

Übungsblatt 6 Software Technik 3.12.2024

Nr.2 b)

Durch Dependency 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

The screenshot shows an IDE with a project named 'swt2024'. The file explorer on the left shows the project structure, including 'src/main/java' and 'test/java'. The main editor displays the code for 'StudentManager.java'. The code defines a 'StudentManager' class with a 'generateStudentDistribution' method that takes a 'courseId' and returns a 'Map<String, List<String>>'. The method uses 'course.getExerciseGroups()' to get a list of exercise groups and then streams them to create a map of group names to student details. A SonarLint issue is reported at the bottom, stating 'Remove this unused "students" private field. 12 days ago'.

```
public class StudentManager {
    * @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) {
        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().map(student -> student.getName()).collect(Collectors.toList());
            result.put(groupName, studentDetails);
        }

        return result;
    }
}
```

SonarLint Current File Report Security Hotspots Taint Vulnerabilities Log

Found 1 issue in 1 file

- StudentManager.java (1 Issue)
- (9, 32) Remove this unused "students" private field. 12 days ago

Rule Locations

Select a finding to display the rule description