

# Effizientes Programmieren

## Debugging in Eclipse



Effizientes Programmieren (19.05.2015)

Pit Pietsch

- 1 Introduction
- 2 Definitionen
- 3 Fehlerarten
- 4 Debugging in Eclipse
- 5 Hausaufgabe

## Praktikum

P. Pietsch

Introduction

Definitionen

Fehlerarten

Debugging in  
Eclipse

Hausaufgabe

# Section 1

## Introduction



## Praktikum

P. Pietsch

Introduction

Definitionen

Fehlerarten

Debugging in  
Eclipse

Hausaufgabe

- Softwaresysteme sind sehr komplex
- Entwickler sind auch nur Menschen
- Programme ohne Fehler sind selten

Als **stabil** gilt ein Programm, wenn es weniger als 0.5 Fehler auf 1000 Zeilen Code aufweist.

## Praktikum

P. Pietsch

Introduction

Definitionen

Fehlerarten

Debugging in  
Eclipse

Hausaufgabe

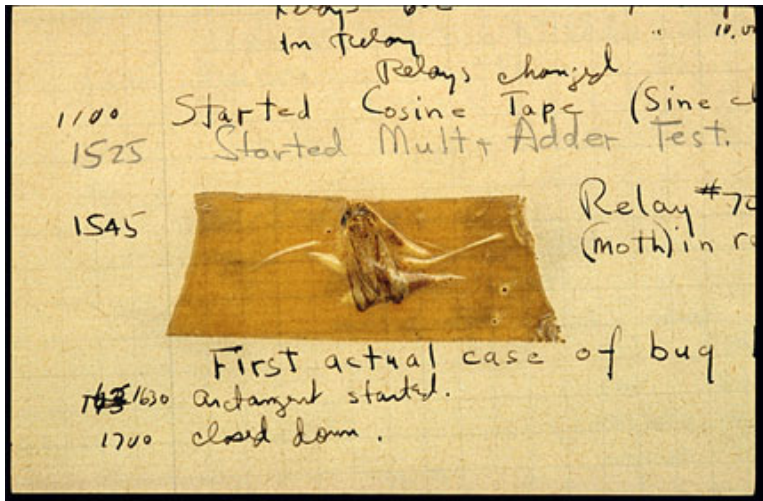


Figure: Gefunden in einem Mark II Aiken Relay Calculator

## Praktikum

P. Pietsch

Introduction

Definitionen

Fehlerarten

Debugging in  
Eclipse

Hausaufgabe

Debugging ist ein methodischer, in der Regel tool-unterstützter, Prozess um Fehler in Quellcode zu finden und diese zu beheben.

Die Ziele des Debuggings sind insbesondere Fehler zu

- ... reproduzieren
- ... isolieren
- ... beheben

## Praktikum

P. Pietsch

Introduction

Definitionen

Fehlerarten

Debugging in  
Eclipse

Hausaufgabe

Reproduzieren kann mit der schwierigste Part sein, weil nicht alle Fehler sind einfach reproduzierbar:

- Keine Garantie über Reihenfolge der Bearbeitung
- Verzögerung zwischen Fehler und Manifestation
- Nebenläufigkeiten (Threads, Synchronisierung, . . .)

## Praktikum

P. Pietsch

Introduction

Definitionen

Fehlerarten

Debugging in  
Eclipse

Hausaufgabe

Wenn der Fehler reproduzierbar ist, dann kann man sich in Ruhe anschauen was genau falsch läuft und danach den Fehler beheben.



## Praktikum

P. Pietsch

Introduction

Definitionen

Fehlerarten

Debugging in  
Eclipse

Hausaufgabe

# Section 2

## Definitionen

**Praktikum**

P. Pietsch

Introduction

Definitionen

Fehlerarten

Debugging in  
Eclipse

Hausaufgabe

Abweichung des IST (beobachtete, ermittelte, berechnete Zustände oder Vorgänge) vom SOLL (festgelegte, korrekte Zustände und Vorgänge), wenn sie die vordefinierte Toleranzgrenze (die auch 0 sein kann) überschreitet.

## Praktikum

P. Pietsch

Introduction

Definitionen

Fehlerarten

Debugging in  
Eclipse

Hausaufgabe

Nach dem *International Software Testing Qualifications Board* setzt sich ein Fehler aus den folgenden Zusammenhängen zusammen:

- **Fehlhandlung:** eine menschliche Handlung, die zu einem Fehlerzustand führt
- **Fehlerzustand:** Ein Defekt einer Systemkomponente, der die geforderte Funktion des Produkts beeinträchtigen kann.
- **Fehlerwirkung:** Die Manifestierung eines Defekts, der bei Ausführung des Systems zu einem inkorrekten Verhalten führt.

