

Lehrstuhl Bioinformatik • Konstantin Pelz

Debugging

Tutorium Bioinformatik
(WS 18/19)

Konstantin: Konstantin.pelz@campus.lmu.de

Homepage: <https://bioinformatik-muenchen.com/studium/propaedeutikum-programmierung-in-der-bioinformatik/>





Compiler Fehler

- Fehler in der Syntax
- Sichtbarkeiten, Deklarationen, Methodenaufrufe
- Import fehlt (Quick-Fix: STRG+Shift+O)

Semantische/ Logische Fehler

- "Das Programm macht nicht das, was es soll"
- Manchmal schwer zu finden. -> Debugger!

```
File f = new File("hallo");
```

File cannot be resolved to a type

7 quick fixes available:

- ← Import 'File' (java.io)
- ⊙ Create class 'File'
- ↔ Change to 'FileLock' (java.nio.channels)
- ↔ Change to 'Files' (java.nio.file)
- ↔ Change to 'FileTime' (java.nio.file.attribute)
- ↔ Change to 'FileDialog' (javax.swing.filechooser)
- ↔ Fix project setup...

Press 'F2' for focus



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

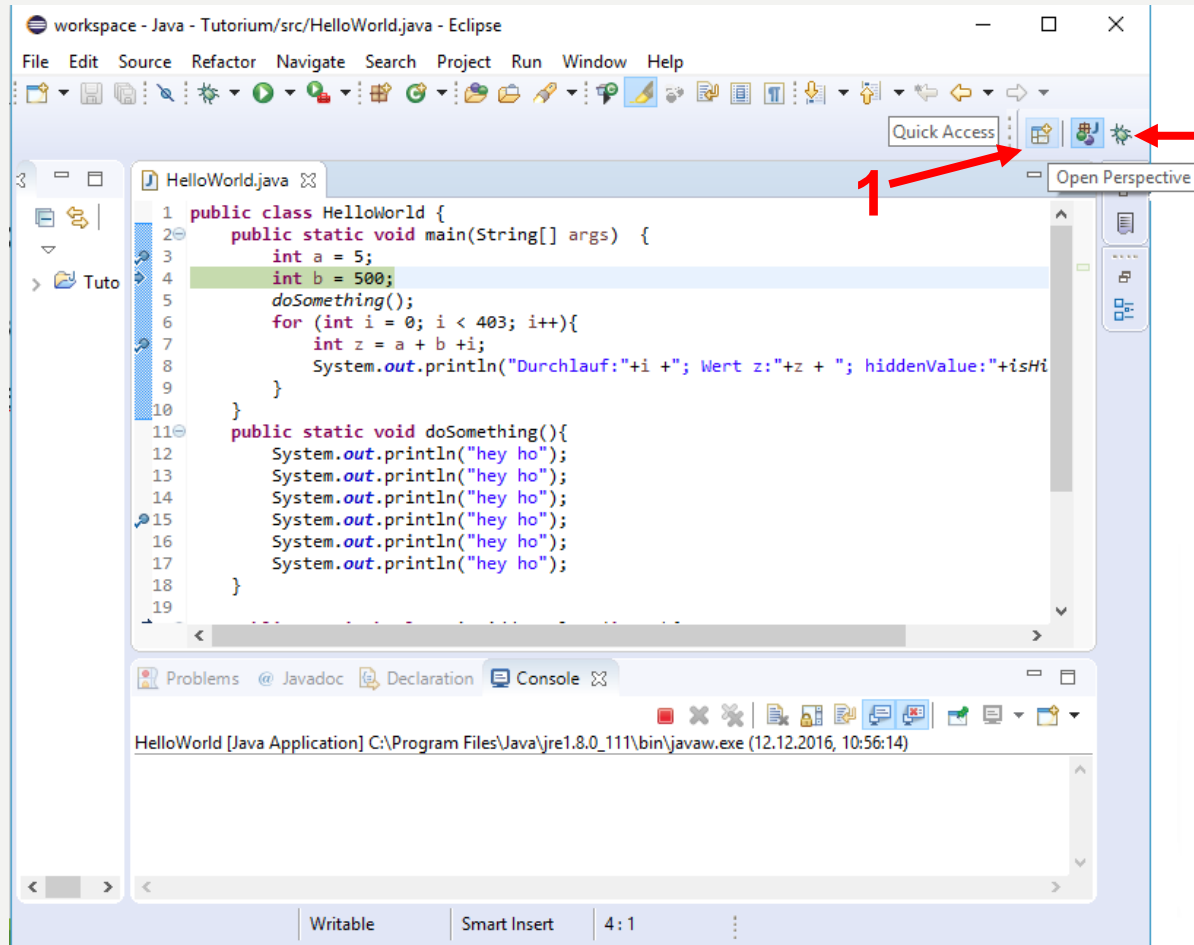
- Anzeige aller Variablen zu einem beliebigen Programmzeitpunkt -> Breakpoints
- Grafische Oberfläche
- Wichtig: sinnvolle Breakpoints

