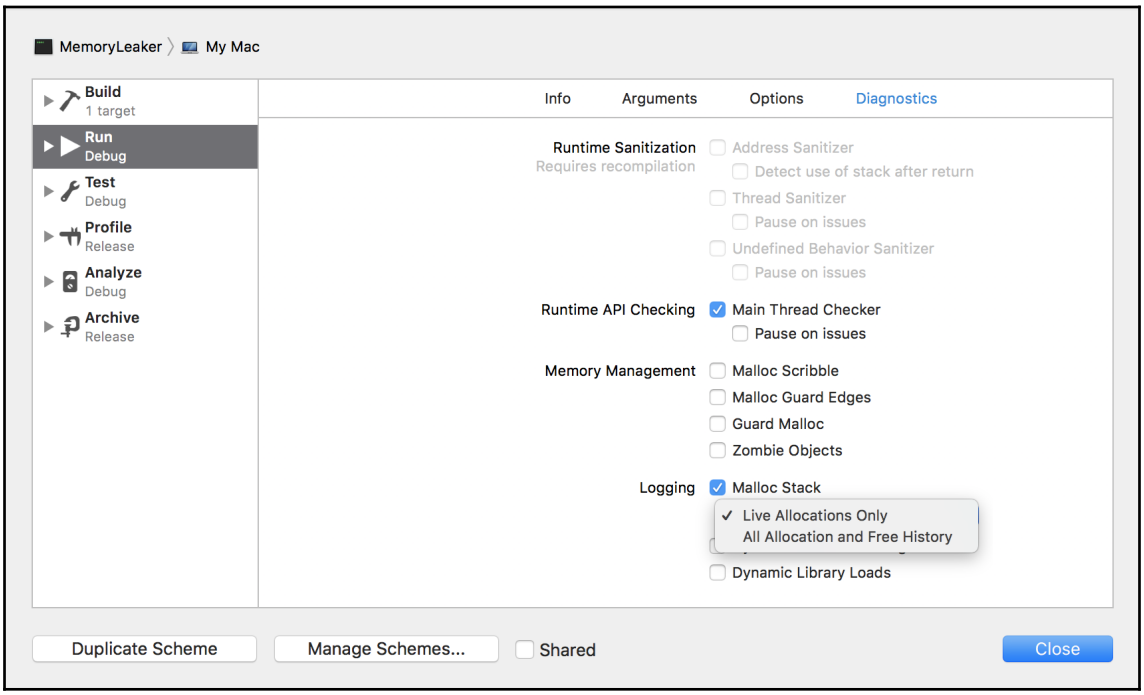
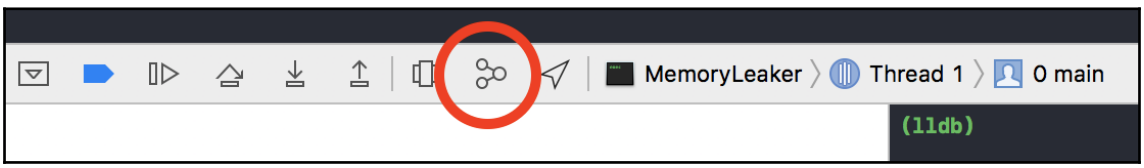
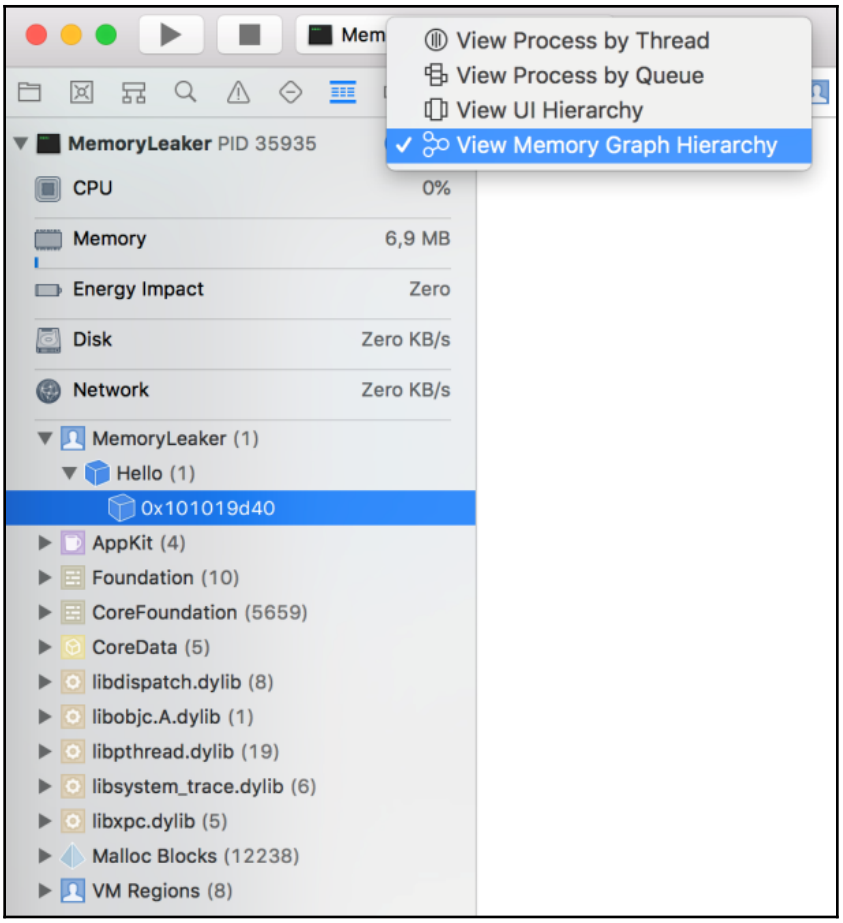
