

Programmieren I

Entwicklungsumgebungen Eclipse, NetBeans und IntelliJ IDEA

Institut für Automation und angewandte Informatik

```
final List<String> allResults = new ArrayList<String>();  
final Map<String, Integer> typeWordResultCount = new HashMap<String, Integer>();  
final Map<String, Integer> typePoints = new HashMap<String, Integer>();  
evaluation.put(type, typePoints);  
  
for (final Sheet sheet : this.sheets) {  
    final String sheetResult = sheet.getPlayerInput(type);  
    if (sheetResult.startsWith(start) && this.isValidWord(sheetResult, type)) {  
        validWordCountForType++;  
        allResults.add(sheetResult);  
    }  
}
```

Was ist Programmieren

- Quellcode schreiben

Texteditor

- Der Maschine verständlich machen („kompilieren“)

javac

- Mit Standardbibliotheken verknüpfen („linken“) // nicht in Java

- Erzeugtes Programm ausführen und Fehler beheben („Debuggen“)

java

Vereinfachung: Ein Tool für alles!

- Entwicklungsumgebungen bieten typischerweise Werkzeuge zum
 - Quellcode editieren
 - Code übersetzen
 - Linken
 - Ausführen
 - Debuggen

- Auch meist vorhanden:
 - Syntax-Highlighting
 - „Live“-Anzeige von Syntax-Fehlern
 - Unterstützung in Form verschiedener Code-Generatoren
 - Integration in verschiedene Tools (z.B. Versionskontrollsysteme)

Entwicklungsumgebungen für Java

- Java Development Kit (JDK)

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

- Englisch: *Integrated Development Environment* → IDE

- Eclipse (aktuell: Version 2019-12)

<http://www.eclipse.org/>



- Apache NetBeans (aktuell: Version 12.2)

<https://netbeans.apache.org>

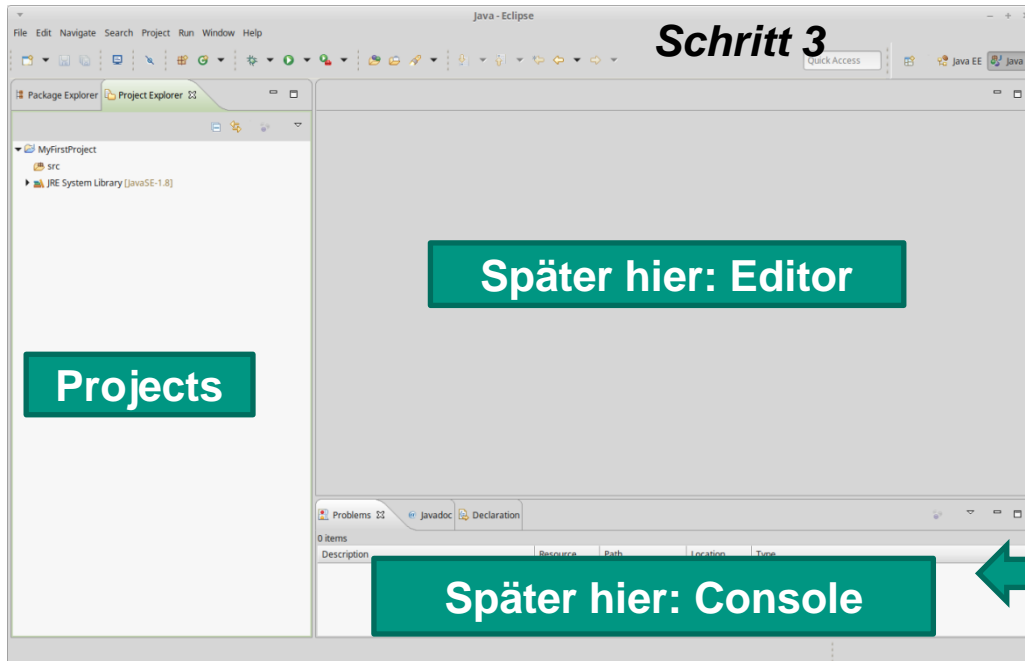
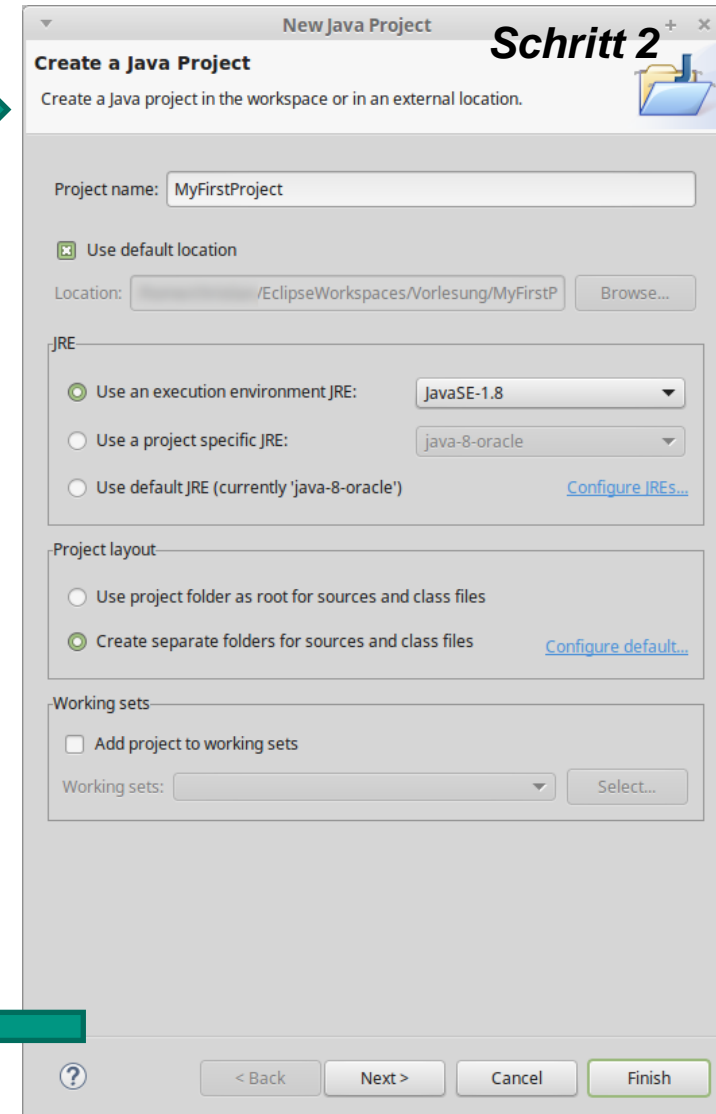
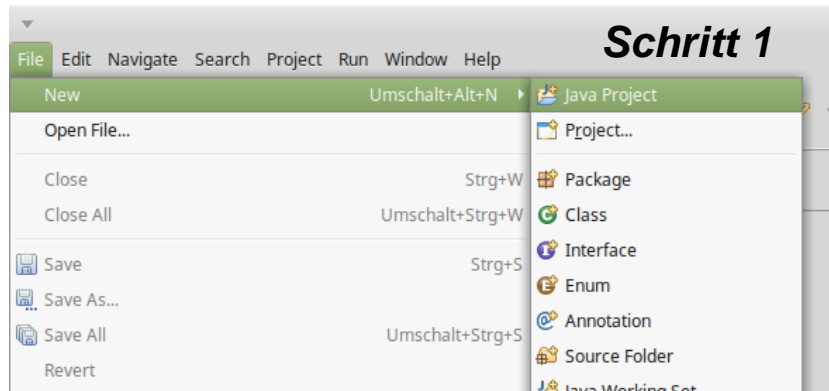


- IntelliJ IDEA (aktuell: Version 2019.3)

