Mastering your eclipseIDE

Java tooling, Tips & Tricks!

Noopur Gupta
Eclipse JDT co-lead
IBM

@noopur2507
@EclipseCon
@EclipseJavaIDE
• Window > Appearance
  • Hide Toolbar, Status Bar
  • Full Screen (Alt + F11)
• Show key bindings when command is invoked
• Zoom in Text Editors
  • Zoom In: Ctrl + + (or) Ctrl + =
  • Zoom Out: Ctrl + -
  • Pinch gestures on touch-enabled devices
• Working Sets
• Preferences > Java > Appearance
  • Abbreviate Package Names with Custom Rules
  • Sort Library Entries Alphabetically in Package Explorer
NAVIGATION

• Quick Access / Find Actions (Ctrl + 3)

• Quick Switch Editor (Ctrl + E) – Open editors list
  • ‘Delete’ key – close editor

• Quick Outline (Ctrl + O, twice to show inherited members)
  • In-place Outline (Ctrl + F3)

• Configure problem severity quick fix (Ctrl + 1)

• Hovers (Preferences > Java > Editor > Hovers)
  • In-place Javadoc (with error/warning on element)
    • Ctrl + Shift and hover, or use Javadoc view
  • In-place source code
    • Shift and hover, or use Declaration view
  • Sticky Hover (F2) – Open attached Javadoc in a Browser
  • Expand vertical ruler icons upon hovering
NAVIGATION

• Go to last edit location (Ctrl + Q)
• Alt + Left/Right – move backward/forward in editor navigation history

• Markers
  • Bookmarks
    • Right-click on left editor pane > Add Bookmark...
    • Bookmarks view
  • Tasks
    • Right-click on left editor pane > Add Task...
    • Tasks view
    • Preferences > Java > Compiler > Task Tags

• Show In (Alt + Shift + W) > System Explorer
INVESTIGATION

- Plug-in Spies
  - Alt + Shift + F1 – Plug-in Selection Spy
  - Alt + Shift + F2 – Plug-in Menu Spy
  - Alt + Shift + F3 – Show Contributing Plug-in
  - Ctrl + Alt + Shift + F9 – Layout Spy

- Plug-in Image Browser

- Externalized Strings
  - Ctrl + Click on key in .properties file to find referencing code

- Show Revision Information
  - To determine who last modified a line of code and when
  - Right-click on left editor pane > Show Revision Information
**INVESTIGATION**

- **UI Responsiveness Monitoring**
  - For automatic detection and logging of stack traces when Eclipse UI is unresponsive
  - Window > Preferences > General > UI Responsiveness Monitoring

- **Java Stack Trace Console**
  - Copy stack trace from log file and Navigate > Open from Clipboard (Ctrl + Shift + V)
  - Click on hyperlinks to jump to the code
  - Click on exception name hyperlink to create an exception breakpoint
• Content Assist - Insert or Overwrite
  • Replace existing text while invoking content assist
  • Press ‘Ctrl’ to toggle while content assist is active

• Content Assist Templates
  • Configure/Create in Templates view, Export/Import in Preferences

• Content Assist Postfix Proposals
  • Preferences > Java > Editor > Content Assist > Advanced

• Auto Activation of Content Assist
  • Auto activation triggers for Java
  • Disable insertion triggers except ‘Enter’
  • Preferences > Java > Editor > Content Assist

• Type Filters
  • Ignore certain types from content assist, imports organization, quick fixes, open type dialog etc.
  • Preferences > Java > Appearance > Type Filters
• Code Minings (as decorative text; via extension point)
  • Preferences > Java > Editor > Code Minings
  • Show references
  • Show implementations

• Show method parameter names
  or, Show method parameter hints – Ctrl + Shift + Space

• Block Selection Mode (Alt + Shift + A)
  • To edit a large number of almost identical lines at the same time

