2023 땅울림 썸머코딩

JUnit5를 이용한 단위테스트

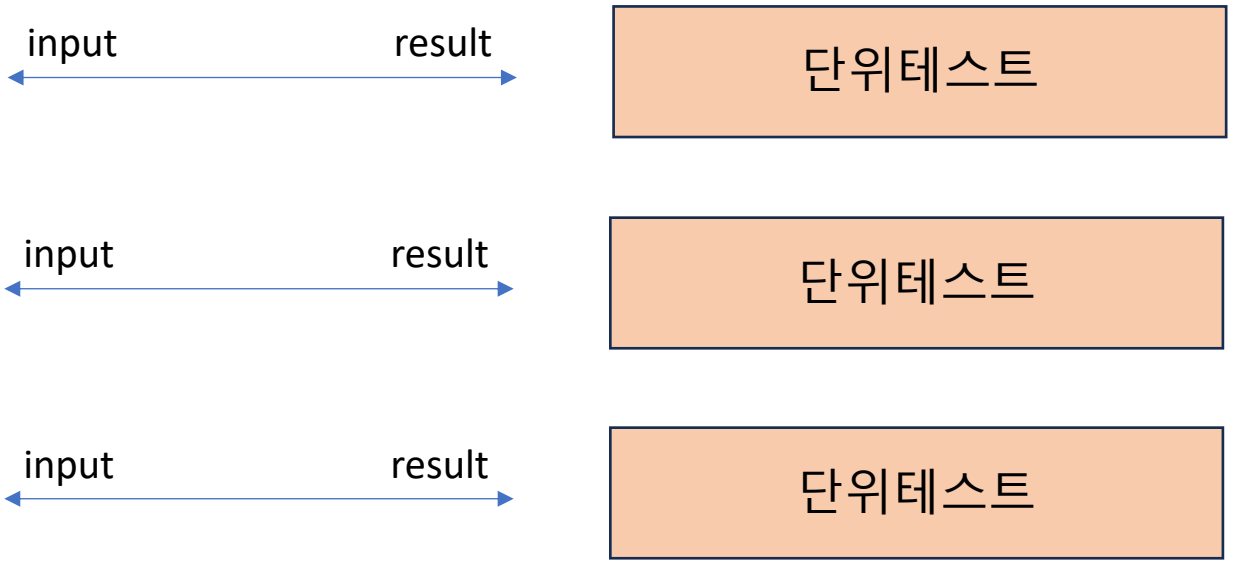
진행자: 김선우

목차

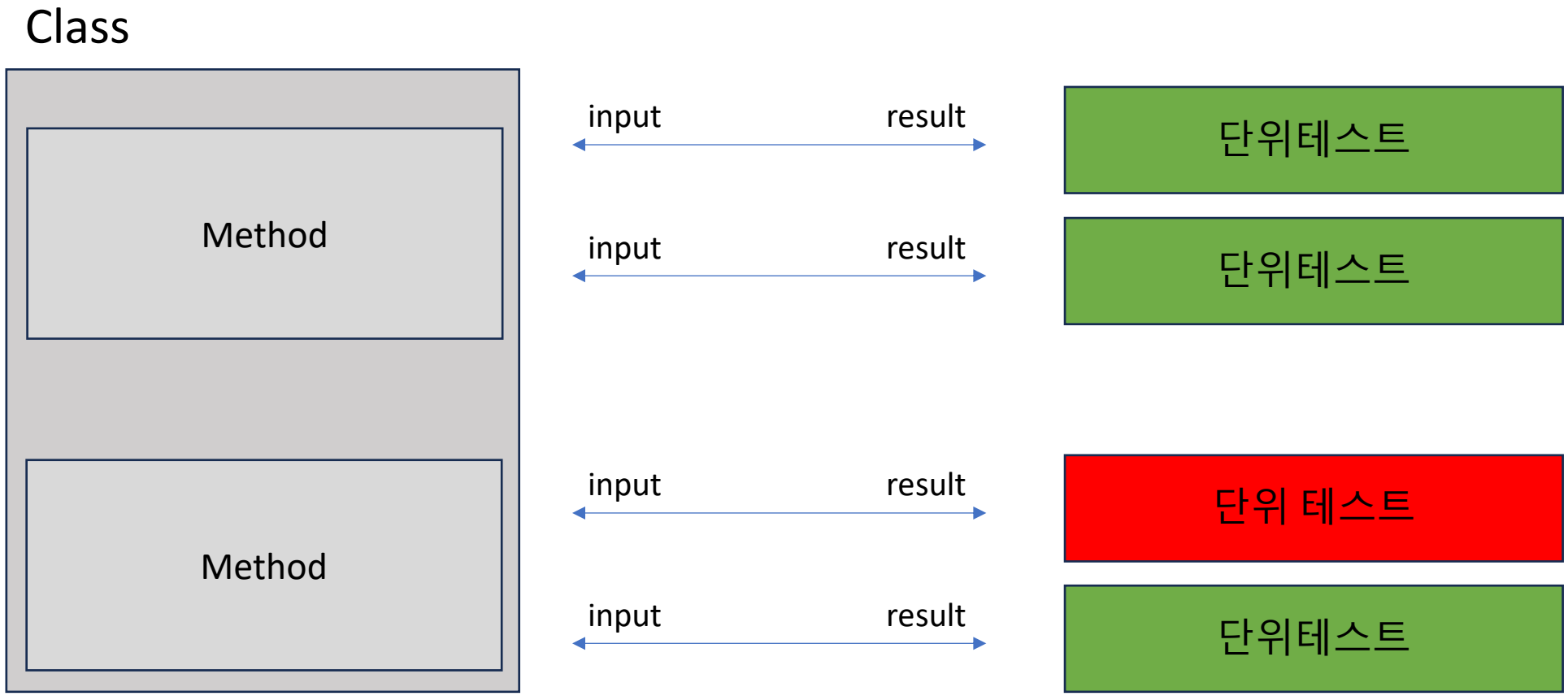
01
단위테스트와 JUnit5

단위테스트와 JUnit5

```
public boolean isInputValid(String input){


   // 입력을 검증하는 메소드

   // 길이, 특수문자 여부 등...



}
```



단위테스트와 JUnit5

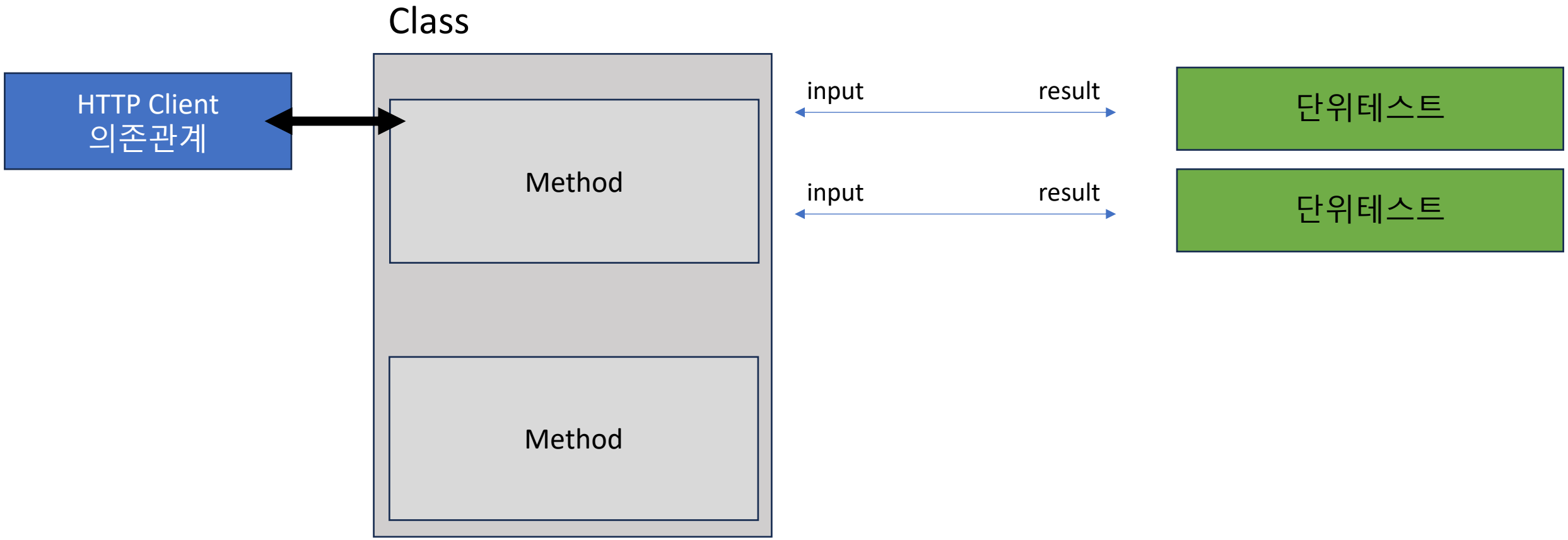


단위테스트와 JUnit5

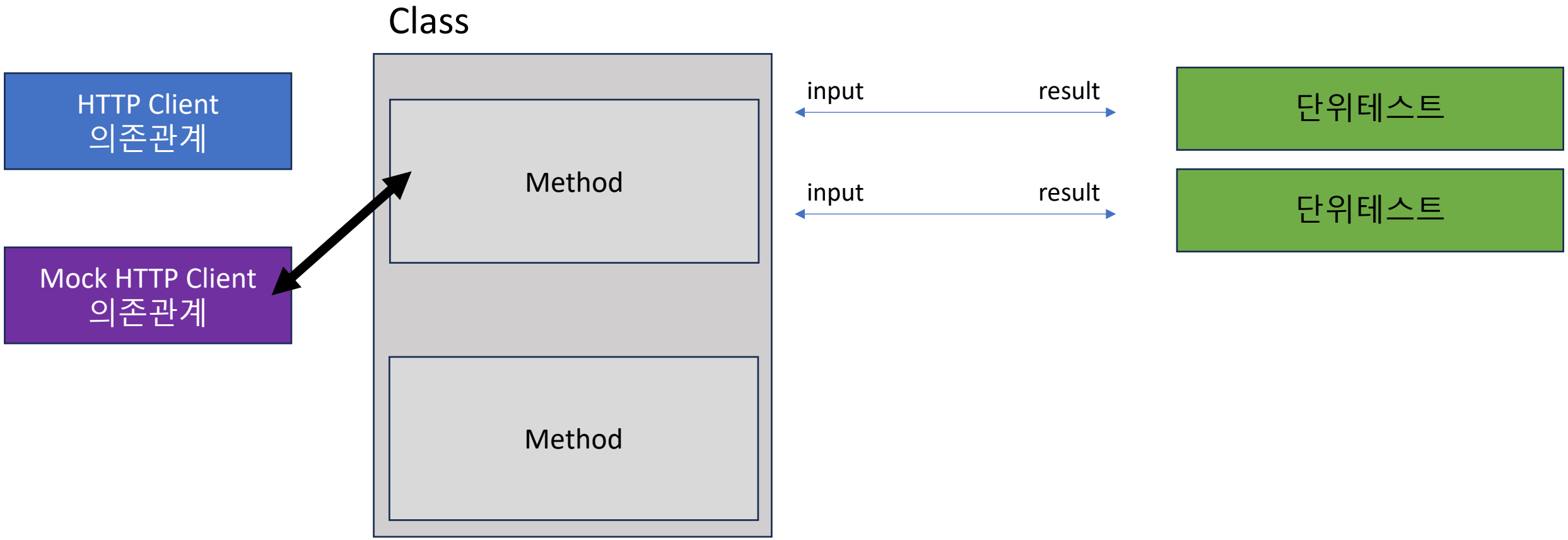
```java
public class Calculator {

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

}
```

```java
@Test
void testIntegerDivision(){
  // Arrange
  Calculator calculator = new Calculator();

  // Act
  int result = calculator.integerDivision( dividend: 4,  divisor: 2);

  //Assert
  Assertions.assertEquals( expected: 2, result,  message: "4/2 should be 2!");
}
```

단위테스트와 JUnit5



단위테스트와 JUnit5



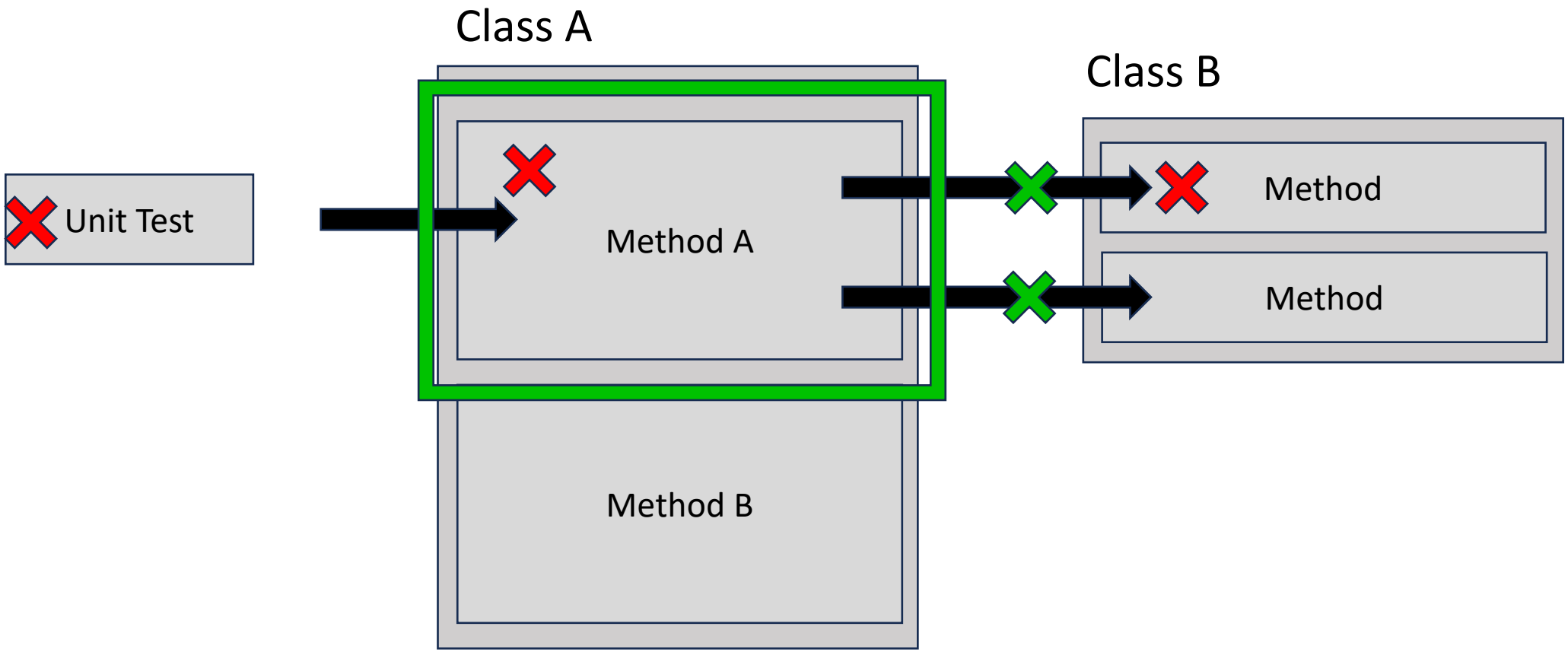
단위테스트를 작성해야하는 이유

- 코드가 제대로 동작하는지 확인하기
- 리팩토링 해도 제대로 동작하는지 확인
- 프로그램의 신뢰성 확보
- 개발 초기 단계에서 버그 발견

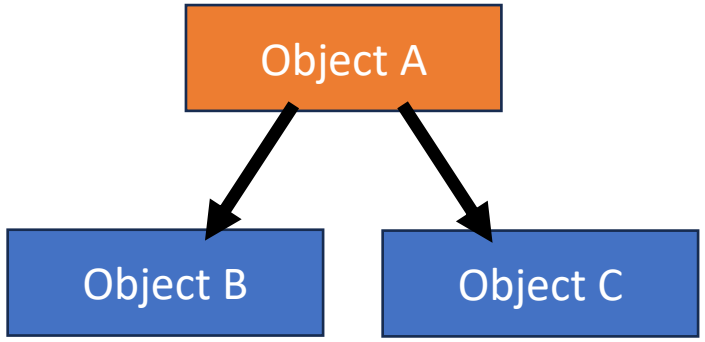
FIRST

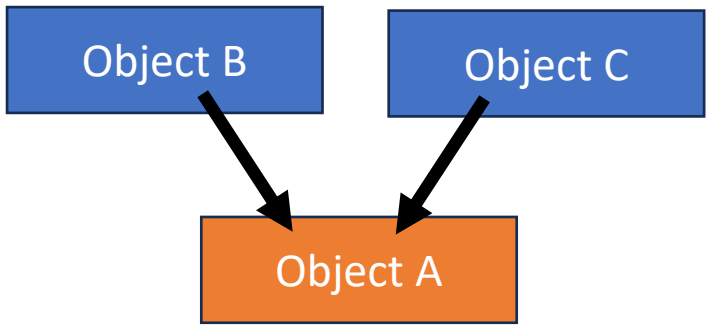
- Fast – 빠르게 동작
- Independent – 다른 테스트 의존 X 독립적이어야함
- Repeatable – 여러번 동작해도 같은 결과
- Self-validating – 단위테스트는 자신을 검증해야한다
- Thorough & Timely – 적재적소에 작성되어야한다

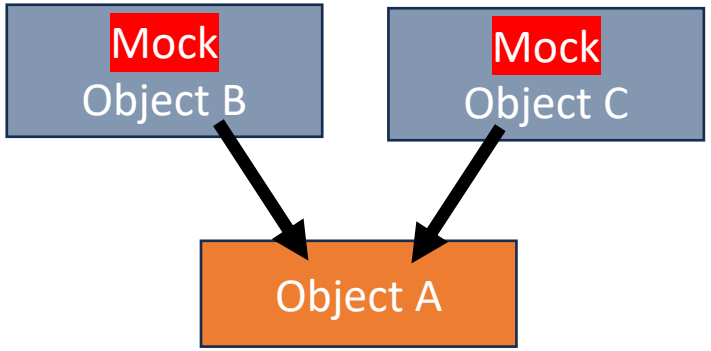
단위테스트와 JUnit5



Dependency Injection

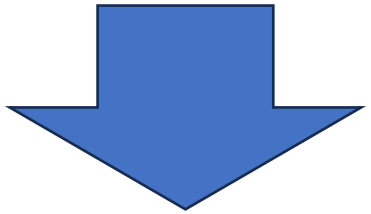






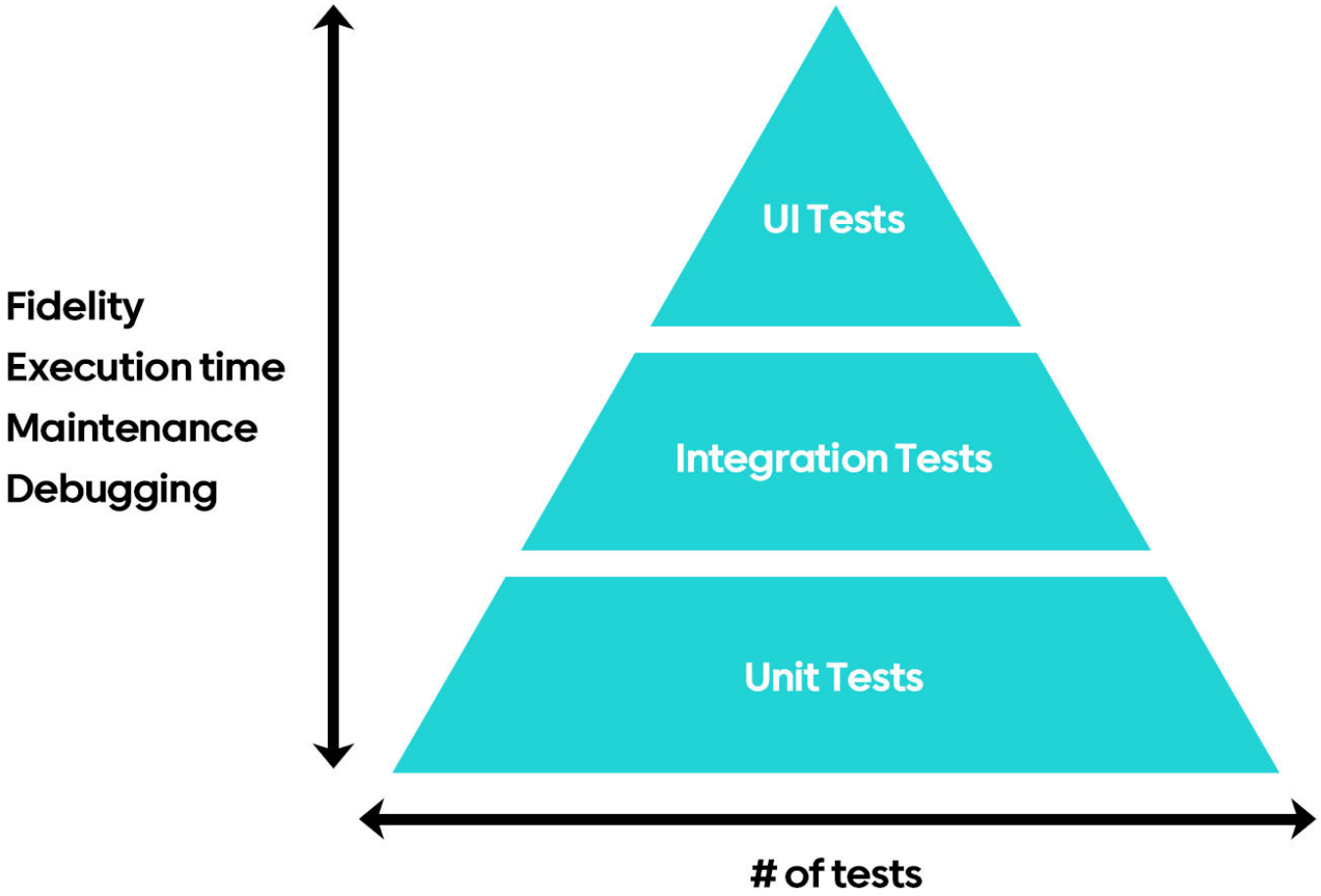
단위테스트와 JUnit5

```
public SignupResult processUserSignUp(FormData formData){

    SignupWebService webservice = new SignupWebserviceImpl();

    return webservice.performSignup(formData)
}
```

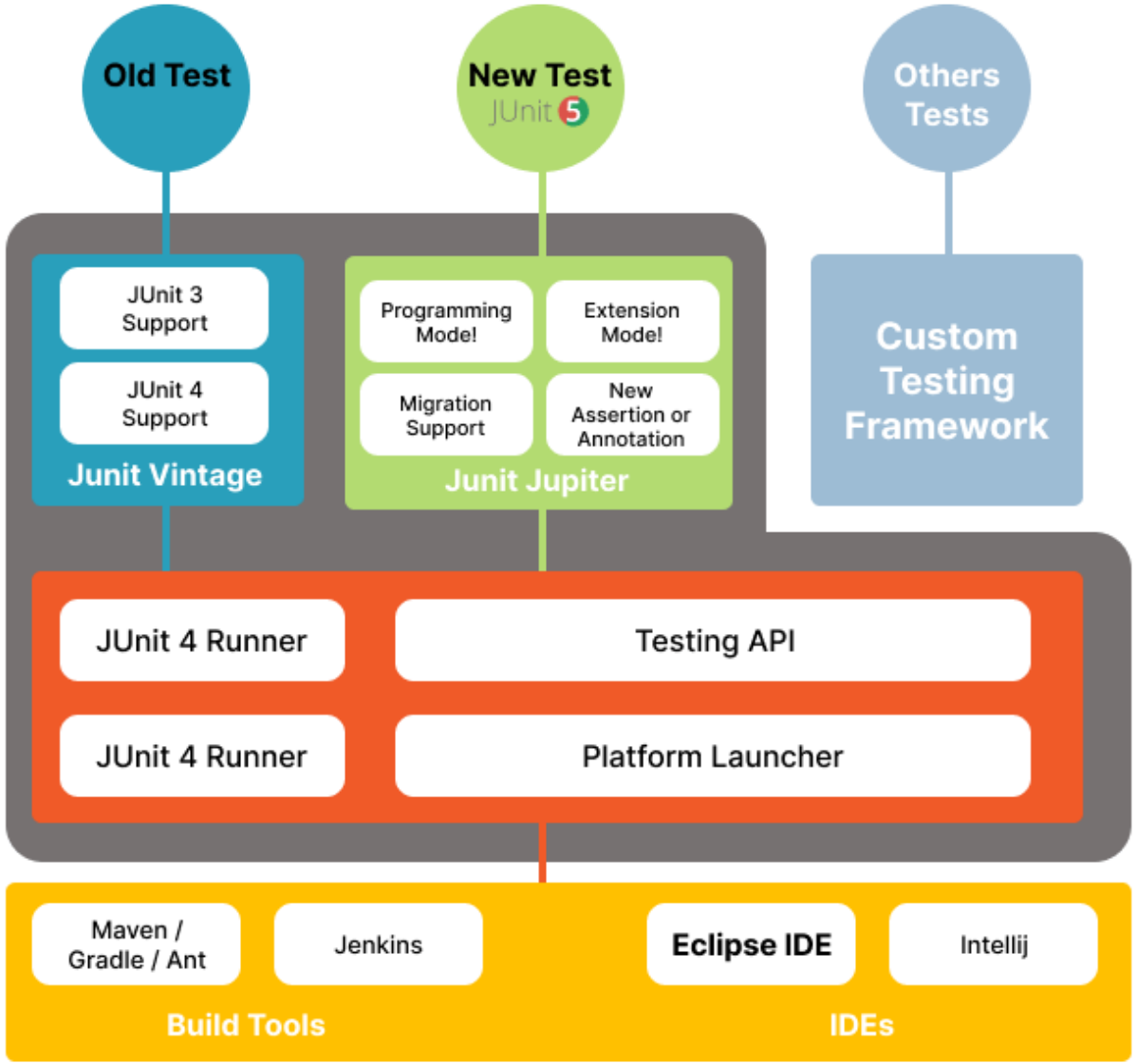


```
public SignupResult processUserSignUp(FormData formData, SignupWebService signupWebService){

    return webservice.performSignup(formData)
}
```

