Name: Johnathan Brandstetter   
Date: 2/20/23  
Week: 6 - Black-box Unit Test the Reservation Class of a Small Bed & Breakfast Reservation System

Embed here a copy of your complete Java unit test source code (e.g., TestReservation.java):

/\*  
Johnathan Brandstetter  
CMIS 330 Software Engineering Principles and Techniques  
02-20-23  
\*/  
  
import java.text.SimpleDateFormat;  
import java.util.\*;  
  
public class TestReservation {  
 private static String *datePattern* = "MMM dd, yyyy";  
 private static SimpleDateFormat *sdf* = new SimpleDateFormat(*datePattern*);  
  
 public static void main(String argv[]) throws Exception {  
 *testConstructorAndGetters*();  
 *testSettersAndGetters*();  
 *testCalculateReservationNumberOfDays*();  
 *testCalculateReservationBillAmount*();  
 }  
 public static void testConstructorAndGetters() {  
 System.*out*.println();  
 System.*out*.println("Testing Constructor and Getters");  
 System.*out*.println("-------------------------------");  
 Reservation r = new Reservation(1, "RoomWBath", "Feb 16, 2023", "Feb 19, 2023");  
 Reservation r2 = new Reservation(5, "RoomWBath", "Feb 16, 2023", "Feb 19, 2023");  
 Assert.*assertNotEqualsUUID*(r.getReservationID(), r2.getReservationID());  
 Assert.*assertEqualsDate*(r.getReservationDate(), new Date() );  
 Assert.*assertEqualsString*(*sdf*.format(r.getReservationDate()), *sdf*.format(new Date()) );  
 Assert.*assertEqualsInt*(r.getGuestID(), 1);  
 Assert.*assertEqualsString*(r.getRoomType(), "RoomWBath");  
 Assert.*assertEqualsString*(r.getReservationStartDate(), "Feb 16, 2023");  
 Assert.*assertEqualsString*(r.getReservationEndDate(), "Feb 19, 2023");  
 }  
  
 public static void testCalculateReservationNumberOfDays() throws Exception {  
 System.*out*.println();  
 System.*out*.println("Testing calculateReservationNumberOfDays");  
 System.*out*.println("-------------------------------");  
 Reservation r = new Reservation(1, "RoomWBath", "Jun 12, 2023", "Jun 14, 2023");  
 Assert.*assertEqualsLong*(r.calculateResersationNumberOfDays(), 2L);  
 Reservation r2 = new Reservation(15, "NormalRoom", "Dec 27, 2022", "Jan 04, 2023");  
 Assert.*assertEqualsLong*(r2.calculateResersationNumberOfDays(), 8L);  
 Reservation r3 = new Reservation(38, "NormalRoom", "Feb 27, 2022", "Mar 02, 2022");  
 Assert.*assertEqualsLong*(r3.calculateResersationNumberOfDays(), 3L);  
 Reservation r4 = new Reservation(66, "RoomWView", "Mar 27, 2023", "Mar 28, 2023");  
 Assert.*assertEqualsLong*(r4.calculateResersationNumberOfDays(), 1L);  
  
 }  
  
 public static void testSettersAndGetters() {  
 System.*out*.println();  
 System.*out*.println("Testing Setters and Getters");  
 System.*out*.println("-------------------------------");  
 Reservation r = new Reservation(5, "NormalRoom", "Feb 07, 2023", "Feb 17, 2023");  
 r.setGuestID(55);  
 Assert.*assertEqualsInt*(r.getGuestID(), 55);  
 r.setRoom("RoomWView");  
 Assert.*assertEqualsString*(r.getRoomType(), "RoomWView");  
 r.setReservationStartDate("Feb 07, 2023");  
 Assert.*assertEqualsString*(r.getReservationStartDate(), "Feb 07, 2023");  
 r.setReservationEndDate("Feb 15, 2023");  
 Assert.*assertEqualsString*(r.getReservationEndDate(), "Feb 15, 2023");  
 }  
  
 public static void testCalculateReservationBillAmount() throws Exception{  
 System.*out*.println();  
 System.*out*.println("Testing calculateReservationBillAmount()");  
 System.*out*.println("-------------------------------");  
 Reservation r = new Reservation(1, "RoomWBath", "Jun 24, 2022", "Jun 24, 2022");  
 Assert.*assertEqualsDouble*(r.calculateReservationBillAmount(), 175);  
 Reservation r2 = new Reservation(5, "RoomWView", "Mar 21, 2024", "Mar 26, 2024");  
 Assert.*assertEqualsDouble*(r2.calculateReservationBillAmount(), 875);  
 Reservation r3 = new Reservation(67, "NormalRoom", "Sep 01, 2023", "Sep 05, 2023");  
 Assert.*assertEqualsDouble*(r3.calculateReservationBillAmount(), 500);  
 }  
}

