Universidade do Algarve **Testing a UDP Implementation** 2024

Lab #7

Nome Turma 1 Número de Aluno

Do not forget to read the Hints section!!

Instructions

1. Download the file `lab07_udp_turma01_student_number.zip` from GitHub.
2. From Eclipse, import the `lab07_udp_turma01_student_number.zip` file above: `File`, `Import ...`, `Existing Projects into Workspace`, `select archive file`, `Browse`, `lab07_udp_turma01_student_number.zip`.
3. Right after importing the Eclipse project, you should rename it. Right-click the project, then select `Refactor`, `Rename`, and then replace `student_number` with your actual student number (including the symbol 'a' at the beginning).
4. The project has a JUnit (Java Unit) test class named `TestUDP` which you will edit to complete this laboratory. The test class is under the `udp_sequential` Java package. Experiment with this test class, for instance, right-click `TestUDP.java` and select `Run As JUnit Test`. A JUnit console will be displayed with all the tests that have passed (green) or failed (red). Initially, all tests should be green.
5. You will work on one single JUnit test in the file `TestUDP.java`. Therefore, if the last digit of your student number is 0, then you will work with the `test_0` JUnit test; if the last digit is 5 then you will work with the `test_5` JUnit test, etc.
6. You will need to do two things. First, you will write the JUnit test as described in the test above. And, you will answer a question that you're asked in the same place. Write your answer as a Java comment.
7. Once you have finished, you are ready to turn in your lab. First, you need to compress your project as a zip file. In Eclipse, right-click your project name and select `Export ...`, `General`, `Archive File`, `Next`, `Save in zip format`, `To archive file`, `lab07_udp_turma01_student_number.zip`.
8. Turn in your lab using **Tutoria**. You do not need to send a PDF file, just the zip file of the Eclipse project. You will turn in the zip file of the eclipse project by the end of the respective lab. Your zip file needs to be edited Eclipse project. It cannot be a project of another IDE.

Hints

- All the variables that you want to access have type `BSet` or `BRelation`; they are defined in the package `eventb_prelude`. Check those classes and the methods they implement.
- Variables `packet`, `sent`, `received`, and `dropped` are sets so their type is `BSet`. If you want to know the values received before or after a transmission, the you can use `udp.get_received()`. If you want to know the value of `sent` after a transmission, then you can use `udp.get_sent()`.
- `BSet` implements various methods for Unioning and Intersecting sets; check those methods out; `BSet` also implements some boolean functions to check if a set contains another, etc. For instance, the union of sets `X` and `Y` can be achieved by writing `X.union(Y)`. If you want to check if `X` contains `Y`, then you can use `X.containsAll(Y)`. If you want to check if `X` is a subset of `Y`, then you can use `X.isSubset(Y)`.

Rúbrica De Avaliação

Value	Description
1	No test has been written, or some code has been written but it is not really testing anything.
2	A test has been written, it works, but it has a major flaw (for instance, it does not test what it is required to test but something else)
3	A test has been written, it works, but it has a minor flaw (for instance, it tests a very different version of what it is required to test).
4	A test has been written, it works, but it has a minor flaw (for instance, it tests a slightly different version of what it is required to test).
5	A test has been written, it works, it tests what it is required to test

Rúbrica De Avaliação para os Invariantes

Value	Description
1	Missing
2	Bad
3	Average
4	Good
5	Excellent