

# Datenbanken und Informationssysteme

## SoSe 2012

**- Anmerkungen zum Debugging -**  
**April 2012**

Verteilte Systeme und Informationssysteme (VSIS)  
Department Informatik  
Universität Hamburg



# Anmerkungen zum Debugging (1)

---

## ***Einfaches System.out.println()***

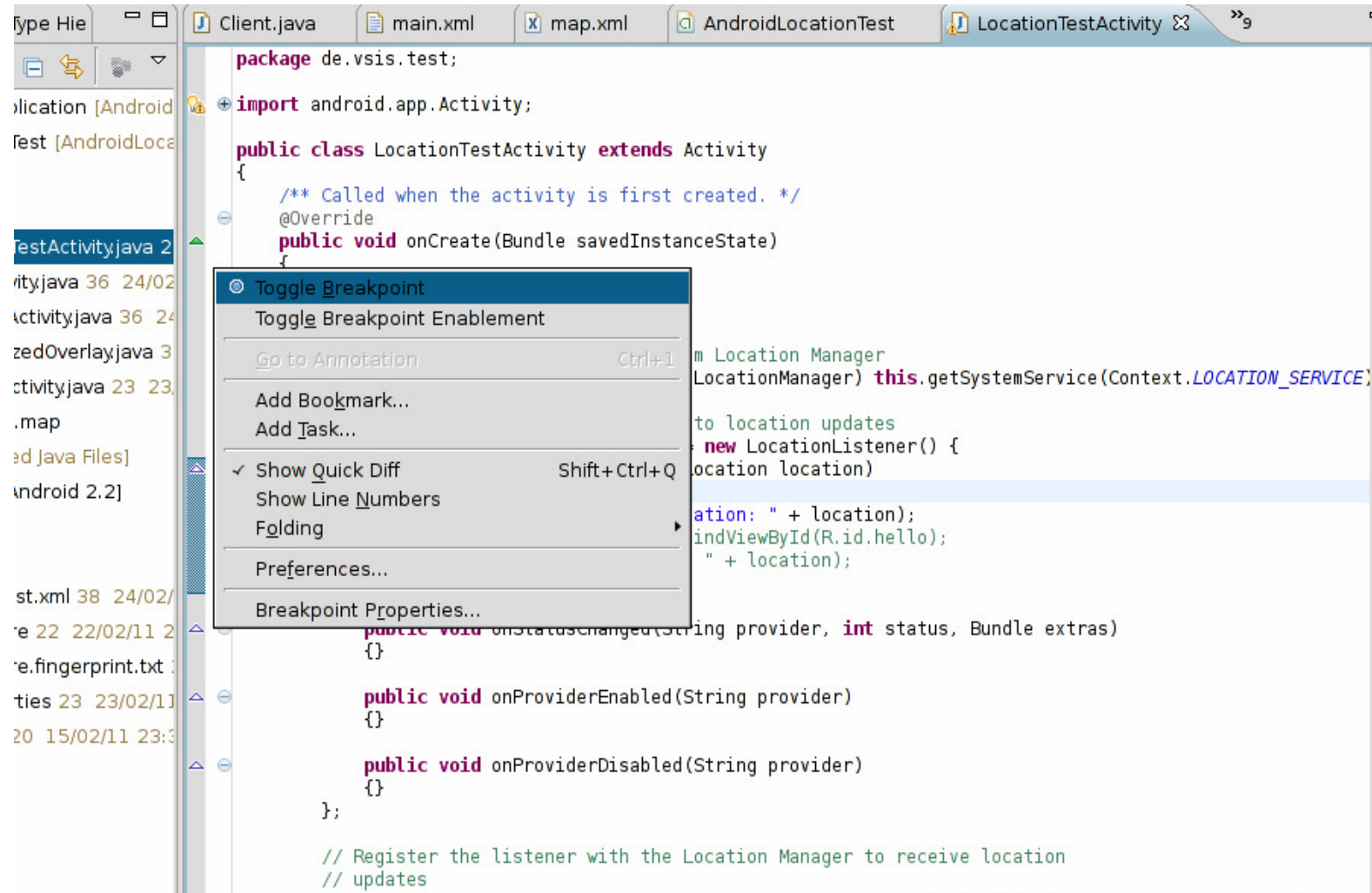
- ◆ Einbauen von print Statements:  

```
int foo = getFancySensorValue();  
System.out.println("Der Wert vom Sensor ist: "+foo);
```
- ◆ Nicht sehr schön / jedes Mal neu Tippen

