Software Product Line Engineering

Summer 2019, Assignment 1

Bauhaus-Universität Weimar

Prof. Dr.-Ing. Norbert Siegmund
Stefan Mühlbauer, M. Sc.

Assignment issued: Friday, 12 April, 2019
Submission due: Thursday, 25 April, 2019, 23:59 CEST
Presentantion: Friday, 26 April, 2019

Task 1: Software Product Lines (8 points)

(a) Define the term software product line (SPL) in your own words. (1 point)

(b) When is a software a product line? Provide a brief description of common characteristics. (1 point)

(c) Could you name an example software that is not a product line? Why is that, or why is it not developed as such? (2 points)

(d) Name three domains or use cases where SPLs are useful. (1 point)

(e) Research the following three software systems and decide whether they are software product lines. Motivate your decisions! (3 points)

BusyBox (`www.busybox.net`), GIMP (`www.gimp.org`), Android (`www.android.com`)

Task 2: Chat Product Line (8 points)

Suppose you are the CTO of a software company which intends to develop a chat application. You have decided to simplify development using SPL concepts.

(a) Analyze the domain of chat applications. Which features might be desired by users? Which features might be desired only by some of them? For which of these features is there an economical pointet? Can you think of "killer features" with which you can stand out from other competitors? (3 point)

(b) Design a brief chat application on class-level using the Unified Modeling Language UML). The application consists of several clients and a single server which broadcasts messages.

The chat application should provide the following functionality (1 point per feature):

(i) Interfaces: The chat client should have a graphical user interface (GUI); additionally, it should be possible to use a simple command line interface (CLI).

(ii) Color: Every single message can have a different font color; each client can change the color for outgoing messages.

(iii) Authentification: Each client must send an authentification message (with password) to the server before being able to send and receive messages.

(iv) Encryption: All message traffic has to be encrypted.

(v) Logging: Both clients and the server maintain a log of received messages.

Task 3: Software Product Line Development (DE students only; 6 points)

(a) The development process for a software product line includes domain and application engineering. What are the outcomes of the last step of domain and application engineering, respectively? (2 points)

(b) What is the difference between *proactive* and *extractive* development of software product lines? (4 points)

Task 4: Object-Oriented Implementation (16 points)

There are several tutorials and example of simple chat applications available on the internet and can be used. Adddi-tionally, we provide an implementation stub to start with. Your implementation is the basis for later assignments and therefore crucial for the remaining course.

Implement a single chat application (one server, multiple clients) in Java. The application should have the following features (4 points each), which can have a simplistic realization.

a) Color (can be shown as text, e.g., `<blue>text</blue>`)

b) Authentification (e.g., hard-coded password in the source code)

c) Two encryption modes (encryption via ROT13 cipher and via exchanging of first two letters of a message)

d) Logging (e.g., use a simple log file)

Submit your answers (PDF) and implementation (jar + source code) as an archive with name, degree program, and matriculation number until 25 April, 23:59 CEST to `stefan.muehlbauer@uni-weimar.de`.