

Unit Testing

Introduction

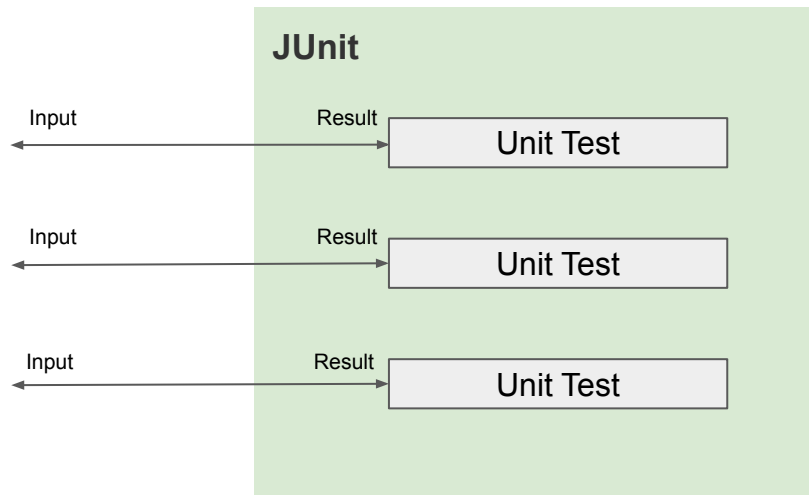
What is a Unit Test

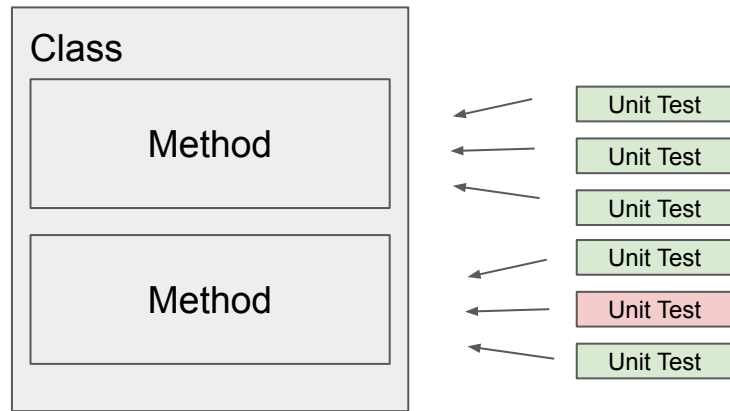
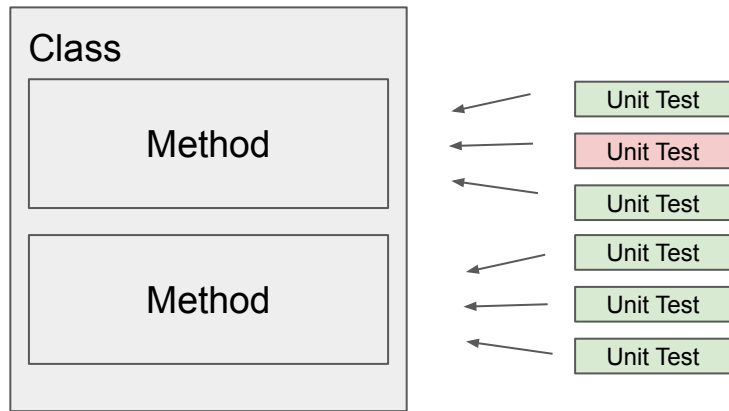
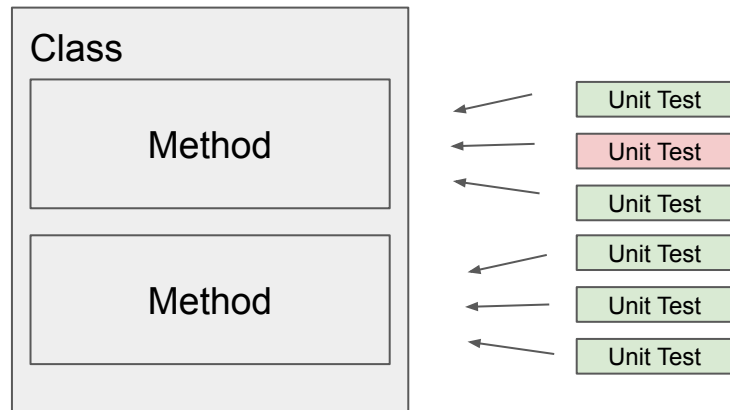
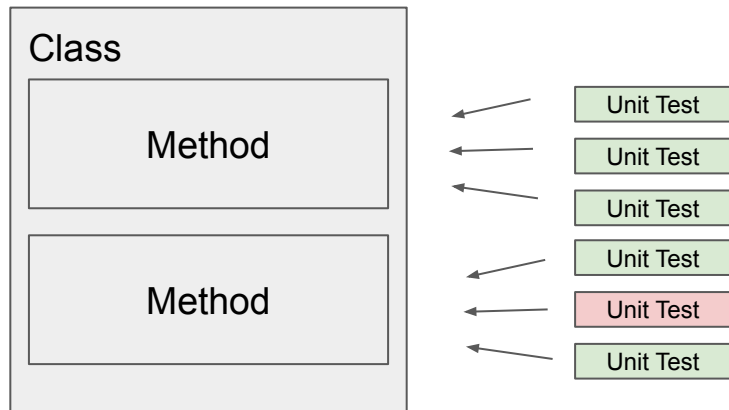
A Unit test is very small method that you write to test some part of your code.

```
public String isValidEmail(String email) {
```

