MSSE672 – Component Based Software

Student: John Michael Kreski

Instructor: Mohammad Abu Matar

Assignment: Week 3 – Geometry App JDBC Integration

Date: 07/20/2025

File Name: HWUnitTestExecution.doc

## Summary

The GeometryApp application now includes complete integration testing of all business logic related to quadrilateral data management using a test-dedicated H2 in-memory database. The test class QuadServiceIntegrationTest validates that JDBC methods work as expected under different scenarios, including inserting data, retrieving types, performing aggregate queries, and enforcing validation rules.

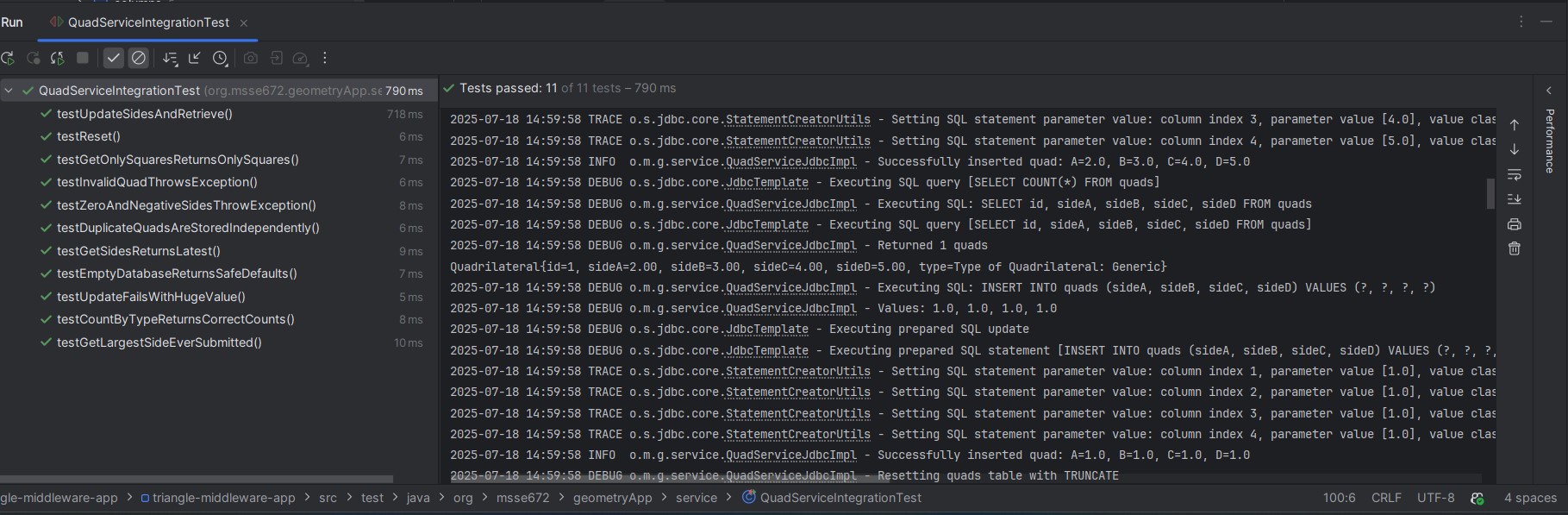
JUnit 5 and Spring Boot's testing framework were used to boot the application context, inject the service layer, and validate behaviors through assertions. Logs were added at each step for traceability, and results were printed in the console during test execution.

The test class ensures data is only persisted if valid, filters work correctly (e.g., squares only), and error cases (invalid inputs, zero/negative values) are handled gracefully.

**JUnit Test Results**

**Screenshot 1 – Test Execution Screenshot**

*All tests pass and validate correct behavior against H2 database with JDBC integration.*

**

**JUnit Coverage Table**

|  |  |  |
| --- | --- | --- |
| **Test Class** | **Method Tested** | **Purpose** |
| QuadServiceItegrationTest | testUpdateSidesAndRetrieve() | Confirms data is inserted and retrieved successfully |
| QuadServiceItegrationTest | testReset() | Confirms DB table is cleared properly |
| QuadServiceItegrationTest | testGetSidesReturnsLatest() | Validates getters for most recent record |
| QuadServiceItegrationTest | testGetOnlySquaresReturnsOnlySquares() | Filters square quads only |
| QuadServiceItegrationTest | testCountByTypeReturnsCorrectCounts() | Validates type grouping via stream API |
| QuadServiceItegrationTest | testGetLargestSideEverSubmitted() | Confirms aggregation logic using MAX/GREATEST |
| QuadServiceItegrationTest | testInvalidQuadThrowsException() | Ensures invalid quads are not inserted |
| QuadServiceItegrationTest | testZeroAndNegativeSidesThrowException() | Confirms defensive programming prevents bad inserts |
|  |  |  |
|  |  |  |
|  |  |  |

**Notes**

* PUT and DELETE REST operations are working but will be expanded in future iterations to include targeted updates/deletes by ID.
* Application is stable, test-backed, and logs SQL operations for traceability.
* Further enhancements will follow after angle-based logic and history tracking are introduced.
* This project will be maintained in a public GitHub repository for version control and learning practice.
* A `.zip` file containing only `.java` files and the `/Docs` folder is submitted as per course instructions.
  + Docs/week3
* Screenshots are provided in this single document for clarity, as per the professor’s recent guidance.
* All screenshots referenced in this document are included in the project ZIP file under the directory: Docs/Week3/assets/.