Assignment 3

Hide Assignment Information

Instructions

Black-box Unit Test the Reservation Class of a Small Bed & Breakfast Reservation System

Software testing is an integral part of any software development projects. It not only provides an assessment of software quality, but also uncovers many of the hidden defects in the software. One technique of software testing is black-box testing or functional testing. In black-box testing, the functions of the software are tested without having access to the software source code.

In this assignment, you will black-box unit test the Reservation class of a small bed & breakfast reservation system. You will be given a Java jar file that includes:

1) a Java class file of the Reservation class and

2) a Java class file of the Assert library class that is used for testing.

Additionally, you will be given the specifications of both of these two given class files.

You will create four (4) sets of black-box functional unit test cases (composed of Selected Inputs, Expected Result, Actual Result, Pass|Fail) to test the constructor and the public methods of the Reservation class. Then, using the provided \*.jar files, you will code Java unit tests to execute your test cases and record their Pass|Fail verdicts. You will also explain your approach to create these black-box functional unit test cases and tests, the steps you followed, and the rationale behind your tests. You will reflect on the assignment and lessons learned.

To prepare for this assignment:

1) Review the assigned reading for the week.

2) Prepare a Java development environment. You can use the Java command line interface environment or any Java IDE (e.g., Eclipse or NetBeans). Java instructions (e.g., how to compile, integrate the provided Java \*.jar file into your development environment, etc.) will be given using the command line interface. Adjust these instructions to your selected Java IDE if you choose to use one.

3) Read the software development requirements for a small bed & breakfast reservation system (SW-Dev-Req-Small-BB-Reservation-Sys.docx)in the attached Assignment-Resources-W6.zip file. These are the same requirements as those for week 2 and week 4 assignments but repeated here for completeness.

4) Read the analysis model for a small bed & breakfast reservation system (Analysis-Model-Small-BB-Reservation-Sys.docx)in the attached Assignment-Resources-W6.zip file. This analysis model was developed based on the software development requirements for a small bed & breakfast reservation system. This is the same analysis model as the analysis model for week 4 assignment but repeated here for completeness.

5) Read the specifications of the Reservation class of the small bed & breakfast reservation system (Specifications-Reservation-Class.docx)in the attached Assignment-Resources-W6.zip file. This is the class for which you will develop black-box unit test cases and tests.

6) Read the specifications of the Assert library class (Specifications-Assert-Class.docx) in the attached Assignment-Resources-W6.zip file. You will use methods from this class to code your black-box unit tests of the Reservation class.

7) Prepare the Java jar file (w6.jar) in the attached Assignment-Resources-W6.zip file for inclusion into your development environment. The Java jar file includes a) the Reservation.class file and b) the Assert.class file

8) You will use the submission template (Submission-Template-W6-LName-FName.docx) to submit your response to this assignment. The submission template is in the attached Assignment-Resources-W6.zip file.

To complete this assignment:

1) Create four (4) sets of black-box functional unit test cases and unit tests to test the following public components of the Reservation class:

a) The constructor & getters methods

b) The setters & getters methods

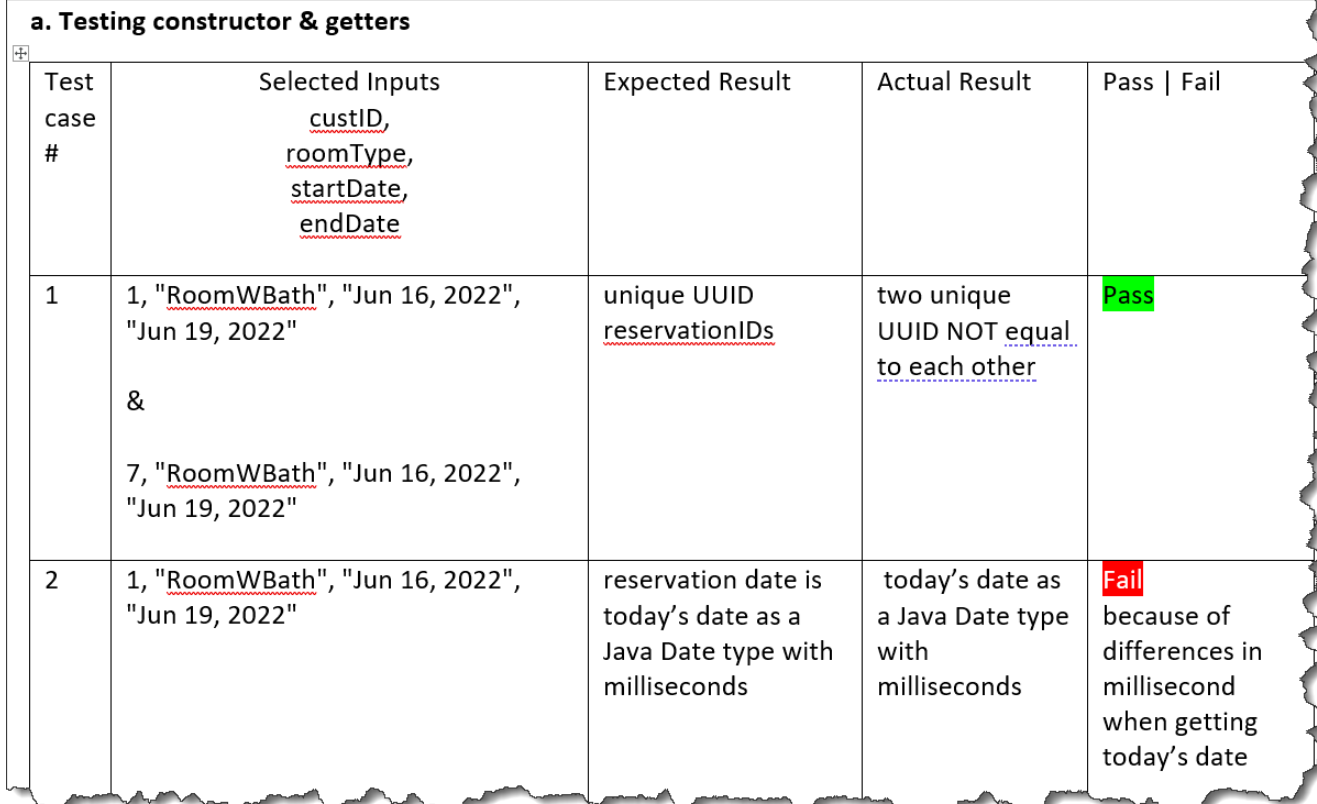
c) The calculateReservationNumberOfDays() method

d) The calculateReservationBillAmount() method

For each of these four (4) sets complete the following:

a) Create a set of black-box functional unit test cases (composed of Selected Inputs, Expected Result, Actual Result, Pass|Fail) to test the designated public component(s) of the Reservation Class.

As an example, here is a portion of the test cases for testing the Constructor & getters methods of the Reservation class:



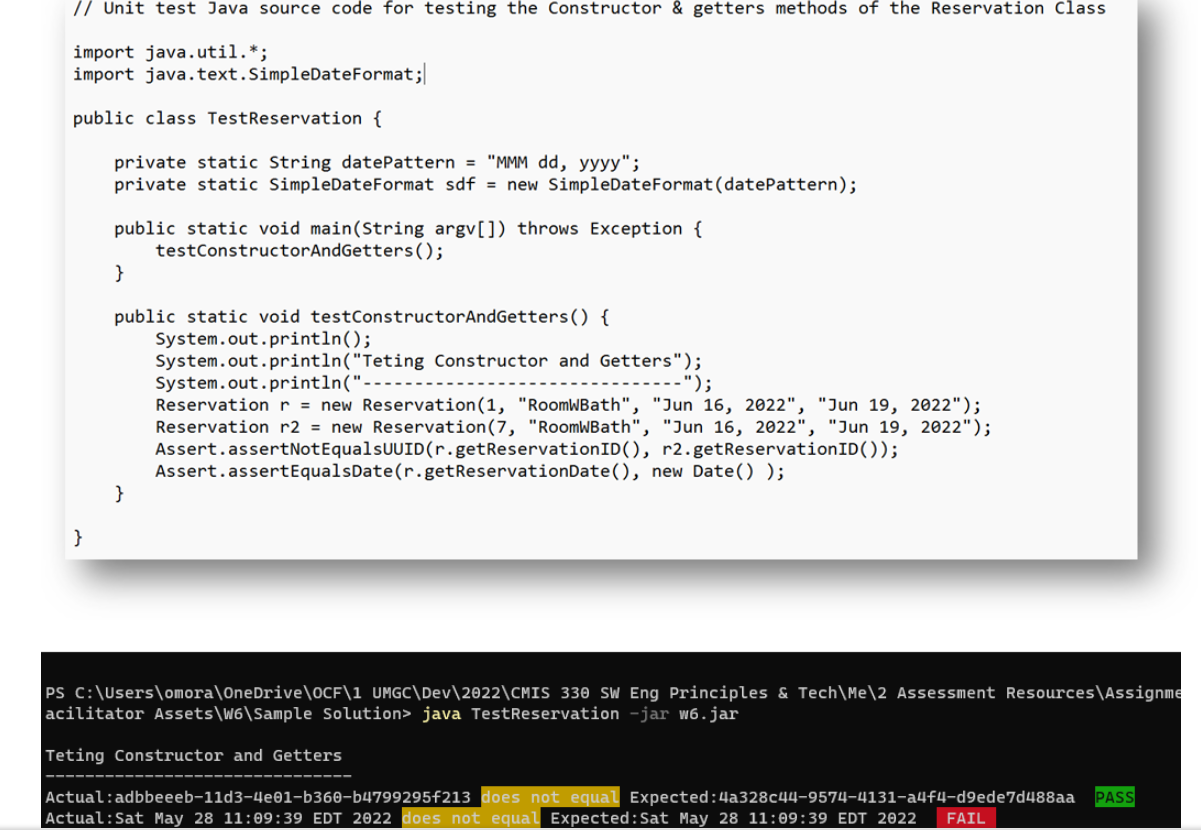
b) Using the provided \*.jar file, code Java unit tests to execute your test cases and record their Pass|Fail verdicts for the designated public component(s) of the Reservation class.

On the Java command line, compile and execute your single test file (e.g., TestReservation.java) using this single java command (note that you need to specify the \*.java extension for your single test file):

java -cp w6.jar TestReservation.java

Take screenshots of both your Java code and its test results.

As an example, here is a portion of the unit test Java source code for testing the Constructor & getters methods of the Reservation class along with the results of executing the test:



c) Explain your approach to black-box unit test designated public component(s) of the Reservation class, the steps you followed to create it, and the rationale behind your test cases and their results.

2) Reflect on your learning experience in this assignment and the lessons you learned.

To submit your response to this assignment:

1) You will use the submission template (Submission-Template-W6-LName-FName.docx) to submit your response to this assignment. The submission template is in the attached Assignment-Resources-W6.zip file.

Change the submission template file name to include your last name and your first name. Use the provided submission template to submit a single Word document that includes in this order:

Your name, data, and course code

Your complete Java unit test source code (e.g., TestReservation.java)

Test cases for testing constructor & getters of the Reservation class

Screenshots of your Java unit test source code and the results of running it

Explanation of your approach, steps, and rationale of your test of the constructor and getters of the Reservation class

Test cases for testing setters & getters of the Reservation class

Screenshots of your Java unit test source code and the results of running it

Explanation of your approach, steps, and rationale of your test of the setters & getters of the Reservation class

Test cases for testing the calculateReservationNumberOfDays() method of the Reservation class

Screenshots of your Java unit test source code and the results of running it

Explanation of your approach, steps, and rationale of your test of the calculateReservationNumberOfDays() of the Reservation class

Test cases for testing the calculateReservationBillAmount() method of the Reservation class

Screenshots of your Java unit test source code and the results of running it

Explanation of your approach, steps, and rationale of your test of the calculateReservationBillAmount() method of the Reservation class

Reflection