단위테스트와 JUnit5

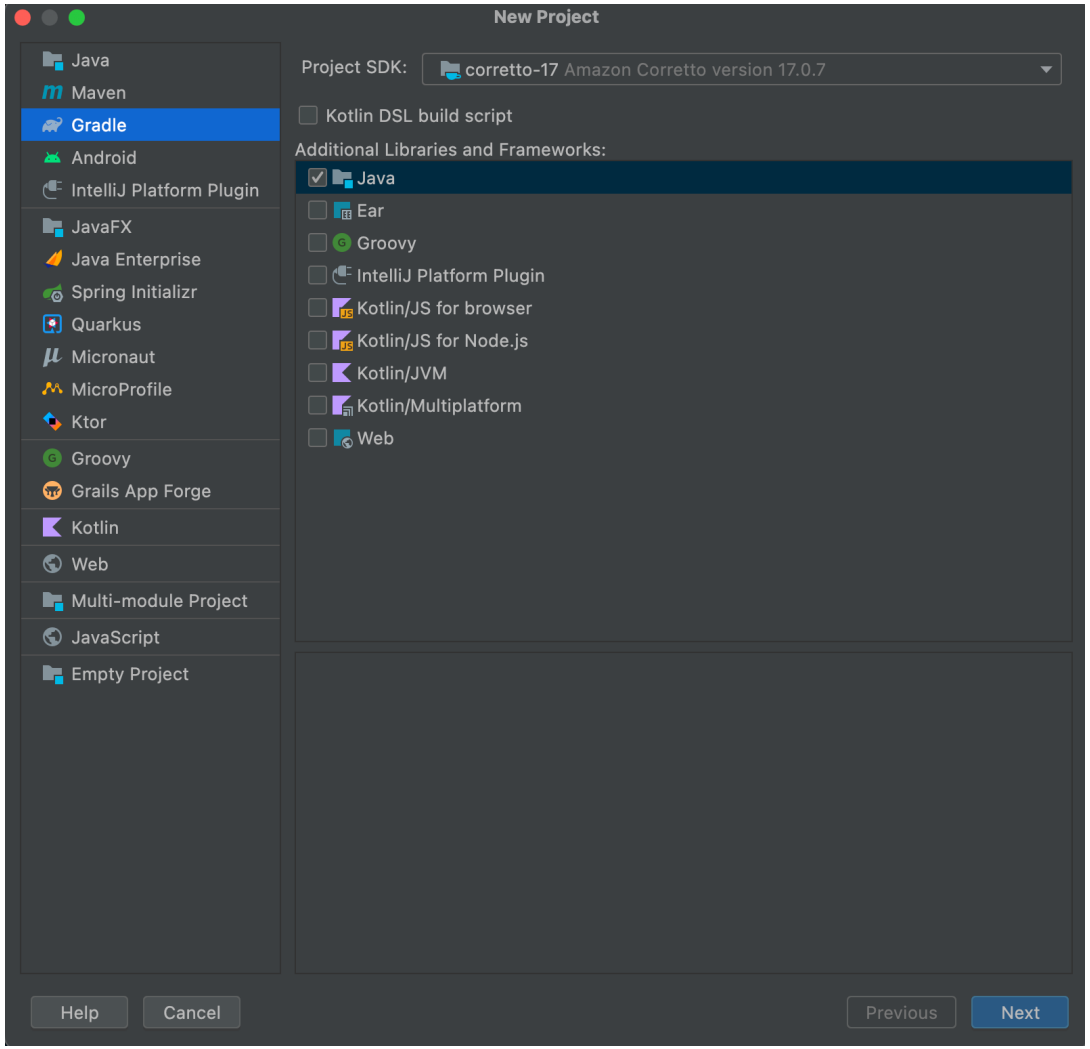


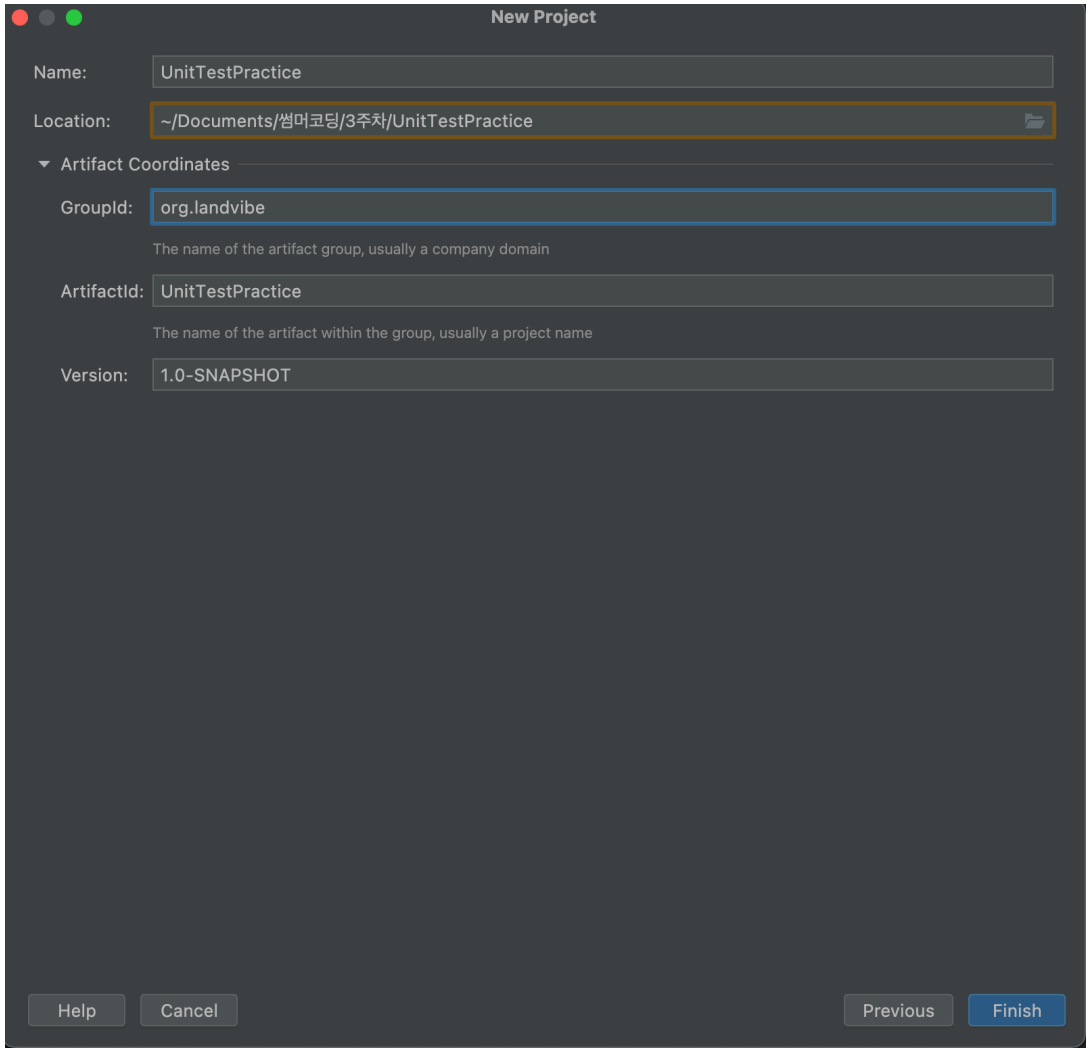
단위테스트와 JUnit5



02
JUnit 기본 문법

JUnit5와 기본문법



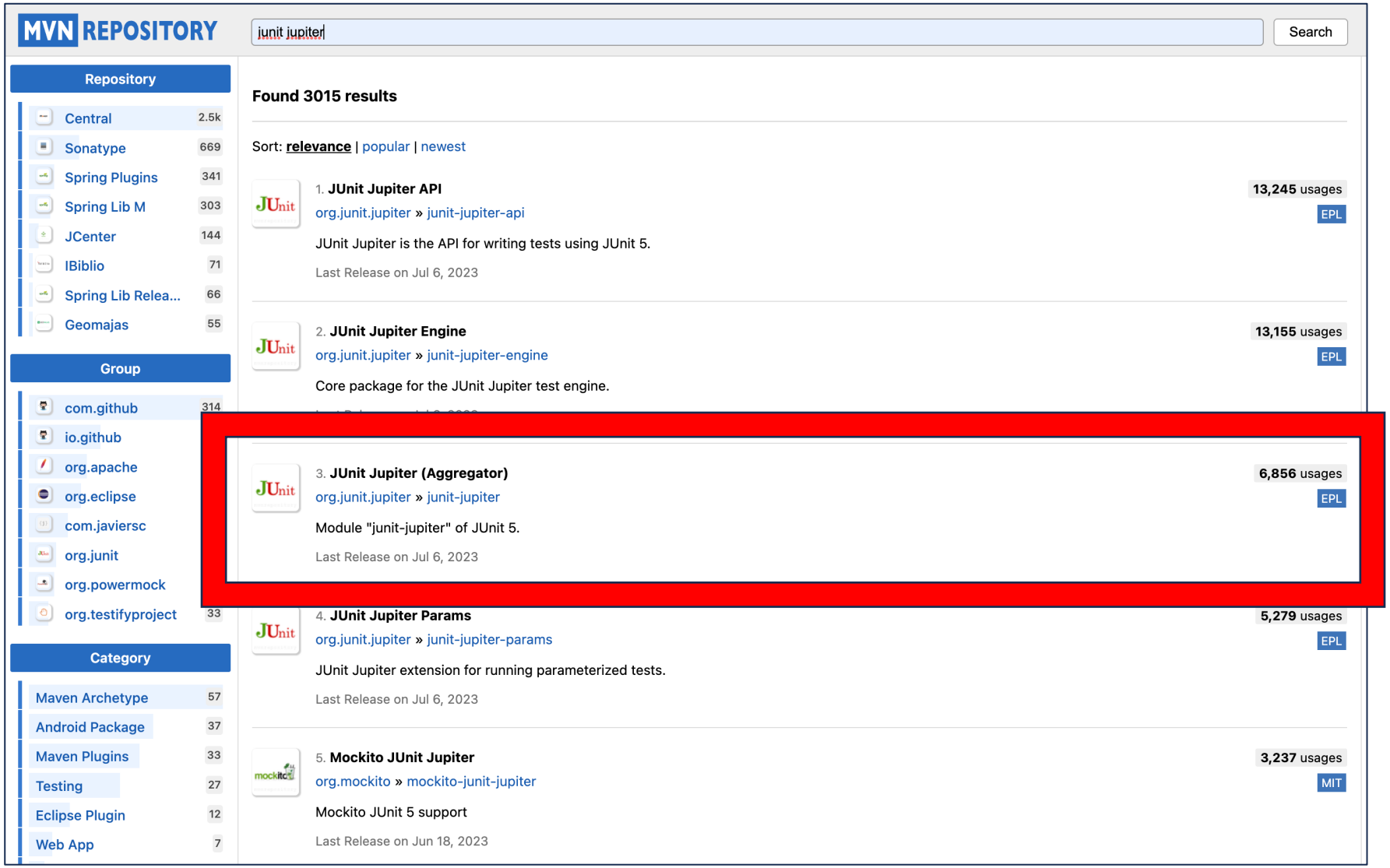


JUnit5와 기본문법

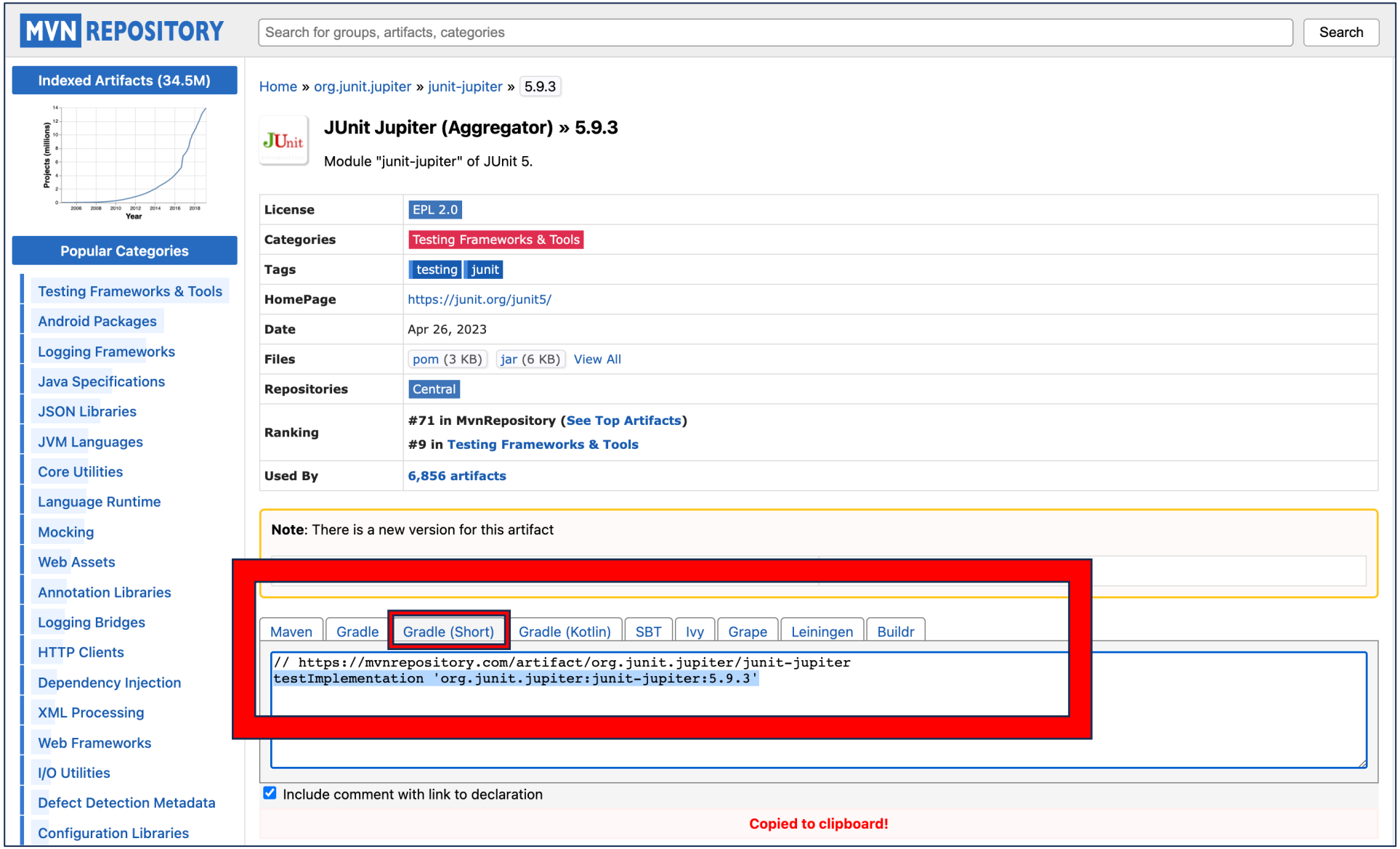
build.gradle (UnitTestPractice)

```
plugins {
    id 'java'
}

group 'org.landvibe'
version '1.0-SNAPSHOT'

repositories {
    mavenCentral()
}

dependencies {
    testImplementation 'org.junit.jupiter:junit-jupiter-api:5.8.1'
    testRuntimeOnly 'org.junit.jupiter:junit-jupiter-engine:5.8.1'
}

test {
    useJUnitPlatform()
}
```

JUnit5와 기본문법



JUnit5와 기본문법

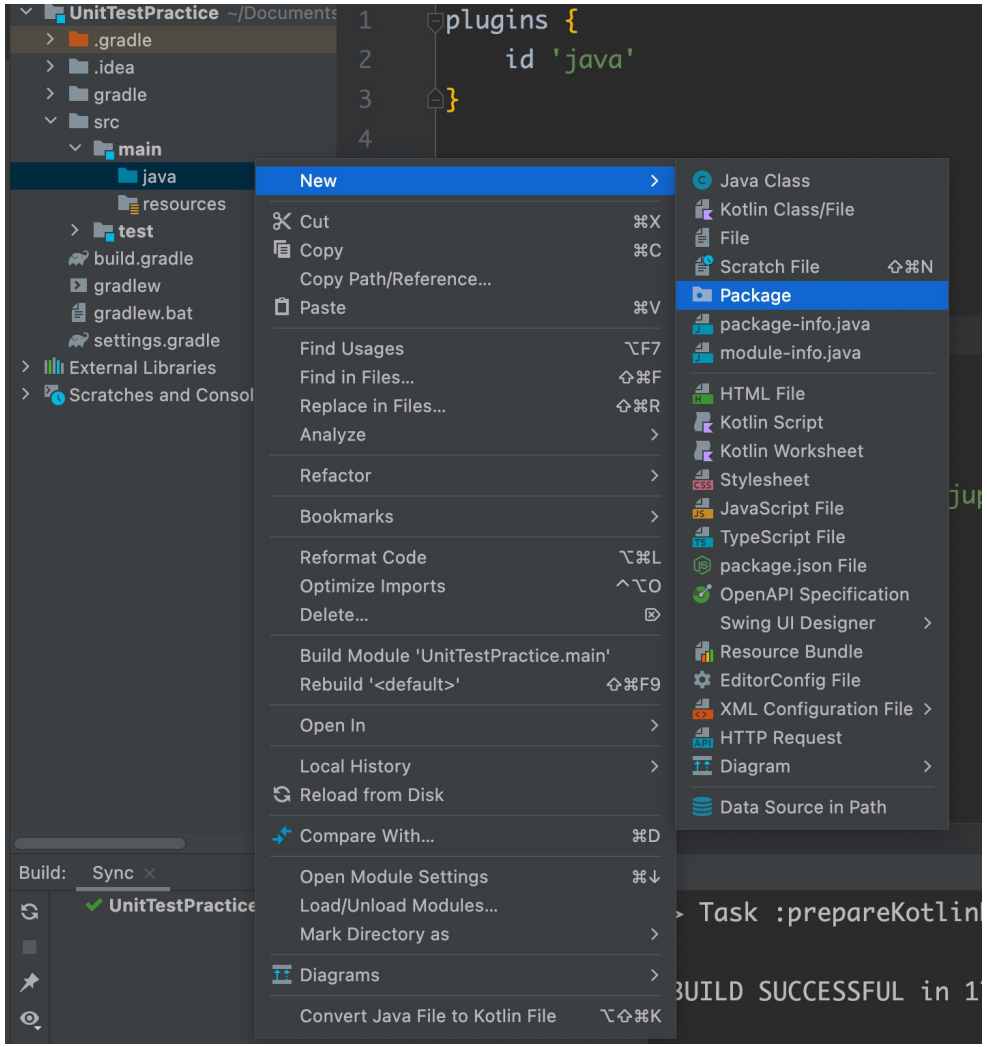


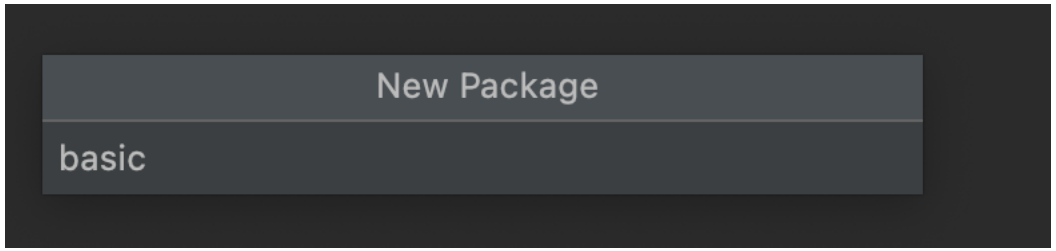
JUnit5와 기본문법

build.gradle (UnitTestPractice)

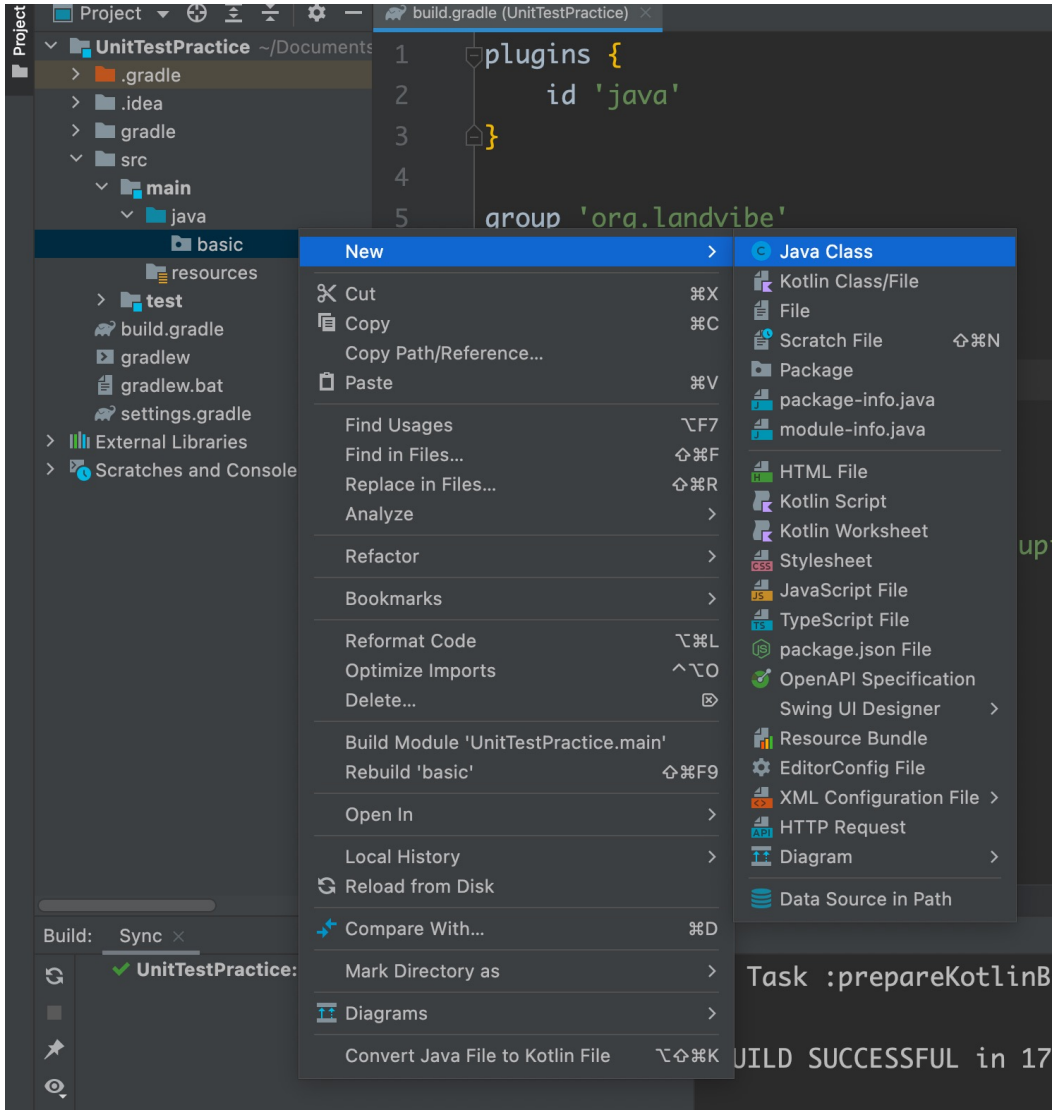
```
plugins {
    id 'java'
}

group 'org.landvibe'
version '1.0-SNAPSHOT'

repositories {
    mavenCentral()
}

dependencies {
    testImplementation 'org.junit.jupiter:junit-jupiter:5.9.3'
}

test {
    useJUnitPlatform()
}
```

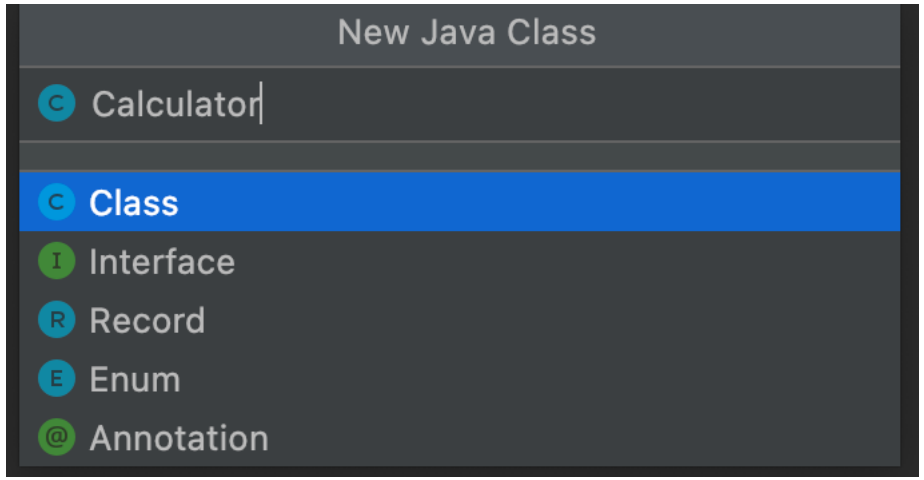
JUnit5와 기본문법





JUnit5와 기본문법



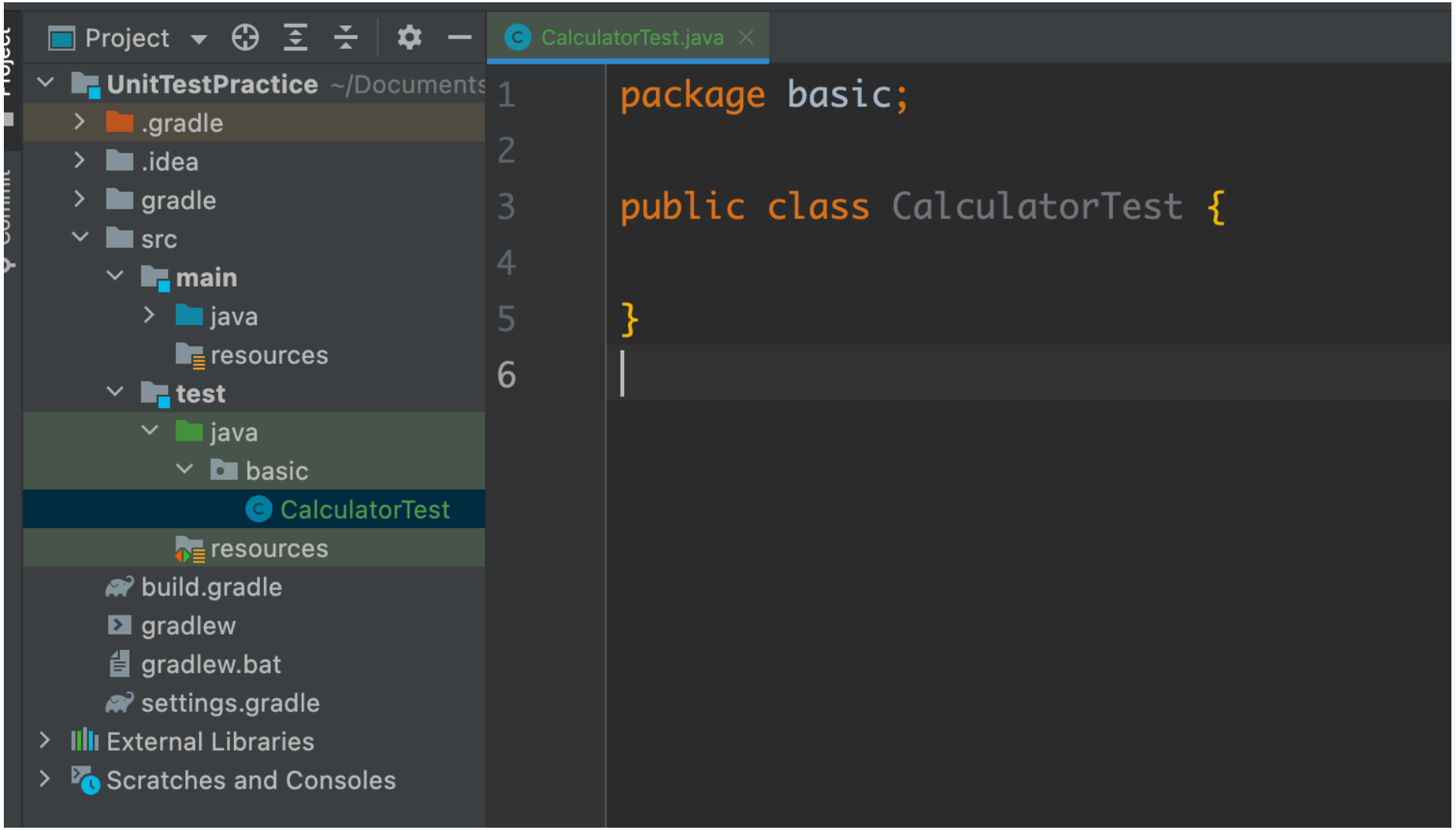


JUnit5와 기본문법

Calculator.java

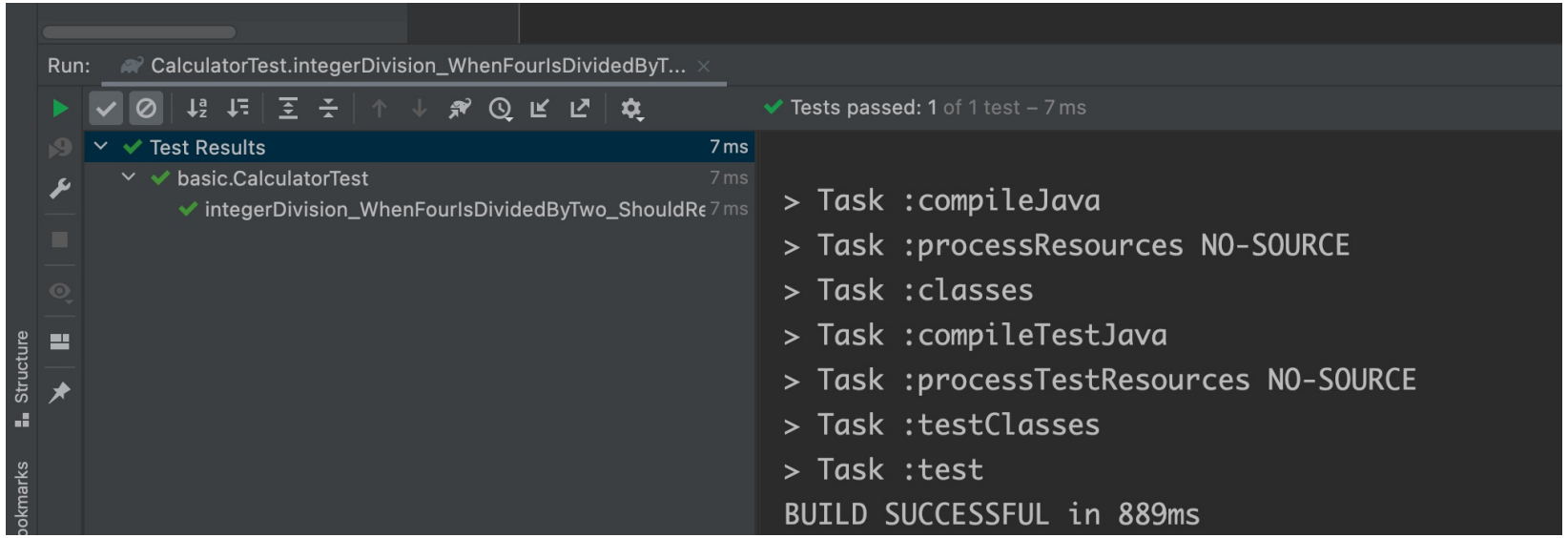
```java
package basic;

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

    public int integerSubtraction(int minuend, int subtrahend){
        return minuend - subtrahend;
    }
}
```