-----------------------------------------------------------------------------------------------------------------------------------

Rubric Criteria:  
Create black-box test cases to test the constructor and the getters methods of the Reservation class 8%  
Your Response:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case # | Selected Inputs guestID, roomType, startDate, endDate | Expected Result | Actual Result | Pass | Fail |
| 1 | 1, "RoomWBath", "Feb 16, 2023", "Feb 19, 2023"  &  7, "RoomWBath", "Feb 16, 2023", "Feb 19, 2023" | Asserts:  unique UUID reservationIDs | Two unique UUID NOT are not equal | Pass |
| 2 | 1, "RoomWBath", " Feb 16, 2022", " Feb 19, 2023" | reservation date is today’s date as a Java Date type with milliseconds | Today’s date as a Java Date type with milliseconds | Fail differences in millisecond when getting today’s date |
| 3 | 1, "RoomWBath", " Feb 16, 2023", " Feb 19, 2023" | reservation date is today’s date using the “MMM dd, yyyy” pattern | Today’s date using the “MMM dd, yyyy” pattern | Pass |
| 4 | 1, "RoomWBath", " Feb 16, 2023", " Feb 19, 2023" | guestID == 1 | 1 | Pass |
| 5 | 1, "RoomWBath", " Feb 16, 2023", " Feb 19, 2023" | roomType == “RoomWBath” | “RoomWBath” | Pass |
| 6 | 1, "RoomWBath", " Feb 16, 2023", " Feb 19, 2023" | startDate == " Feb 19, 2023" | " Feb 19, 2023" | Pass |
| 7 | 1, "RoomWBath", " Feb 16, 2023", " Feb 19, 2023" | endDate ==  " Feb 19, 2023"" | " Feb 19, 2023" | Pass |

Rubric Criteria:  
Execute, using w6.jar, unit tests for the constructor and the getters method of the Reservation class. Document the unit tests code and results via screenshots 10%  
Your Response:

A picture containing text

Description automatically generated

Rubric Criteria:  
Explain approach, steps, and rationale of the test cases and unit tests of testing the constructor and the getters method of the Reservation class 5%  
Your Response:

My approach towards this assignment section was to review the assignment questions and target them according to the pattern. Multiple sections of the assignment request different parts, from assert assigned to the reservation class with getting a start date of a room type and then assigning it to userID requests. The steps I took towards it were to build the reservation and give it a unique id with the current date of the class. Then, attach information to it to review the test case.

-----------------------------------------------------------------------------------------------------------------------------------

Rubric Criteria:  
Create black-box test cases to test the setters and the getters methods of the Reservation class 8%  
Your Response:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case # | Selected Inputs guestID, roomType, startDate, endDate | Expected Result | Actual Result | Pass | Fail |
| 1 | 7, "NormalRoom", "Feb 07, 2023",  "Feb 17, 2023" & set guestID = 55 | guestID == 55 | 55 | Pass |
| 2 | 7, "NormalRoom", "Feb 07, 2023",  "Feb 17, 2023" & set roomType = “RoomWView” | roomType == “RoomWView” | “RoomWView” | Pass |
| 3 | 7, "NormalRoom", "Feb 07, 2023", "Feb 17, 2023" & set startDate = “Feb 10, 2023” | startDate = “Feb 10, 2023” | “Feb 07, 2023” | Fail fail because the setReservationStartDate() sets the endDate and not the startDate. Should look like this: this.reservation**Start**Date = newValue; |
| 4 | 7, "NormalRoom", "Feb 07, 2023", "Feb 17, 2023" & set endDate = “Feb 15, 2023” | endDate = “Feb 15, 2023” | “Feb 15, 2023” | Pass |

Rubric Criteria:  
Execute, using w6.jar, unit tests for the setters and the getters method of the Reservation class. Document the unit tests code and results via screenshots 10%  
Your Response:

A picture containing text

Description automatically generated

Rubric Criteria:  
Explain approach, steps, and rationale of the test cases and unit tests of testing the setters and the getters method of the Reservation class 5%  
Your Response:

The approach I had targeted for this section of the constructor and getter method of the reservation class was to have a nice layout per section to make it easier to read. All of it follows the assignment accordingly. However, there is a failure as the process it takes to call the method is slower than the processing time itself.

