Software Construction - Assignment 5

First of all, create a new folder for this assignment in your GitHub repository. Before starting the assignment, copy and paste the code developed for Assignment 1.

Please note that the folder used in the previous assignment must still contain the original code.

***Part 1*:** In the first part of this assignment, we ask you to test your Snake and Ladders game.

Use a copy of the code of Assignment 1 as the starting point.

Then, complete the following points:

1. Implement unit test cases for all the functionalities of your code. *Think about **how** each step of your code should work and **what** would break it.*

2. Document all test cases using Javadoc. The Javadocs can be auto-generated, but you need to complete it with all method information and what the test is covering.

3. If you discover a bug on your code when testing:

    1. Fix the code.

    2. Write a natural language description of **how** you found the bug and **how** you fixed it.

Address all of these 3 points in an **ANSWERS.md** file and add it to the assignment folder. If you did not find any bug in part 3, please specify it.

To complete Part 1, all tests must pass.

A guide to Javadoc can be found here:
https://www.tutorialspoint.com/java/java_documentation.htm

The tags @param and @return should always be indicated (when applicable).

Please note: Auto-generated tests and incomplete Javadocs will not be accepted.

Part 2: You will find the next part of the assignment inside the assignment5.zip file on OLAT. It contains a system source code of the program and its unit tests. Please, complete the following points:

1. Copy the source code to your Assignment 5 folder.

2. Discover why the test cases are failing and fix the code so the tests pass. To complete part 2, all tests must pass.

3. Write in natural language **why** the test cases were failing and **how** you fixed them.

Address all of these 3 points in the **ANSWERS.md** file and add it to the assignment folder. Clearly state that your answers refer to part 2.

Attention: The tests must not be modified. We expect the source code to be fixed and the unmodified test cases to run without failures.

Part 3: In the final part of the assignment we will work with Test Driven Development (TDD). Imagine you are asked to implement the UNO game. **You will find the set of rules in the "UNO rules" file in the Assignment 5 folder on OLAT.**

Then, complete the following points:

1. Identify how the system should behave:

   a. Which inputs it should accept, which it should not.

   b. What it should do with the inputs, and what it should not

   c. What are the expected outputs, which ones should not happen.

   Write a short answer to these three points in the ANSWERS.md file, clearly stating that your answers refer to part 3.

2. Draw a class diagram of the design of your system. Make appropriate use of inheritance and interfaces and show all the class attributes and methods that are fundamental for your design.

3. Implement test cases for all functionalities of the system and document these tests using Javadoc. The Javadocs can be auto-generated, but you need to complete it with all method information and what the test is covering.

You must not write the implementation of the game, only the test cases.

Note 1: Auto-generated tests and incomplete Javadocs will not be accepted. The class must not be auto-generated.

Note 2: You should not implement the system, only the unit tests. This way, the unit tests will fail as there is no code. We will evaluate whether you thought about all cases that should be tested in this scenario.

Deadline: Friday December 13 2019 at 18:00.

Please remember to make use of the ANSWERS.md file to inform us about the choices you took, and you would like us to consider during our correction.

We will only evaluate the code present on the Master branch at the time of the deadline.