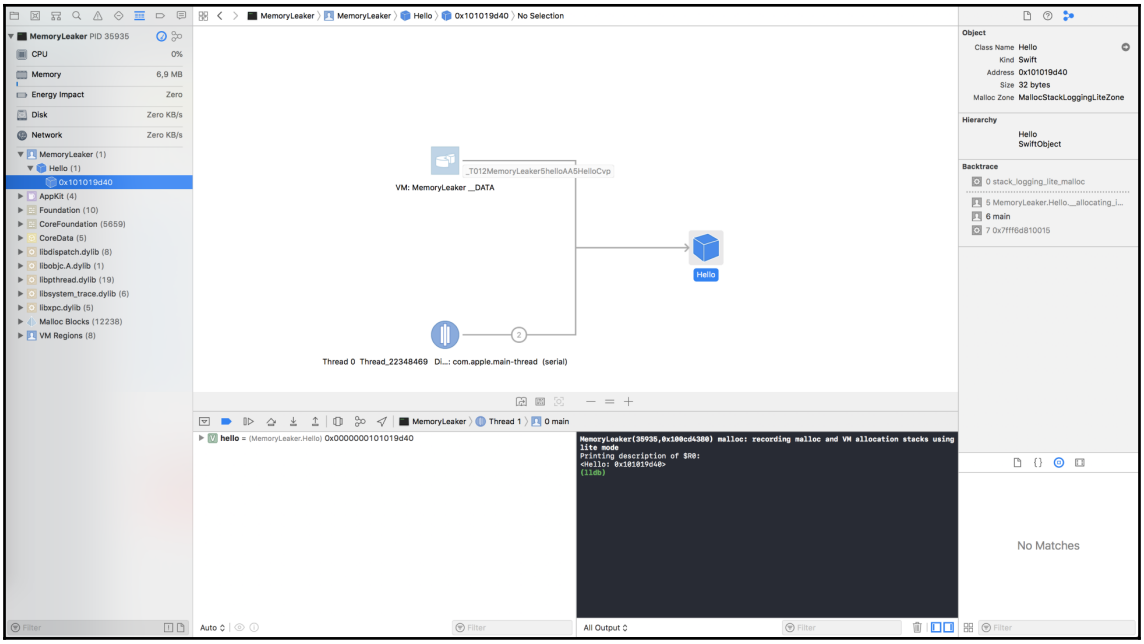
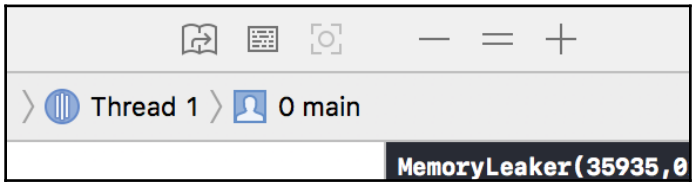
Chapter 2: Understanding ARC and Memory Management

