

SCALE FOR PROJECT

JAVA / DAY 07

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

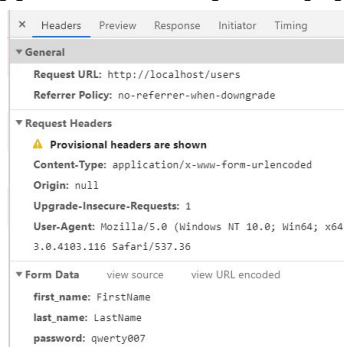
Exercise 00 - Work With Classes

1. В пакете `classes` присутствуют не менее 2-х классов?
2. Каждый из этих классов отвечает требованиям по своему наполнению?
3. Программа корректно работает для любого из этих классов?
4. Вывод программы соответствует заданию?
5. Программа корректно работает, если в пакет `classes` включить ваш корректный класс?

Yes | No

Exercise 01 - Annotations - SOURCE

1. У аннотаций указан `RetentionPolicy.SOURCE`?
2. HTML-формы генерируются при выполнении `mvn clean compile`?
3. Формы генерируются в папке `target/classes`?
4. Включите в проект свой корректный класс, программа выполняется правильно?
5. Откройте сгенерированную форму в браузере, через инструменты разработчика браузера убедитесь в корректности отправки формы. Соответствует ли вывод?



Yes | No

Exercise 02 - ORM

1. У аннотаций указан `RetentionPolicy.RUNTIME`?
2. Классы, помеченные аннотацией `@OrmEntity` соответствуют требованиям?
3. Приложение выводит сгенерированный SQL-код в консоль во время выполнения?
4. Параметр `length` у аннотации `@OrmEntity` влияет на длину строки в таблице?
5. Параметр `name` у аннотации `@OrmEntity` влияет на название столбца в таблице?
6. В `OrmManager`-е корректно реализованы все три метода?
7. `OrmManager` удаляет уже созданные таблицы?

Yes | No