

# Objektorientierte Softwareentwicklung

Vorlesung im Sommersemester 2010

---

Prof. Dr. Andreas Weber

[weber@cs.uni-bonn.de](mailto:weber@cs.uni-bonn.de)

Dr. Günter Kniesel

[gk@cs.uni-bonn.de](mailto:gk@cs.uni-bonn.de)

# Das Team

## ● Ihr Dozent



Dr. Günter Kniesel

Institut für Informatik III

Römerstr. 164

Raum A 107

E-Mail: [gk@cs.uni-bonn.de](mailto:gk@cs.uni-bonn.de)

Tel.: 73-4511

Forschungsgebiete

- Softwaretechnologie
- Programmiersprachen
- Software-Analyse und -Transformation

## ● Ihre Tutoren

◆ [oose-tutoren@lists.iai.uni-bonn.de](mailto:oose-tutoren@lists.iai.uni-bonn.de)

## ● Ihre Kommilitonen

◆ [oose-vorlesung@lists.iai.uni-bonn.de](mailto:oose-vorlesung@lists.iai.uni-bonn.de)

# Vorlesung „Objektorientierte Softwareentwicklung“

Kapitel 0: Organisatorisches und grober Themenüberblick

---

## Organisatorisches

---

- ❑ Website
  - ❑ Vorlesungsumfang und Arbeitsaufwand
  - ❑ Tutoriumskonzept und Tutoriums anmeldung

# Webseiten zur OOSE-Vorlesung

Unter <http://oose.cs.uni-bonn.de/> finden Sie:

- Vorlesungsfolien
- Übungsaufgaben
- Literatur
- Hinweise zu Java, Eclipse, SVN, ...
- Termine
- Mitteilungen
- Prüfungsergebnisse
- ...

Alternativ: <http://sewiki.iai.uni-bonn.de/teaching/lectures/oose/2010/>

# „Stundenplan“

- Vorlesung (in Hörsaal 1)
  - ◆ 2 SWS
  - ◆ 11.15 -12.45 Uhr
- Tutorien (im Terminalpool A106)
  - ◆ 3 SWS
  - ◆ Praxis steht im Vordergrund!
- Zeitaufwand: 3 bis 5 Stunden pro Woche
  - ◆ Vorlesungsnachbereitung, Aufgabebearbeitung, Tutorien

# Tutorien

- Kleinstgruppen
  - ◆ 3 TeilnehmerInnen pro Gruppe
- Intensivbetreuung
  - ◆ 1 Tutor pro Gruppe, 45 Minuten exklusiv für 3 TeilnehmerInnen
- Praxisorientierung
  - ◆ Reichlich Programmierübungen
  - ◆ Elektronische Abgabe über Versionsverwaltungssystem „Subversion“
  - ◆ Vorführung und Besprechung mit Tutor direkt am Rechner

# Grundsätzliches zu den Tutorien

- Programmierübungen stehen im Vordergrund
  - ◆ Sie sind für das Verständnis der Vorlesungsinhalte unerlässlich
  - ◆ Programmieren lernen ohne zu programmieren ist nicht möglich!
- Gruppenarbeit
  - ◆ Sie dürfen und sollen die Aufgaben gemeinsam lösen
  - ◆ Es wird erwartet, dass Sie jede der Aufgaben präsentieren können, egal wer sie bearbeitet hat.
- Präsentation der Ergebnisse ist genauso wichtig wie ihre Erarbeitung
  - ◆ Die Tutoren sind angehalten, jedes Gruppenmitglied über jede der Aufgaben zu befragen.
  - ◆ Punkte gibt es nur für die Aufgaben, die Sie auch selbst zufriedenstellend präsentieren können, bzw. zu denen Sie schlüssig Fragen beantworten können.

# Organisatorisches zu den Tutorien

- Online-Anmeldung

- ◆ Angeben wann Sie **nicht** können
- ◆ Automatische Zuordnung der möglichen Gruppen
- ◆ Bei Überbelegung wird gewürfelt
  - ⇒ Je weniger „unmögliche“ Gruppen Sie angeben, um so höher die Chance eine der verbleibenden zu bekommen
- ◆ Zeitraum
  - ⇒ ab Mittwoch, 14.4.2010, 13 Uhr bis Donnerstag, 15.4.2010, 20 Uhr

- Link unter

- ◆ <https://sewiki.iai.uni-bonn.de/teaching/lectures/oose/2010/tutorien>
- ◆ ab heute Nachmittag zugänglich



# Übungsblatt 0

- Es ist jetzt schon online
- Sie sollten sofort anfangen es zu bearbeiten!

## Wichtige Inhalte

- Registrierungen!
- Eclipse!
- Subversion!

