

Programmierung 2

Kapitel 2a
Debuggen

Testen vs. Debuggen

Ist die Software korrekt?

Testen

Warum ist die Software nicht korrekt?

Wo ist der Fehler?

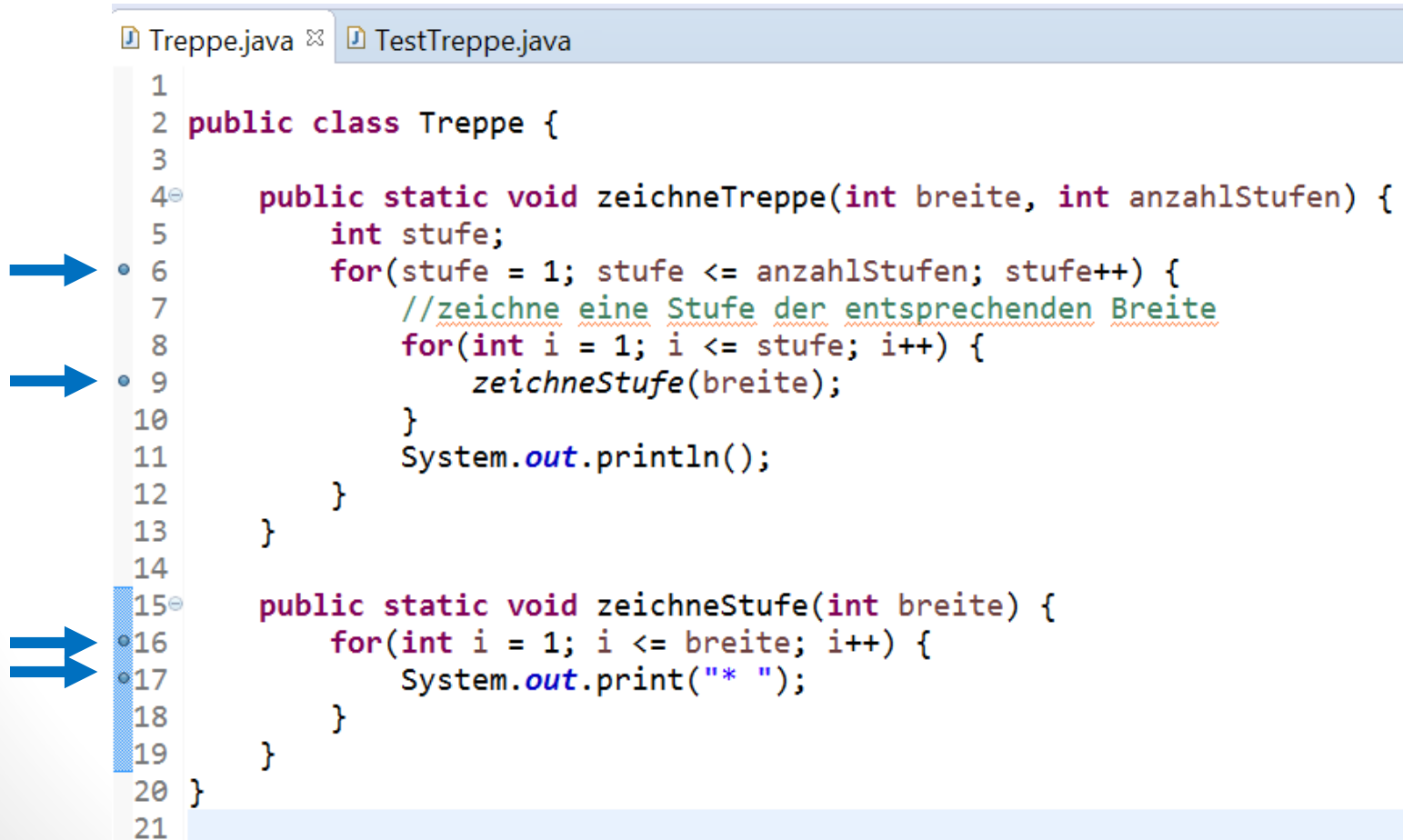
Debuggen

Was kann ein Debugger

- Programm anhalten → **Breakpoint** setzen
- Variablenbelegung anschauen und verändern
- Programm fortsetzen (mehrere Modi)

