

Programmieren in Java

Einführung

Prof. Dr. Peter Thiemann

Albert-Ludwigs-Universität Freiburg, Germany

SS 2015

Koordinaten

▶ **Vorlesung:** Mo, 16ct-18 Uhr, HS 101-00-026

▶ **Dozent:** Prof. Dr. Peter Thiemann

Gebäude 079, Raum 00-015

Telefon: 0761 203 -8051/-8247

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

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

▶ **Homepage der Vorlesung:**

`http://proglang.informatik.uni-freiburg.de/teaching/java/2015`

Kommunikation

- ▶ Folienvortrag, Aufzeichnung

Kommunikation

- ▶ Folienvortrag, Aufzeichnung
- ▶ Live-Hacking

Kommunikation

- ▶ Folienvortrag, Aufzeichnung
- ▶ Live-Hacking
- ▶ Interaktion: Twitter #ufrjava @ProglangUniFr

Kommunikation

- ▶ Folienvortrag, Aufzeichnung
- ▶ Live-Hacking
- ▶ Interaktion: Twitter #ufrjava @ProglangUniFr
- ▶ Forum
(<https://daphne.informatik.uni-freiburg.de/forum/viewforum.php?f=383>)

Kommunikation

- ▶ Folienvortrag, Aufzeichnung
- ▶ Live-Hacking
- ▶ Interaktion: Twitter #ufrjava @ProglangUniFr
- ▶ Forum
(<https://daphne.informatik.uni-freiburg.de/forum/viewforum.php?f=383>)
- ▶ Subversion (Einreichen von Lösungen)

Ziele aus dem Modulhandbuch

Die Studierenden erlernen weiterführende objekt-orientierte Programmierkonzepte anhand der Sprache Java. Sie können eigenständig Programme unter Verwendung einer graphischen Entwicklungsumgebung für Java erstellen. Sie sind in der Lage, Java-Bibliotheken in eigenen Programmen einzusetzen.

Inhalt

- ▶ Objekt-orientierte Programmierkonzepte
- ▶ Benutzung der Java Entwicklungswerkzeuge (Eclipse)
- ▶ Erstellung von Java Programmen
- ▶ Benutzung von Java Bibliotheken
- ▶ Ressourcen zu den Themen Objekt-Orientierung, Java, Software-Entwicklung
- ▶ Eigenständige Recherche

... und was dahinter steckt

- ▶ Programmieren erfordert
 - ▶ Beherrschung einer Programmiersprache
 - ▶ Verständnis des zu lösenden Problems
 - ▶ Kreativität
 - ▶ Arbeitseinsatz

... und was dahinter steckt

- ▶ Programmieren erfordert
 - ▶ Beherrschung einer Programmiersprache
 - ▶ Verständnis des zu lösenden Problems
 - ▶ Kreativität
 - ▶ Arbeitseinsatz
- ▶ Wie werde ich ein (besserer) Programmierer?

... und was dahinter steckt

- ▶ Programmieren erfordert
 - ▶ Beherrschung einer Programmiersprache
 - ▶ Verständnis des zu lösenden Problems
 - ▶ Kreativität
 - ▶ Arbeitseinsatz
- ▶ Wie werde ich ein (besserer) Programmierer?
- ▶ Programmierung kennt „Standardsituationen“, für die es bewährte Lösungen (*Patterns* oder *Best Practices*) gibt.

... und was dahinter steckt

- ▶ Programmieren erfordert
 - ▶ Beherrschung einer Programmiersprache
 - ▶ Verständnis des zu lösenden Problems
 - ▶ Kreativität
 - ▶ Arbeitseinsatz
- ▶ Wie werde ich ein (besserer) Programmierer?
- ▶ Programmierung kennt „Standardsituationen“, für die es bewährte Lösungen (*Patterns* oder *Best Practices*) gibt.
- ▶ Ziel der Vorlesung ist also:
 - ▶ Sie kennen die Standardsituationen + Lösungsmöglichkeiten.
 - ▶ Sie verschwenden keine Zeit mit Standardsituationen, sondern können Ihre Energie für Verständnis und Kreativität einsetzen.

