

SCALE FOR PROJECT

JAVA / DAY 06

Introduction

The methodology of School 21 makes sense only if peer-to-peer assessments are done seriously. This document will help you to do it properly.

- Please, stay courteous, polite, respectful and constructive in all communications during this assessment. The bond of trust between community 21 and you depends on it.
- Highlight possible malfunctions of the work done by the person and take the time to discuss and debate it.
- Keep in mind that sometimes there can be differences in interpretation of the tasks and the scope of features. Please, stay open-minded to the vision of the other.

Guidelines

- Evaluate only the files that are on the GIT repository of the student or group.
- Doublecheck that the GIT repository is the one corresponding to the student or the group as long as to the project.
- Meticulously check that nothing malicious has been used to mislead you and have you assess something except the content of the official repository.
- If you have not finished the project yet, it is compulsory to read the entire instruction before starting the review.
- Use the special flags in the scale to report an empty or non-functional solution as long as a case of cheating. In these cases, the assessment is completed and the final grade is 0 (or in a case of cheating is -42). However, except for a case of cheating, you are encouraged to continue reviewing the project to identify the problems that caused the situation in order to avoid them for the next assessment.
- You must stop giving points from the first wrong exercise even if the following exercises are correct.

Attachments

- [The exercises](#)

Preliminaries

Respect the rules:

- The repository contains the work of the student (or group).
- The student is able to explain their work at any time during the assessment.
- The general rules are respected throughout the assessment.

Yes | No

Exercise 00 - First Test

1. Структура проекта соответствует заданию?
2. Методы `isPrime` и `digitsSum` реализованы корректно?
3. Реализовано ровно пять тест-метода?
4. Используются указанные в задании аннотации?
5. Подготовлен `csv` файл?
6. В `csv` файлах не менее 10 чисел?
7. Обеспечено 100% покрытие тестами?
8. Все тесты обрабатывают корректно?
9. Тесты запускаются через команду `mvn clean compile test`?

Yes | No

Exercise 01 - EmbeddedDatabase

1. Структура проекта соответствует заданию?
2. Используются зависимости `spring-jdbc` и `hsqldb`?
3. Подготовлены файлы `schema.sql` и `data.sql`?
4. Метод `testConnection` проверяет соединение и работает корректно?
5. Тесты запускаются через команду `mvn clean compile test`?

Yes | No

Exercise 02 - Test For JDBC Repository

1. Структура проекта соответствует заданию?
2. Используется in-memory база данных?
3. Тест-методы изолированы друг от друга?
4. Тест `findById` реализован и обрабатывает корректно?
5. Тест `update` реализован и обрабатывает корректно?
6. Тест `save` реализован и обрабатывает корректно?
7. Тест `delete` реализован и обрабатывает корректно?

8. Тест `findAll` реализован и отрабатывает корректно?
9. Тесты запускаются через команду `mvn clean compile test`?

Yes | No

Exercise 03 - Test For Service

1. Структура проекта соответствует заданию?
2. Используется mock-объект `UsersRepository`?
3. Метод `authenticate` реализован в соответствии с логикой, приведенной в задании?
4. Используется stubbing методов `findById` и `update`?
5. Тест-методы реализованы для всех случаев и отрабатывают корректно?
6. Тесты запускаются через команду `mvn clean compile test`?

Yes | No