Radboud University

Dr. Daniel Strüber
Dr. Carlos Diego Damasceno

Radboud University
Institute for Computing
and Information Sciences

Autumn term 2022/23
Published: 03.10.2022
To be submitted:
10.10.2022 13:30

Software Product Lines

Assignment 5

Tasks 1-3 are designed as live exercises for the tutorial session – for this, we will split up in breakup groups. It's not required to submit anything for these tasks.

Tasks 4 is to be completed offline and to be submitted until the deadline.

Task 1: Feature orientation - terms

Clarify the meaning of and connection between the terms "feature", "collaboration", "role", "mixin", "jampack", "class" and "object". Illustrate the meaning of each term using the chat product line as an example. How does feature-orientation solve the problem of inflexible extension hierarchies?

Task 2: Role-based design of chat implementation

Define a role-based design for your chat implementation (as we did in the lecture for graphs). The individual features should be modelled by collaborations.

Include new and refined methods and fields into the roles.

Define the design as a class diagram, for example, using www.lucidchart.com, https://creately.com/diagram-type/class-diagram, Microsoft Visio, or PowerPoint.

Task 3: Mixin and jampack

Under which circumstances would you prefer mixin composition to jampack, and the other way around?

Task 4: Feature-oriented implementation

a) Evolve your chat-product line towards feature-oriented programming.*

Hints:

- I recommend using FeatureIDE with FeatureHouse, which on my system works without problems. See https://github.com/FeatureIDE/FeatureIDE/wiki/Tutorial Using AHEAD instead of FeatureHouse I encounter a bug, even on the simple program from the tutorial.
- Start with the base program and extend it with the features from the previous assignments.

* Modified version for smaller groups:
3-person groups may omit support for logging.
2-person groups may omit support for logging and authentication.

b) After finishing task (a), add a new feature: sound output. This feature plays a sound on every received and sent message in the client. A simple solution could just play beep sound, using `Toolkit.getDefaultToolkit().beep();`

c) Generate and manually test a few (at least three) variants.

Submit a zip archive with the source code of your implementation, and a test report PDF. The test report should include for each tested variant: a) the used feature selection, and b) a screenshot of the resulting system when executed.