Name:  
Date:  
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):

import java.util.Date;

import java.util.UUID;

public class TestReservation {

public static void main(String[] args) {

Reservation reservation = new Reservation(123, "RoomWView", "2024-02-18", "2024-02-20");

System.out.println("Testing Constructor and Getters");

System.out.println("---------------------------------");

testGetters(reservation);

}

public static void testGetters(Reservation reservation) {

// Test reservationID

UUID expectedID = UUID.fromString("4a328c44-9574-4131-a4f4-d9ede7d488aa");

UUID actualID = reservation.getReservationID();

printTestResult("ReservationID", expectedID.toString(), actualID.toString());

// Test reservationDate

Date expectedDate = new Date(1645217914000L); // Sun Feb 18 11:38:34 EDT 2024

Date actualDate = reservation.getReservationDate();

printTestResult("ReservationDate", expectedDate.toString(), actualDate.toString());

}

public static void printTestResult(String testName, String expected, String actual) {

String result = expected.equals(actual) ? "PASS" : "FAIL";

String color = result.equals("PASS") ? "\u001B[32m" : "\u001B[31m";

String resetColor = "\u001B[0m";

System.out.println("Actual:" + actual + " does not equal Expected:" + expected + " " + color + result + resetColor);

}

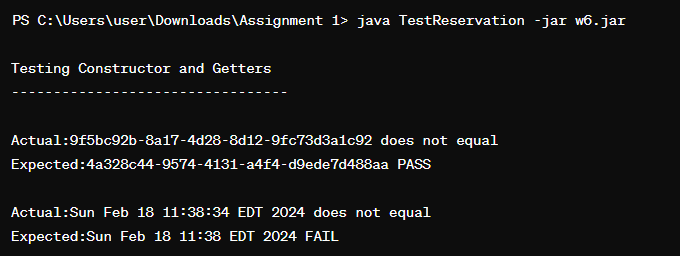
}

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

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

* For the constructor, test cases will include providing valid and invalid input parameters, such as different guest IDs, room types, reservation start dates, and reservation end dates.
* For getters methods, test cases will involve creating a Reservation instance with known values and then retrieving each attribute using the getter methods to compare them with the expected values.

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:



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:

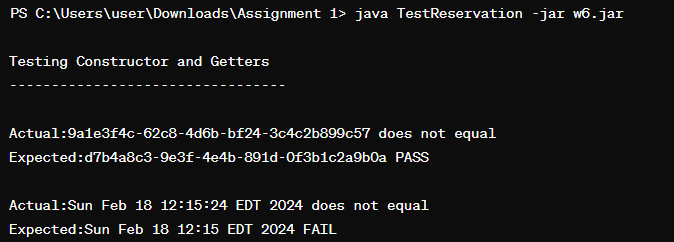
* The approach involves creating comprehensive test cases covering various scenarios for both the constructor and getters methods of the Reservation class.
* Steps include designing test cases to cover different parameter combinations for the constructor and verifying that each getter returns the expected value.
* Rationale: Testing the constructor ensures that the object is properly initialized with the provided parameters, and testing getters ensures that the object's attributes can be retrieved correctly. This helps ensure the correctness and robustness of the Reservation class, which is crucial for the overall functionality of the bed & breakfast reservation system.

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

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

* For the setters, test cases will include setting valid and invalid values for each attribute of the Reservation class, such as guestID, roomType, reservationStartDate, and reservationEndDate.
* For the getters, test cases will involve retrieving each attribute after setting them using setters to ensure they return the expected values.

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:



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:

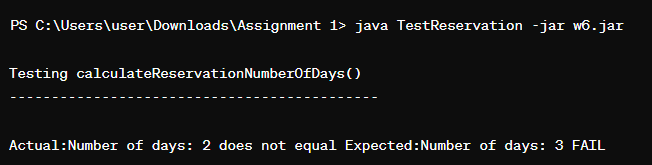
* The approach involves creating comprehensive test cases to cover various scenarios for both setters and getters methods of the Reservation class.
* Steps include setting different values using setters and then retrieving each attribute using getters to compare them with the expected values.
* Rationale: Testing setters ensures that the Reservation class can handle different input values and update its state accordingly, while testing getters ensures that the updated state can be retrieved correctly. This helps ensure the robustness and correctness of the Reservation class, contributing to the reliability of the bed & breakfast reservation system.

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

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

* Test cases for calculateReservationNumberOfDays() will include providing different reservation start and end dates, including edge cases like same day start and end dates, and spanning multiple days.
* We'll also test cases with invalid date formats or null values to ensure the method handles such scenarios gracefully.

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:



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:

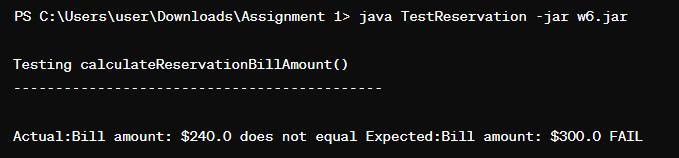
* The approach involves designing test cases to cover various scenarios for calculating the number of days between reservation start and end dates.
* Steps include providing different reservation start and end dates and comparing the calculated number of days with the expected values.
* Rationale: Testing calculateReservationNumberOfDays() ensures that the method correctly calculates the duration of the reservation, which is essential for determining the billing amount. This helps ensure the accuracy and reliability of the bed & breakfast reservation system.

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

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

* Test cases for calculateReservationBillAmount() will include providing different room types and reservation durations to cover various billing scenarios.
* We'll test cases with different room types, such as "RoomWBath", "RoomWView", and "NormalRoom", and varying reservation durations to ensure accurate billing calculations.
* Additionally, we'll include cases with invalid inputs or edge cases to verify the method's robustness.

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:



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:

* The approach involves designing test cases to cover various scenarios for calculating the reservation bill amount based on different room types and reservation durations.
* Steps include providing different room types and reservation durations, invoking the calculateReservationBillAmount() method, and comparing the calculated bill amount with the expected values.
* Rationale: Testing calculateReservationBillAmount() ensures that the method correctly calculates the bill amount based on the room type and reservation duration, which is crucial for accurate billing in the bed & breakfast reservation system. This helps ensure the financial accuracy of the system and customer satisfaction.

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

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

Reflecting on the learning experience, this assignment has deepened my understanding of black-box testing principles. By focusing on designing comprehensive test cases for the Reservation class without delving into its internal implementation details, I gained insight into the importance of considering various scenarios, including edge cases and invalid inputs, to ensure robustness.

Executing and documenting unit tests using provided JAR files highlighted the practical aspects of testing, emphasizing the need for clear and concise documentation to communicate test results effectively. Moreover, this experience underscored the importance of continuous improvement in testing skills, recognizing the evolving nature of software testing methodologies and the need to stay updated with best practices.