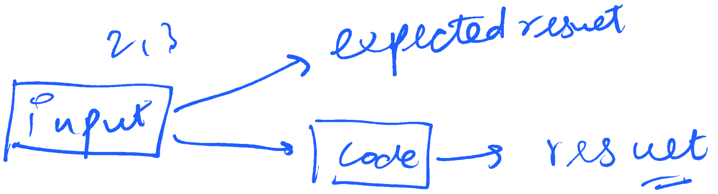
Unit Testing

Monday, 10 June 2024 9:06 PM

↳ check the correctness of your code.

Test cases.



expected result == result produced by the code.

Why do we need to write Test cases for our applications

→ test the correctness

→ to test diff. situations (diff set of inputs)

→ write test cases so that even after some changes, overall functionality of the code should not change

Types of testing

→ Unit Testing ✓

→ System Testing

→ Functional Testing

→ Fat Testing

→ Automation tests

→ Integration testing

→ Load Testing / Perf Testing

Unit Tests

↳ testing smallest working unit of your code

Calculator {

add( )

sub( )

div( )

multiply( )

;

Framework for unit Testing

JUnit] → testing Java apps

Show off knowledge

Junit 5 = Junit Platform + → actually runs the test case with JVM

Junit Vintage + → backward compatibility with Junit 4

Junit Jupiter → new functionalities of Junit 5

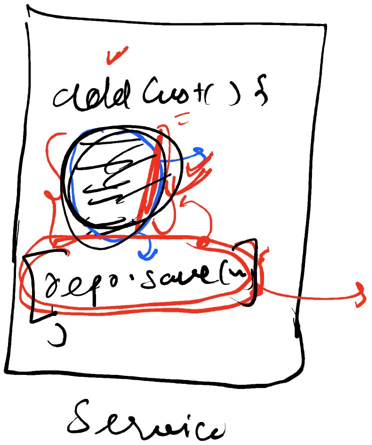
Some imp points to remember

•) Test cases are always written inside the 'test' folder (not the main folder).

•) naming of test case functions is very imp.

(A) The best way to understand a new code base is to write test cases.

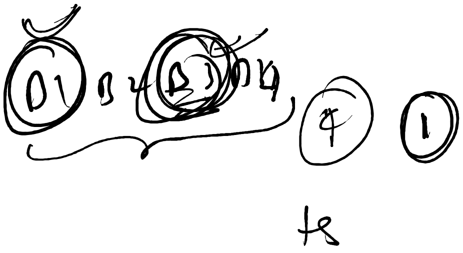
Mockito



when (repo.save (n) -> new Custo)

Mockito -> mocking the external dependencies

Mojito



```
{add( ) {
  x = x+5;
  y = y+7;
  x -> fun( )
  ret x+y
}
```