Übungsblatt 6 Software Technik 3.12.2024

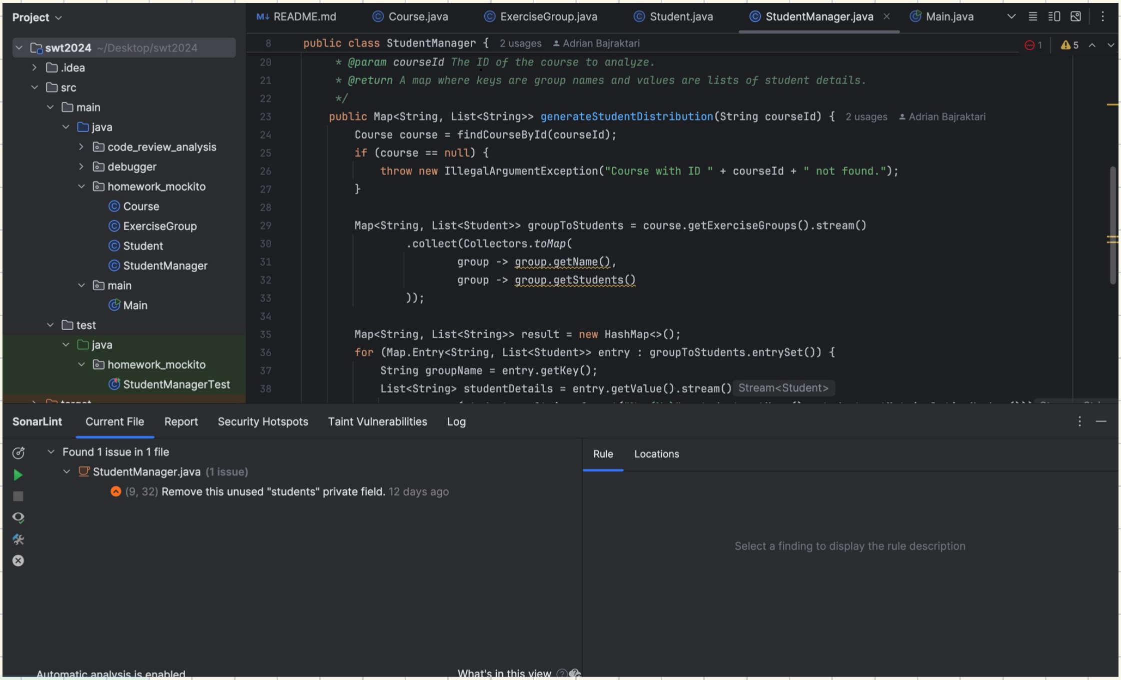
Nr.2 b)

Durch Dependecy injections kann man flexibel zwischen mocks und echten objekten wechseln.

```
// Unit-Test mit Mock
List<Course> mockCourses = Mockito.mock(List.class);
StudentManager manager = new StudentManager(null, mockCourses);

// Integrationstest mit echten Objekten
List<Course> realCourses = new ArrayList<>();
StudentManager manager = new StudentManager(null, realCourses);
```

Nr.3



```
public class StudentManager {  2 usages  Adrian Bajraktari
     * @param courseId The ID of the course to analyze.
     * @return A map where keys are group names and values are lists of student details.
     */
    public Map<String, List<String>> generateStudentDistribution(String courseId) {  2 usages  Adrian Bajraktari
        Course course = findCourseById(courseId);
        if (course == null) {
            throw new IllegalArgumentException("Course with ID " + courseId + " not found.");
        }

        Map<String, List<Student>> groupToStudents = course.getExerciseGroups().stream()
                .collect(Collectors.toMap(
                        group -> group.getName(),
                        group -> group.getStudents()
                ));

        Map<String, List<String>> result = new HashMap<>();
        for (Map.Entry<String, List<Student>> entry : groupToStudents.entrySet()) {
            String groupName = entry.getKey();
            List<String> studentDetails = entry.getValue().stream() Stream<Student>
```

Found 1 issue in 1 file

StudentManager.java (1 issue)

(9, 32) Remove this unused "students" private field. 12 days ago

Select a finding to display the rule description

Automatic analysis is enabled

What's in this view