3. Mischung aus Ausgabe und Debugger



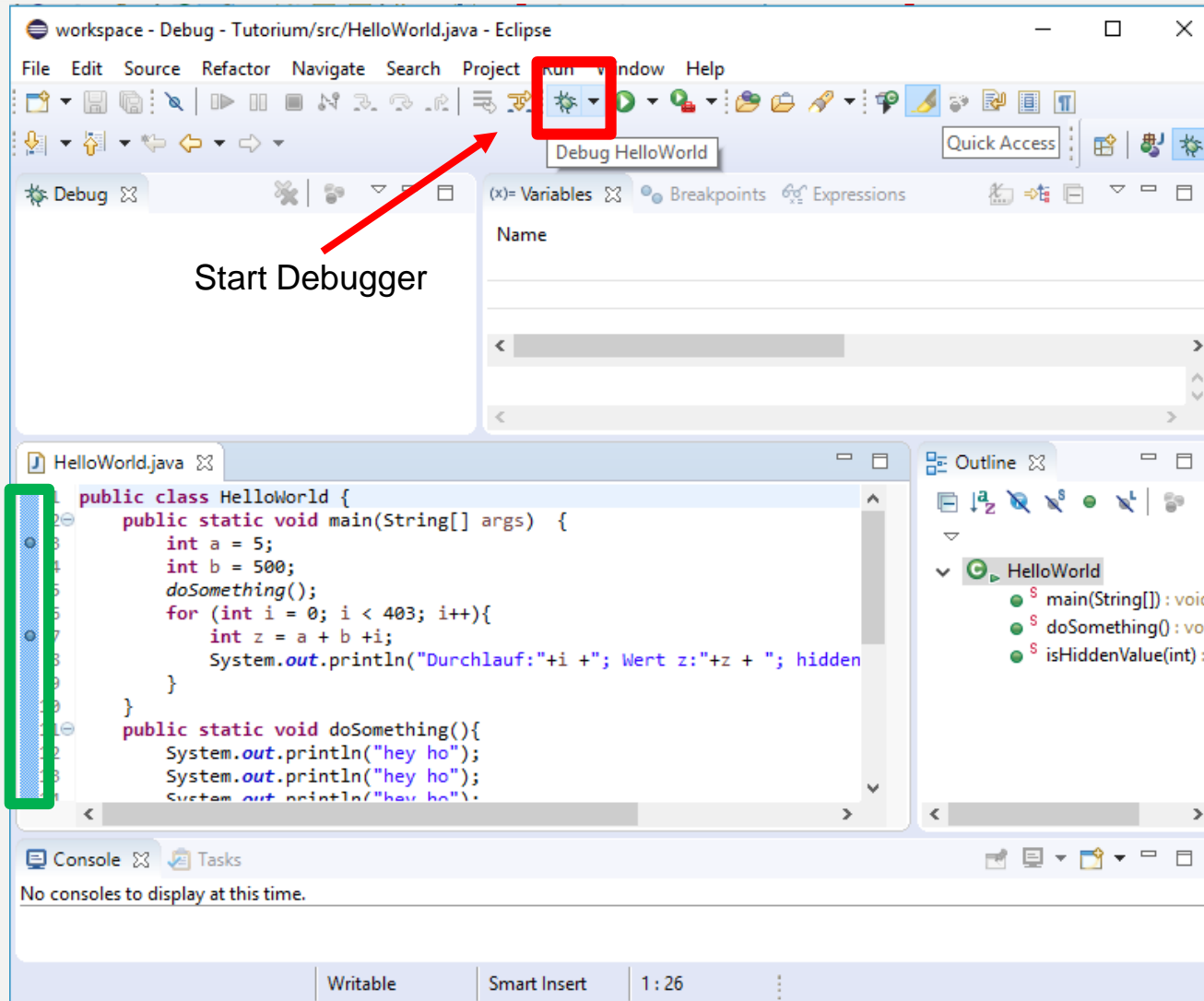
- 1. Zur Debug-Perspektive wechseln**
- 2. Breakpoints setzen**
- 3. Debugging starten**
- 4. Weitere Analysen durchführen**