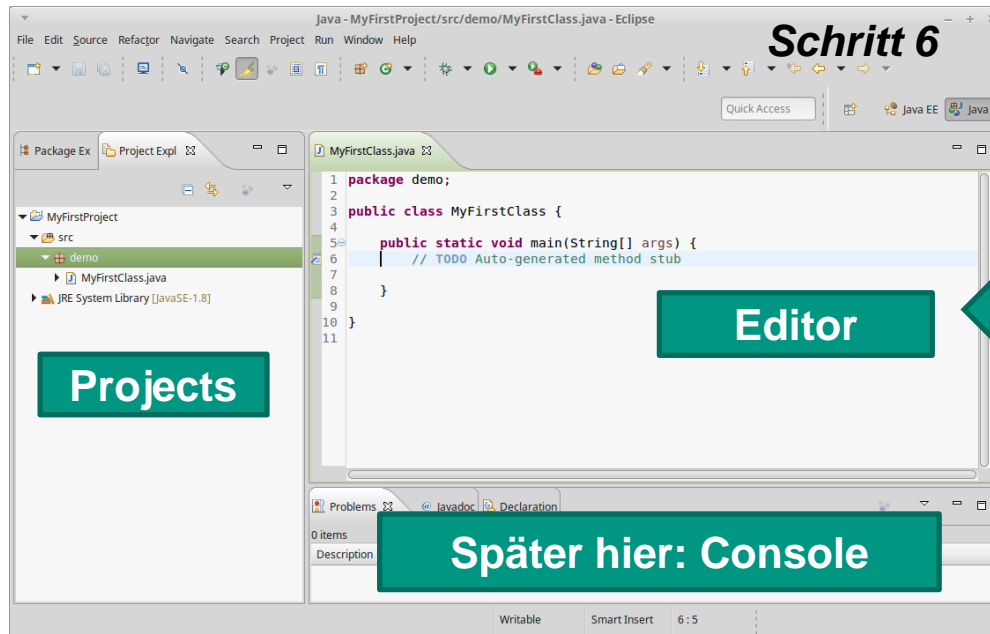
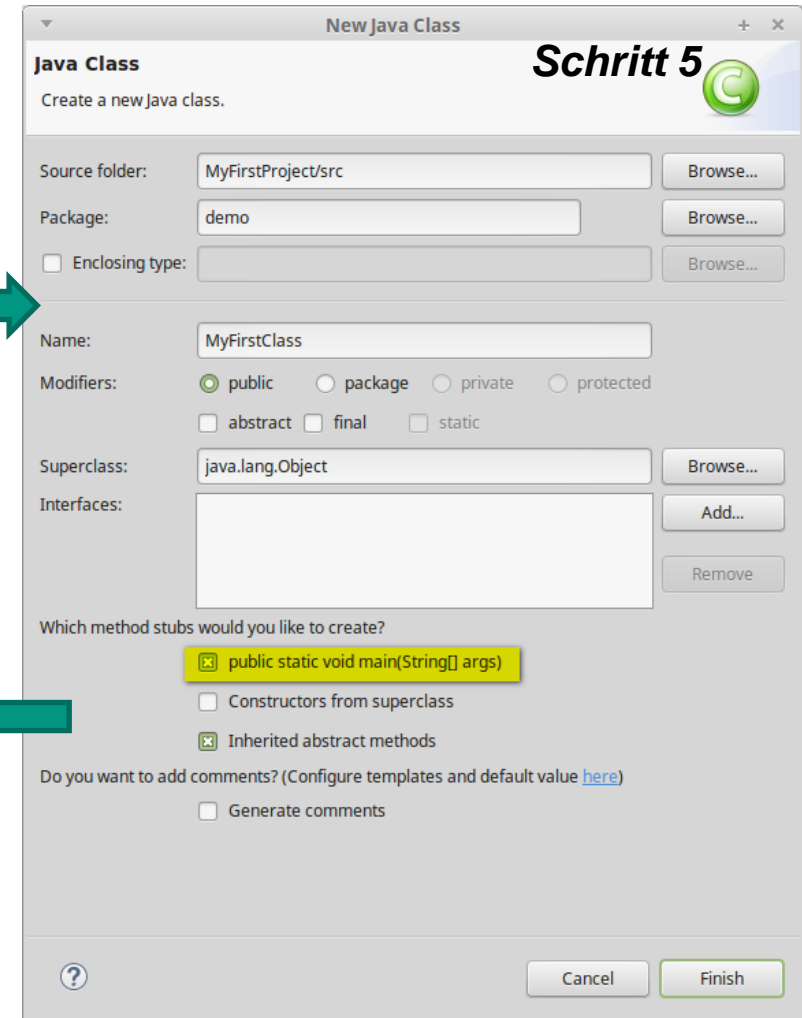
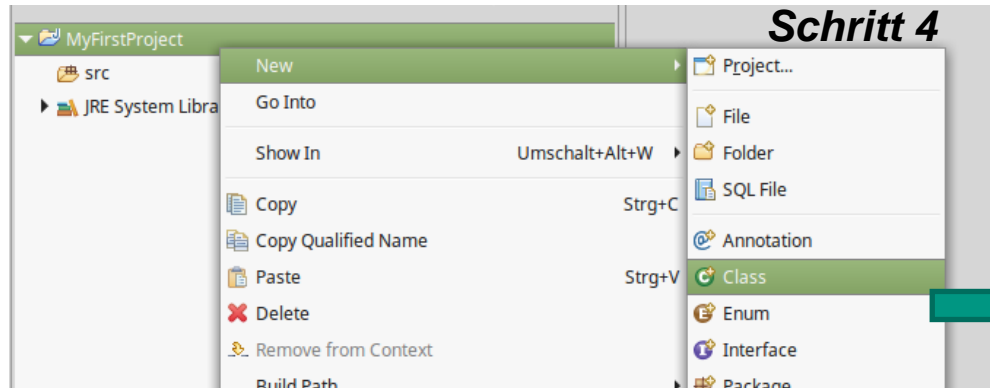
<https://www.jetbrains.com/idea/>



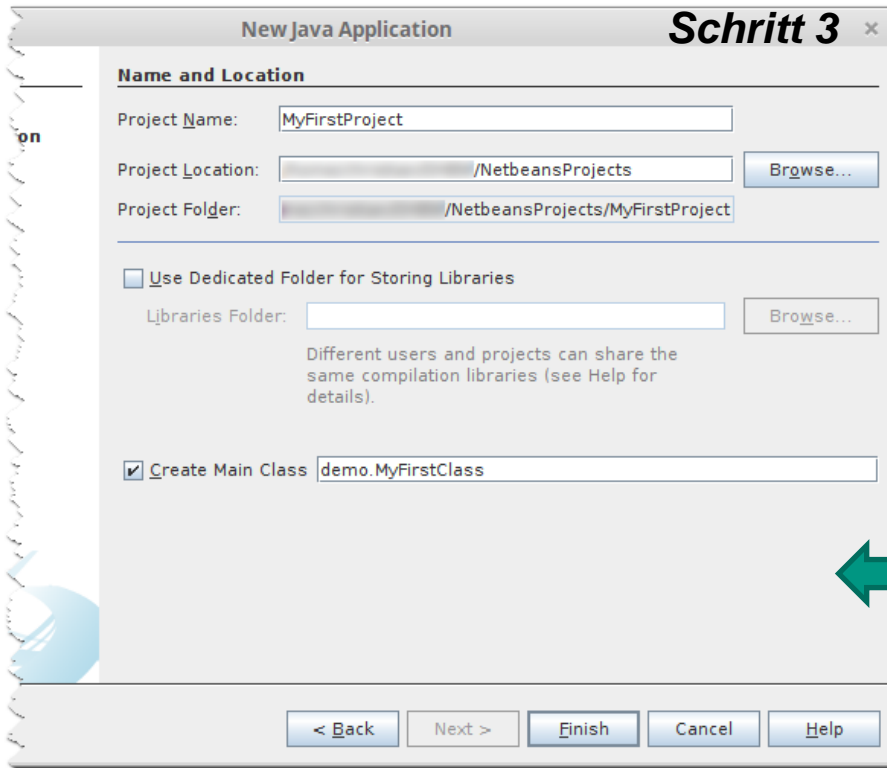
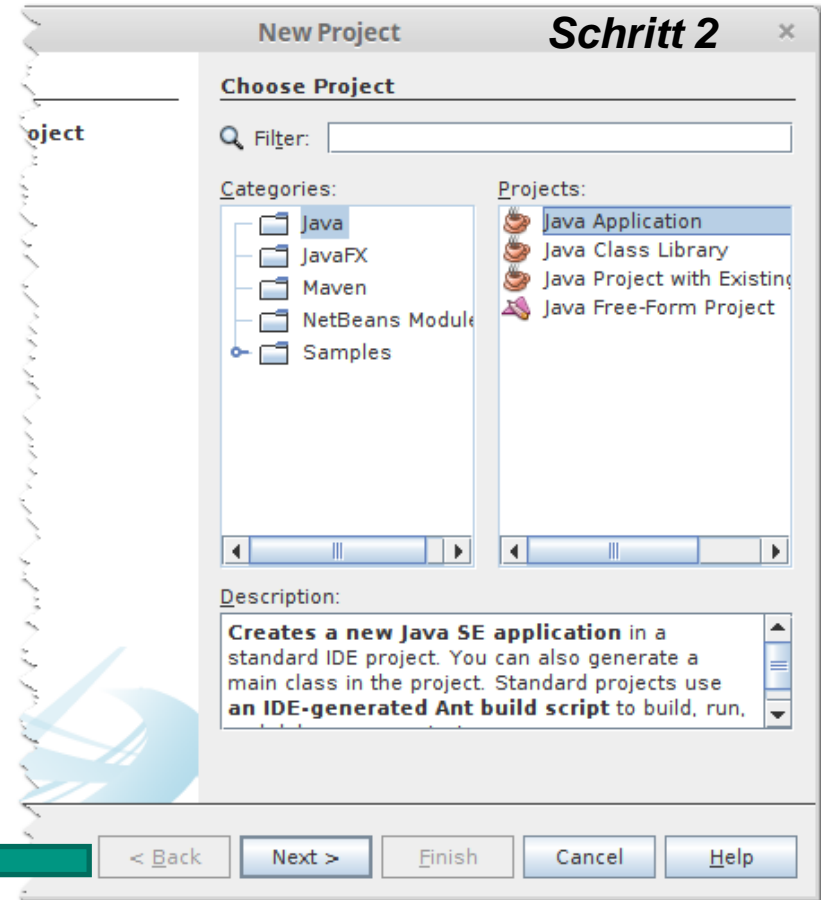
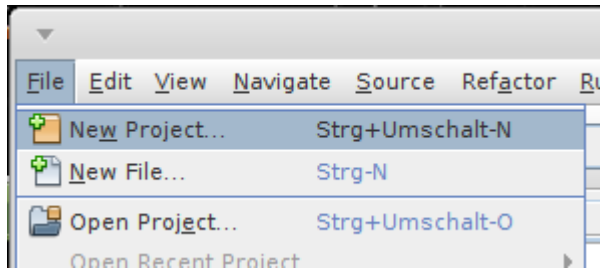
Neues Projekt (1) – Eclipse I