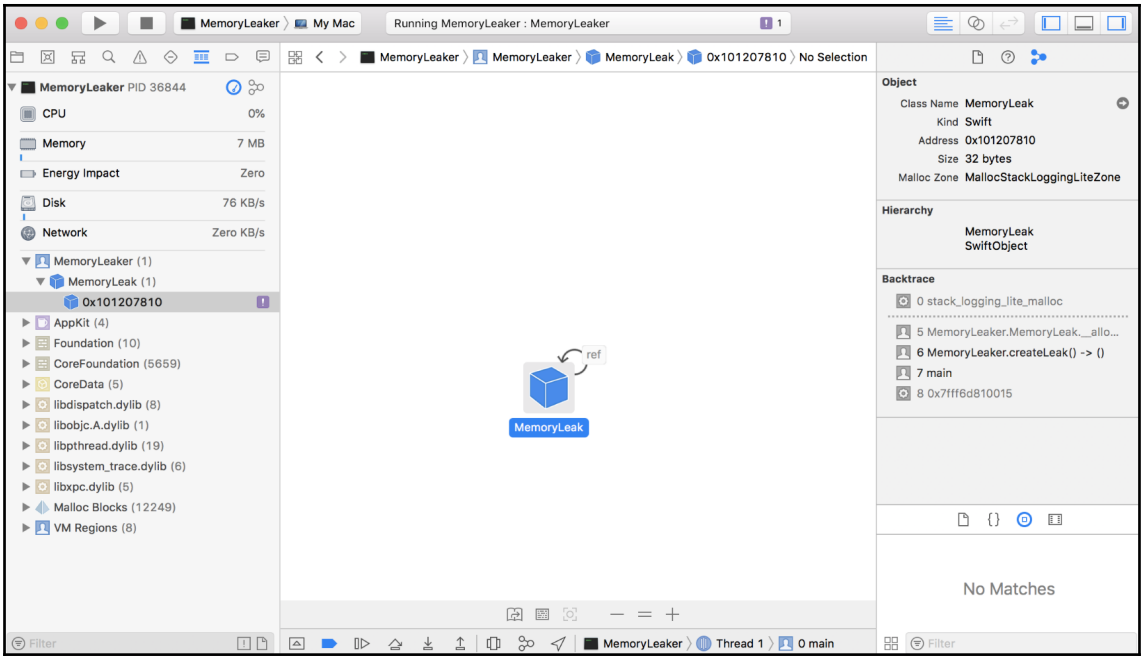


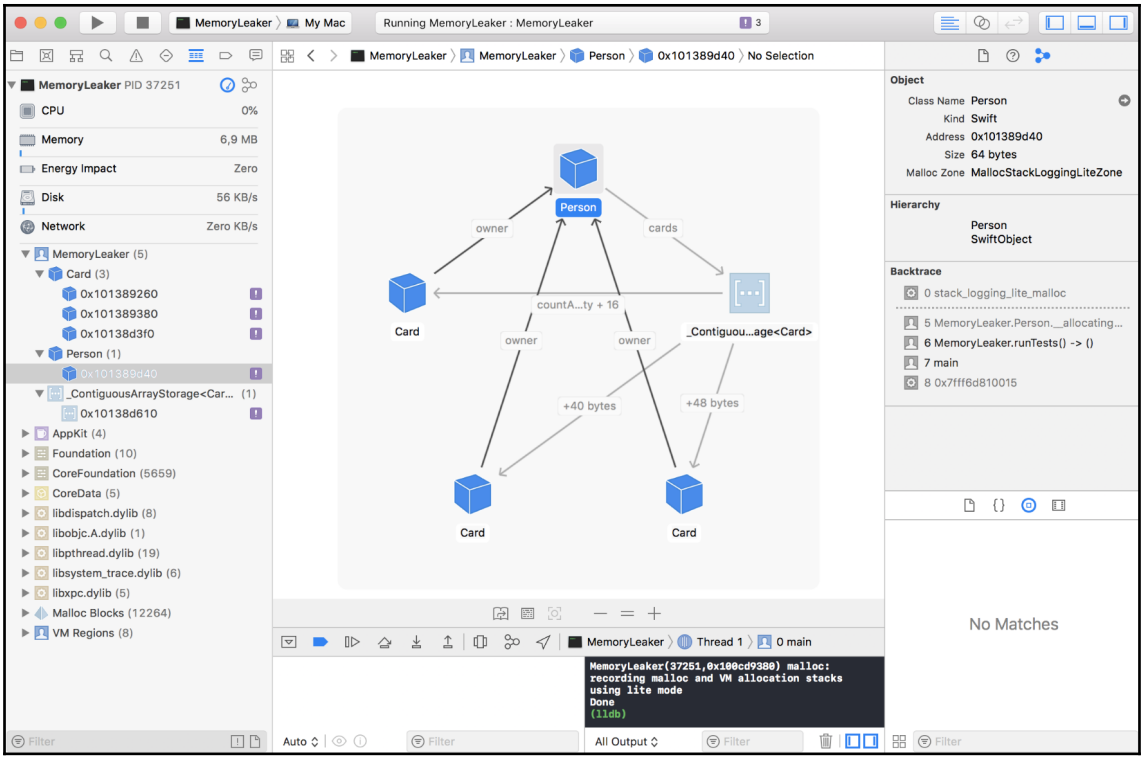



Mem
View Process by Thread
View Process by Queue
View UI Hierarchy
View Memory Graph Hierarchy
MemoryLeaker PID 35935
CPU 0%
Memory 6,9 MB
Energy Impact Zero
Disk Zero KB/s
Network Zero KB/s
MemoryLeaker (1)
Hello (1)
0x101019d40
AppKit (4)
Foundation (10)
CoreFoundation (5659)
CoreData (5)
libdispatch.dylib (8)
libobjc.A.dylib (1)
libpthread.dylib (19)
libsystem_trace.dylib (6)
libxpc.dylib (5)
Malloc Blocks (12238)
VM Regions (8)

