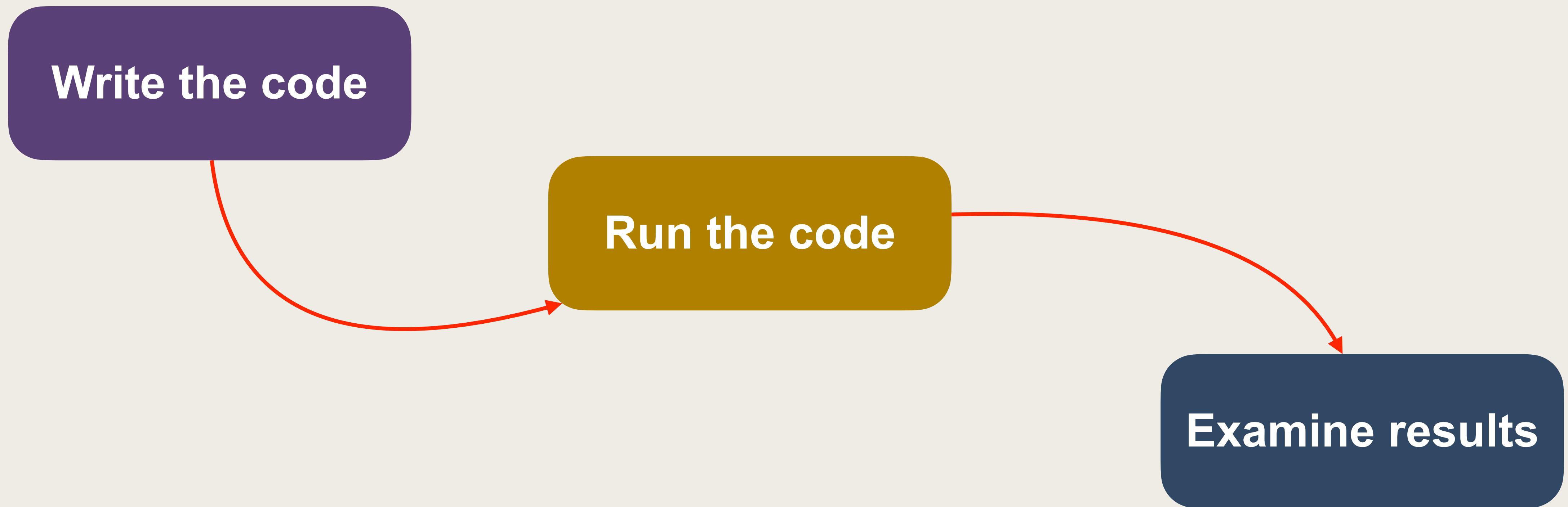


Unit Testing Overview



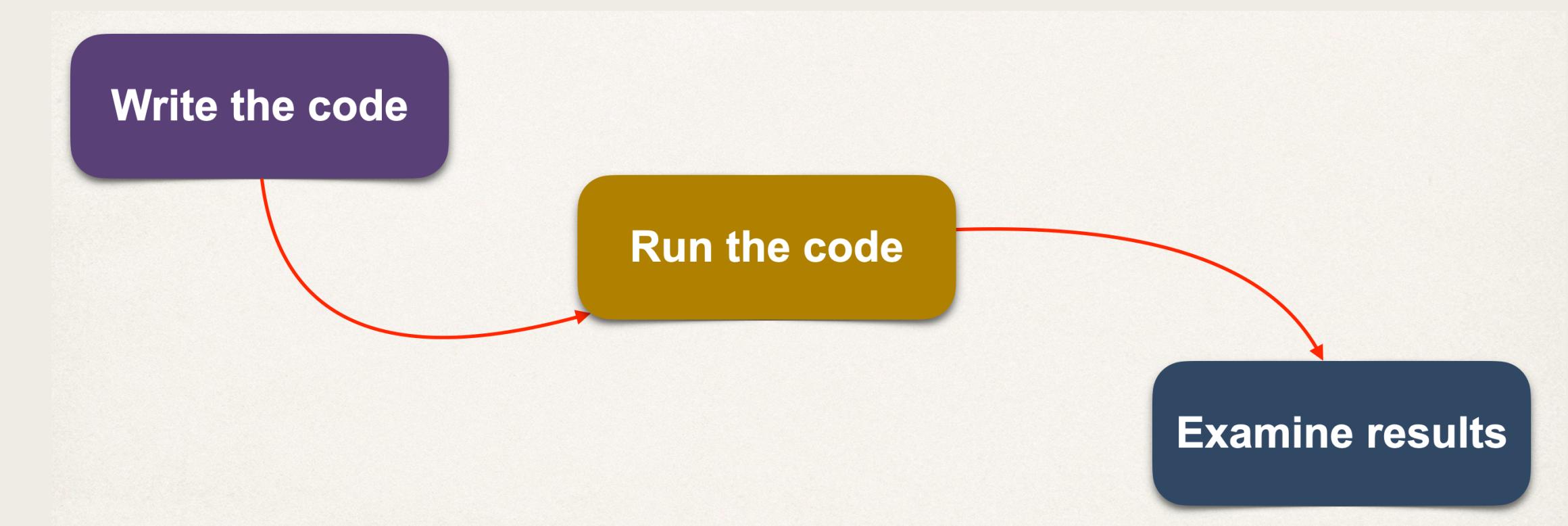
How Do We Test Code?

