

# Programmieren in Java

Peter Thiemann

Universität Freiburg, Germany

SS 2010

# Organisatorisches

- ▶ **Vorlesung** Mo, 16-17(18) Uhr ct, HS 101-00-026
- ▶ **Dozent** Prof. Dr. Peter Thiemann

Gebäude 079, Raum 00-015

Telefon: 0761 203 8051/8247

E-mail: `thiemann at informatik uni-freiburg de`

Web: <http://www.informatik.uni-freiburg.de/~thiemann>

- ▶ **Informationen** Homepage der Vorlesung über  
<http://proglang.informatik.uni-freiburg.de/>

# Literatur

- ▶ Felleisen, Findler, Flatt, Gray, Krishnamurthi, Proulx.  
*How to Design Classes*. 2008.
- ▶ Joshua Bloch.  
*Effective Java*. Prentice Hall, 2001.
- ▶ Peter Sestoft.  
*Java Precisely, Second Edition*. The MIT Press, August 2005.  
ISBN 0-262-69325-9.
- ▶ Martin Fowler.  
*UML Distilled, Third Edition*. Addison-Wesley, 2004.
- ▶ Matthias Felleisen, Dan Friedman.  
*A Little Java, A Few Patterns*. MIT Press, 1998.
- ▶ Arnd Poetzsch-Heffter.  
*Konzepte objektorientierter Programmierung. Mit einer Einführung in Java*. Springer, 2000.

# Organisation

Zwei Varianten:

- ▶ Bachelor Informatik
  - ▶ 1 SWS Vorlesung Mo 16-17
  - ▶ 2 SWS Übung
- ▶ Bachelor ESE
  - ▶ 2 SWS Vorlesung Mo 16-18
  - ▶ 2 SWS Übung

# Übungen

- ▶ 2 Stunden pro Woche – ab nächster Woche
- ▶ Kontakt: Konrad Anton  
`anton@informatik.uni-freiburg.de`

# Übungen

- ▶ 2 Stunden pro Woche – ab nächster Woche
- ▶ Kontakt: Konrad Anton  
anton@informatik.uni-freiburg.de
- ▶ erste Semesterhälfte: Betreutes Programmieren
  - ▶ 2 Stunden Zeit für ein Aufgabenblatt
  - ▶ jeder für sich im Pool
  - ▶ erforderlich: *Account*
  - ▶ keine Anwesenheitspflicht
  - ▶ zum **Bestehen** erforderlich: 33% der Punkte der letzten  $n$  Übungsblätter für alle  $n \geq 1$

# Übungen

- ▶ 2 Stunden pro Woche – ab nächster Woche
- ▶ Kontakt: Konrad Anton  
anton@informatik.uni-freiburg.de
- ▶ erste Semesterhälfte: Betreutes Programmieren
  - ▶ 2 Stunden Zeit für ein Aufgabenblatt
  - ▶ jeder für sich im Pool
  - ▶ erforderlich: *Account*
  - ▶ keine Anwesenheitspflicht
  - ▶ zum **Bestehen** erforderlich: 33% der Punkte der letzten  $n$  Übungsblätter für alle  $n \geq 1$
- ▶ zweite Semesterhälfte: Projekt
  - ▶ in Zweiergruppen
  - ▶ Abschlusspräsentation
  - ▶ zum **Bestehen** erforderlich: erfolgreiche Abschlusspräsentation

# Übungen

- ▶ 2 Stunden pro Woche – ab nächster Woche
- ▶ Kontakt: Konrad Anton  
anton@informatik.uni-freiburg.de
- ▶ erste Semesterhälfte: Betreutes Programmieren
  - ▶ 2 Stunden Zeit für ein Aufgabenblatt
  - ▶ jeder für sich im Pool
  - ▶ erforderlich: *Account*
  - ▶ keine Anwesenheitspflicht
  - ▶ zum **Bestehen** erforderlich: 33% der Punkte der letzten  $n$  Übungsblätter für alle  $n \geq 1$
- ▶ zweite Semesterhälfte: Projekt
  - ▶ in Zweiergruppen
  - ▶ Abschlusspräsentation
  - ▶ zum **Bestehen** erforderlich: erfolgreiche Abschlusspräsentation
- ▶ Anmeldung ab Dienstag 14.00st per Web (Adresse: siehe Homepage)



