

IntelliJ Cheatsheet

Wenn von «Cursor» gesprochen wird, ist hiermit **NICHT** der Mauszeiger gemeint, sondern der blinkende Strich der zeigt, wo Buchstaben auftauchen sollen wenn man tippt.

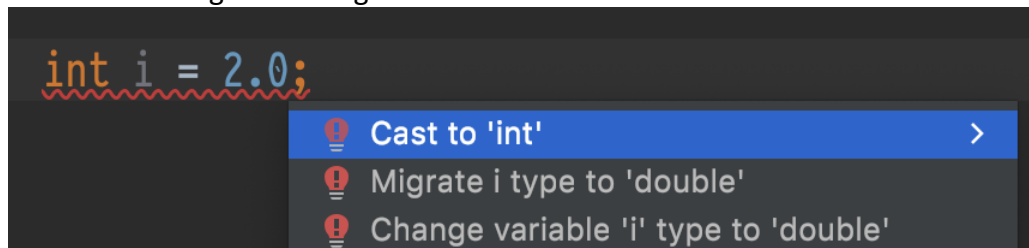
Bei Shortcuts werden zur besseren Lesbarkeit Grossbuchstaben gezeigt, es ist aber nicht gemeint, dass die Shift-Taste gedrückt werden muss, ausser im Shortcut wird explizit «Shift» erwähnt.

Abkürzungen im Code Editor

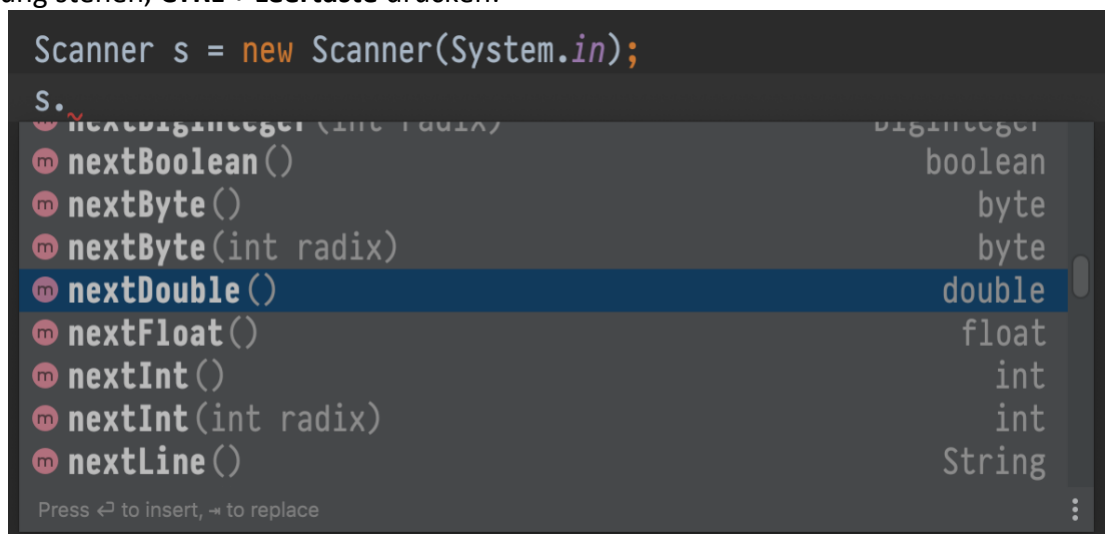
- **psvm** tippen, mit Enter-Taste bestätigen erstellt main-Methode
- **sout** tippen, mit Enter-Taste bestätigen erstellt `System.out.println();`
- **fori** tippen, mit Enter-Taste bestätigen erstellt for-Schleife mit Index
 - danach jeweils Eingaben machen und mit Enter bestätigen um der Reihe nach Variablennamen, Bedingung und Schleifenkörper zu definieren.
- **foreach** tippen, mit Enter-Taste bestätigen erstellt for-each-Schleife
 - danach jeweils Eingaben machen und mit Enter bestätigen um das vor dem Doppelpunkt, nach dem Doppelpunkt und Schleifenkörper zu definieren.

