



***Software
Technology
Group***

TU Darmstadt | FB Informatik

Software Engineering Design

1. Getting Started

Prof. Dr. Mira Mezini

Dipl.-Inform. Christoph Bockisch

Dipl.-Ing. Michael Haupt

Design Needs Principles

A quote from the early days of object-orientation:

"OOP is like teenage sex."

Everyone talks about it all of the time
(without really knowing what they're talking about);
everyone claims to be doing it;
everyone thinks everyone else is doing it;

those few who are actually doing it:

- are not practicing it safely;
- are doing it poorly, and
- are sure it'll be better next time.

Principles are required in addition to mechanisms.

Design Needs Principles

- software is like mathematical theorems
 - good notation, adequate environments and powerful tools help
 - *human creativity* remains the main influence



- help is provided through...
 - object-oriented design patterns
 - object-oriented frameworks
 - other advanced design techniques
- the problem with "good style":
 - we can only *recommend* it



Organisation

- time and coordinates
 - Mondays, 13:30 - 16:05, room S2/02/C205
 - Thursdays, 11:40 - 14:15, room S2/02/C205
 - exam (written): Thursday, Feb 16, 2006, room ... will follow
- integrated course:
 - lectures where we present topics to you
 - talk where you present topics to the course
 - essays where you can deepen your knowledge
 - labs where we work and discuss together
 - homework you solve and present to the course
- you can gain bonus points
 - up to 16% of the exam

Homework

- homework
 - 4 exercises in total
 - at least six days for solution
 - you may work in groups (up to 3)
 - for solution deliver
 - code and documentation (preferably slides)
 - a day before lecture where exercise is discussed
 - we will review each solution and will either accept or reject it
 - you receive 3 %-points for an accepted solution
 - you may have to present your solution
 - **topics needed for exam**
- eAssignment
 - we publish assignments
 - you submit solutions
 - eclipse plug-in
 - will provide as bundle with eclipse installation
 - more details together with first homework

Essays / Talks

- essays
 - one subject each Thursday (last subject Feb. 2)
 - each subject can be summarized by several students
 - you get a paper or a chapter from a book which you will summarize
 - approx. 1000 words
 - one week for finishing essay
 - if your essay is not elaborated carefully enough we can reject it
 - if the essay is accepted, you get 4 bonus %-points
 - accepted essays will be published
 - it is not necessary to read essays for the exam
- talks
 - two students should give a talk instead of write an essay
 - about 20 minutes + discussion
 - also get 4 %-points for giving talk
 - dates
 - subject on Jan 9 -> talk on Jan 16
 - subject on Jan 16 -> talk on Jan 23
 - discuss the slides with us Friday before the talk
 - **subject are relevant for exam**

Labs

- will have labs integrated in lecture
- please bring your notebook or arrange with someone
- we will announce labs in lecture before the lab
- when lab is announced
 - please visit homepage the day before next lecture
 - download and install material provided there
 - please leave empty one row when seating
- **topics are relevant for exam**

Exam

- exam is Feb. 16 during lecture hours
- no requirements to attend it
- for bachelor / master
 - you must have registered at the "Prüfungssekretariat"
- for diploma students
 - the grade will be used in your diploma exam

Communication

- information on the lecture, materials and assignments can be found on the website:
<http://www.st.informatik.tu-darmstadt.de/static/pages/lectures/se/ws05-06/design/index.html>
- if you have questions / problems, you can contact us
 - Michael Haupt: haupt@informatik.tu-darmstadt.de
 - Christoph Bockisch: bockisch@informatik.tu-darmstadt.de
- or discuss them on the mailing list
 - for subscription, send an email to se-design-05-request@jsb.st.informatik.tu-darmstadt.de with the body
subscribe <your email address>
 - for posting a mail, send an email to se-design-05@jsb.st.informatik.tu-darmstadt.de
 - each member of the course can read and answer emails
 - we will moderate the list and answer mails when necessary
- important announcements will be made on the website and posted on the mailing list

Topics Covered

- history of programming concepts unto OOP
- modularity and principles of good design
- design patterns
 - idioms, design patterns, architectural patterns
 - some specific design patterns covered in depth
- refactoring: improving existing code and designs
- reuse of design and code: frameworks
 - case study: GEF (Graph Editing Framework)
- aspect-oriented programming
 - increased support for modularity at language level
 - AspectJ, CaesarJ
- programming languages vs. design patterns

Books

- Budd: Introduction to Object-Oriented Programming
- Meyer: Object-Oriented Software Construction
- Gamma et al.: Design Patterns. Elements of Reusable Object-Oriented Software
- Fowler: Refactoring
- Fayad et al.: Building Application Frameworks
- Laddad: AspectJ in Action