TDD

(Test Driven Development)

iOS D조 쌍쌍바

TDD란?

Test Driven Development

테스트 주도 개발이라는 개발 방법론

TDD를 하는 이유

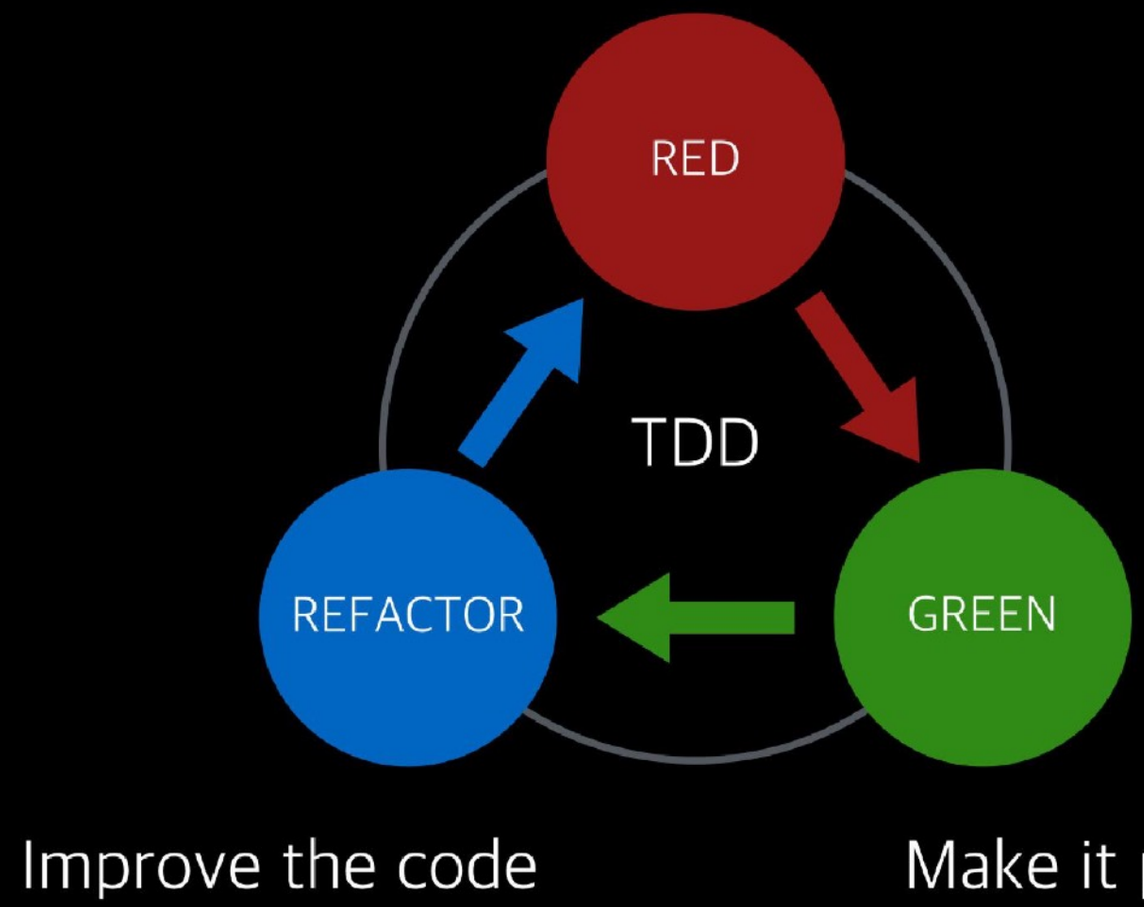
소프트웨어 개발 시 작은 부분이 수정될 경우
관련된 모든 부분에 있어서 테스트가 다시 이루어져야 한다.

이전에 정상적으로 동작 하던 기능이 기능 추가 후에도
여전히 정상적으로 동작할 것이라고 감히 예상할 수 없기 때문이다.

TDD는 이러한 문제를 테스트 코드 작성을
통해 해결하는 방법을 제시한다.

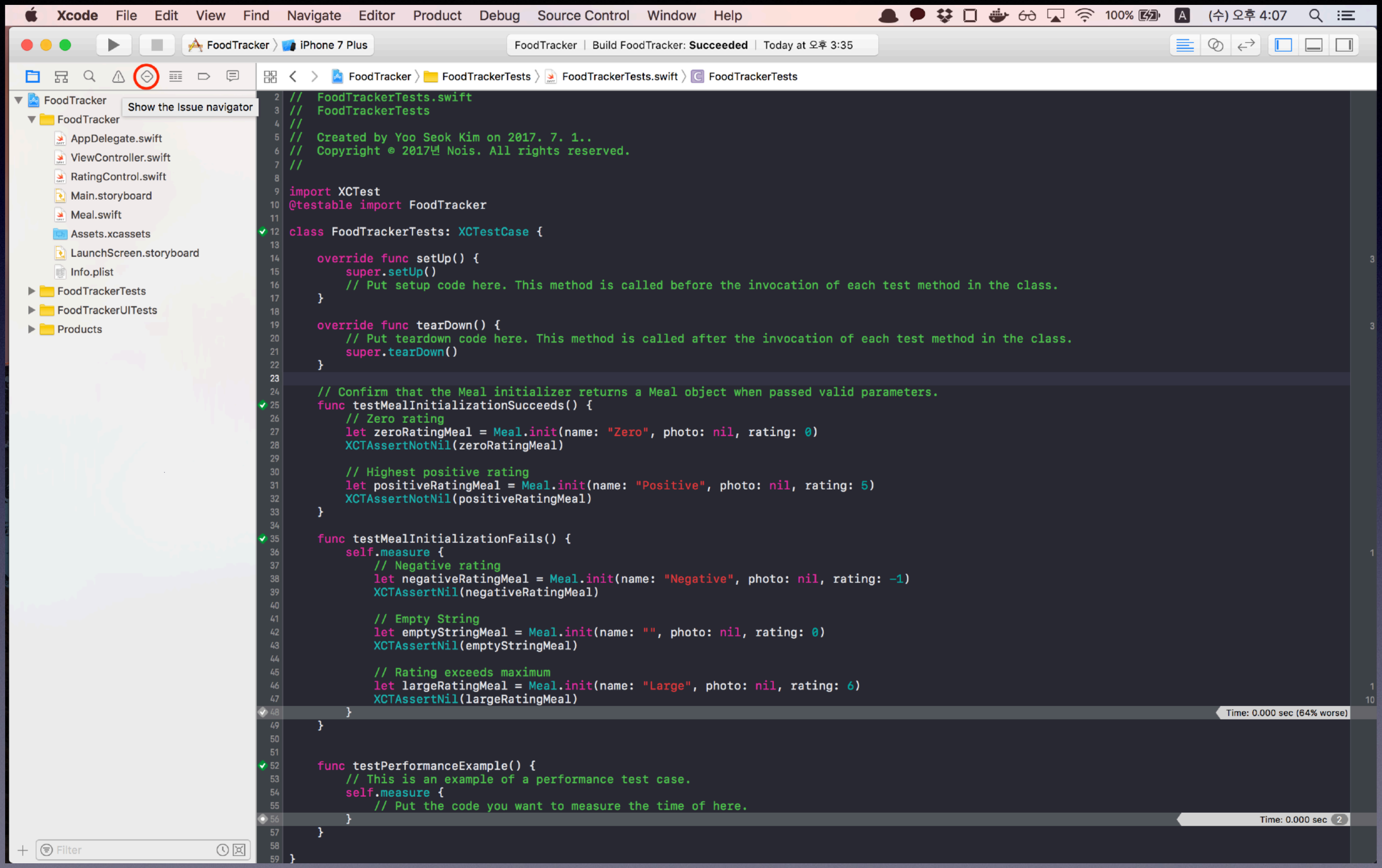
TDD란?