- Manual testing ... maybe this???



What's Wrong?

- Requires human interaction and analysis
- Test case is not reproducible
- No way to automate the test
- What if I wanted to
 - Run a battery of tests for each code commit
 - Produce a report of test results



What is Unit Testing?

- Testing an individual unit of code for correctness
- Provide fixed inputs
- Expect known output

Calculator

`add(int x, int y) : int`

Test Cases

`add(5, 2)`



`add(-3, -8)`



`add(-3, 1)`



`add(0, 0)`



Test Cases

capitalize("Luv2code")



LUV2CODE

capitalize("LUV2CODE")



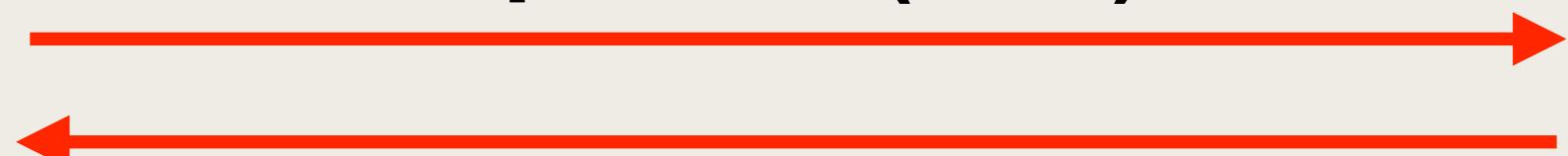
LUV2CODE

capitalize("luv2code")



LUV2CODE

capitalize(null)



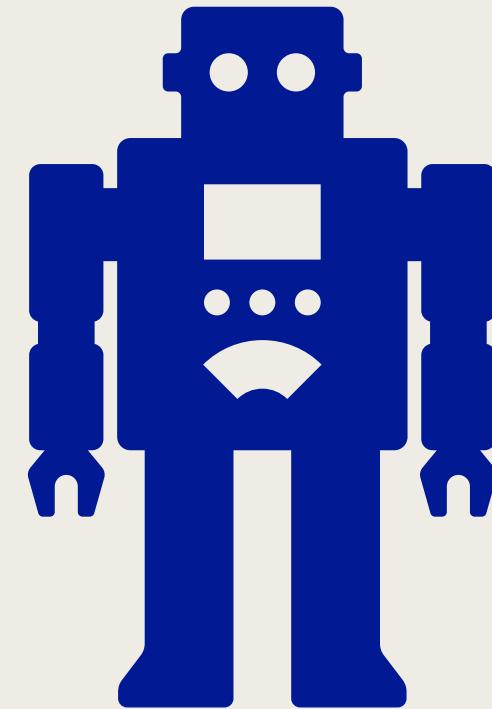
???

StringUtils

capitalize(String data) : String

Benefits of Unit Testing

- Automated tests
- Better code design
- Fewer bugs and higher reliability
- Increases confidence for code refactoring ... did I break anything??
- Basic requirement for
 - DevOps and Build Pipelines
 - Continuous Integration / Continuous Deployment (CI / CD)



Integration Testing

- Test multiple components together as part of a test plan
- Determine if software units work together as expected
- Identify any negative side effects due to integration
- Can test using mocks / stubs
- Can also test using live integrations (database, file system)