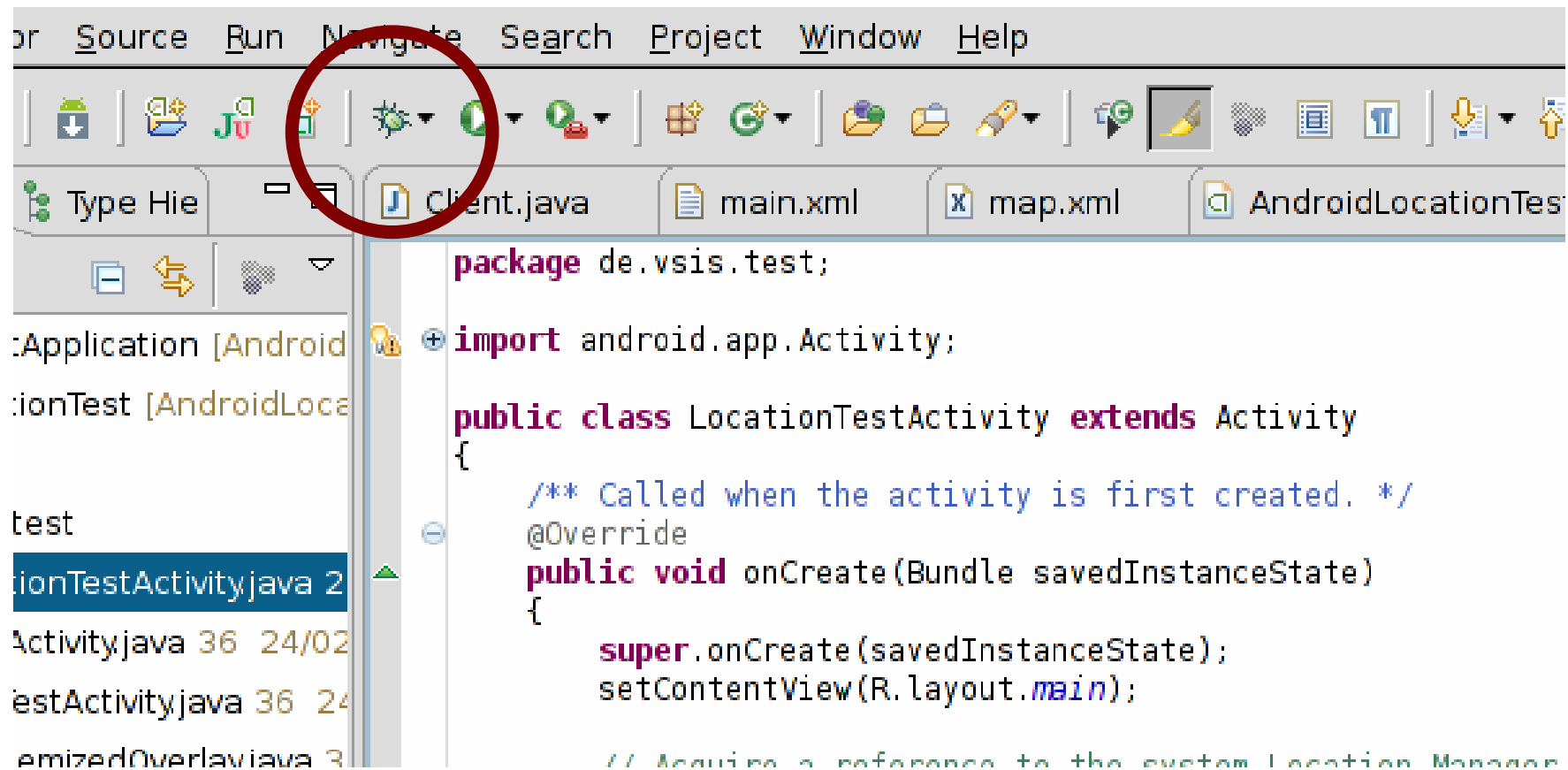
## ***Der Eclipse Debugger***

- ◆ Keine Quellcodeveränderung
- ◆ Inspektion von Variablen
- ◆ Setzen von Breakpoints

