

SCALE FOR PROJECT

JAVA / DAY 05

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 - Tables & Entities

1. Структура проекта соответствует заданию?
2. Структура моделей соответствует заданию?
3. Предусмотрены реализации методов equals, hashCode, toString?
4. Существуют тестовые данные в количестве не менее 5 для каждой таблицы?
5. Используются реляционные связи?
6. Идентификаторы сущностей генерируются самой СУБД?

Yes | No

Exercise 01 - Read / Find

1. Реализованы как интерфейс, так и его имплементация?
2. Репозиторий использует HikariCP в качестве входного параметра?
3. Зависимость HikariCP указана в pom.xml?
4. В Program.java продемонстрирована корректная работа реализуемого метода?

Yes | No

Exercise 02 - Create/Save

1. В случае неверно указанных идентификаторов подсущностей выбрасывается исключение NotSavedSubEntityException?
2. После сохранения сущности у нее будет указан сгенерированный СУБД идентификатор?
3. В Program.java продемонстрирована корректная работа реализуемого метода?

Yes | No

Exercise 03 - Update

1. В Program.java продемонстрирована корректная работа реализуемого метода?
2. Будет ли обновлен внешний ключ в БД, если у сообщения заменить автора на другого?
3. Будет ли выброшено исключение NotSavedSubEntityException, если в сообщении указать автора с несуществующим id?

Yes | No

Exercise 04 - Find All

1. Реализован как интерфейс, так и его имплементация?
2. Механизм пагинации возвращает корректное количество результатов нужной страницы для тестовых данных?
3. Используется не более одного запроса в СУБД?
4. Корректно происходит заполнение подсущностей?
5. В Program.java продемонстрирована корректная работа реализуемого метода?

Yes | No