Radboud University

Dr. Daniel Strüber

Radboud University
Institute for Computing
and Information Sciences

Autumn term 2024/25
Published: 02.09.2024
To be completed:
09.09.2024 13:30

Software Product Lines

Assignment 1

Task 1: Find a group

This is a practical course, and you will work on several assignments with a completion time of one or two weeks each. You're not on your own: the assignments are designed for teams of three persons (exceptions are possible but should not be the rule).

We will allocate time to form groups in today's course session. If you don't have a group by the end of the session, please use the forum to coordinate.

We recommend to take the following considerations into account:

- Do you prefer to work in in-person meetings or virtually?
- Ideally, you would want to work with group members of similar programming experience and skill level. For coordination, you can use skill levels *beginner* (beginner experience with programming), *experienced* (you have developed a few programs already), and *advanced* (you have done a lot of programming).

Once you have found your group, please register your group in Brightspace. The groups are set up for self-assignment, so just, register your group under a free group number.

Task 2: Download and install FeatureIDE

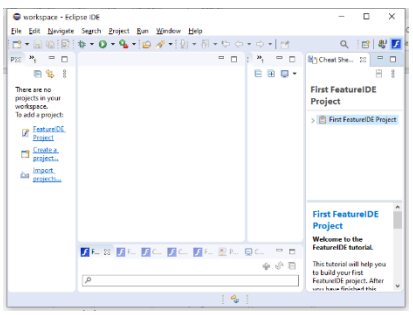
In this course, we will use FeatureIDE as the main tool for modeling and implementing software product lines. FeatureIDE comes as either an installable plug-in for the Eclipse IDE, or as a prepackaged bundle of Eclipse with all required plug-ins. We recommend to use the prepackaged bundle. Follow the instructions given below:

a) Download the suitable bundle from https://featureide.github.io -> Downloads
Choose the latest version (v.3.11.1) of the bundle "Eclipse with FeatureIDE, JDT, CDT, and AJDT".
This will lead to a download of an archive file of about 800 MB.

b) Unzip the bundle on your local hard-drive, for example (in Windows) under C:\spl\eclipse

c) At the first start, Eclipse asks you to specify a directory for the workspace to be used. I recommend to use the default, the directory " ../workspace" in the same folder as your Eclipse installation.

d) The started Eclipse should look as follows:



Task 3: Chat implementation (to be submitted until 09.09.)

Implement a *simple* chat application with Java. The program consists of multiple clients that can connect to the server and exchange messages over the server.

The chat application must have the following features.

1. **Color**: Each message can have a text color. In the client, it's possible to configure the text color for all outgoing messages.
2. **Authentication**: The client has to send a special message to the server, including a password, before it can send and receive messages.
3. **Two encryption algorithms**: All message exchanges are encrypted. To each message, two encryption algorithms are applied (the second one to the result of the first one).
4. **Log**: All clients and the server have a log, recording all received messages.

The features can be implemented in a minimal way (for example, encryption by ROT13 or reverting the message; color only shown as additional text; authentication with a fixed password; log writes to a file).

Tutorials and basic implementations of chats using sockets are available on the internet, for example, at https://www.ashishmyles.com/tutorials/tcpchat/index.html

You are allowed to use Generative AI tooling (e.g., ChatGPT or Copilot) to generate your solution or parts of it, as long as you can take responsibility for the resulting solution, for example, by explaining how it works.

Hint: We will use this example as a basis for upcoming assignments, in which we consider certain language extensions of Java. Since some of these extensions only support the core Java features, please avoid advanced Java concepts from newer Java versions; specifically, generics and lambdas (we know, might be awkward).

Please submit your solution to this task as a Zip file containing the Java code. If you're using Eclipse (which we recommend), use Eclipse's export functionality *(File -> Export -> Archive File).*