THE THREE LAWS OF TDD

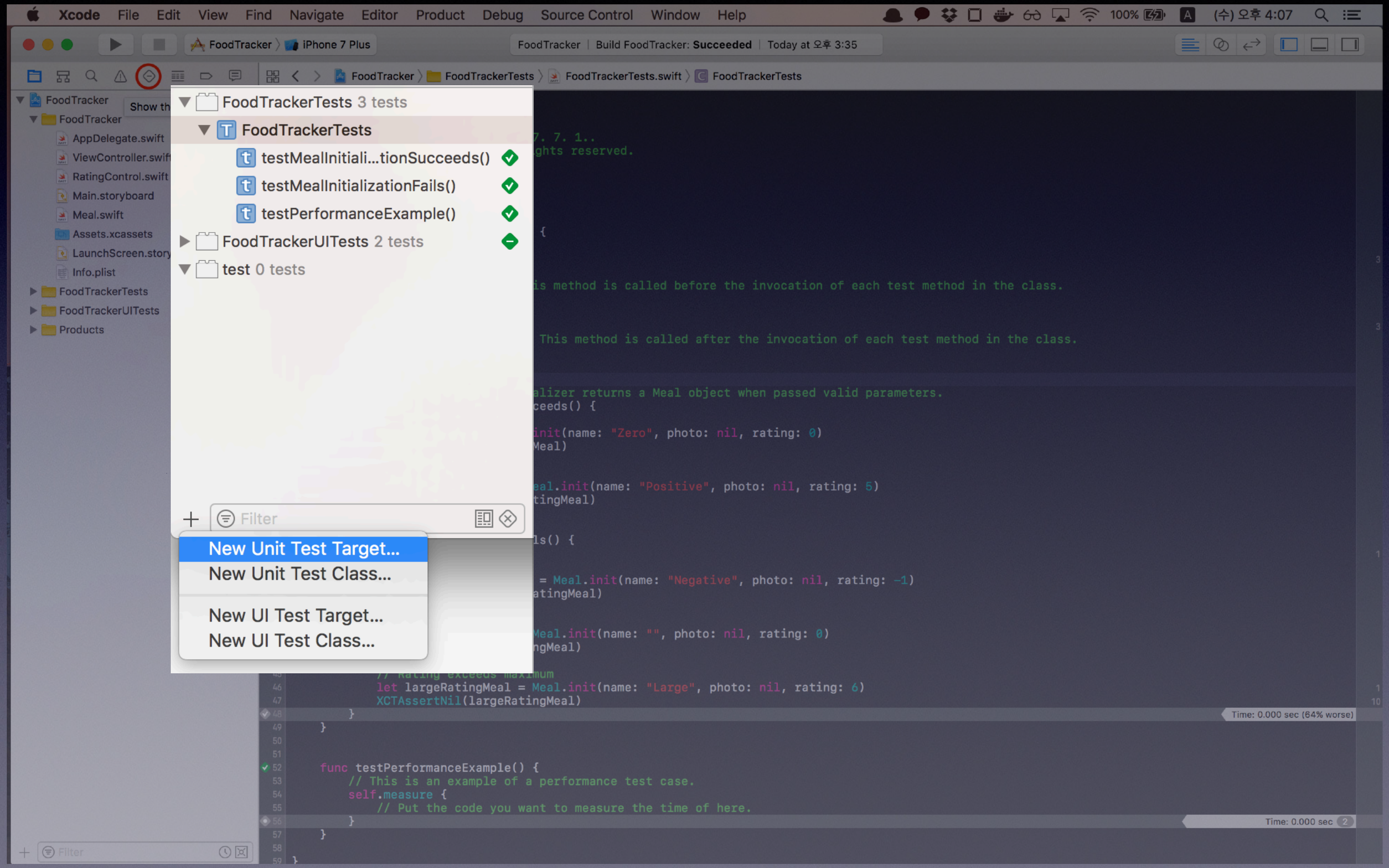


1. 실패하는 테스트 코드 작성
2. 성공하는 테스트 코드 작성
3. 작성한 코드 리팩토링

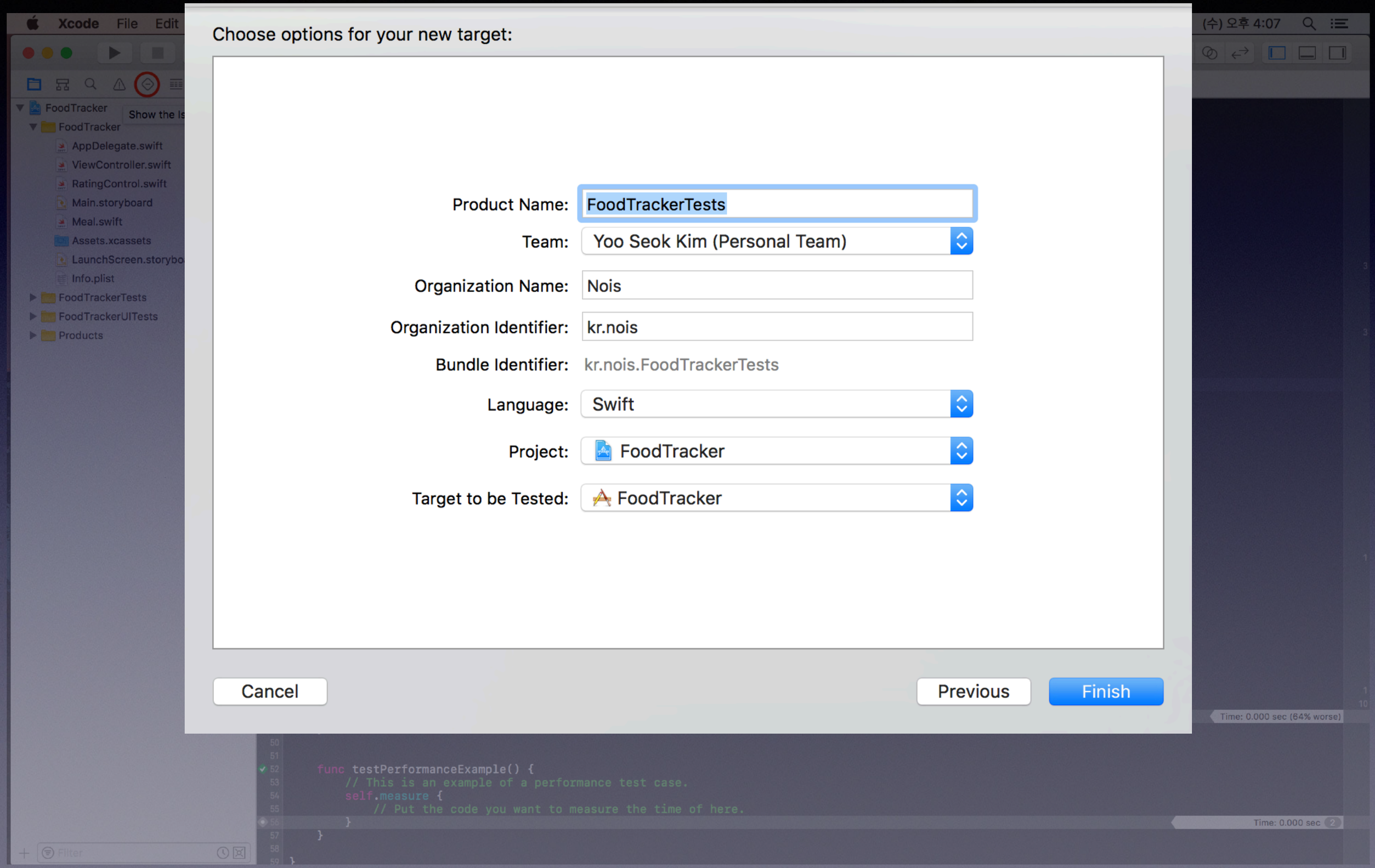
TDD Unit Test 적용해보기



TDD Unit Test 적용해보기



TDD Unit Test 적용해보기



TDD Unit Test 적용해보기

```swift
import XCTest

class Tests: XCTestCase {

    override func setUp() {
        super.setUp()
        // Put setup code here. This method is called before the invocation of each test method in the class.
    }

    override func tearDown() {
        // Put teardown code here. This method is called after the invocation of each test method in the class.
        super.tearDown()
    }

    func testExample() {
        // This is an example of a functional test case.
        // Use XCTAssert and related functions to verify your tests produce the correct results.
    }

    func testPerformanceExample() {
        // This is an example of a performance test case.
        self.measure {
            // Put the code you want to measure the time of here.
        }
    }

}
```

TDD Unit Test 적용해보기

```
import XCTest

class Tests: XCTestCase {

    override func setUp() {
        super.setUp()
        // Put setup code here. This method is called before the invocation of each test method in the class.
    }

    override func tearDown() {
        // Put teardown code here. This method is called after the invocation of each test method in the class.
        super.tearDown()
    }

    func testExample() {
        // This is an example of a functional test case.
        // Use XCTAssert and related functions to verify your tests produce the correct results.
    }

    func testPerformanceExample() {
        // This is an example of a performance test case.
        self.measure {
            // Put the code you want to measure the time of here.
        }
    }

}
```

TDD Unit Test 적용해보기

```
import XCTest

class Tests: XCTestCase {

    override func setUp() {
        super.setUp()
        // Put setup code here. This method is called before the invocation of each test method in the class.
    }

    override func tearDown() {
        // Put teardown code here. This method is called after the invocation of each test method in the class.
        super.tearDown()
    }

    func testExample() {
        // This is an example of a functional test case.
        // Use XCTAssert and related functions to verify your tests produce the correct results.
    }

    func testPerformanceExample() {
        // This is an example of a performance test case.
        self.measure {
            // Put the code you want to measure the time of here.
        }
    }

}
```

TDD Unit Test 적용해보기

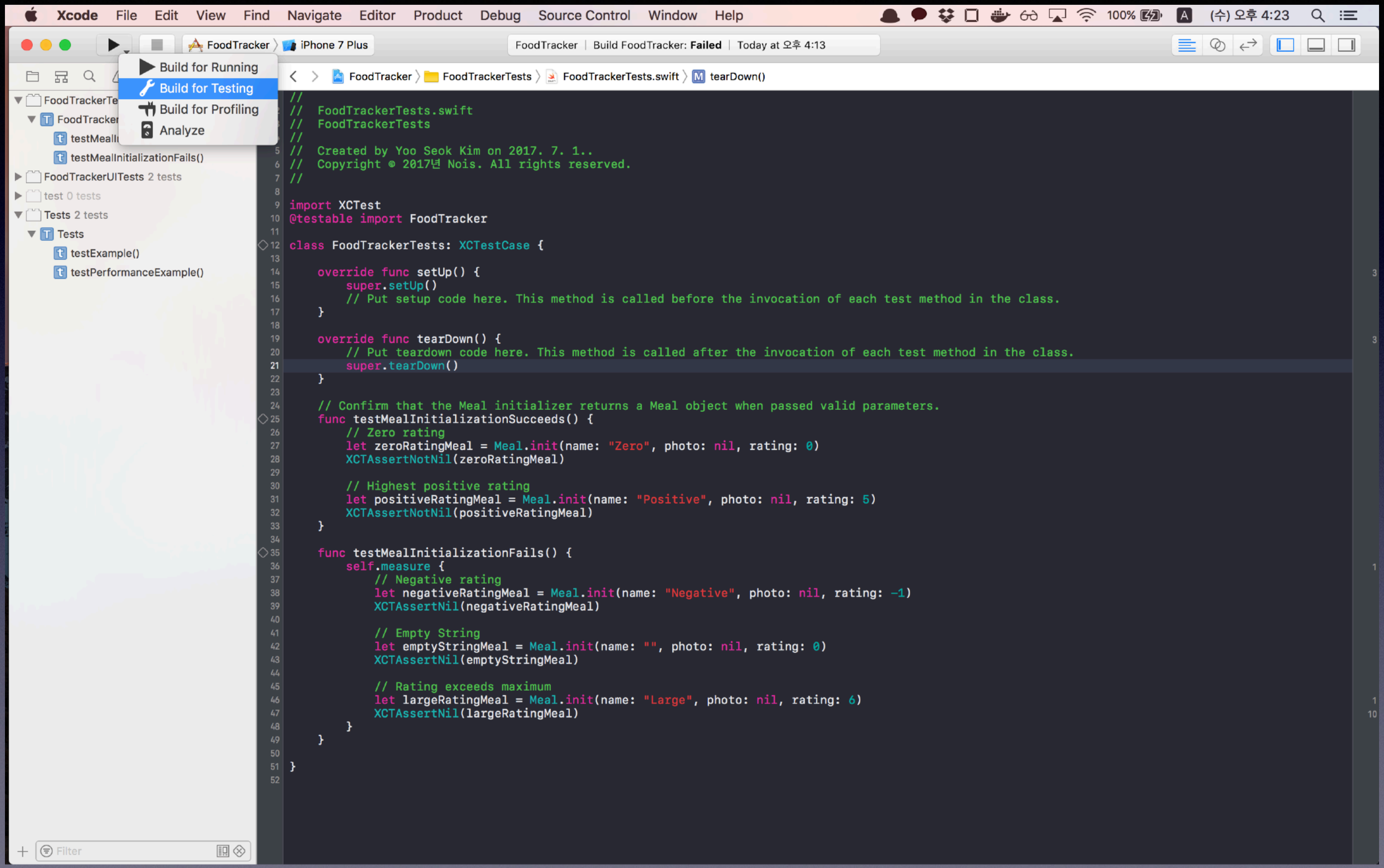
```swift
import UIKit

class Meal {

    //MARK: Properties

    var name: String
    var photo: UIImage?
    var rating: Int

    //MARK: Initialization

    init?(name: String, photo: UIImage?, rating: Int) {
        // Initialization should fail if there is no name or if the rating is negative.
        if name.isEmpty || rating < 0  {
            return nil
        }

        // Initialize stored properties.
        self.name = name
        self.photo = photo
        self.rating = rating
    }
}
```

TDD Unit Test 적용해보기

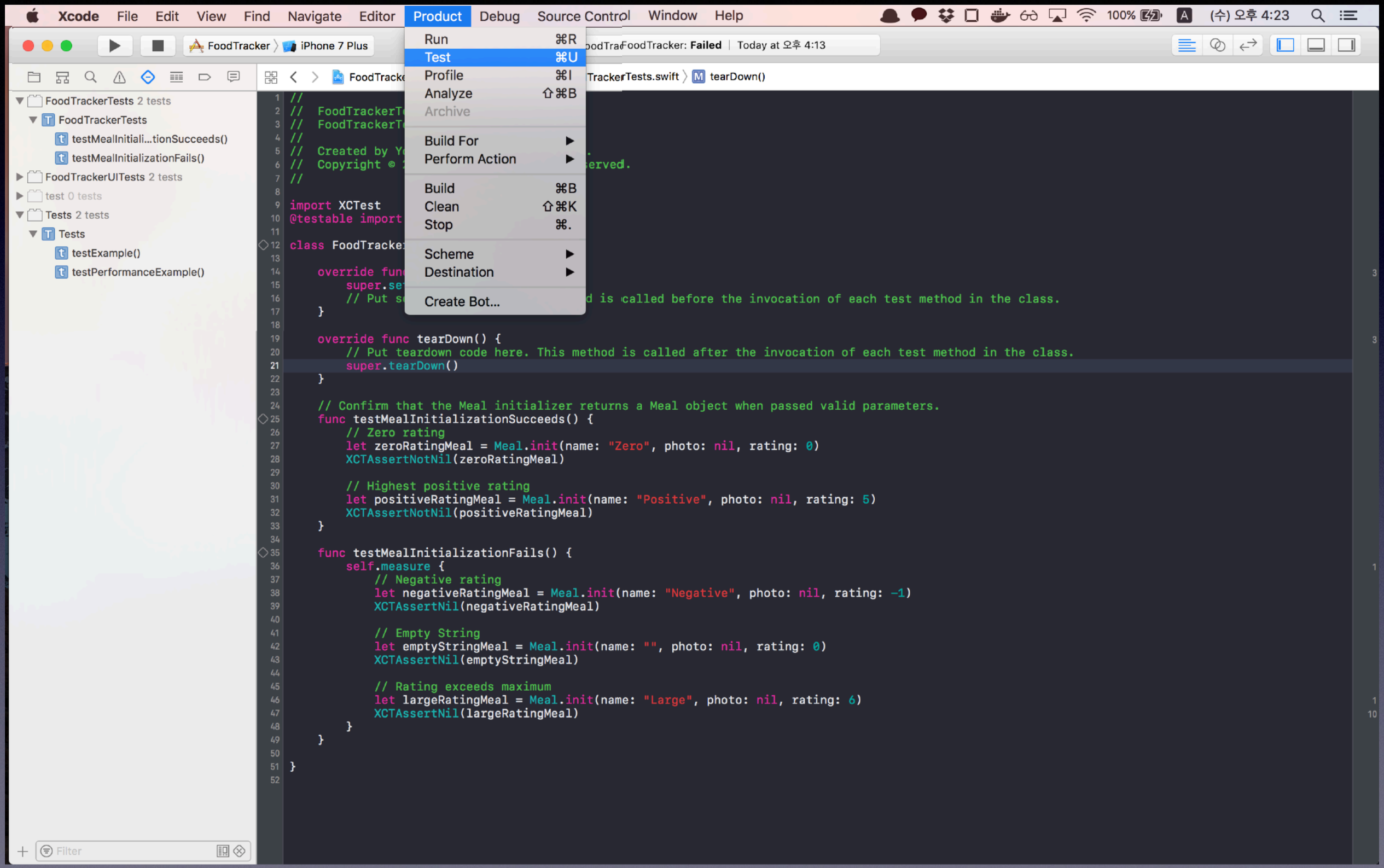
```swift
// Confirm that the Meal initializer returns a Meal object when passed valid parameters.
func testMealInitializationSucceeds() {
    // Zero rating
    let zeroRatingMeal = Meal.init(name: "Zero", photo: nil, rating: 0)
    XCTAssertNotNil(zeroRatingMeal)

    // Highest positive rating
    let positiveRatingMeal = Meal.init(name: "Positive", photo: nil, rating: 5)
    XCTAssertNotNil(positiveRatingMeal)
}
```

```swift
func testMealInitializationFails() {
    self.measure {
        // Negative rating
        let negativeRatingMeal = Meal.init(name: "Negative", photo: nil, rating: -1)
        XCTAssertNil(negativeRatingMeal)

        // Empty String
        let emptyStringMeal = Meal.init(name: "", photo: nil, rating: 0)
        XCTAssertNil(emptyStringMeal)

        // Rating exceeds maximum
        let largeRatingMeal = Meal.init(name: "Large", photo: nil, rating: 6)
        XCTAssertNil(largeRatingMeal)
    }
}
```

