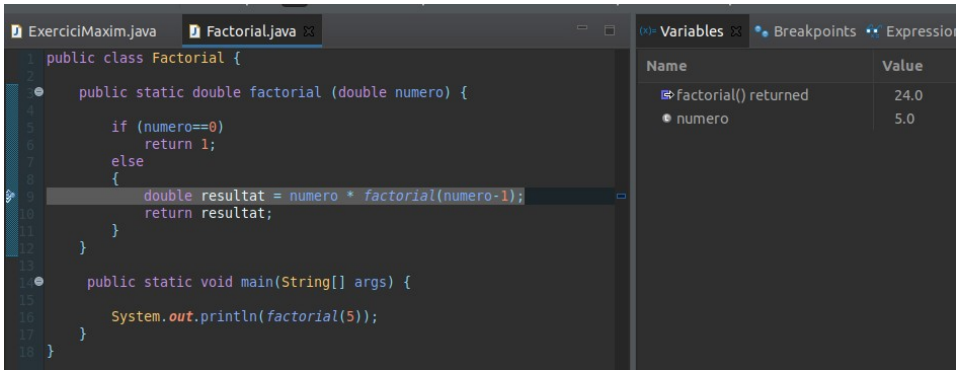
```
1.  public class Factorial {
2.
3.   public static double factorial (double numero) {
4.
5.        if (numero==0)
6.             return 1;
7.        else
8.        {
9.             double resultat = numero * factorial(numero-1);
10.            return resultat;
11.       }
12.  }
13.
14.  public static void main(String[] args) {
15.
16.       System.out.println(factorial(5));
17.  }
18. }
19. //El codi a copiar està l'annex.
```

1. Es demana depurar el codi pas a pas, introduint els punts d'interrupció necessaris i mostrant el valor que van tenint les variables. Fes un informe amb el seguiment del valor de les variables amb el suport de captures de pantalla. Mínim factorial a calcular per fer el treball: 5.

   Se ha puesto un BreakPoint en la línea 9, ya que desde ahí se puede ver las llamadas que la función se hace a si misma.

   Parte en la que se hacen las llamadas recursivas

Parte final, obteniendo el resultado



2. Fes servir les abreviatures de teclat per realitzar la debuggació. Enumera-les i explica quina funció fan.

- F5 --> "Paso a paso"
- F6 --> "Paso por encima"
- F7 --> "Paso de retorno"
- F8 --> "Reanudar"
- Ctrl + Mayús + B -->"Cambiar Breakpoint"
- Ctrl + F2 --> "Teminar ejecución"

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

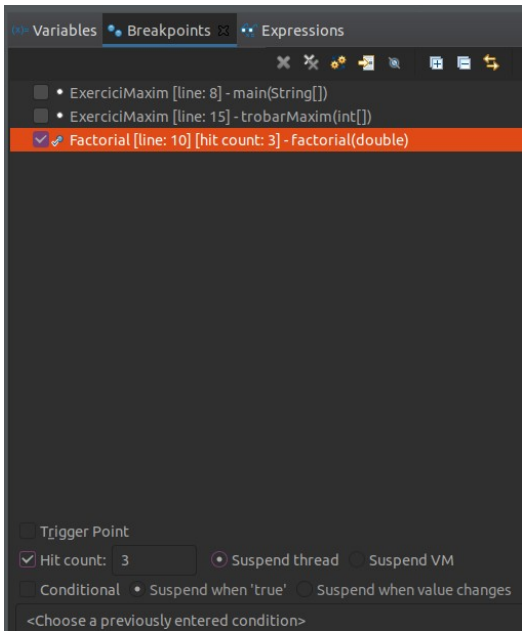
       Si tiene sentido si hay un código en el que simplemente queremos ver lo que nos da la función. En caso de que querramos inspeccionar la función; no tiene sentido, ya que recoge directamente lo que devuelve.
    b. Com podem fer per veure com funciona a partir d'aquest punt la funció factorial? Quin és la abreviatura de teclat?