Allgemeines im Umgang mit IntelliJ

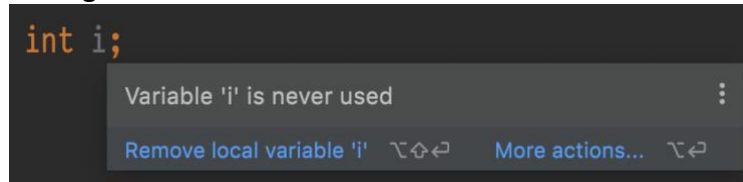
- **Alt + Enter** drücken zeigt Vorschläge an:



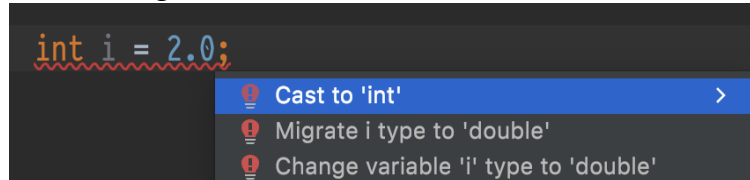
-
- Für Autovervollständigung, z. B. um zu sehen was beim Scanner für Methoden zur Verfügung stehen, **CTRL + Leertaste** drücken:



- Hinweise von IntelliJ (bei Verwendung Color Scheme «Darcula»)
 - **Grauer Text:** Hinweis auf eine Verbesserungsmöglichkeit im Code, Erklärung und Vorschläge bei Hover.



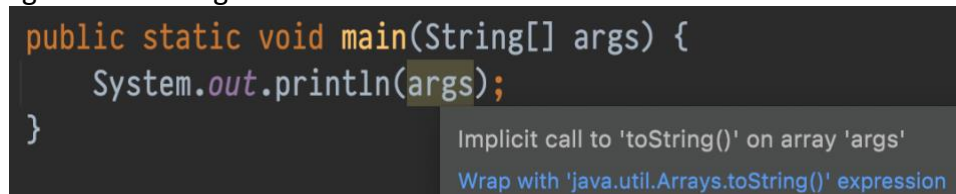
- **Rot unterstrichen:** Es wurde ein Fehler erkannt. **ALT + Enter** drücken zeigt mögliche Behebung des Problems.



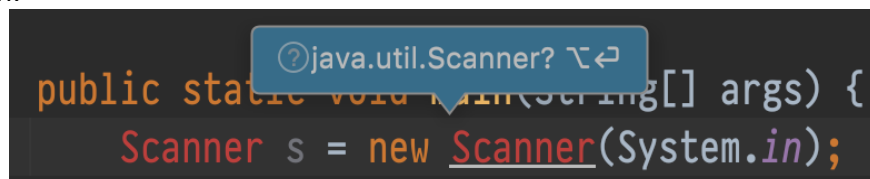
- **Teil des Texts ist rot unterstrichen** Fehler, bei Hover Erklärung und Vorschlag.



- **Gelb markiert:** Warnung, weist auf mögliche Probleme hin, bei Hover Erklärung und Vorschlag.



- **Rote Schrift:** das Geschriebene ist unbekannt, wenn blaue Sprechblase sichtbar dann fehlt ein `import`, dieser kann über **ALT + Enter** hinzugefügt werden.



- UML-Diagramm aus Code generieren
 - Rechtsklick auf das Package mit den Klassen zu denen man das UML-Diagramm generieren will
 - Diagrams → Show Diagram → Java classes diagram

- Auf mehreren Zeilen gleichzeitig Text schreiben / ändern
 - Wenn man mehrere Zeilen direkt untereinander bearbeiten möchte:
 - **Alt-Taste** oder **mittlere Maustaste** gedrückt halten
 - Bei erster Zeile im Code mit der **linken Maustaste** klicken und gedrückt halten
 - Maus nach unten bewegen, bis man bei der letzten Zeile ist
 - **Linke Maustaste** und **Alt-Taste** oder **mittlere Maustaste** wieder loslassen
 - Nun sollten mehrere Cursor sichtbar sein, tippt man nun, tippt man mit allen Cursors gleichzeitig, tippt man auf eine Pfeiltaste bewegt sich der Cursor bei allen gleichzeitig.

