

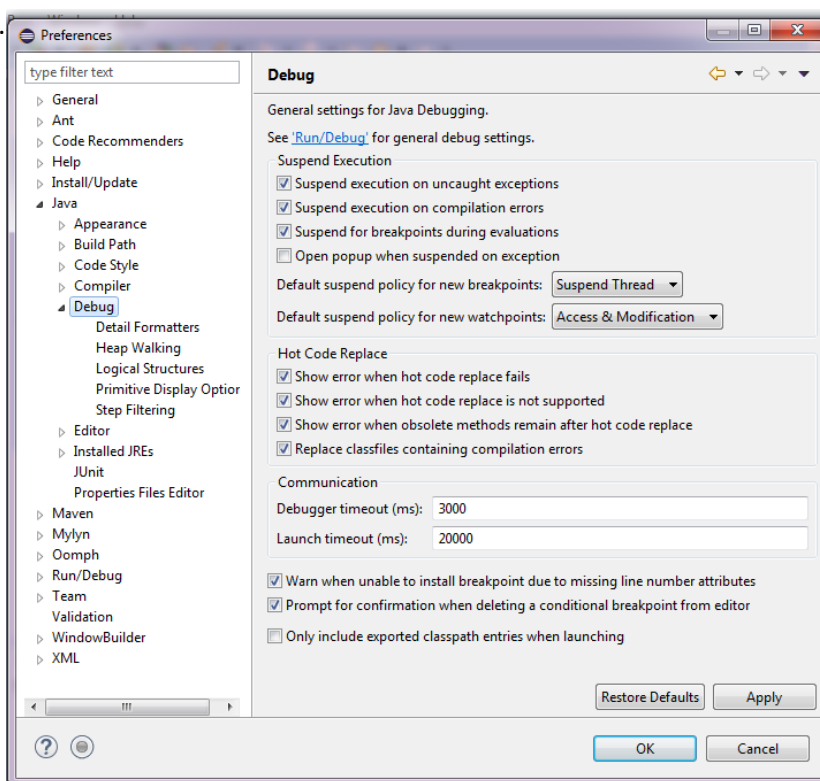
IDE ECLIPSE – PERSPECTIVA DEBUG

Buscar Bugs, en proyectos complejos es una tarea difícil y que consume mucho tiempo. Para estos casos un debugger poderoso es de mucha utilidad. La IDE eclipse tiene un debugger muy potente incorporado.

Este debugger tiene 2 modos de operación local y remoto. Veremos el local.

- En la configuración, en la pestaña JAVA, DEBUG, STEP filtering, se puede configurar que CLASES SERAN SALTEADAS cuando se debuggee paso a paso el programa.

Configuración.