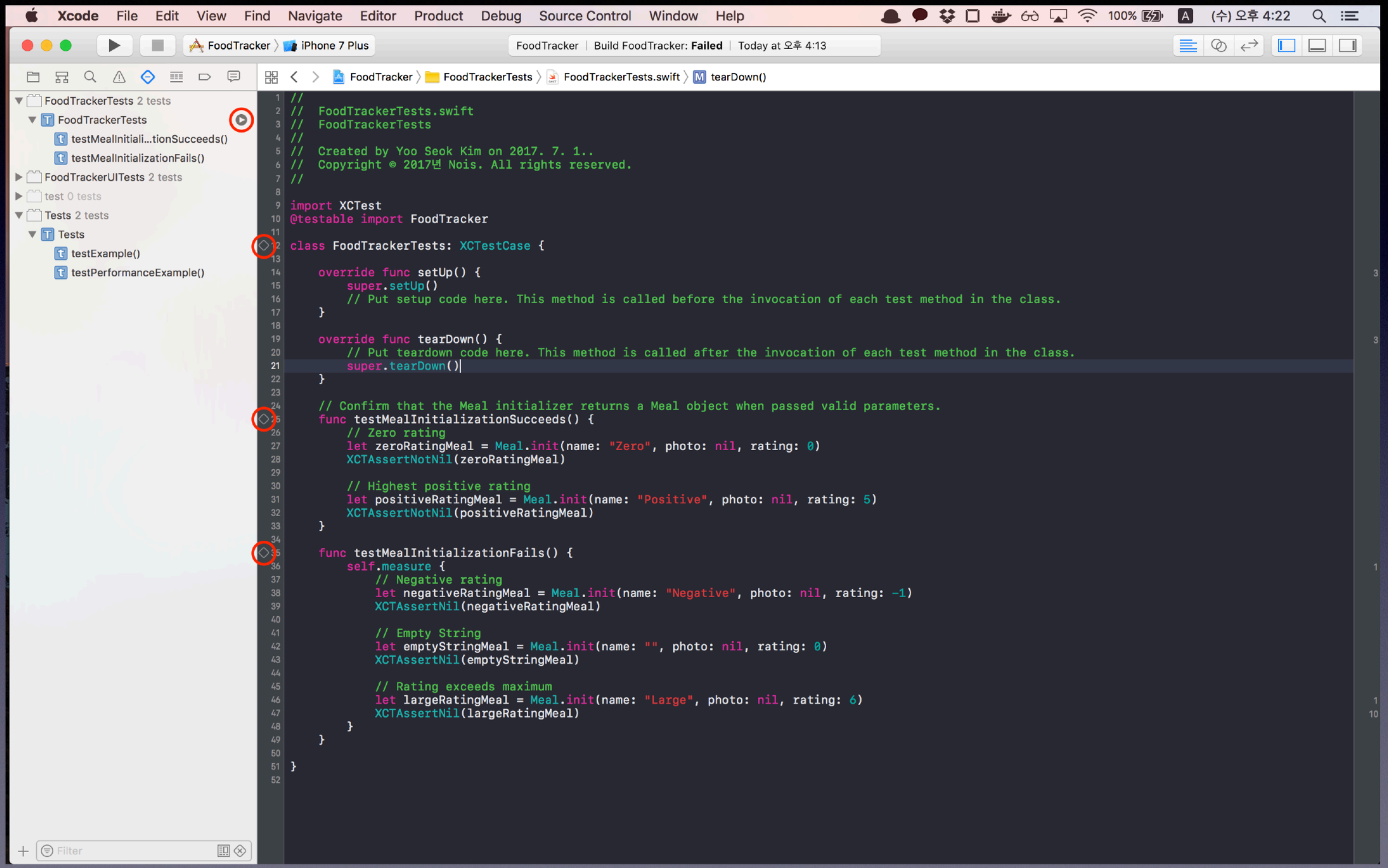
TDD Unit Test 적용해보기



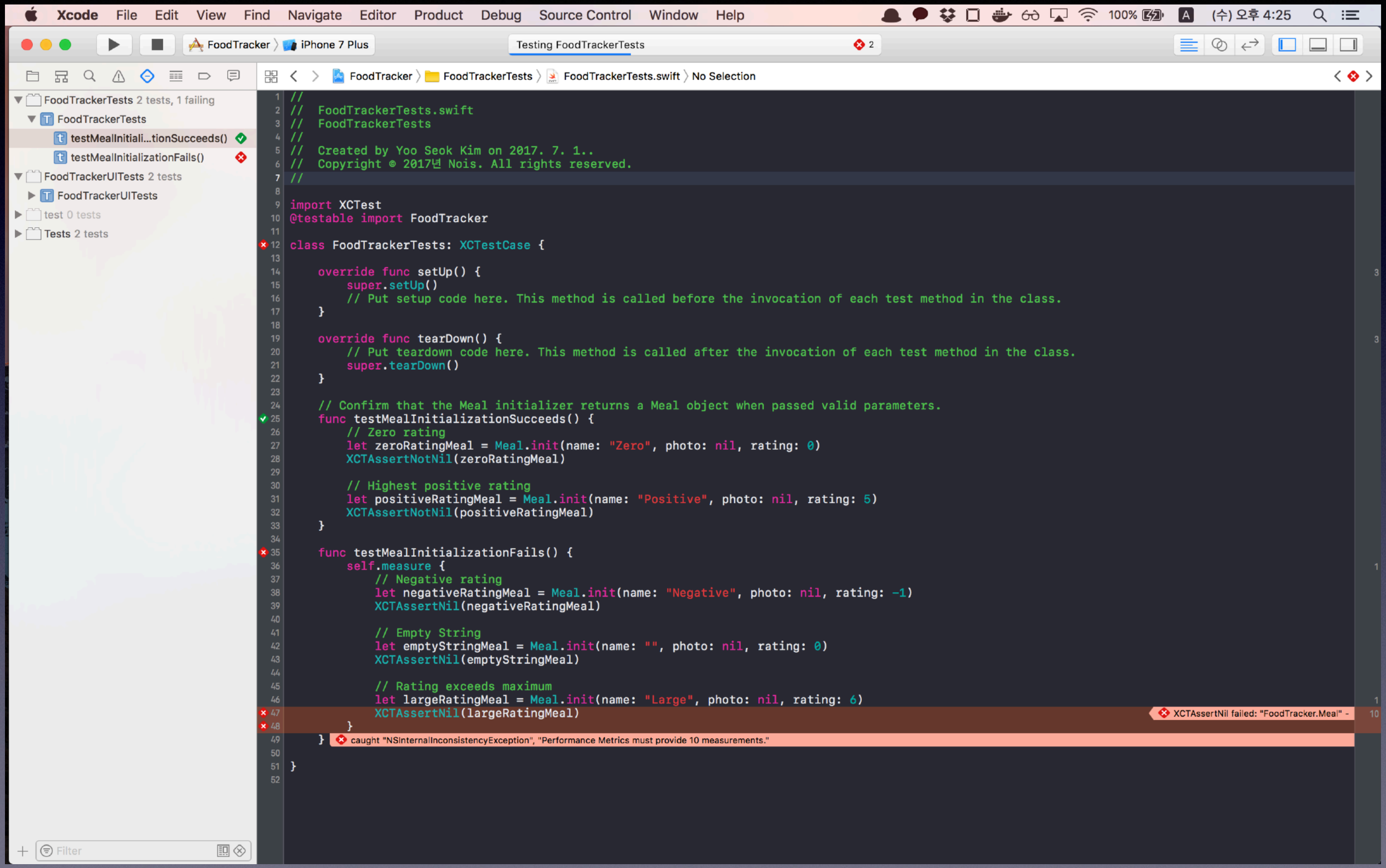
TDD Unit Test 적용해보기



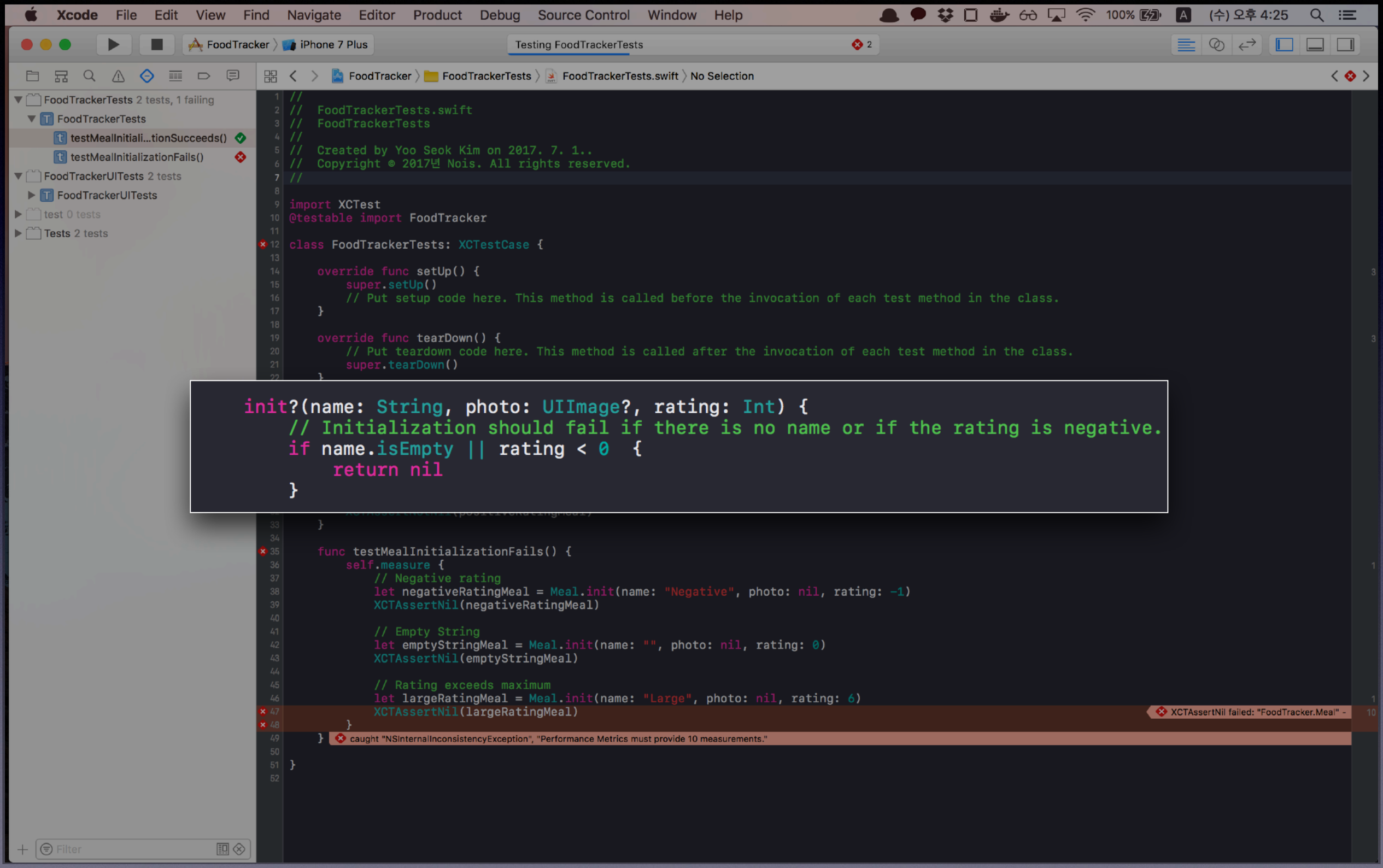
TDD Unit Test 적용해보기



TDD Unit Test 적용해보기



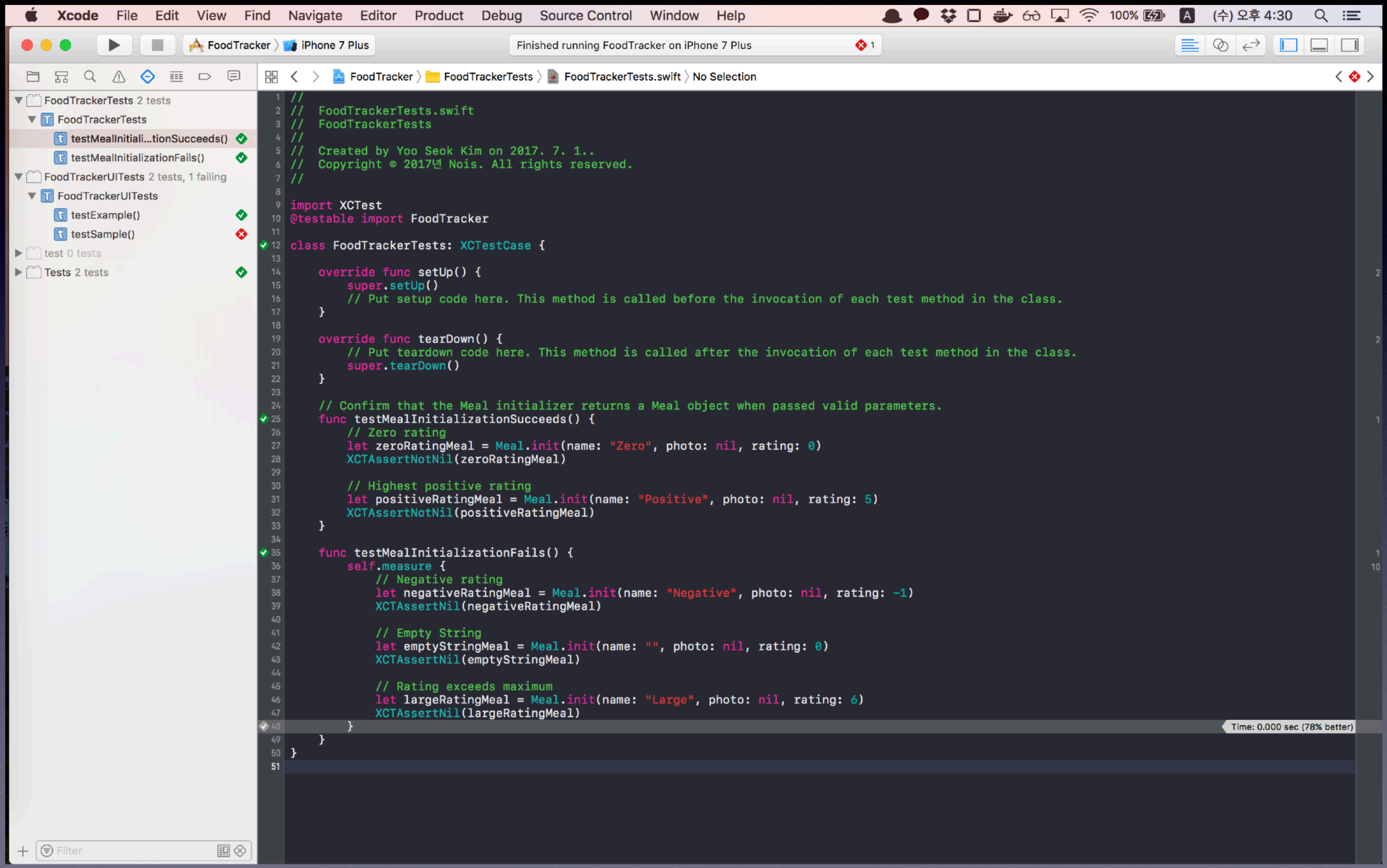
TDD Unit Test 적용해보기



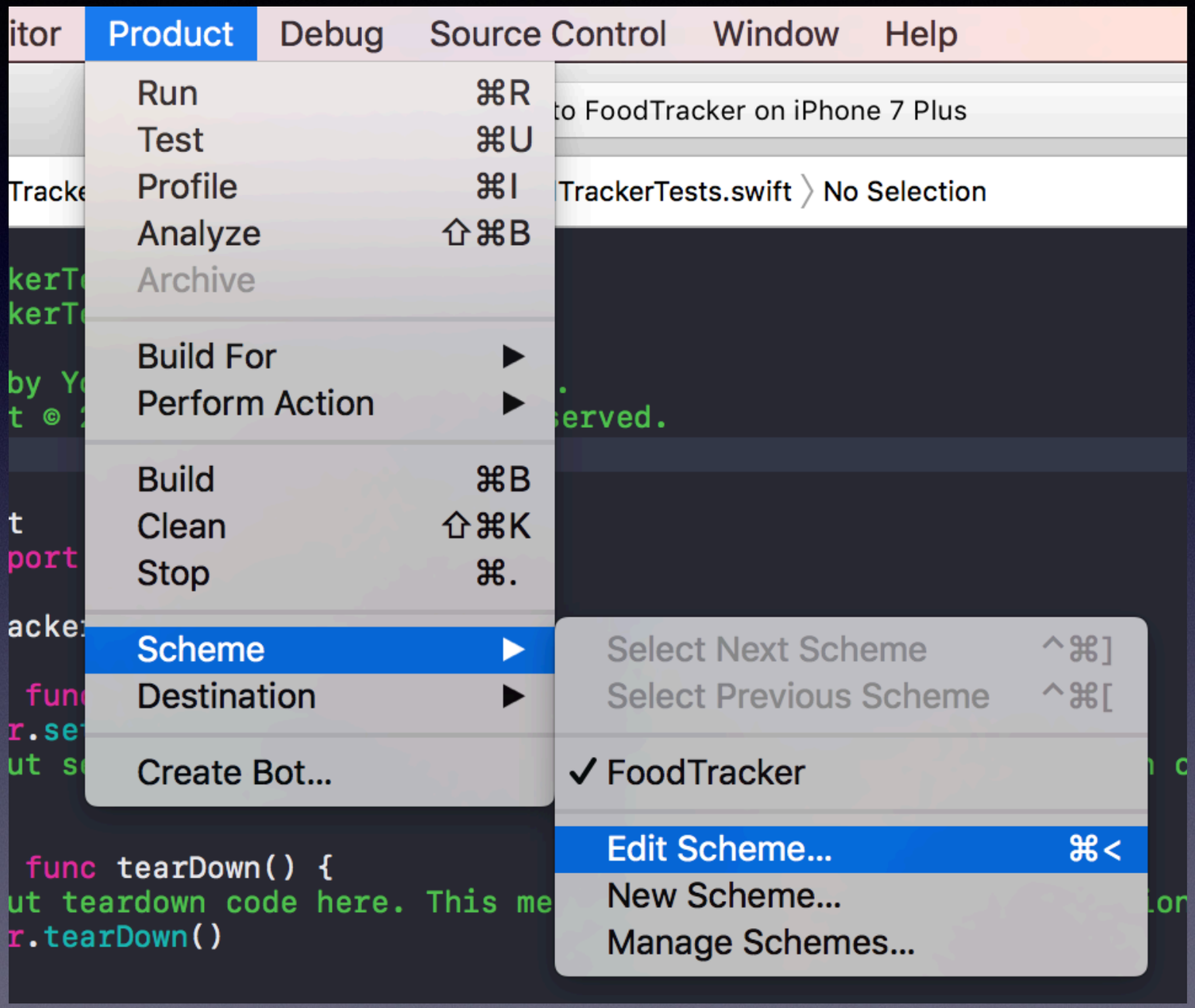
TDD Unit Test 적용해보기

```swift
import UIKit

class Meal {

    //MARK: Properties

    var name: String
    var photo: UIImage?
    var rating: Int

    //MARK: Initialization

    init?(name: String, photo: UIImage?, rating: Int) {
        // The name must not be empty
        guard !name.isEmpty else {
            return nil
        }

        // The rating must be between 0 and 5 inclusively
        guard (rating >= 0) && (rating <= 5) else {
            return nil
        }

        // Initialize stored properties.
        self.name = name
        self.photo = photo
        self.rating = rating
    }
}
```

