

El debugger de NetBeans

El debugger o depurador de NetBeans permite:

- Ejecutar el código fuente paso a paso
- Ejecutar métodos del JDK paso a paso.
- Utilizar breakpoint para detener la ejecución del programa y poder observar el estado de las variables.
- Conocer el valor que toma cada variable o expresión según se van ejecutando las líneas de código.
- Modificar el valor de una variable sobre la marcha y continuar la ejecución.

Mediante el debugger podemos:

- Encontrar de forma rápida y fácil errores en el programa.
- Entender mejor el flujo de ejecución del programa.

INICIAR EL DEBUGGER

Hay varias formas de iniciar la ejecución del programa en modo depuración:

1: Debug -> Debug Main Project (Ctrl. + F5) o pulsando sobre el botón:



También pulsando con el botón derecho sobre un proyecto en explorador de proyectos y seleccionar Debug

El programa se ejecuta hasta llegar al primer breakpoint. Si no han establecido breakpoints, el programa se ejecutará normalmente hasta el final.

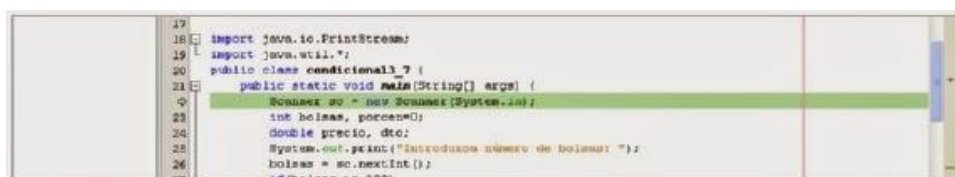
2: Debug -> Run to Cursor (F4)

Se ejecuta el programa hasta la instrucción donde se encuentra el cursor.







3: Debug -> Step Into (F7)

Comienza la depuración desde la primera línea del método main. El depurador se detiene esperando que decidamos el modo de depuración.

Una vez iniciada la depuración y el depurador se detiene, la siguiente línea de código que se va a ejecutar aparece en verde, con una flecha verde a su izquierda:



En este punto la depuración puede continuar utilizando distintas opciones:

	Step Over (F8) Ejecuta una línea de código. Si la instrucción es una llamada a un método, ejecuta el método sin entrar dentro del código del método.
	Step Into (F7) Ejecuta una línea de código. Si la instrucción es una llamada a un método, salta al método y continúa la ejecución por la primera línea del método.
	Step Out (Ctrl + F7) Ejecuta una línea de código. Si la línea de código actual se encuentra dentro de un método, se ejecutarán todas las instrucciones que queden del método y se vuelve a la instrucción desde la que se llamó al método.
	Run to Cursor (F4) Se ejecuta el programa hasta la instrucción donde se encuentra el cursor.
	Continue (F5) La ejecución del programa continúa hasta el siguiente breakpoint. Si no existe un breakpoint se ejecuta hasta el final.
	Finish Debugger Session (Mayúsculas + F5). Termina la depuración del programa.

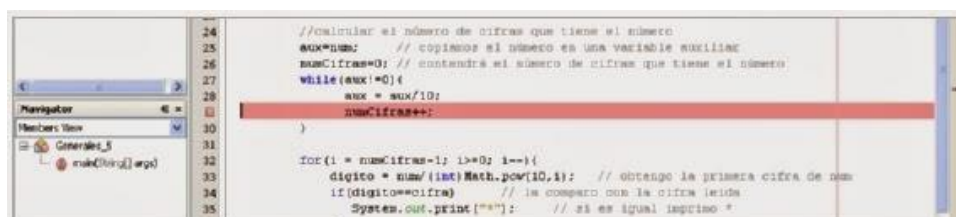
ESTABLECER BREAKPOINTS

Un breakpoint o punto de interrupción es una marca que indica al depurador que debe detenerse cuando la ejecución del programa llegue a ella.

Cuando el programa se detiene en un breakpoint podemos:

- Examinar los valores actuales de las variables.
- Detectar cuando se crea un objeto.
- Continuar la depuración línea a línea del programa.

Para fijar un breakpoint se pulsa sobre el número de línea donde se desea colocar. La línea queda resaltada en color rojo con una marca del mismo color en el margen izquierdo.



También se puede crear seleccionando: Debug -> New Breakpoint

Para eliminar un breakpoint se pulsa sobre el cuadrado rojo.

Para eliminarlos todos: botón derecho en la ventana de breakpoints -> Delete All

Para desactivar un breakpoint, botón derecho sobre la marca roja -> breakpoint -> desmarcar Enable.



Para desactivarlos todos: botón derecho en la ventana de breakpoints -> Disable All

Los breakpoints desactivados aparecen en gris.



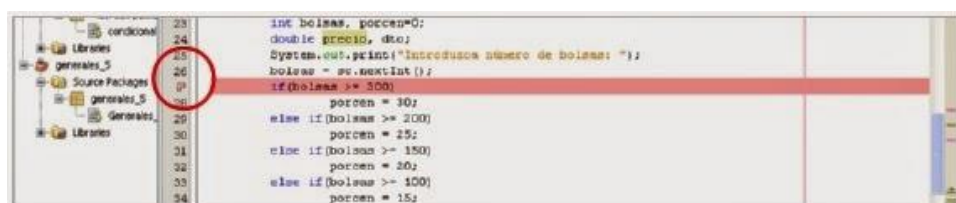
BREAKPOINTS CONDICIONALES

Podemos hacer que la depuración se detenga en un breakpoint solo si se cumple una determinada condición o si la línea de código se ha ejecutado un número determinado de veces.