JUnit5와 기본문법



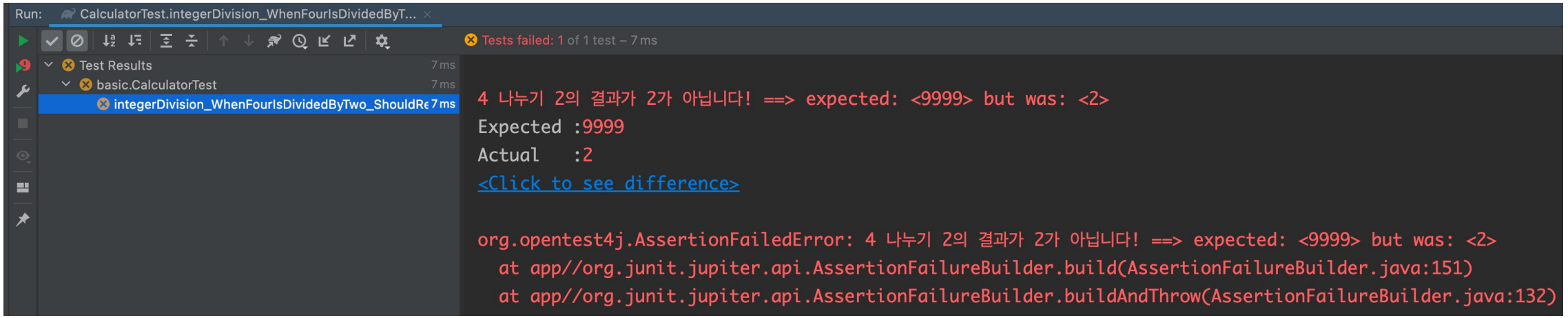
JUnit5와 기본문법

```java
public class CalculatorTest {

    @Test
    void integerDivision_WhenFourIsDividedByTwo_ShouldReturnTwo(){
        Calculator calculator = new Calculator();
        int result = calculator.integerDivision( dividend: 4,  divisor: 2);
        Assertions.assertEquals( expected: 2, result);
    }
}
```



JUnit5와 기본문법

```java
@Test
void integerDivision_WhenFourIsDividedByTwo_ShouldReturnTwo(){
    Calculator calculator = new Calculator();
    int result = calculator.integerDivision( dividend: 4,  divisor: 2);
    Assertions.assertEquals( expected: 9999, result,  message: "4 나누기 2의 결과가 2가 아닙니다!");
}
}
```



JUnit5와 기본문법

```java
@Test
void integerDivision_WhenFourIsDividedByTwo_ShouldReturnTwo() {
    // Arrange (given)
    Calculator calculator = new Calculator();
    int dividend = 4;
    int divisor = 2;
    int expectedResult = 2;

    // Act (when)
    int result = calculator.integerDivision(dividend, divisor);

    // Assert (then)
    assertEquals(expectedResult, result, message: "4 나누기 2의 결과가 2가 아닙니다!");
}
```

JUnit5와 기본문법

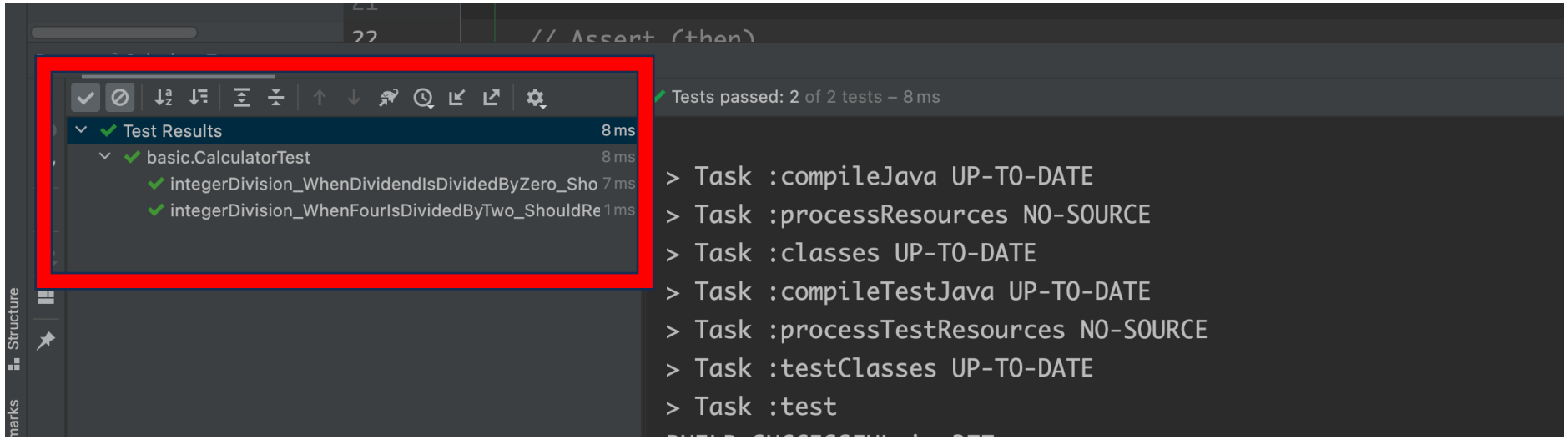
```java
@Test
void integerDivision_WhenDividendIsDividedByZero_ShouldThrowArithmeticException() {
    // Arrange (given)
    Calculator calculator = new Calculator();
    int dividend = 4;
    int divisor = 0;

    // Act, Assert
    Assertions.assertThrows(ArithmeticException.class, () -> {
        calculator.integerDivision(dividend, divisor);
    }, message: "0으로 나눌 경우에는 ArithmeticException을 발생시킨다");
}
```

JUnit5와 기본문법

```java
@Test
void integerDivision_WhenDividendIsDividedByZero_ShouldThrowArithmeticException() {
    // Arrange
    Calculator calculator = new Calculator();
    int dividend = 4;
    int divisor = 0;
    String expectedExceptionMessage = "/ by zero";

    // Act
    ArithmeticException actualException = assertThrows(ArithmeticException.class, () -> {
        calculator.integerDivision(dividend, divisor);
    }, message: "0으로 나눌 경우에는 ArithmeticException을 발생시킨다");

    // Assert
    assertEquals(expectedExceptionMessage, actualException.getMessage(),
            message: "Unexpected exception message");
}
```

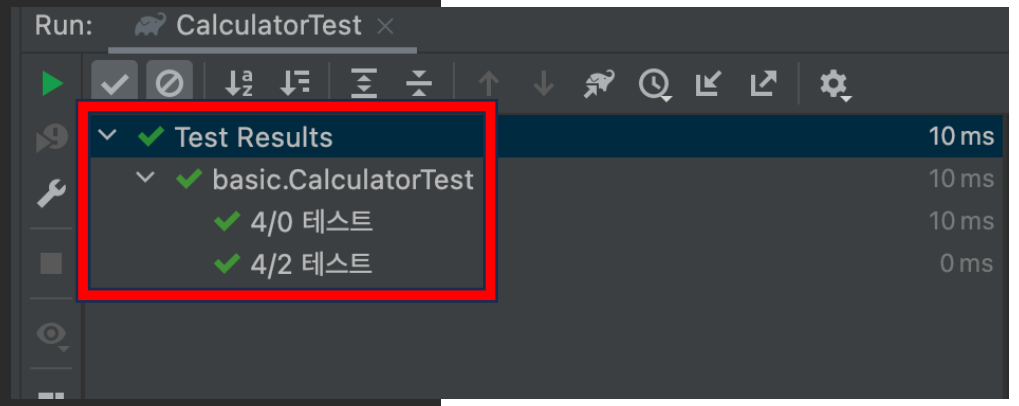
JUnit5와 기본문법

```java
public class CalculatorTest {

  @Test
  void integerDivision_WhenFourIsDividedByTwo_ShouldReturnTwo() {
    // Arrange (given)
    Calculator calculator = new Calculator();
    int dividend = 4;
```



JUnit5와 기본문법

```java
@DisplayName("4/2 테스트")
@Test
void integerDivision_WhenFourIsDividedByTwo_ShouldReturnTwo() {
    // Arrange (given)
    Calculator calculator = new Calculator();
    int dividend = 4;
    int divisor = 2;
    int expectedResult = 2;

    // Act (when)
    int result = calculator.integerDivision(dividend, divisor);

    // Assert (then)
    assertEquals(expectedResult, result, message: "4 나누기 2의 결과가 2가 아닙니다!");
}

@DisplayName("4/0 테스트")
@Test
void integerDivision_WhenDividendIsDividedByZero_ShouldThrowArithmeticException() {
    // Arrange
```



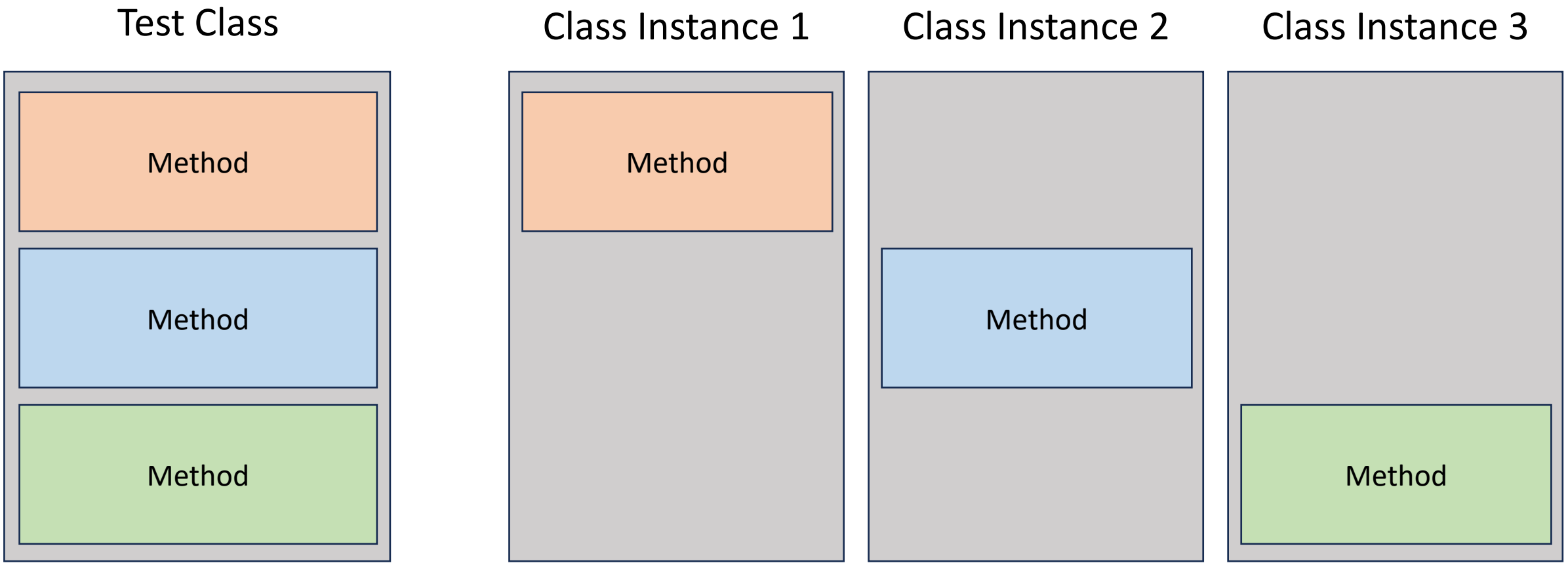
JUnit5와 기본문법

```java
@DisplayName("33-1 테스트")
@Test
void integerSubtraction_When33Minus1_ShouldReturn32() {
    // Arrange
    Calculator calculator = new Calculator();
    int minuend = 33;
    int subtrahend = 1;
    int expectedResult = 32;

    // Act
    int result = calculator.integerSubtraction( minuend: 33,  subtrahend: 1);

    // Assert
    assertEquals( expected: 32, result,
            message: minuend + "-" + subtrahend + " 결과값이 " + expectedResult + "와 다릅니다.");
}
```

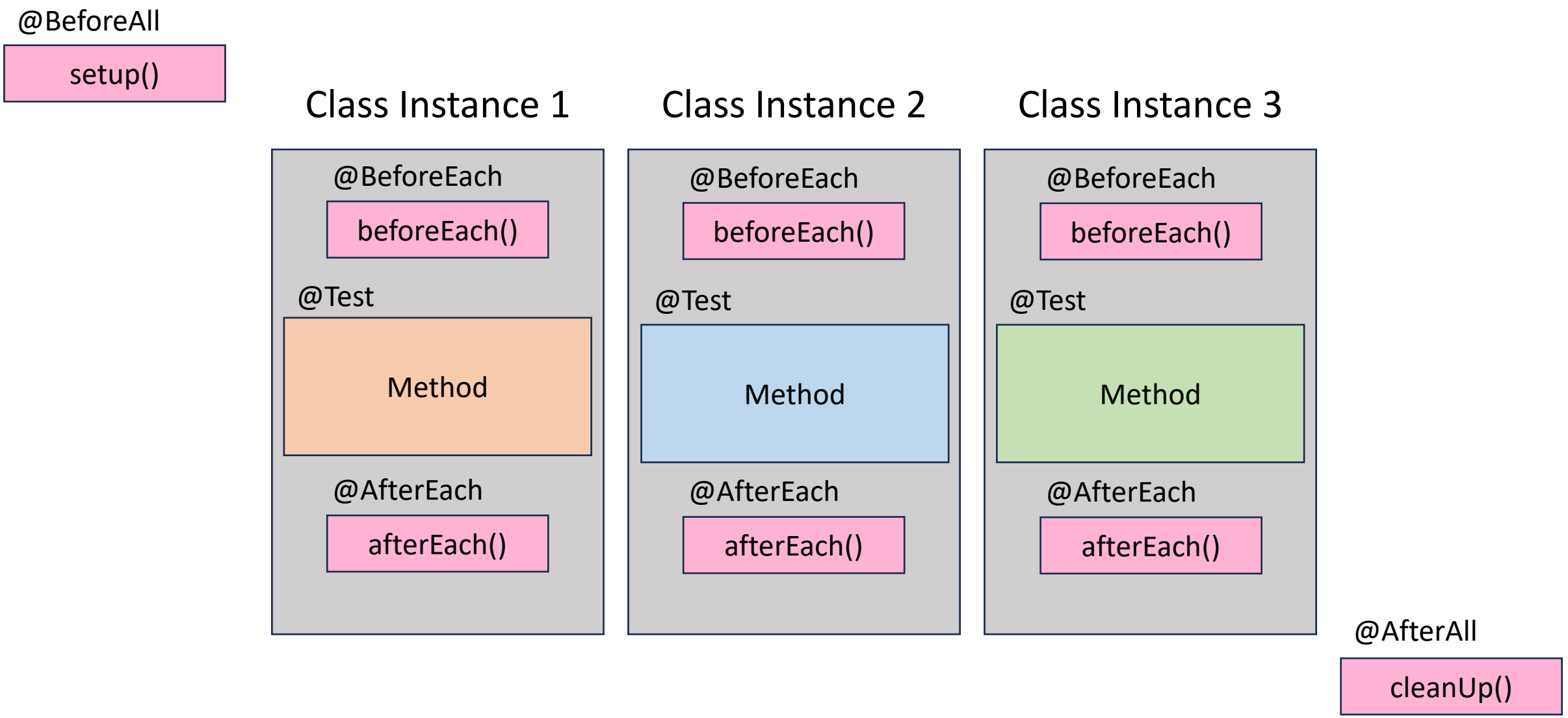
JUnit5와 기본문법

```java
@DisplayName("33-1 테스트")
@Test
void integerSubtraction_When33Minus1_ShouldReturn32() {
    // Arrange
    Calculator calculator = new Calculator();
    int minuend = 33;
    int subtrahend = 1;
    int expectedResult = 32;

    // Act
    int result = calculator.integerSubtraction( minuend: 33,  subtrahend: 1);

    // Assert
    assertEquals( expected: 32, result,
        () -> minuend + "-" + subtrahend + " 결과값이 " + expectedResult + "와 다릅니다.");
}
```

JUnit5와 기본문법



JUnit5와 기본문법



JUnit5와 기본문법

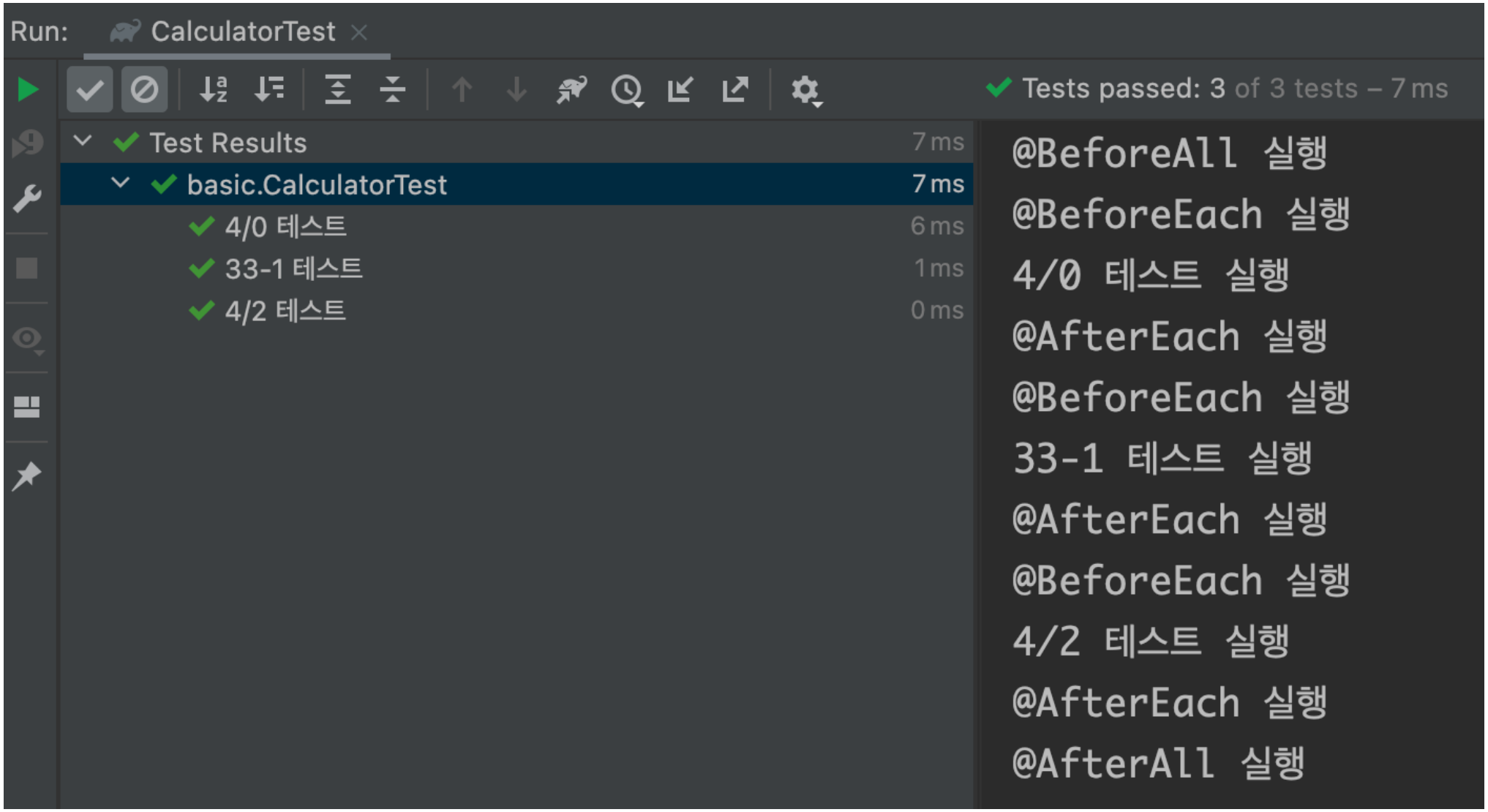
```java
public class CalculatorTest {

  @BeforeAll
  static void setup() {
    System.out.println("@BeforeAll 실행");
  }

  @AfterAll
  static void cleanup() {
    System.out.println("@AfterAll 실행");
  }

  @BeforeEach
  void beforeEachTestMethod() {
    System.out.println("@BeforeEach 실행");
  }

  @AfterEach
  void afterEachTestMethod() {
    System.out.println("@AfterEach 실행");
  }
```

```java
@DisplayName("4/2 테스트")
@Test
void integerDivision_WhenFourIsDividedByTwo_ShouldReturnTwo() {

  System.out.println("4/2 테스트 실행");
```

```java
@DisplayName("4/0 테스트")
@Test
void integerDivision_WhenDividendIsDividedByZero_ShouldThrowArithmeticException() {

  System.out.println("4/0 테스트 실행");
```

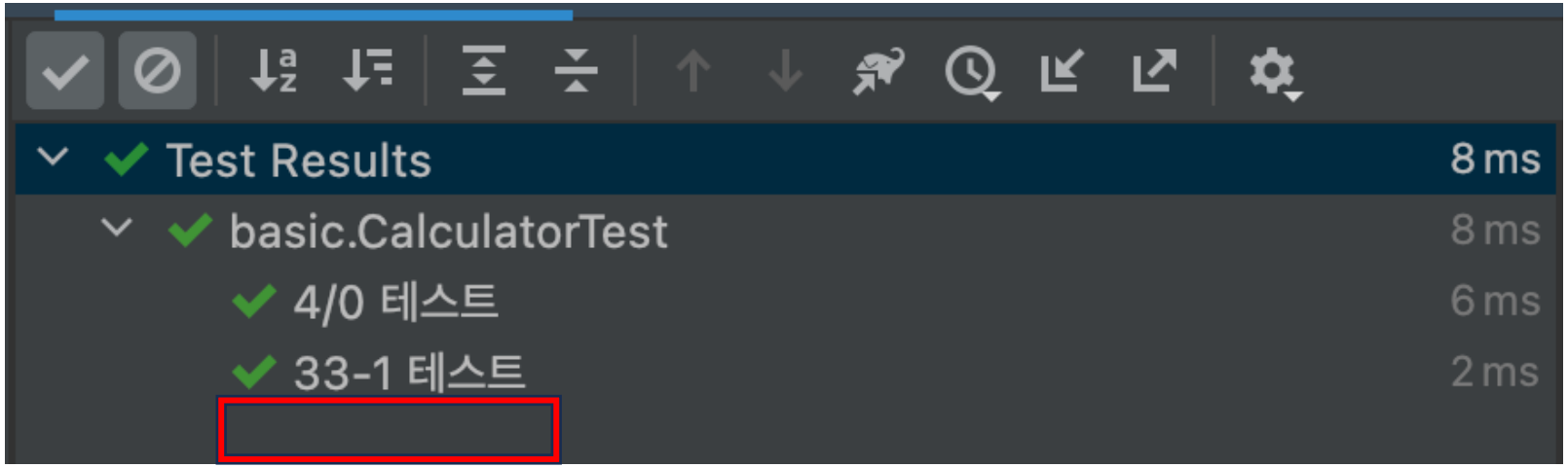
```java
@DisplayName("33-1 테스트")
@Test
void integerSubtraction_When33Minus1_ShouldReturn32() {

  System.out.println("33-1 테스트 실행");
```

JUnit5와 기본문법



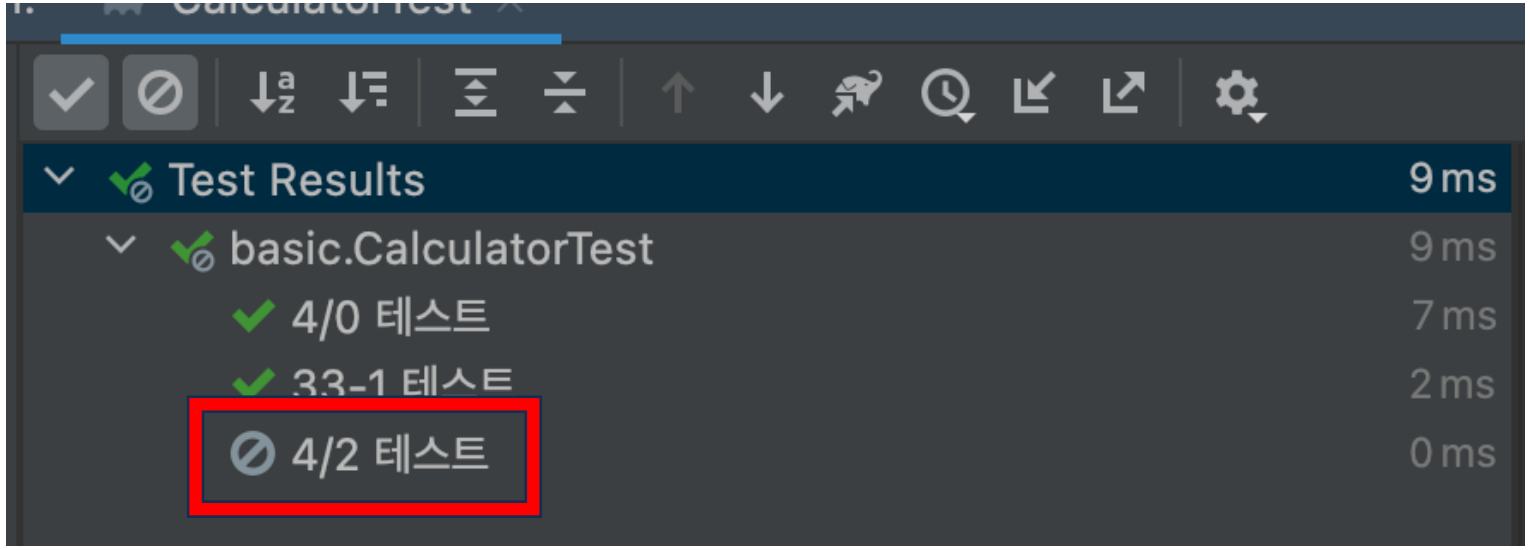
JUnit5와 기본문법

```
@DisplayName("4/2 테스트")
//@Test
void integerDivision_WhenFourIsDividedByTwo_ShouldReturnTwo() {
```



JUnit5와 기본문법

```
@Disabled
@DisplayName("4/2 테스트")
@Test
void integerDivision_WhenFourIsDividedByTwo_ShouldReturnTwo() {
```



JUnit5와 기본문법

```java
public class CalculatorTest {

  Calculator calculator;

  @BeforeAll
  static void setup() {
    System.out.println("@BeforeAll 실행");
  }

  @AfterAll
  static void cleanup() {
    System.out.println("@AfterAll 실행");
  }

  @BeforeEach
  void beforeEachTestMethod() {
    calculator = new Calculator();
    System.out.println("@BeforeEach 실행");
  }

  @AfterEach
  void afterEachTestMethod() {
    System.out.println("@AfterEach 실행");
  }
```

```java
@DisplayName("4/2 테스트")
@Test
void integerDivision_WhenFourIsDividedByTwo_ShouldReturnTwo() {

  System.out.println("4/2 테스트 실행");

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

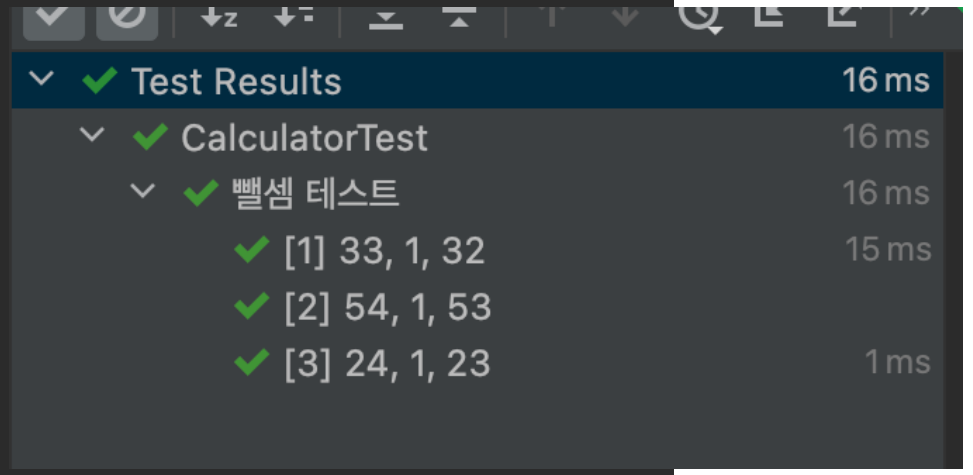
```java
@DisplayName("4/0 테스트")
@Test
void integerDivision_WhenDividendIsDividedByZero_ShouldThrowArithmeticException() {

  System.out.println("4/0 테스트 실행");

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

```java
@DisplayName("33-1 테스트")
@Test
void integerSubtraction_When33Minus1_ShouldReturn32() {

  System.out.println("33-1 테스트 실행");

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

03
JUnit 심화문법

JUnit5 심화문법

