Task „Studienleistung“ (v1.0.0)

The task has to be solved individually and without the help of generative AI. If (very) similar solutions or AI-generated solutions are detected, all affected parties will fail the task.

Background

This „Studienleistung“ task is a comprehensive task that connects the individual pieces covered in our first four lectures. It requires basic programming and software engineering skills that you should have acquired as part of your bachelor program studies.

Task 1: Creating an application

Create a small financial application that helps the user to calculate various parameters around paying back a loan. The application shall be realized as a command line program that takes all arguments as parameters via standard streams (STDIN, STDOUT). Error handling is not required.

Input: Debt value (in euros), interest rate (percentage), payback duration (months)

Output: monthly rate, number of rates, overall debt (debt value plus total interest)

Task 2: Automation and Enhancements

Make building, testing and quality assessing your program automatable by adding a build script, unit tests (including mutation testing), metrics computation and identification of bad smells. Version all your artificacts using a version control system and create a continuous integration pipeline that runs all the aforementioned steps automatically each time a new commit to the code repository is issued.

Deliverables (via OLAT, Deadline to be announced)

- Source code
- Unit tests including mutation tests
- Build automation script
- Results of computed metrics
- Results of identified bad smells
- Demonstation video showing off all parts in action (by running a successful build and showing the artifacts and committing a breaking change afterwards)

Tool support

In general you have freedom of choice regarding tools and languages. You are also allowed to use any services (SaaS) of your choice.

The use of Java (as programming language), JUnit (for the Unit Tests), Pitest (for mutation tests), Spotbugs (for bad smell identification), Ant (for the build script), OpenSCC (for metrics), Jenkins (for CI), Git (for version control) and Forgejo (for Git hosting) is recommended.

Individual work *Please submit all work via OLAT (not by E-Mail!)*