... und was dahinter steckt

- ▶ Programmieren erfordert
 - ▶ Beherrschung einer Programmiersprache
 - ▶ Verständnis des zu lösenden Problems
 - ▶ Kreativität
 - ▶ Arbeitseinsatz
- ▶ Wie werde ich ein (besserer) Programmierer?
- ▶ Programmierung kennt „Standardsituationen“, für die es bewährte Lösungen (*Patterns* oder *Best Practices*) gibt.
- ▶ Ziel der Vorlesung ist also:
 - ▶ Sie kennen die Standardsituationen + Lösungsmöglichkeiten.
 - ▶ Sie verschwenden keine Zeit mit Standardsituationen, sondern können Ihre Energie für Verständnis und Kreativität einsetzen.
- ▶ Chance der Übung
 - ▶ Trainieren der Standardsituationen

Literatur (Design und Programmierung)

- ▶ Felleisen, Findler, Flatt, Gray, Krishnamurthi, Proulx.
How to Design Classes. 2012.
<http://www.ccs.neu.edu/home/matthias/HtDC/htdc.pdf>
- ▶ Matthias Felleisen, Dan Friedman.
A Little Java, A Few Patterns. MIT Press, 1997.
- ▶ Allen Holub.
Holub on Patterns: Learning Design Patterns by Looking at Code.
Apress, 2004.
- ▶ Steve McConnell.
Code Complete: A Practical Handbook of Software Construction, Second Edition. Microsoft Press, 2004.
- ▶ Arnd Poetzsch-Heffter.
Konzepte objektorientierter Programmierung. Mit einer Einführung in Java. 2. Auflage. Springer, 2009.

Literatur (Java, UML)

- ▶ Peter Sestoft.
Java Precisely, Second Edition. The MIT Press, August 2005. ISBN 0-262-69325-9.
- ▶ James Gosling, Bill Joy, Guy Steele, Gilad Bracha, Alex Buckley.
The Java Language Specification, Java SE 8 Edition .
Addison-Wesley Professional, 2014.
- ▶ Joshua Bloch.
Effective Java. Prentice Hall, 2001.
- ▶ Martin Fowler.
UML Distilled, Third Edition. Addison-Wesley, 2003.
- ▶ Kirill Fakhroutdinov.
The Unified Modeling Language. <http://www.uml-diagrams.org/>

Übungen

- ▶ **Poolstunden:**

Mi 12-14 ct Pool 082-028/029

Do 14-16 ct Pool 082-021/029

Do 16-18 ct Pool 082-028/029

- ▶ **Betreuung:** Luminous Fennell, Manuel Geffken

Gebäude 079, Raum 00-013

Telefon: 0761 203 -8053/-8050

E-mail: {fennell, geffken} at cs uni-freiburg de

- ▶ **Tutoren:** David Spisla, David Zschocke, Gabriel Kalweit, Jannis Limperg, Max Lohmann, Nicolas Riesterer, Sebastian Dufner

- ▶ Teilnahme freiwillig, die Tutoren helfen bei den Aufgaben

- ▶ Empfehlung: hingehen, falls ein Blatt mehr als 2-3 Stunden in Anspruch nimmt

Übungsmodus

- ▶ **Neues Aufgabenblatt:** i.d.R. montags auf Daphne, nach der Vorlesung.
- ▶ **Abgabe:** Einzelabgabe, keine Teams.
i.d.R. sonntags 23:59h, über Subversion/Eclipse.
(Je nach Fortschritt der Vorlesung kann es auch zweiwöchige Blätter geben)
- ▶ Zum **Bestehen des Kurses** müssen
 - ▶ Blatt 0 vollständig bearbeitet und abgegeben werden (Ausgabe: heute!)
 - ▶ mindestens 50% der Gesamtpunktzahl erreicht werden.
- ▶ **Kooperation:** Ideen austauschen: ✓, diskutieren: ✓, Code austauschen: ✗, Code aus dem Internet kopieren: ✗.
- ▶ **Jede Abgabe muss ausschließlich selbstgeschriebenen Code enthalten.** Wir führen eine stichprobenartige Überprüfung nach Plagiaten durch. Alle Beteiligten an einem entdeckten Plagiatsversuch werden von der weiteren Teilnahme ausgeschlossen und der Kurs wird als nicht bestanden gewertet.

Get Started!

1. Bei Daphne für “ProgrammierenJavaSS2015” anmelden
`https://daphne.informatik.uni-freiburg.de/ss2015/ProgrammierenJavaSS2015/`
2. Übungsblatt 0 herunterladen
3. Bei Problemen in die Übung gehen
4. Ende der Woche wird Ihnen ein Tutor zugeteilt

Fragen?