## Java

- ▶ gratis verfügbar: <http://java.sun.com/>,  
<http://www.ibm.com/developerworks/java/jdk/>
- ▶ mehr zu Java: ab der nächsten Woche

Entwicklungsumgebung: Eclipse

# Eclipse

## Info 1: DrScheme



(Quelle: Björn Laczay, CC-by-2.0)

# Eclipse

Jetzt: Eclipse

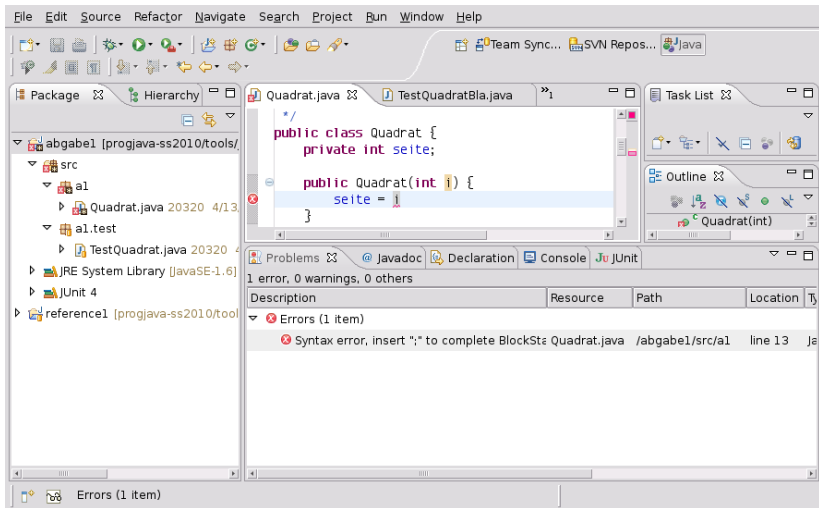


Eclipse (Quelle: Wikimedia Commons, public domain)

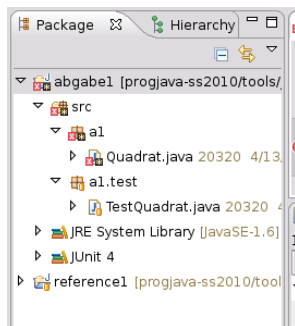
# Eclipse

- ▶ Entwicklungsumgebung
- ▶ Ursprünglich von IBM (Nachfolger der VisualAge-Serie)
- ▶ Plug-In-Architektur
- ▶ Plug-Ins für fast alles (Programmiersprachen, Versionsverwaltungen, Modellierung, Diagramme, ... )  
verfügbar
- ▶ bei uns: JDT (Java Development Tools)
- ▶ <http://www.eclipse.org>

# Eclipse

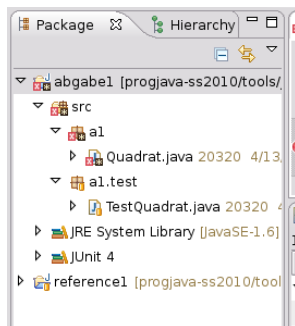


# Workspace



- ▶ Workspace enthält Projekte
- ▶ Verschiedene Arten von Projekten (z.B. Java, L<sup>A</sup>T<sub>E</sub>X, ...)
- ▶ Jedes Projekt enthält einen Verzeichnisbaum

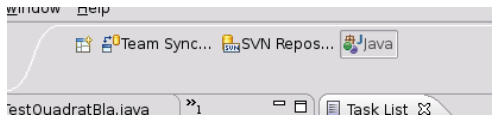
# Workspace



- ▶ Workspace enthält Projekte
- ▶ Verschiedene Arten von Projekten (z.B. Java,  $\text{\LaTeX}$ , ...)
- ▶ Jedes Projekt enthält einen Verzeichnisbaum
- ▶ vorläufig reicht uns ein Projekt (gleichzeitig)

# Oberfläche

- ▶ Editoren, Views
  - ▶ Java Editor
  - ▶ Package Explorer
  - ▶ Problems View
- ▶ Vorgefertigte Arrangements von Editoren und Views:  
Perspektiven



- ▶ Java
- ▶ Java Debugging



# Eclipse

Demo.