IntelliJ

* 1. Schnelles Editieren über Shortcuts

|  |  |
| --- | --- |
| **Shortcuts** | **Was passiert?** |
| Doppel-Shift | Fenster zum überall suchen |
| Strg+Shift+A | Kommando suchen und ausführen |
| Alt+Enter | Schnellreparatur bei Fehlern |
| F2 Shift+F2 | Sprung zum nächsten Fehler |
| Strg+E | Fenster mit letzten geöffneten Dateien |
| Strg+Alt+L | Code formatieren |
| Strg+W Strg+Shift+W | Code expandieren |
| Strg+Q | Schnell-Dokumentation |

* 1. Projektmanagement  
       
     Inhalte können mit Git-Plug-In problemlos auf Git hoch- und runtergeladen werden. Updates, die jemand anderer im Git-Projekt gemacht hat, werden auch aktualisiert.  
     Über „Code With Me“-Plug-In kann gleichzeitig am Code gearbeitet werden.
  2. Wichtige Plugins  
       
     Git  
     Code With Me  
     Maven  
     JavaFX  
     Für diverse Sprachen (Groovy, Markdown, YAML, Kotlin)
  3. Code-Funktionen  
       
     Inspektion und Kontextaktionen  
     Intelligente Code-Completion  
     Sprachinjektionen (Code-Abschnitte in anderen Sprachen einfügen)  
     Diagramme  
     Live-Vorlagen (Einfügen von Standardkonstrukten)  
     Projektweites Refactoring
  4. Navigation / Struktur  
       
     Mit Alt+1 wird die Struktur des Projektes am linken Rand angezeigt,
  5. Coderefactoring-Funktionen  
       
     automatische Projektweite Coderefactoring Funktion  
     Im „Refactor“-Reiter sind alle Funktionen.
  6. Code-Style-Funktionen / Formatierung  
       
     Strg+Alt+L automatische Formatierung für geöffneten Bereich  
     Strg+Alt+Shift+L automatische Formatierung für gesamtes Projekt  
     Strg+Alt+S öffnet Einstellungen -> Benutzerspezifische Formatierungen können eingestellt werden
  7. Code Inspektionen  
       
     IntelliJ überprüft Qualität und Gültigkeit des Codes automatisch im Hintergrund. Tipps und Hilfen werden mit Glühbirnen-Symbol oder Alt+Enter angezeigt.
  8. UML-Generierung  
       
     Mit richtigem Plug-In können UMLs generiert werden. (UML-Generator)
  9. Pro Tips  
       
     Strg+Q öffnet die Schnelldokumentation  
     Bei der automatischen Codecompletion kann die Case Sensitivity eingestellt werden.  
     Strg+Leertaste Basis Autovervollständigung  
     Strg+Shift+Leertaste Verbesserte Autovervollständigung  
     Strg+Shift+Enter Statement Autovervollständigung  
     Alt+/ Hippie-Completion. Zeigt alle verfügbaren Autovervollständigungen an  
     mit .if: wird das betreffende Objekt in eine If-Schleife eingebaut -> Postfix  
     Postfixes können auch individuell gestaltet und geändert werden

Visual Studio Code

1. Schnelles Editieren über Shortcuts

|  |  |
| --- | --- |
| **Shortcuts** | **Was passiert?** |
| Strg+, | Benutzer-Einstellungen |
| F1 | Zeigt mögliche Kommandos |
| Strg+Shift+\ | Springt zu dazugehöriger Klammer |
| Strg+Shift+[ Strg+Shift+] | Code expandieren |
| Strg+Shift+M | Zeigt Problem-Fenster |
| F8 Shift+F8 | Sprint zum nächsten Fehler |
| F5 | Startet Programm |

1. Projektmanagement  
     
   Mit Git-Plug-In kann Code über Git-Plattform geteilt werden.  
   Über Remote-SSH kann Visual Studio mit anderen Maschinen verbunden werden.
2. Wichtige Plugins  
     
   Debugger For Java  
   HTML CSS Support  
   JavaScript and TypeScript  
   Maven for Java  
   Spring Boot
3. Code-Funktionen  
     
   Intelligente Code-Komplettierung  
   Code-Navigation  
   Code Refactoring  
   Verschiedene Debugging-Funktionen
4. Navigation / Struktur  
     
   Struktur kann mit Strg+Shift+E am linken Rand angezeigt werden.
5. Coderefactoring-Funktionen  
     
   Strg+. Schnell-Reparatur und -Refactoring  
   Rechtsklick -> Refactor
6. Code-Style-Funktionen / Formatierung  
     
   Shift+Alt+F automatische Codeformatierung  
   Es gibt Plug-Ins für viele Programmiersprachen
7. Code Inspektionen  
     
   Wieder Plug-Ins für viele Programmiersprachen
8. UML-Generierung  
     
   Plug-Ins zur UML-Generierung erhältlich
9. Pro Tips  
     
   Multicursor-Editing -> zum ändern in mehreren Zeilen gleichzeitig  
   Strg+Shift+P -> öffnet Kommando-Palette  
   Benutzerdefinierte Shortcuts können eingestellt werden

/FSE\_LAND/01\_IDEs/Untitled

1. Arbeite dich in JavaDoc ein. Das bedeutet,
   1. du kennst und verwendest alle wichtigen JavaDoc-Funktionen für die Dokumentation von Code,
   2. kannst eine Dokumentation erzeugen und
   3. kannst eine Dokumentation lesen.
2. Alle gelernten Funktionen für IDEs und alle gelernten JavaDoc-Aspekte müssen anhand eines praktischen Projektes praktisch verwendet / ausprobiert / vorgezeigt werden können.
3. Dokumentiere deine Arbeit in angemessener Form (Videotagebuch, Zusammenfassung, Präsentation)