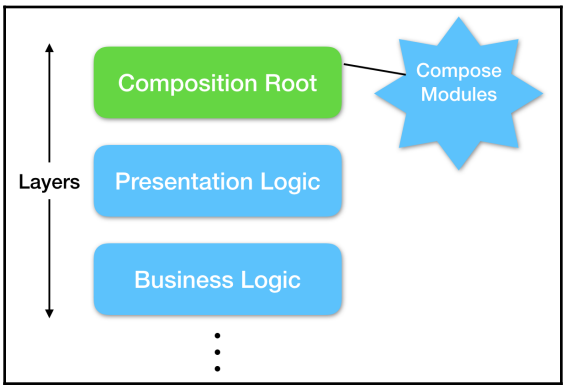

MemoryLeaker PID 35935
CPU 0%
Memory 6,9 MB
Energy Impact Zero
Disk Zero KB/s
Network Zero KB/s
MemoryLeaker (1)
Hello (1)
0x101019d40
AppKit (4)
Foundation (10)
CoreFoundation (5659)
CoreData (5)
libdispatch.dylib (8)
libobjc.A.dylib (1)
libpthread.dylib (19)
libsystem_trace.dylib (6)
libxpc.dylib (5)
Malloc Blocks (12238)
VM Regions (8)
MemoryLeaker 〉 MemoryLeaker 〉 Hello 〉 0x101019d40 〉 No Selection
_T012MemoryLeaker5helloAA5HelloCvp
VM: MemoryLeaker __DATA
Hello
Thread 0 Thread_22348469 Di...: com.apple.main-thread (serial)
MemoryLeaker 〉 Thread 1 〉 0 main
hello = (MemoryLeaker.Hello) 0x0000000101019d40
MemoryLeaker(35935,0x100cd4380) malloc: recording malloc and VM allocation stacks using lite mode
Printing description of $R0:
<Hello: 0x101019d40>
(lldb)
Object
Class Name Hello
Kind Swift
Address 0x101019d40
Size 32 bytes
Malloc Zone MallocStackLoggingLiteZone
Hierarchy
Hello
SwiftObject
Backtrace
0 stack_logging_lite_malloc
5 MemoryLeaker.Hello__allocating_i...
6 main
7 0x7fff6d810015
No Matches
Auto
All Output

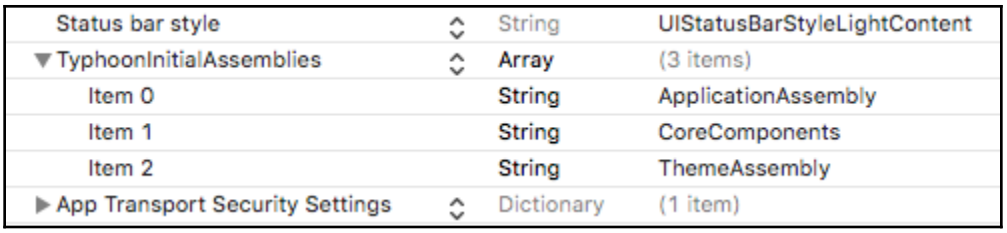

Thread 1 〉 0 main
MemoryLeaker(35935,0


MemoryLeaker 〉 My Mac
Running MemoryLeaker : MemoryLeaker
MemoryLeaker 〉 MemoryLeaker 〉 MemoryLeak 〉 0x101207810 〉 No Selection
MemoryLeaker PID 36844
CPU 0%
Memory 7 MB
Energy Impact Zero
Disk 76 KB/s
Network Zero KB/s
MemoryLeaker (1)
MemoryLeak (1)
0x101207810
AppKit (4)
Foundation (10)
CoreFoundation (5659)
CoreData (5)
libdispatch.dylib (8)
libobjc.A.dylib (1)
libpthread.dylib (19)
libsystem_trace.dylib (6)
libxpc.dylib (5)
Malloc Blocks (12249)
VM Regions (8)
ref
MemoryLeak
Object
Class Name MemoryLeak
Kind Swift
Address 0x101207810
Size 32 bytes
Malloc Zone MallocStackLoggingLiteZone
Hierarchy
MemoryLeak
SwiftObject
Backtrace
0 stack_logging_lite_malloc
5 MemoryLeaker.MemoryLeak.__allo...
6 MemoryLeaker.createLeak() -> ()
7 main
8 0x7fff6d810015
No Matches
MemoryLeaker 〉 Thread 1 〉 0 main


