

Java Grundlagen 1

Bioinformatik Bachelor
Propädeutikum WS 2019/2020
17. Dezember 2019
Leopold Endres

debugging

[de-buhg-ing] -verb.

1. being the detective in a crime movie where you are also the murderer.

1. Was ist Debugging?



- Fehlersuche und -behebung im Programmcode
- Zwei Fehlerarten:
 - Compiler-Fehler: Fehler in der Syntax, Deklarationen, Methodenaufruf
→ in der IDE rot unterringelt
 - Logischer Fehler: Programm macht nicht das, was es soll, wirft aber keinen Error
→ nicht in der IDE markiert, Debugger oft hilfreich

2. Die Wunderwaffe: IntelliJ



- In einer IDE wie IntelliJ sind viele Debugging Methoden integriert
- Immer wenn etwas rot unterringelt ist, gibt es einen Compiler-Fehler
→ Programm kann nicht kompiliert und ausgeführt werden
- **Tipp:**
 1. Klick mit dem Cursor auf einen rot markierten Befehl
 2. Drück ALT + Enter #auf dem Mac und ALT + Enter auf dem PC
 3. Lösungsvorschläge werden angezeigt
 4. Wähle mit der Enter-Taste den obersten Lösungsvorschlag (meist die korrekte Lösung)

3. Weitere IntelliJ-Tipps

- Auto-complete:
sout -> System.out.println("");
main -> komplette main-Methode
for -> for-Schleife
- IntelliJ kann euch auch ganze Methoden generieren:

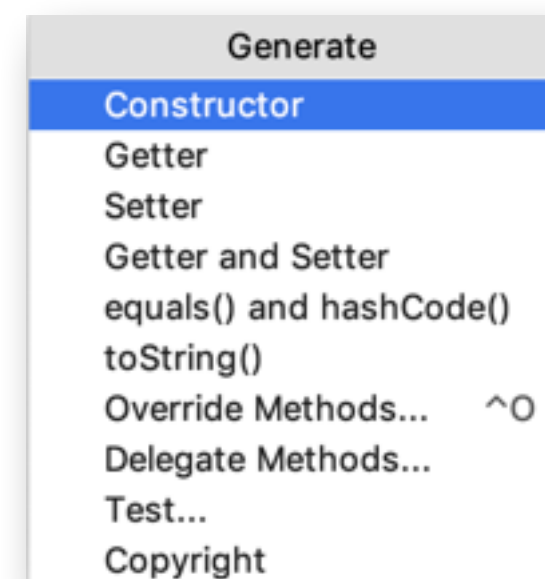
Beispiel Student-Klasse

```
public class Student {
```

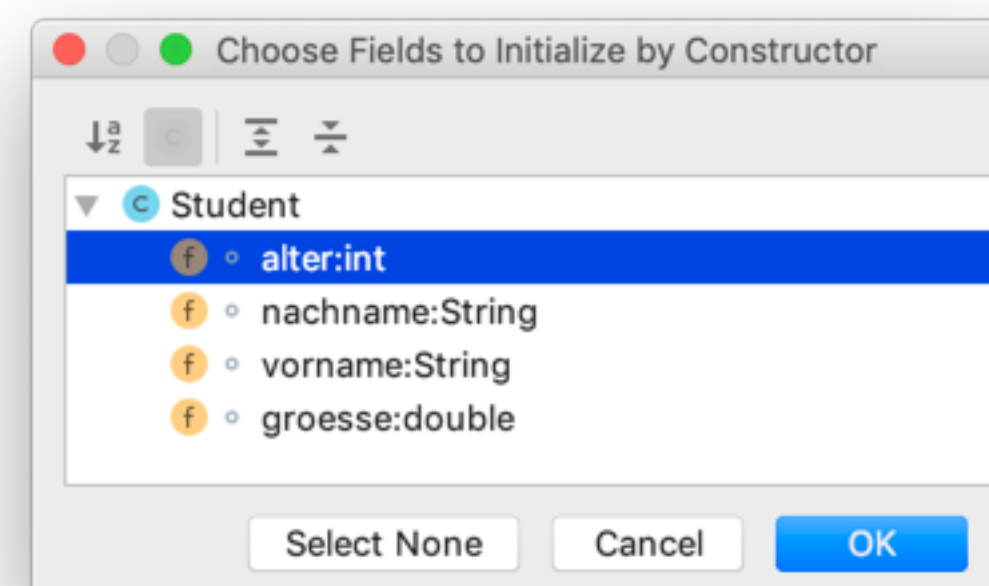
Unter Code -> Generate
auswählen oder Strg + N
als shortcut

```
    int alter;  
    String nachname;  
    String vorname;  
    double groesse;
```

```
}
```



Alle auswählen



```
public class Student {
```

```
    int alter;  
    String nachname;  
    String vorname;  
    double groesse;
```

```
    public Student(int alter, String nachname,  
String vorname, double groesse) {  
        this.alter = alter;  
        this.nachname = nachname;  
        this.vorname = vorname;  
        this.groesse = groesse;
```

```
    }  
}
```

4. Debugging-Methoden



1. Ausgabe von Werten einer Variablen an bestimmten Stellen im Programm
 - `System.out.println("xx");` - Wichtig:
 - Hierbei keine Fehler machen (z.B.: Variablen verändern) - Sinnvolle Ausgaben
2. Debugger verwenden
 - Breakpoint setzen
 - Programm Schritt für Schritt durchgehen
 - Anzeige aller Variablen zu einem beliebigen Programmzeitpunkt
3. Mischung aus Ausgabe und Debugger

5. Final Thoughts



- Probiert den Debugger aus! Er wird euch helfen und kann euch sehr viel Zeit und Frust ersparen.
- Tipps zur allgemeinen Fehlerbehebung:
 - Testet einzelne Abschnitte eures Programm.
 - Wenn es nicht funktioniert, muss es einen Fehler geben.
 - Fragt eure Tutoren und Kommilitonen nach Hilfe.
 - google, google, google
 - and most importantly: Keep your head up!

6. Debugger-Live-Demonstration

