

# **PS Software Engineering**

## **WS 2012/13**

Andreas Naderlinger

# Allgemeines

- Wöchentlich, Dienstag 08:30 - 10:00  
– Gruppe Naderlinger: T03
- **08:30 s.t.**
- Termine: PLUSonline
- Homepage zum PS:

**[www.softwareresearch.net](http://www.softwareresearch.net)** → Teaching

# Allgemeines (2)

- Anwesenheitspflicht !
- Keine wöchentlichen Tests, keine Abschlussprüfung
- Wöchentliche Aufgaben
  - Selbständiges Erarbeiten von Hintergrundwissen
  - Praktische Programmier-Übungen: **Java**  
**...knapp 2 Wochen Zeit**  
**...jeder einzeln**
- Insgesamt ~ 12 Aufgaben
  - Bachelor (4 ECTS)
  - Lehramt (3 ECTS)

# Beurteilung (bei 12 Aufgaben)

- Bachelor (4 ECTS)

– <=6:	5
7:	4
8:	3
9,10:	2
11,12:	1

... als Grundlage  
sowie

- Qualität der Abgaben
- Präsentation/Theoretische Fragen
- Mitarbeit

# Programmieraufgaben

- **Java** 6 bzw. 7 (nur standard library)
  - Also kein SavitchIn etc.

- Abgabe jeweils vor dem PS per E-Mail:

andreas.naderlinger@cs.uni-salzburg.at

- Subject: **PS SE Assignment 1**
- .zip-File with directory ,your\_last\_name‘ containing
  - ☒ .java-Files (no .class-Files, no Binaries)
  - ☒ Readme.txt (Name, Mat.Nr., how to run it, design decisions?, ...)

# “Programmieren im Großen”

- Die Entwicklung von kleinen Programmen unterscheidet sich fundamental von der Entwicklung großer Programme.
- Große Programme zu entwickeln heißt nicht „nur“ mehr Code schreiben zu müssen.
- Somit nicht bloß durch mehr Manpower zu beherrschen!

# Unterschiedliche Aufgaben

- Kleine *Programme*
  - “Coden” und “Ausprobieren”
  - “... habe ich schnell mal programmiert”
- Mittlere bis (sehr) große *Softwaresysteme*
  - Analyse
  - Spezifikation
  - Entwurf
  - Implementierung
  - Validierung
  - Wartung
  - **Dokumentation**
  - **Qualitätssicherung**
  - **Projektmanagement**

# Topics covered in WS2011...

- OO Design (Inheritance, Interfaces, Polymorphism, ...)
- Design Patterns (Composite, Iterator, Decorator, Visitor, Observer, MVC)
- UML
- Best practices (Exception handling, ...)
- TestDrivenDevelopment: JUnit
- Threads / Synchronization (System- vs. Application Programmer)
- Java Collection Framework
- Java Streams
- Generics
- Java Swing
- XML
- Reflection
- Ant, Versioning (Subversion:SVN), Jar