```java
@DisplayName("뺄셈 테스트")
@ParameterizedTest
@MethodSource("integerSubtraction")
void integerSubtraction_1(int minuend, int subtrahend, int expectedResult) {

    int result = calculator.integerSubtraction(minuend, subtrahend);

    assertEquals(expectedResult, result,
        () -> minuend + "-" + subtrahend + " 결과값이 " + expectedResult + "와 다릅니다.");
}

private static Stream<Arguments> integerSubtraction() {
    return Stream.of(
        Arguments.of(33, 1, 32),
        Arguments.of(54, 1, 53),
        Arguments.of(24, 1, 23)
    );
}
```

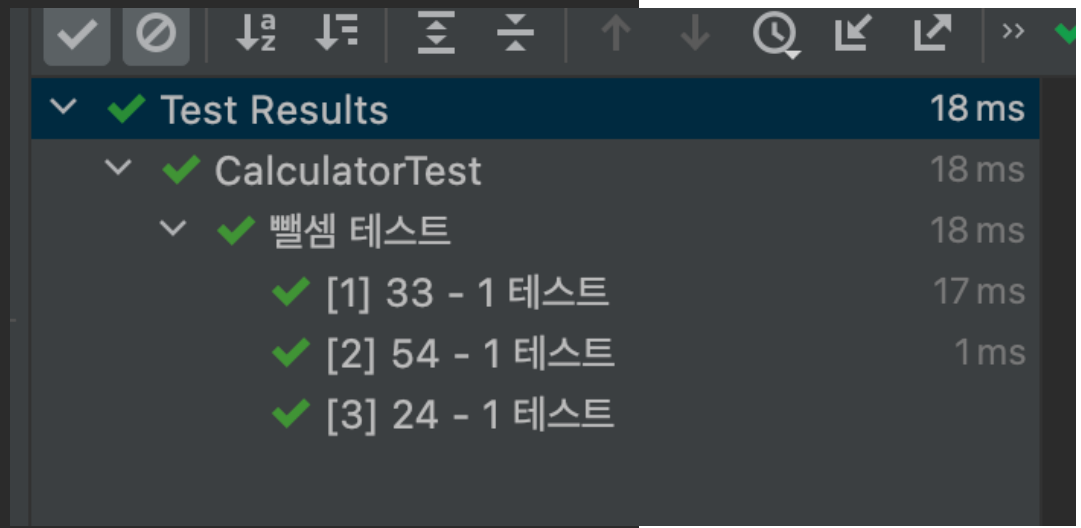


JUnit5 심화문법

```java
@DisplayName("뺄셈 테스트")
@ParameterizedTest
@MethodSource()
void integerSubtraction_1(int minuend, int subtrahend, int expectedResult) {

    int result = calculator.integerSubtraction(minuend, subtrahend);

    assertEquals(expectedResult, result,
        () -> minuend + "-" + subtrahend + " 결과값이 " + expectedResult + "와 다릅니다.");
}

private static Stream<Arguments> integerSubtraction_1() {
    return Stream.of(
        Arguments.of(33, 1, 32),
        Arguments.of(54, 1, 53),
        Arguments.of(24, 1, 23)
    );
}
```

JUnit5 심화문법

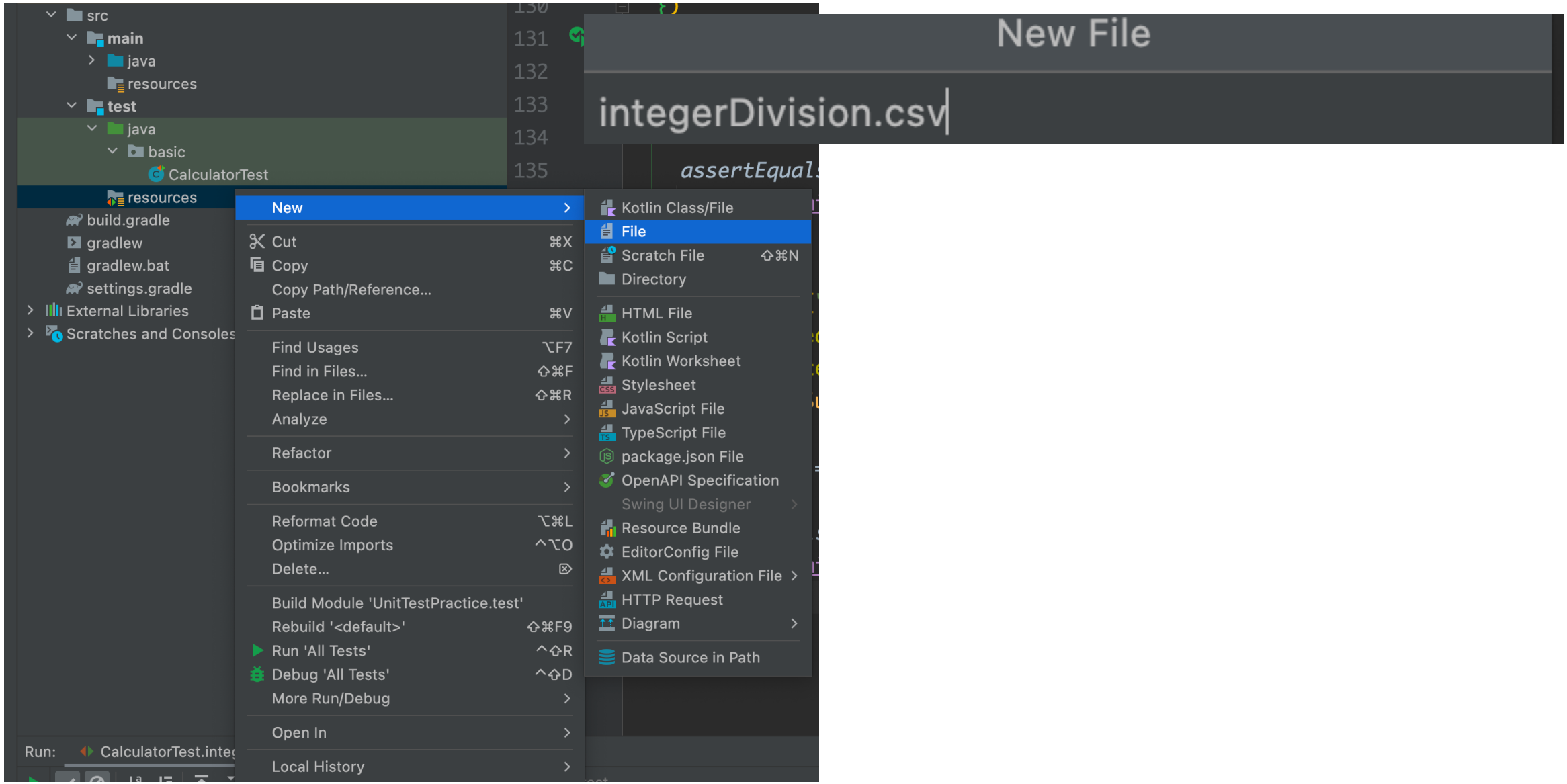
```java
@DisplayName("뺄셈 테스트")
@ParameterizedTest(name ="[{index}] {0} - {1} 테스트")
@MethodSource()
void integerSubtraction_1(int minuend, int subtrahend, int expectedResult) {

  int result = calculator.integerSubtraction(minuend, subtrahend);

  assertEquals(expectedResult, result,
      () -> minuend + "-" + subtrahend + " 결과값이 " + expectedResult + "와 다릅니다.");
}

private static Stream<Arguments> integerSubtraction_1() {
  return Stream.of(
      Arguments.of(33, 1, 32),
      Arguments.of(54, 1, 53),
      Arguments.of(24, 1, 23)
  );
}
```



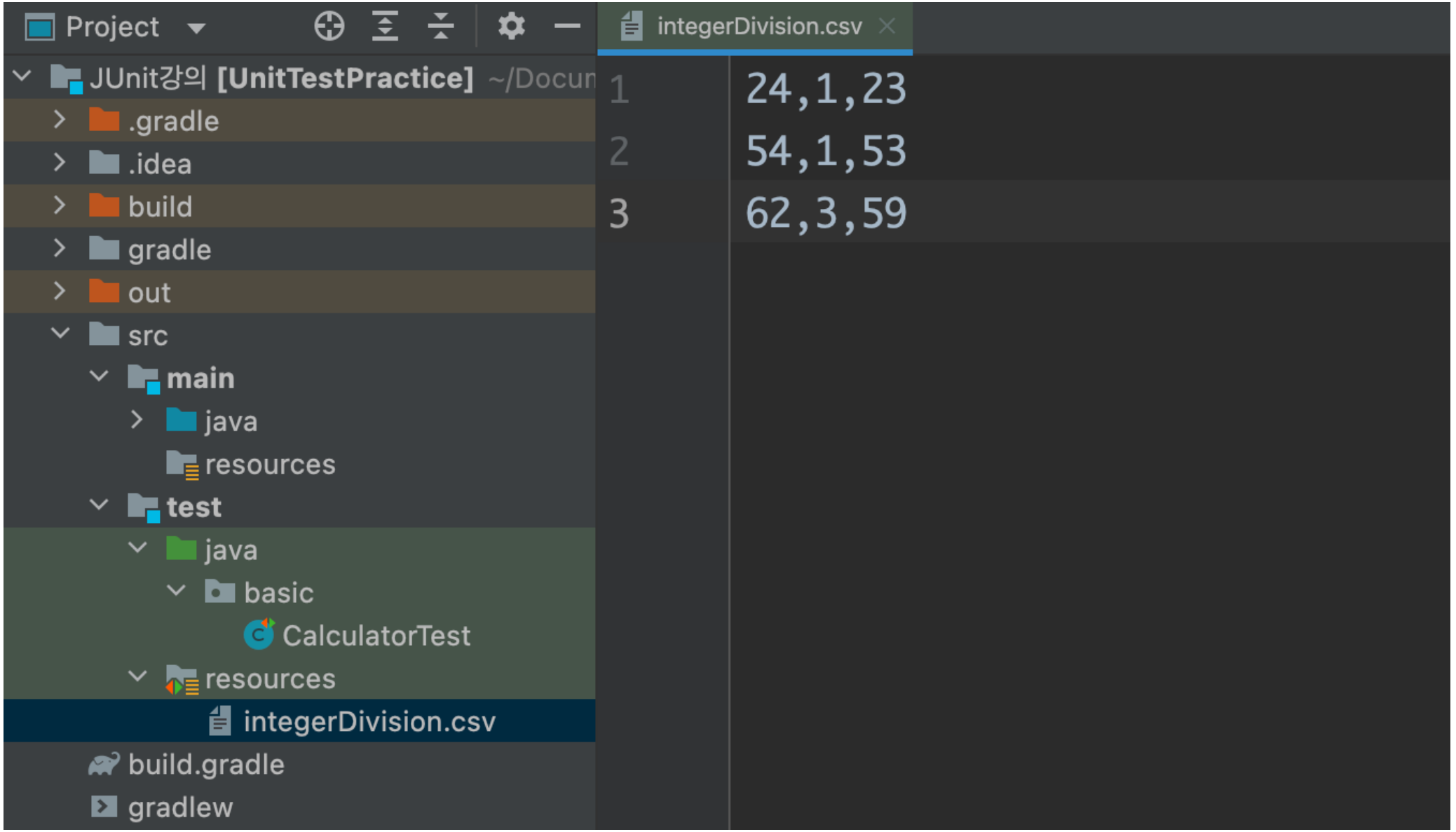
JUnit5 심화문법


```java
@DisplayName("빨셈 테스트 CsvSoruce")
@ParameterizedTest(name = "[{index}] {0} - {1} 테스트")
@CsvSource({
    "20, 17, 3",
    "17, 12, 5",
    "26, 24, 2"
})
void integerSubtraction_2(int minuend, int subtrahend, int expectedResult) {

  int result = calculator.integerSubtraction(minuend, subtrahend);
  assertEquals(expectedResult, result,
      () -> minuend + "-" + subtrahend + " 결과값이 " + expectedResult + "와 다릅니다.");
}
```

JUnit5 심화문법

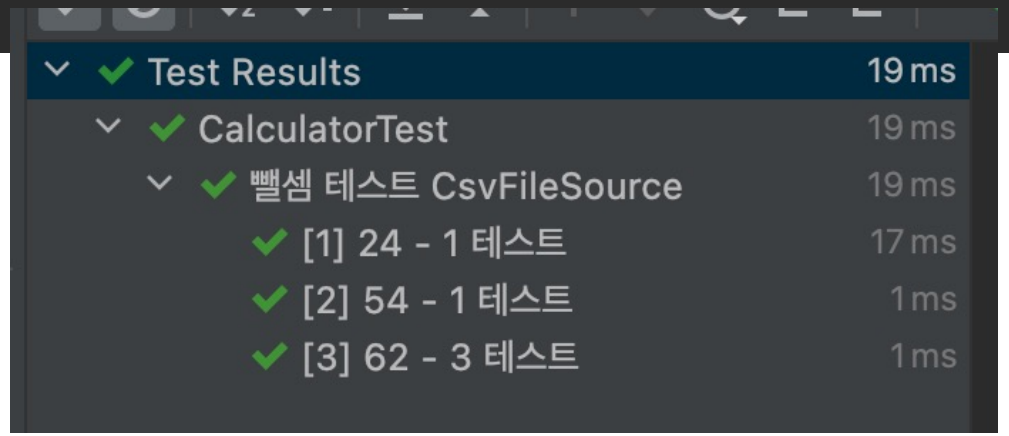

src
main
java
resources
test
java
basic
CalculatorTest
resources
build.gradle
gradlew
gradlew.bat
settings.gradle
External Libraries
Scratches and Consoles
New
Cut ⌘X
Copy ⌘C
Copy Path/Reference...
Paste ⌘V
Find Usages ⌥F7
Find in Files... ⇧⌘F
Replace in Files... ⇧⌘R
Analyze
Refactor
Bookmarks
Reformat Code ⌥⌘L
Optimize Imports ^⌥O
Delete...
Build Module 'UnitTestPractice.test'
Rebuild '<default>' ⇧⌘F9
Run 'All Tests' ^⇧R
Debug 'All Tests' ^⇧D
More Run/Debug
Open In
Local History
Kotlin Class/File
File
Scratch File ⇧⌘N
Directory
HTML File
Kotlin Script
Kotlin Worksheet
Stylesheet
JavaScript File
TypeScript File
package.json File
OpenAPI Specification
Swing UI Designer
Resource Bundle
EditorConfig File
XML Configuration File
HTTP Request
Diagram
Data Source in Path
Run:
CalculatorTest.inte
131
132
133
134
135
assertEqual
New File
integerDivision.csv

JUnit5 심화문법


Project
JUnit강의 [UnitTestPractice] ~/Docun
.gradle
.idea
build
gradle
out
src
main
java
resources
test
java
basic
CalculatorTest
resources
integerDivision.csv
build.gradle
gradlew
integerDivision.csv
1 24,1,23
2 54,1,53
3 62,3,59

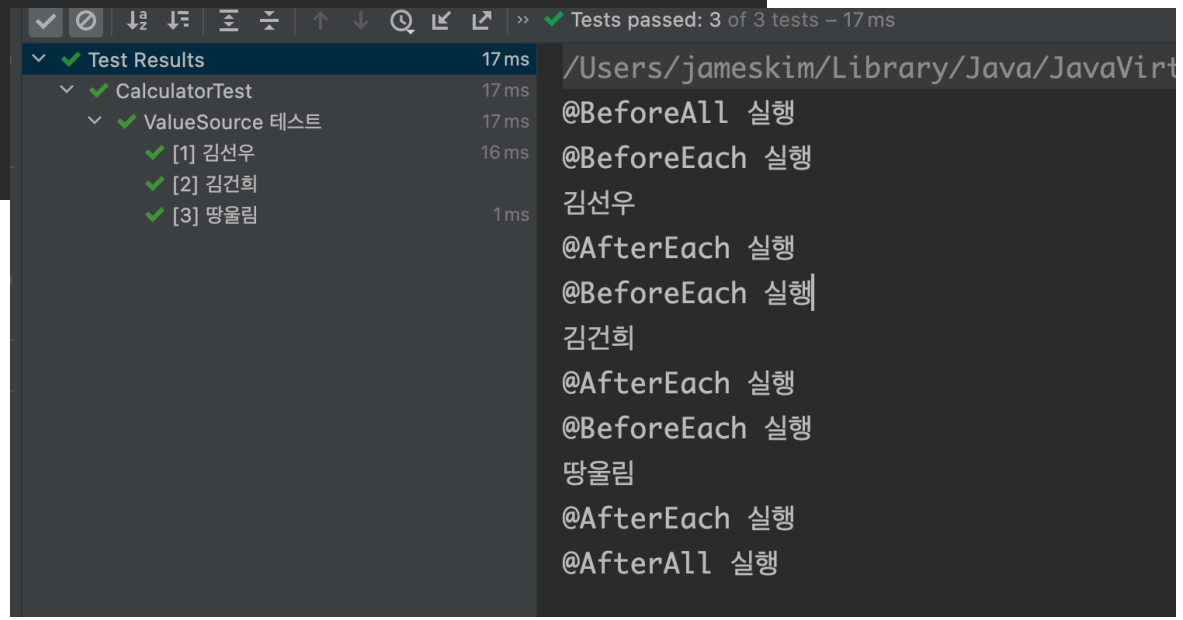
JUnit5 심화문법

```java
@DisplayName("빨셈 테스트 CsvFileSource")
@ParameterizedTest(name = "[{index}] {0} - {1} 테스트")
@CsvFileSource(resources = "/integerDivision.csv")
void integerSubtraction4(int minuend, int subtrahend, int expectedResult) {

    int result = calculator.integerSubtraction(minuend, subtrahend);

    assertEquals(expectedResult, result,
        () -> minuend + "-" + subtrahend + " 결과값이 " + expectedResult + "와 다릅니다.");
}
```

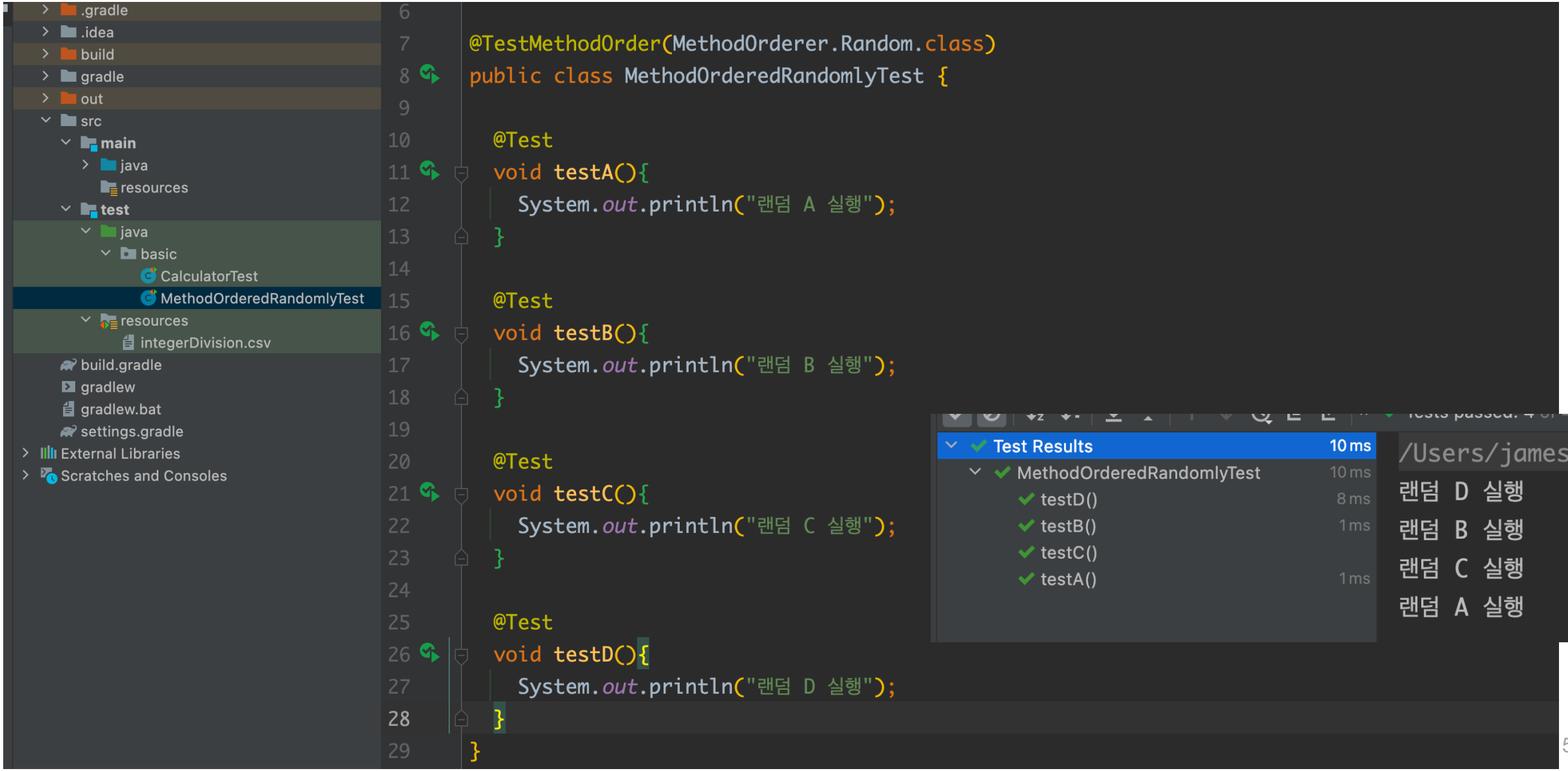


JUnit5 심화문법

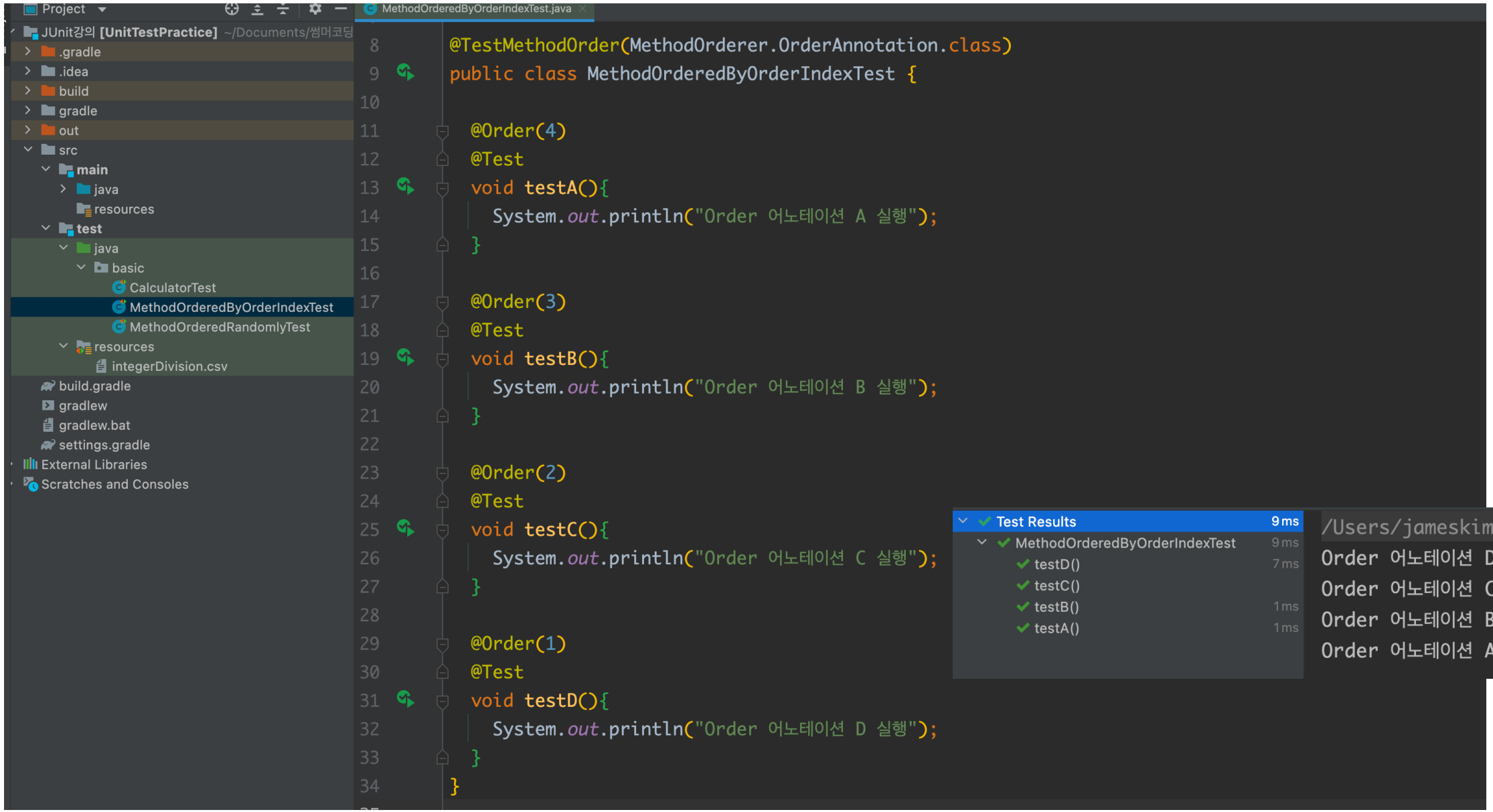
```java
@DisplayName("ValueSource 테스트")
@ParameterizedTest
@ValueSource(strings={"김선우", "김건희", "땅울림"})
void valueSourceDemonstration(String firstname){
    System.out.println(firstname);
    assertNotNull(firstname);
}
```



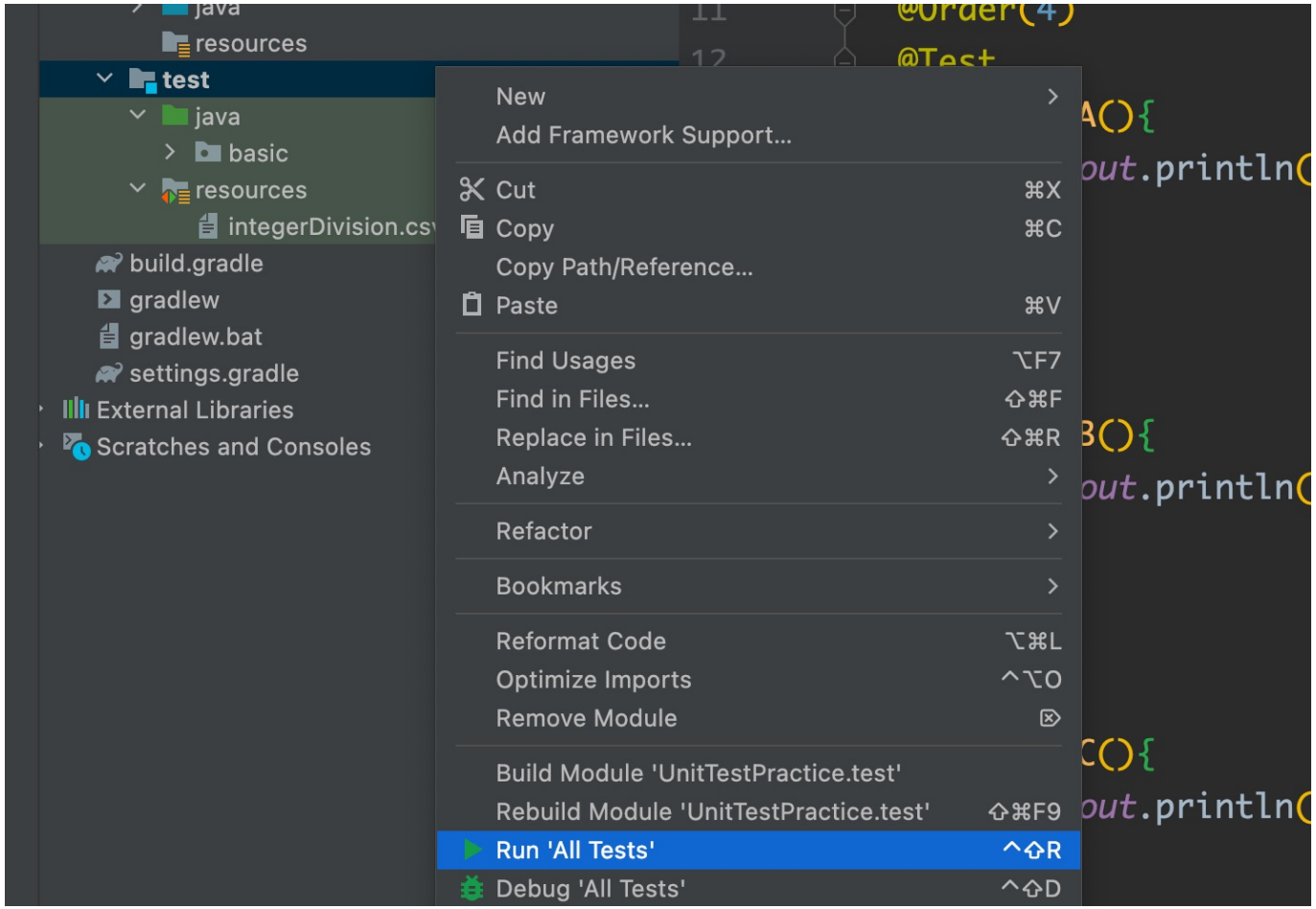
JUnit5 심화문법

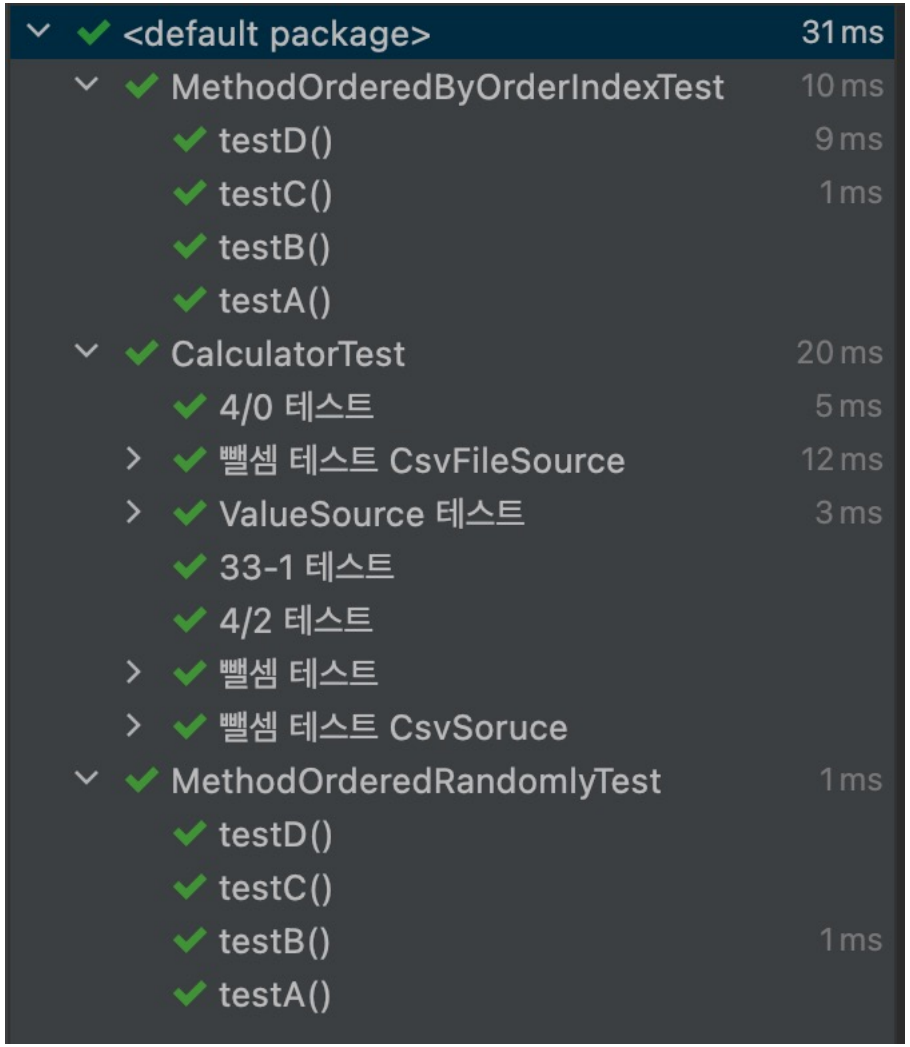

.gradle
.idea
build
gradle
out
src
main
java
resources
test
java
basic
CalculatorTest
MethodOrderedRandomlyTest
resources
integerDivision.csv
build.gradle
gradlew
gradlew.bat
settings.gradle
External Libraries
Scratches and Consoles
@TestMethodOrder(MethodOrderer.Random.class)
public class MethodOrderedRandomlyTest {
@Test
void testA(){
System.out.println("랜덤 A 실행");
}
@Test
void testB(){
System.out.println("랜덤 B 실행");
}
@Test
void testC(){
System.out.println("랜덤 C 실행");
}
@Test
void testD(){
System.out.println("랜덤 D 실행");
}
}
Test Results 10 ms
MethodOrderedRandomlyTest 10 ms
testD() 8 ms
testB() 1 ms
testC()
testA() 1 ms
/Users/jameskim/
랜덤 D 실행
랜덤 B 실행
랜덤 C 실행
랜덤 A 실행

JUnit5 심화문법


Project
JUnit강의 [UnitTestPractice] ~/Documents/썸머코딩
.gradle
.idea
build
gradle
out
src
main
java
resources
test
java
basic
CalculatorTest
MethodOrderedByOrderIndexTest
MethodOrderedRandomlyTest
resources
integerDivision.csv
build.gradle
gradlew
gradlew.bat
settings.gradle
External Libraries
Scratches and Consoles
MethodOrderedByOrderIndexTest.java
@TestMethodOrder(MethodOrderer.OrderAnnotation.class)
public class MethodOrderedByOrderIndexTest {
@Order(4)
@Test
void testA(){
System.out.println("Order 어노테이션 A 실행");
}
@Order(3)
@Test
void testB(){
System.out.println("Order 어노테이션 B 실행");
}
@Order(2)
@Test
void testC(){
System.out.println("Order 어노테이션 C 실행");
}
@Order(1)
@Test
void testD(){
System.out.println("Order 어노테이션 D 실행");
}
}
Test Results 9 ms
MethodOrderedByOrderIndexTest 9 ms
testD() 7 ms
testC()
testB() 1 ms
testA() 1 ms
/Users/jameskim/Library/Java/J
Order 어노테이션 D 실행
Order 어노테이션 C 실행
Order 어노테이션 B 실행
Order 어노테이션 A 실행

JUnit5 심화문법


resources
test
java
basic
resources
integerDivision.csv
build.gradle
gradlew
gradlew.bat
settings.gradle
External Libraries
Scratches and Consoles
New
Add Framework Support...
Cut ⌘X
Copy ⌘C
Copy Path/Reference...
Paste ⌘V
Find Usages ⌥F7
Find in Files... ⇧⌘F
Replace in Files... ⇧⌘R
Analyze
Refactor
Bookmarks
Reformat Code ⌥⌘L
Optimize Imports ^⌥O
Remove Module
Build Module 'UnitTestPractice.test'
Rebuild Module 'UnitTestPractice.test' ⇧⌘F9
Run 'All Tests' ^⇧R
Debug 'All Tests' ^⇧D
@Order(4)
@Test
out.println


<default package> 31 ms
MethodOrderedByOrderIndexTest 10 ms
testD() 9 ms
testC() 1 ms
testB()
testA()
CalculatorTest 20 ms
4/0 테스트 5 ms
뺄셈 테스트 CsvFileSource 12 ms
ValueSource 테스트 3 ms
33-1 테스트
4/2 테스트
뺄셈 테스트
뺄셈 테스트 CsvSoruce
MethodOrderedRandomlyTest 1 ms
testD()
testC()
testB() 1 ms
testA()

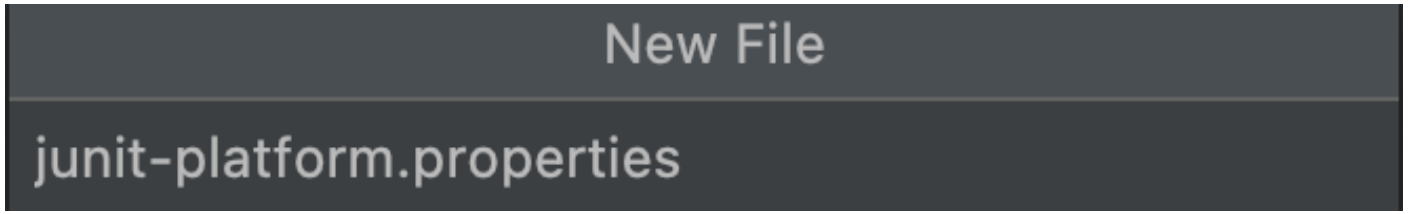
JUnit5 심화문법

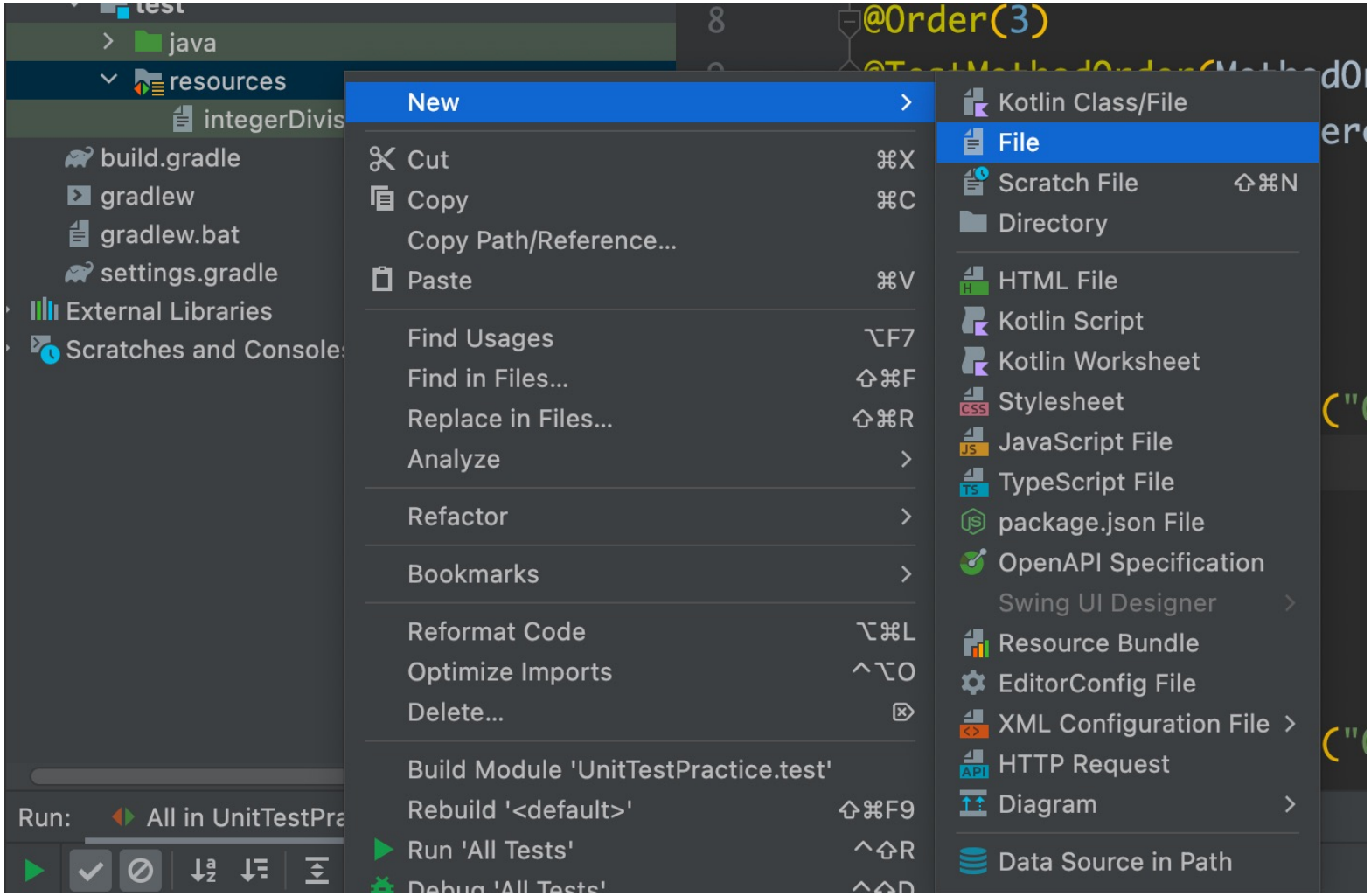
```
@Order(1)
@TestMethodOrder(MethodOrderer.Random.class)
public class MethodOrderedRandomlyTest {
```