## Praktikum

P. Pietsch

Introduction

Definitionen

Fehlerarten

Debugging in  
Eclipse

Hausaufgabe

# Section 3

## Fehlerarten



## Praktikum

P. Pietsch

Introduction

Definitionen

Fehlerarten

Debugging in  
Eclipse

Hausaufgabe

Kompilierfehler verhindern das korrekte Kompilieren des Programms. Man kann zwischen zwei Arten unterscheiden:

- **Lexikalische Fehler** sind nicht interpretierbare Zeichenketten (unbekannte Befehle, Variablenamen, ...)
- **Syntaktische Fehler** sind Verstöße gegen die Grammatik der Programmiersprache (Klammern, Typen, ...)

Beide Fehlerarten lassen sich statisch überprüfen und werden in den meisten Entwicklungsumgebungen direkt angezeigt, z.B. bei Eclipse rot unterkringelt.

## Praktikum

P. Pietsch

Introduction

Definitionen

Fehlerarten

Debugging in  
Eclipse

Hausaufgabe

- **Semantische Fehler:** Syntaktisch fehlerfreier, aber inhaltlich fehlerhafter Code (Falsche Befehle genutzt ...).
- **Logische Fehler:** Falscher Lösungsansatz...
- **Design Fehler:** Richtiger Ansatz, falsch umgesetzt. Fehler im Grundkonzept
- **Laufzeitfehler:** Fehler, der während der Ausführung des Programms auftritt. Führt zu Programmabsturz oder fehlerhaften Ergebnissen.

## Praktikum

P. Pietsch

Introduction

Definitionen

Fehlerarten

Debugging in  
Eclipse

Hausaufgabe

# Section 4

## Debugging in Eclipse

## Praktikum

P. Pietsch

Introduction

Definitionen

Fehlerarten

Debugging in  
Eclipse

Hausaufgabe

Debugging erlaubt die Schritt-für-Schritt Ausführung eines Programmes. Dabei können der Sourcecode, Variablen, etc., zur Laufzeit betrachtet analysiert werden.

Debugging erlaubt es, die Ausführung des Programmcodes an einer beliebigen Stelle anzuhalten. Dies kann geschehen an ...

- einer bestimmten Programmzeile (Breakpoint)
- im Fehlerfall (Exception Breakpoint)
- unter bestimmten Bedingungen (Conditional Breakpoint)
- wenn sich Werte ändern (Watchpoint)



## Praktikum

P. Pietsch

Introduction

Definitionen

Fehlerarten

Debugging in  
Eclipse

Hausaufgabe

Eclipse erlaubt es ein Programm im Debugging-Modus zu starten. Sollte in diesem Modus ein Breakpoint/Watchpoint aktiviert werden, dann wechselt Eclipse in die *Debugging Perspective*. Diese bietet verschiedene Views mit denen die Ausführung des Programmes kontrolliert werden kann.

## Praktikum

P. Pietsch

Introduction

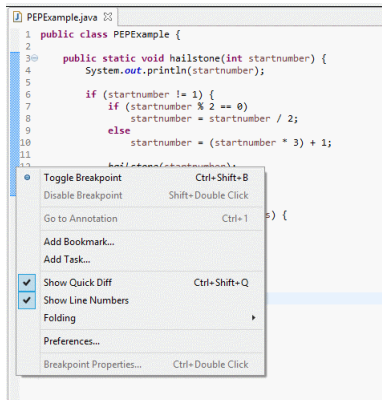
Definitionen

Fehlerarten

Debugging in  
Eclipse

Hausaufgabe

Klick mit der rechten Maustaste auf den schmalen Rahmen am Rand des Sourcecode-Editors - *Toggle Breakpoint*



```
1 public class PEPExample {
2
3 public static void hailstone(int startnumber) {
4     System.out.println(startnumber);
5
6     if (startnumber != 1) {
7         if (startnumber % 2 == 0)
8             startnumber = startnumber / 2;
9         else
10            startnumber = (startnumber * 3) + 1;
11
12    }
13 }
```

## Praktikum

P. Pietsch

Introduction

Definitionen

Fehlerarten

Debugging in  
Eclipse

Hausaufgabe

Beim Ausführen im Debug Modus wird dann die Ausführung des Programms an der gesetzten Zeile angehalten und es wird in die Debug Perspektive gewechselt.



