About

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 well as to the project.
- Meticulously check that nothing malicious has been used to mislead you and have you assessed 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 well as a case of cheating. In these cases, the assessment is completed and the final grade is 0 (or, in a case of cheating, -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.

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.

Нет

✓ Да

Main part

Exercise 00 – Models

- Are both domain knowledge models developed?
- Do model fields match the task requirements?
- Are all model fields private?
- Do models contain validity checks for values of transaction amounts and user balance?
- Is there a program file demonstrating how the described classes work?

Нет

✓ Да

Exercise 01 – ID Generator

- Does new identifier differ from the preceding identifier by one?
- Is Singleton pattern used to ensure existence of a single class instance?
- Does User class constructor have identifier initialization logics?
- Is "get" the only method defined for the user identifier?
- Is there a program file demonstrating how the described classes work?

Нет

✓ Да

Exercise 02 – List of Users

- Are both interface and its implementing class implemented?
- Does the class implementation imply array size increase in case of an overflow?
- Is required exception thrown when searching for a nonexistent user?
- Is there a program file demonstrating how the described classes work?

Нет

✓ Да

Exercise 03 – List of Transactions

- Are both interface and its implementing class implemented?
- Does User class contain the required field for storing a transaction list?
- Is a new transaction added within O(1)?
- Is transaction list implemented as a linked list?

- Is a required exception thrown when searching for a nonexistent transaction?
- Is there a program file demonstrating how the described classes work?

Нет

[✓ Да]

Exercise 04 – Business Logic

- Does the implemented service contain all the methods specified in the task?
- Does the service have a logic for creating a transaction pair with the same identifier for recipient and sender?
- Is a required exception thrown when attempting to transfer an amount exceeding user balance?
- Do names of each additional methods of each class fully comply with their intended use?
- Is there a program file demonstrating how the described classes work?

Нет

[✓ Да]

Exercise 05 – Menu

- Does the program support two-mode startup?
- Does the program output a warning about invalid input data?
- Will the program output all unconfirmed transfers correctly if all transfers of a specific user are deleted in Dev mode?

Нет

[✓ Да]

✓ Evaluate