Course PM: OO programming project, TDA367/DIT212, LP4 2016

Course responsible, examiner and lecturer: Joachim von Hacht, hajo@chalmers.se
Assistants: See course site
Course site: http://www.cse.chalmers.se/edu/course/TDA367

General

This course is intended to make you familiar with the principal activities of software construction: requirements elicitation, analysis, design, implementation, testing, and documentation. The idea is to practice what you have learned in the courses TDA545/DIT012 and TDA550/DIT952. On top of that, you will gain experience in specifying, designing and implementing a system from scratch, in a (small) team.

Course uses English for most written documents (all code in English). Spoken language is Swedish (sadly with a lot of svenglish computer terms).

Collaboration with LSP310

This applies only to students taking the course “Kommunikation och ingenjörskompetens”, LSP310. NOTE: The oral presentation is part of the grading for both courses. See examination below.

Schedule

See TimeEdit (link from course page > About).

Literature

Recommended;

- Clean Code, Prentice Hall, Robert C. Martin
- Domain driven design, Addison-Wesley, Eric Evans
- Jan Skansholm, Java Direkt med Swing, latest edition, Studentlitteratur
- Joshua Bloch, Effective Java latest edition, Addison-Wesley

Environment

IDE
We recommend the NetBeans IDE (but any IDE will do).

Version control system
Git is mandatory. We recommend using the command line Git.

Build automation software
Mandatory is Maven or Gradle

Organization

Project

Each group is supposed to deliver a project. For project details see Project PM on course site > Project.

Groups

The project is carried out in groups of four students. We will only allow group size different from four when this is necessary because the total number of students is not congruent to zero modulo four.

Because of collaboration with LSP310 (scheduling) we prefer pure Chalmers and GU groups. If any questions contact course responsible.

Lectures

There are a few introductory lectures, see course site. The lectures will mostly be a real time demonstration of the software process you are supposed to use.

Workshops

During the first weeks there are two workshops to help you (your group) to quickly get going. The workshops are designed to be self explanatory, should be possible to do anytime/anywhere. If you need help visit the workshop session. For schedule see course site > About.

The Git-workshop is mandatory and should be presented during the workshop session or possible (if not finished) during the supervised meeting, see below.

Seminar

There is an obligatory seminar, see schedule. The group has about 10 min. to present the project and the domain model. Be well prepared, tight schedule!

Supervised meetings

Each week you will have a meeting with your supervisor. The meetings are compulsory! If you miss a meeting you will have to do a complementary assignment (code or write a paper)! This is also applicable if you have too many late arrivals.

Group meetings

The group is assumed to organize two group meeting each week. The meetings should be documented (upload to git-repo, use agenda template on course site).

Examination

Basic requirements for the group to pass;

- Have demonstrated the Git-workshop.
- Have handled in and got the project accepted.
- Each week have organized two meetings for the group.
- Have done an oral presentation of the project.
- Have attended some of the other groups presentations and acted as opponents on one of them (at least one day presence).

For individual member of groups to pass;

- Have fulfilled the group requirements.
- Have visited all mandatory meetings (or handed in the complementary assignment).
- Have been an active member of the group.
- Have participated sufficiently to the final deliveries.
- Have spoken at the oral presentation.
- Have asked some questions at the presentation (as an opponent).
- Have delivered a self estimation, see course page.

It will be possible to trace individual code contributions. If you work in pairs (or similar) you must take turns when committing into the version handling system. Else your contribution will be lost!

Grades

Individual (U/3/4/5, U/G/VG). Grading will be done in two phases;

1. Group grade; Depending on the project and the oral presentation a group grade will be assigned.
2. Individual grade: We'll estimate each students contribution to the project and presentation. High or low contributing students could get higher or lower grades but normally the group and individual grade will end up the same. The mean of the individual grades should be close to the project grade.