Core

Ctrl+1	Quickfix
Ctrl+3	Open "Quick Access" search
Ctrl+␣	Content Assist
Alt+⇧Shift+Z	Surround with (do, for, if, try, ..)
Ctrl+Z	Undo
Ctrl+⇧Shift+Z	Redo

Testing

Alt+⇧Shift+X, T	execute Tests	
Ctrl+R	execute Tests	moreUnit
Ctrl+J	jump to test or implementation	moreUnit

Comment

Ctrl+/	Comment / uncomment line or selection (adds '//')
Ctrl+⇧Shift+7	Add Block Comment around selection (adds '/... */')
Ctrl+⇧Shift+ß	Remove Block Comment
Alt+⇧Shift+J	Add Element Comment (adds '/** ... */' JavaDoc)

Text Editing

Ctrl+⇧Shift+Y	lowercase selected text
Ctrl+⇧Shift+X	uppercase selected text
Ctrl+D	Delete line
Ctrl+⇧Shift+Del	Delete until end of line
Ctrl+Del	Delete next element
Ctrl+⌫	Delete previous element
Ctrl+Alt+↓/↑	Copy line below/above
Alt+/	Propose word completion - after typing at least one letter. Repeatedly press until reaching correct name
Alt+⇧Shift+A	Column (block) selection, Multiline block editing
⇧Shift+Enter	Adds a blank line below the current line
Ctrl+⇧Shift+Enter	Adds a blank line above the current line

Generate

Alt+⇧Shift+S, R	Generate Getters and Setters

Movement

Alt+↑/↓	Move current line or selection up or down
Ctrl+←/→	Jump one word to the left / right
Ctrl+⇧Shift+↑/↓	Jump to previous / next method
Ctrl+L	Jump to Line Number. To hide/show line numbers, press Ctrl+F10 and select 'Show Line Numbers'
Ctrl+Q	Jump to last location edited
Ctrl+./,	Jump to next / previous warning or error
Ctrl+↑	Scroll up
Ctrl+↓	Scroll down
Ctrl+⇧Shift+P	Go to the matching bracket
Alt+⇧Shift+↑/↓	Select the previous / next syntactical element
Alt+⇧Shift+↑/↓/←/→	Extending / reducing the selection of the previous / next syntactical element

Tip: Use bookmarks. Add them on left margin of the code view (where you set the breakpoints) with "right click → Add bookmark". Add the Bookmarks-View (Window → Show View → Bookmarks).

Selection

Alt+⇧Shift+↑	Expand selection to current element
Alt+⇧Shift+↓	Reduce previously expanded selection by one step

Search / Information

Ctrl+⇧Shift+T	Open Type
Ctrl+⇧Shift+H	Open Type In Hierarchy
Ctrl+⇧Shift+R	Open Resource
Ctrl+O	Show Outline **(second press shows the inherited members)**

Search / Information (cont)

F2	Open class, method or variable information (tooltip text)
F3	Open Declaration: Jump to Declaration of selected class, method or parameter
F4	Open Type Hierarchy window for selected item
Ctrl+T	Show Quick Type Hierarchy **(toggles between supertypes and subtypes when pressed again)**
Ctrl+U	Find occurrences of expression in current file
Ctrl+mouse over method	Open Declaration or Implementation
Ctrl+⇧Shift+L	List of shortcuts (depending on the context)
Ctrl+⇧Shift+␣	Method Parameter Hints cursor is located at the parameter specification for method reference
Ctrl+⇧Shift+G	Search workspace for references to selected method or variable
Ctrl+G	Search workspace for declarations of to the item under the cursor
Ctrl+F	Quick search
Alt+↲	Show properties (in Package or Project Explorer)

Cleanup

Ctrl+⇧Shift+O	Organize Imports
Ctrl+⇧Shift+F	Format Source
Ctrl+I	Intend selected source
Ctrl+⇧Shift+M	Add Import / also static imports