Breakpoints setzen

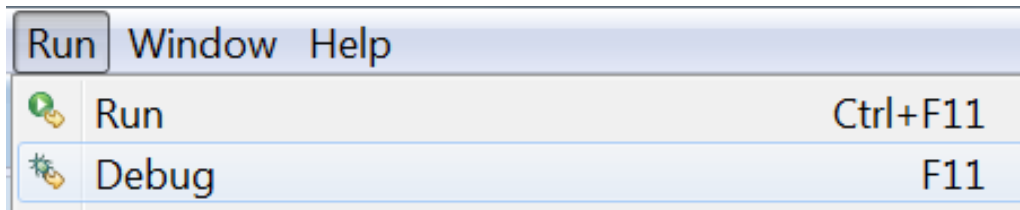
in Eclipse: Doppelklick an den Rand des Programmfensters →
kleiner blauer Punkt ● erscheint



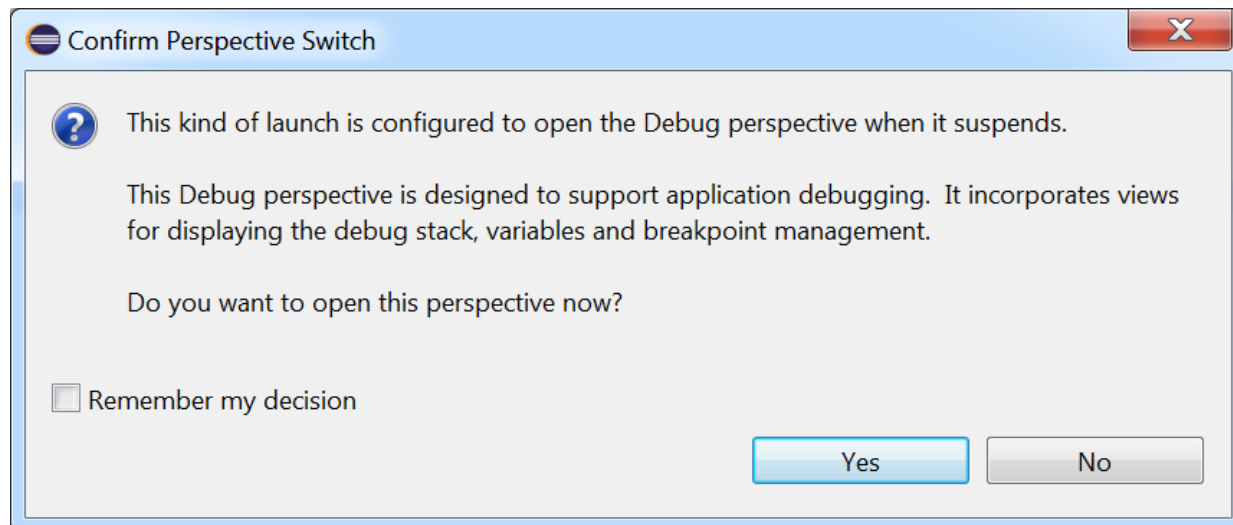
```
Treppe.java x TestTreppe.java
1
2 public class Treppe {
3
4     public static void zeichneTreppe(int breite, int anzahlStufen) {
5         int stufe;
6         for(stufe = 1; stufe <= anzahlStufen; stufe++) {
7             //zeichne eine Stufe der entsprechenden Breite
8             for(int i = 1; i <= stufe; i++) {
9                 zeichneStufe(breite);
10            }
11            System.out.println();
12        }
13    }
14
15    public static void zeichneStufe(int breite) {
16        for(int i = 1; i <= breite; i++) {
17            System.out.print("* ");
18        }
19    }
20 }
21
```

Debugger starten

Run → Debug (F11)



bevor Debug-Ansicht geöffnet



Debug-Modus

Variablen- und Breakpoint-Ansicht

The screenshot shows the Eclipse IDE in Debug mode. The top toolbar includes buttons for 'Quick Access', 'Java', and 'Debug'. The main window is divided into several panes:

- Aufrufhierarchie (Call Hierarchy):** Located in the top-left pane, it shows the call stack. The current thread is 'Thread [main] (Suspended (breakpoint at line 9 in Treppe))'. The current method is 'Treppe.zeichneTreppe(int, int) line: 9'. The caller is 'TestTreppe.main(String[]) line: 4'.
- Variablen- und Breakpoint-Ansicht (Variables and Breakpoints View):** Located in the top-right pane. It shows the 'Variables' tab with a table of variables and their values:

Name	Value
breite	3
anzahlStufen	7
stufe	1
i	1
- Source-Code und Breakpoints (Source Code and Breakpoints View):** Located in the bottom-left pane. It shows the source code of 'Treppe.java'. The current line is 9, which is highlighted in green. A green arrow points to this line, indicating the 'aktuelle Position' (Current Position). The code is:

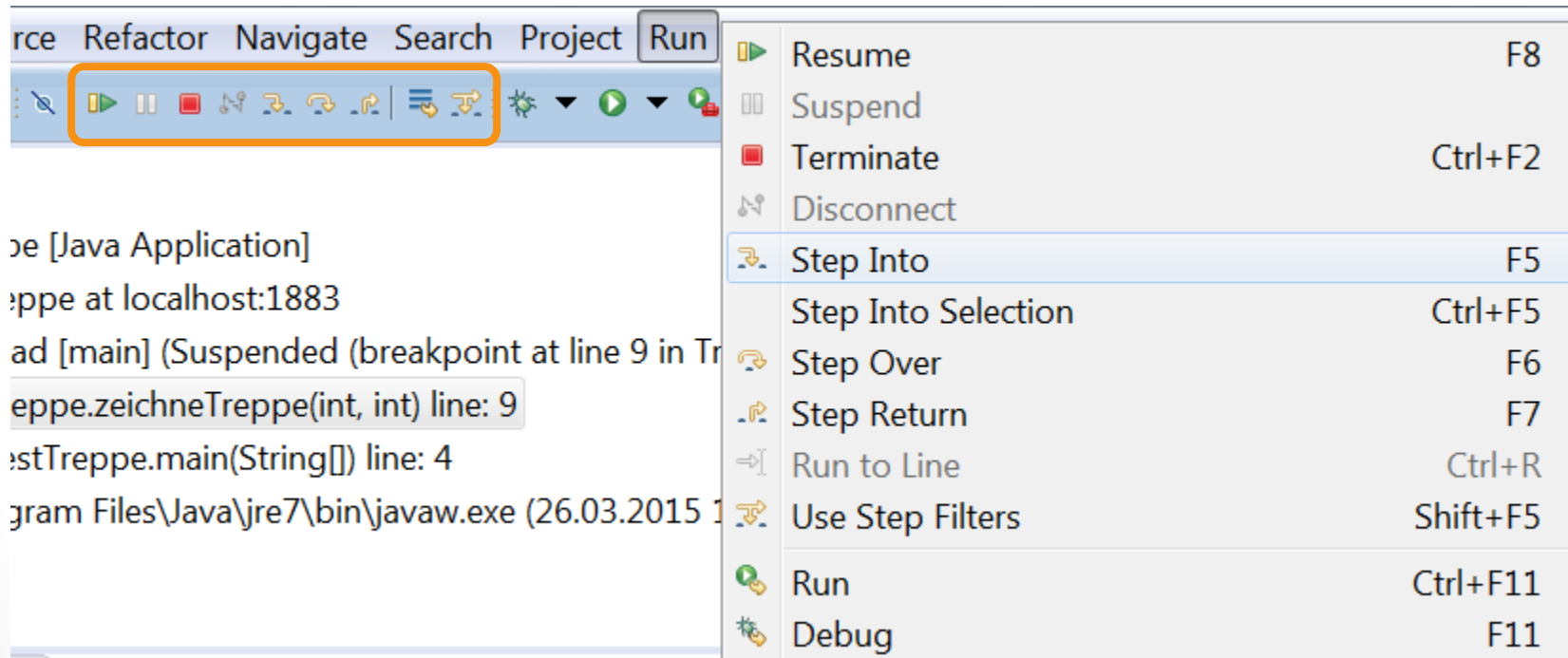
```
1 public class Treppe {  
2  
3  
4 public static void zeichneTreppe(int breite, int anzahlStufen)  
5     int stufe;  
6     for(stufe = 1; stufe <= anzahlStufen; stufe++) {  
7         //zeichne eine Stufe der entsprechenden Breite  
8         for(int i = 1; i <= stufe; i++) {  
9             zeichneStufe(breite);  
10        }  
11        System.out.println();  
12    }  
13 }
```
- Gliederung-Ansicht (Outline View):** Located in the bottom-right pane. It shows the structure of the 'Treppe' class, including the methods 'zeichneTreppe(int, int) : void' and 'zeichneStufe(int) : void'.
- aktuelle Position (Current Position):** Located in the bottom-center pane. It shows the current position in the source code, which is line 9.
- BildschirmAusgabe (Screen Output):** Located in the bottom-most pane. It shows the output of the program, which is a series of asterisks: '* * * * *