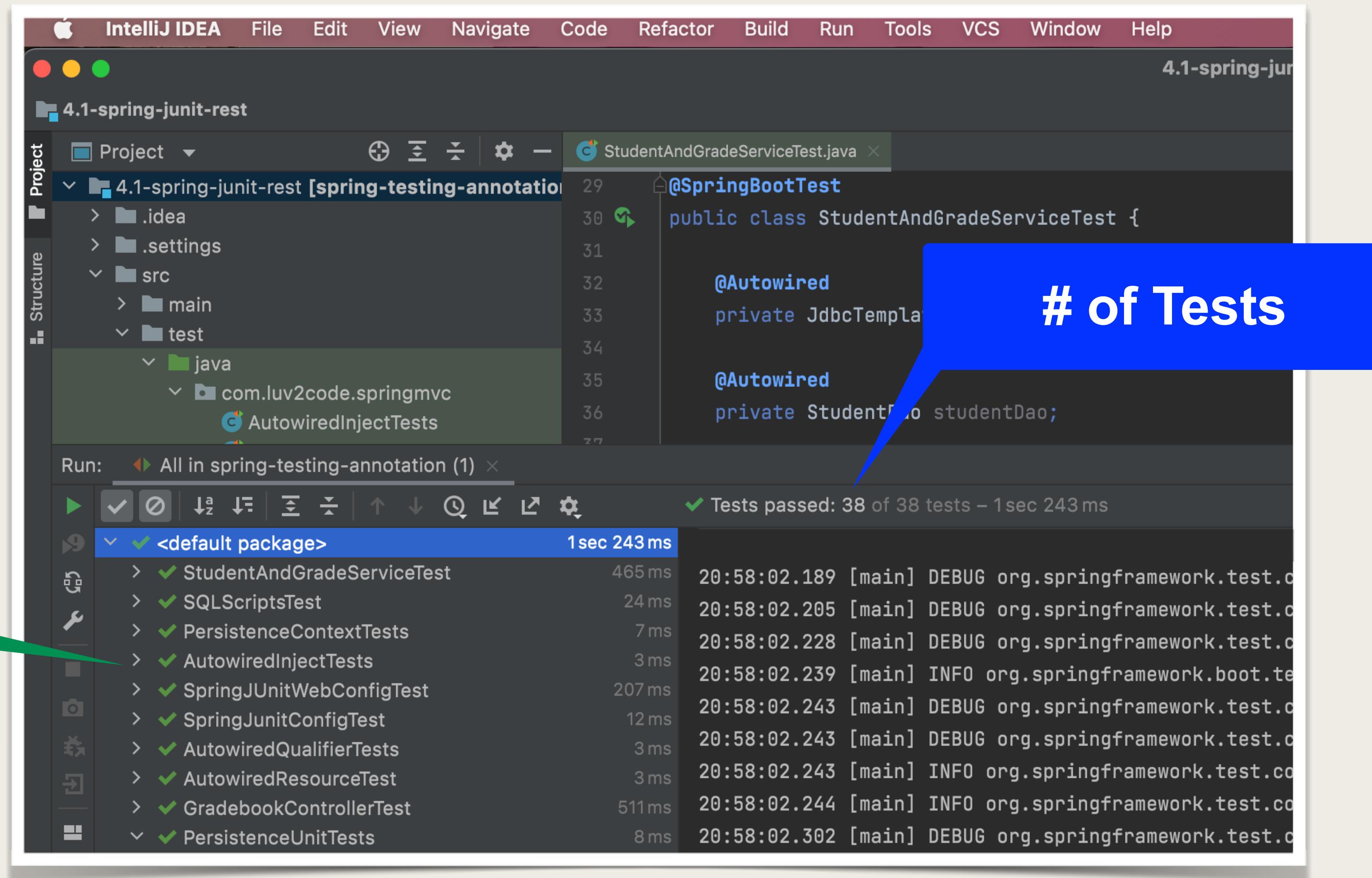
Unit Testing Frameworks

- JUnit
 - Supports creating test cases
 - Automation of the test cases with pass / fail
 - Utilities for test setup, teardown and assertions
- Mockito
 - Create mocks and stubs
 - Minimize dependencies on external components

Unit Testing Tooling Support

- IntelliJ

Test Results
Pass / Fail



The screenshot shows the IntelliJ IDEA interface with a project named "4.1-spring-junit-rest". The "StudentAndGradeServiceTest.java" file is open, displaying code annotated with @SpringBootTest and @Autowired. A blue callout bubble points to the "Run" tool window at the bottom right, which displays the test results. The results show 38 tests passed out of 38, with a total execution time of 1 sec 243 ms. The log output below the results shows various DEBUG and INFO logs from the Spring framework.

Test	Time	Log Output
StudentAndGradeServiceTest	465 ms	20:58:02.189 [main] DEBUG org.springframework.test.context.junit4.SpringJUnit4ClassRunner\$4.runTestMethod() ...
SQLScriptsTest	24 ms	20:58:02.205 [main] DEBUG org.springframework.test.context.junit4.SpringJUnit4ClassRunner\$4.runTestMethod() ...
PersistenceContextTests	7 ms	20:58:02.228 [main] DEBUG org.springframework.test.context.junit4.SpringJUnit4ClassRunner\$4.runTestMethod() ...
AutowiredInjectTests	3 ms	20:58:02.239 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBootstrapper ...
SpringJUnitWebConfigTest	207 ms	20:58:02.243 [main] DEBUG org.springframework.test.context.junit4.SpringJUnit4ClassRunner\$4.runTestMethod() ...
SpringJUnitConfigTest	12 ms	20:58:02.243 [main] DEBUG org.springframework.test.context.junit4.SpringJUnit4ClassRunner\$4.runTestMethod() ...
AutowiredQualifierTests	3 ms	20:58:02.243 [main] DEBUG org.springframework.test.context.junit4.SpringJUnit4ClassRunner\$4.runTestMethod() ...
AutowiredResourceTest	3 ms	20:58:02.243 [main] INFO org.springframework.test.context.junit4.SpringJUnit4ClassRunner\$4.runTestMethod() ...
GradebookControllerTest	511 ms	20:58:02.244 [main] INFO org.springframework.test.context.junit4.SpringJUnit4ClassRunner\$4.runTestMethod() ...
PersistenceUnitTests	8 ms	20:58:02.302 [main] DEBUG org.springframework.test.context.junit4.SpringJUnit4ClassRunner\$4.runTestMethod() ...

Unit Testing Tooling Support

- Eclipse