Refactoring

Alt+⇧Shift+R	Rename Element
Ctrl+2, R	Rename locally (in file), faster than Alt+⇧Shift+R
Alt+⇧Shift+L	Extract to local variable
Alt+⇧Shift+M	Extract to method
Alt+⇧Shift+T	Opens the context-sensitive refactoring menu
Alt+⇧Shift+R	Rename
Alt+⇧Shift+V	Move
Alt+⇧Shift+C	Change Method Signature
Alt+⇧Shift+M	Extract Method
Alt+⇧Shift+L	Extract Local Variable
Alt+⇧Shift+I	Inline

Editor Window

F12	Jump to Editor Window
Ctrl+PgDn/PgUp	Next / previous Editor - in the **order of their tabs** in the editor area
Ctrl+M	Maximize or un-maximize current Editor Window
Ctrl+E	Show list of open Editors. Use arrow keys and enter to switch
Alt+←/→	Backward / Forward editor navigation history
Alt+-	Open Editor Window Option menu
Ctrl+F6	Next Editor - if released it immediately, Eclipse will take you to the **last active editor**

Editor Window (cont)

Ctrl+⇧Shift+F6	Previous Editor	
Ctrl+E	Quick Switch Editor	
Alt+F7	Multipage editor - next Page	
Alt+⇧Shift+F7	Multipage editor - previous Page	
Ctrl+-	**Toogle Split Editor** (Horizontal) - since Luna	**old:** Ctrl+_
Ctrl+<	**Toogle Split Editor** (Vertical) - since Luna	**old:** Ctrl+{
Ctrl+⇧Shift+"Num Keypad /"	Collapse all code blocks	
Ctrl+⇧Shift+"Num Keypad *"	Open all code blocks	
⇧Shift (and hover over element)	Show source (in tooltip)	settings: Window → Preferences → Java → Editor → Hovers

Outline View

Reorder class members with Drag & Drop	Disable "Sort Order" - otherwise not possible
	Automatic reordering: Select the class, Source → Sort Members

Package / Project Explorer

Alt+F5	Maven → Update Project...

Problems View

Configure to view only warnings and errors associated with a particular resource or group of resources	Problems View - Configure Contents e.g. select "on any element in same project" to restrict to current selected project

Templates (Java → Editor → Templates)

test	Created test method
for	Creates for loop
new	Creates new instance
try	Creates try..catch

Templates (Java → Editor → Templates) (cont)

sysout, syserr, systrace	Creates System.out...	
strf	Format string	here
if	if null/if not null	
for	iterate over **map**	
static_final	static final field	
before	**junit** before method	
after	**junit** after method	
beforeclass	**junit** beforeclass method	
afterclass	**junit** afterclass method	
quote	surround selection with quotes	here

Tip: It is possible to select text and use completion (Ctrl+␣) including typing of template name

Changed Eclipse Keybindings

Windows → Preferences → General → Keys

Required modifications due to the german QWERTZ keyboard layout:

1. Add Block Comment - **new:** Ctrl+⇧Shift+7, **old:** Ctrl+⇧Shift+/
2. Remove Block Comment - **new:** Ctrl+⇧Shift+ß, **old:** Ctrl+⇧Shift+\
3. Toogle Split Editor (Horizontal) - **new:** Ctrl+⇧Shift+-, **old:** Ctrl+_
4. Toogle Split Editor (Vertical) - **new:** Ctrl+⇧Shift+<, **old:** Ctrl+{

Update 12/2016: had to add ⇧Shift to 3. and 4. because of new Eclipse Neon Zoom functionality with Ctrl+-

Project Specific Configuration

1. Import project **Java formatter configuration**

2. Download **Checkstyle configuration** file from sonar server and reference the config file as a file resource url

3. Download **Findbugs configuration** file from sonar server and add the config file under "include filter files"

4. Download **PMD configuration** file from sonar server and first "clear all" existing rules and then import the config file

