External docs for topic 9

To develop a test strategy for both RegisterClass and RegistClassAssertion, it is essential to take into account the distinctions between utilizing exceptions and assertions in Java, as well as the recommended practices for testing these methodologies.

**Test Strategy for RegisterClass (Using Exceptions)**

1. Test Activation: Validate the successful activation of the tutorial space.
2. Test Slot Reservation: Ensure that slots can be reserved up to the maximum available limit.
3. Test Slot Overbooking: Attempt to reserve a slot when none are available, anticipating an exception to be raised.

**Test Strategy for RegisterClassAssertion (Using Assertions)**

1. Test Activation: Confirm the successful activation of the tutorial space.
2. Test Slot Reservation: Verify the ability to reserve slots up to the maximum available limit.
3. Test Slot Overbooking: Attempt to reserve a slot when none are available, expecting an assertion error to be triggered.

**Exceptions:**

Exceptions are employed to handle anticipated conditions during program execution, such as errors or unexpected situations. They are valuable for validating user input or managing errors that may arise during normal operations, such as file not found or network connection issues. In the context of RegisterClass, exceptions are utilized to manage scenarios where a slot cannot be reserved due to unavailability.

**Assertions:**

Assertions are utilized to verify conditions that should never occur, primarily for debugging and development purposes to detect bugs early. They are typically disabled in production environments. In the context of RegisterClassAsssertion, assertions are used to ensure that the tutorial space is always activated or not fully allocated when enrolling a student, serving to catch programming errors that should not occur.

**When to Use Each Approach**

**Use Exceptions:** When expecting specific conditions to arise during normal program operation, such as handling invalid user input or network errors. Exceptions are also suitable for recoverable conditions, enabling the program to continue running or take alternative actions.

**Use Assertions:** When aiming to identify programming errors that should not occur under normal circumstances, particularly during development and testing phases to ensure expected code behavior. They are not intended to handle runtime errors occurring in production.

**Test table**

| **Test ID** | **Test Description/Justification** | **Actual Data for This Test** | **Expected Output** | **Actual Desk Check Result** | **Desk Check Outcome** |
| --- | --- | --- | --- | --- | --- |
| T01 | Activate tutorial space | Activate tutorial space with 5 slots | Successful activation | Successful activation | Pass |
| T02 | Reserve slot | Reserve a slot when 5 slots are available | Slot reserved, 4 slots remaining | Slot reserved, 4 slots remaining | Pass |
| T03 | Reserve slot when none left | Attempt to reserve a slot when none are left | Exception thrown: "There are no more slots left" | Exception thrown: "There are no more slots left" | Pass |
| T04 | Activate tutorial space (Assertion) | Activate tutorial space with 5 slots | Successful activation | Successful activation | Pass |
| T05 | Reserve slot (Assertion) | Reserve a slot when 5 slots are available | Slot reserved, 4 slots remaining | Slot reserved, 4 slots remaining | Pass |
| T06 | Reserve slot when none left (Assertion) | Attempt to reserve a slot when none are left | AssertionError thrown: "There are no more slots left" | AssertionError thrown: "There are no more slots left" | Pass |