Debugging

Debugging is the process of identifying and fixing any issues in the source code of a program.

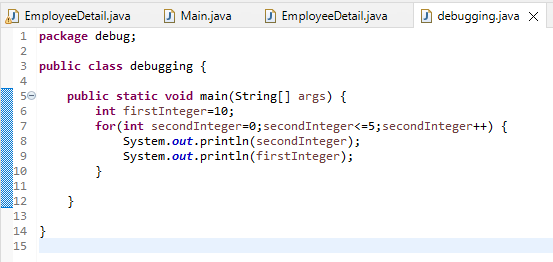
Debugging tools that make it easier for the developer to understand the code and how the machines understand.

Debugging in eclipse

Eclipse allows running an application in debugging mode which helps with stepping through each line of the code in a program.

In eclipse, a debugging perspective which helps to inspect code and make the debugging process very effective.

Example:

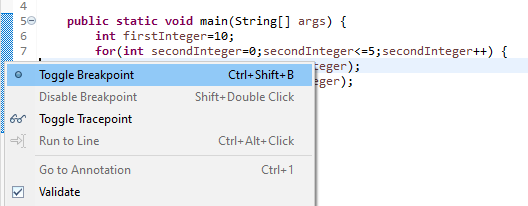


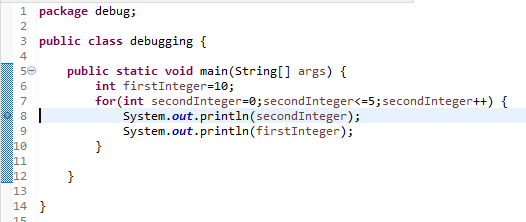
Set Breakpoints:

A breakpoint is appoint in code where the program execution pauses during debugging.

This allows the programmer to inspect code and the flow of execution at the defined breakpoint.

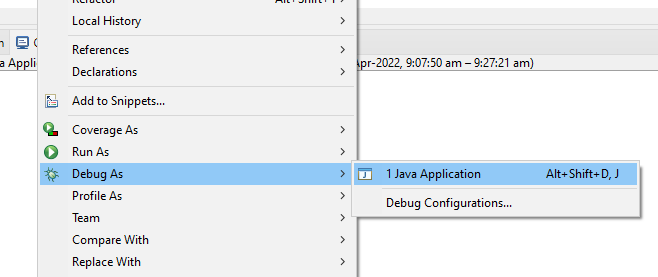
To define a breakpoint by clicking on the left margin in the editor or right click and select toggle Breakpoint.



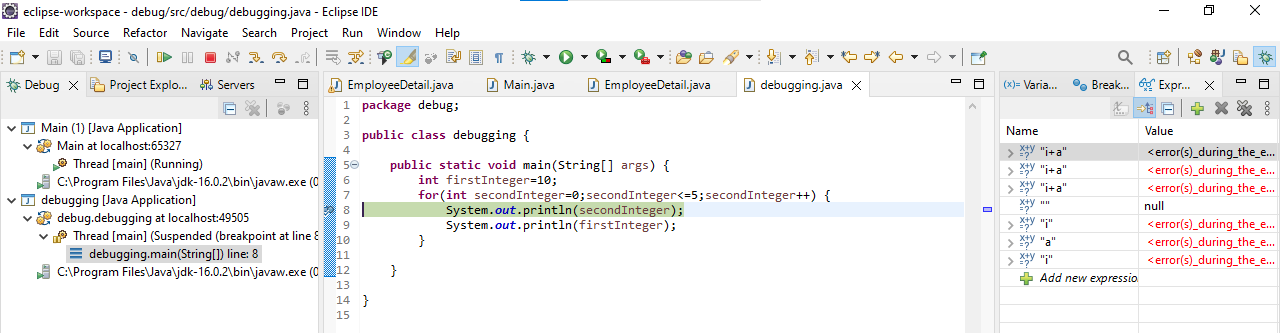


Debug mode:

To debug the application, within the java editor and select Debug As -> Java Application.

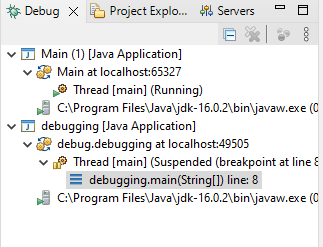


After selecting debugging perspective



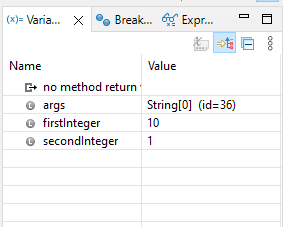
Debug:

Displays the call stack which helps in determining the flow of execution of the program until the breakpoint is reached.



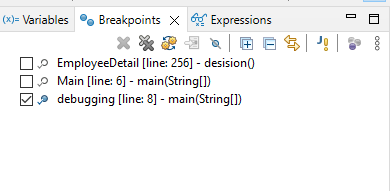
Variables:

Displays fields and defined variables in the current stack.



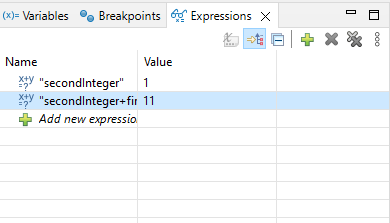
Breakpoints:

Shows a list of all breakpoints in the code and enabling or disabling breakpoints.



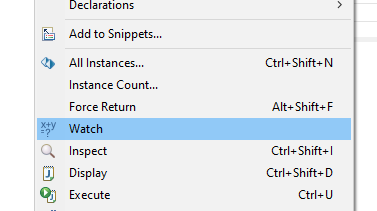
Expressions:

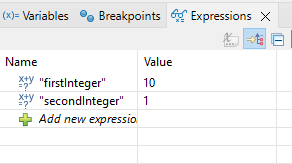
Allows defining custom java expressions to inspect their values.



Check the variables values in expressions:

To track the value of the firstInteger and secondInteger variables, they can be added to the expressions view by right clicking on each and then clicking watch.





Eclipse provides several buttons in the toolbar for controlling the flow of execution of the program.



Resume: Resumes normal execution of the program until the next breakpoint is reached.

Step into: Executes the current line of the code and dives into the next line of the code in the program execution. If the current line calls a method, the debugger steps into the method.

Step over: Executes the current line of the code and goes to the next line without stepping into any method calls or associated scope of the current line.

Step return: Steps back out of the current method and returns to the method.

Terminate: The debugger can stopped by pressing the terminate icon in the toolbar.