Code that needs to be tested


```
}
```





Unit Test Example

```
public class Calculator {  
  
    public int integerDivision(int dividend, int divisor) {  
        return dividend / divisor;  
    }  
  
}
```



```
@Test  
void testIntegerDivision_whenValidValuesProvided_shouldReturnExpectedResult() {
```


```
    //Arrange  
    Calculator calculator = new Calculator();
```

```
    //Act  
    int result = calculator.integerDivision(4,2);
```


```
    //Assert  
    assertEquals(2, result, "4/2 should have returned 2");
```

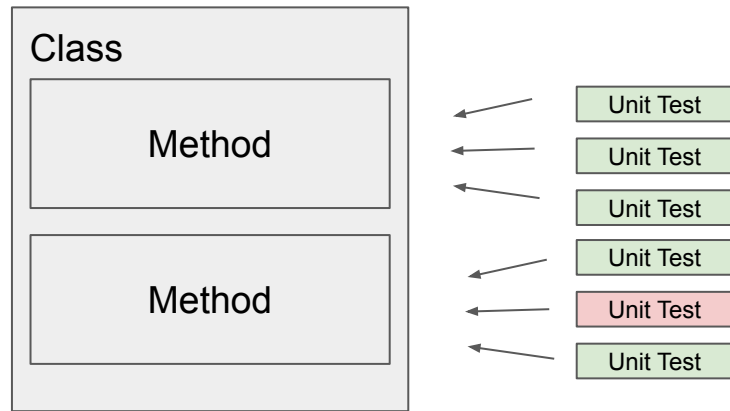
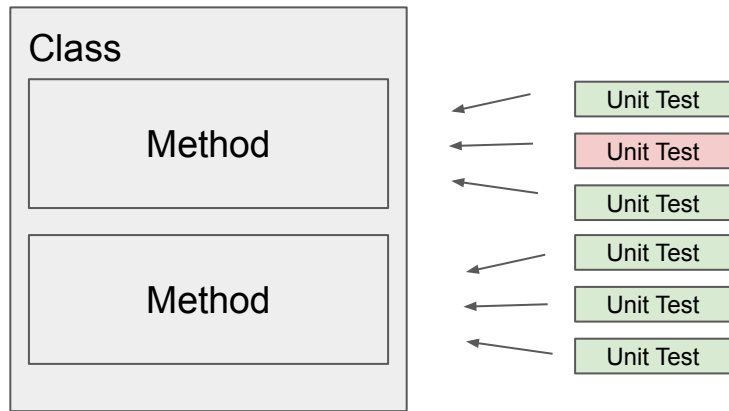
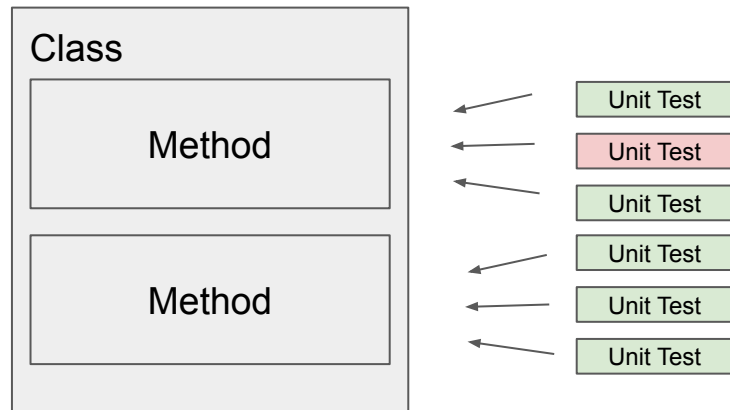
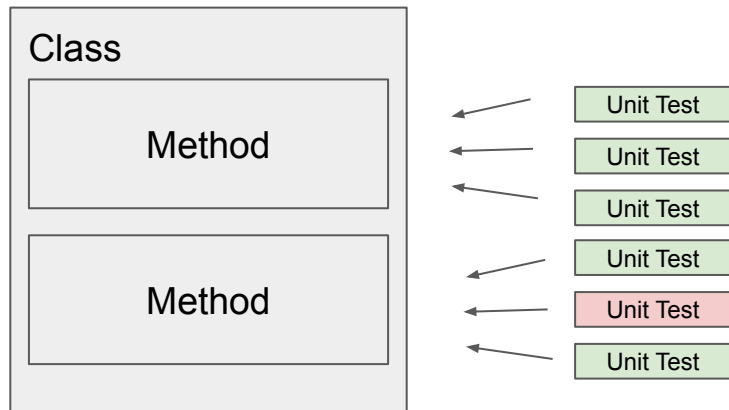
```
}
```

```
12 @Test
13 void integerDivision() {
14     // Arrange
15     Calculator calculator = new Calculator();
16
17     // Act
18     int actualResult = calculator.integerDivision( dividend: 4, divisor: 2);
19
20     // Assert
21     assertEquals( expected: 2, actualResult);
22 }
```



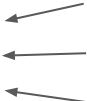
```
12 @Test
13 void integerDivision() {
14     // Arrange
15     Calculator calculator = new Calculator();
16
17     // Act
18     int actualResult = calculator.integerDivision( dividend: 4, divisor: 2);
19
20     // Assert
21     assertEquals( expected: 2, actualResult);
22 }
```





Mock HTTP Client

HTTP Client.
Dependency.



Unit Test

Unit Test

Unit Test



- All parameters are valid
- One parameter has invalid value
- Invalid Response from HTTP Client

Why write Unit Tests?

- Make sure that the code works,
- Code works with valid and invalid input parameters,
- Code works now and in the future,
- Other code still works even after you made changes.

Testing Pyramid

