Organization
Teaser
As long as there were no machines, programming was no problem at all; when we had a few weak computers, programming became a mild problem, and now we have gigantic computers, programming has become a gigantic problem.

E.W. Dijkstra

The Humble Programmer

The Team
The Team

Dr. Michael Eichberg  Ralf Mitschke  Johannes Lerch

Contact

WWW

Forum
http://www.d120.de/forum/
→ kanonische Einführungsveranstaltungen
→ Einführung in Software Engineering
The Team

Dr. Michael Eichberg
Office hour:
Thursdays, 17:00-18:00 hours (S2 | 02 A206)

Ralf Mitschke
Office hour:
Tuesdays, 15:00-16:00 hours (S2 | 02 A207)
Student Assistants
Organization
Lecture

- Thursdays 13:30-15:00 in S2 02 | C205 and C110
- The slides are in English
  (Key terms will be translated into German.)
- The slides will generally be available after the lecture
  (I will try hard to make a preliminary version available the day before the lecture.)
- The slides and exercises can be downloaded from:
  [https://cage.st.informatik.tu-darmstadt.de/eise/public](https://cage.st.informatik.tu-darmstadt.de/eise/public)
  (SVN)
  (RBG account required)
Exercises

• Every week, we will have an exercise, starting Thursday, Oct. 27th, ~15:00-15:45 in S2 02 | C205 and C110
• Team exercises (not-graded) are expected to be solved in teams of 3 to 4 students
• Graded exercises need to be solved on your own
• Register for the exercise using Webreg; sign-up for the exercise until Oct. 26th
  https://www.dekanat.informatik.tu-darmstadt.de/webreg/index.php
  WebReg is only used to sign-up for the exercise.
• Sign-up as a team; if you don’t have a team, we will assign you to a team
Written Exam

• The exam will be on February, 22nd 2012 - 10:00 - 12:00
  (The room will be announced in due time. The exam will take 90min.)

• You need to register for the exam in TUCaN
  (There are no further prerequisites; “everyone” can attend the exam.)

• The exam will be an open book exam

• The exam will enable you to choose the tasks that you are particular good at
  (I.e., it will not be possible to solve the entire exam in 90 minutes.)
“Additional Information”

- The document stored in the SVN named “Weitere Informationen” contains information relevant for the written exam (e.g., the bonus system) and the lecture (All statements in the document are binding.)
Related Bibliography

- Design Patterns: Elements of Reusable Object-Oriented Software
  - Erich Gamma
  - Richard Helm
  - Ralph Johnson
  - John Vlissides
  - Martin Fowler
- Writing Effective Use Cases
  - Alistair Cockburn
- Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process
  - Craig Larman
- 97 Things Every Programmer Should Know
  - Edited by Kent Be Nelson
- The Passionate Programmer: Creating a Joyful Career in Software Development
  - Craig Fowler
Essential Bibliography

• Design Patterns - Elements of Reusable Object-Oriented Software; Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides; Addison-Wesley, 1995

• Applying UML and Patterns - An Introduction to Object-oriented Analysis and Design; Craig Larman; Prentice Hall
A Recommended / Very Useful Podcast

http://www.se-radio.net/
External Talks / Events

• **Capgemini**
  Excursion / One-day Workshop
  December, 1st, 2011

• **ACCSO** - Working Title:
  “Beschleunigte Softwareentwicklung”
  January, 19th, 2012
The Lecture
Basic programming skills are required.

- Basic knowledge of **object-oriented programming** concepts is necessary. I.e., you should readily understand the following terms:
  - class, interface
  - object
  - inheritance
  - polymorphism
- Working knowledge of the **Java** programming language
• To get a brief overview of “all” areas of software engineering
• To understand agile software development processes
• To be able to perform object-oriented analysis and design
• To be able to read and create basic UML diagrams
• To get first hands-on experience and to learn to use basic software development tools
The goal is to enable you to systematically carry out small(er) commercial or open-source software projects.
Introduction to Software Engineering

- What is Software Engineering (A)
- Software Project Management
- Software Engineering Tools
- Object-Oriented Analysis and Design
- UML Modeling
Introduction to Software Engineering

Object-Oriented Analysis and Design

UML Modeling

- General Design Goals
- Responsibility Driven Design / GRASPPrinciples
- Design Heuristics
- Domain Modeling
- Design Patterns

- Low Coupling
- Singleton
- Template Method
- Class Diagrams
- Sequence Diagrams
- Communication Diagrams

- High Cohesion
- "What is Software Engineering?"
Content Distribution

"General" Software Engineering Topics:
- Software Quality
- Requirements Engineering
- Software Engineering Processes
- ...

- General Design Principles
- Object-Oriented Design Heuristics
- Object-Oriented Design Patterns
Developing Competence

Information

Knowledge

Ability

Attitude

Competence

Professionalism

+ cross-linking

+ application

+ assessment (e.g., the choice of a specific design/process)

+ suitability / adequacy

+ responsibility