Plugins

Testing Support	moreUnit	
	Infinitest	
Test Coverage	EclEmma	
Source Code Analyzer	FindBugs	
	Checkstyle	
	PMD	Update-Site
	Usus	

Plugins (cont)

	⚡ SonarCube or SonarLint	Ignore certain warnings with //NOSONAR or @SuppressWarnings("squid:S00112")
Refactoring	JDeodorant	
Subversion Eclipse Team Provider Plugin	Subclipse	
	use Subclipse instead of Subversive because it is **much faster**	
	use JavaHL native/JNI provider	
Task based working	Mylyn	
	Nice: Task-focused Change Sets	
	Subclipse includes an optional Mylyn connector	
JavaDoc	JAutoDoc	
Maven integration	M2Eclipse (m2e)	
	add needed m2e connectors. follow pom.xml hints/problems. e.g:	m2e project configurators for Eclipse WTP
		m2e connector for build-helper-maven-plugin
		m2e connector for the Eclipse JDT Compiler
		m2e connector for the Maven Dependency Plugin
		Groovy-Eclipse m2e integration
Decompiler	Eclipse Class Decompiler	

XUbuntu specific Configuration Changes

Delete following keybindings in **XFCE Settings → Window Manager → Keyboard**:
1. Ctrl+Alt+↓/↑
2. Ctrl+F6
3. Alt+F5

If running **IBus-Applet** delete keybinding definition for "Ctrl+Space" (Content Assist) in IBus-Settings

Use **gnome-color-chooser** and change options:
1. **This only works for Eclipse < 4.5, because Eclipse Mars 4.5 and higher uses GTK3 by default (gtkrc-2.0 settings will not be used).** You can force Eclipse 4.5 and higher to use GTK2 with the following export or change GTK3 theme options: export SWT_GTK3=0
2. Tooltip colors (background: yellow, foreground: black)
3. Global colors → Input field selected (foreground: black)

Disable "pink/purple highlights" in Xubuntu 14.10 and higher: disable custom hightlight-color in **gtk-theme-config**

Settings

show "." files in package explorer	**"Package Explorer → View Menu → Filters → uncheck .*"** or "Package Explorer → Customize View… → Filters → uncheck .*"	
Package Explorer Option "Link with editor"	Link file to package explorer	
Favorites Imports	Windows → Preferences → Java → Editor → Content Assist → Favorites, New Type...	x^1
	Add types for Assertion library Hamcrest and/or AssertJ, Mockito and JUnit-Assert	

Settings (cont)	
Show line numbers	Window → Preferences → General → Editors → Text Editors → "Show line numbers"
Show print margin	Window → Preferences → General → Editors → Text Editors → "Show print margin"
Paste multiline strings	Window → Preferences → Java → Editor → Typing → "Escape text when pasting into a string literal"
Import code templates collection	**x²**
Import code template "surround with quotes"	**x²**
MoreUnit Configuration	Test source folder: "src/test/java"
	Test Type: "Junit4"
	uncheck "Use test-prefix for test-methods"
	Rule for naming test files → Pattern: ${srcFile}(Test|IT|ITCase)
	if you have legacy tests with Test prefix: Rule for naming test files → Pattern: (Test)${srcFile}(Test|IT|ITCase)

Settings (cont)		
Formatter Options	Window → Preferences → Java → Code Style → Formatter → Off/On Tags **for //@formatter:off and //@formatter:on**	here
	Check the **Never join lines** preference accessible on the Line Wrapping and the Comments tabs of the Java Formatter preference page	here
Improve Performance	Window → Preferences → Validation, Disable All	
	Add **-Xverify:none** to eclipse.ini	tell the VM not to validated all the .class files it is loading
	Window → Preferences →General → Startup and Shutdown	remove plugins not needed
	increase -Xms and -Xmx heap memory options (start and max) in eclipse.ini	e.g. -Xms512m -Xmx2048m
	increase -XX:MaxPermSize option (if java < 1.8)	e.g. -XX:MaxPermSize=512m
Enable informative Eclipse Compiler Warnings	Window → Preferences → Java → Compiler → Errors/Warnings	**set to Warning and keep sub checkboxes checked (if any)**
	Potential Programming problems	Potential resource leak
	Unnecessary code	Value of parameter is not used

