

SCALE FOR PROJECT

JAVA / RUSH 00

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 - Give up, you're surrounded

1. Решение представлено двумя maven-проектами?
2. В папке resources Game-проекта присутствуют оба properties-файла?
3. Все зависимости в проекте Game указаны как dependency?
4. Команда `mvn clean package` в корне проекта Game приводит к сборке архивов приложения в папке target?
5. В одном из архивов присутствуют все необходимые зависимости?
6. Один из архивов является “запускаемым” из консоли?
7. При некорректных входных параметрах генерируется исключение `IllegalParametersException`?
8. `IllegalParametersException` - потомок `RuntimeException`?
9. Программа поддерживает запуск в production и dev-режимах? (Необходимо учесть все отличия в режимах работы, указанных в тексте задания).
10. Запустите программу не менее 10-ти раз. Во всех случаях игрок имеет возможность добраться до целевой точки?
11. Запустите программу не менее 10-ти раз. При генерации программы учтены все входные параметры?
12. Программа поддерживает “цветной” вывод?
13. Изменения в properties-файлах изменяют логику работы программы при пересборке приложения?
14. Добавьте properties-файл с каким-либо режимом, отличным от production и dev. При пересборке приложения и указании необходимого профиля программа корректно обрабатывает новый режим?
15. Получилось пройти один игровой сценарий? (выигрыш игрока, проигрыш игрока).

Yes | No