Para fijar un breakpoint condicional:

Botón derecho sobre el breakpoint -> Breakpoint > Properties

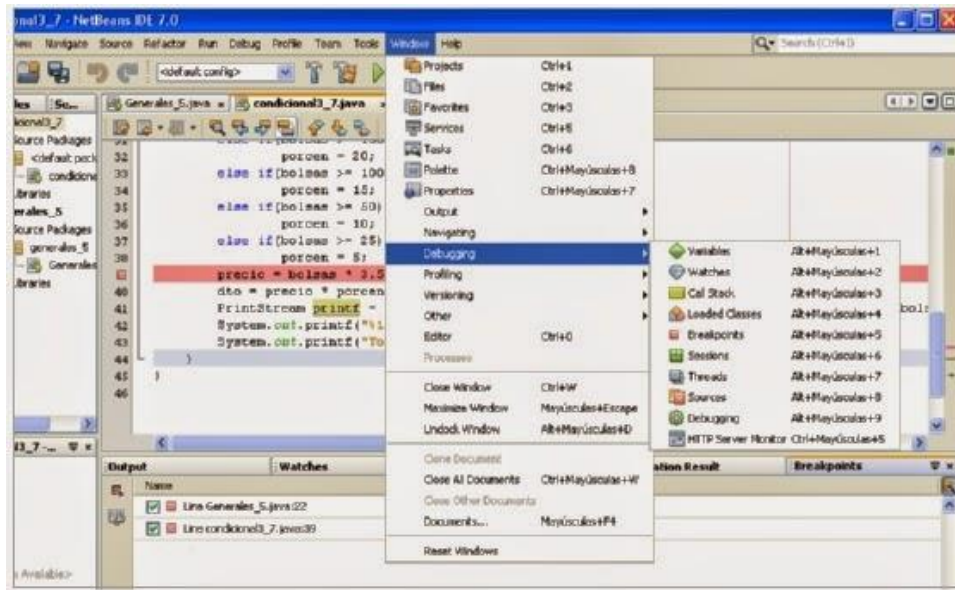
La marca cambia de aspecto:



DEBUGGING WINDOWS

En el proceso de depuración se usan distintas ventanas situadas bajo la ventana de código. Algunas aparecen automáticamente.

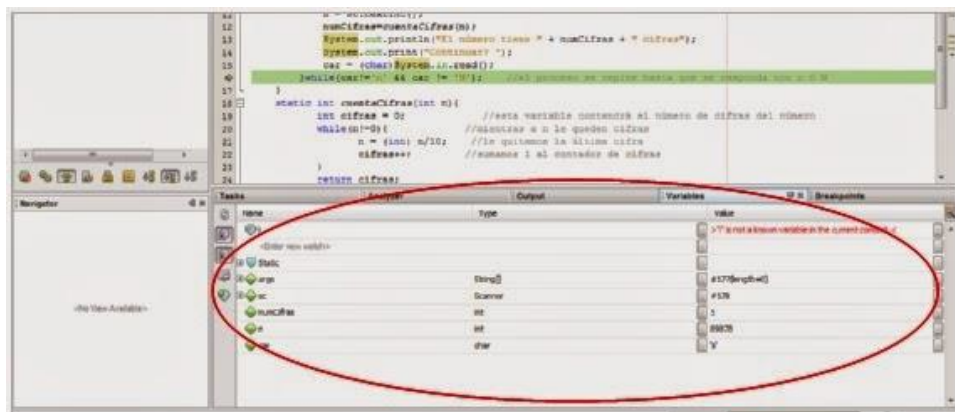
Para mostrarlas pulsar Windows-> Debugging y seleccionar la que queramos.



Las más importantes son: Breakpoints, Variables, Watches y Call Stack.

LA VENTANA DE VARIABLES LOCALES.

En esta ventana se muestran las variables, su tipo y su valor actual.



El debugger permite cambiar el valor de una variable local en esta ventana y continuar la ejecución del programa usando el nuevo valor de la variable.

AÑADIR UN WATCH

Para obtener el valor de una variable durante la depuración de un programa se sitúa el ratón sobre la variable y se mostrará, el depurador mostrará el tipo y el valor de la variable.

Otra forma de monitorizar el valor de una variable es mediante un watch.

Para agregar un watch a una variable o una expresión, selecciona la variable o expresión a supervisar y a continuación pulsar. Debug-> New Watch.

O también: botón derecho sobre la variable-> New Watch

Las variables a las que se les ha asignado un watch aparecen en la ventana Watches:



También aparecen en primer lugar en la ventana de variables.

MOSTRAR LA BARRA DE HERRAMIENTAS DEL DEPURADOR

La barra de herramientas del depurador aparece cuando iniciamos la depuración. Para mostrarla siempre: View -> Toolbars -> Debug.