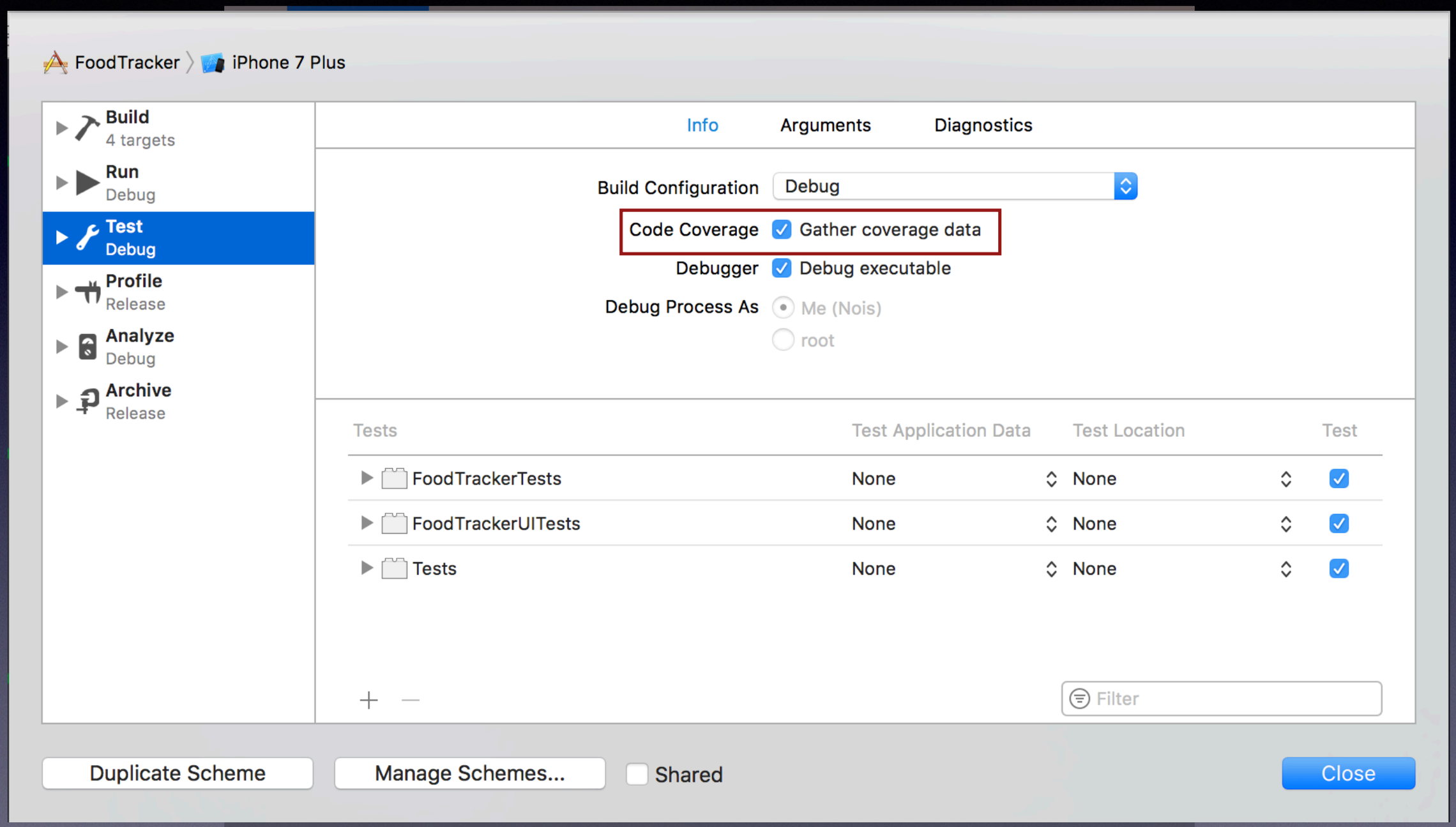
TDD Unit Test 적용해보기



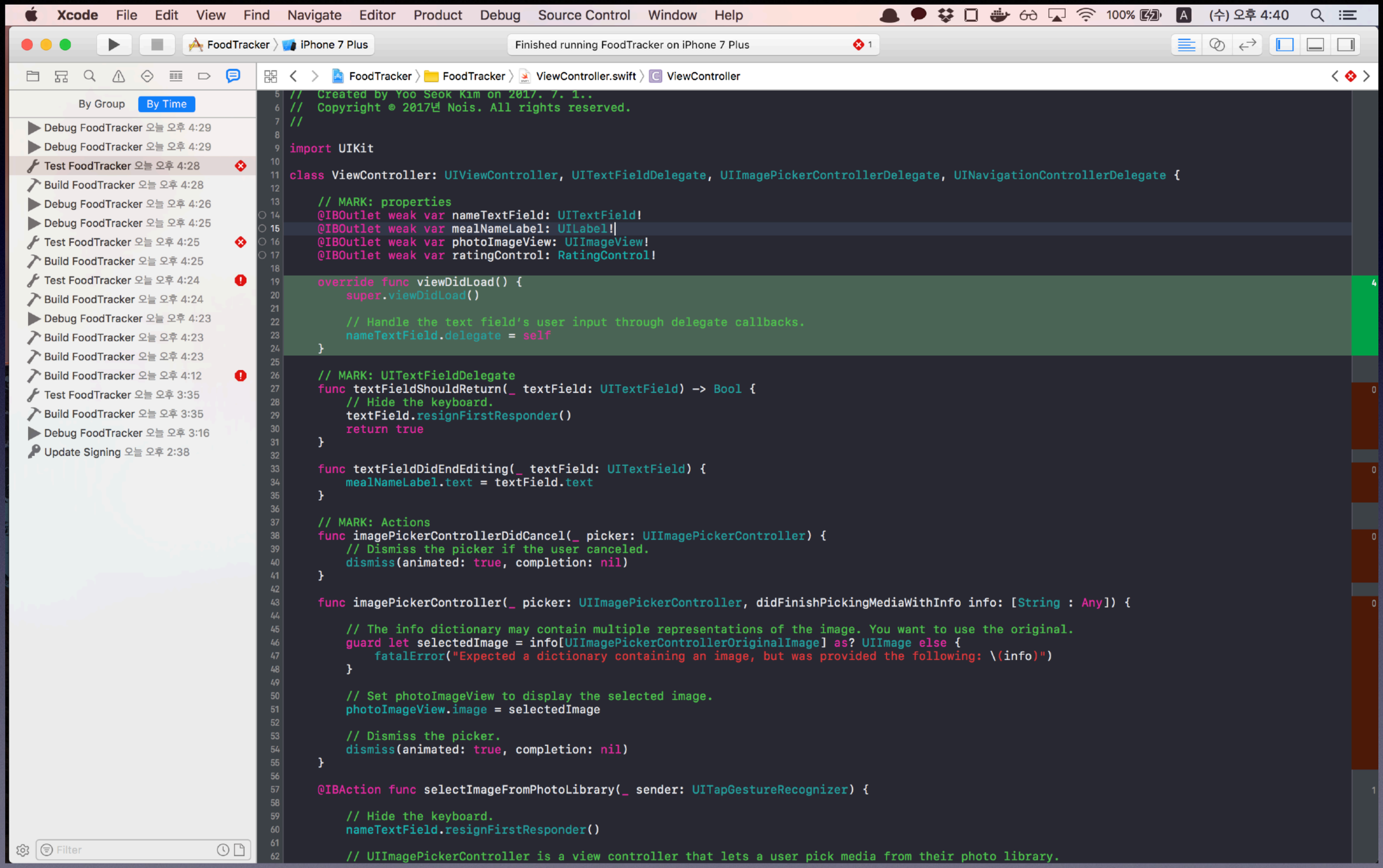
TDD Unit Test 적용해보기



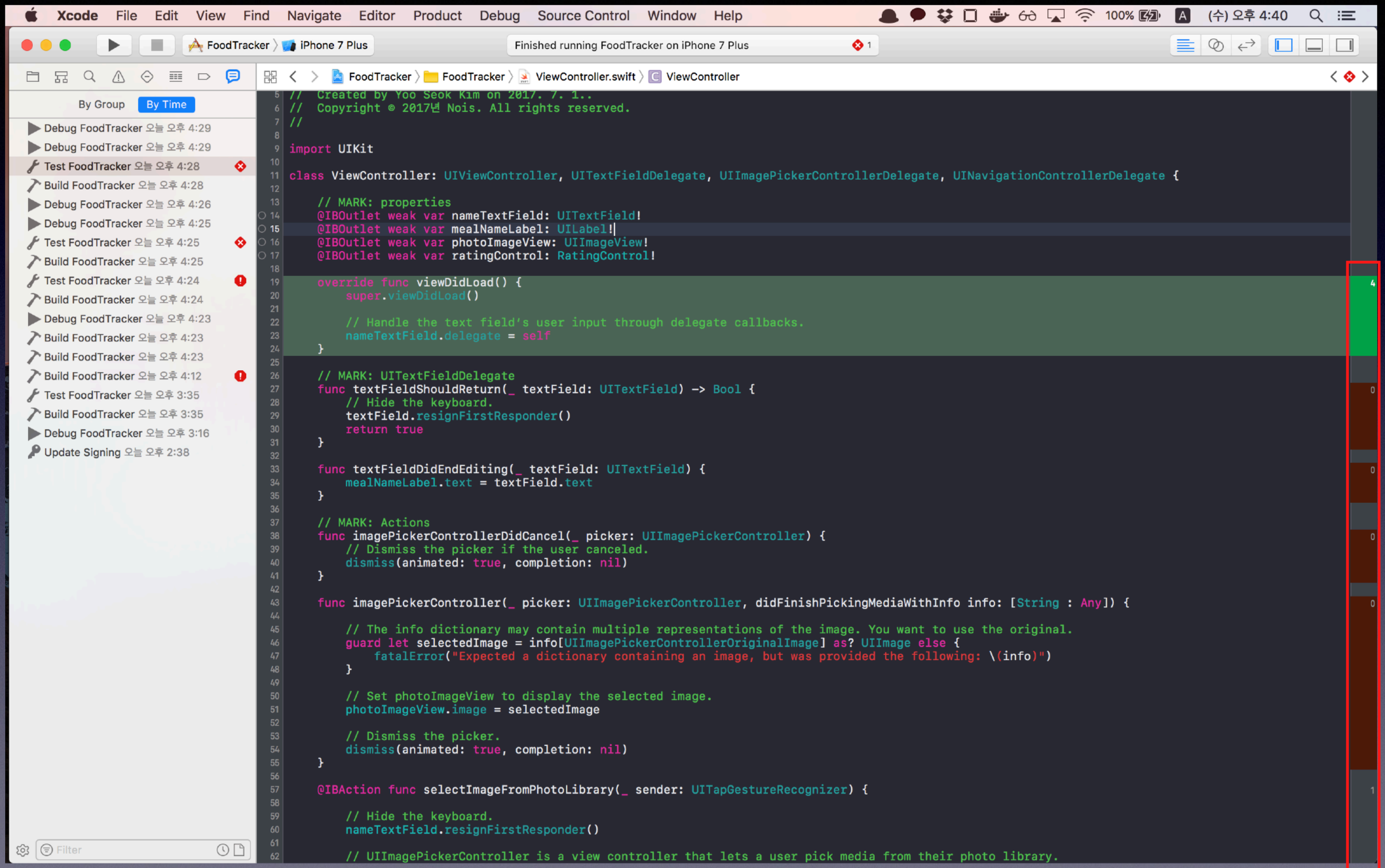
TDD Unit Test 적용해보기



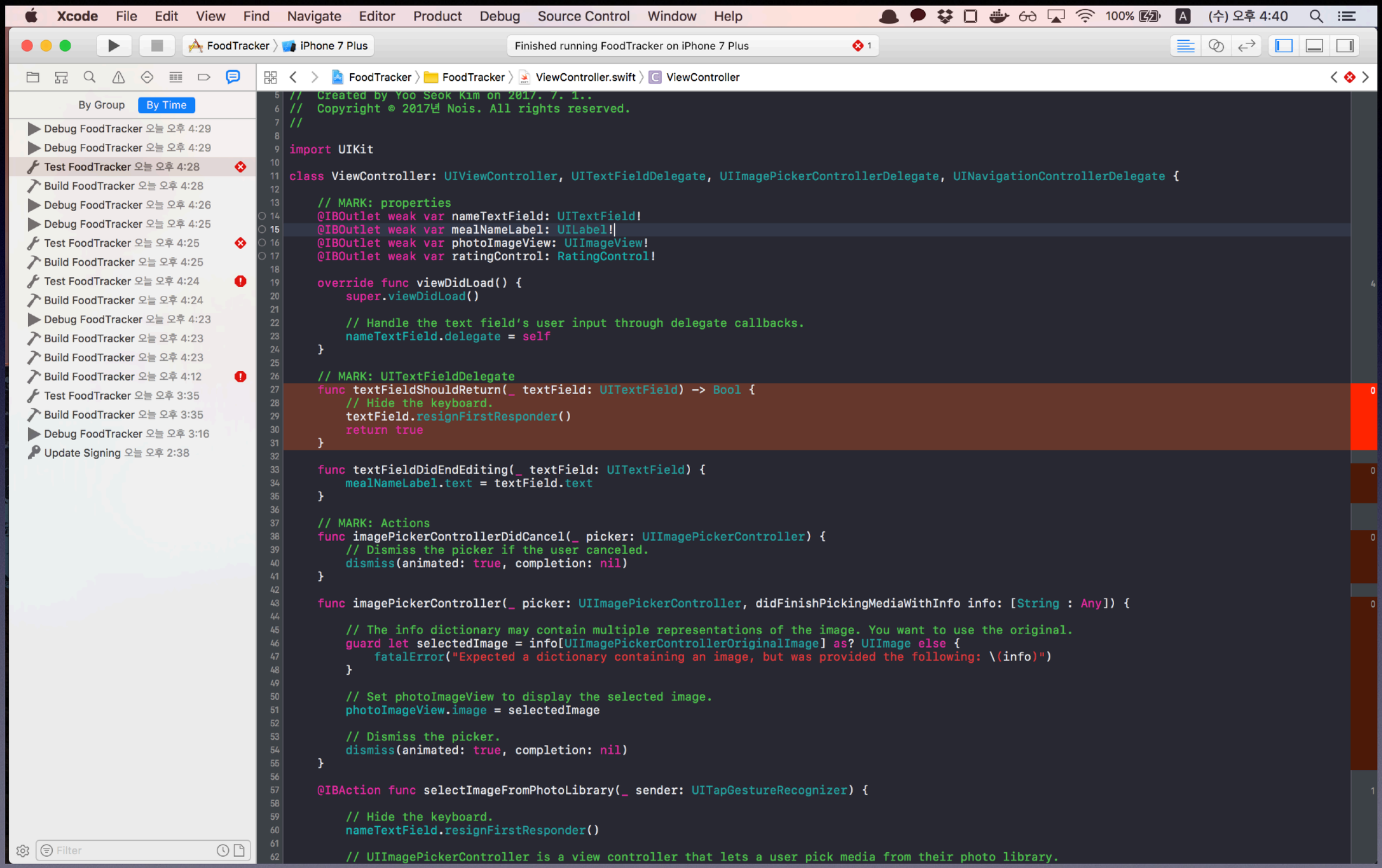
TDD Unit Test 적용해보기



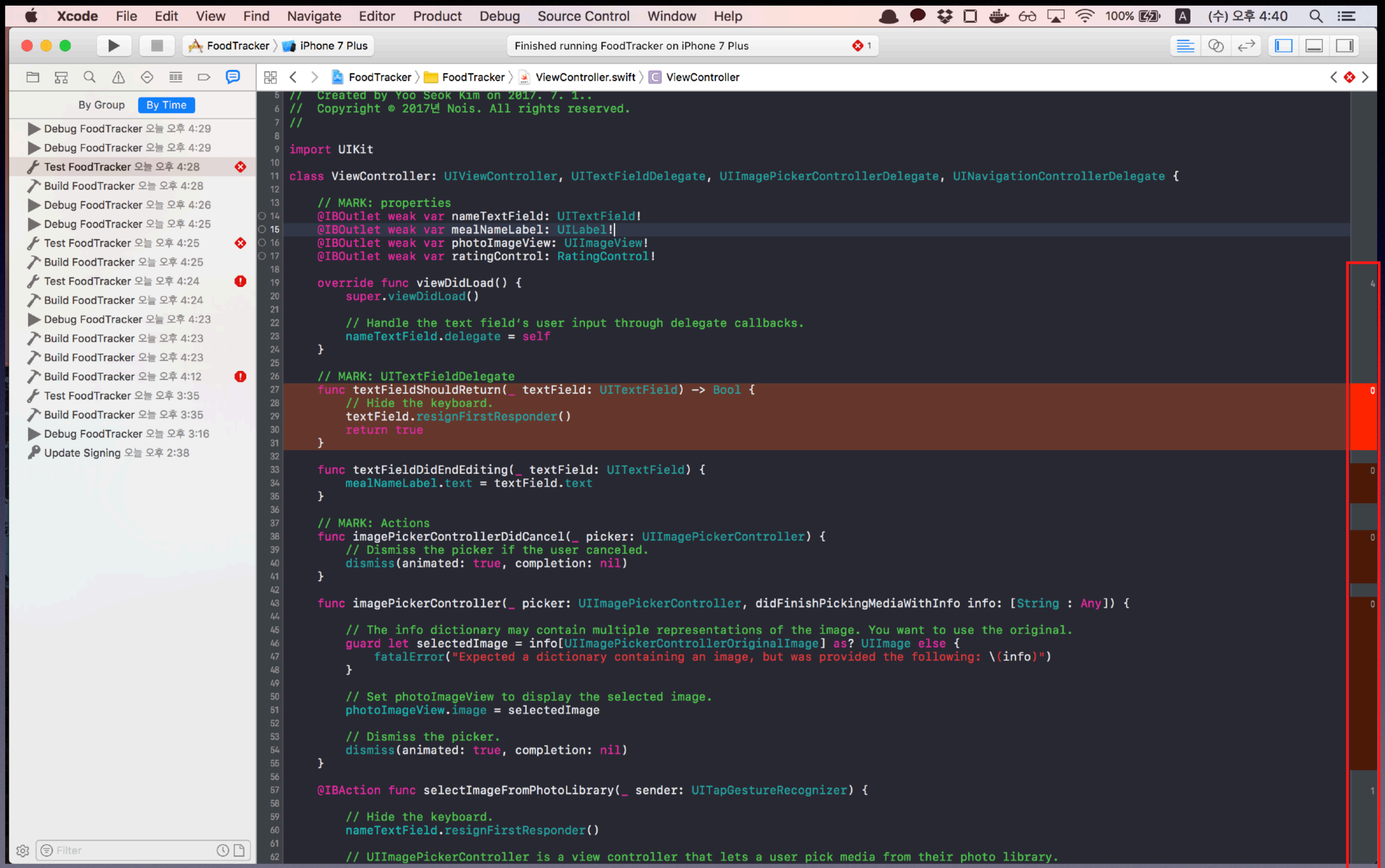
TDD Unit Test 적용해보기



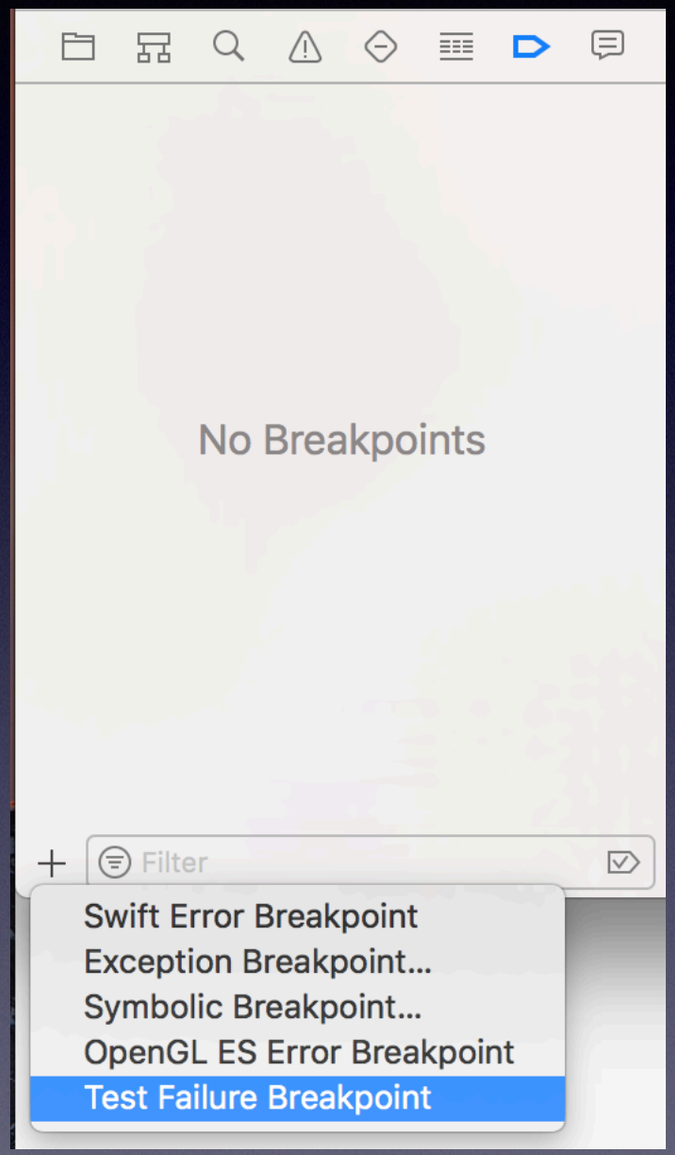
TDD Unit Test 적용해보기



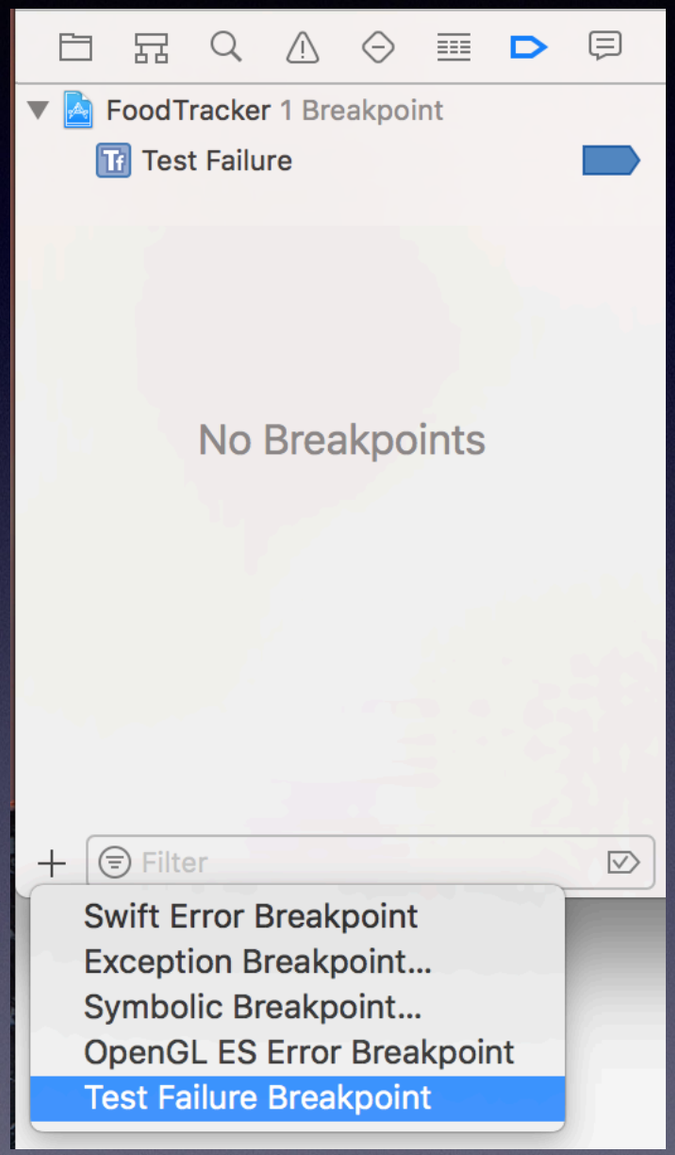
TDD Unit Test 적용해보기



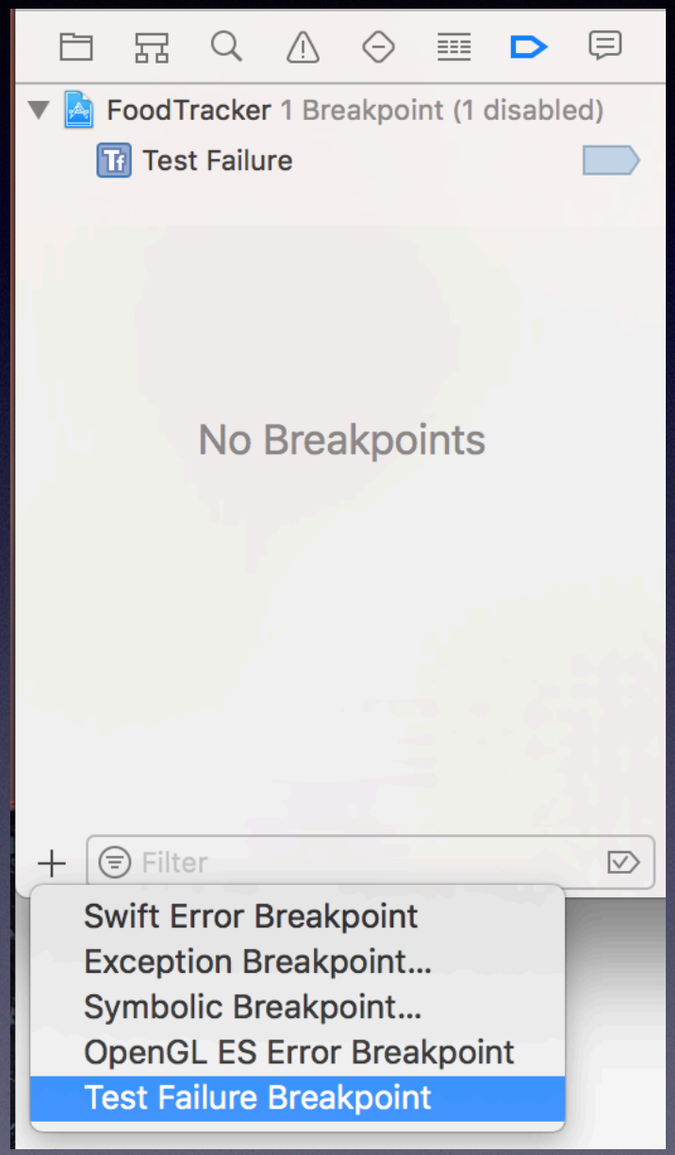
TDD Unit Test 적용해보기



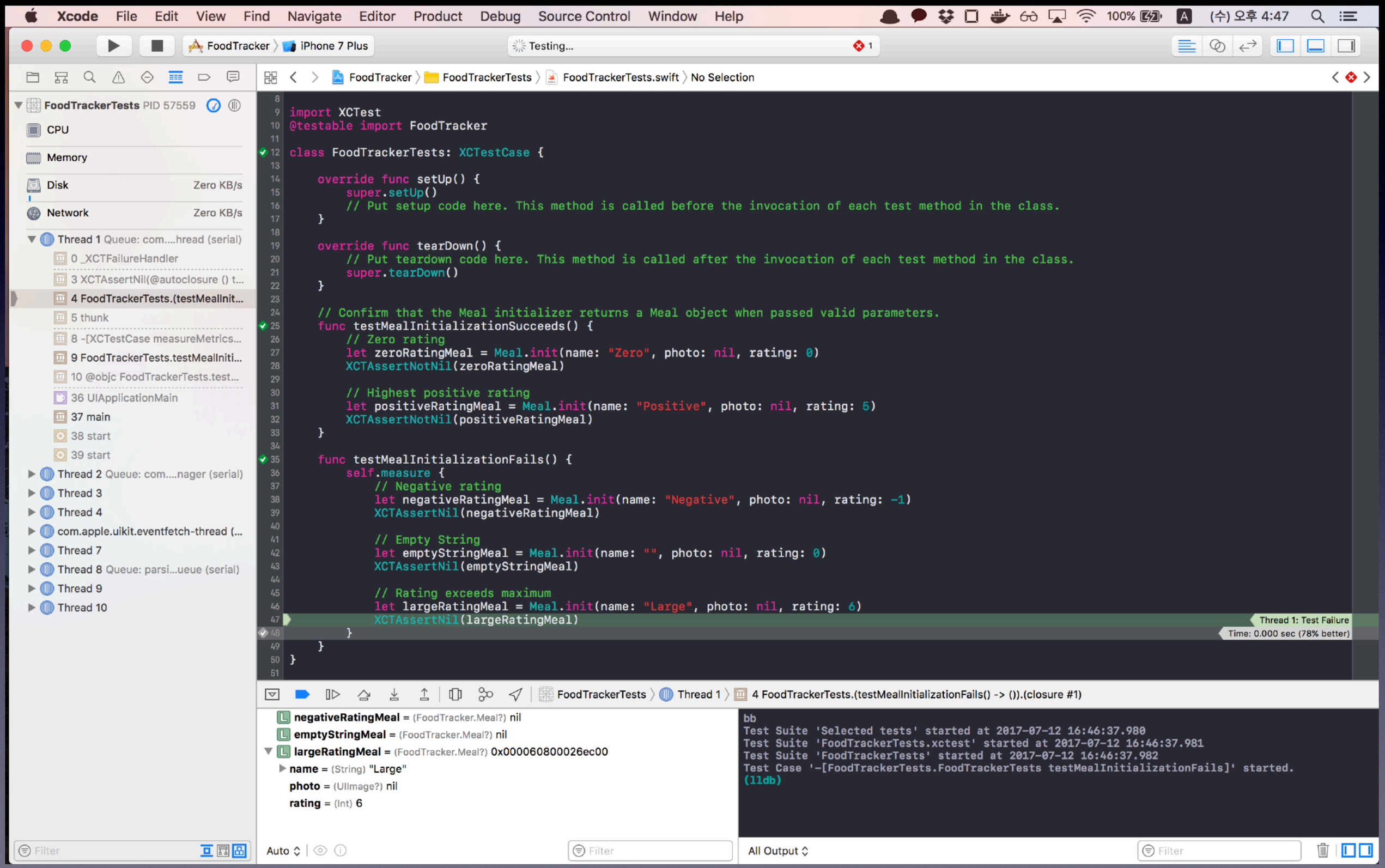
TDD Unit Test 적용해보기



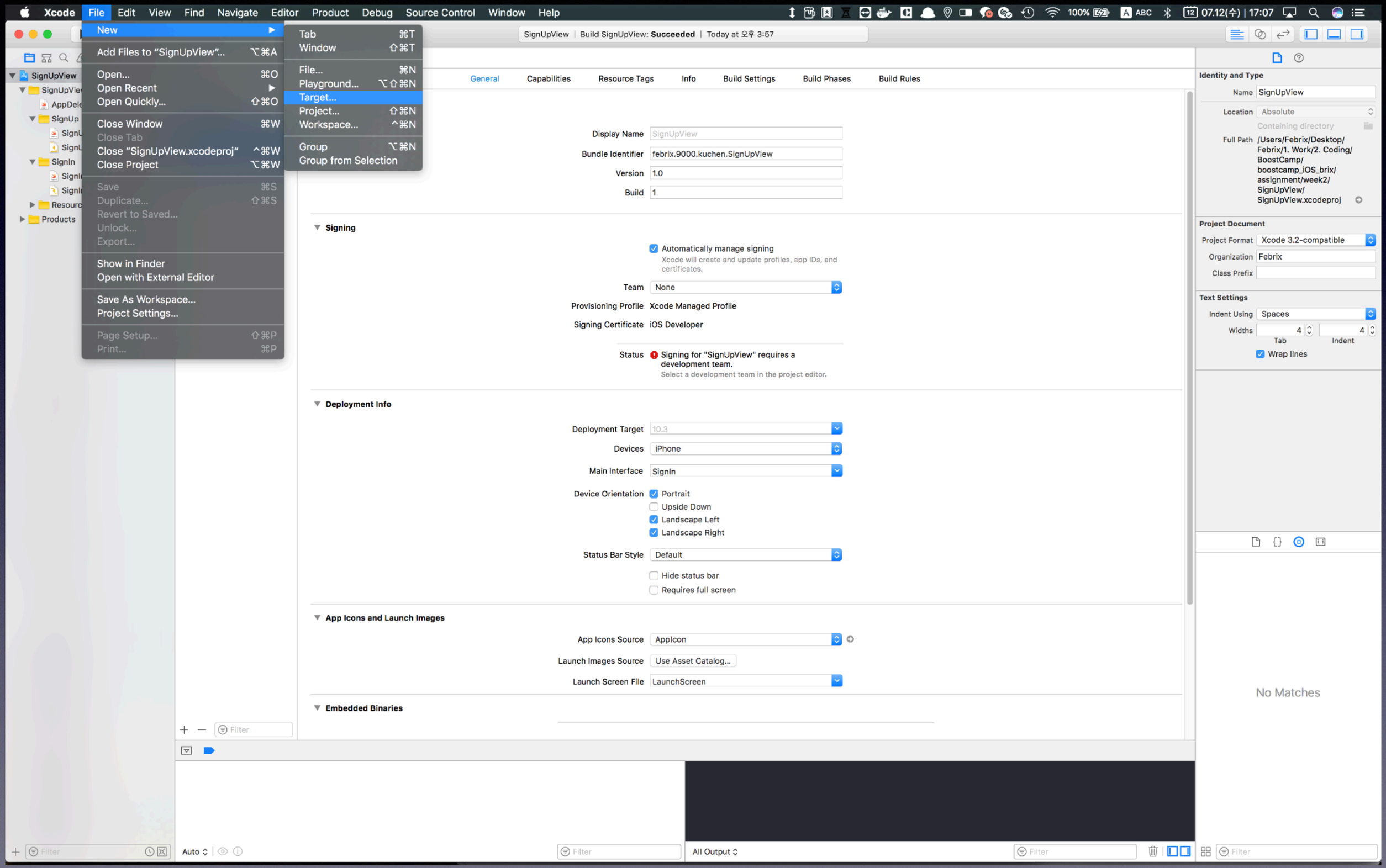
TDD Unit Test 적용해보기



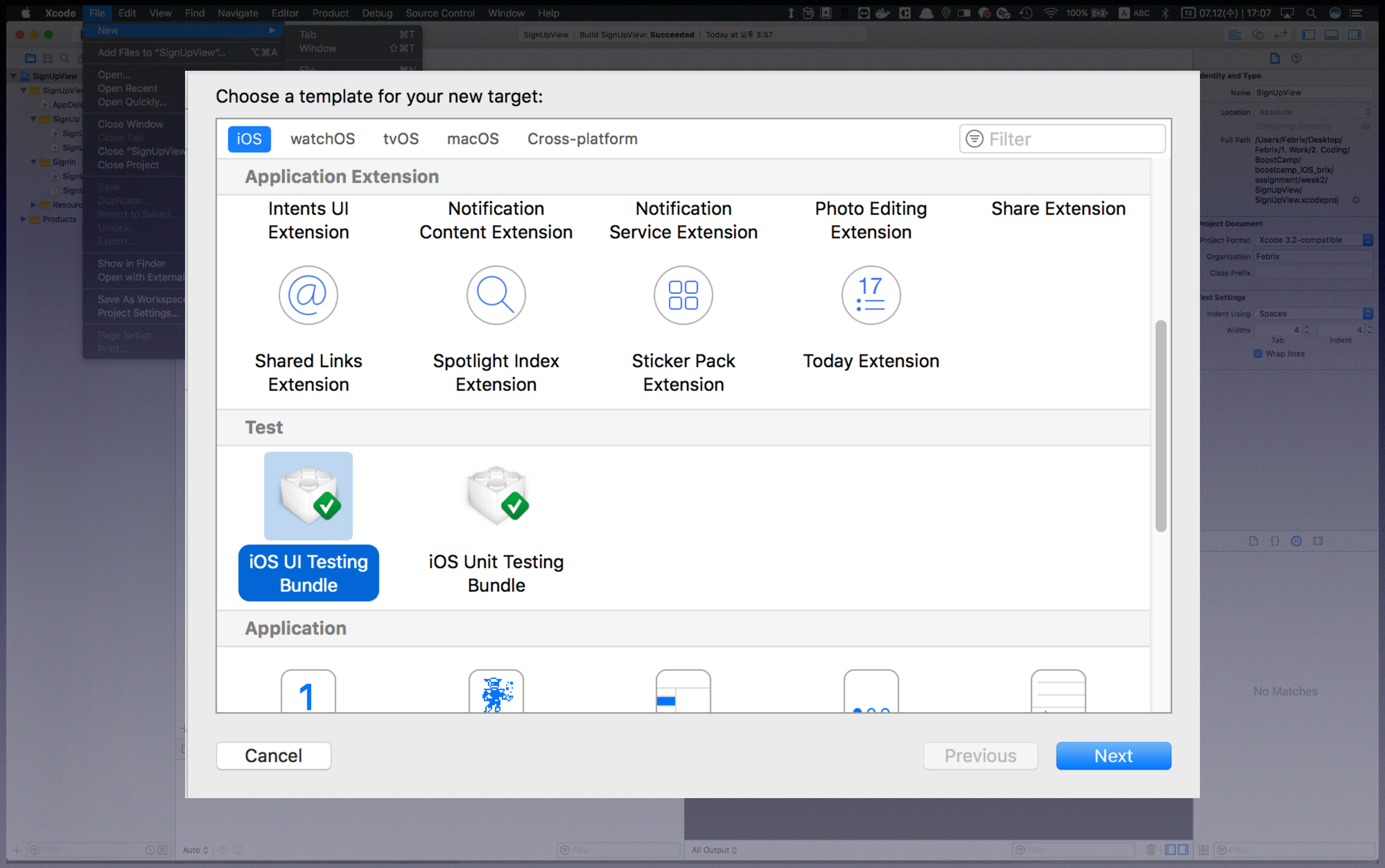
TDD Unit Test 적용해보기



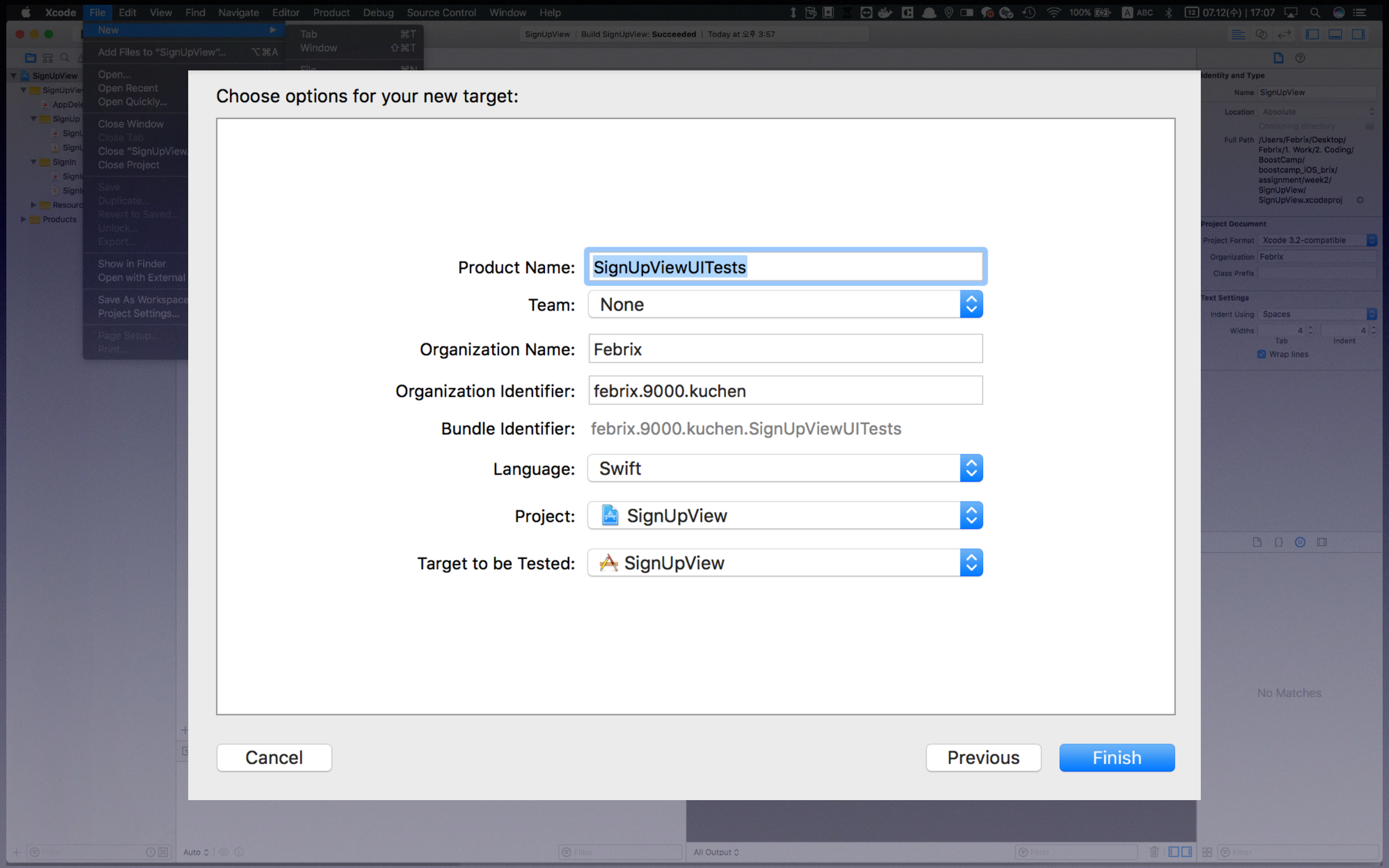
TDD UI Test 적용해보기



TDD UI Test 적용해보기



TDD UI Test 적용해보기



TDD UI Test 적용해보기

SignUpView 〉 SignUpViewUITests 〉 SignUpViewUITests.swift 〉 testExample()

```swift
//
//  SignUpViewUITests.swift
//  SignUpViewUITests
//
//  Created by Febrix on 2017. 7. 12..
//  Copyright © 2017년 Febrix. All rights reserved.
//

import XCTest