MemoryLeaker
My Mac
Running MemoryLeaker : MemoryLeaker
MemoryLeaker PID 37251
CPU 0%
Memory 6,9 MB
Energy Impact Zero
Disk 56 KB/s
Network Zero KB/s
MemoryLeaker (5)
Card (3)
0x101389260
0x101389380
0x10138d3f0
Person (1)
0x101389d40
_ContiguousArrayStorage<Car... (1)
0x10138d610
AppKit (4)
Foundation (10)
CoreFoundation (5659)
CoreData (5)
libdispatch.dylib (8)
libobjc.A.dylib (1)
libpthread.dylib (19)
libsystem_trace.dylib (6)
libxpc.dylib (5)
Malloc Blocks (12264)
VM Regions (8)
Person
0x101389d40
No Selection
Person
owner
cards
countA...ty + 16
Card
_Contiguou...age<Card>
owner
owner
+40 bytes
+48 bytes
Card
Card
Object
Class Name Person
Kind Swift
Address 0x101389d40
Size 64 bytes
Malloc Zone MallocStackLoggingLiteZone
Hierarchy
Person
SwiftObject
Backtrace
0 stack_logging_lite_malloc
5 MemoryLeaker.Person.__allocating...
6 MemoryLeaker.runTests() -> ()
7 main
8 0x7fff6d810015
No Matches
Thread 1
0 main
MemoryLeaker(37251,0x100cd9380) malloc:
recording malloc and VM allocation stacks
using lite mode
Done
(lldb)
Auto
All Output

Chapter 11: Implementing Dependency Injection

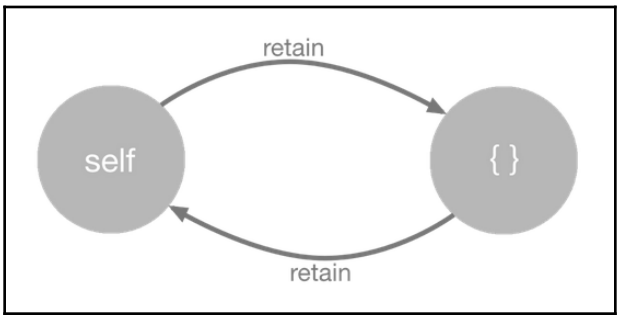




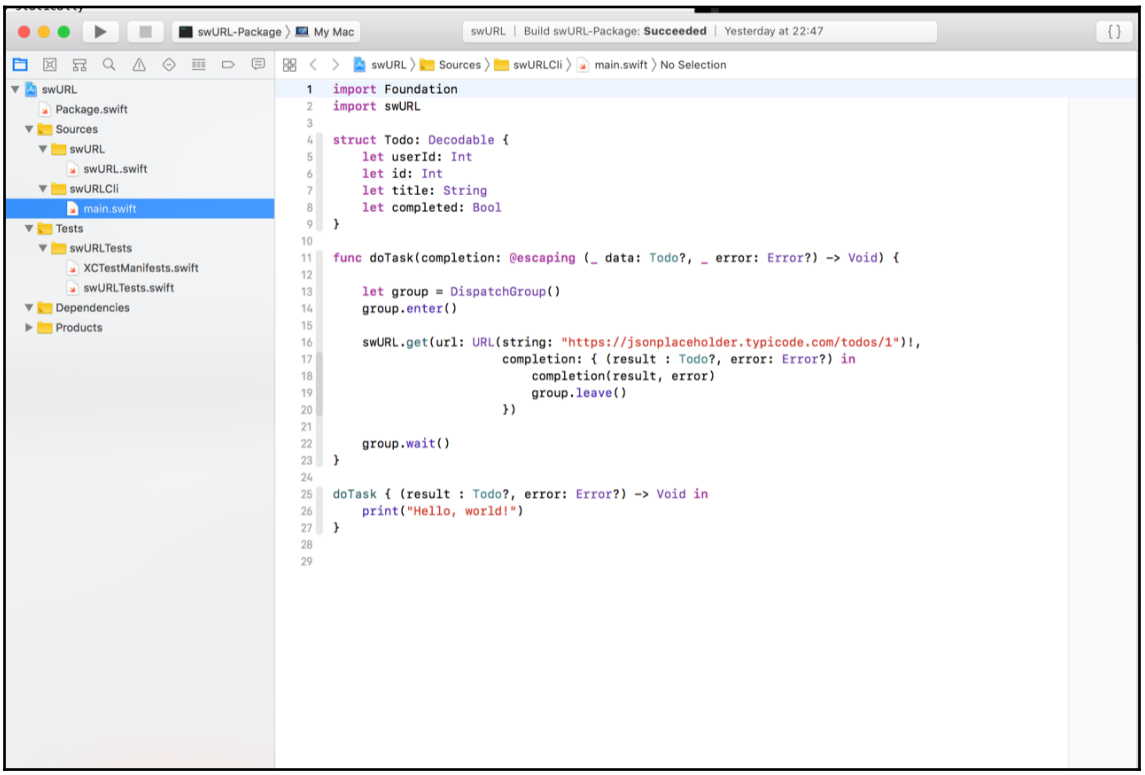
Status bar style	String	UIStatusBarStyleLightContent
▼ TyphoonInitialAssemblies	Array	(3 items)
Item 0	String	ApplicationAssembly
Item 1	String	CoreComponents
Item 2	String	ThemeAssembly
▶ App Transport Security Settings	Dictionary	(1 item)

