



Practice sheet 3 – due on 2010-11-08

Learning Aims:

- Familiarization with the Open Source application JabRef
- Exploration of the source code of a big application
- Introduction to an Open Source project: Bugfixing
- Application of the learned skills via JUnit and TDD

Outlook: In the following practice sheet you are asked to extend JabRef which is an open source bibliography reference manager! Therefore, learn as much as possible about JabRef with the help of this practice sheet.

Task 3 – 1: Familiarization with the functions of JabRef

The aim of this task is to get to know the main functions of JabRef. Do the following:

1. Make sure you have installed an up-to-date version of Eclipse.
2. Carry out an SVN import of the project JabRef in Eclipse: Go via *File → New → Other → SVN → Checkout Projects from SVN*.
In case there is no SVN wizard, you have to install Subclipse¹.
3. Chose as location:
<https://jabref.svn.sourceforge.net/svnroot/jabref/trunk/jabref>
4. In the case that some error will occur you may manually categorize the reason for this as warning: *Project > Properties > Java Compiler > Errors/Warnings > Deprecated and restricted API > Forbidden Reference > Warning*
5. Execute the Ant-Target² »run« of the file build.xml. While doing so, further targets (e.g. for compiling) will be executed. If all goes well, the program will be started after some seconds³.
6. Now study the documentation⁴ of JabRef in order to complete the following tasks/functions:
 - a. Work with literature tables: meaning of colors, sort sequences ...
 - b. Add an entry to JabRef and change it
 - c. Groups: generated statically, dynamically and automatically
 - d. Linking of entries with files (e.g. PDF)
 - e. Carry out the integrated online search
 - f. Automatically generate BibTeX keys
 - g. Mechanisms for the central administration of frequently used words or phrases (e.g. for keywords, journals, ...)
 - h. Import/Export of external files

The aim of this task is to enable you to introduce your fellow students in the tutorial to the use and the features of JabRef.

¹ Subclipse for Eclipse >3.2 (Update site): http://subclipse.tigris.org/update_1.6.x

² Information on Ant: <http://de.wikipedia.org/wiki/Ant>; <http://ant.apache.org/>

³ In future, you can also start JabRef by calling the main class (see task 3-2b).

⁴ <http://jabref.sourceforge.net/documentation.php>



Practice sheet 3 – due on 2010-11-08

Task 3-2: (Introduction to the source code of JabRef)

The aim of this task is to introduce you to the code of JabRef and testing the use of TDD while further developing JabRef. Please do again work **in pairs**. Execute all programming **via TDD**.

[In the case you want to participate in the Saros experiment, providing 4 points, please solve this task in a distributed environment using Saros (for further information and setup steps see <http://www.saros-project.org>)]

1. For test purposes, execute all unit tests in your JabRef-Eclipse project. How do you best proceed? What is the result?
2. Which is the main class of JabRef (the one with which JabRef is started)? Describe roughly what happens at the start!
3. On <http://sourceforge.net/projects/jabref/develop> open the area *Tracker* → *Bugs* and chose one bug fulfilling the following criteria:
 - a. The error description is understandable and comprehensible.
 - b. You can reproduce the error in the development branch aka trunk (the version you already checked out).
 - c. It is no problem if comments with approaches already exist, as long as they do not completely anticipate the solution.
 - d. In case of questions or doubts, first write a blog entry in the "Softwaretechnik Blog" or send your questions to our mailing list `se_v_sw2!` Should you feel confident, you may very well add clarifying comments directly to the appropriate entry of the bug tracker.
4. Now have a closer look at the entry. Understand its error message (possibly have another look at the JabRef documentation). Go to the code of JabRef and do the following:
 - a. Look for the affected class(es)/method(s) in your Eclipse project.
 - b. Are there unit tests for this/these class(es)/method(s) already? Where?
 - c. **Remove the defect reported via TDD** (that is write the necessary test and change or extend the code). Record yourselves with Camtasia, should you be interested in your approach.
 - d. Even if you know how to proceed: Do **not** send your solution to the project. Your fellow students should also have the chance to solve the task.
 - e. Prepare to present your approach and solution in the tutorial. Pay attention to the following points:
 - Which functional areas of JabRef are concerned?
 - Explain the problem behind the error message in detail.
 - Which classes are concerned in JabRef?
 - Are there unit tests for these classes? Why do these not report any failure in this respect?
 - Describe the written tests.
 - Explain the written code.
 - Sketch the solution aspired.
 - f. Describe the experiences you have had with pair programming and TDD in this case. Where these methods helpful? Why so (give examples)?