Summary
Today

- Following slides have keywords of roughly each lecture
- This is mainly about *students* asking questions, we do not repeat the whole semester in half an hour.
Object-Orientation Basics

- class, instance
- interface (type)
- method, data member
- inheritance, overriding
- polymorphism, late binding, casting
- reuse: composition vs. inheritance
- meta-class
Development Process

- Process models:
  - waterfall
  - spiral
  - evolutionary
  - Unified Process (UP), incremental and iterative

- Details of UP:
  - inception
  - elaboration
  - construction
  - transition

- Unified Modeling language (UML)
Analysis Phase

- Use cases
  - identifying use cases
  - describing use cases

- UML: Use case diagrams
  - use case
  - actor
  - inclusion, extension, generalization
Design Phase

- **Formal Specification**
  - Introduction to the concept of Alloy

- **Specifying structure and behavior**
  - approaches to each: class diagrams, state machine, Petri nets
  - scenario-based and declarative ones

- **UML:**
  - structure: (conceptual) class diagrams
    - classes
    - associations
    - ...
  - behavior: sequence, activity, state, ... diagrams
Implementation Phase

- Implementation Level UML diagrams
  - difference to / connection with other diagrams
  - how to derive?
Tools Overview

- Which software is a tool? (and which is not?)
- Importance of tools
- Software development methodologies and their relationship with tools
- Use (when and for what) of tools:
  - project management
  - documentation
  - model-centric
  - repository
  - code-centric
  - testing
- Runtime tools
Model-centric Tools

- Kinds:
  - diagrammatic
  - requirements management

- Approaches in requirements analysis

- Model-code synchronization and its relationship to software development methodologies
  - code generation
  - MDA

- Project management tools
  - Activity organization
  - Gantt and Pert charts
OO-Libraries and Extensions

- Reuse of code
  - source code
  - modularization
  - parameterization
  - generation

- Different kinds of extensions:
  - library/toolkit
  - framework
  - component
  - code generator

- Writing extensions
Version and Change Tracking Tools

- Central aspects:
  - propagating changes
  - merging changes
  - development history

- Versions and revisions
  - revision/version graph
  - extend, merge, split

- Conflicts and their resolution
Integrated Development Environments

- Development of complex application requires tools support
- Central functionality:
  - code editing, compilation, running, debugging
  - often: refactorings, documentation support, ...
  - extensibility by plugins
- Source code is not just a one-dimensional string of characters!
Documentation Tools

- Importance of documentation
- Kinds of documentation
  - for developers: on source code
  - for API-users: API docs
  - (for end-users: manuals)
Exam: Formal Stuff

- Preliminary dates:
  - OOAD: 19 Feb 2008
  - LTOOD: 27 Feb 2008

- Watch the Prüfungsamt’s homepage for updates, times, and rooms!