## Praktikum

P. Pietsch

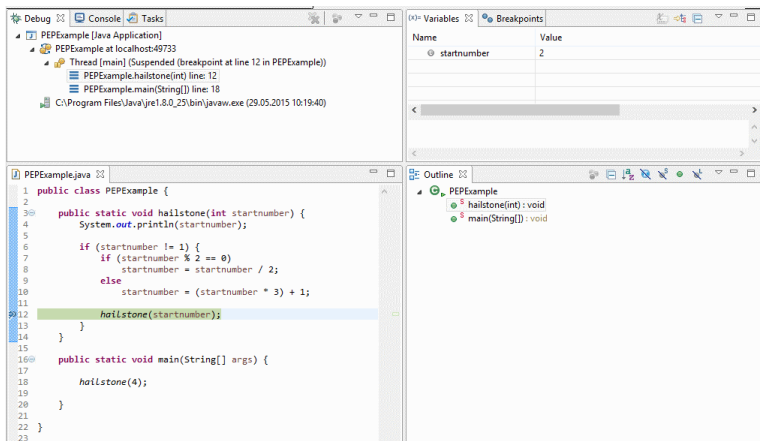
Introduction

Definitionen

Fehlerarten

 Debugging in  
Eclipse

Hausaufgabe



The screenshot shows the Eclipse IDE in Debug Perspective. The top-left pane displays the Debug Console and the Hierarchy view, showing the application structure with a breakpoint set at line 12 in the `PEPEExample` class. The top-right pane shows the Variables window with a table:

Name	Value
startnumber	2

The bottom-left pane shows the source code of `PEPEExample.java` with line 12 highlighted:

```

1 public class PEPEExample {
2
3 public static void hailstone(int startnumber) {
4     System.out.println(startnumber);
5
6     if (startnumber != 1) {
7         if (startnumber % 2 == 0)
8             startnumber = startnumber / 2;
9         else
10            startnumber = (startnumber * 3) + 1;
11
12    hailstone(startnumber);
13    }
14 }
15
16 public static void main(String[] args) {
17     hailstone(4);
18 }
19
20 }
21
22 }
23
  
```

The bottom-right pane shows the Outline view with the following structure:

- PEPEExample
  - hailstone(int): void
  - main(String[]): void

## Praktikum

P. Pietsch

Introduction

Definitionen

Fehlerarten

Debugging in  
Eclipse

Hausaufgabe

In der Debug Perspektive kann die weitere Ausführung des Programms schrittweise gesteuert werden.

- **F5** führt die aktuelle Zeile Code aus, hält danach wieder an.
- **F6** führt die aktuelle Zeile Code aus, ohne in Methodenaufrufe abzustiegen, hält danach an.
- **F7** führt die aktuelle Methode zu Ende aus, hält dann nach dem Aufrufer wieder an.
- **F8** führt das Programm weiter aus, hält bei dem nächsten Breakpoint an.



F8      Stop      F5    F6    F7

## Praktikum

P. Pietsch

Introduction

Definitionen

Fehlerarten

Debugging in  
Eclipse

Hausaufgabe

In dieser View werden die gerade aktuellen Variablen und ihre Werte angezeigt. Diese können auch zur Laufzeit verändert werden.

Name	Value
startnumber	2

## Praktikum

P. Pietsch

Introduction

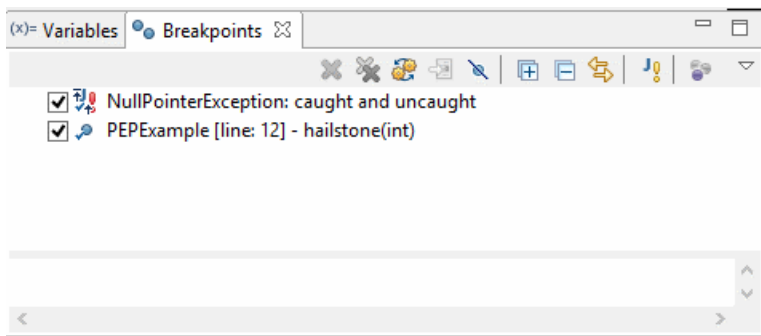
Definitionen

Fehlerarten

Debugging in  
Eclipse

Hausaufgabe

Breakpoints können über den *Breakpoint View* bequem an/ausgeschaltet werden. Ausserdem können einzelne Breakpoints oder alle Breakpoints gelöscht werden.