# Breakpoints setzen



# Debugger starten

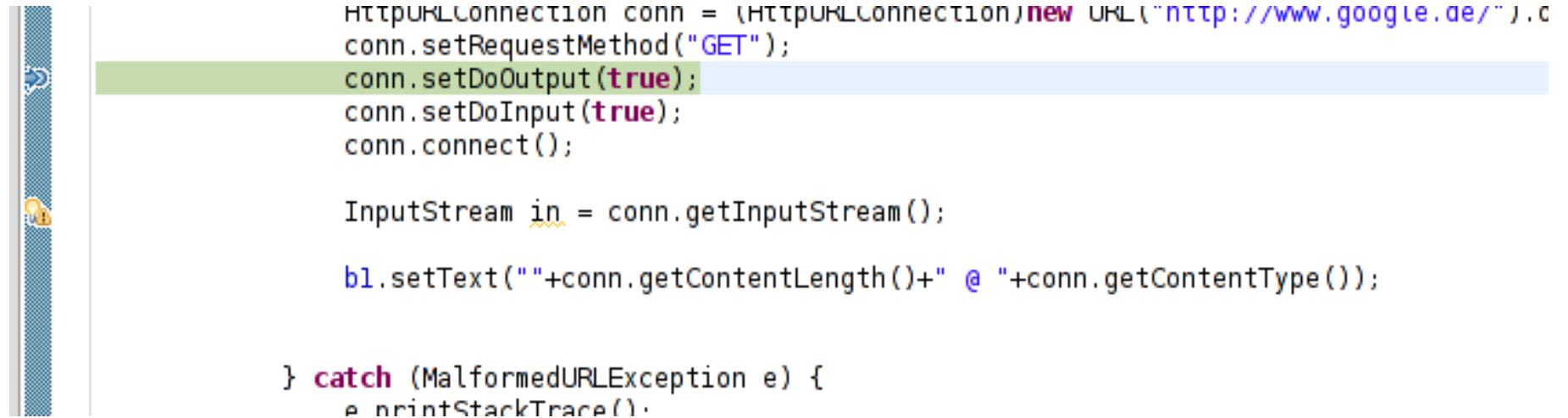


**Kleiner Käfer**



**der Run → Debug As → ...**

# Wenn der Debugger läuft



```
HttpURLConnection conn = (HttpURLConnection)new URL("http://www.google.de/").c
conn.setRequestMethod("GET");
conn.setDoOutput(true);
conn.setDoInput(true);
conn.connect();

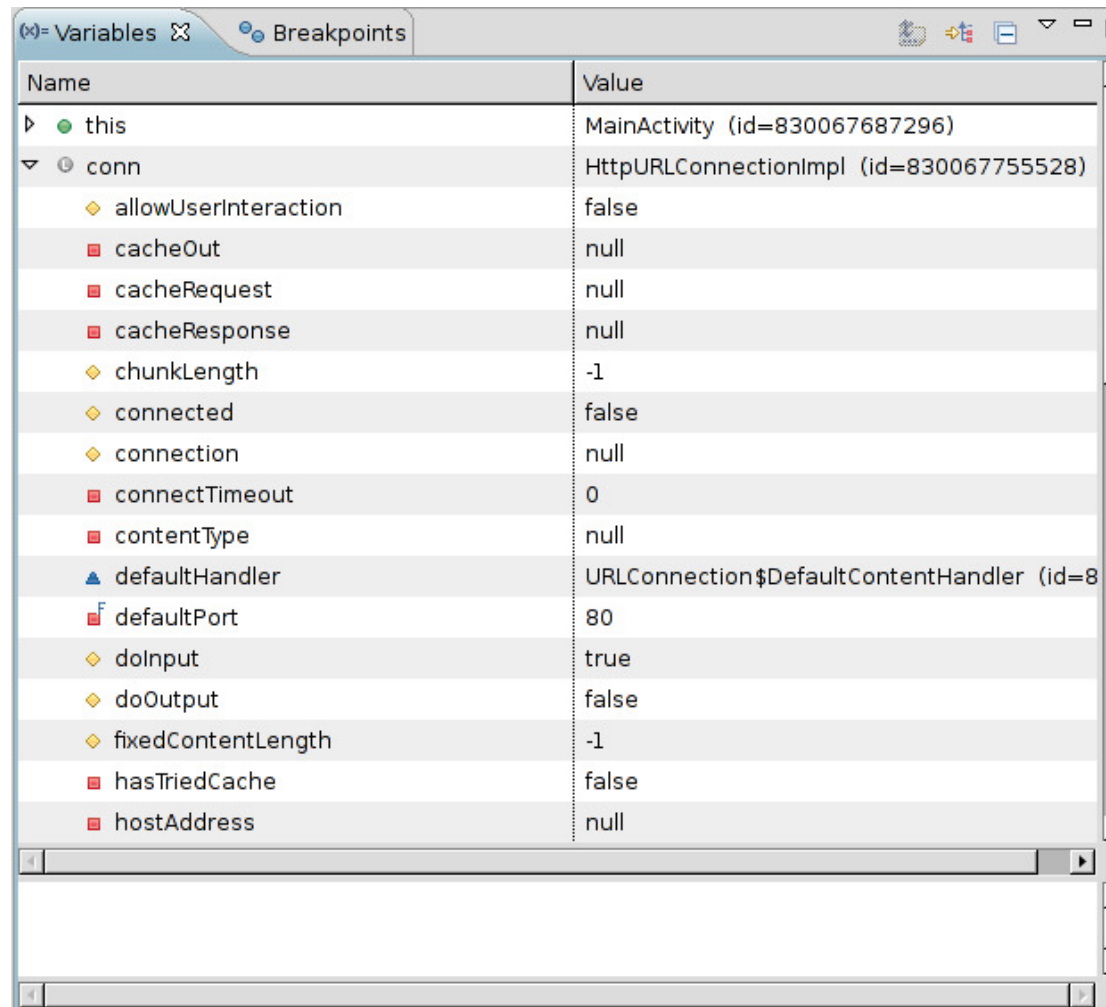
InputStream in = conn.getInputStream();

b1.setText(""+conn.getContentLength()+" @ "+conn.getContentType());

} catch (MalformedURLException e) {
    e.printStackTrace();
}
```