```
@Order(2)
public class CalculatorTest {
```

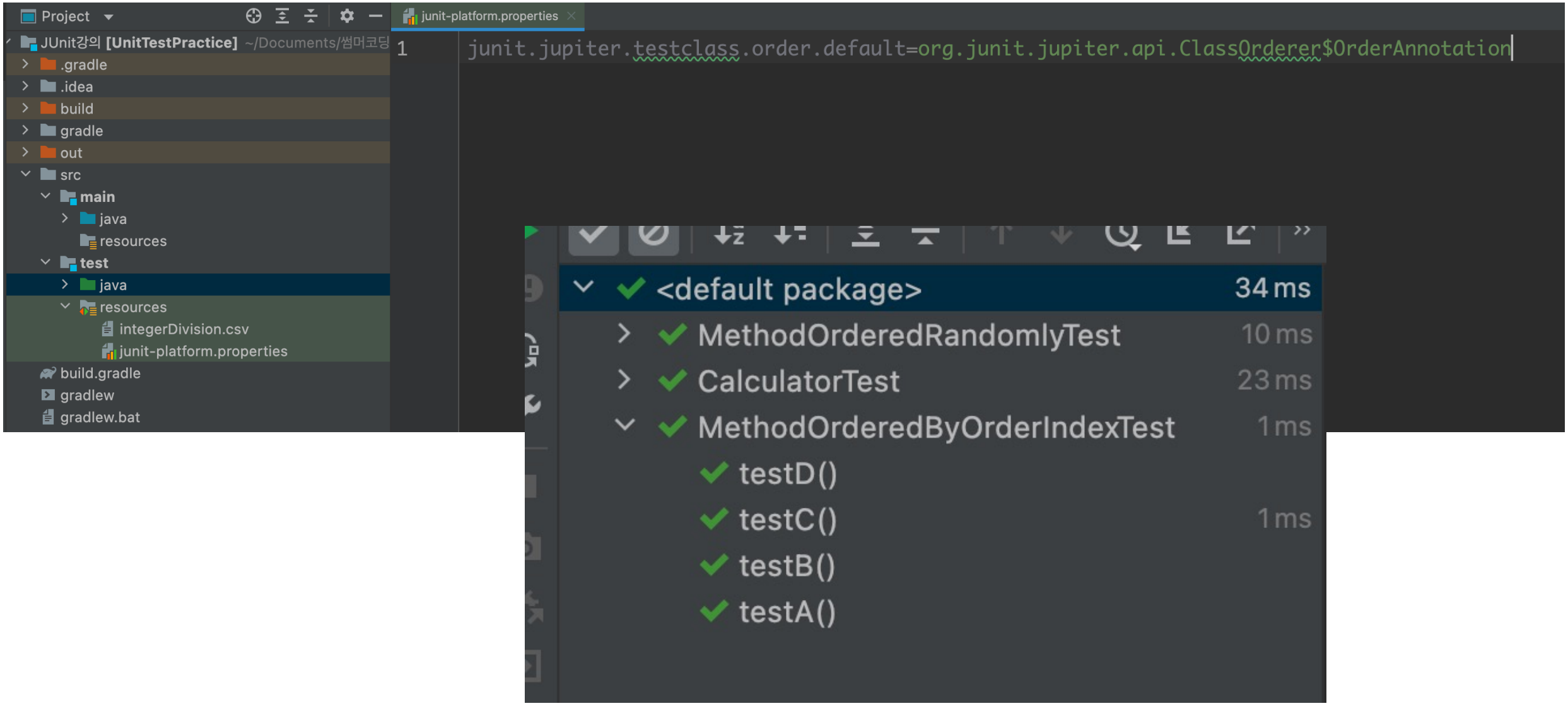
```
@Order(3)
@TestMethodOrder(MethodOrderer.OrderAnnotation.class)
public class MethodOrderedByOrderIndexTest {
```

JUnit5 심화문법

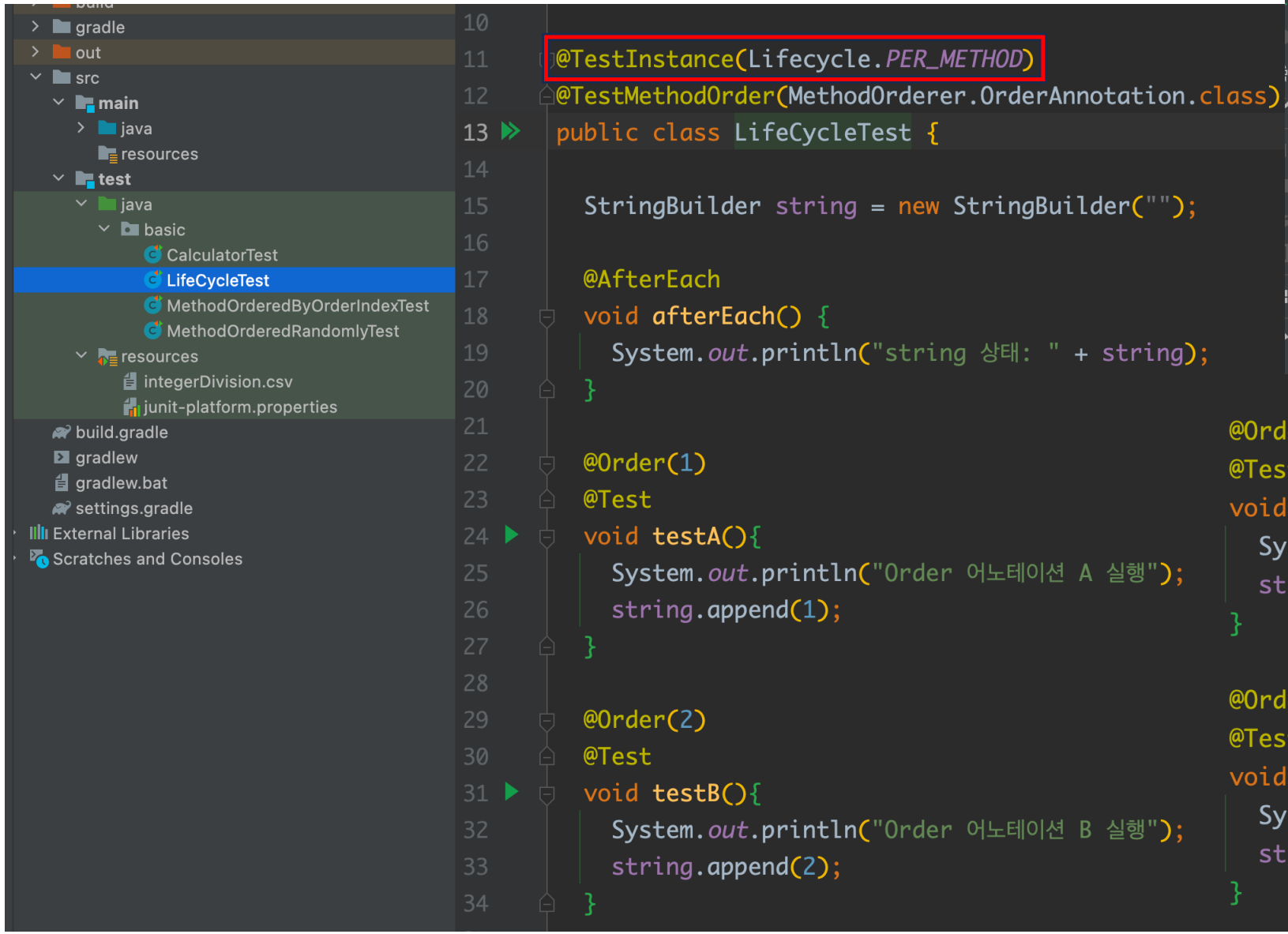

New File
junit-platform.properties


java
resources
integerDivis
build.gradle
gradlew
gradlew.bat
settings.gradle
External Libraries
Scratches and Console
Run:
All in UnitTestPra
@Order(3)
New
Cut ⌘X
Copy ⌘C
Copy Path/Reference...
Paste ⌘V
Find Usages ⌥F7
Find in Files... ⇧⌘F
Replace in Files... ⇧⌘R
Analyze
Refactor
Bookmarks
Reformat Code ⌥⌘L
Optimize Imports ^⌥O
Delete...
Build Module 'UnitTestPractice.test'
Rebuild '<default>' ⇧⌘F9
Run 'All Tests' ^⇧R
Kotlin Class/File
File
Scratch File ⇧⌘N
Directory
HTML File
Kotlin Script
Kotlin Worksheet
Stylesheet
JavaScript File
TypeScript File
package.json File
OpenAPI Specification
Swing UI Designer
Resource Bundle
EditorConfig File
XML Configuration File
HTTP Request
Diagram
Data Source in Path

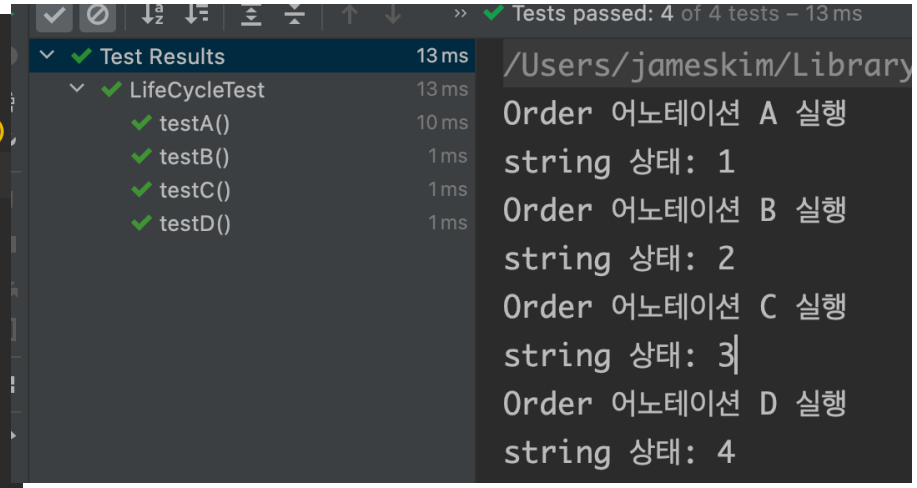
JUnit5 심화문법


Project
JUnit강의 [UnitTestPractice] ~/Documents/썸머코딩
.gradle
.idea
build
gradle
out
src
main
java
resources
test
java
resources
integerDivision.csv
junit-platform.properties
build.gradle
gradlew
gradlew.bat
junit-platform.properties
1
junit.jupiter.testclass.order.default=org.junit.jupiter.api.ClassOrderer$OrderAnnotation
<default package> 34 ms
MethodOrderedRandomlyTest 10 ms
CalculatorTest 23 ms
MethodOrderedByOrderIndexTest 1 ms
testD()
testC() 1 ms
testB()
testA()

JUnit5 심화문법



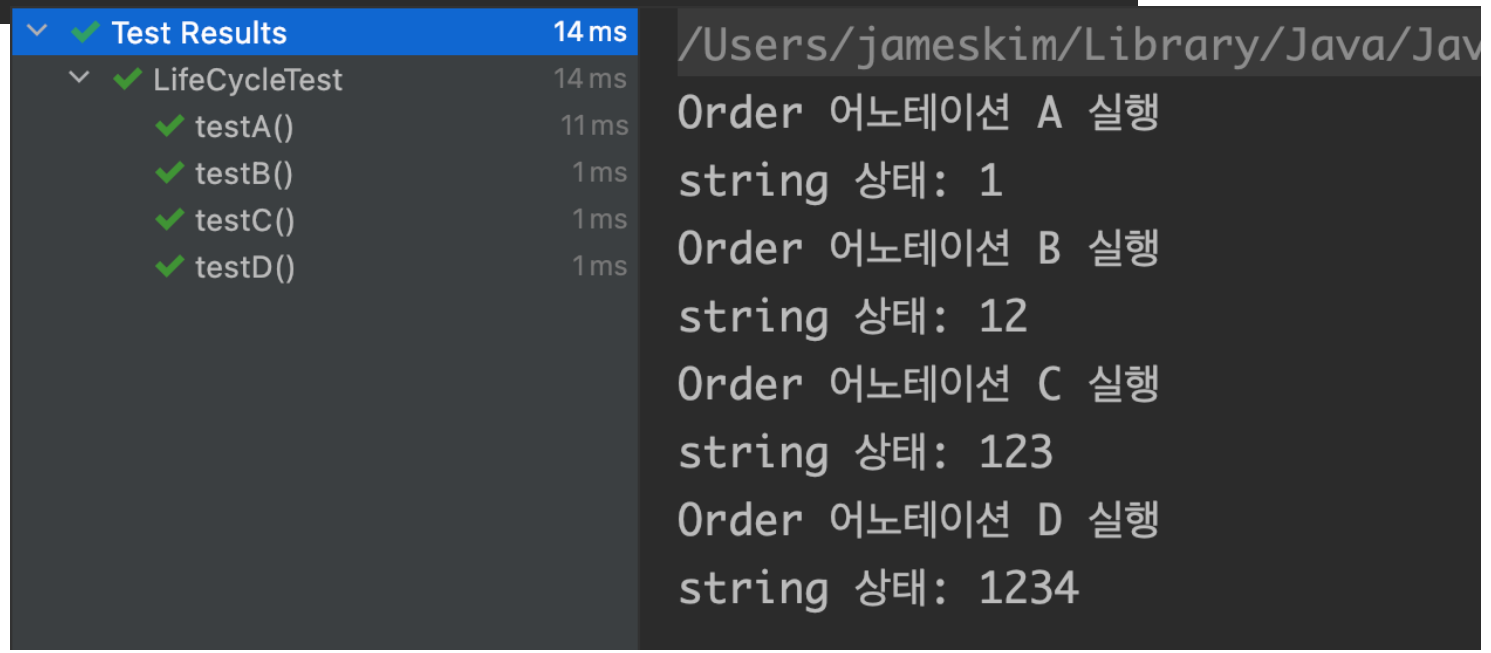

```java
@TestInstance(Lifecycle.PER_METHOD)
@TestMethodOrder(MethodOrderer.OrderAnnotation.class)
public class LifeCycleTest {

    StringBuilder string = new StringBuilder("");

    @AfterEach
    void afterEach() {
        System.out.println("string 상태: " + string);
    }

    @Order(1)
    @Test
    void testA(){
        System.out.println("Order 어노테이션 A 실행");
        string.append(1);
    }

    @Order(2)
    @Test
    void testB(){
        System.out.println("Order 어노테이션 B 실행");
        string.append(2);
    }
```




```java
    @Order(3)
    @Test
    void testC(){
        System.out.println("Order 어노테이션 C 실행");
        string.append(3);
    }

    @Order(4)
    @Test
    void testD(){
        System.out.println("Order 어노테이션 D 실행");
        string.append(4);
    }
```

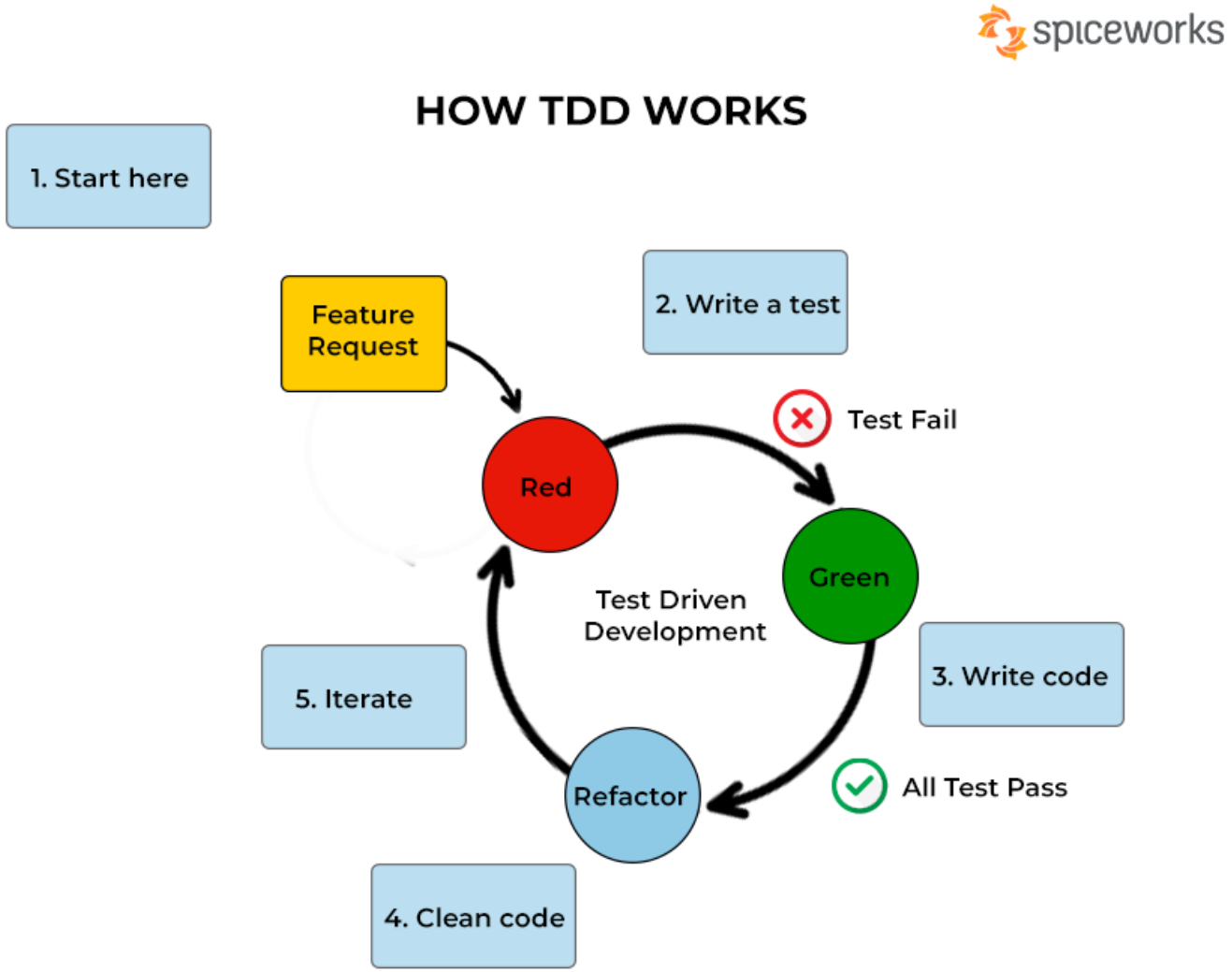
JUnit5 심화문법