## Praktikum

P. Pietsch

Introduction

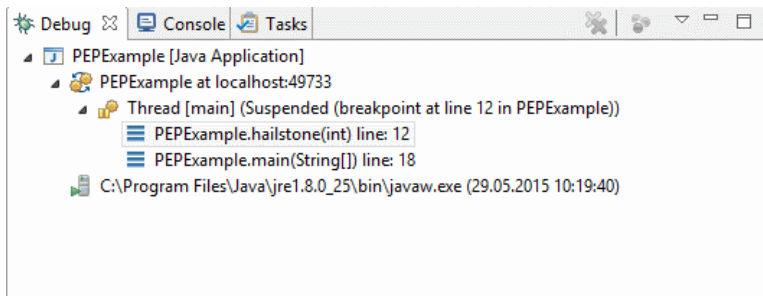
Definitionen

Fehlerarten

Debugging in  
Eclipse

Hausaufgabe

Der Execution Stack zeigt welche Teile eines Programmes gerade ausgeführt werden und wie sie ineinandergreifen.



The screenshot shows the Eclipse IDE's Debug Console. The top bar includes 'Debug', 'Console', and 'Tasks' tabs. The main area displays the execution stack for 'PEPEXample [Java Application]'. The stack is as follows:

- PEPEXample at localhost:49733
  - Thread [main] (Suspended (breakpoint at line 12 in PEPEXample))
    - PEPEXample.hailstone(int) line: 12
    - PEPEXample.main(String[]) line: 18
  - C:\Program Files\Java\jre1.8.0\_25\bin\javaw.exe (29.05.2015 10:19:40)



## Praktikum

P. Pietsch

Introduction

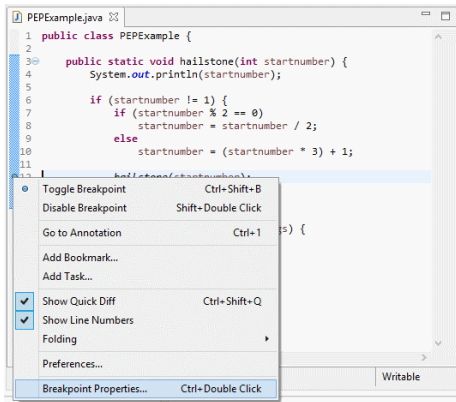
Definitionen

Fehlerarten

Debugging in  
Eclipse

Hausaufgabe

Öffnet man das Kontext-Menü auf einem Breakpoint, dann kann man seine Eigenschaften unter *Breakpoint Properties* ändern...



Unter den Breakpoint Properties kann man verschiedenste Eigenschaften des Breakpoints verändern.

## Praktikum

P. Pietsch

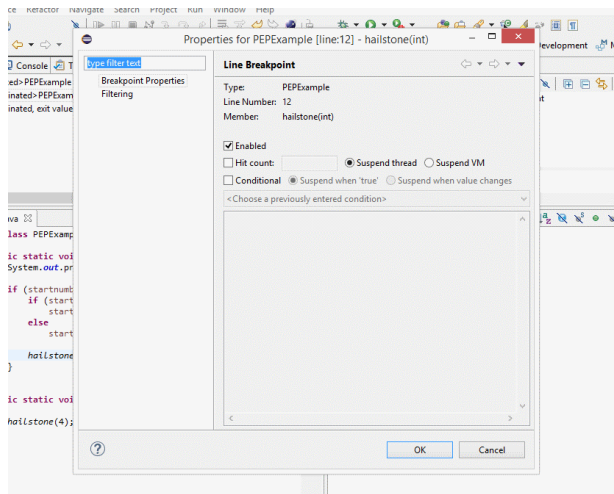
Introduction

Definitionen

Fehlerarten

Debugging in  
Eclipse

Hausaufgabe



## Praktikum

P. Pietsch

Introduction

Definitionen

Fehlerarten

Debugging in  
Eclipse

Hausaufgabe

Die beiden wichtigsten Eigenschaften sind:

- **Hit Count:** Der Breakpoint schlägt nur an, wenn die Zeile zum x-ten Mal ausgeführt wurde.
- **Conditional:** Der Breakpoint schlägt nur an, wenn die Bedingung erfüllt ist. Beispielsweise, wenn eine Variable einen bestimmten Wert hat.

## Praktikum

P. Pietsch

Introduction

Definitionen

Fehlerarten

Debugging in  
Eclipse

Hausaufgabe

- **Method Breakpoint:** Hält an wenn die Methode ausgeführt wird.
- **Watch Point:** Hält an, wenn sich der Wert eines Feldes ändert.
- **Exception Breakpoint:** Hält an, wenn eine Exception geworfen wird.

## Praktikum

P. Pietsch

Introduction

Definitionen

Fehlerarten

Debugging in  
Eclipse

Hausaufgabe

# Section 5

## Hausaufgabe

## Praktikum

P. Pietsch

Introduction

Definitionen

Fehlerarten

Debugging in  
Eclipse

Hausaufgabe

`http://eclipsetutorial.sourceforge.net/debugger.html`