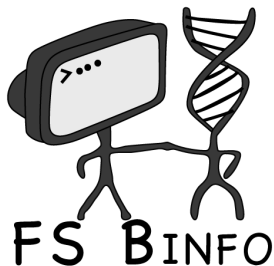


# Debuggen mit Eclipse



Anna Dieckmann  
Lorenz Maier

Fachschaft Bioinformatik

12. Januar 2009

# Was ist Debuggen?

---

## Debugging

- Vorgang der Fehlersuche und der Fehlerbeseitigung in einem Programm oder Quellcode
- kann entweder **manuell** oder mit Hilfe eines **Debuggers** durchgeführt werden

# Manuell Debuggen

---

- von Hand nachrechnen („Mini-Beispiel“)
- `System.out.println()` verwenden

```
int a = 0;
int i = 0;
int j = 0;

while (i < 10) {
    j = 0;
    while (j < 10) {
        a++;
        j++;
    }
}
```

⇒ Was macht dieser Code falsch?

- Lösung: Die äußere `while`-Schleife terminiert nicht, da das Inkrement von `i` fehlt.

# Debuggen mit Eclipse

---

- **Debug Perspective:** Code, Output, Variablen, Breakpoints, etc.
- **Breakpoints:** Programm pausieren lassen
- Resume, Step Into, Step Over, Terminate
- Conditional Breakpoints
- Ausführen mit dem kleinen Käfer neben dem *run*-Button

Java - DebuggerBsp/src/bugs/GaussFormel.java - Eclipse SDK

File Edit Source Refactor Navigate Search Project Run Window Help

DebuggerBsp

- asdbio
- CBS
- DebuggerBsp
  - src
    - bugs
    - entities
  - JRE System Library [java-...
  - epic\_links
  - Rezepte
  - taxonArrays

GaussFormel.java

```
1 package bugs;
2
3 public class GaussFormel {
4
5     public static void main(String[] args) {
6         int n = 100;
7         int sum = 0;
8         int formel = (n*(n+1))/2;
9
10        for(int i=1; i<n; i++) {
11            sum += i;
12        }
13        System.out.println("Aufsummieren ergibt: " + sum
14            + " und die Gauß-Formel: " + formel);
15        System.out.print("Vergleich der Berechnung: ");
16        System.out.print(sum==formel);
17    }
18 }
19
```

Outline

- bugs
  - GaussFormel
    - main(String[]): void

Console

```
<terminated> GaussFormel [Java Application] /usr/lib/jvm/java-6-openjdk/bin/java (Jan 12, 2010 12:07:04 AM)
Aufsummieren ergibt:4950 und die Gauß-Formel: 5050
Vergleich der Berechnung: false
```

Debug - DebuggerBsp/src/bugs/GaussFormel.java - Eclipse SDK

File Edit Source Refactor Navigate Search Project Run Window Help

Debug Hierarchy Variables Breakpoints Expressions

<terminated>GaussFormel [Java Application]  
    <terminated, exit value: 0>/usr/lib/jvm/java-6-openjdk/bin/java [Jan

GaussFormel.java

DebuggerBsp > src > bugs > GaussFormel > main(String[]) : void

```
4
5 public static void main(String[] args) {
6     int n = 100;
7     int sum = 0;
8     int formel = (n*(n+1))/2;
9
10    for(int i=1; i<n; i++) {
11        sum += i;
12    }
13    System.out.println("Aufsummieren ergibt:" + sum
14                        + " und die Gauß-Formel: " + formel);

```

Outline

- bugs
  - GaussFormel
    - main(String[]) : void

Console

<terminated> GaussFormel [Java Application] /usr/lib/jvm/java-6-openjdk/bin/java (Jan 12, 2010 12:07:04 AM)  
Aufsummieren ergibt:4950 und die Gauß-Formel: 5050  
Vergleich der Berechnung: false

Writable Smart Insert 10 : 33

Debug - DebuggerBsp/src/bugs/GaussFormel.java - Eclipse SDK

File Edit Source Refactor Navigate Search Project Run Window Help

Debug Hierarchy Variables Breakpoints Expressions

GaussFormel [Java Application]  
bugs.GaussFormel at localhost:33577  
Thread [main] (Suspended (breakpoint at line 11 in GaussFormel))  
GaussFormel.main(String[]) line: 11  
/usr/lib/jvm/java-6-openjdk/bin/java (Jan 12, 2010 12:10:48 AM)

GaussFormel.java  
DebuggerBsp > src > bugs > GaussFormel > main(String[]) : void

```
7   int sum = 0;  
8   int formel = (n*(n+1))/2;  
9  
10  for(int i=1; i<n; i++) {  
11      sum += i;  
12  }  
13  System.out.println("Aufsummieren ergibt: " + sum  
14      + " und die Gauß-Formel: " + formel);  
15  System.out.println("Trennen ist das Beste!");
```

Outline  
bugs  
GaussFormel  
main(String[]) : void

Console  
GaussFormel [Java Application] /usr/lib/jvm/java-6-openjdk/bin/java (Jan 12, 2010 12:10:48 AM)

