Ejecución en modo depuración en Eclipse

Estíbaliz Martínez

Eclipse es un entorno de desarrollo muy completo que ofrece multitud de posibilidades. Una de ellas es permitir a los desarrolladores la depuración de su código ("debug") línea a línea.

Para ello únicamente hay que insertar un punto de ruptura en el código ("breakpoint") y ejecutar la aplicación en modo debug.

Inserción de puntos de ruptura en el código

Existen diversas formas de poner un punto de ruptura, pero la más habitual es hacer doble click con el ratón en la columna situada a la izquierda del código, en la línea que queramos introducir el punto de ruptura.



Una vez establecido el punto de ruptura quedará marcado con un círculo azul.



Ejecución en modo depuración.

Una vez establecido el punto de ruptura, ejecutamos la aplicación en modo depuración desde el menú Run \rightarrow Debug (o bien mediante el botón de la barra de herramientas).

En ese momento, aparecerá una ventana emergente preguntando si deseamos cambiar a la perspectiva debug. Esta perspectiva permite ejecutar paso a paso el código de nuestra aplicación e incluso ver el valor de las variables.

Poly Standardstation in Stine				
ebug - Ejempios/src/Autoboxing.java - Eclipse				
File Fait Zonice Kelactor Maxidate Sealich Buolect Rin Murgow Helb				
📫 • 🔜 🗟 🏇 • 🔾 • 🌯 • 🈕 🗁 🛷 • 🍄 🌙 🐲 🗏 🕤 ½] - {} - \$ - ↓ -	🐉 Java		
🏇 Debug 🖄 🥄 🦓 🕪 💠 🔳 🗗 🗎 🕺 🦘 🗟 🧮 🕱 👘 🍸 🗖 🗖	🕪 Variables 🕴 💁 Breakpoints 👘 🎼	~ - D)		
😑 🗾 Autoboxing [Java Application]	Name Value			
Autoboxing at localhost:3825	args String[0] (id=16)	String[0] (id=16)		
Inread (main) (Suspended (breakpoint at line 5 in Autoboxing))	E C int1 Integer (id=17)			
C:\Archivos de programa\lava\ire6\bin\iavaw.exe (11/11/2011 14:08:06)	a vaue 128			
	128	A.		
	***	~		
	<	>		
🕅 Autoboxing. java 😒	- D 🗄 Outine 🕄 👘 🖓 🔌 💉 🔹 🔧	□ □ □		
	A 😑 🕒 Autoboxing			
public class Autoboxing {	• S main(String[]) : void			
public static void main(String args[]) (
Integer intl=new Integer(128);				
Integer int2=new Integer(128);				
if (intl equals(int2))(
System out println("Son igueles!"):				
else				
System.out.println("Son diferentes! ¿?");				
}				
}	v			
Console S2 P Tacks				
Autoboving [Java Application] C-Varchivos de programa/Java/ire6/Jbin/Javaw.eve (11/11/20				
Autopowing (Java Application) Criptic into de programatizava (reolumit) avaiv. 6x8 (11/11/20	711 17:00:00)	-		
		>		
i 0 °				
		_		

Para controlar la ejecución del código, nos apoyaremos en los siguientes botones:

3 • 🔡 🗟 🗄 🎋 • 🔕 • 🥵 • 🤌 🛱 🌮 🖓 💓 🔳 🖬 🗄 🐕	• 🖓 • 🏷 🗘 • 🔿	
Debug 🛛 💦 🥻 🕪 🔢 🔳 🕅 🔍 🕾 🐟 🗟 😿 🌍 🌱 🗖	🗱 Variables 🛿 📀	
Autoboxing [Java Application]	Name	
🖶 🔗 Autoboxing at localhost:3825	I args	
😑 🐠 Thread [main] (Suspended (breakpoint at line 5 in Autoboxing))	🗉 🔍 int1	
Autoboxing.main(String[]) line: 5	🚽 🖌 value	
🔄 🖳 C+\Archivos de programa\lava\ire6\bin\iavaw eve (11/11/2011 14:08:06)		

Que nos permiten entre otras opciones.

- Ejecutar línea a línea el código sin entrar en los métodos.
- Ejecutar línea a línea el código entrando en los métodos.
- Ejecutar hasta el siguiente punto de ruptura.
- Finalizar la ejecución

Como cambiar de perspectiva.

Si posteriormente el desarrollador quiere cambiar de perspectiva, puedo hacerlo apoyándose en los botones de la esquina superior derecha.

arch <u>Project R</u> un <u>W</u> indow <u>H</u> elp							
🍅 🖨 🔹 🗣 🌛 🗢 🔳 👔 🕴 👱	• 🖓 • 🍫 🔶 • ⇔ •				🏇 Debug	🖏 Java	
84 3. 👁	😡= Variables 😫 💁 Breakp	oints			An 📲 🗉		
	Name		Value				
	args		String[0]	(id=16)			
eakpoint at line 5 in Autoboxing))	🗉 🔍 int1		Integer (id=17)			
ne: 5	e ^r value		128				
\bin\javaw.exe (11/11/2011 14:08:06)							
	128					< >	
	<					>	
🗖 🗖 🔡 Outline 🖄 🔰 🖓 💐 💩 🗙 🏱 🗖 🗖							
<pre>{ in(String args[]){ ger(128); ger(128); </pre>		Θ _Γ Α	utoboxing ^S main(Str	ring[]) : voi	d		
- <u>j</u> == (=== / /							

O a través de las opciones de menús disponibles en Window

