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 2. Second Sprint

Hand out: 14.06.2024, 18:00
Hand in: 21.06.2024, 14:00

Exercise 1. This Sprint's Work

This week, your task is to add room and course management to Klipsias.

a) Complete the following tasks to implement the room management functionality:

- Create the models `University`, `Building`, `Room`, and `RoomOccupancy`. Make them entities the way you have already learned in the last sprint. Make sure that all models have an ID. This is important for loading from database.
- Universities should have a `name`. Furthermore, they should have a list of `buildings`, `students`, and `employees.`

> **Hint:** Spring uses annotations to mark database relations between models. For example, you can model a one-to-many relation between `University` and `Student` by annotating a list of students in the university model with `@OneToMany(mappedBy = "university")`. Vice versa, you'd then add an attribute `university` to the student model with the following annotations:
> `@ManyToOne`
> `@JoinColumn(name = "university_id")`
> The join column annotation tells Spring which database column holds the reference to university. Always remember to add relations in both directions wherever necessary.

- Buildings should have a `name` and a list of `rooms`. Keep in mind that a room can only be in one building but a building might have multiple rooms.
- Rooms should have a `number` (which should be a string to allow for numbers such as ”1.234”, a number of `seats` and a marker `isAuditorium`. Also, they should have a list of `occupancies`.
- Use the `RoomOccupancy` model to model a many-to-many relationship between rooms and courses. Each room may be occupied by various different courses. Also, a course may use various different rooms. Thus, the model should reference an `course`, and a `room`. It should have a `occupancyTime` attribute of type `LocalDateTime`.

b) Now that the basic room management works, you should implement the course management. Complete the following tasks:

- Create the models `Course`, `Enrollment`, and `TeachingShift`.
- Courses should have a `name`, a list of `roomOccupancies`, a list of `enrollments`, and a list of `teachingShifts`.
- The `Enrollment` model should reference a `course`, and a `student`. Think of the required annotations to model a many-to-many relationship. Additionally, a `semester` should be stored as a string.
- Students can enroll in a course. Similarly, employees have teaching shifts. Thus, the teaching shift model should be developed similarly to the enrollment model but for employees instead of students.

[6 points]

$\sum$ **6.0 points**