Swinject

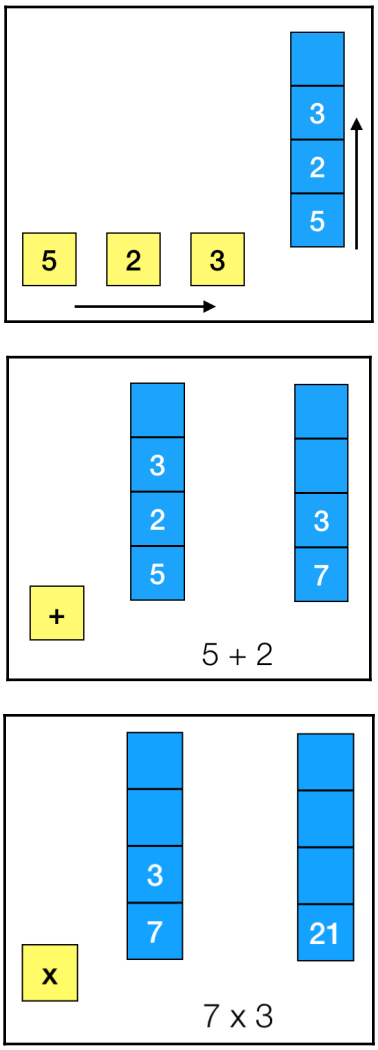
Chapter 12: Futures, Promises, and Reactive Programming