Si se puede, usando F5.

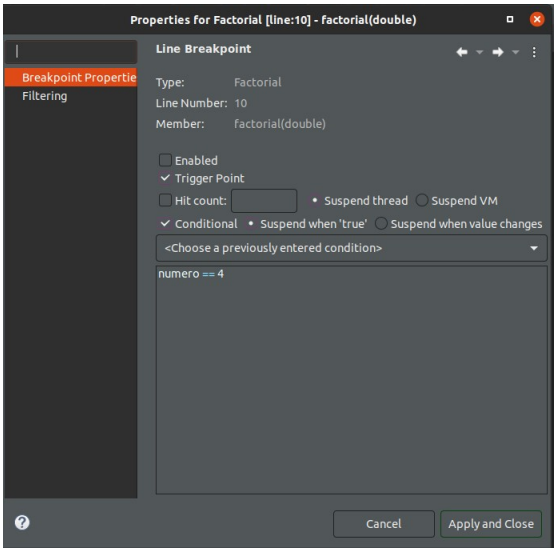
4. Si el punt de Breakpoint es troba a la linia 6, cuantes vegades pararà l'execució abans d'acabar el programa?

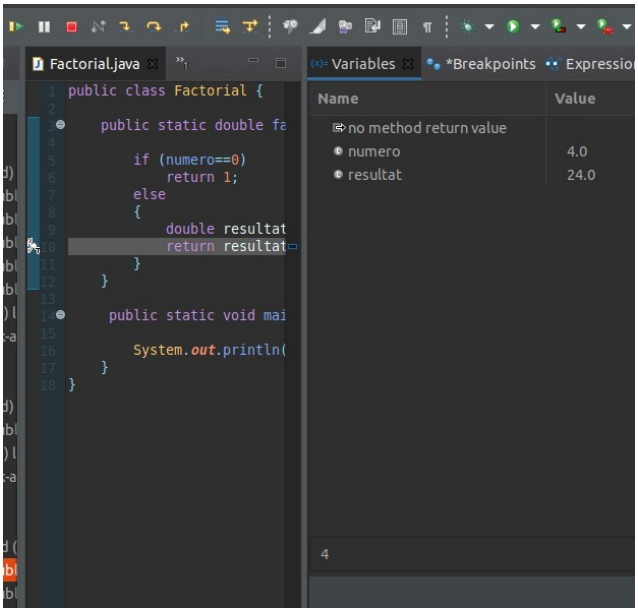
   Para 13 veces, hace las ejecuciones del else.

5. Poseu un breakpoint a la linia 10 que només s'aturi la tercera vegada que passa.

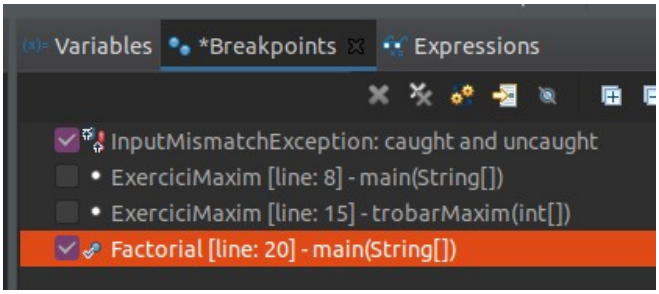


6. Canvieu les propietats del breakpoint pq s'aturi quan `numero =4`.





7. Extra: Creus que aquest codi pot tenir cap tipus de problema? Explica quin.
   De primeras no.
8. Extra: Creus que aquest codi pot tenir cap tipus d'optimització? Tan si la resposta es sí com si es no, explica per què.
   No (de primeras), el método de la recursividad parece bastante óptimo.
9. 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).



```
public class Factorial {

	public static double factorial (double numero) {

		if (numero==0)
			return 1;
		else
		{
			double resultat = numero * factorial(numero-1);
			return resultat;
		}
	}

	 public static void main(String[] args) {

		System.out.println(factorial(5));
	}
}
```