class SignUpViewUITests: XCTestCase {

    override func setUp() {
        super.setUp()

        // Put setup code here. This method is called before the invocation of each test method

        // In UI tests it is usually best to stop immediately when a failure occurs.
        continueAfterFailure = false
        // UI tests must launch the application that they test. Doing this in setup will make s
        XCUIApplication().launch()

        // In UI tests it's important to set the initial state - such as interface orientation
    }

    override func tearDown() {
        // Put teardown code here. This method is called after the invocation of each test meth
        super.tearDown()
    }

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

    }

}
```

TDD UI Test 적용해보기

SignUpView 〉 SignUpViewUITests 〉 SignUpViewUITests.swift 〉 testExample()

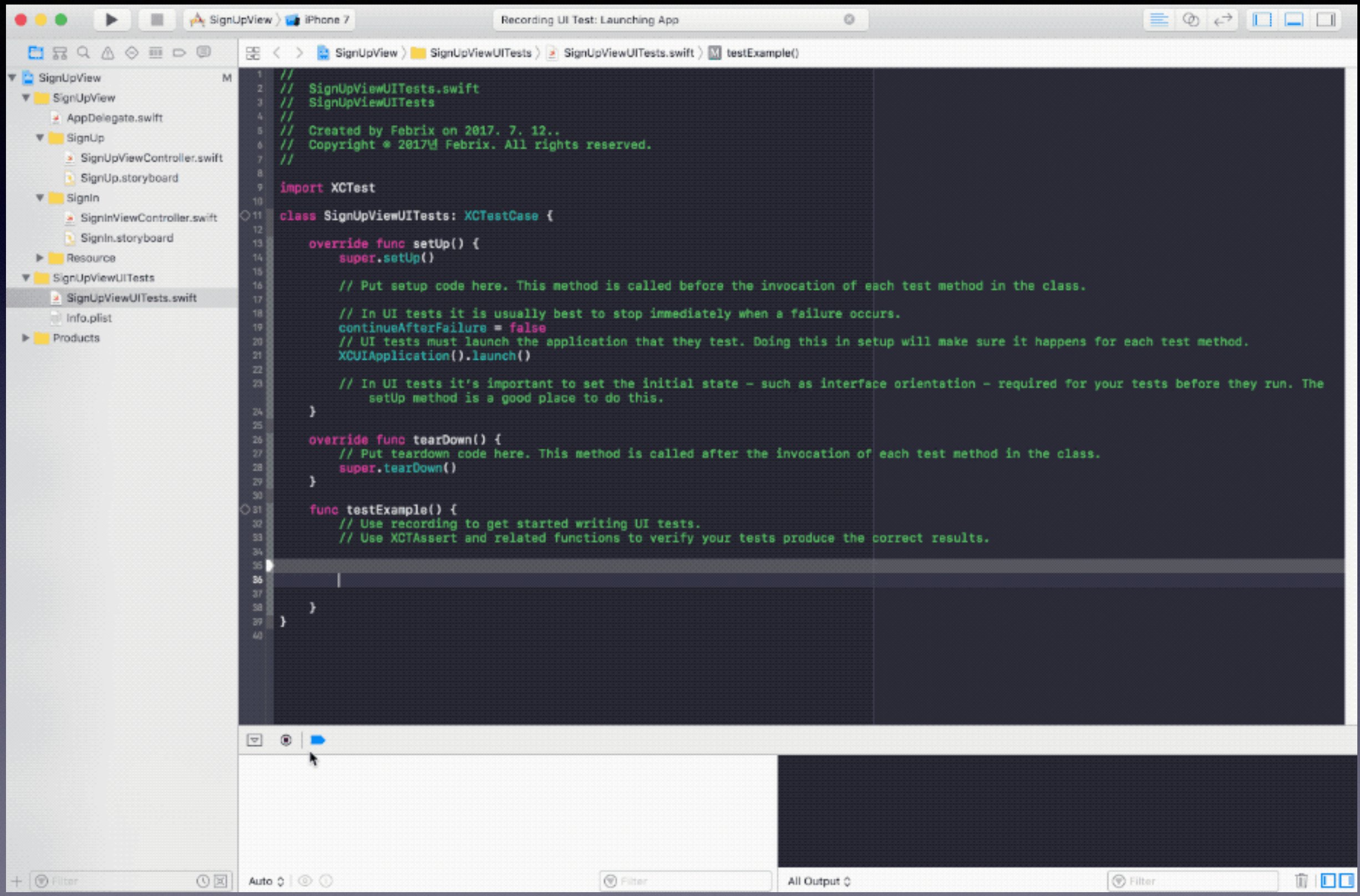
```
1  //
2  //  SignUpViewUITests.swift
3  //  SignUpViewUITests
4  //
5  //  Created by Febrix on 2017. 7. 12..
6  //  Copyright © 2017년 Febrix. All rights reserved.
7  //
8
9  import XCTest