Debug - DebuggerBsp/src/bugs/GaussFormel.java - Eclipse SDK

File Edit Source Refactor Navigate Search Project Run Window Help

Debug Hierarchy Variables Breakpoints Expressions

GaussFormel [Java Application]  
bugs.GaussFormel at localhost:41655  
Thread [main] (Suspended (breakpoint at line 13 in GaussFormel))  
GaussFormel.main(String[]) line: 13  
/usr/lib/jvm/java-6-openjdk/bin/java (Jan 12, 2010 12:18:36 AM)

Name	Value
args	String[] (id=15)
n	100
sum	4950
formel	5050

GaussFormel.java  
DebuggerBsp > src > bugs > GaussFormel > main(String[]): void

```
9  
10     for(int i=1; i<n; i++) {  
11         sum += i;  
12     }  
13     System.out.println("Aufsummieren ergibt: " + sum  
14         + " und die Gauß-Formel: " + formel);  
15     System.out.print("Vergleich der Berechnung: ");  
16     System.out.print(sum==formel);  
17
```

Outline  
bugs  
GaussFormel  
main(String[]): void

Console  
GaussFormel [Java Application] /usr/lib/jvm/java-6-openjdk/bin/java (Jan 12, 2010 12:18:36 AM)

Writable Smart Insert 13 : 1

Debug - DebuggerBsp/src/bugs/GaussFormel.java - Eclipse SDK

File Edit Source Refactor Navigate Search Project Run Window Help

Java Debug

Debug Hierarchy

GaussFormel [Java Application]

- bugs.GaussFormel at localhost:33577
  - Thread [main] (Suspended (breakpoint at line 11 in GaussFormel))
    - GaussFormel.main(String[]): line: 11

/usr/lib/jvm/java-6-openjdk/bin/java (Jan 12, 2010 12:10:48 AM)

Variables Breakpoints Expressions

Name	Value
n	100
sum	6
formel	5050
i	4

100

GaussFormel.java

DebuggerBsp > src > bugs > GaussFormel > main(String[]): void

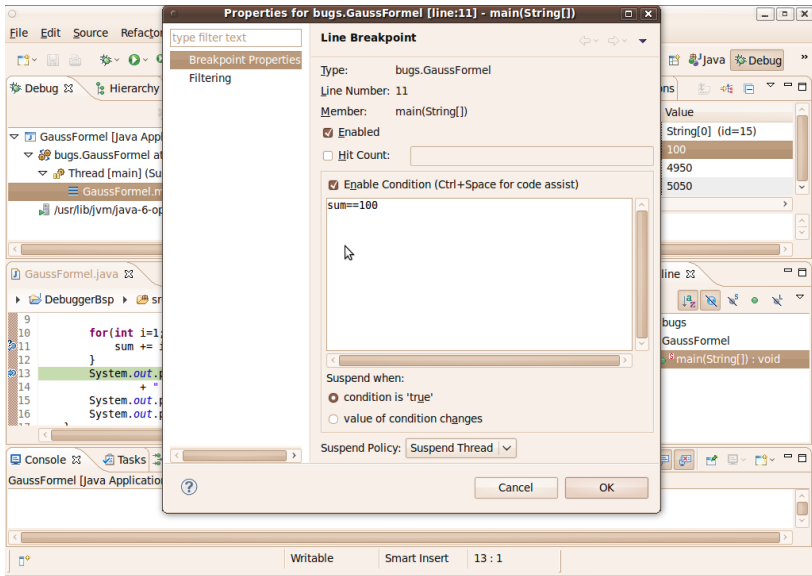
```
4
5 public static void main(String[] args) {
6     int n = 100;
7     int sum = 0;
8     int formel = (n*(n+1))/2;
9
10    for(int i=1; i<n; i++) {
11        sum += i;
```

Outline

- bugs
  - GaussFormel
    - main(String[]): void

Console Tasks Call Hierarchy Search

GaussFormel [Java Application] /usr/lib/jvm/java-6-openjdk/bin/java (Jan 12, 2010 12:10:48 AM)



# Typische Fehler

---

- Indexzählfehler / Laufvariable nicht gesetzt
- Null-Zeiger-Zugriff
- „Um eins daneben“-Fehler
- Rundungsfehler
- „Copy & Paste“ Artefakte
- Objekte falsch kopiert
- ...

## Aufgabe

Das Java-Projekt enthält verschiedene Klassen, die unterschiedlich schwere Fehler beinhalten. Arbeitet Euch soweit wie möglich durch diese durch. Wir werden uns am Ende die Lösungen gemeinsam anschauen.

## ssh = Secure SHell

- verschlüsselte Netzwerkverbindung zu einem Server/Computer
- `ssh -l Benutzername Servername`
- zB `maxmuster@musterPC:~$ ssh -l mustermann remote.cip.ifi.lmu.de`, wobei *mustermann* dem CIP-Login entspricht

## scp = Secure CoPy

- verschlüsselten Übertragung von Daten zwischen zwei Computern über ein Rechnernetz.
- `scp Benutzername@Servername:Quellpfad Zielpfad`
- zB `scp mustermann@remote.cip.ifi.lmu.de:/tmp/debugging.tar.gz .`
- `.` für Zielpfad

⇒ Datei wird in das aktuelle Verzeichnis kopiert

# Online-Tutorials

---

- <http://www.eclipse.org/resources/resource.php?id=405>
- <http://eclipsetutorial.sourceforge.net/debugger.html>
- <http://www.ibm.com/developerworks/library/os-ecbug/>