# Prinzip der Tutorien

- Wir nehmen Sie an der Hand, erwarten aber, dass Sie mitkommen!
  - ◆ Beim ersten Mal erklären wir alles noch ganz genau
  - ◆ ... danach erklären wir es gar nicht mehr!
  - ◆ Nehmen Sie also die Übungen ernst und machen Sie auch scheinbar „Einfaches“, es hat seinen Sinn, auch wenn Sie ihn nicht sofort erkennen
    - ⇒ Insbesondere: „Übungsblatt 0“ soll sicherstellen, dass für alle Beteiligten ihre Infrastruktur (Rechnerzugang, Tutorienanmeldung, SVN-Zugang, ...) reibungslos funktioniert,
    - ⇒ Wer Probleme hat, soll sie sofort melden, dann helfen wir Ihnen!
    - ⇒ Spätere Meldungen gelten als Ausrede für nicht erbrachte Übungsleistungen!
- Regelmäßige aktive Übungsbearbeitung ist die beste Vorbereitung auf die Klausur und ihr späteres Berufsleben!

# Grundsätzliches zum Lernen

## ● Chinesisches Sprichwort

◆ Was ich höre, vergesse ich.

◆ Was ich sehe, erinnere ich.

⇒ *Sehen + Hören* fördert die Erinnerung → Besuch der Vorlesung!

◆ Worüber ich spreche, verstehe ich.

⇒ *Selbst durchdenken und Formulieren* fördert das Verständnis  
→ Mitarbeit in der Vorlesung!

◆ Was ich tue, verinnerliche ich.

⇒ *Tun* ist der Schlüssel zu nachhaltigem Verständnis  
→ Aktives Bearbeiten der Übungen und aktive Tutoriumsteilnahme!

## ● Moderne Kognitionspsychologie

◆ ... bestätigt obiges in jeder Hinsicht!

# Rechnerzugang für Nebenfächler

- Für einen Account in der Informatik benötigen Sie eine Bestätigung, dass Sie an einer Übungsgruppe teilnehmen
  - ◆ **Sie können diese nächste Woche von Ihrem Tutor / ihrer Tutorin in Ihrer Übungsgruppe erhalten**

# Vorlesung „Objektorientierte Softwareentwicklung“

Kapitel 0: Organisatorisches und grober Themenüberblick

---

## Vorlesungs-Inhalte

---

Grober Themen-Überblick

Literatur

# Ziele der Vorlesung

## ● Inhaltliche Ziele

- ◆ Ihnen die Konzepte moderner, weitverbreiteter und extrem praxisrelevanter Programmiersprachen zu vermitteln
- ◆ Ihnen den Zusammenhang zwischen Programmiersprachenkonzepten und Design / Modellierung zu vermitteln
- ◆ Ihnen eine erste Idee davon zu vermitteln, dass Informatik nicht nur technische sondern auch soziale Kompetenzen erfordert (Teamarbeit, Präsentation, ...)

## ● Strategische Ziele

- ◆ Ihnen behilflich zu sein, sich klar zu werden, ob dies das richtige Studium für Sie ist
- ◆ Nach dem 2-ten Semester ist ein Drittel der Studienzeit um!  
Es wird Zeit sich zu entscheiden!
- ◆ Wenn Sie sich am Semesterende nicht sicher sind, dass Sie diese oder ähnliche Inhalte für die nächsten 40 Jahre (!) interessieren könnten, sollten sie ein Gespräch mit einem Dozenten, Studienberater, etc. suchen!

# Grobe Struktur der Vorlesung

- Imperative Anteile objektorientierter Sprachen
  - ◆ Von Pascal zu „Java ohne Objekten“
- Objektorientierte Sprachkonzepte
  - ◆ Ziemlich viele, auf den ersten Blick oft stark miteinander verwoben.
  - ◆ Wir versuchen sie möglichst entzerrt zu präsentieren und ihren Nutzen für Sie klar zu machen.
- Objektorientierte Modellierungskonzepte
  - ◆ Abstraktionen der Sprachkonzepte und ihre graphische Notation in der „Unified Modelling Language (UML)“
  - ◆ Elementarer Schritt um die Konzepte über Sprachgrenzen hinweg zu verstehen und anzuwenden
- Objektorientierte Bibliotheken und „Frameworks“
  - ◆ ... sind ein wesentlicher Teil der Stärke des OOP-Ansatzes
  - ◆ ... werden am Beispiel von graphischen Benutzerschnittstellen verdeutlicht

# OOSE im Kontext

- 1. Semester: „**Imperative Programmierung**“
  - ◆ OOSE baut darauf auf
  - ◆ OOSE legt die Grundlage für Bewältigung der nächsten Komplexitätsstufe
  
- 3. Semester: „**Softwaretechnologie (SWT)**“
  - ◆ SWT baut auf OOSE auf
  - ◆ SWT vertieft OO Design und OO Modellierung
  - ◆ SWT behandelt kompletten Softwarelebenszyklus
    - ⇒ von der Anforderungserhebung
    - ⇒ über den System- und Objektentwurf-Entwurf
    - ⇒ ...
    - ⇒ bis zur Qualitätssicherung durch systematisches Testen