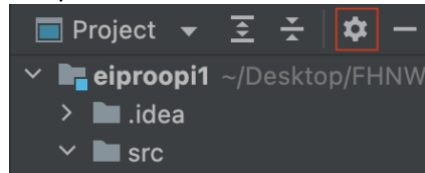
```
int var0 = 1;
int var1 = 2;
int var2 = 3;
int var3 = 4;
```

- Wenn man mehrere Zeilen bearbeiten möchte, die **NICHT** direkt untereinander liegen:
 - Cursor an erster Zeile platzieren
 - **Alt + Shift** gedrückt halten und mit der **linken Maustaste** dorthin klicken, wo man einen zusätzlichen Cursor haben will.

```
int v0 = 1;
int var1 = 2;
int variable2 = 3;
```

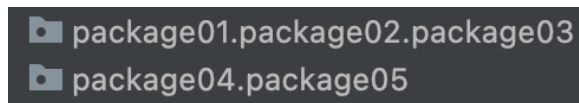
- Wie generiert man Getter & Setter, Konstruktoren usw.?
 - Oben in der Leiste auf **Code > Generate** klicken, dann auswählen was man möchte.

- Wie ändert man die Darstellung von mehreren verschachtelten Packages?
 - Links bei «Project» (bei den Ordnern) auf das Zahnrad-Icon (rot markiert im Screenshot) klicken:

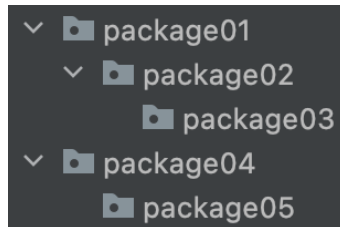


- Dann auf «**Tree Appearance**» gehen
 - Sicherstellen, dass bei «**Flatten Packages**» **kein** Häkchen ist (sonst entfernen)
 - Je nach dem welche Darstellung man bevorzugt, Häkchen bei «**Compact Middle Packages**» setzen oder entfernen (Empfehlung: Häkchen entfernen)

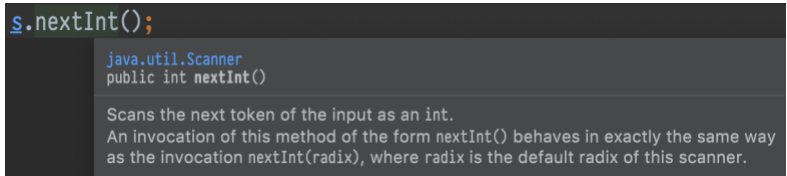
- Mit Häkchen:



- Ohne Häkchen:



Shortcuts

- JavaDoc anschauen
 - Cursor positionieren bei Methode, Klasse etc.
 - Windows / Linux: **Ctrl + Q**
 - macOS: **Ctrl+ J**
- 
- - Code formatieren
 - Windows / Linux: **Ctrl + Alt + L**
 - macOS: **Alt + Cmd + L**
 - Mehrere Zeilen Code gleichzeitig kommentieren / auskommentieren
 - Zeilen markieren, dann oben auf «**Code**» klicken
 - Entweder auf «**Comment with Line Comment**» klicken, um jede Zeile einzeln mit `//` zu kommentieren
 - Oder auf «**Comment with Block Comment**» klicken, um die Zeilen mit `/*` und `*/` zu kommentieren
 - Tipp: In den IntelliJ Einstellungen unter «**Keymap**» nach «**Comment with Line Comment**» und «**Comment with Block Comment**» suchen und eigene Shortcuts für diese definieren, die Standard-Shortcuts sind mit nicht-amerikanischer Tastatur nur schwer auszuführen.
 - Zeile duplizieren
 - Windows / Linux: **Ctrl + D**
 - macOS: **Cmd + D**
 - Aktuelle Zeile mit nächster Zeile vertauschen
 - Windows / Linux: **Ctrl + Shift + Pfeil nach unten**
 - macOS: **Cmd + Shift + Pfeil nach unten**
 - Aktuelle Zeile mit vorheriger Zeile vertauschen
 - Windows / Linux: **Ctrl + Shift + Pfeil nach oben**
 - macOS: **Cmd + Shift + Pfeil nach oben**
 - Variablenname und alle Verwendungen des Variablennamens gleichzeitig ändern
 - **Rechtsklick** auf Variablenname, im Kontextmenü auf «**Refactor**», dann auf «**Rename...**» klicken (**Shift + F6**)
 - Neuen Namen eingeben und mit Enter bestätigen.
 - Als Methode extrahieren
 - Code markieren, aus dem Methode werden soll
 - Rechtsklick → Refactor → Extract Method (macOS: **Alt + Cmd + M**, Windows / Linux: **Ctrl + Alt + M**)
 - Name der Methode eingeben und mit Enter bestätigen
 - Zu Test-Klasse gehen
 - Windows / Linux: **Ctrl + Shift + T**
 - macOS: **Cmd + Shift + T**

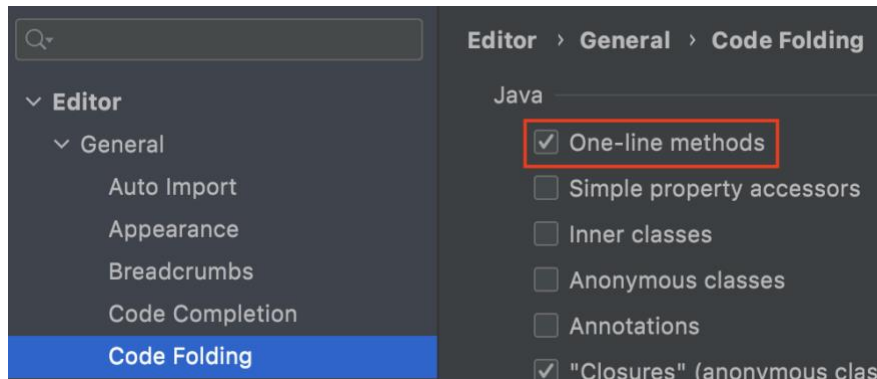
Einstellungen

Die folgenden Anweisungen beziehen sich auf das «Preferences» Fenster in IntelliJ, welches man auf Windows mit **Ctrl + Alt + S** und auf macOS mit **Cmd + ,** öffnen kann.

- Wie kann ich deaktivieren, dass IntelliJ einzeilige Methoden «einklappt»?

```
public int getColor() { return color; }
```

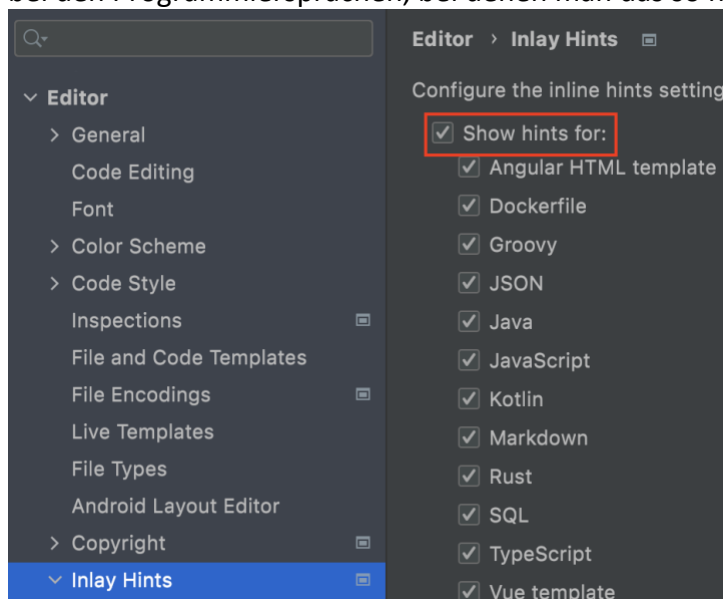
-
- Unter **Editor > General > Code Folding** bei der Option **“One-line methods”** das Häkchen entfernen:



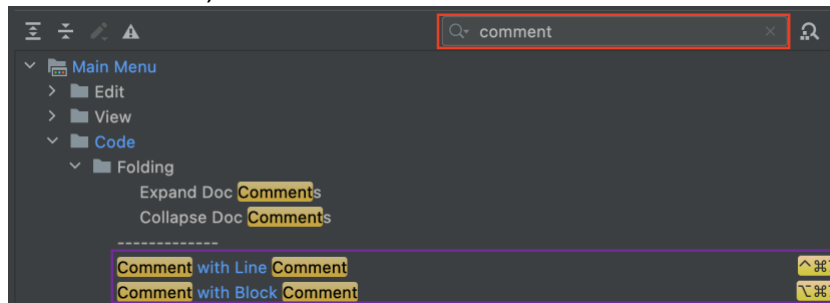
-
- Wie kann ich deaktivieren, dass mir IntelliJ bei Methodenaufrufen einen Hinweis anzeigt, wie die Parameter heissen (rot markiert im folgenden Bild):

```
Fahrzeug f1 = new Fahrzeug(name: "Fahrzeug1", speed: 0);
```

-
- Möchte man das nur in gewissen Situationen deaktivieren, kann man wenn so eine Situation auftritt bei der Codestelle **Alt + Enter** drücken und dann auswählen **«Do not show hints for current method»**
- Möchte man das immer deaktiviert haben, in den Preferences unter **Editor > Inlay Hints** das Häkchen bei **«Show hints for:»** entfernen (oder nur bei den Programmiersprachen, bei denen man das so haben will)



- Wie kann ich Shortcuts ändern?
 - Unter Keymap im Suchfeld (rot markiert) nach dem Shortcut suchen, z. B. nach «**comment**», danach einen **Rechtsklick** machen auf den Shortcut den man ändern möchte, z. B. «**Comment with Line Comment**»:



- Mit «**Add keyboard shortcut**» kann man einen neuen Shortcut hinzufügen.
- Mit «**Remove ...**» kann man Shortcuts löschen, was besonders dann Sinn macht wenn man die bisher zugewiesenen für etwas anderes verwenden möchte.
- Meine Empfehlung für Kommentar-Shortcuts:
 - Die 7 im Shortcut kommt davon, dass der Slash (/) auf der Taste 7 ist
 - «**Comment with Line Comment**» (Kommentar der Zeile(n) mit //)
 - Windows / Linux:
 - Ctrl + Shift + 7
 - Mac:
 - Ctrl + Cmd + 7
 - «**Comment with Block Comment**» (Kommentar der Zeile(n) mit /* */)
 - Windows / Linux:
 - Alt + Shift + 7
 - Mac:
 - Alt + Cmd + 7
- Automatische Imports und Import-Vorschläge von gewissen Packages verhindern
 - Es gibt gewisse Packages, von den man nicht importieren möchte, z. B.:
 - das Package sun welches Java-interne Klassen enthält
 - bei oopI2 wird JavaFX verwendet, welches ähnliche Klassennamen hat wie andere GUI-Frameworks (Swing, AWT)
 - Um zu verhindern, dass IntelliJ den Import von solchen Klassen vorschlägt, unter **Editor > General > Auto Import** im Abschnitt «**Java**» unter «**Exclude from auto-import and completion:**» folgende Einträge hinzufügen:
 - sun.*
 - java.awt.*
 - javax.swing.*