• Expand selection to enclosing element (Alt + Shift + Up / Down)
• Duplicate lines (Ctrl + Alt + Up / Down)
• Move lines (Alt + Up / Down)
• Delete lines (Ctrl + D)
• Smart Step Into (Ctrl + F5 or Ctrl + Alt + Click)
  o To step into a single method within a series of chained or nested method calls

• Step Filters
  o To filter out specified classes and packages while stepping into code
  o Preferences > Java > Debug > Step Filtering

• Run to Line (Ctrl + Alt + Click on left editor pane)

• Grouping and Sharing breakpoints – Breakpoints view
  o Sort by name or creation time
  o Group by various categories or create working sets
  o Share by exporting/importing breakpoint groups
DEBUGGING

• Tracepoint
  • A point in code where the debugger does not break the execution but only prints to console
  • Prevents the addition of print statements in code while debugging
  • Can be used to debug race conditions or to see the order of execution of threads
  • Run > Toggle Tracepoint

• Trigger Point
  • Set a breakpoint as trigger point in Breakpoint Properties
  • All the breakpoints (except trigger points) are disabled until a trigger point is hit
  • All the trigger points are disabled after a trigger point is hit and all the trigger points will be re-enabled for the next run
  • Remove all trigger points – context-menu action in Breakpoints view

• Hit count - The breakpoint suspends the execution of a thread only when it is encountered for the Nth time.
• Paste Code on Package Explorer
  • A new Java project will be created and the file will be opened in Java editor

• Java 13 – Switch Expressions, Text Blocks – As preview features
  • Enable preview features in Preferences
  • Switch Expressions and Enhanced Switch Statements
    • Add missing case statements and default case
    • Split case labels
    • Ctrl + Click on ‘case’ and ‘default’ to jump to the target location in switch

• Text blocks
  • Double-click to select text block content
  • Refactoring e.g. Extract to local variable (replace all occurrences), Inline local variable
  • Ctrl + I to indent
Java 11 and Java 10 - ‘var’ support
• Declare extracted local variable’s type as ‘var’
• Replace ‘var’ with inferred type & vice-versa
• Hover to view the inferred type for ‘var’
• Ctrl + Click to navigate to the inferred type of ‘var’

Java 9 – Java Platform Module System
• Convert to modular project - create module-info.java action
• Add libraries to module path
• Configure properties of module graph
• Quick fix to identify and add required module to module-info.java
• Create and export a non-empty package in a single step from module-info.java
• Quick fix to provide a service implementation from module-info.java
Java 8 – Lambda Expressions and Method References

- Convert anonymous class to lambda and back
- Convert lambda to method reference and back
- Add inferred lambda parameter types
- View and navigate to the functional method
- Refactoring on functional interface instances e.g. Change method signature

- [https://wiki.eclipse.org/JavaX/Examples](https://wiki.eclipse.org/JavaX/Examples), X is Java version
JUnit 5 TOOLING

• Quick assist to create JUnit Jupiter test class and test method

• Grouped assertions and combined result comparison

• Navigate to failing source location
  • Double-click on entry in JUnit view’s failure trace
  • Click "Show Stack Trace in Console View" button and use the hyperlinks

• Tagging and Filtering (@Tag)
  • Configure Tags dialog in JUnit launch configuration

• Embracing JUnit 5 with Eclipse IDE – Webinar
RESOURCES

• List all keyboard shortcuts (Ctrl + Shift + L)
• Configure keyboard shortcuts (Ctrl + Shift + L - twice): Preferences > General > Keys
• Eclipse Help > Tips and Tricks...
• Eclipse Help > Welcome > What’s new
• New and Noteworthy: https://www.eclipse.org/eclipse/news/4.x
• Twitter
  • @EclipseJavaIDE
  • #EclipseTips
• Instagram
EVALUATE THE SESSIONS

Sign in and vote using the conference app or eclipsecon.org

-1 0 +1
Thank You!

@noopur2507
@EclipseCon
@EclipseJavaIDE