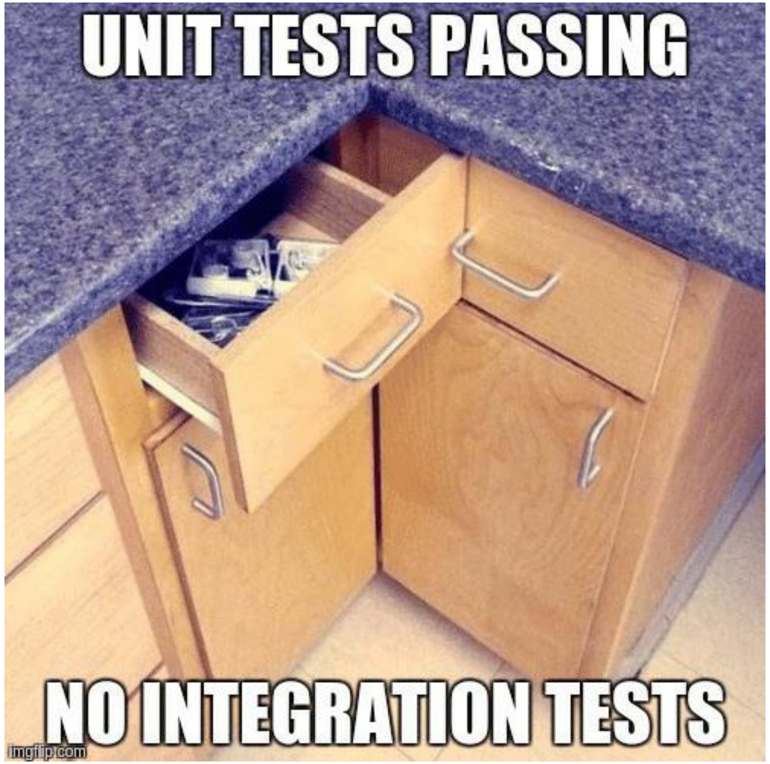
UNIT TESTS PASSING
NO INTEGRATION TESTS
imgflip.com

Module 2-7

Integration Testing

Objectives

- What is an integration test?
- DAO Integration testing

Integration Testing

- Broad category of tests that validate integration between
  - Units of code
  - Outside dependencies such as databases or network resources

Integration Testing

- Use same tools as unit tests (i.e. Junit)
- Usually slower than unit tests (but still measured in ms)
- More complex to write and debug
- Can have dependencies on outside resources like files or a database

DAO Integration Testing

DAOs exist solely to interact with database
Best tested with integration tests

Rules of testing:

- DRY – production code should be DRY – don’t repeat yourself
- WET – testing code should be WET – write everything twice

DAO Integration Testing

Integration tests with a database should ensure that the DAO code functions correctly:

- SELECT statements are tested by inserting dummy data before the test
- INSERT statements are tested by searching for the data
- UPDATE statements are tested by verifying dummy data has been changed
- DELETE statements are tested by seeing if dummy data is missing

DAO Integration Testing

Tests should be:

- Repeatable – If test passes/fails on first execution, it should pass/fail on second execution if no code has changed
- Independent – A test should be able to run on its own, independently of other tests, OR together with other tests and have the same result either way
- Obvious – When a test fails, it should be as obvious as possible as to why it failed

How to manage test data

- Remotely Hosted Shared Test Database
    - Advantages:
        - Easy setup
        - Production-like software and (possibly) hardware
    - Disadvantages
        - Unreliable and brittle
        - Last of test isolation
        - Temptation to rely on existing data (which can change)

How to manage test data

- Locally Hosted Test Database
  - Advantages
    - Production-like software
    - Reliable (local control)
    - Isolation
  - Disadvantages
    - Requires local hardware resources
    - RDBMS needs to be installed and managed

How to manage test data

- Embedded, In-memory Database
  - Advantages
    - Very Reliable
    - Consistent across dev machines (managed by source control)
    - Lightweight
  - Disadvantages
    - Not same software used in production
    - Cannot use proprietary features of production RDBMS

Mocking

- Make a replica or imitation
- Creating objects that simulate the behavior of real objects
- Typically used in unit testing, but we need to create fake data in order to test CRUD statements

Let’s Code!