-----------------------------------------------------------------------------------------------------------------------------------

Rubric Criteria:  
Create black-box test cases to test the calculateReservationNumberOfDays() method of the Reservation class 8%  
Your Response:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case # | Selected Inputs guestID, roomType, startDate, endDate | Expected Result | Actual Result | Pass | Fail |
| 1 | 1, "RoomWBath", "Jun 12, 2023", "Jun 14, 2023" | 2L | 2L | Pass |
| 2 | 11, "NormalRoom", "Dec 27, 2022", "Jan 04, 2023" | 8L | 8L | Pass |
| 3 | 34, "NormalRoom", "Feb 27, 2022", "Mar 02, 2022" | 3L | 3L | Pass |
| 4 | 66, "RoomWView", "Jul 27, 2023", "Jul 28, 2023" | 1L | 1L | Pass |

Rubric Criteria:  
Execute, using w6.jar, unit tests for the calculateReservationNumberOfDays() method of the Reservation class. Document the unit tests code and results via screenshots 10%  
Your Response:

Graphical user interface, text

Description automatically generated

Rubric Criteria:  
Explain approach, steps, and rationale of the test cases and unit tests of testing the calculateReservationNumberOfDays() method of the Reservation class 5%  
Your Response:

The approach I had was to calculate the number of reservation steps targeted between the requested time and the assigned time. If it was in the parameters, it would pass against the assigned values. The rationale behind this shows the consistency of the equations used in the extended class to be called upon to assist the number of days for reservations.

-----------------------------------------------------------------------------------------------------------------------------------

Rubric Criteria:  
Create black-box test cases to test the calculateReservationBillAmount() method of the Reservation class 8%  
Your Response:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case # | Selected Inputs custID, roomType, startDate, endDate | Expected Result | Actual Result | Pass | Fail |
| 1 | 1, "RoomWBath", "Jun 24, 2023", "Jun 24, 2022" | 400 | 0 | Fail fails because we are using the var private long reservationNumberOfDays which is not initialized Also implementing not enough time to calculatebetween the dates even if it passed through. |
| 2 | 5, "RoomWView", "Mar 21, 2024", "Mar 26, 2024" | 875 | 875 | Pass |
| 3 | 67, "NormalRoom", "Sep 01, 2023", "Sep 05, 2023" | 500 | 480 | Fail fails because we are using the wrong rate for a normal room should be 125 instead of 120: It should return:  return (**125** \* calculateReversationNumberOfDays()); |

Rubric Criteria:  
Execute, using w6.jar, unit tests for the calculateReservationBillAmount() method of the Reservation class. Document the unit tests code and results via screenshots 10%  
Your Response:

Graphical user interface

Description automatically generated

Rubric Criteria:  
Explain approach, steps, and rationale of the test cases and unit tests of testing the calculateReservationBillAmount() method of the Reservation class 5%  
Your Response:2  
My approach to calculating the bill amount was to view the inputs and equations assigned in assert and reservation. Then, section the dates to have a different amount of time to not equal in it as it would need a longer reservation to equal the assigned amount. My steps were to evaluate the days and the cost time in between. I then had to write that on paper and multiply it to see what would happen. Then following the same predetermined guide would help me with that, as I did not understand it until I read the assigned example for this assignment. That was not easy to research. The rationale behind the test cases was to keep it simple after following the layout on paper and have something close or similar to the assignment but update the given year as we are unsure if you can preview previous registration in the assignment.-----------------------------------------------------------------------------------------------------------------------------------

Rubric Criteria:  
Reflect on the learning experience and lessons learned 8%  
Your Response:

I learned the most during this assignment. I did a lot more research than the previous assignments, and It was also confusing on the instructions used. Researching each section took some time, but in the end, cross-referencing it with the assigned information given us as the test examples. I am interested in the other sections not assigned to this assignment. Parts of the class in the assignment are unnecessary unless it is a multipurpose class for other schools or classes. It has also been a little since I have had to write a java class, so I had to recall certain information outputs. I also learned a lot more about the debugger function, going more in depth on sections that didn’t work I would end up tracing back and stepping through certain sections as long as they were assigned to the correct function.

I was however, was able to figure out how to assign the .jar with IntelliJ but it was a little different. However, I also compared it to eclipse and couldn’t pull the breakdown of the classes inside unless I unzipped it with 7zip. But it was interesting dissecting all the provided information and classes. However, I am still not one hundred percent on my assignment. Below I have attached all besides the final one of bill amount as I still haven’t initialized it. However adjusting the alternatives on my testcase making the dates equal to the final amount passed the rest.

Text

Description automatically generated