Debug Perspektive mit 1 oder 2 öffnen



Gesetzte
Breakpoints

Start Debugger

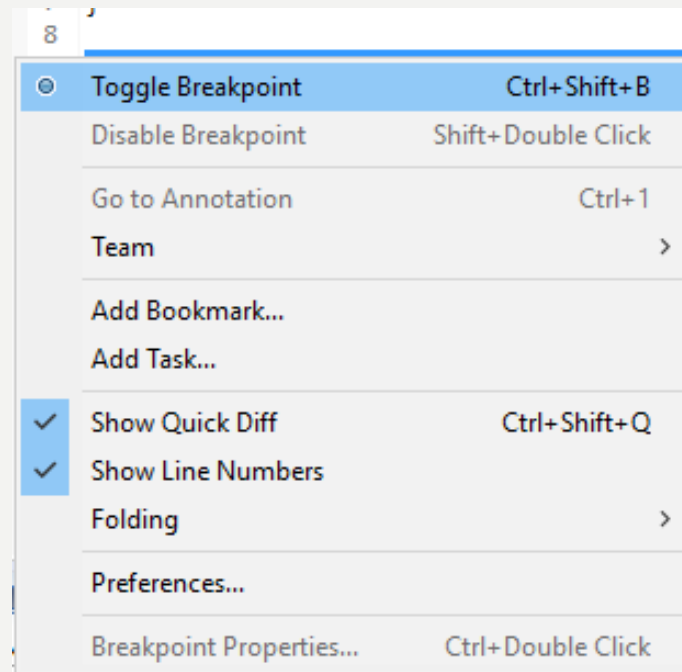


The screenshot shows the Eclipse IDE interface for a Java project named "workspace - Debug - Tutorium/src/HelloWorld.java". The main editor displays the source code of the "HelloWorld" class. Two breakpoints are set on the first line of the "main" method and the first line of the "doSomething" method. A red box highlights the "Run" button in the toolbar, with a red arrow pointing to it and the text "Start Debugger". A green box highlights the breakpoint markers in the left margin of the code editor, with a green arrow pointing to it and the text "Gesetzte Breakpoints". The "Debug" console is open, showing the "Variables" tab. The "Outline" view on the right shows the class structure with the "main" method selected. The "Console" view at the bottom shows "No consoles to display at this time."

```
public class HelloWorld {  
    public static void main(String[] args) {  
        int a = 5;  
        int b = 500;  
        doSomething();  
        for (int i = 0; i < 403; i++){  
            int z = a + b + i;  
            System.out.println("Durchlauf:"+i + "; Wert z:"+z + "; hidden  
        }  
    }  
    public static void doSomething(){  
        System.out.println("hey ho");  
        System.out.println("hey ho");  
        System.out.println("hey ho");  
    }  
}
```

