
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

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

Development Process

- o Process models:
 - n waterfall
 - n spiral
 - n evolutionary
 - n Unified Process (UP), incremental and iterative
- o Details of UP:
 - n inception
 - n elaboration
 - n construction
 - n transition
- o Unified Modeling language (UML)

Analysis Phase

- o Use cases
 - n identifying use cases
 - n describing use cases
- o UML: Use case diagrams
 - n use case
 - n actor
 - n inclusion, extension, generalization

Design Phase

- o Formal Specification
 - n Introduction to the concept of Alloy
- o Specifying structure and behavior
 - n approaches to each: class diagrams, state machine, Petri nets
 - n scenario-based and declarative ones
- o UML:
 - n structure: (conceptual) class diagrams
 - o classes
 - o associations
 - o ...
 - n behavior: sequence, activity, state, ... diagrams

Implementation Phase

- o Implementation Level UML diagrams
 - n difference to / connection with other diagrams
 - n how to derive?

Tools Overview

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

Model-centric Tools

- o Kinds:
 - n diagrammatic
 - n requirements management
- o Approaches in requirements analysis
- o Model-code synchronization and its relationship to software development methodologies
 - n code generation
 - n MDA
- o Project management tools
 - n Activity organization
 - n Gantt and Pert charts

OO-Libraries and Extensions

- o Reuse of code
 - n source code
 - n modularization
 - n parameterization
 - n generation
- o Different kinds of extensions:
 - n library/toolkit
 - n framework
 - n component
 - n code generator
- o Writing extensions

Tools

- o Central aspects:
 - n propagating changes
 - n merging changes
 - n development history
- o Versions and revisions
 - n revision/version graph
 - n extend, merge, split
- o Conflicts and their resolution

Integrated Development Environments

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

Documentation Tools

- o Importance of documentation
- o Kinds of documentation
 - n for developers: on source code
 - n for API-users: API docs
 - n (for end-users: manuals)

Exam: Formal Stuff

- o Preliminary dates:
 - n OOAD: 19 Feb 2008
 - n LTOOD: 27 Feb 2008
- o Watch the Prüfungsamt's homepage for updates, times, and rooms!