Department Mathematik/Informatik, Abteilung Informatik
Software & System Engineering Group
Software Technology Group
Lecture Object-Oriented Software Engineering, SS 2024

Alexander Korn, Adrian Bajraktari,
Prof. Dr. Michael Felderer

UNIVERSITÄT ZU KÖLN

Project Sheet 3. Third Sprint

Hand out: 21.06.2024, 18:00
Hand in: 28.06.2024, 14:00

Exercise 1. This Sprint's Work

This week, your task is to continue working on the room and course management and to implement some basic institute and chair management.

a) Begin by merging the updated template from `https://github.com/lxndio/oose-24-project` into your project.

> **Hint:** If you have forked the template repository (the *upstream* repository) to create your own repository, this page from the GitHub documentation explains how you can update your repository: `https://docs.github.com/de/pull-requests/collaborating-with-pull-requests/working-with-forks/fork-a-repo`.

b) First, add the institute and chair management:

- Add models `OrganizationalUnit`, `Institute`, and `Chair`. The latter two should be derived from the organizational unit, which should not be useable itself. All of the models should have an `id`, and a `name`.
- Institutes and chairs should have a plausible relationship. An institute should store a string `providesStudySubject` specifying which study subject is offered by the institute.
- A chair should have an employee as the chair owner and be located in a building.
- Remember to implement getter and setter methods where necessary.
- Create (Spring) repositories for `Institute`, and `Chair`. Remember what you have already learned about derived query methods and implement query methods that you think are useful for the given repositories. You can add query methods whenever you need them.

c) Complete the following tasks to extend the room and course management functionality:

- Create repositories for `University`, `Building`, `Room`, and `Course`.
- Chairs and courses should have a plausible relationship. A chair of course may offer many courses but a course is always only offered by one chair at a time.

d) Finally, you should add controllers to allow showing rooms and courses in the web application:

- Create a course controller. The controller should get the course repository and provide an endpoint `/courses` which takes `sort_by` and `sort_asc` query parameters and implements them as the student controller does. Both query parameters should again be added to the model. Additionally, the sorted list of `courses` should be added to the model. The controller should return the `courses` view.
- Create a room controller. The controller should be implemented similarly to the course controller.

[6 points]

$\sum$ **6.0 points**