```java
@TestInstance(Lifecycle.PER_CLASS)
@TestMethodOrder(MethodOrderer.OrderAnnotation.class)
public class LifeCycleTest {

    StringBuilder string = new StringBuilder("");
```



04
테스트 주도 개발(TDD)

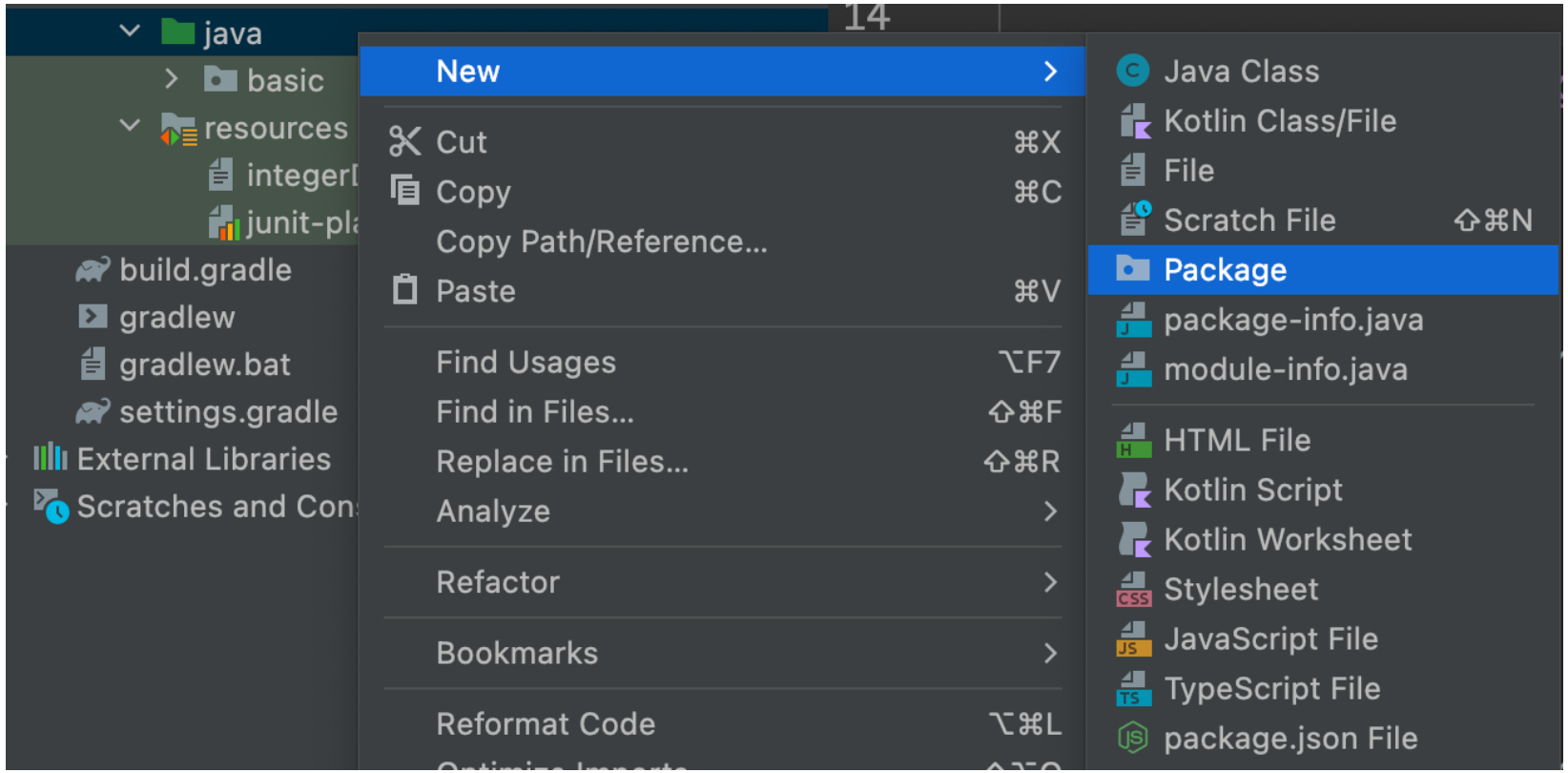
테스트 주도 개발(TDD)

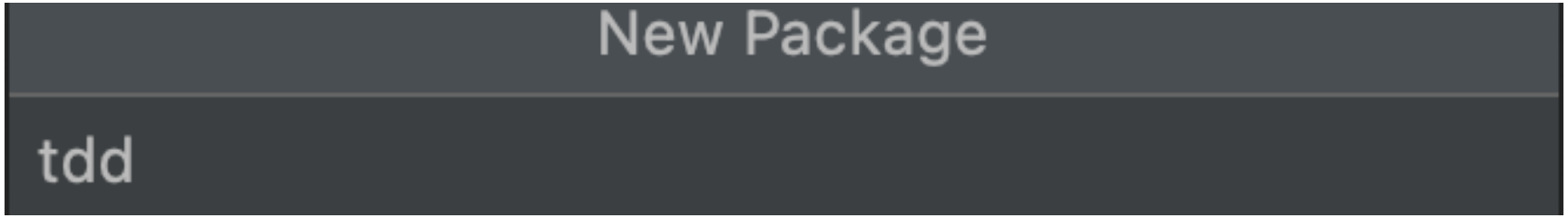

spiceworks
HOW TDD WORKS
1. Start here
Feature Request
2. Write a test
Test Fail
Red
Green
Test Driven Development
3. Write code
All Test Pass
Refactor
4. Clean code
5. Iterate

요구사항

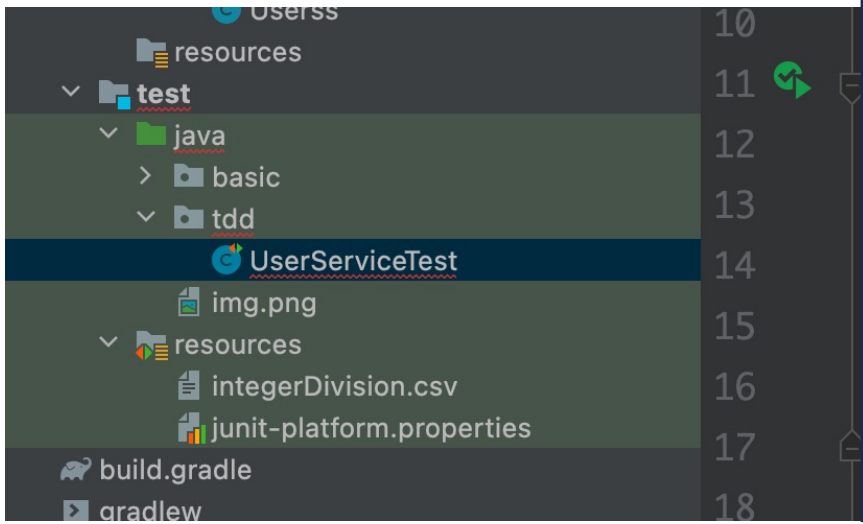
회원가입을 구현하시오
회원가입시 이름, 이메일, 비밀번호만 요구

테스트 주도 개발(TDD)


java
basic
resources
integerI
junit-pla
build.gradle
gradlew
gradlew.bat
settings.gradle
External Libraries
Scratches and Cons
New
Cut ⌘X
Copy ⌘C
Copy Path/Reference...
Paste ⌘V
Find Usages ⌥F7
Find in Files... ⇧⌘F
Replace in Files... ⇧⌘R
Analyze
Refactor
Bookmarks
Reformat Code ⌥⌘L
Java Class
Kotlin Class/File
File
Scratch File ⇧⌘N
Package
package-info.java
module-info.java
HTML File
Kotlin Script
Kotlin Worksheet
Stylesheet
JavaScript File
TypeScript File
package.json File

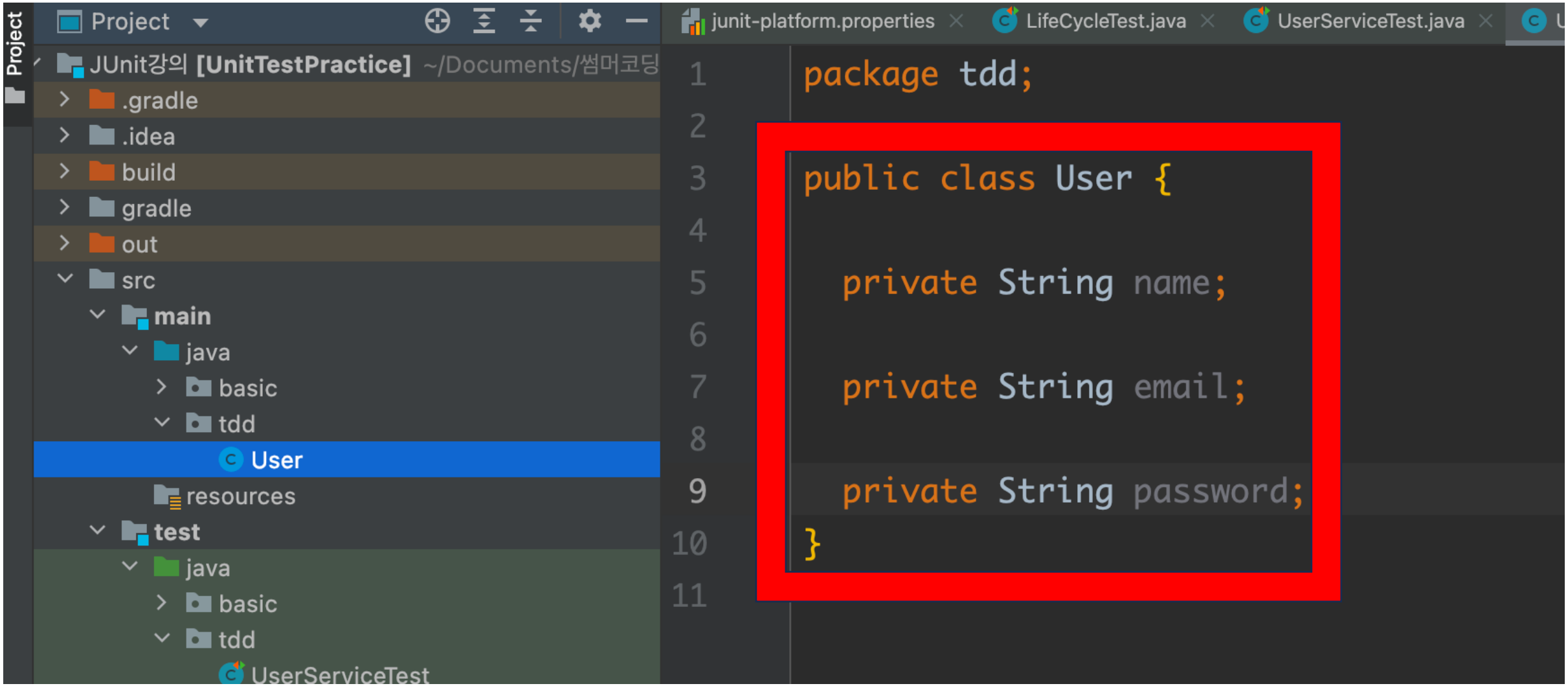

New Package
tdd

테스트 주도 개발(TDD)

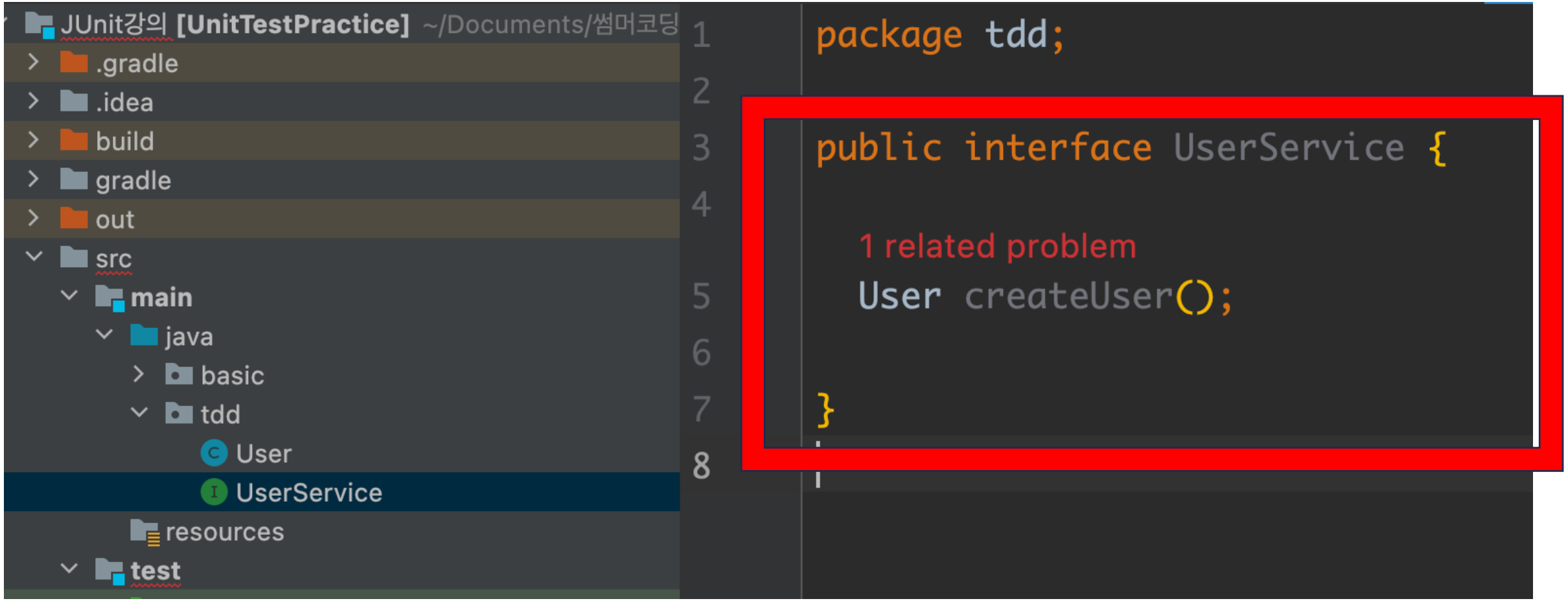



```
@Test
public void createUser_whenUserDetailsProvided_returnUserObject(){

  User user = userService.createUser();

  assertNotNull(user, "회원가입한 사용자는 null이면 안됩니다.");

}
```

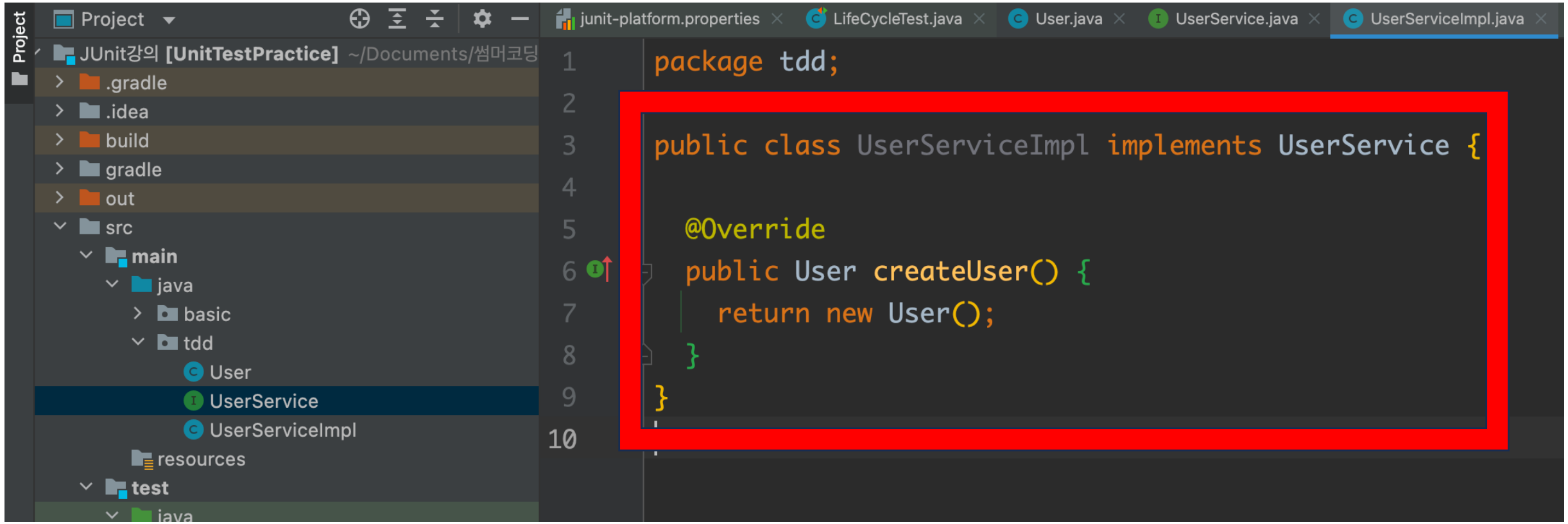
테스트 주도 개발(TDD)


Project
JUnit강의 [UnitTestPractice] ~/Documents/썸머코딩
.gradle
.idea
build
gradle
out
src
main
java
basic
tdd
User
resources
test
java
basic
tdd
UserServiceTest
junit-platform.properties
LifeCycleTest.java
UserServiceTest.java
package tdd;
public class User {
private String name;
private String email;
private String password;
}

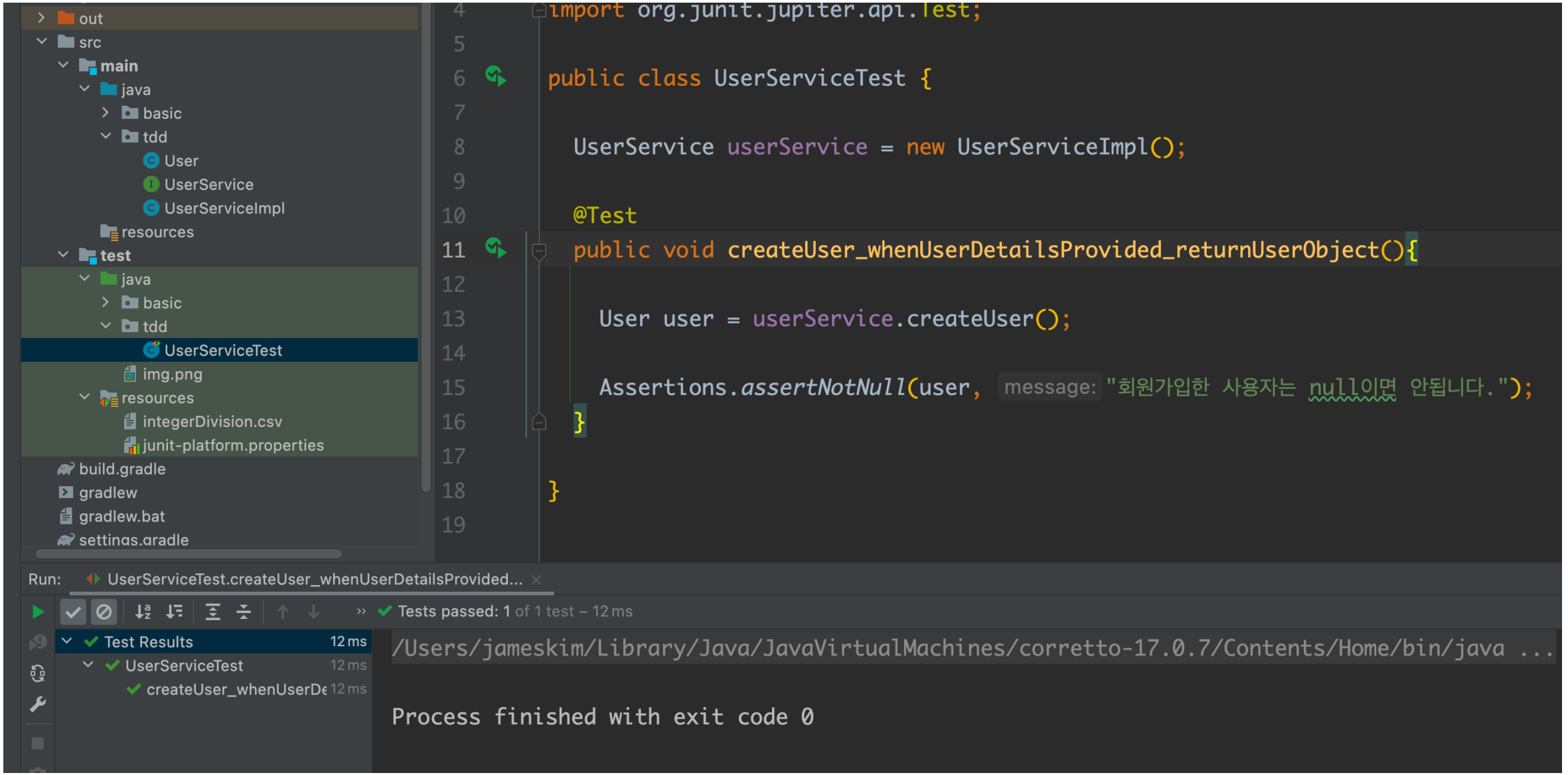
테스트 주도 개발(TDD)


JUnit강의 [UnitTestPractice] ~/Documents/썸머코딩
.gradle
.idea
build
gradle
out
src
main
java
basic
tdd
User
UserService
resources
test
package tdd;
public interface UserService {
1 related problem
User createUser();
}

테스트 주도 개발(TDD)


Project
JUnit강의 [UnitTestPractice] ~/Documents/썸머코딩
.gradle
.idea
build
gradle
out
src
main
java
basic
tdd
User
UserService
UserServiceImpl
resources
test
java
junit-platform.properties
LifeCycleTest.java
User.java
UserService.java
UserServiceImpl.java
package tdd;
public class UserServiceImpl implements UserService {
@Override
public User createUser() {
return new User();
}
}

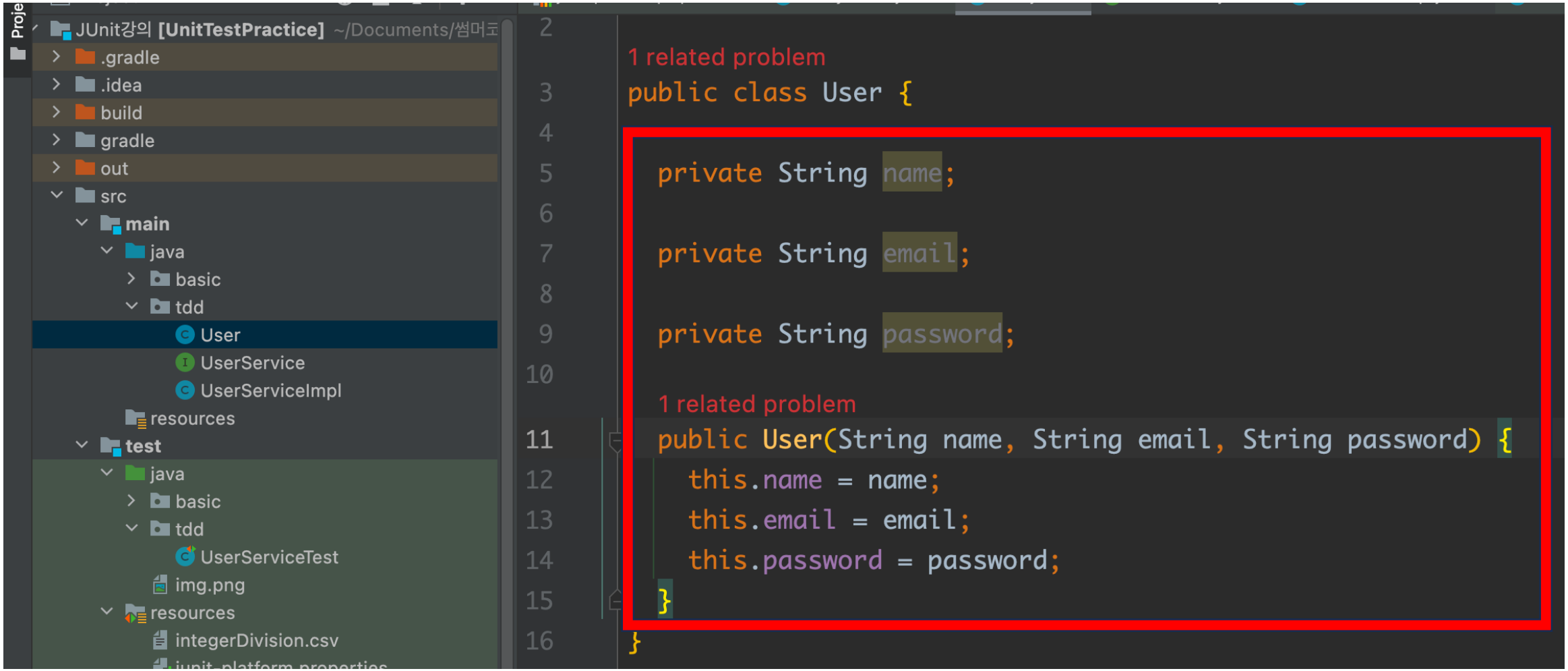
테스트 주도 개발(TDD)


out
src
main
java
basic
tdd
User
UserService
UserServiceImpl
resources
test
java
basic
tdd
UserServiceTest
img.png
resources
integerDivision.csv
junit-platform.properties
build.gradle
gradlew
gradlew.bat
settings.gradle
import org.junit.jupiter.api.Test;
public class UserServiceTest {
UserService userService = new UserServiceImpl();
@Test
public void createUser_whenUserDetailsProvided_returnUserObject(){
User user = userService.createUser();
Assertions.assertNotNull(user, message: "회원가입한 사용자는 null이면 안됩니다.");
}
}
Run: UserServiceTest.createUser_whenUserDetailsProvided...
Tests passed: 1 of 1 test – 12 ms
Test Results 12 ms
UserServiceTest 12 ms
createUser_whenUserDe 12 ms
/Users/jameskim/Library/Java/JavaVirtualMachines/corretto-17.0.7/Contents/Home/bin/java ...
Process finished with exit code 0

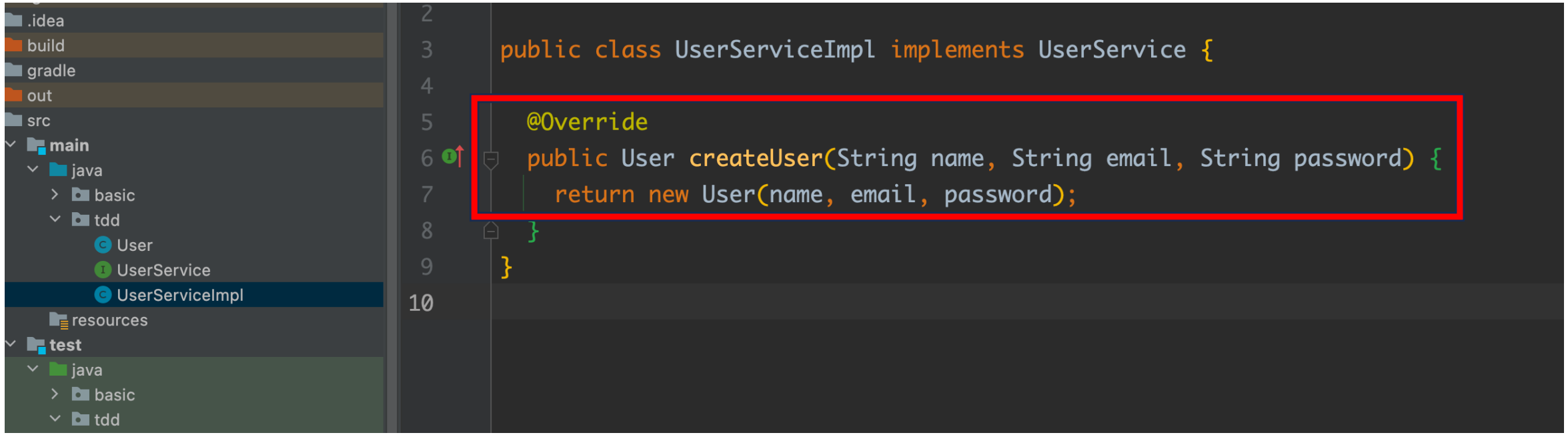
테스트 주도 개발(TDD)

```
@Test
public void createUser_whenUserDetailsProvided_returnUserObject(){

    String name = "홍길동";
    String email = "gildong@naver.com";
    String password = "qwerty123";

    User user = userService.createUser(name, email, password);