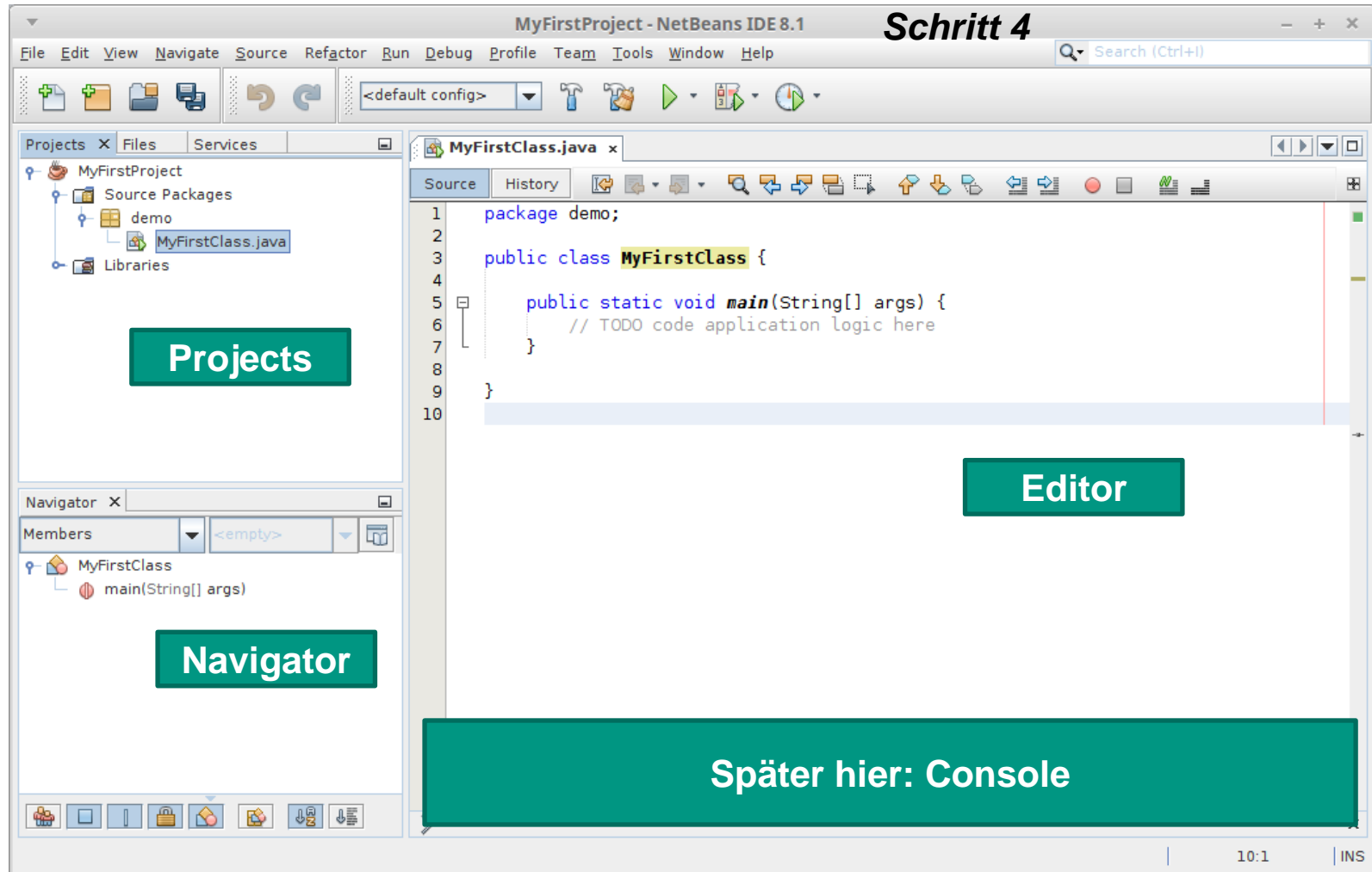
Neues Projekt (2) – Eclipse II



Neues Projekt (3) – NetBeans I

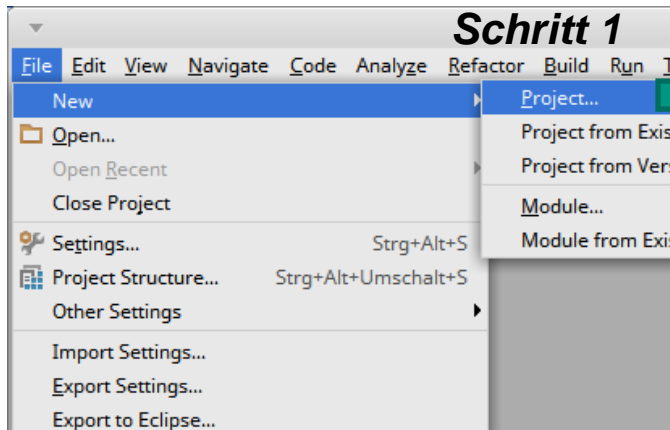


Neues Projekt (4) – NetBeans II

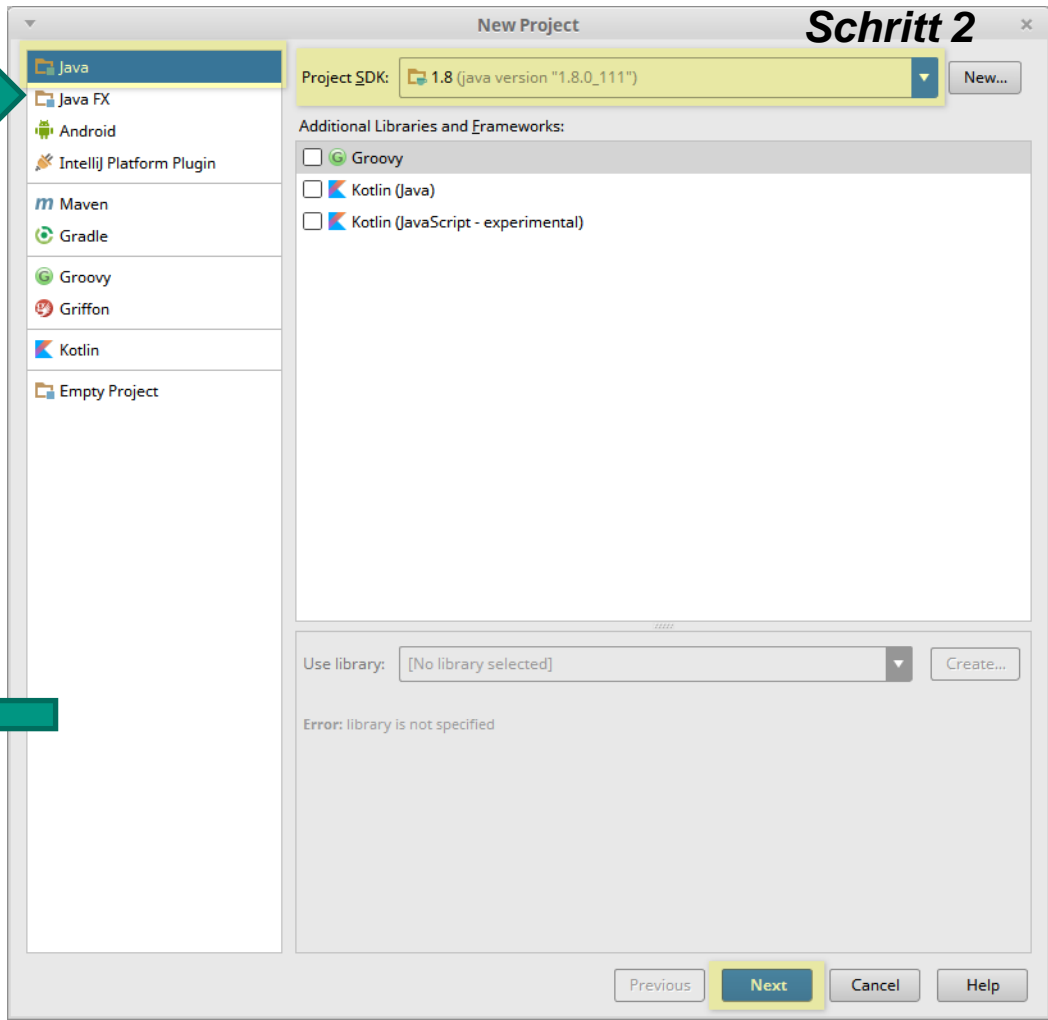


Neues Projekt (5) – IntelliJ IDEA I


Schritt 1



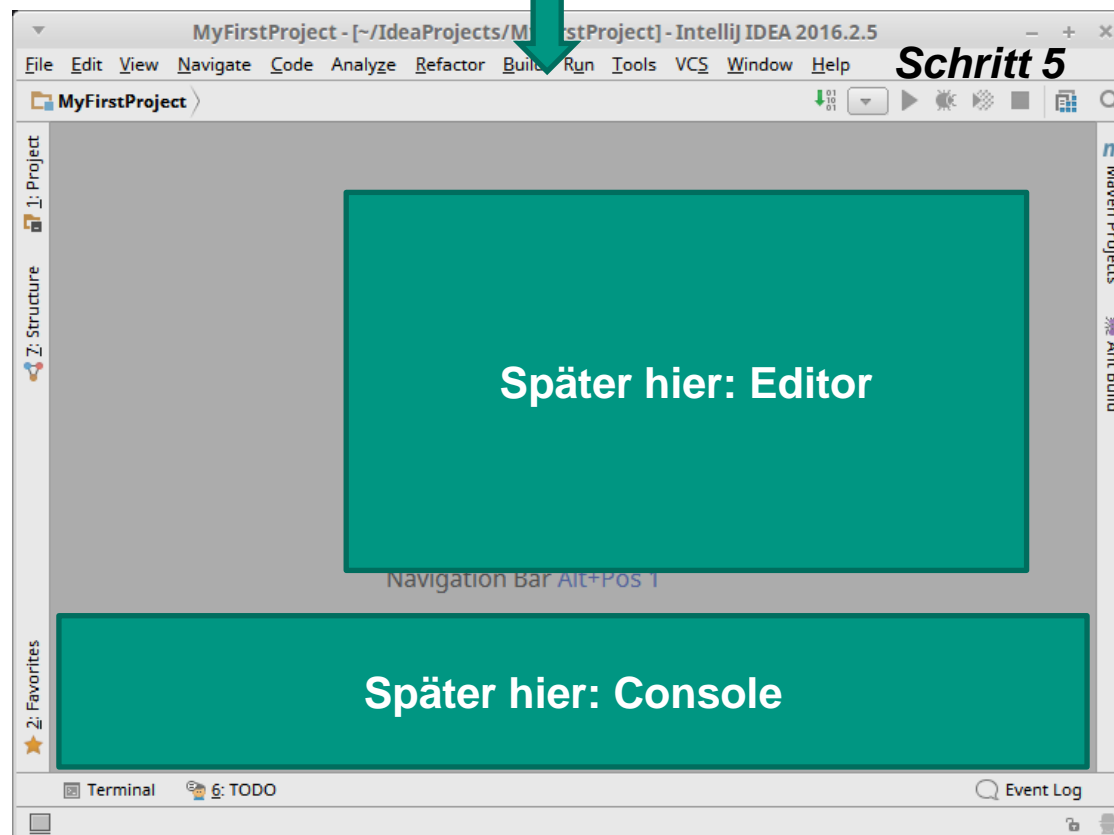
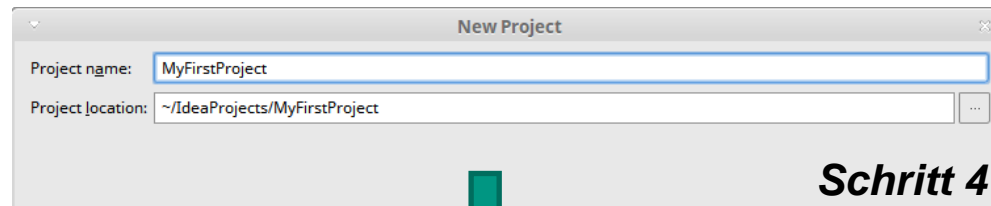
Schritt 2



