Details of TestCases

1. Basic Command Execution:

- Inputting commands like ‘rotate(East)’, ‘activatePanels()’, and ‘collectData()’ sequentially to observe the satellite's state change, ensuring fundamental command functionality.

2. Invalid Command Handling:

- Testing the system's response to invalid commands, such as mistyped directions in the rotate command or attempting to collect data without activating the solar panels, to verify robust error handling.

3. Multiline Command Input:

- Providing multiline commands like rotate(North)\nactivatePanels()\ncollectData() to validate the script's ability to process and execute multiple commands in succession.

4. Sequential Command Execution:

- Issuing a series of diverse commands, such as rotating in different directions, activating and deactivating panels, and collecting data, to assess the cumulative effect on the satellite's state.

5. User-Friendly Exit:

- Testing the 'exit' command to gracefully terminate the program and verifying that the final state of the satellite is displayed before exiting.

6. Logging and Error Reporting:

- Deliberately introducing errors, such as providing an unsupported command, and examining the log output to ensure that the script appropriately logs errors and warnings for debugging purposes.

These test cases cover various aspects of the Satellite Command System, ensuring that the code behaves correctly, handles errors gracefully, and provides a user-friendly interface for controlling the simulated satellite.