Debugging-Aktionen

- Resume: Programm bis zum nächsten Breakpoint ausführen
- Step Into: wenn aktuelle Position = Methodenkopf, dann in die Methode reinspringen (Ausführung in der 1. Zeile anhalten)
- Step Return: angesprungene Methode zu Ende ausführen und wieder verlassen
- Step Over: aktuelle Programmzeile ausführen (wenn Methodenaufruf, dann vollständig abarbeiten)
- Run to Line: Programm bis zu der Zeile ausführen, in der sich der Cursor befindet
- Terminate: Ausführung des Programms beenden

Aktion auswählen

über Toolbar, Menü oder Tastenkombination wählen



The screenshot shows an IDE interface with the 'Run' menu open. The menu items and their shortcuts are as follows:

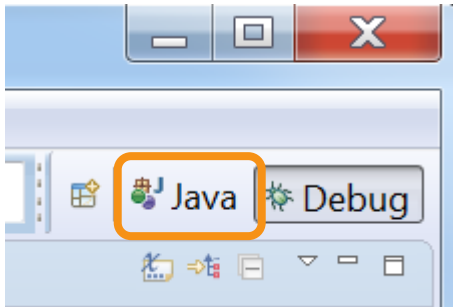
Action	Shortcut
Resume	F8
Suspend	
Terminate	Ctrl+F2
Disconnect	
Step Into	F5
Step Into Selection	Ctrl+F5
Step Over	F6
Step Return	F7
Run to Line	Ctrl+R
Use Step Filters	Shift+F5
Run	Ctrl+F11
Debug	F11

The toolbar at the top shows the 'Run' button (a green play icon) highlighted with an orange box. Below the toolbar, the IDE's output window shows the following text:

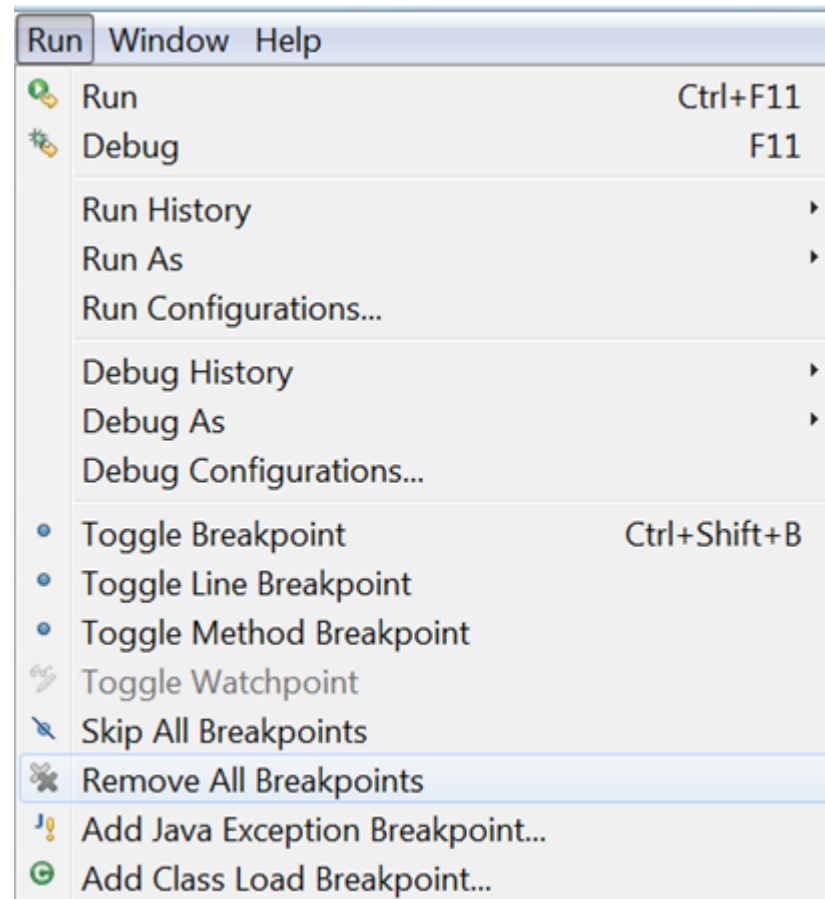
```
de [Java Application]  
ppe at localhost:1883  
ad [main] (Suspended (breakpoint at line 9 in Tr  
eppe.zeichneTreppe(int, int) line: 9  
stTreppe.main(String[]) line: 4  
gram Files\Java\jre7\bin\javaw.exe (26.03.2015 1
```


Nach dem Debuggen

zurück zur Programm-Ansicht



ggf. Breakpoint wieder löschen



Verständnisfragen



- Wann braucht man einen Debugger?
- Was kann man mit Hilfe eines Debuggers tun?