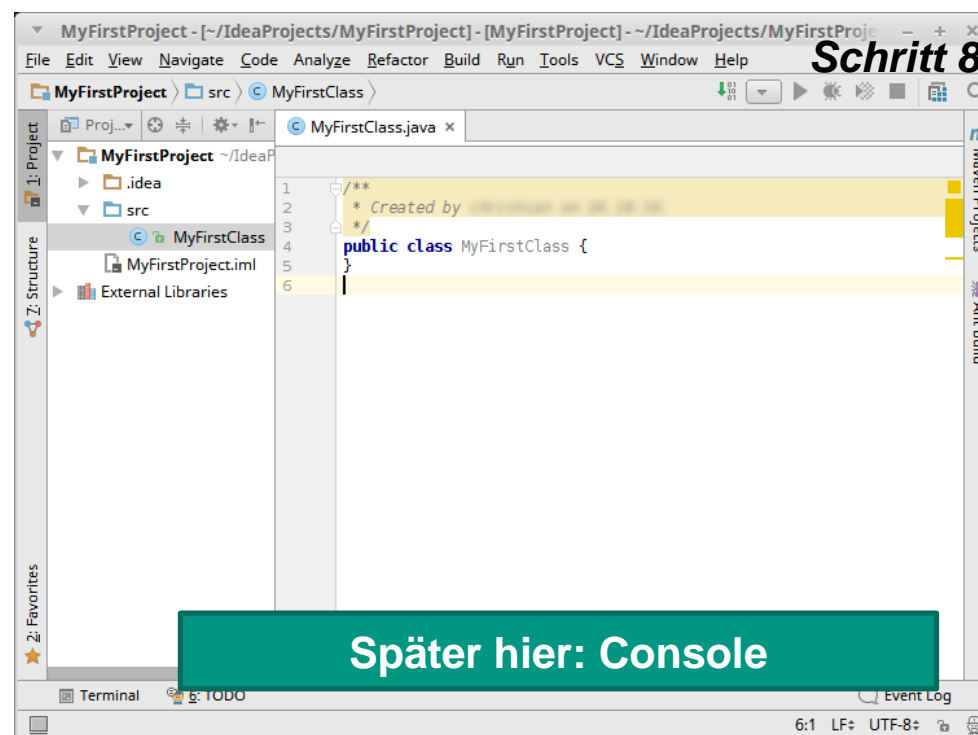
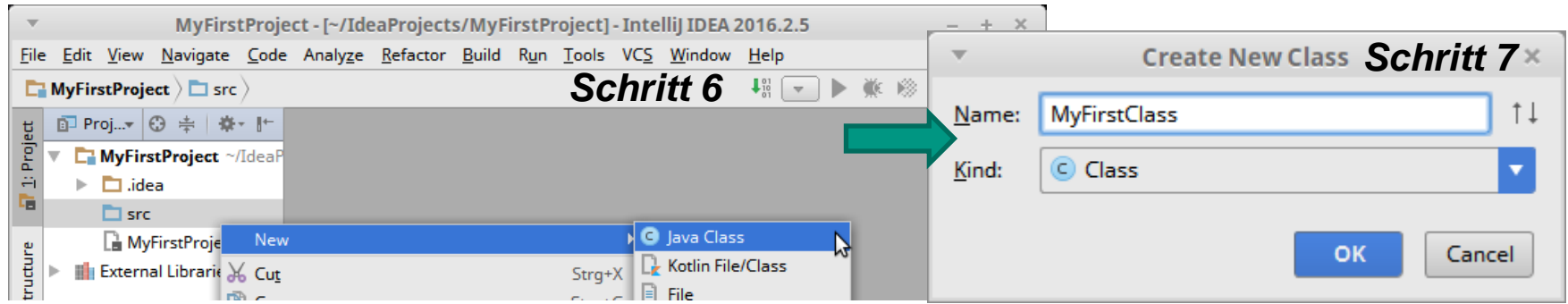
Schritt 3



Neues Projekt (6) – IntelliJ IDEA II

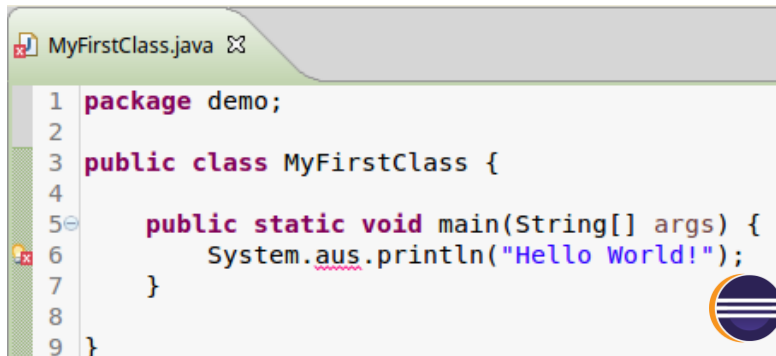


Neues Projekt (7) – IntelliJ IDEA III



Projekt compilieren

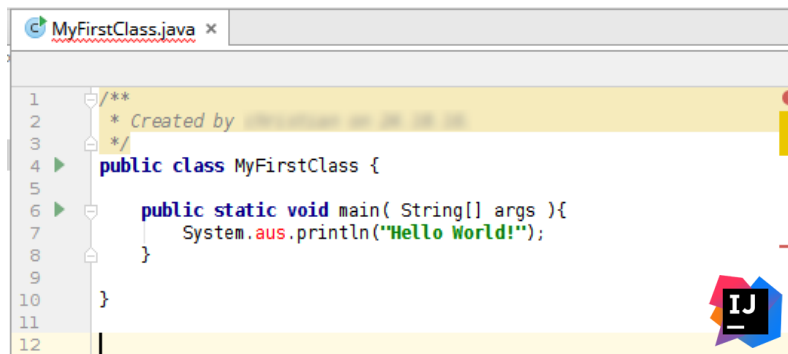
- IDEs übersetzen Java-Dateien beim Speichern in der Regel automatisch und zeigen ggf. Fehler an



```

1 package demo;
2
3 public class MyFirstClass {
4
5     public static void main(String[] args) {
6         System.out.println('Hello World!');
7     }
8
9 }

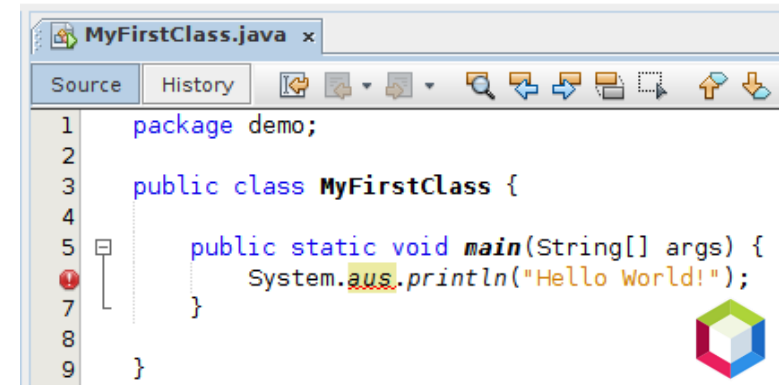
```



```

1 /**
2  * Created by ...
3  */
4 public class MyFirstClass {
5
6     public static void main( String[] args ){
7         System.out.println('Hello World!');
8     }
9
10 }
11
12

```



```

1 package demo;
2
3 public class MyFirstClass {
4
5     public static void main(String[] args) {
6         System.out.println('Hello World!');
7     }
8
9 }

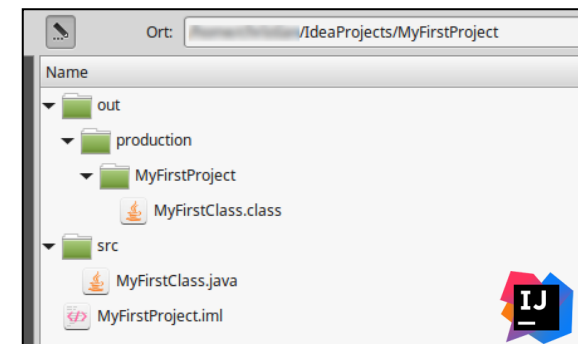
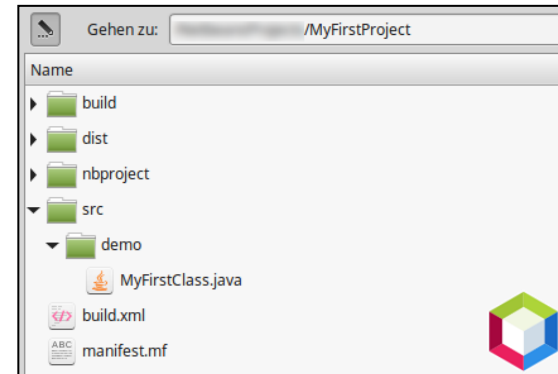
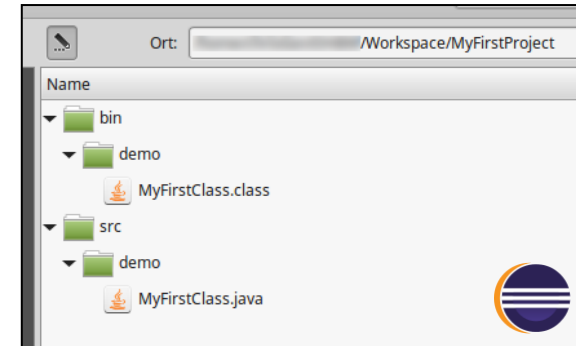
```

