Testing java code with junit and Mockito

* Introduction to unit testing and junit 5
* Mockito
* Test driven development
* Intellij idea community – free
* Regular java project
* Maven
* Gradle

What is a unit test

Eg method

Public String isEmailValid(String email) {

}

Unit test 1

Unit test 2

Unit test 3

Method may contain very complex logic

Unit test need not contain biz logic

More than 1 unit test for most of the methods

Public class Calculator {

Public int integerDivision(int dividend, int divisor) {

Return dividend/divisor;

}

}

@Test

Void testIntegerDivision\_whenValidValuesProvided\_shouldReturnExpectedResult() {

Calculator calculator = new Calculator();

Int result = calculator.integerDivision(4,2);

assertEquals(2,result, “4/2 should have returned 2”);

// Arrange

//Act

//Assert

}

Unit tests – mock objects

1. Make sure code works
2. Code works with valid and invalid params
3. Code works now and in future
4. Other code still works even after you made changes

F – unit tests run fast

I – unit tests are independent

R – repeatable

S – self validating

T – thorough and timely

Unit test – testing isolated small pieces of code with fake and mock dependencies

Integration – application code is tested without mocking database or http connections

Junit 5 – junit platform + junit Jupiter + junit vintage