MSSE672 – Component Based Software

Student: John Michael Kreski

Instructor: Mohammad Abu Matar

Assignment: Week 4 – Geometry App Hibernate Integration

Date: 07/27/2025

File Name: HWUnitTestExecution.doc

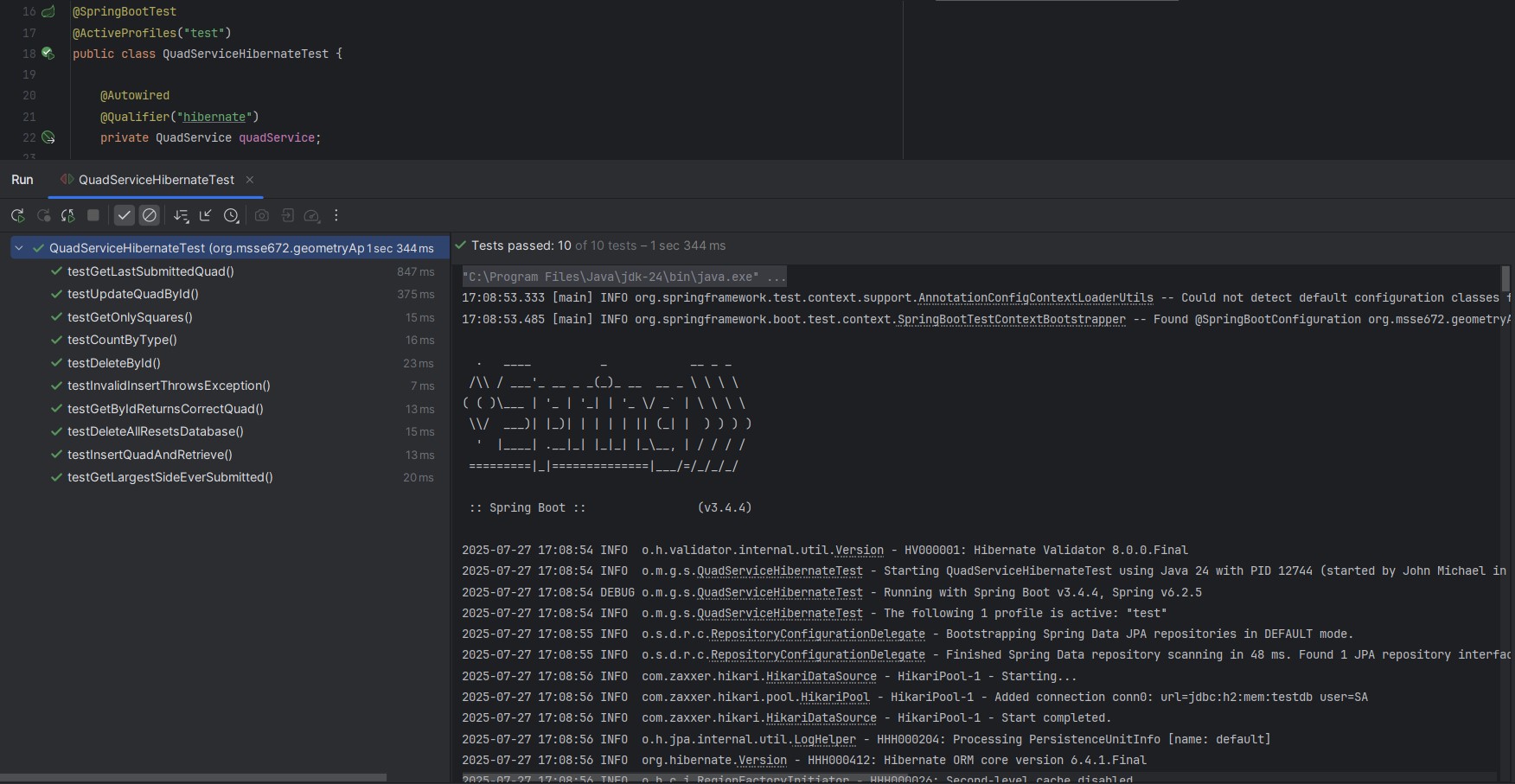
## Summary

This week, a dedicated test class, QuadServiceHibernateTest, was added to verify the correctness of the Hibernate-based service implementation (QuadServiceHibernateImpl). The class is annotated with @SpringBootTest and runs under the test profile to ensure a safe, isolated environment using an H2 in-memory database.

The test suite validates that Hibernate correctly handles CRUD operations, classification logic, aggregation queries, and business rule enforcement. Each test resets the state before execution, ensuring independence and reliability. Assertions confirm that Hibernate behaves consistently with JDBC, reinforcing the interchangeable design of the QuadService interface.

**JUnit Test Results**

**Screenshot 1 – Test Execution Screenshot**

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

**JUnit Coverage Table (QuadServiceHibernateTest.java)**

|  |  |
| --- | --- |
| **Test Method** | **Purpose** |
| testInsertQuadAndRetrieve() | Confirms a quad can be saved and retrieved |
| testGetByIdReturnsCorrectQuad() | Verifies that a quad can be fetched by its unique ID |
| testUpdateQuadById() | Ensures updates to an existing quad are persisted correctly |
| testDeleteById() | Validates that deleting a quad by ID removes it from the DB |
| testGetLastSubmittedQuad() | Confirms that the last inserted quad is accurately returned |
| testGetOnlySquares() | Filters and returns only square-shaped quads |
| testCountByType() | Confirms quad types are classified and counted properly |
| testGetLargestSideEverSubmitted() | Aggregates and returns the largest side length across all quads |
| testInvalidInsertThrowsException() | Ensures business rule violations (e.g., 1+2+3 < 100) raise exceptions |
| testDeleteAllResetsDatabase() | Confirms the reset method clears all persisted quads |

**Notes**

* The tests confirm that Hibernate correctly handles both successful and exceptional scenarios.
* Business validation logic is respected across all inserts and updates.
* The @Qualifier("hibernate") annotation ensures the correct implementation is injected during tests.
* Logs from the Hibernate service show entity lifecycle activity, helping confirm persistence behavior.
* This test suite complements the existing QuadServiceIntegrationTest which validates JDBC functionality.
* All tests are isolated, fast, and safe for CI/CD pipelines or local development.