10
11 class SignUpViewUITests: XCTestCase {
12
13     override func setUp() {
14         super.setUp()
15
16         // Put setup code here. This method is called before the invocation of each test method
17
18         // In UI tests it is usually best to stop immediately when a failure occurs.
19         continueAfterFailure = false
20         // UI tests must launch the application that they test. Doing this in setup will make s
21         XCUIApplication().launch()
22
23         // In UI tests it's important to set the initial state - such as interface orientation
24     }
25
26     override func tearDown() {
27         // Put teardown code here. This method is called after the invocation of each test meth
28         super.tearDown()
29     }
30
31     func testExample() {
32         // Use recording to get started writing UI tests.
33         // Use XCTAssert and related functions to verify your tests produce the correct results
34
35     }
36
37 }
38
```

TDD UI Test 적용해보기

```
30
31      func testExample() {
32          // Use recording to get started writing UI tests.
33          // Use XCTAssert and related functions to verify your tests produce the correct results.
34          XCUIDevice.shared().orientation = .portrait
35          XCUIDevice.shared().orientation = .portrait
36          XCUIDevice.shared().orientation = .faceUp
37          XCUIDevice.shared().orientation = .portrait
38
39          let app = XCUIApplication()
40          let idTextField = app.textFields["ID"]
41          idTextField.tap()
42          idTextField.typeText("test")
43
44          let passwordSecureTextField = app.secureTextFields["Password"]
45          passwordSecureTextField.tap()
46
47          let moreKey = app.keys["more"]
48          moreKey.tap()
49          moreKey.tap()
50          passwordSecureTextField.typeText("1234")
51          app.buttons["Sign In"].tap()
52
53      }
54  }
55
```

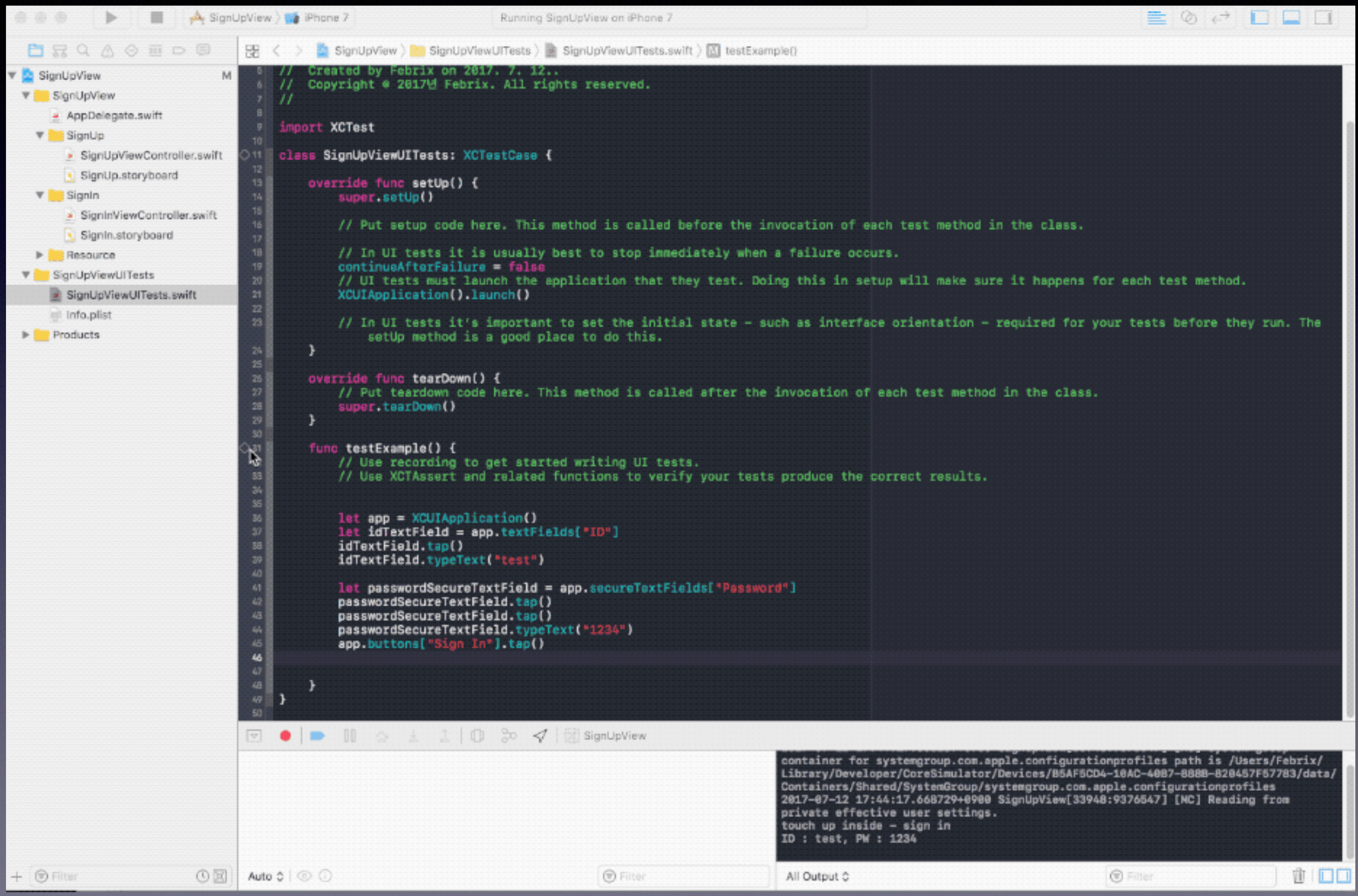
TDD UI Test 적용해보기

UI Recording



TDD UI Test 적용해보기

Test



TDD의 장단점

장점

작성한 코드가 테스트를 반드시 통과해야하므로 소스코드의 품질이 향상되고 디버깅 시간이 절약된다.
또한 유지보수와 리팩토링이 용이해진다.

단점

프로토타이핑에 부적합하다.
테스트 코드를 작성하는 비용이 든다.

감사합니다