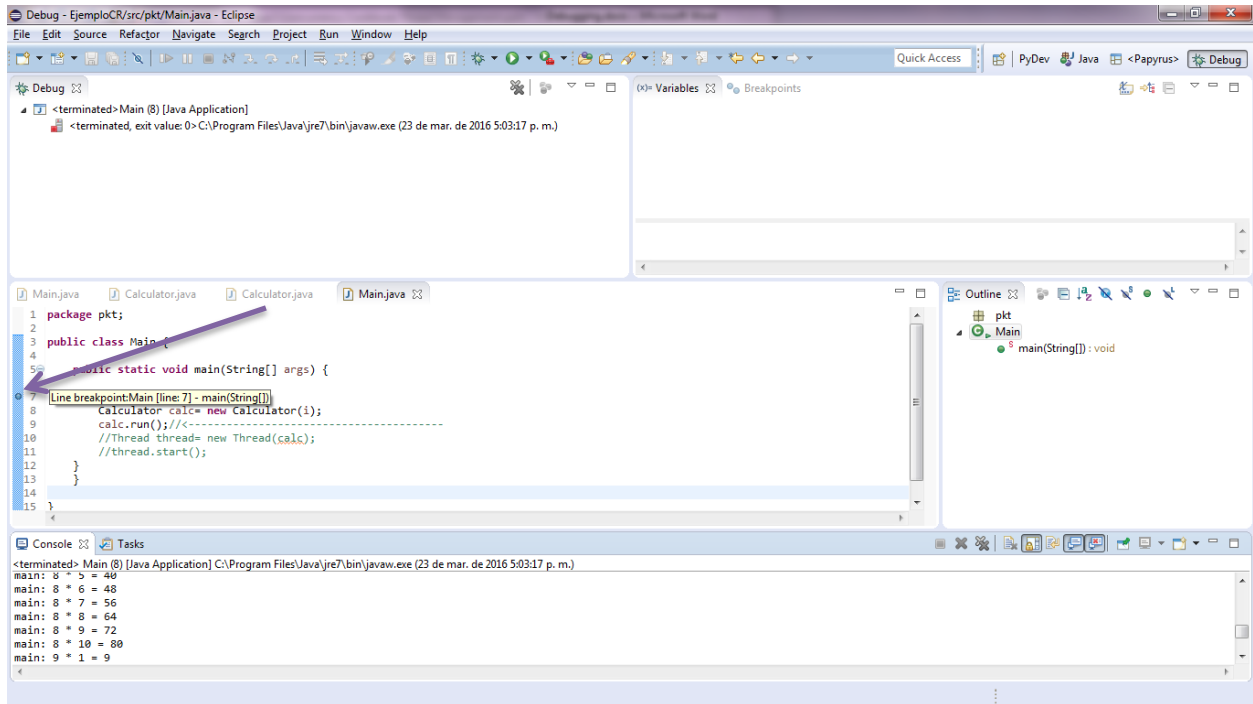
Por defecto vienen tildadas las java clas lang, para ser salteadas, ud puede agregar lo que ud desee.

- Ud. Puede comenzar a debuggear haciendo click en el icono del bug en el workbench.



- La vista del modo debugg, es igual a la de desarrollo, solo que con 2 ventanas más.
 - Una ventana, la superior izquierda, muestra los hilos activos.
 - En la ventana superior derecha, podemos observar las variables y sus valores, asociadas a la ejecución actual. Además en esta ventana, hay otro tab de breakpoints.

- Settear breakpoints, para detener en ese momento la ejecución del programa, se logra, haciendo doble click en el margen izquierdo del código. En cualquier ventana. Si quiero sacarlo, solo debo doble click de nuevo en el mismo lugar.



Una vez seteado el breakpoint y lanzado el debug, se detendrá la ejecución al llegar a este punto. Aquí se presentaran las siguientes alternativas:

- 1-presionar F8, con esto continuara la ejecución hasta encontrar el próximo breakpoint.
- 2-terminar la ejecución presionando stop.
- 3-ejecutar el método que sigue, con la tecla f5.
- 4-Saltar el método que sigue (instrucción por instrucción), con la tecla f6.

- Puedes verificar el valor de las variables de tu programa, en la ventana superior derecha. O con posicionar el mouse por sobre la variable que tenemos interés en ver su valor.
- Haciendo doble click en la variable, en la ventana de variables (superior derecha), se puede modificar su valor actual.

En la mayoría de las modificaciones menores de código se puede hacer hotswap, lo que implica que es posible modificar el código fuente, y guardar y continuar con la sesión de debug. Para modificaciones mayores, se recomienda cerrar y recomenzar la ejecución del debug.

- En la pestaña de breakpoints se pueden manejar los mismos, se pueden borrar determinados breakpoints, y se observa una vista general de todos los breakpoints definidos en el código.

- Con el check box, al lado de cada breakpoint se puede deshabilitar temporalmente ese breakpoint o activarlo.
- Al hacer click sobre el breakpoint, aparecen algunas opciones avanzadas. Por ejemplo, se puede setear un hit point, el cual indica que el breakpoint se activara luego de pasar esa cantidad de veces por el mismo. Tambien, se pueden setear opciones condicionales.
- Tambien hay una opción, interesante, add java exception breakpoint, para que si hay una exception uncaught, se detenga la ejecución sin abortar el programa y mostrar el stack trace. Se puede ver las variables y que sucedió.
- Tambien esta la opción de deshabilitar todos los breakpoints.