Assessing Adequacy and Code Coverage Analysis with JaCoCo

We want to try to unit test thoroughly, and need to make sure we have executed every line of the program.

One way to do this is to measure which code structures in the SUT we have tested.

A tool for doing this automatically for Java programs is called JaCoCo (Java Code Coverage)

https://www.jacoco.org/jacoco/index.html

Flakey tests & how to avoid them

Flaky Tests are a Real Problem

Flakey tests are a real problem:

TAP system at Google:

- 1.6M test failures on avg./day
- 73K out of 1.6M (4.56%) test failures are caused by flakey test.

Rothermel Study:

- Changing OS, Java version & timing can lead to >50% of GUI tests being flakey.

What makes a Test Flakey?

- Concurrency
- Timing
- Change in environment.

Concurrency / Timing:

Concurrency introduces subtle failures into code and tests —

multiple threads, multiple proces

Results from multiple threads may be generated in :

- different orders, timings

and may legitimately produce different results if threads are not independent from one another.

Change in Environment :

Factors leading to flakiness —

- change in date/time.
- underlying OS
- GUI framework
- amount of available memory
- processor speed
- network speed
- disk speed.
- JVM version
- location of objects in memory

eg - default implementation of hashCode uses pointer location of object.

This means hash codes are unstable across different platforms.