Dateien und Dateibaum im Projektverzeichnis

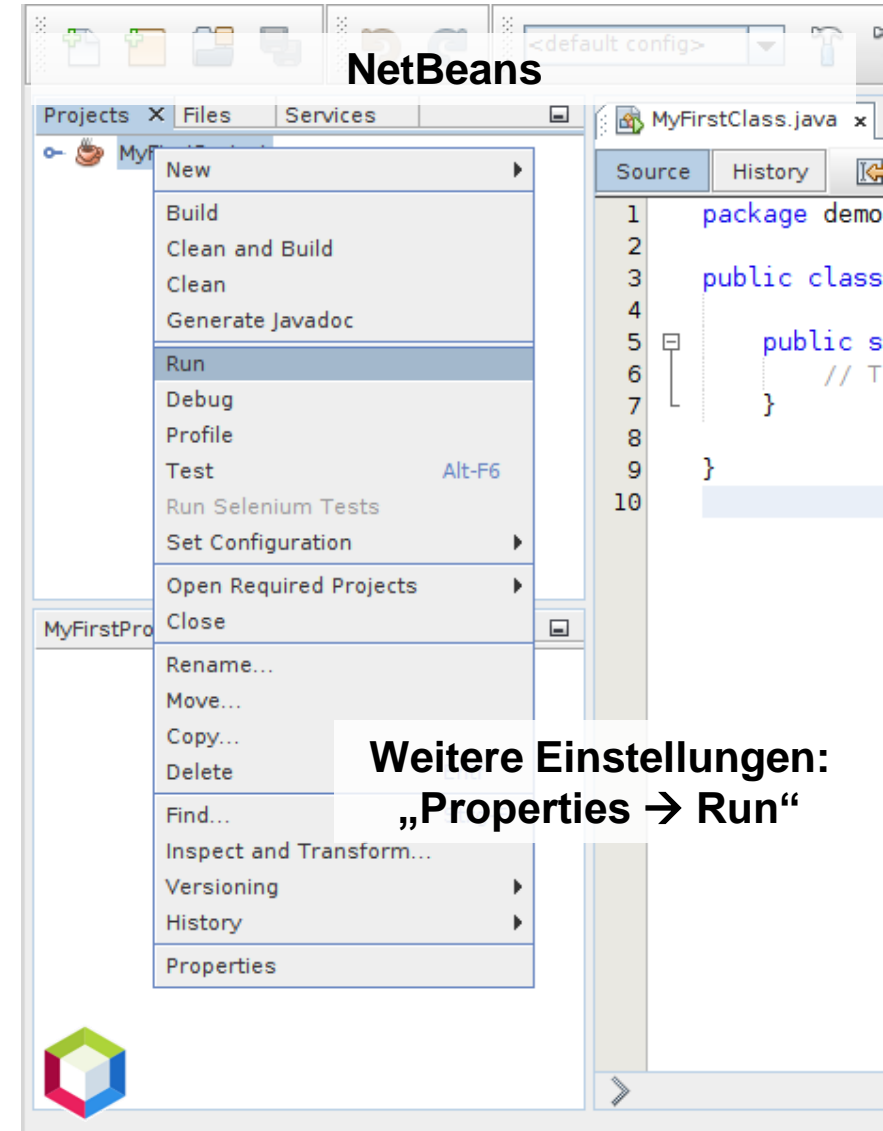
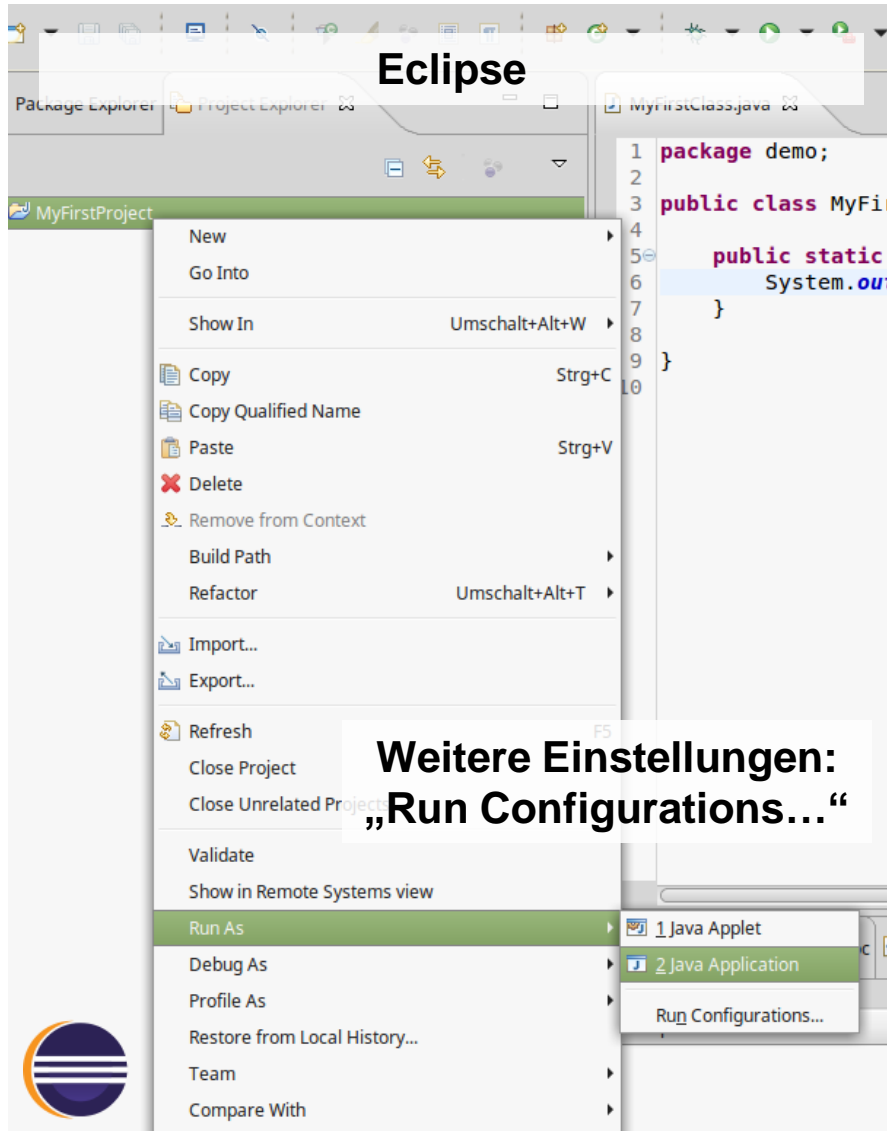
- **src** („Source“)
 - Quellcode / `.java`-Dateien

- **bin** (Eclipse)
 - build** (NetBeans)
 - out** (IntelliJ IDEA)
 - Bytecode / `.class`-Dateien

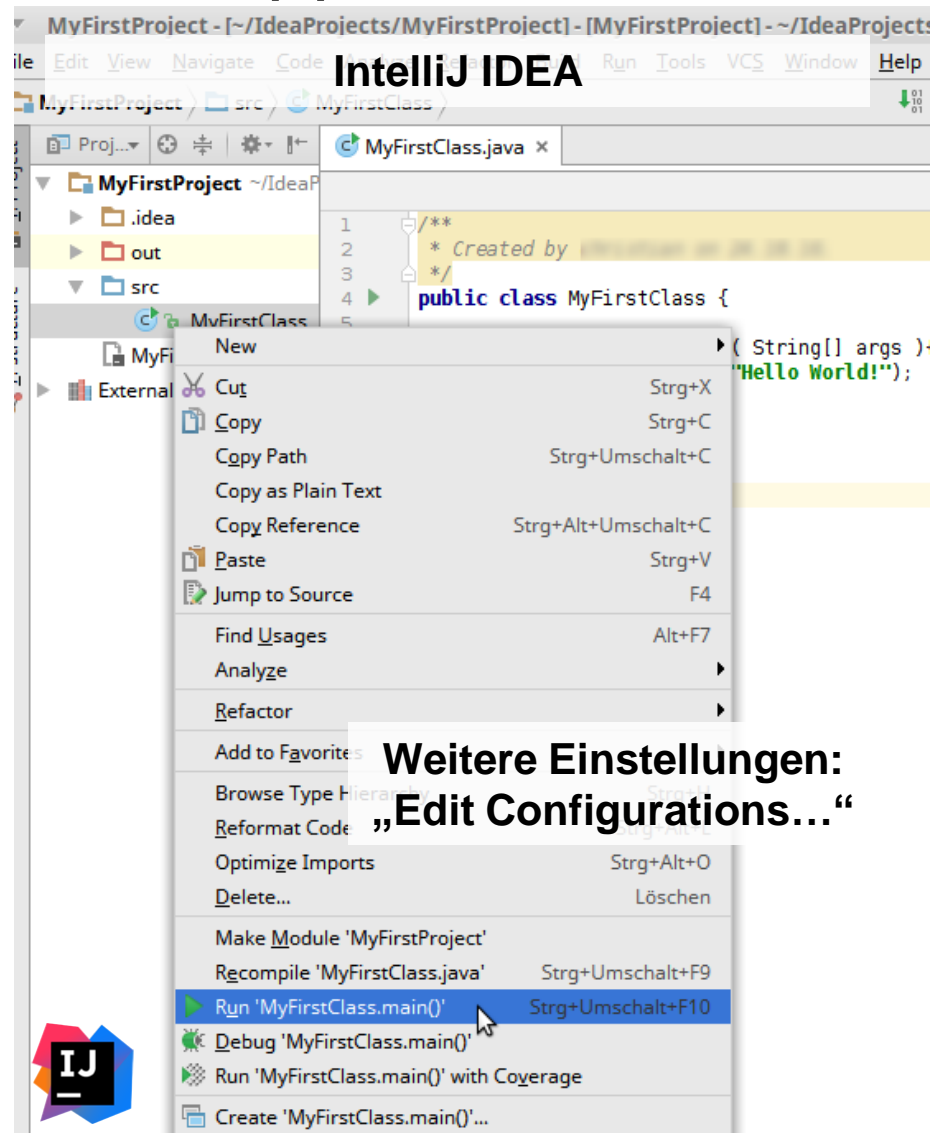
- NetBeans-Spezialität:
 - dist** („Distribution“)
 - `.jar`-Archiv / Dokumentation



Programm ausführen (1)



Programm ausführen (2)



**Weitere Einstellungen:
„Edit Configurations...“**



Effektives Arbeiten: Tastaturkürzel / Hotkeys

- Viele Aktionen können durch geschickten Einsatz von Tastatur oder Maus vereinfacht werden
→ Beschleunigung des Arbeitsablaufs
- Viele Tastatur- und Mausaktionen sind programmübergreifend gleich
- Viele Werkzeuge halten Standards zu Tastaturbelegungen bereit

Systemweite Tastenkombinationen

- **Strg** + **C** Selektierten Text kopieren (in Zwischenablage)
- **Strg** + **X** Selektierten Text ausschneiden (in Zwischenablage)
- **Strg** + **V** Inhalt der Zwischenablage einfügen
- **Strg** + **P** Drucken
- **Strg** + **S** Speichern
- **Strg** + **Z** Aktion rückgängig machen („Undo“)
- **Strg** + **Y** Aktion erneut ausführen („Redo“)
- **Strg** + **A** Gesamten Text selektieren

Man kann damit sogar reich werden!