    Assertions.assertNotNull(user, message: "회원가입한 사용자는 null이면 안됩니다.");
}
}
```

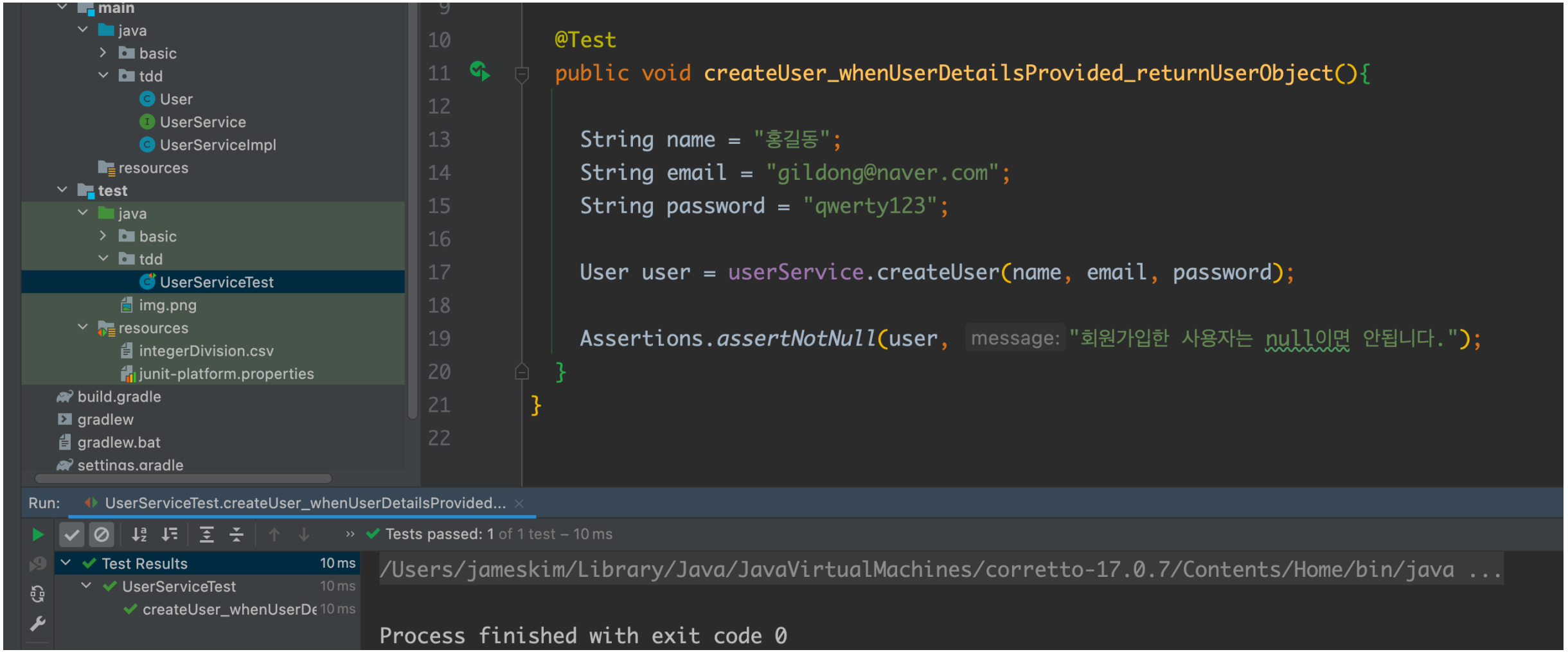
테스트 주도 개발(TDD)


JUnit강의 [UnitTestPractice] ~/Documents/썸머코
.gradle
.idea
build
gradle
out
src
main
java
basic
tdd
User
UserService
UserServiceImpl
resources
test
java
basic
tdd
UserServiceTest
img.png
resources
integerDivision.csv
1 related problem
public class User {
private String name;
private String email;
private String password;
1 related problem
public User(String name, String email, String password) {
this.name = name;
this.email = email;
this.password = password;
}
}

테스트 주도 개발(TDD)


.idea
build
gradle
out
src
main
java
basic
tdd
User
UserService
UserServiceImpl
resources
test
java
basic
tdd
public class UserServiceImpl implements UserService {
@Override
public User createUser(String name, String email, String password) {
return new User(name, email, password);
}
}

테스트 주도 개발(TDD)


main
java
basic
tdd
User
UserService
UserServiceImpl
resources
test
java
basic
tdd
UserServiceTest
img.png
resources
integerDivision.csv
junit-platform.properties
build.gradle
gradlew
gradlew.bat
settings.gradle
@Test
public void createUser_whenUserDetailsProvided_returnUserObject(){
String name = "홍길동";
String email = "gildong@naver.com";
String password = "qwerty123";
User user = userService.createUser(name, email, password);
Assertions.assertNotNull(user, message: "회원가입한 사용자는 null이면 안됩니다.");
}
}
Run: UserServiceTest.createUser_whenUserDetailsProvided...
Tests passed: 1 of 1 test – 10 ms
Test Results 10 ms
UserServiceTest 10 ms
createUser_whenUserDe 10 ms
/Users/jameskim/Library/Java/JavaVirtualMachines/corretto-17.0.7/Contents/Home/bin/java ...
Process finished with exit code 0

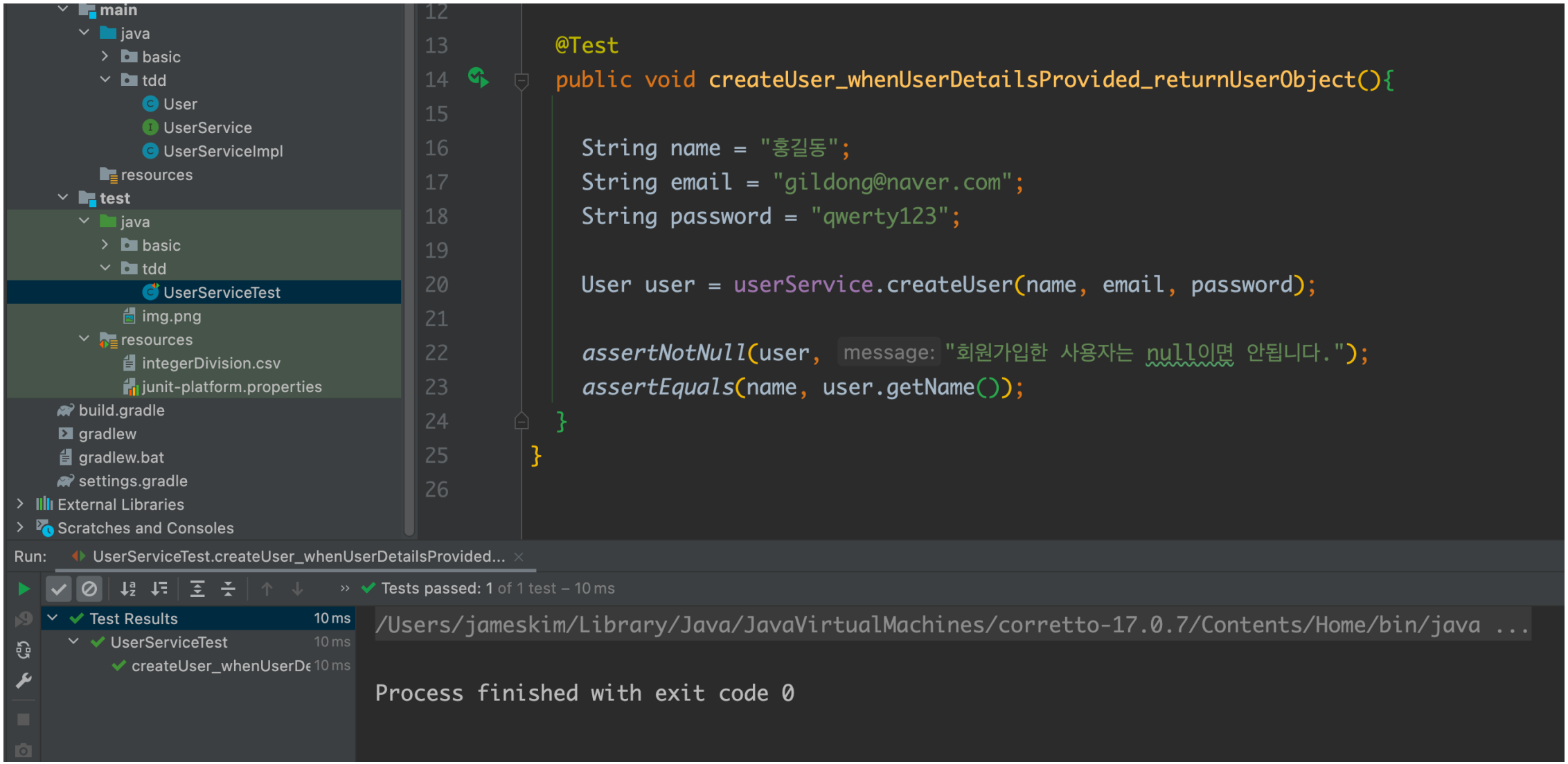
테스트 주도 개발(TDD)

```java
@Test
public void createUser_whenUserDetailsProvided_returnUserObject(){

    String name = "홍길동";
    String email = "gildong@naver.com";
    String password = "qwerty123";

    User user = userService.createUser(name, email, password);

    assertNotNull(user, message: "회원가입한 사용자는 null이면 안됩니다.");
    assertEquals(name, user.getName());
}
```

테스트 주도 개발(TD

```java
public class User {

  private String name;

  private String email;

  private String password;

  public User(String name, String email, String password) {
    this.name = name;
    this.email = email;
    this.password = password;
  }

  public String getName() {
    return name;
  }
}
```

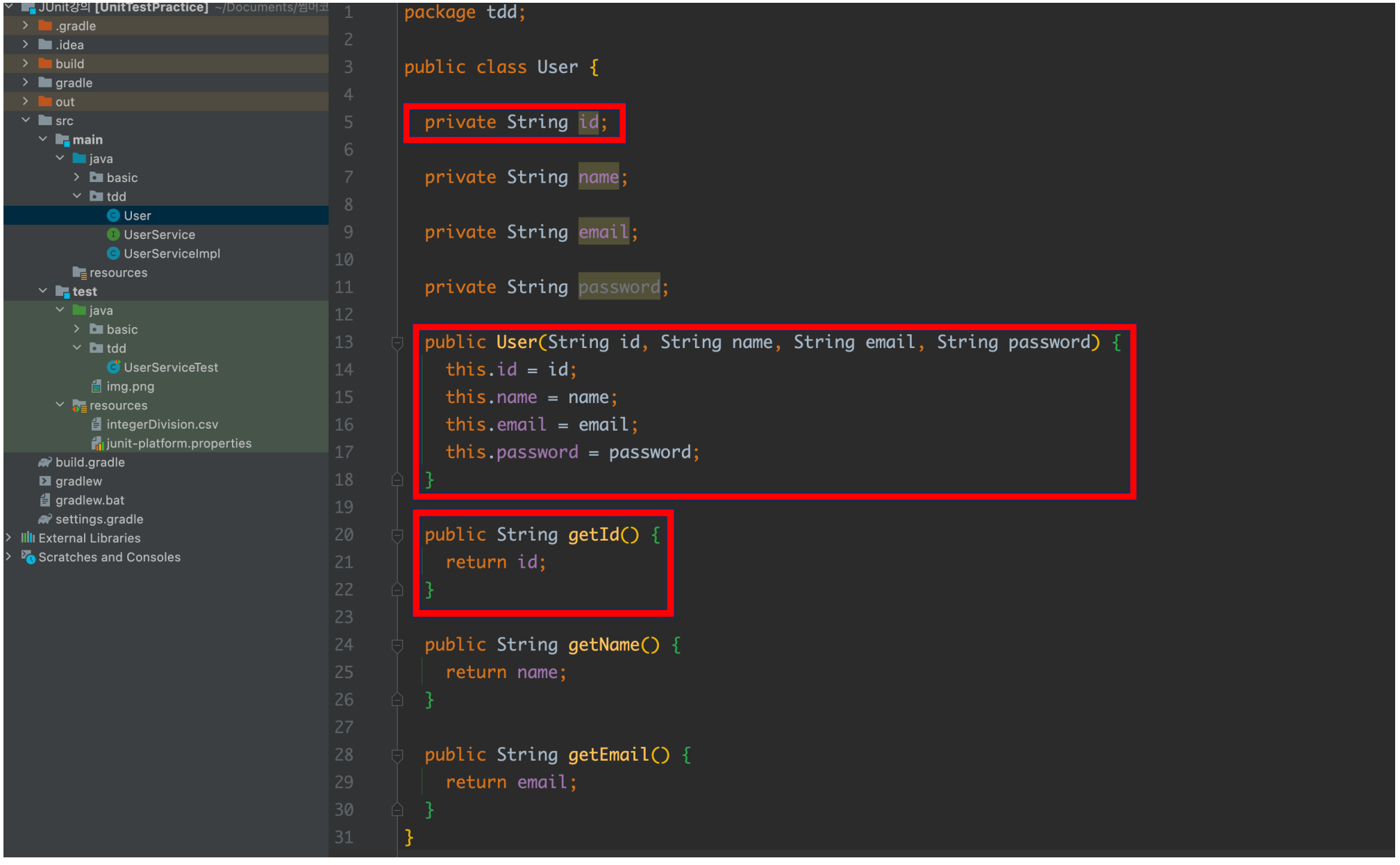
테스트 주도 개발(TDD)


main
java
basic
tdd
User
UserService
UserServiceImpl
resources
test
java
basic
tdd
UserServiceTest
img.png
resources
integerDivision.csv
junit-platform.properties
build.gradle
gradlew
gradlew.bat
settings.gradle
External Libraries
Scratches and Consoles
@Test
public void createUser_whenUserDetailsProvided_returnUserObject(){
String name = "홍길동";
String email = "gildong@naver.com";
String password = "qwerty123";
User user = userService.createUser(name, email, password);
assertNotNull(user, message: "회원가입한 사용자는 null이면 안됩니다.");
assertEquals(name, user.getName());
}
}
Run: UserServiceTest.createUser_whenUserDetailsProvided...
Tests passed: 1 of 1 test – 10 ms
Test Results 10 ms
UserServiceTest 10 ms
createUser_whenUserDe 10 ms
/Users/jameskim/Library/Java/JavaVirtualMachines/corretto-17.0.7/Contents/Home/bin/java ...
Process finished with exit code 0

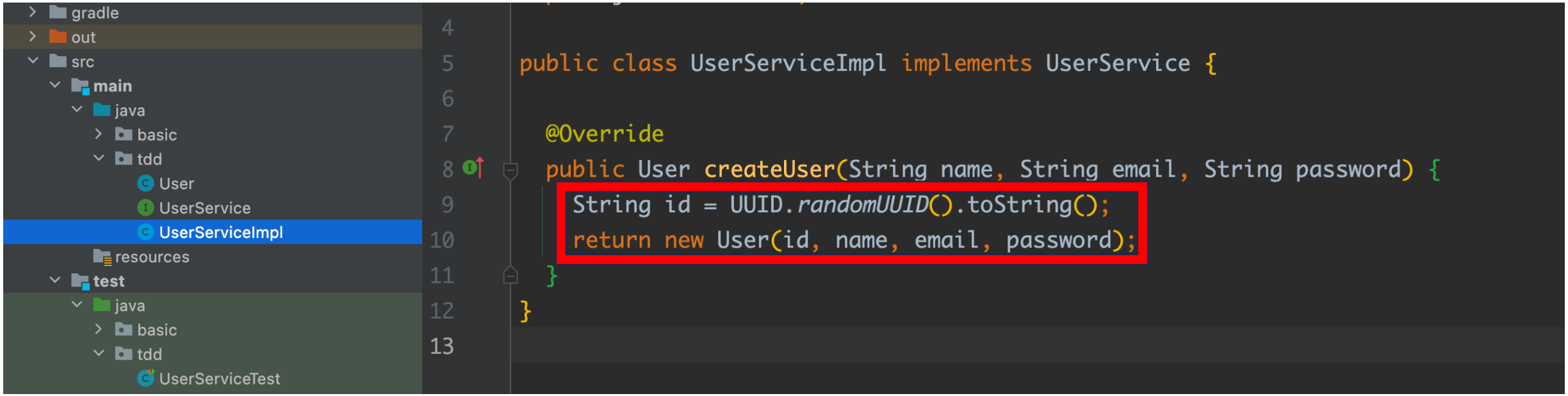
테스트 주도 개발(TDD)

```
@Test
public void createUser_whenUserDetailsProvided_returnUserObject(){

  String name = "홍길동";
  String email = "gildong@naver.com";
  String password = "qwerty123";

  User user = userService.createUser(name, email, password);

  assertNotNull(user, message: "회원가입한 사용자는 null이면 안됩니다.");
  assertEquals(name, user.getName());
  assertEquals(email, user.getEmail());
  assertNotNull(user.getId(), "사용자 id가 존재하지 않습니다.");
}
```

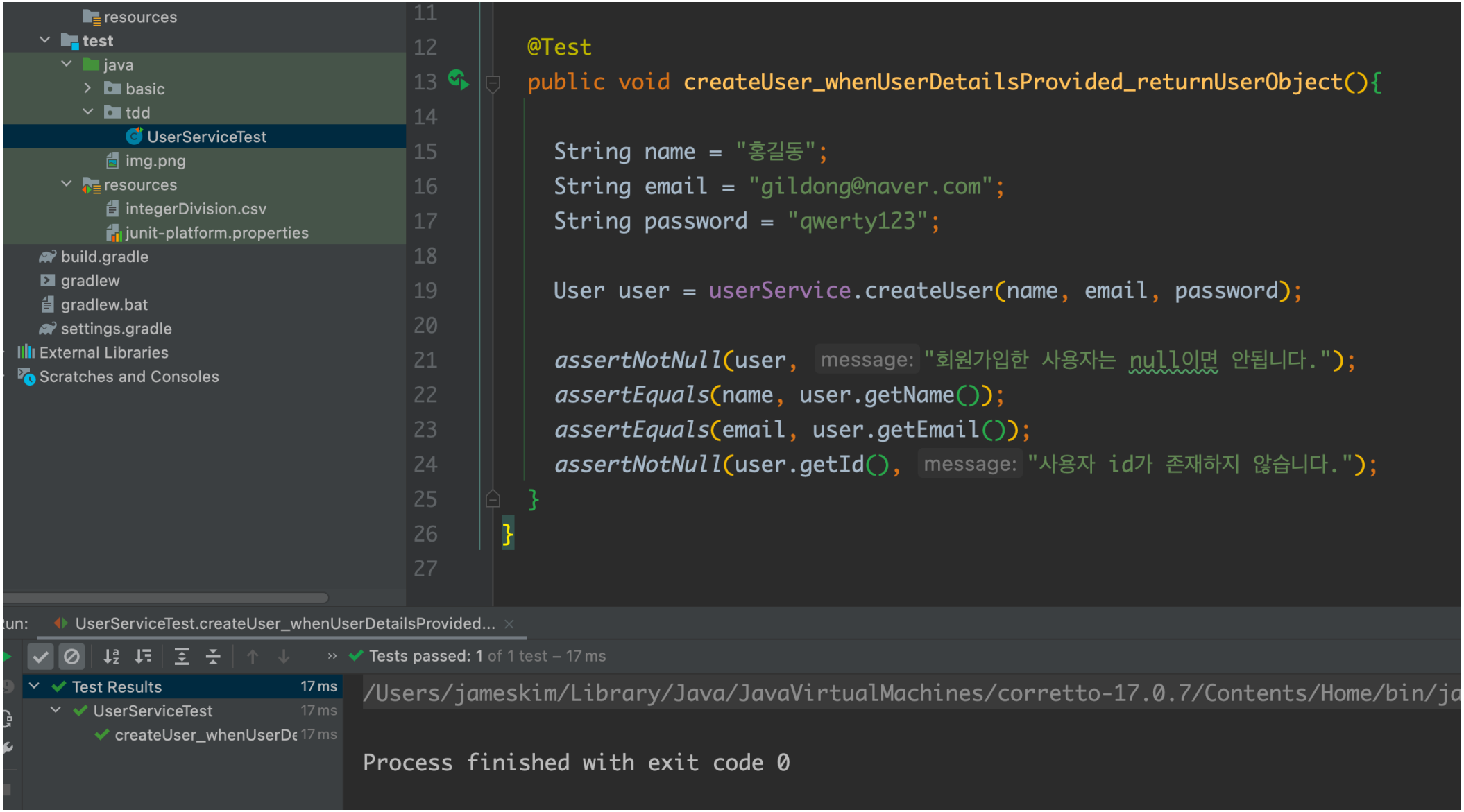
테스트 주도 개발(TDD)


JUnit강의 [UnitTestPractice] ~/Documents/쌤머프
.gradle
.idea
build
gradle
out
src
main
java
basic
tdd
User
UserService
UserServiceImpl
resources
test
java
basic
tdd
UserServiceTest
img.png
resources
integerDivision.csv
junit-platform.properties
build.gradle
gradlew
gradlew.bat
settings.gradle
External Libraries
Scratches and Consoles
package tdd;
public class User {
private String id;
private String name;
private String email;
private String password;
public User(String id, String name, String email, String password) {
this.id = id;
this.name = name;
this.email = email;
this.password = password;
}
public String getId() {
return id;
}
public String getName() {
return name;
}
public String getEmail() {
return email;
}
}

테스트 주도 개발(TDD)


gradle
out
src
main
java
basic
tdd
User
UserService
UserServiceImpl
resources
test
java
basic
tdd
UserServiceTest
public class UserServiceImpl implements UserService {
@Override
public User createUser(String name, String email, String password) {
String id = UUID.randomUUID().toString();
return new User(id, name, email, password);
}
}

테스트 주도 개발(TDD)


resources
test
java
basic
tdd
UserServiceTest
img.png
resources
integerDivision.csv
junit-platform.properties
build.gradle
gradlew
gradlew.bat
settings.gradle
External Libraries
Scratches and Consoles
@Test
public void createUser_whenUserDetailsProvided_returnUserObject(){
String name = "홍길동";
String email = "gildong@naver.com";
String password = "qwerty123";
User user = userService.createUser(name, email, password);
assertNotNull(user, message: "회원가입한 사용자는 null이면 안됩니다.");
assertEquals(name, user.getName());
assertEquals(email, user.getEmail());
assertNotNull(user.getId(), message: "사용자 id가 존재하지 않습니다.");
}
}
UserServiceTest.createUser_whenUserDetailsProvided...
Tests passed: 1 of 1 test – 17 ms
Test Results 17 ms
UserServiceTest 17 ms
createUser_whenUserDe 17 ms
/Users/jameskim/Library/Java/JavaVirtualMachines/corretto-17.0.7/Contents/Home/bin/ja
Process finished with exit code 0

테스트 주도 개발(TDD)

직접 해보세요!

```
@DisplayName("사용자 생성시 빈 이름 추가")
@Test
void createUser_whenNameIsEmpty_throwsIllegalArgumentException() {




}
```

테스트 주도 개발(TDD)

```
public class UserServiceTest {

    UserService userService = new UserServiceImpl();

    String name = "홍길동";
    String email = "gildong@naver.com"
    String password = "qwerty123";

    @DisplayName("사용자 생성")
    @Test
    public void createUser_whenUserDetailsProvided_returnUserObject() {

        User user = userService.createUser(name, email, password);

        assertNotNull(user, message: "회원가입한 사용자는 null이면 안됩니다.");
```

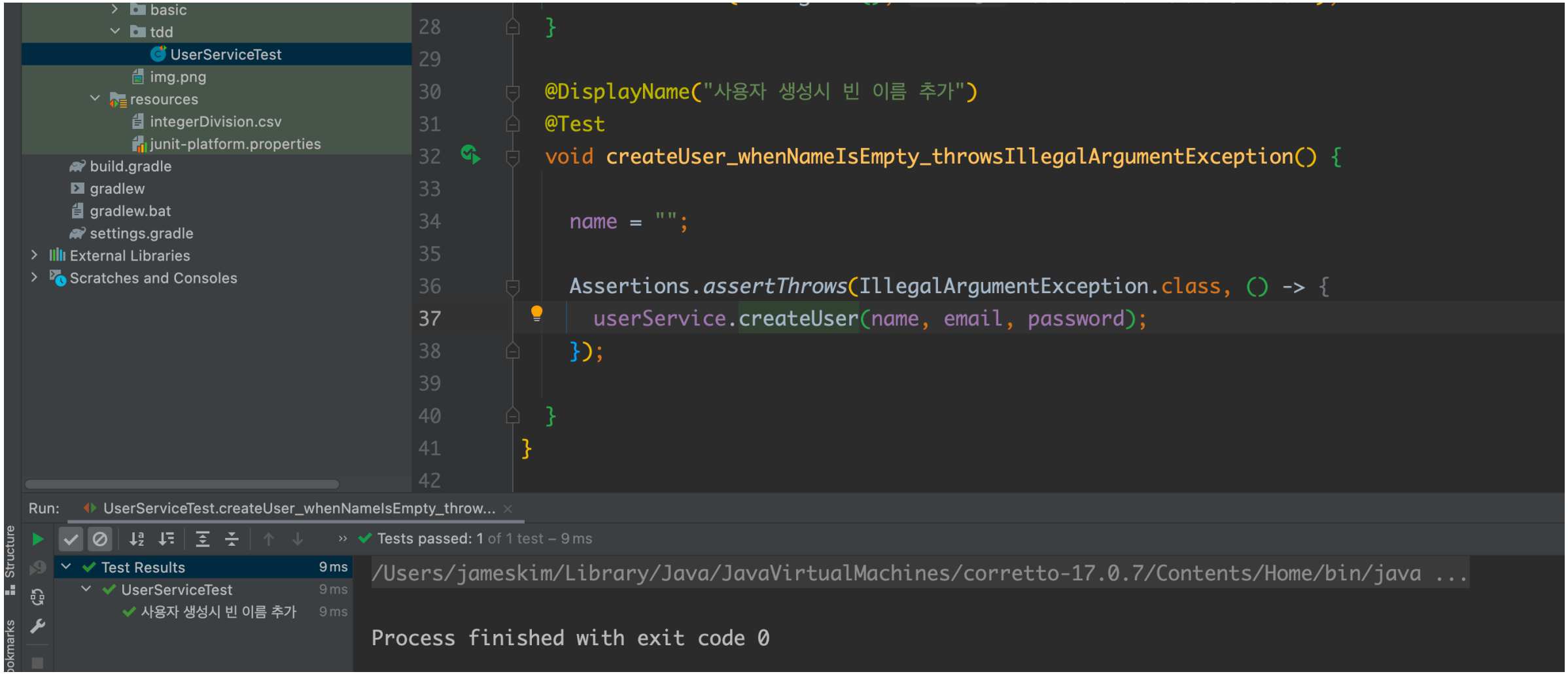
테스트 주도 개발(TDD)

```
@DisplayName("사용자 생성시 빈 이름 추가")
@Test
void createUser_whenNameIsEmpty_throwsIllegalArgumentException() {

    name = "";

    Assertions.assertThrows(IllegalArgumentException.class, () -> {
        userService.createUser(name, email, password);
    });

}
```

테스트 주도 개발(TDD)

```java
public class UserServiceImpl implements UserService {

  @Override
  public User createUser(String name, String email, String password) {

    if(name == null || name.trim().length() == 0){
      throw new IllegalArgumentException("이름이 존재하지 않습니다.");
    }

    String id = UUID.randomUUID().toString();
    return new User(id, name, email, password);
  }
}
```

테스트 주도 개발(TDD)


basic
tdd
UserServiceTest
img.png
resources
integerDivision.csv
junit-platform.properties
build.gradle
gradlew
gradlew.bat
settings.gradle
External Libraries
Scratches and Consoles
}
@DisplayName("사용자 생성시 빈 이름 추가")
@Test
void createUser_whenNameIsEmpty_throwsIllegalArgumentException() {
name = "";
Assertions.assertThrows(IllegalArgumentException.class, () -> {
userService.createUser(name, email, password);
});
}
}
Run: UserServiceTest.createUser_whenNameIsEmpty_throw...
Tests passed: 1 of 1 test – 9 ms
Test Results 9 ms
UserServiceTest 9 ms
사용자 생성시 빈 이름 추가 9 ms
/Users/jameskim/Library/Java/JavaVirtualMachines/corretto-17.0.7/Contents/Home/bin/java ...
Process finished with exit code 0

테스트 주도 개발(TDD)

```
@DisplayName("사용자 생성시 빈 이름 추가")
@Test
void createUser_whenNameIsEmpty_throwsIllegalArgumentException() {

    name = "";
    String expectedExceptionMessage = "이름이 존재하지 않습니다.";