Settings (cont)

		Unused type parameter
		Unnecessary declaration of thrown exception
		Unnecessary 'else' statement
		Unnecessary cast of 'instanceof' operation
	Generic types	Redundant type arguments (1.7 and higher)
	Annotations	Unhandled token in '@SuppressWarnings' set to **"Ignore"**
	Null analysis	Redundant null check
Indent using spaces	Window → Preferences → XML Files → Editor	Select "Indent using spaces" and set "Indentation Size: 4"
	Window → Preferences → General → Editors → Text Editors	Set "Displayed tab width: 4" and select "Insert spaces for tabs"
Save Actions	Window → Preferences →Java →Editor → Save Actions	
	check "Perform the selected actions on save"	
	check "Format source code", "Format all lines"	
	check "Organize imports"	
	check "Additional actions" and **add the following:**	

Settings (cont)

Code Style	• *"Use blocks in if/while/for/do statements"*	
	• *Use lambda where possible*	**if you use Java >= 8**
Add final modifier to private fields*	• *Add final modifier to private fields*	
	• *Add final modifier to local parameter*	
	• *Add final modifier to local variables*	
Unused Code	• *Remove unused imports*	
	• *Add missing '@Override' annotations*	
	• *Add missing '@Override' annotations to implementations of interface methods*	
	• *Add missing '@Deprecated' annotations*	
	• *Remove unnecessary casts*	
	• *Remove unnecessary '$NON-NLS$' tags*	
Limit maximum number of open editors	Window → Preferences → General → Editors	
	check "Close editors automatically"	
	choose value (e.g. 20) for "Number of opened editors before closing"	

Settings (cont)

JDT Core/Null Analysis to avoid NPEs	Window → Preferences → Java → Compiler → Errors/Warnings, Section *Null analysis*	if you use Java < 8 (cannot use Optional) and cannot use Guava Optional e.g. for legacy projects
	check "Enable annotation-based null analysis"	you can use the Eclipse annotations, the JSR-305 annotations (old), custom ones or the JSR-308 annotations from Checker Framework
Maven	Window → Preferences → Maven → User Interface	
	check "Open XML page in the POM editor by default"	**Reason:** show pom.xml source code is faster than loading the "Overview" tab all the time
JUnit	Window → Preferences → Java → JUnit enable "Add 'ea' to VM arguments when creating new JUnit launch configuration"	enable JVM assertions by default
Show heap memory status	Window → Preferences → General	enable "Show heap status"
Enable "-Quick Diff"	Window → Preferences → General → Editors → Text Editors → Quick Diff	enable checkboxes and select correct reference source option

x¹: TDD-Style mockito: org.mockito.Mockito, BDD-Style mockito: org.mockito.BDDMockito, org.junit.Assert, org.hamcrest.Matchers, org.assertj.core.api.Assertions
x²: Java → Editor → Templates

General

Refresh **all** Projects (with some closed in between)	In **Package Explorer**	Ctrl+A, F5
Use Working Sets for better performance	Package Explorer → open the drop-down menu → Select Working Set	
	If Ctrl+⇧Shift+R and Ctrl+⇧Shift+T does not reflect active Working Set check the small arrow on the top right of the dialogue to select Working Set	

Troubleshooting Projects

Maven dependency issues	**try without eclipse project resolution**
	clean projects
	Error message `The archive: <pa th_ to_ jar _` `does not exist.`

Troubleshooting Projects (cont)

you can use this command to search for the projects referencing old jar: `find -name ' .clas spath' -exec grep -Hin 'examp le- 1.2.3.jar' {} \;`