Quelle: RTL Television / „Wer wird Millionär? Für den Verein zur Million“ (11.12.2017)

Positionierung des Cursors mit der Tastatur

- **Frage:** Wozu? Dafür gibt es ja die Maus!
- **Antwort:** Meist viel schneller, da Texteingabe sowieso mit Tastatur
- Den Cursor ...

- ... um einzelne Zeichen und Zeilen versetzen:



- ... um einzelne Worte versetzen:



- ... an Anfang/Ende der Zeile versetzen:



- ... an Anfang/Ende der Datei versetzen:



- ... an zuletzt editierte Position springen:



Selektieren von Text mit der Tastatur

- **Frage:** Wozu? Dafür gibt es ja die Maus!
- **Antwort:** Innerhalb eines Absatzes meist schneller
- Selektion (von der aktuellen Cursorposition aus) ...

- ... einzelner Zeichen:

Shift + **→** / **←**

- ... einzelner Zeilen:

Shift + **↑** / **↓**

- ... einzelner Worte:

Strg + **Shift** + **→** / **←**

- ... bis Anfang/Ende der Zeile:

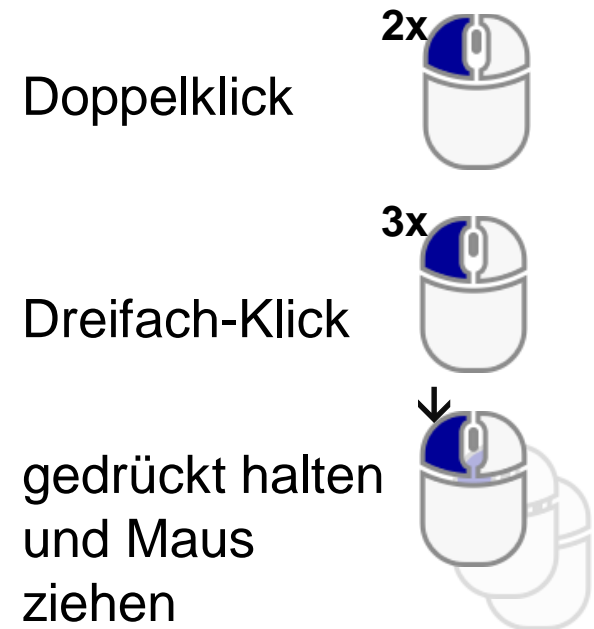
Shift + **Pos1** / **Ende**

- ... bis Anfang/Ende der Datei:

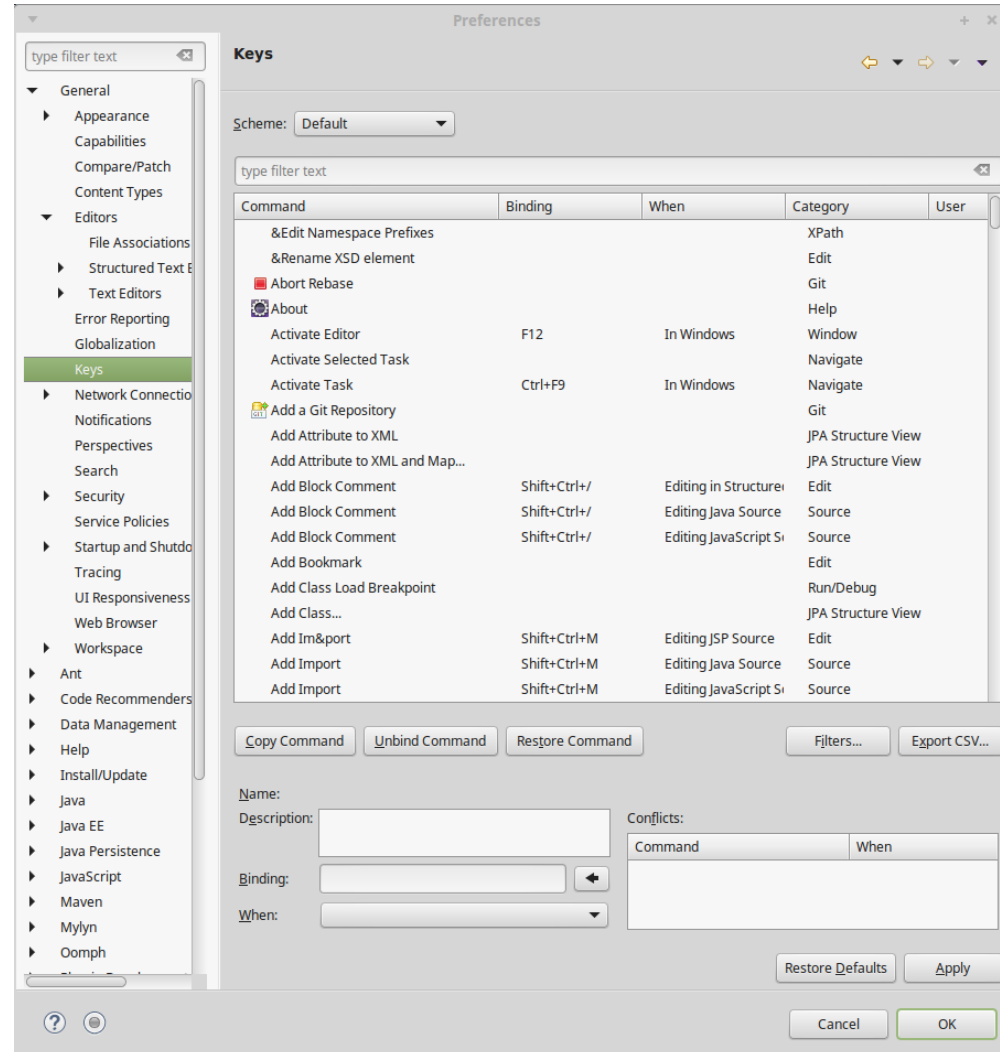
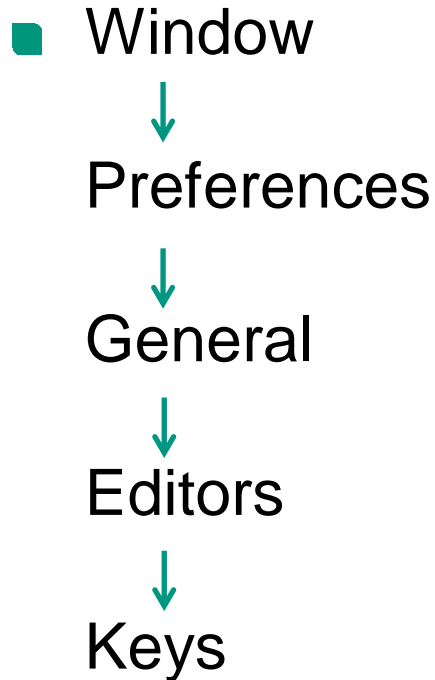
Strg + **Shift** + **Pos1** / **Ende**

Positionierung des Cursors mit der Maus

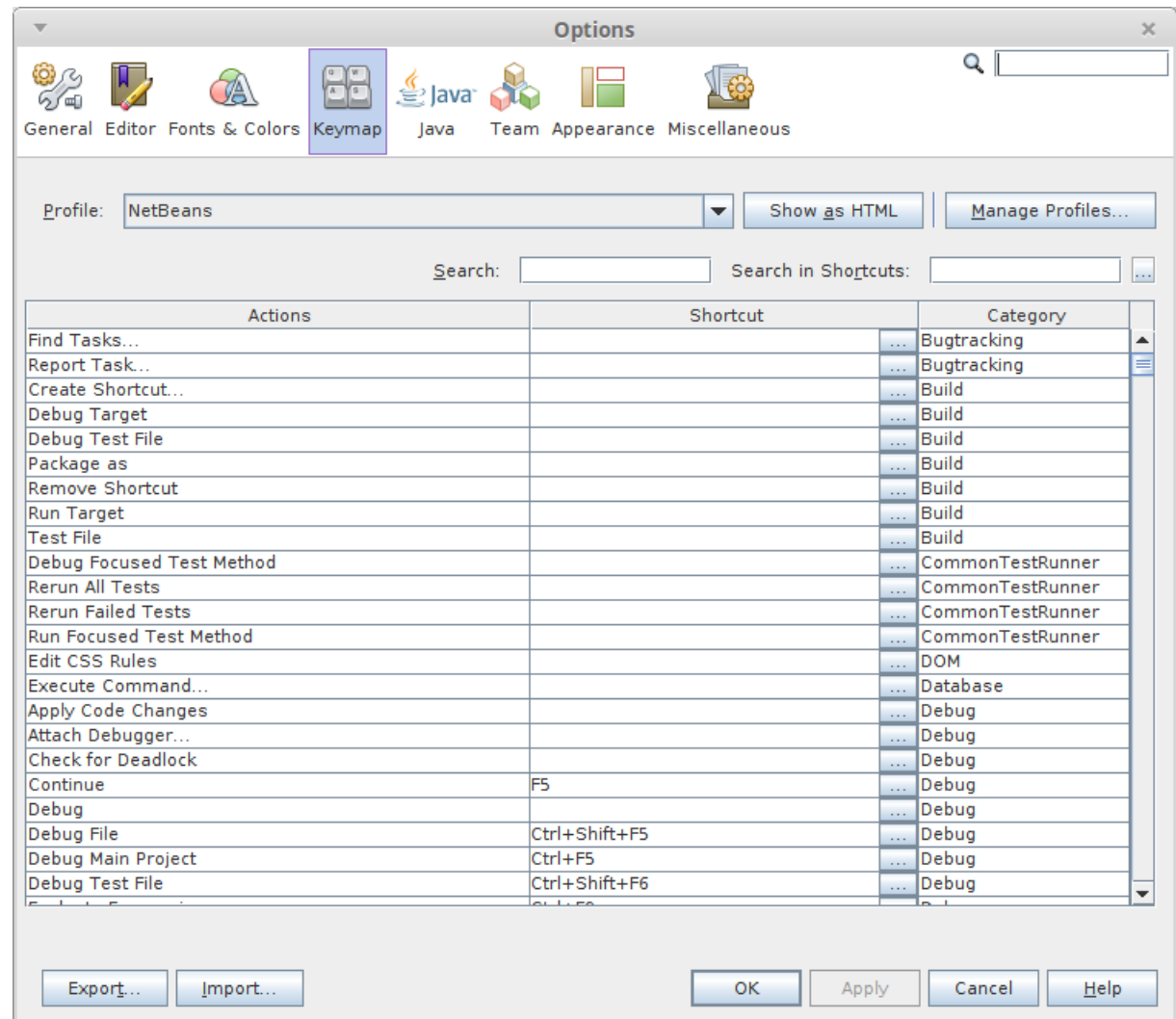
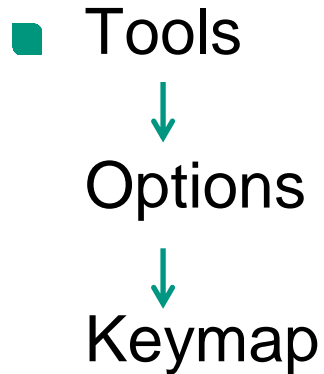
- Vorteil der Maus: „große Strecken“ zurücklegen
- Selektion (ohne Rücksicht auf aktuelle Cursorposition) ...
 - ... einzelner Worte
(zwischen 2 Trennzeichen, z.B. Leerzeichen, Punkt oder Klammer)
 - ... einzelner Zeilen
 - ... Selektion erweitern



