Fresher Android

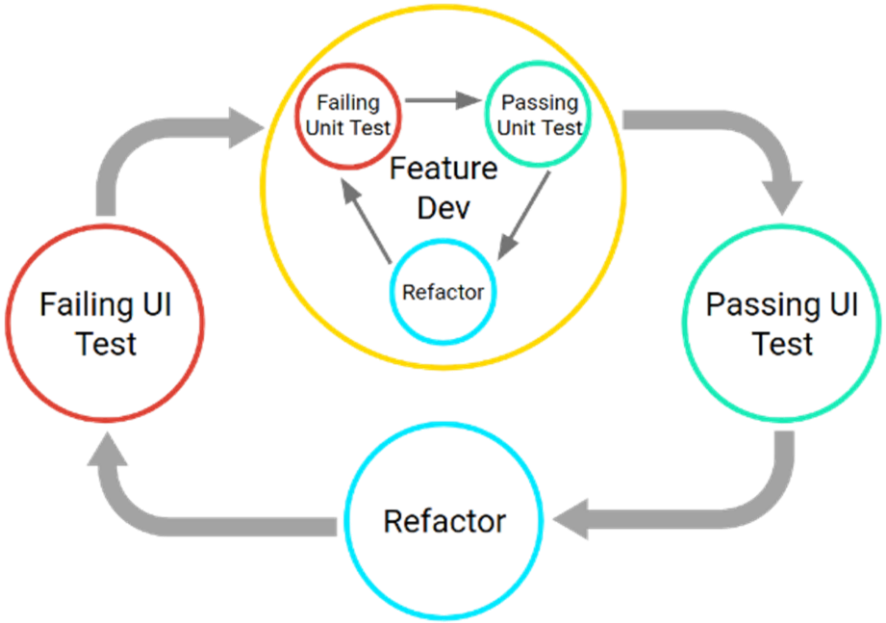
UnitTest – Day 1

GLOBAL SMART TECHNOLOGIES

Introduction

Introduction

Fundamentals of Testing



GLOBAL SMART TECHNOLOGIES

FPT

Configure environment

#1. Configure your test environment

- Organize test directories based on execution environment
- Consider tradeoffs of running tests on different types of devices
- Consider whether to use test doubles

#2. The basic framework testing

UI Testing :

- Espresso
- UIAutomator
- Appium

Local unit test :

- Mockito
- MockK
- Robolectric

Functions

1. Configure your test environment
2. The basic framework testing

FPT | GLOBAL SMART TECHNOLOGIES

MockK

#1. What is MockK?

- MockK is a mocking framework, Kotlin-based library that is used for effective unit testing of Kotlin applications.

- MockK is used to mock interfaces so that a dummy functionality can be added to a mock interface that can be used in unit testing.

#2. Installation

- All you need to get started is just to add a dependency to MockK library.

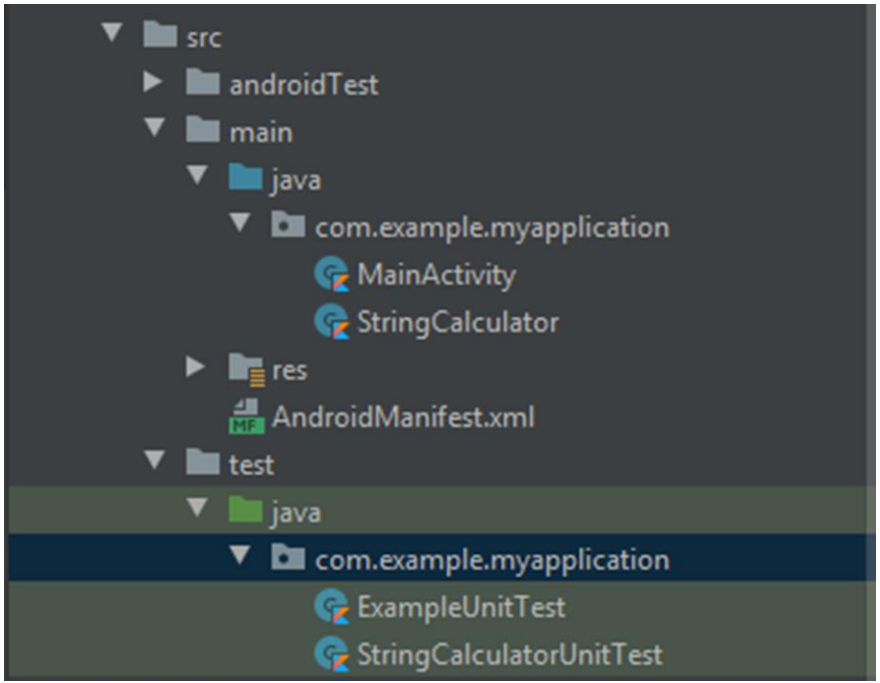
On build.gradle file :

```
dependencies {
        testImplementation "io.mockk:mockk:{version}"
    }
```

With Kotlin 1.2 use version : 1.9.kotlin12
With Kotlin 1.3 use version : 1.9

#3. How to create class test?

- Create folder & class test



#4. Mock object with MockK

- **Mock**

  Ex: *val car = mockk<**Car**>()*

  or

  ```
  @MockK
  lateinit var car1: Car
  ```

- **Spy**

  Ex: *val car = **spyk(Car())***

  or

  ```
  @SpyK
  lateinit var car1: Car
  ```

#5. Mocking method with MockK

- **every** { car.**drive**(**Direction**.**NORTH**) } returns **Outcome**.**OK**

- **every** { **func**() } returns **Car**()

#6. Verify

- verify { car.drive(Direction.NORTH) }

- assertEquals(3, MockObj.add(1, 2))

Functions

1. What is MockK?
2. Installation
3. How to create class test?
4. Mock object with MockK
5. Mocking method with MockK
6. Verify

Lesson Summary

- Introduction
- Configure environment
- MockK

Thank you