

Object-Oriented Programming with Java - Advanced Course

Project Description

Robin Müller-Bady

Winter 2014/2015

1 Introduction

The examination of this module is in form of a group project. The resulting program has to be fully functional and documented. Each team should come together and work autonomously and is responsible for distributing tasks fair and equal among all team members. The task must be solved by the team. Help of external people or any fraudulent activities, e.g. copying source code from the Internet except external libraries, will result in, at least, disqualification or legal consequences. Any suspicious activity of that kind will be reported to the examination board for further investigation. All conditions of the examination regulation and module handbook apply.

The following constraints apply to the module examination:

Start Date 1st January 2015, 00:00 AM

Submission Date 12th February 2015 @ “Studiengangssekretariat” (Build. 1 - 215)

Team size 2 to 4 team members (announced in advance)

Topic The presented project from Section 3

Submission Details The following things are required for a successful submission

1. A working program as runnable *.jar file including all necessary libraries and the source code (be careful with the discussed jar-in-jar problem) and the (fully functional) zipped project (eclipse or netbeans) including all necessary files e.g. project files for importing. The source code must be commented as mentioned in the clean code lecture. Make sure that your program runs at least under Linux with JDK 7. Note that the machines in the lab have an installed JDK 6 version.

Please notice that a dysfunctional program will be marked 0 points!

2. Documentation as separate document (PDF file) including technical details of the program, user documentation and presentation of the individual team member performance. A weekly milestone document has to be submitted.
3. The GUI must be developed “by hand”. GUI builders, e.g. Netbeans’ Swing GUI Builder (formerly project Matisse), are not allowed.
4. Submit the software (e.g. as CD, DVD, ...) and printed documents at the study secretary (“Studiengangssekretariat”) in building 1, room 215. The disk must also contain a copy of the documentation as PDF.
5. The declaration of authorship (German: “Ehrenwörtliche Erklärung”) must be signed by each team member and added to the submitted documentation. You find an example text below.

Presentation After the submission each individual (person) has to present the team solution and the participation to it. The presentation should take around 15 to 20 minutes per individual.

Hints It is also possible to implement only parts of the given requirements. In case you do that, please make sure that your program is running even in that case. If parts of the specified program are missing, points will be deducted.

It is necessary to successfully submit every required item by the closing date. In addition, a successful participation in the presentation is necessary.

Declaration of Authorship

Example for a declaration of authorship (“Ehrenwörtliche Erklärung”):

I hereby declare that the submitted project is my own unaided work or the unaided work of our team. All direct or indirect sources used are acknowledged as references.

I am aware that the project in digital form can be examined for the use of unauthorized aid and in order to determine whether the project as a whole or parts incorporated in it may be deemed as plagiarism. For the comparison of my work with existing sources I agree that it shall be entered in a database where it shall also remain after examination, to enable comparison with future projects submitted. Further rights of reproduction and usage, however, are not granted here.

This work was not previously presented to another examination board and has not been published.

First and last name

City, date and signature

2 Grading Criteria

The grading of the projects is based on the given lectures and the individual autonomous learning of the topic. The module is passed with a total of 50 points (50%). See Table 1 for details of the grading structure. For a detailed view of the necessary requirements see Table 2.

Table 1: Points vs. Grades

Grade	5,0	4,0	3,7	3,3	3,0	2,7	2,3	2,0	1,7	1,3	1,0
Points	<50	50	55	60	65	70	75	80	85	90	95 ⁺

Table 2: Requirements

Feature	Points	Comment
Runnable Program	30	The running program is covering the described basic functionalities as described in Section 3.
Milestones	20	Milestones are submitted as required. The requirements of the milestones are specified in Section 3.
Object-Orientation	5	The program was developed using an adequate object orientation. At least one abstract method and one interface must be developed and used.
Collections	5	Information will be stored in suitable Java Collections.
Error Handling	5	The program contains adequate error handling. At least one own exception (extending any Exception) must be programmed.
Streams and Files	5	The program works with streams and files. At least one reading and writing file access should be made.
Threads	5	The program contains at least two threads (from which one is “main”).
Clean Code	20	The program is refactored. Learned guidelines from the lectures are applied.
Documentation	5	The documentation inside and outside the code is present as described in Section 3.
Additional Features	5	The program can contain additional features to gather some extra points. Additional features are to be discussed with the lecturer in advance in order to be counted.
Total	105	

3 Project - Graphical Email Client

The task is to develop an email client for sending and receiving emails from a given email server. Besides a logical program, a graphical user interface has to be developed to offer users a comfortable way of interacting with the program.

The basic functionalities are as follows:

- The graphical user interface shows all emails that exist in the “Inbox” and the “Sent” folder in an overview in tabular form. At least sender, receiver (only “to” field), subject and date are shown.
- It is possible to show the details of an email in a new window. At least the sender, (all) receivers, subject, date and email content are shown.
- The user has the possibility to write and send a new email to given receivers. It is possible to write an email with receivers in “TO”, “CC” or “BCC”.
- It must be also be possible to answer a received email. The String “Re:” should be appended in front of the subject of the mail that will be answered. The content of the answered mail should be copied and preceded by an indicator that indicates, that this is cited text.
- There should exist an “update” button to refresh the emails from the server in your table view.
- Before sending emails “sanity checks” should be made, e.g. exclude false email addresses.
- Configuration properties are read from a file that exists inside the Java project. Only one protocol for receiving emails has to be covered (POP3 or IMAP)
- SMTP should be used for sending mails
- Only one email provider has to be covered (e.g. gmail)

The requirements for the documentation are

- JavaDoc is applied for each public method inside a class except getter and setter methods.
- Other source code documentation/comments are applied as necessary.
- Documentation outside the code is submitted as PDF file containing user- and technical documentation having adequate descriptions, graphics, diagrams etc. Following elements must be present:
 - User handbook
 - Technical description
 - UML diagram(s)
 - Work distribution among team members preferably as table
 - Milestone documents
- Milestone documents containing the current state of the project and the participation of the team members (has to be submitted separately each week via email¹). Submitting the milestones is mandatory. The submission deadline the milestones for a given week is Sundays at 23:59 (timestamp of the email counts, consider to send it early).

It is recommended to use the JavaMail API in version 1.5.2. ² or other APIs/Frameworks that cover the necessary functionalities.

¹Email address: mueller-bady@fb2.fra-uas.de

²<https://Java.net/projects/Javamail>