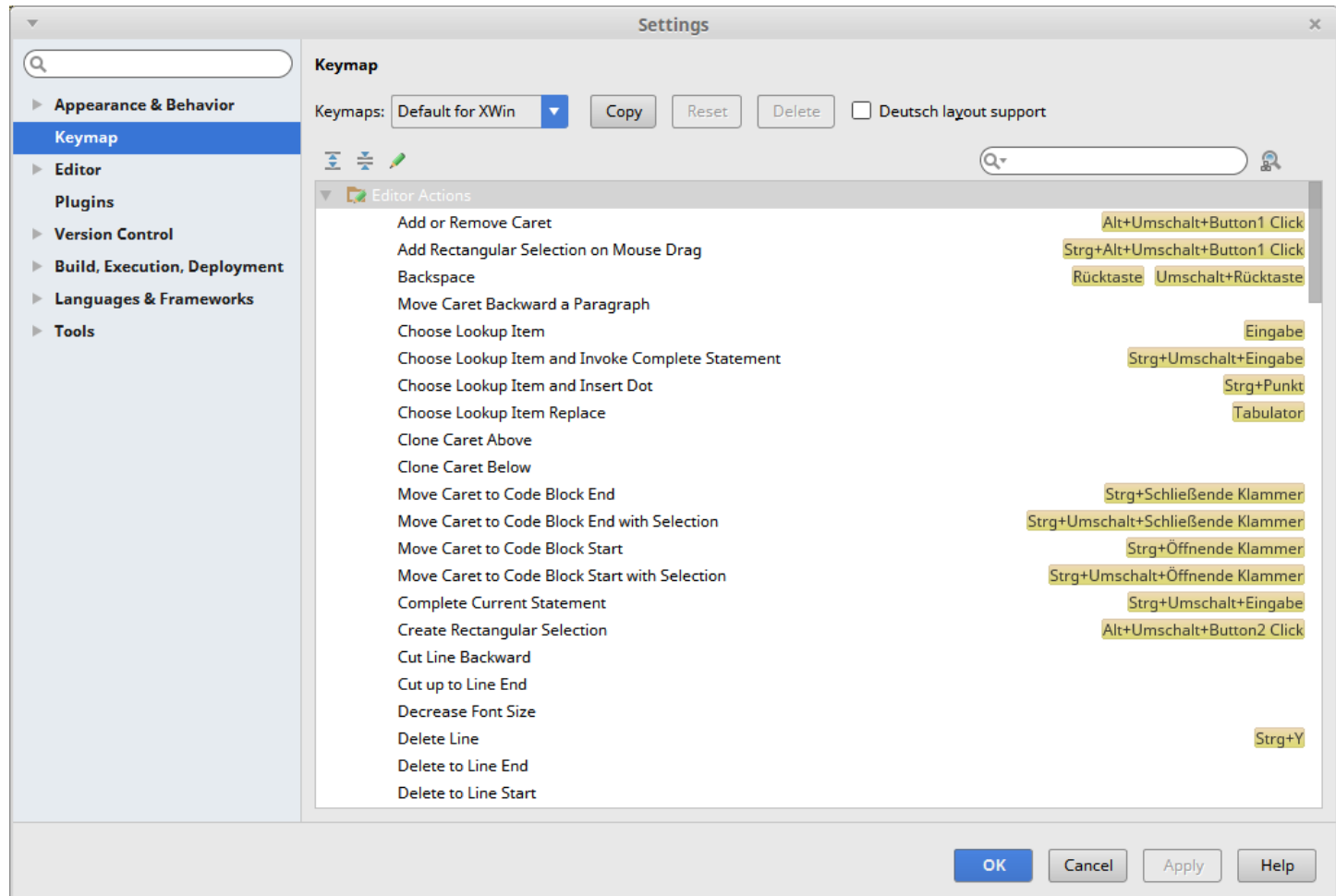
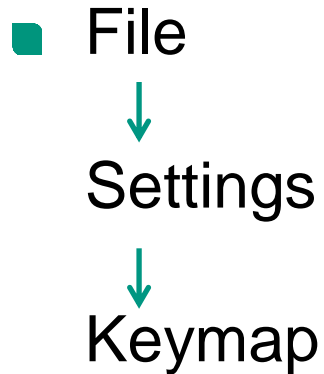
Tastenkombinationen in Eclipse



Tastenkombinationen in NetBeans



Tastenkombinationen in IntelliJ IDEA



Nützliche Beispiele

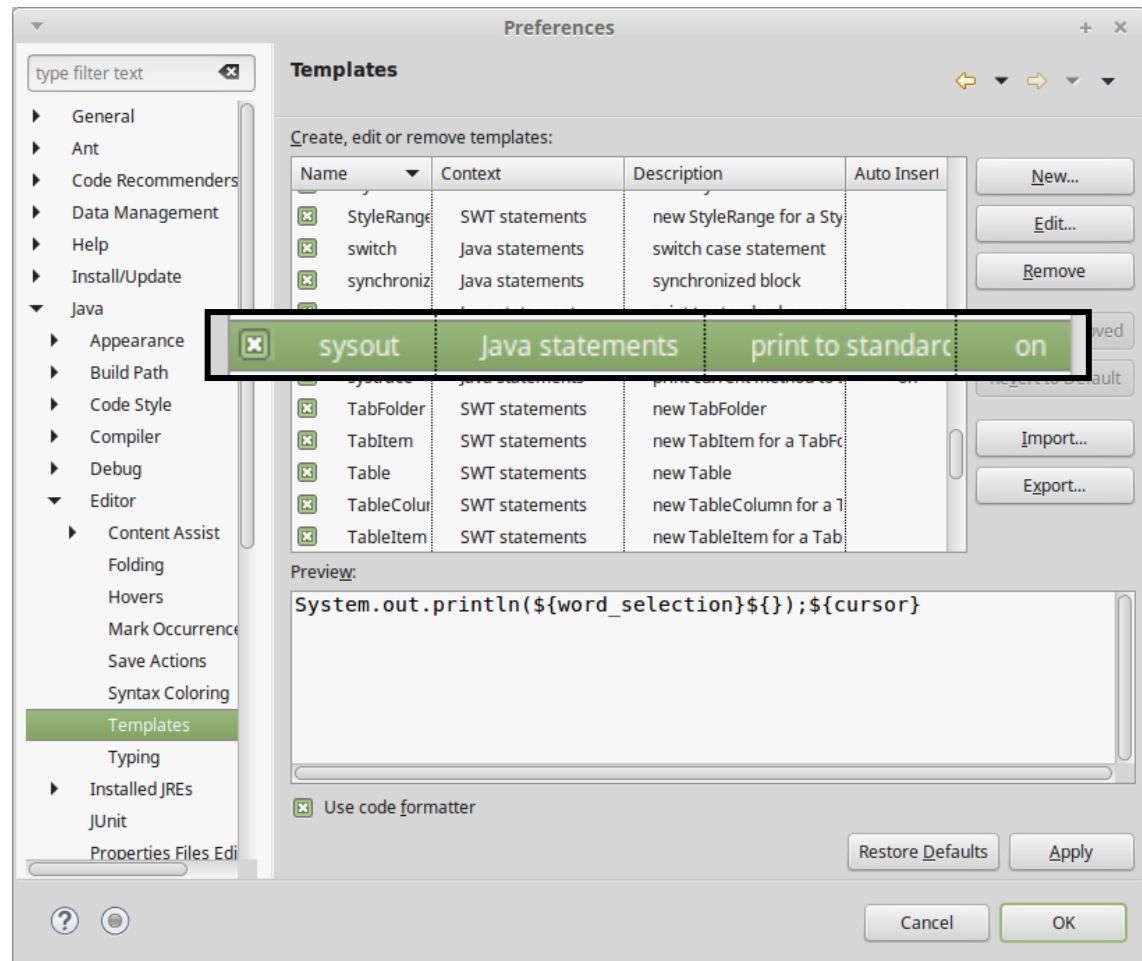
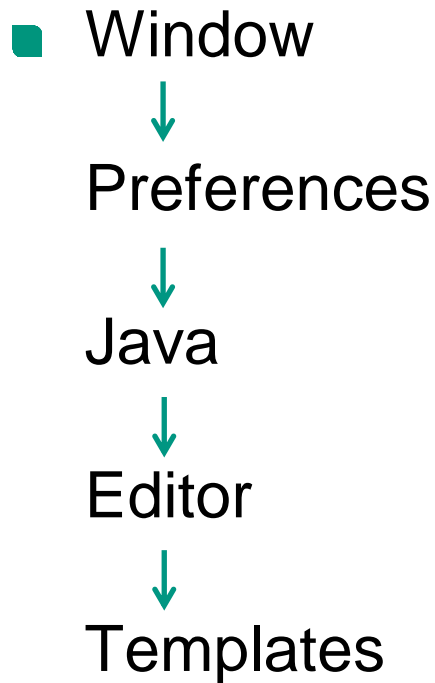
Aktion	Eclipse	NetBeans	IntelliJ IDEA
Programm ausführen	STRG + F11	STRG + F11	STRG + F10
Aktuelle Zeile löschen	STRG + D	STRG + E	STRG + Y
Aktuelle Zeile kopieren	STRG+ALT + ↑ / ↓	STRG + SHIFT + ↑ / ↓	STRG + D
In aktueller Datei suchen	STRG + F	STRG + F	STRG + F
Im Projekt suchen	STRG + H	STRG + SHIFT + F	STRG + SHIFT + F
Parameter-Info	STRG+SPACE (am Anfang der Argumente)	STRG+SPACE (am Anfang der Argumente)	STRG + P