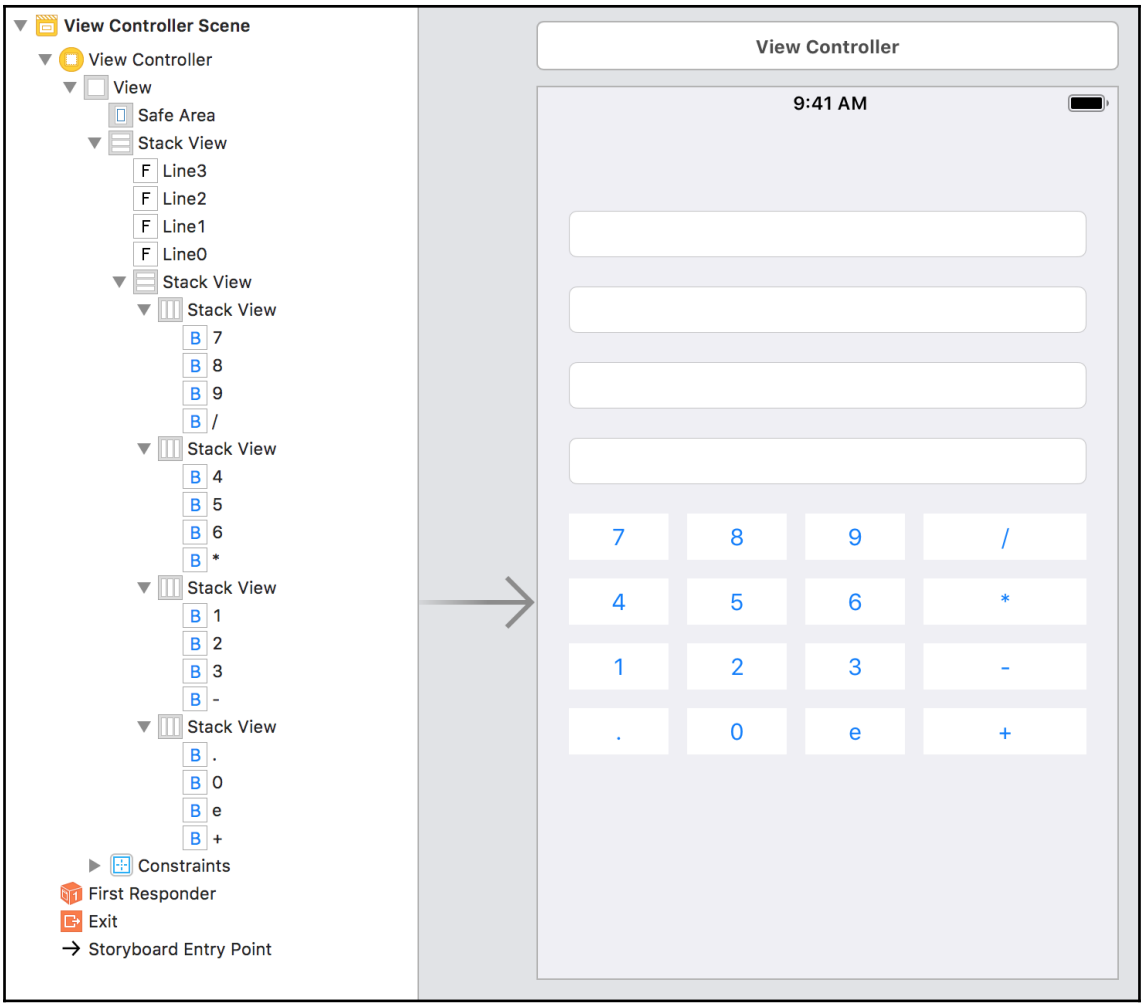


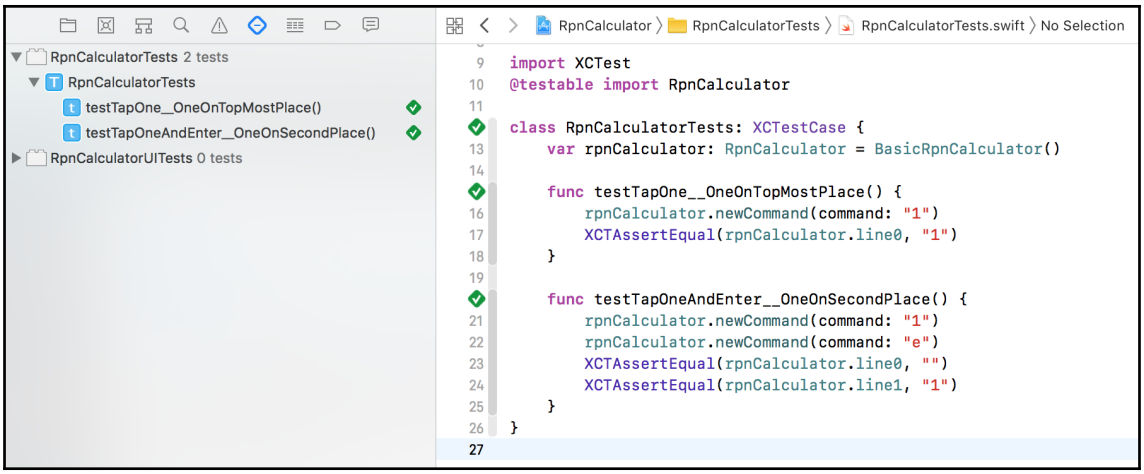
Chapter 13: Modularize Your Apps with Swift Package Manager



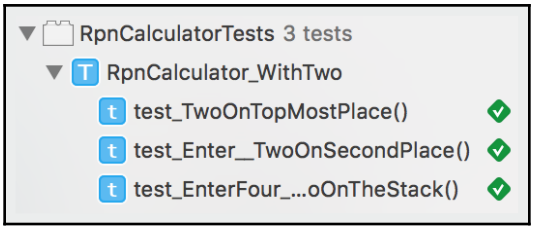
Chapter 14: Testing Your Code with Unit and UI Tests




View Controller Scene
View Controller
View
Safe Area
Stack View
Line3
Line2
Line1
Line0
Stack View
Stack View
7
8
9
/
Stack View
4
5
6
*
Stack View
1
2
3
-
Stack View
.
0
e
+
Constraints
First Responder
Exit
Storyboard Entry Point
View Controller
9:41 AM
7
8
9
/
4
5
6
*
1
2
3
-
.
0
e
+




```
func test_TwoEnterFour__FourTwoOnTheStack() {
    rpnCalculator.new(element: "2")
    rpnCalculator.new(element: "e")
    rpnCalculator.new(element: "4")
    XCTAssertEqual(rpnCalculator.line0, "4")  XCTAssertEqual failed: ("") is not equal to ("2") -
    XCTAssertEqual(rpnCalculator.line1, "2")
}
```