***Debugger pausiert die Anwendung, wenn Breakpoint erreicht ist***

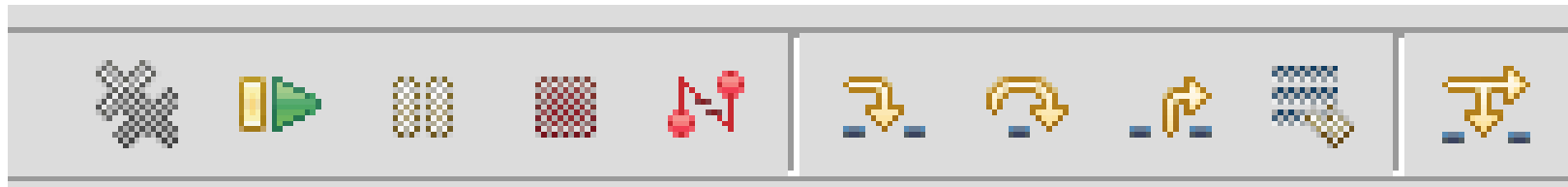
# Inspektion von Variablen



Name	Value
▶ this	MainActivity (id=830067687296)
▼ conn	URLConnectionImpl (id=830067755528)
allowUserInteraction	false
cacheOut	null
cacheRequest	null
cacheResponse	null
chunkLength	-1
connected	false
connection	null
connectTimeout	0
contentType	null
defaultHandler	URLConnection\$DefaultContentHandler (id=830067755528)
defaultPort	80
doInput	true
doOutput	false
fixedContentLength	-1
hasTriedCache	false
hostAddress	null

***Es können Variablen inspiziert oder weitere Breakpoints gesetzt werden.***

# Stepping



*springen („nächste*

*Über Statement weg-*

*Zeile") - F8*

*In Funktion springen - F6*

*Weiter (bis zum nächsten Breakpoint) - F8*

# Anmerkungen zum Debugging (2)

---

## ***Zusammenfassung***

- ♦ Sehr komfortables Werkzeug
- ♦ Gute Einsicht in Laufzeitverhalten
- ♦ Wenn eine Frage aufkommt:
  1. Lösung des Problems durch den Debugger?
  2. Falls nein: Kann der Debugger das Problem eingrenzen?

***Es ist keine Schande zu Debuggen! Im Gegenteil!***



# Ende

---