Code-Templates (1)

- Code-Templates sind vorgefertigte Schablonen für längere (und häufig genutzte) Quellcode-Fragmente, die aus Abkürzungen erzeugt werden können
- Gerade zu Beginn häufiges Beispiel: Ausgabe auf Konsole
 - Statt `System.out.println("");` ; nur `sysout` bzw. `sout` tippen
- Abkürzungen und Anwendung können sich je nach IDE unterscheiden

Code-Templates (2) - Eclipse

■ Anwendung: **Strg** + **Space**



Code-Templates (3) - NetBeans

- Anwendung: Je nach Konfiguration, Standard: **Tab**

- Tools



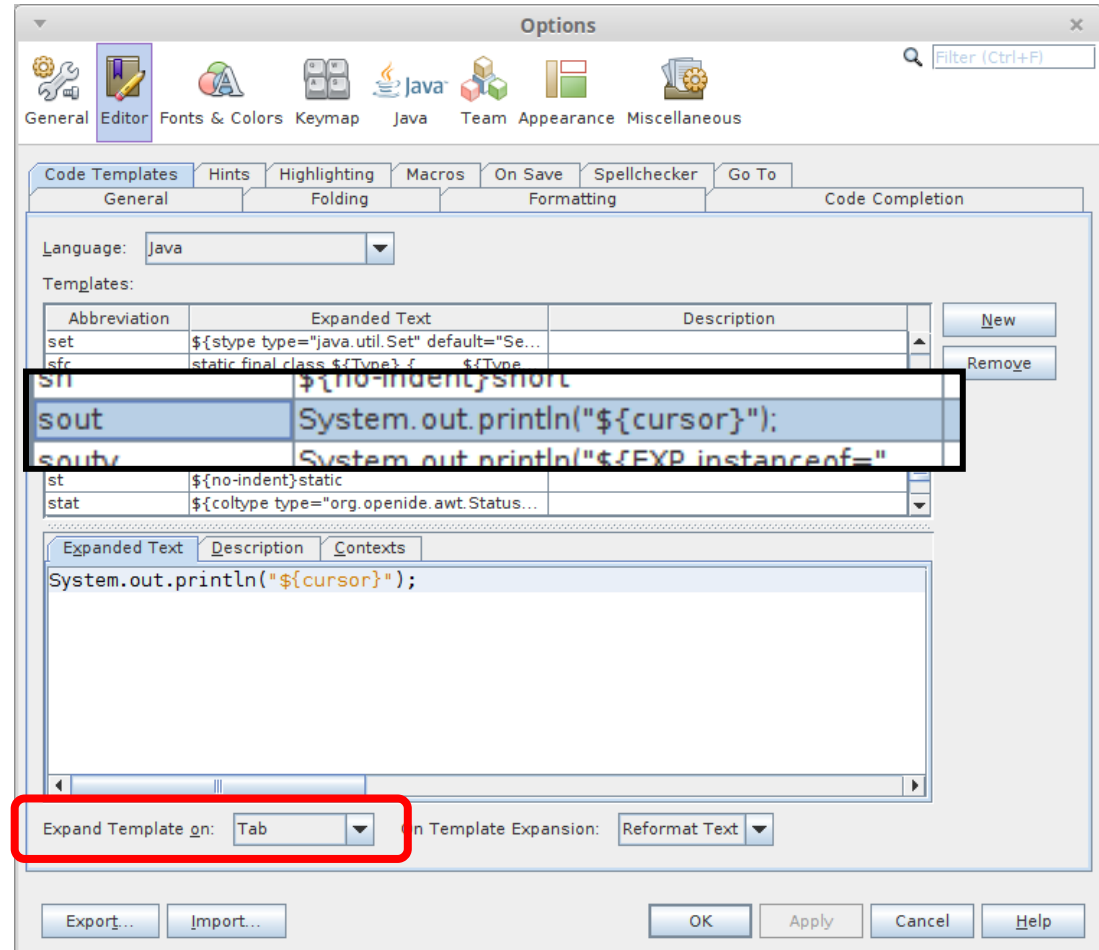
Options



Editor

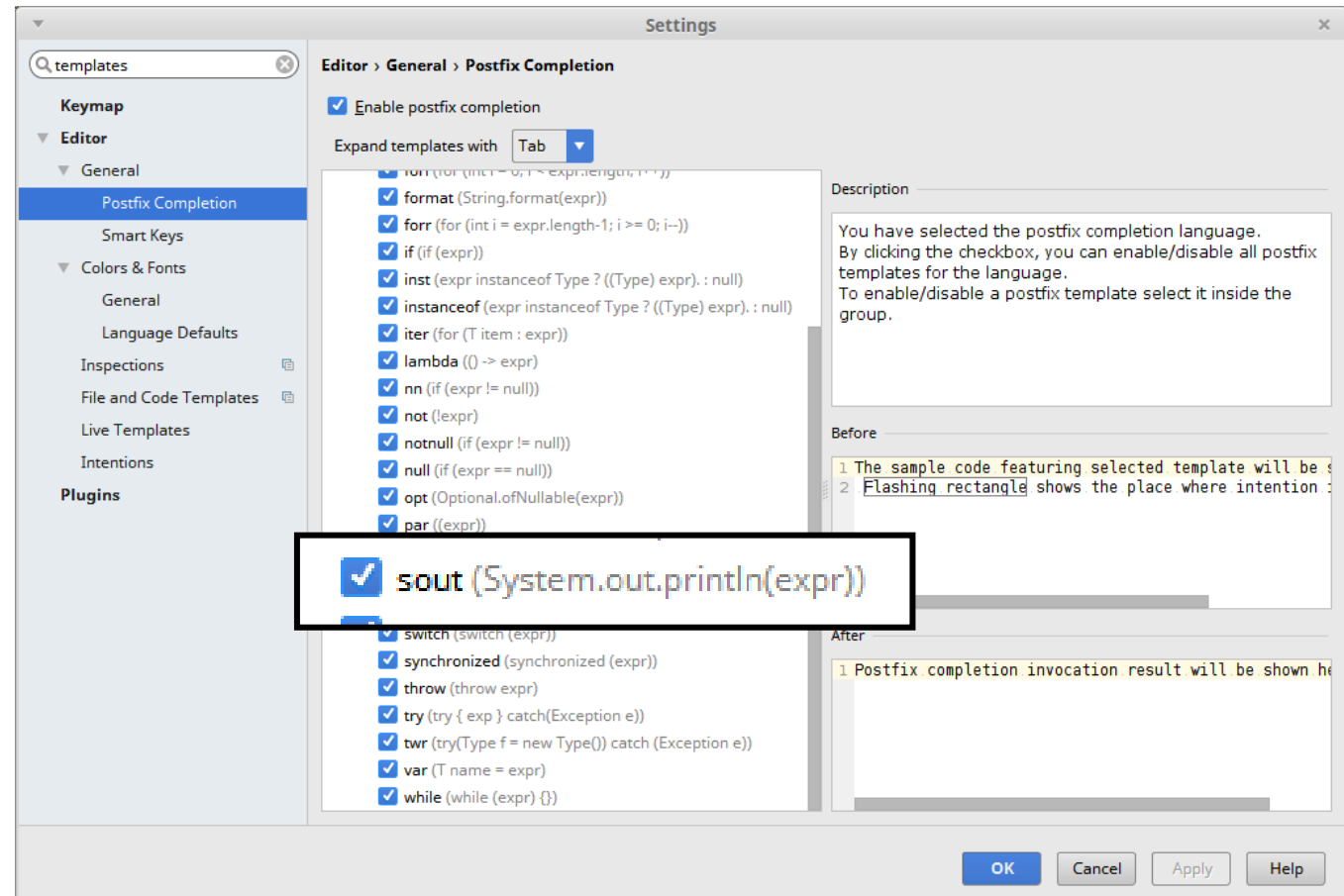
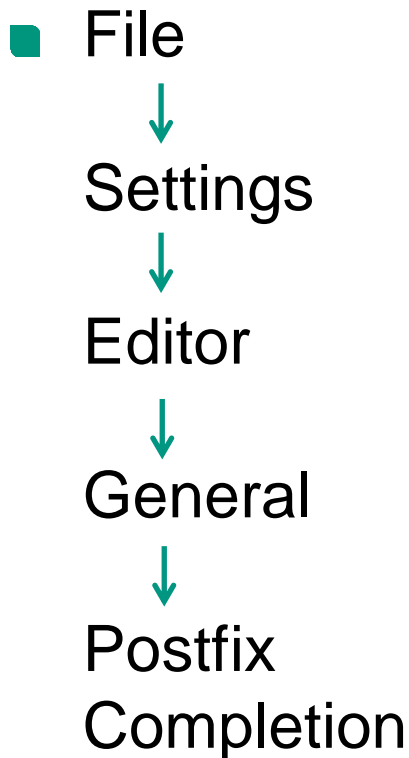


Code Templates



Code-Templates (4) IntelliJ IDEA

■ Anwendung: meist einfach Tippen



Weitere Hilfen

- **Wichtig:**

Werkzeug zum täglichen Arbeiten sollte man gut kennen!

- Lästige „Standard-Arbeiten“ können einem moderne Entwicklungsumgebungen oft abnehmen

- Üben, Üben, Üben ...

➔ **Resultat:**
große Zeitersparnis und effektiveres Programmieren