

Simplification (automated test)

Package: hotel.checkout	
Classname: CheckoutCTL	Method: roomIdEntered
Description: All Service charges were reported as 0.00 after entering roomId during checkout option, this test is generated to verify the reported bug.	
Test Case: CheckoutCTLTest	
S/N	Implementation
1	<pre>/** * Test of roomIdEntered method, of class CheckoutCTL. */ @Test public void testRoomIdEnteredDefaultService() throws Exception { int roomNumber = 301; double total = 7.00; Hotel hotel = HotelHelper.loadHotel(); CheckoutCTL instance = new CheckoutCTL(hotel); instance.roomIdEntered(roomNumber); String actualOutput = outContent.toString(); String actualTotalString = actualOutput.substring(actualOutput.indexOf("Total")+ 8); double actualTotal = Double.parseDouble(actualTotalString); assertEquals(total, actualTotal, 0); }</pre>
2	<pre>/** * Test of roomIdEntered method, of class CheckoutCTL. */ @Test public void testRoomIdEnteredTwoActiveService() throws Exception { int roomNumber = 301; double costRoomService = 7.00; double costBarFridge = 10.0; double total = costRoomService + costBarFridge; Hotel hotel = HotelHelper.loadHotel(); RecordServiceCTL recordService = new RecordServiceCTL(hotel); recordService.roomNumberEntered(roomNumber); recordService.serviceDetailsEntered(ServiceType.BAR_FRIDGE, costBarFridge); CheckoutCTL instance = new CheckoutCTL(hotel); instance.roomIdEntered(roomNumber); String actualOutput = outContent.toString(); String actualTotalString = actualOutput.substring(actualOutput.indexOf("Total")+ 8); double actualTotal = Double.parseDouble(actualTotalString); assertEquals(total, actualTotal, 0); }</pre>

3

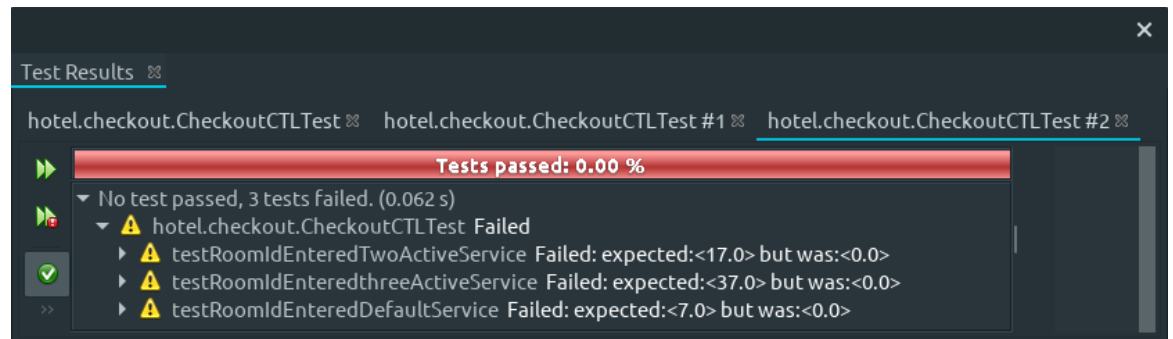
```
/**
 * Test of roomIdEntered method, of class CheckoutCTL.
 */
@Test
public void testRoomIdEnteredthreeActiveService() throws Exception {
    int roomNumber = 301;
    double costRoomService = 7.00;
    double costBarFridge = 10.0;
    double costRestaurant = 20.0;
    double total = costRoomService + costBarFridge + costRestaurant;
    Hotel hotel = HotelHelper.loadHotel();

    RecordServiceCTL recordService = new RecordServiceCTL(hotel);
    recordService.roomNumberEntered(roomNumber);
    recordService.serviceDetailsEntered(ServiceType.BAR_FRIDGE, costBarFridge );

    recordService = new RecordServiceCTL(hotel);
    recordService.roomNumberEntered(roomNumber);
    recordService.serviceDetailsEntered(ServiceType.RESTAURANT, costRestaurant );

    CheckoutCTL instance = new CheckoutCTL(hotel);
    instance.roomIdEntered(roomNumber);
    String actualOutput = outContent.toString();
    String actualTotalString = actualOutput.substring(actualOutput.indexOf("Total")+ 8);
    double actualTotal = Double.parseDouble(actualTotalString);
    assertEquals(total, actualTotal, 0);
}
```

Running
Test



The screenshot shows a 'Test Results' window with a dark theme. At the top, there are three tabs: 'hotel.checkout.CheckoutCTLTest', 'hotel.checkout.CheckoutCTLTest #1', and 'hotel.checkout.CheckoutCTLTest #2'. Below the tabs, a red progress bar indicates 'Tests passed: 0.00 %'. A summary line states 'No test passed, 3 tests failed. (0.062 s)'. A tree view shows the test results for 'hotel.checkout.CheckoutCTLTest', which is marked as 'Failed'. Under this, three specific tests are listed, all marked as 'Failed' with yellow warning icons: 'testRoomIdEnteredTwoActiveService' (expected: <17.0> but was: <0.0>), 'testRoomIdEnteredthreeActiveService' (expected: <37.0> but was: <0.0>), and 'testRoomIdEnteredDefaultService' (expected: <7.0> but was: <0.0>). On the left side of the window, there are icons for running tests: a green play button, a red stop button, a green checkmark, and a double right arrow.