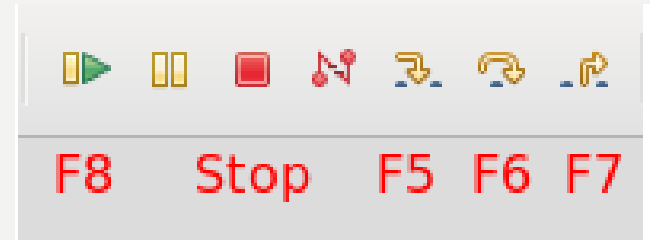


- Programm wird hier angehalten
- Doppelklick auf die Zeile oder Rechtsklick + Auswahl
- Breakpoint Properties (fortgeschritten)
 - Bedingungen zum Anhalten checken
 - z.B. Fehler tritt erst nach 1000 Durchläufen auf.





- F5: Step into
- F6: Step over
- F7: Step reverse
- F8: Resume
- Strg + F2: Terminate
- In Methode springen
- Zur nächsten Zeile im Code (ggf. über Methode springen)
- Aus Methode rauspringen. (Step into rückgängig machen)
- Zum nächsten Breakpoint navigieren
- Beenden des Debuggens



- Strg + 1
- Vorschläge von Eclipse anzeigen lassen, wenn was rot unterringelt ist
- Rot Unterringeltes anklicken -> Verbesserungsvorschläge auswählen
- Und viele mehr



- **Variables:** Alle sichtbaren Variablen am momentanen Breakpoint
- **Breakpoints:** Alle gesetzten Breakpoints + Breakpoint Properties
- **Expressions:** Alle selbst gesetzten Variablen/ Ausdrücke, die unter Beobachtung stehen

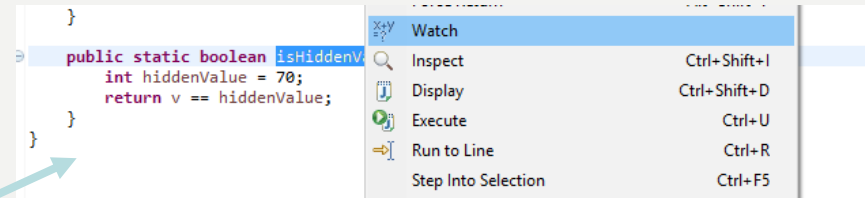
The screenshot shows the Eclipse IDE's 'Variables' view. The window title is '(x)= Variables'. Below the title bar are three tabs: 'Variables', 'Breakpoints', and 'Expressions'. The 'Variables' tab is active, displaying a table with two columns: 'Name' and 'Value'. The table contains four rows of variables: 'args' with value 'String[0] (id=16)', 'a' with value '5', 'b' with value '500', and 'i' with value '37'. The row for 'i' is highlighted in yellow. The 'Name' column has a small circular icon to the left of each variable name. The 'Value' column shows the current value of each variable at the breakpoint.

Name	Value
args	String[0] (id=16)
a	5
b	500
i	37



- Watch

- Anzeigen von ausgewählten Variablen, Methoden, Ausdrücken
- Syntax wichtig, sonst Error
- Kann nicht in die Zukunft schauen



- Markieren > Rechtsklick > Watch
- "Expressions" Tab > "Add new Expression"

Name	Value
<code>int b = 500;</code>	null
<code>"b"</code>	<error(s)_during_the_evaluation>
Add new expression	



- **Probiert es aus, es wird euch helfen!**
- **Testet die unterschiedlichen Arten des Debuggens**
- **Strategie um einen Bug zu finden:**
 - **Finde Punkt an dem noch alles funktioniert, starte hier mit der Fehlersuche**
 - **Mache Pausen**
 - **Frage Leute, die das ähnliche Problem haben könnten (Kommilitonen, Internet)**
 - **Verstehe den Fehler und eigne dir ggf. Zusatzwissen an**
 - **Nicht verzweifeln!**
- **<http://www.vogella.com/tutorials/EclipseDebugging/article.html>**

debugging

[de-buhg-ing] -verb.

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