

Shortcuts

Öffnen und Umschalten von Ansichten

Strg-Tab (Strg-`)	Wechsel zwischen offenen Dokumenten in Nutzungsreihenfolge
Shift-Escape	Fenster vergrößern
Strg-F4/Strg-W	Gewähltes Fenster schließen
Strg-Shift-F4	Alle Fenster schließen
Shift-F10	Kontextmenü öffnen
Strg-PgHoch / PgRunter	Wechsel zwischen offenen Dokumenten in Tabreihenfolge
Strg-Alt-T	Kürzlich geschlossene Datei öffnen
Strg-Alt-PgHoch / PgRunter	Umschalten zwischen Editortypen
Alt-Mouse Wheel Hoch / Runter	In den Text rein/raus zoomen
Strg-Shift-S	Inspiziermodus wechseln

Finden, Suchen und Ersetzen

Strg-F3	Wort an einzufügender Stelle suchen
F3/Shift-F3	Nächstes/Vorheriges in Dokument finden
Strg-F/H	Finden/ersetzen in Dokument
Alt-F7	Verwendungen finden
Strg-Shift-F/H	Finden/ersetzen in Projekten
Alt-Shift-U	Ergebnisse von Verwendungen Suchen
Alt-Shift-H	Hervorhebungen von Suchergebnissen ausschalten
Strg-R	Umbenennen
Strg-U, dann U	Auswahl großschreiben
Strg-U, dann L	Auswahl kleinschreiben
Strg-U, dann S	Schreibung der Auswahl wechseln
Strg-Shift-V	Formatiertes einfügen
Strg-Shift-D	Chronik der Zwischenablage anzeigen
Strg-I	Zur Schnellsuche wechseln
Alt-Shift-L	Dateipfad kopieren

Durch den Quellcode navigieren

Strg-O/Alt-Shift-O	Zu Typ/Datei gehen
Strg-Shift-T	Zu JUnit-Test gehen
Strg-Shift-B	Zu Ursprung gehen
Strg-B	Zu Deklaration gehen
Strg-G	Zu Zeile gehen
Strg-Shift-M	Zwischen Lesezeichen Hinzufügen/Entfernen wechseln
Strg-Shift-Punkt /Comma	Nächstes/vorheriges Lesezeichen

Strg-Punkt /Comma	Nächste/vorherige Verwendung/ Kompilierungsfehler
Alt-Shift-Punkt /Comma	Nächstes/vorheriges Element wählen
Strg-Shift-1/2/3	In Projekten/Dateien/Favoriten wählen
Strg-[Wechseln des Textcursors zur passenden Klammer
Strg-K/Strg-Shift K	Nächste/vorherige Wortübereinstimmung
Alt-Links/Alt-Rechts/Strg-Q	Zurück/vor zu letzter Bearbeitung gehen
Alt Hoch / Runter	Nächstes/vorheriges markiertes Auftreten

Coding in Java

Alt-Insert	Code generieren
Strg-Shift-I	Alle Klassenimporte fixen
Alt-Shift-I	Importe der gewählten Klasse fixen
Alt-Shift-F	Formatauswahl
Alt-Shift Links/Rechts/ Hoch/Runter	Zeilen nach links/rechts/oben/unten verschieben
Strg-Shift-R	Rechteckige Auswahl
Strg-Shift-Hoch/D	Zeilen nach oben/unten kopieren
Strg/Alt-F12	Element/Hierarchie inspizieren
Strg-Shift-C/ Strg- /	Kommentarzeilen hinzufügen/entfernen
Strg-E	Aktuelle Zeile löschen

Kompilieren, testen und ausführen

F9	Paket/Datei kompilieren
F11	Hahochprojekt bauen
Shift-F11	Hahochprojekt säubern und bauen
Strg-Q	Abfrageparameter setzen
Strg-Shift-U	Unit-Test erstellen
Strg-F6/Alt-F6	Unit-Test auf Datei/Projekt ausführen
F6/Shift-F6	Hahochdatei/Projekt ausführen

Debugging

Strg-F5	Fehlersuche in Hahochprojekt starten
Strg-Shift-F5/ F6	Fehlersuche/Test in aktueller Datei starten
Shift-F5/F5	Fehlersuche stoppen/fortfahren
F4	Zu Cursorposition in Datei springen
F7/F8	Reinspringen/überspringen
Strg-F7	Rausspringen

Strg-Alt-Hoch/Runter	Zu aufgerufenen/aufrufenden Methode springen
Strg-F9	Ausdruck evaluieren

Strg-F8	Haltepunkt schalten
Strg-Shift-F8/ F7	Neuer Haltepunkt/Beobachtungspunkt

Code Vervollständigungen

En	Enumeration
Ex	Exception
Ob	Object
Psf	public static final
Psfb	public static final boolean
Psfi	public static final int
Psfs	public static final String
St	String
ab	abstract
bcom	/**/
bo	boolean
br	break;
ca	catch (
cl	class
db	double
dowhile	do {} while (condition);
eq	equals
ex	extends
fa	false
fi	final
fl	float
forc	for (Iterator it = collection.iterator(); it.hasNext();) {Object elem = (Object) it.next();}
fore	for (Object elem : iterable) {}
fori	for (int i = 0; i < arr.length; i++) {}
forl	for (int i = 0; i < lst.size(); i++) { Object object = lst.get(i); }
forst	for (StringTokenizer st = new StringTokenizer(""); st.hasMoreTokens();) {}
forv	for (int i = 0; i < vct.size(); i++) {Object object = vct.elementAt(i);}
fy	finally{ }
ie	interface
ifelse	if (condition){}else {}
iff	if (exp) {}
im	implements
inst	if (exp instanceof Object) {Object obj = (Object) exp;}
iof	instanceof
ir	import
le	length
newo	Object name = new Object (args);
pe	protected

pr	private
psf	private static final
psfb	private static final boolean
psfi	private static final int
psfs	private static final String
psvm	public static void main(String[] args){}
pu	public
re	return
serr	System.err.println (" ");
sout	System.out.println (" ");
soutv	System.out.println("Object = " + Object);
st	static
su	shoher
sw	switch (var) { case val: break; default: throw new AssertionError();}
sy	synchronized
th	throws
trycatch	try {} catch (Exception e) {}
tw	throw
wh	while (
whileit	while (it.hasNext()) {Object elem = (Object) it.next();}
whilen	while (en.hasMoreElements()) {Object elem = (Object) en.nextElement();}
whilexp	while (exp) {}