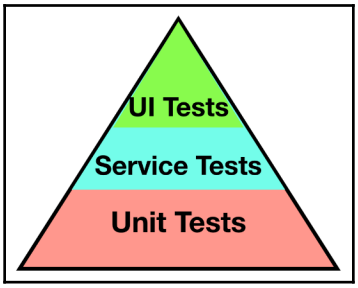

```
    func test__TwoEnterThreMultiply__SixOnFirstPlace() {
        assertThatEntering(elements: ["2", "e", "3", "*"],
                           resultShouldBeEqualTo: "6.00")
    }
}

extension CalculatorAsserting {
    func assertThatEntering(elements: [String],
                            resultShouldBeEqualTo result: String) {
        elements.forEach {
            rpnCalculator.new(element: $0)
        }
        XCTAssertEqual(rpnCalculator.line0, result)  XCTAssertEqual failed: ("3*") is not equal to ("6.00") -
    }
}
```

```
class MessagingTests: XCTestCase {
    func sum(_ a: Float, _ b: Float) -> Float {
        return a*b
    }

    func testSum() {
        XCTAssertEqual(sum(2, 3), 5, "2 + 3 must be equal to 5")
    }
}
```

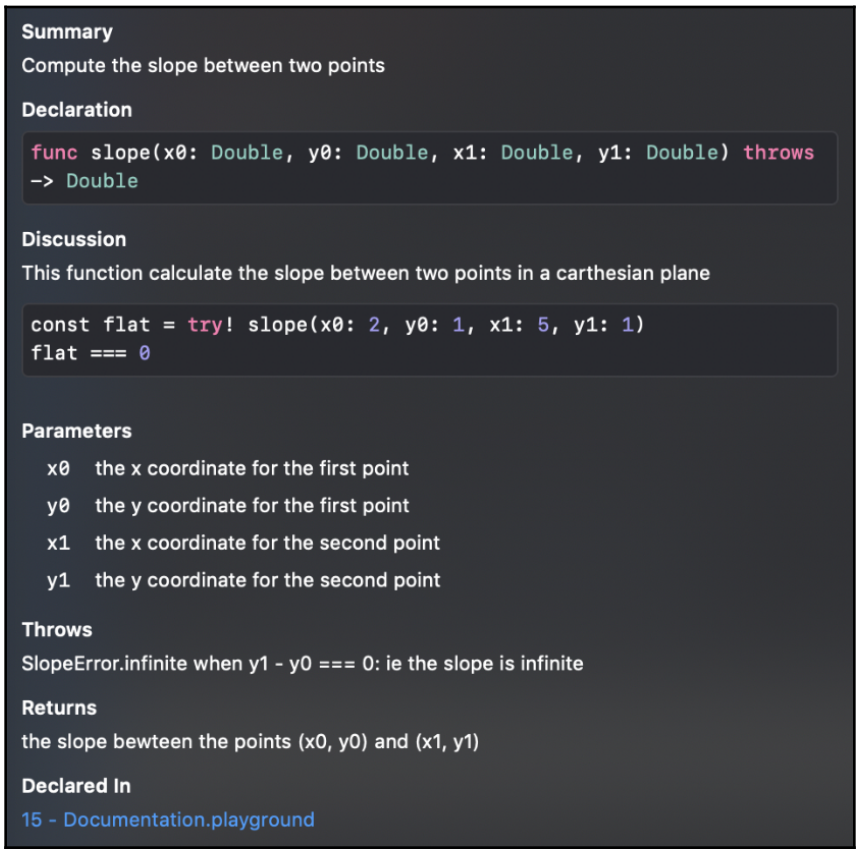
XCTAssertEqual failed: ("6.0") is not equal to ("5.0") - 2 + 3 must be equal to 5




```
class RpnCalculatorUITests: XCTestCase {
    override func setUp() {
        continueAfterFailure = false
        XCUIApplication().launch()
    }

    func testAddition() {
        // Use recording to get started writing UI tests.
        // Use XCTAssert and related functions to verify your tests produce the correct results.
    }
}
```

Chapter 15: Going Out in the Open (Source)



Summary

Compute the slope between two points

Declaration

```
func slope(x0: Double, y0: Double, x1: Double, y1: Double) throws
-> Double
```

Discussion

This function calculate the slope between two points in a carthesian plane

```
const flat = try! slope(x0: 2, y0: 1, x1: 5, y1: 1)
flat === 0
```

Parameters

- `x0` the x coordinate for the first point
- `y0` the y coordinate for the first point
- `x1` the x coordinate for the second point
- `y1` the y coordinate for the second point

Throws

SlopeError.infinite when y1 - y0 === 0: ie the slope is infinite

Returns

the slope bewteen the points (x0, y0) and (x1, y1)

Declared In

15 - Documentation.playground

The **rich** capabilities of markdown help your render text *nicely*. You can mention code inline with the single backtick.