    IllegalArgumentException thrown = Assertions.assertThrows(
        IllegalArgumentException.class, () -> {
          userService.createUser(name, email, password);
        }, message: "이름이 존재하지 않을 때는 IllegalArgumentException을 반환해야합니다.");

    assertEquals(expectedExceptionMessage, thrown.getMessage());
}
}
```

테스트 주도 개발(TDD)

05
Mockito

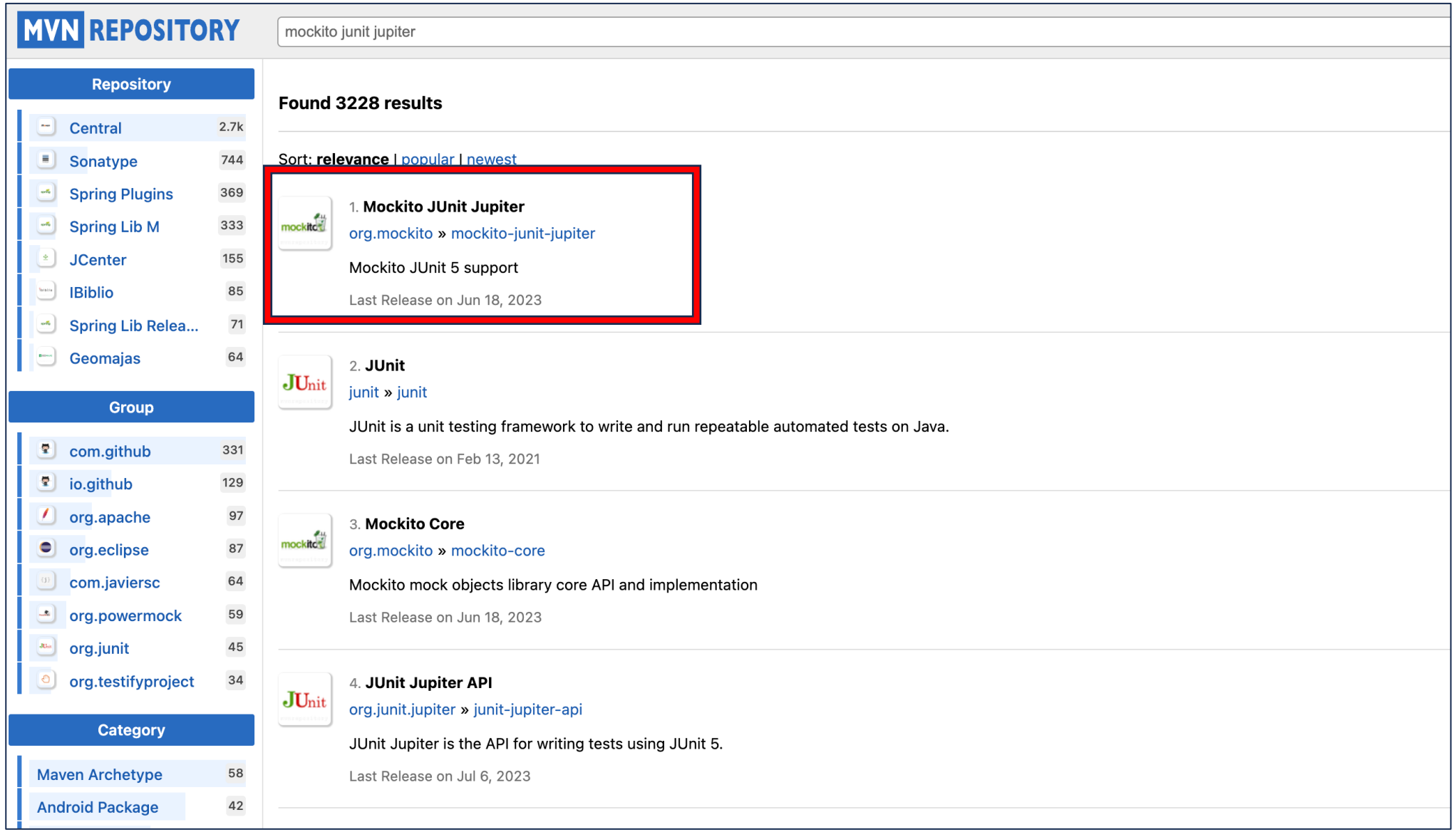
Test Double

Mock, Fake, Spy, Stub, Dummy …

Mockito

```
public User createUser ( UserInfo userInfo, DAO mySqlDao ){

    // 코드

    String userId = mySqlDao.save(user);

    // 코드

}
```

Mockito


MVN REPOSITORY
mockito junit jupiter
Repository
Central 2.7k
Sonatype 744
Spring Plugins 369
Spring Lib M 333
JCenter 155
IBiblio 85
Spring Lib Relea... 71
Geomajas 64
Group
com.github 331
io.github 129
org.apache 97
org.eclipse 87
com.javiersc 64
org.powermock 59
org.junit 45
org.testifyproject 34
Category
Maven Archetype 58
Android Package 42
Found 3228 results
Sort: relevance | popular | newest
1. Mockito JUnit Jupiter
org.mockito » mockito-junit-jupiter
Mockito JUnit 5 support
Last Release on Jun 18, 2023
2. JUnit
junit » junit
JUnit is a unit testing framework to write and run repeatable automated tests on Java.
Last Release on Feb 13, 2021
3. Mockito Core
org.mockito » mockito-core
Mockito mock objects library core API and implementation
Last Release on Jun 18, 2023
4. JUnit Jupiter API
org.junit.jupiter » junit-jupiter-api
JUnit Jupiter is the API for writing tests using JUnit 5.
Last Release on Jul 6, 2023

Mockito

Mockito JUnit Jupiter » 5.4.0

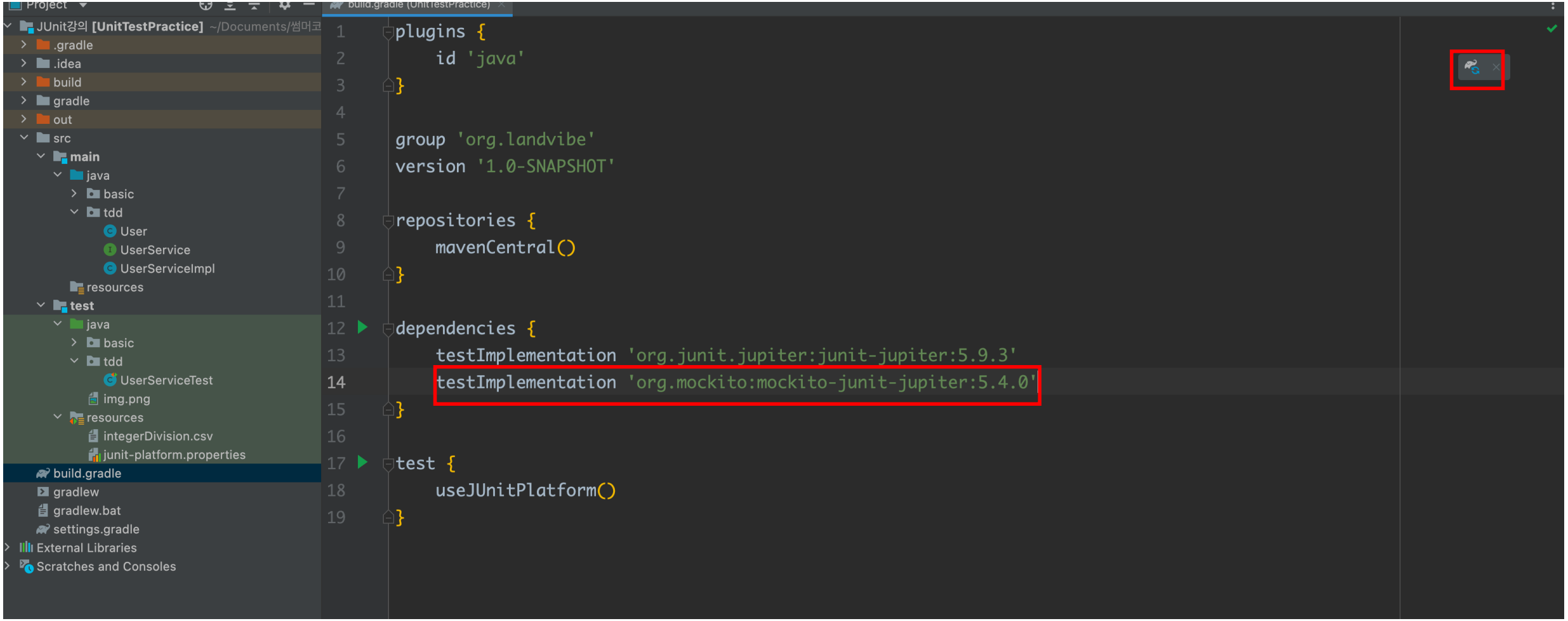
Mockito JUnit 5 support

License	MIT
Categories	Mocking
Tags	mock mocking junit testing mockito
HomePage	https://github.com/mockito/mockito
Date	Jun 18, 2023
Files	pom (2 KB) jar (5 KB) View All
Repositories	Central
Ranking	**#141 in MvnRepository (See Top Artifacts)** **#5 in Mocking**
Used By	**3,237 artifacts**

Maven | Gradle | Gradle (Short) | Gradle (Kotlin) | SBT | Ivy | Grape | Leiningen | Buildr

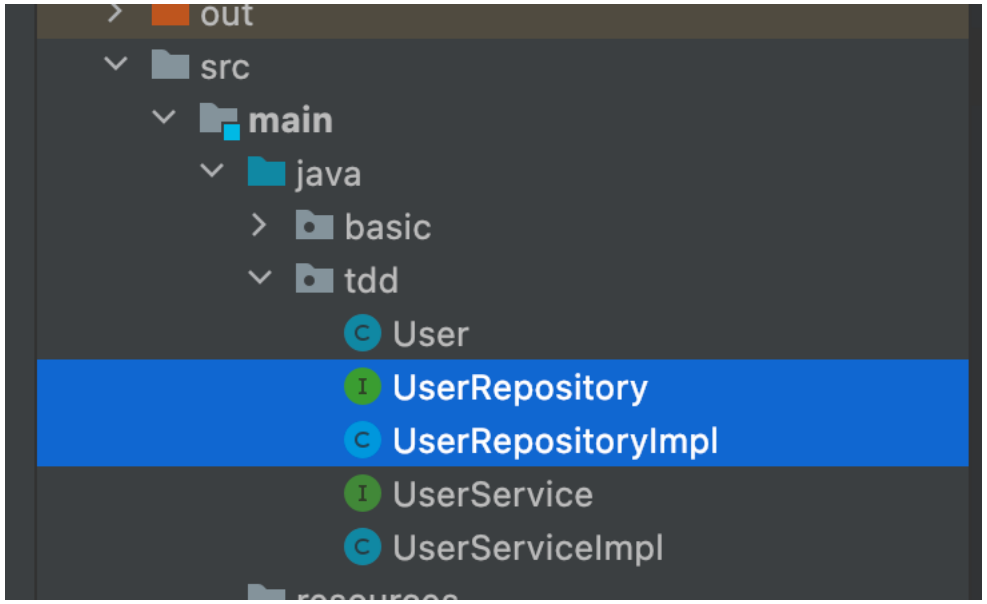
```
// https://mvnrepository.com/artifact/org.mockito/mockito-junit-jupiter
testImplementation 'org.mockito:mockito-junit-jupiter:5.4.0'
```

Mockito


Project
JUnit강의 [UnitTestPractice] ~/Documents/썸머코
.gradle
.idea
build
gradle
out
src
main
java
basic
tdd
User
UserService
UserServiceImpl
resources
test
java
basic
tdd
UserServiceTest
img.png
resources
integerDivision.csv
junit-platform.properties
build.gradle
gradlew
gradlew.bat
settings.gradle
External Libraries
Scratches and Consoles
build.gradle (UnitTestPractice)
plugins {
id 'java'
}
group 'org.landvibe'
version '1.0-SNAPSHOT'
repositories {
mavenCentral()
}
dependencies {
testImplementation 'org.junit.jupiter:junit-jupiter:5.9.3'
testImplementation 'org.mockito:mockito-junit-jupiter:5.4.0'
}
test {
useJUnitPlatform()
}

Mockito

```java
public interface UserRepository {

  boolean save(User user);
}
```

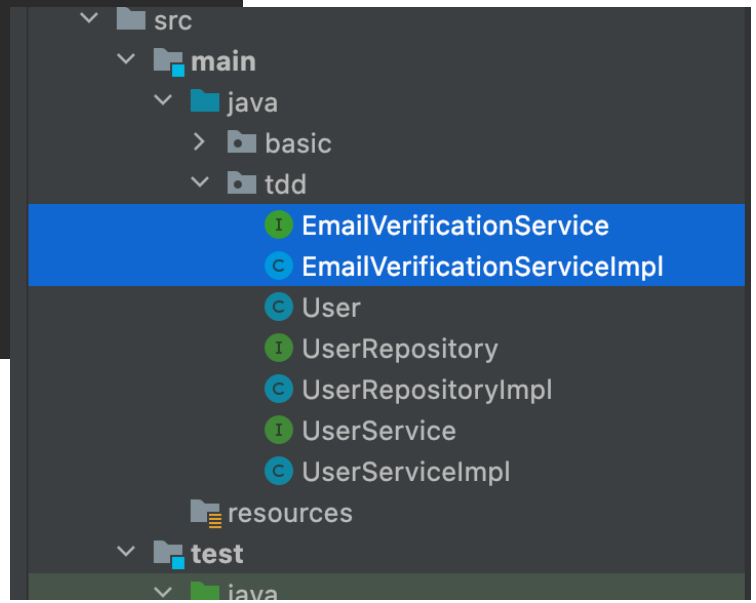



```java
public class UserRepositoryImpl implements UserRepository {

  Map<String, User> users = new HashMap<>();

  @Override
  public boolean save(User user) {

    boolean returnValue = false;

    if (!users.containsKey(user.getId())) {
      users.put(user.getId(), user);
      returnValue = true;
    }

    return returnValue;
  }
}
```

Mockito

```java
public interface EmailVerificationService {

    void scheduleEmailConfirmation(User user);
}
```

```java
public class EmailVerificationServiceImpl implements EmailVerificationService {

    @Override
    public void scheduleEmailConfirmation(User user) {
        System.out.println("scheduleEmailConfirmation is called");
    }
}
```



Mockito

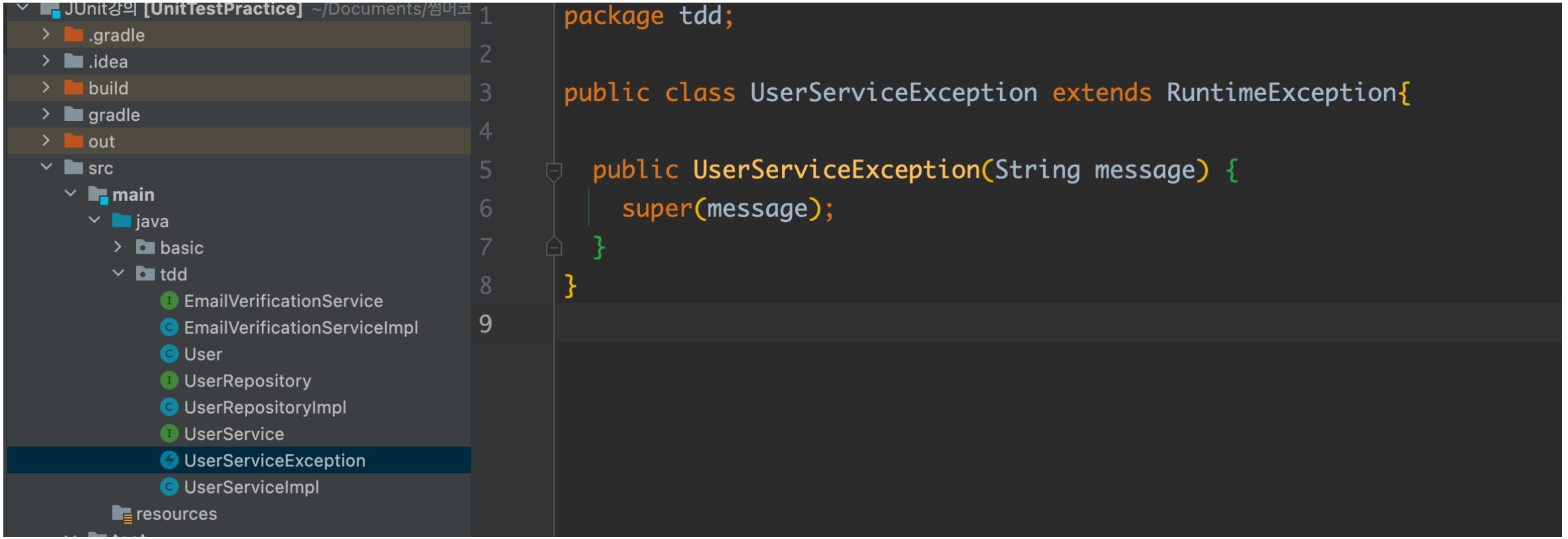
```
@ExtendWith(MockitoExtension.class)
public class MockitoTest {

  UserService userService = new UserServiceImpl();
```

Mockito

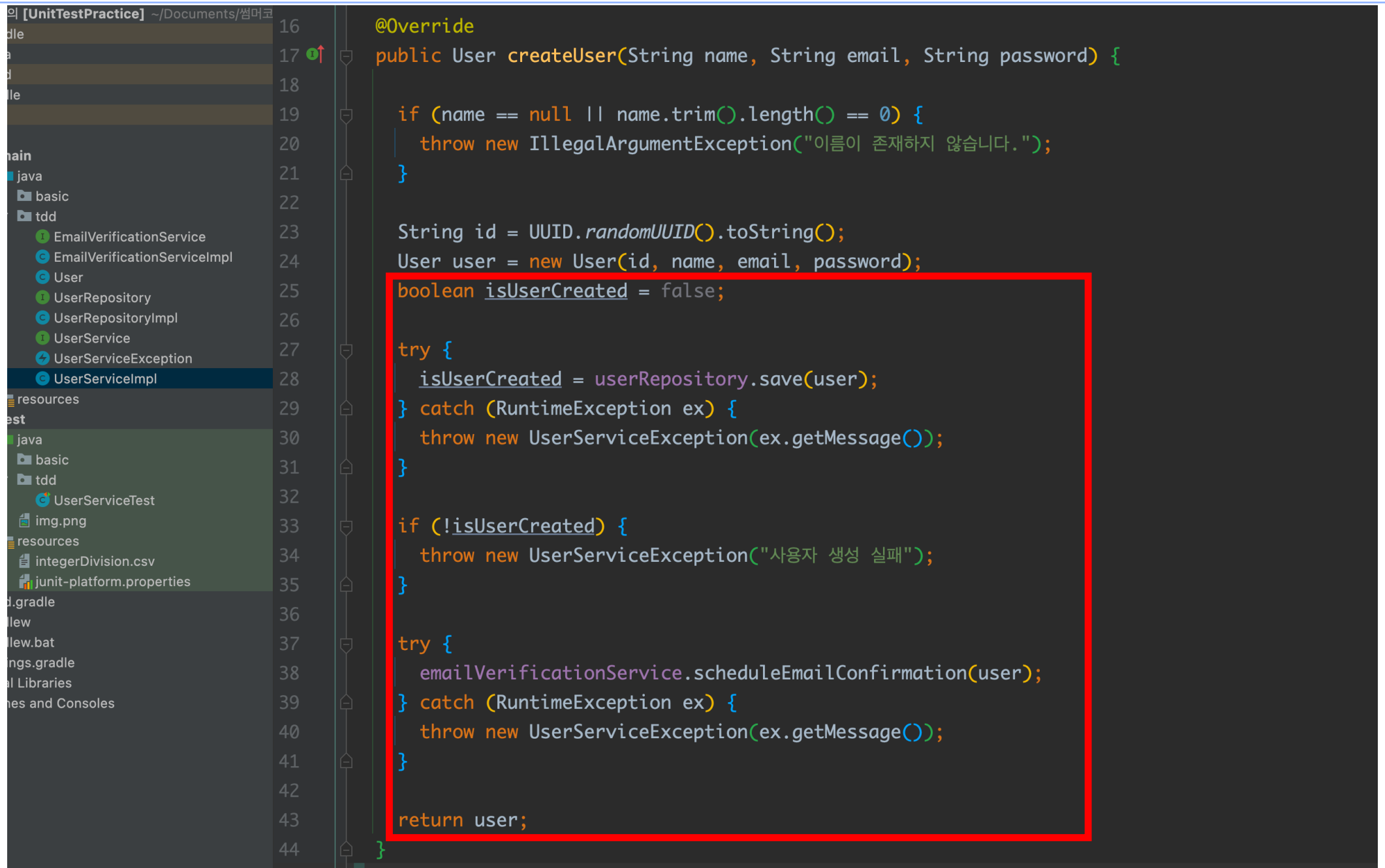
```
1 related problem
public class UserServiceImpl implements UserService {

  private UserRepository userRepository;
  private EmailVerificationService emailVerificationService;

  1 related problem
  public UserServiceImpl(UserRepository userRepository,
      EmailVerificationService emailVerificationService) {
    this.userRepository = userRepository;
    this.emailVerificationService = emailVerificationService;
  }
```

```
@ExtendWith(MockitoExtension.class)
public class UserServiceTest {

  @InjectMocks
  UserServiceImpl userService;

  @Mock
  EmailVerificationServiceImpl emailVerificationService;

  @Mock
  UserRepositoryImpl userRepository;
```

Mockito


JUnit강의 [UnitTestPractice] ~/Documents/쌤머코
.gradle
.idea
build
gradle
out
src
main
java
basic
tdd
EmailVerificationService
EmailVerificationServiceImpl
User
UserRepository
UserRepositoryImpl
UserService
UserServiceException
UserServiceImpl
resources
1 package tdd;
2
3 public class UserServiceException extends RuntimeException{
4
5 public UserServiceException(String message) {
6 super(message);
7 }
8 }
9

Mockito


[UnitTestPractice] ~/Documents/썸머코
java
basic
tdd
EmailVerificationService
EmailVerificationServiceImpl
User
UserRepository
UserRepositoryImpl
UserService
UserServiceException
UserServiceImpl
resources
java
basic
tdd
UserServiceTest
img.png
resources
integerDivision.csv
junit-platform.properties
Libraries
and Consoles
@Override
public User createUser(String name, String email, String password) {
if (name == null || name.trim().length() == 0) {
throw new IllegalArgumentException("이름이 존재하지 않습니다.");
}
String id = UUID.randomUUID().toString();
User user = new User(id, name, email, password);
boolean isUserCreated = false;
try {
isUserCreated = userRepository.save(user);
} catch (RuntimeException ex) {
throw new UserServiceException(ex.getMessage());
}
if (!isUserCreated) {
throw new UserServiceException("사용자 생성 실패");
}
try {
emailVerificationService.scheduleEmailConfirmation(user);
} catch (RuntimeException ex) {
throw new UserServiceException(ex.getMessage());
}
return user;
}

Mockito

```java
@DisplayName("사용자 생성")
@Test
public void createUser_whenUserDetailsProvided_returnUserObject() {

    when(userRepository.save(any(User.class))).thenReturn(true);

    User user = userService.createUser(name, email, password);

    assertNotNull(user,  message: "회원가입한 사용자는 null이면 안됩니다.");
    assertEquals(name, user.getName());
    assertEquals(email, user.getEmail());
    assertNotNull(user.getId(),  message: "사용자 id가 존재하지 않습니다.");
    verify(userRepository)
            .save(any(User.class));
    verify(userRepository, Mockito.times( wantedNumberOfInvocations: 1))
            .save(any(User.class));
}
```

Mockito

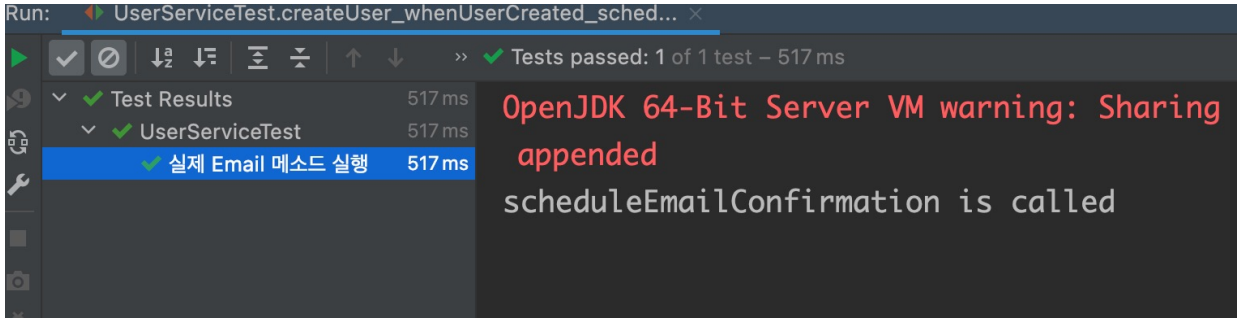

강의 [UnitTestPractice] ~/Documents/썸머코
adle
ea
ld
dle
main
java
basic
tdd
EmailNotificationServiceException
EmailVerificationService
EmailVerificationServiceImpl
User
UserRepository
1 package tdd;
2
3 public class EmailNotificationServiceException extends RuntimeException{
4
5 public EmailNotificationServiceException(String message) {
6 super(message);
7 }
8 }
9

Mockito

```java
@DisplayName("사용자 생성시 이메일 예외 발생")
@Test
void createUser_whenEmailNotificationExceptionThrown_throwsUserServiceException() {
    // Arrange
    when(userRepository.save(any(User.class))).thenReturn(true);

    doThrow(EmailNotificationServiceException.class) Stubber
            .when(emailVerificationService) EmailVerificationServiceImpl
            .scheduleEmailConfirmation(any(User.class));

    // Act & Assert
    Assertions.assertThrows(UserServiceException.class, () -> {
        userService.createUser(name, email, password);
    }, message: "UserServiceException이 발생해야 합니다.");

    // Assert
    verify(emailVerificationService, times( wantedNumberOfInvocations: 1)).
            scheduleEmailConfirmation(any(User.class));
}
```

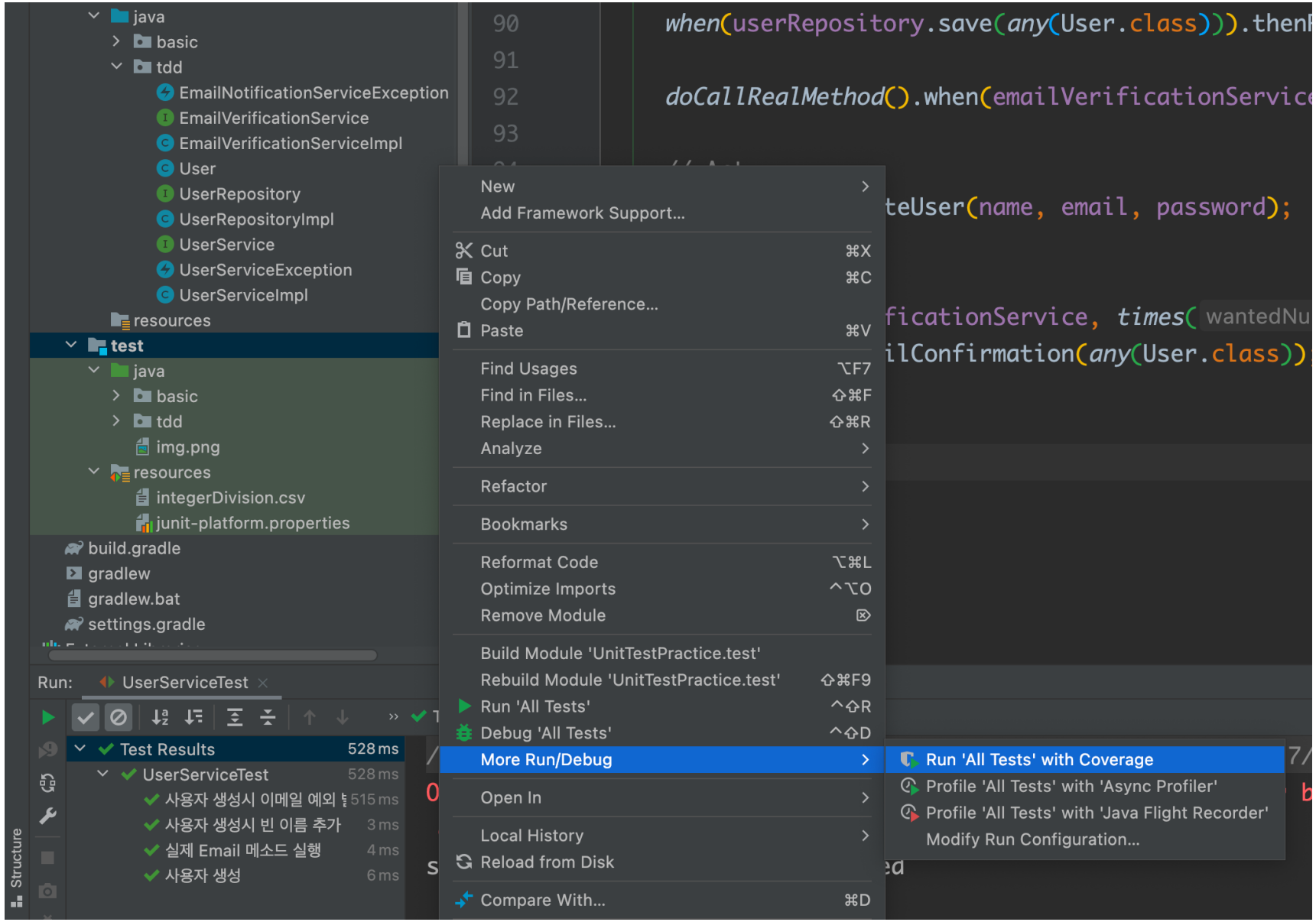
Mockito

```java
@DisplayName("실제 Email 메소드 실행")
@Test
void createUser_whenUserCreated_schedulesEmailConfirmation() {
    // Arrange
    when(userRepository.save(any(User.class))).thenReturn(true);

    doCallRealMethod().when(emailVerificationService).scheduleEmailConfirmation(any(User.class));

    // Act
    userService.createUser(name, email, password);

    // Assert
    verify(emailVerificationService, times( wantedNumberOfInvocations: 1))
            .scheduleEmailConfirmation(any(User.class));
}
```

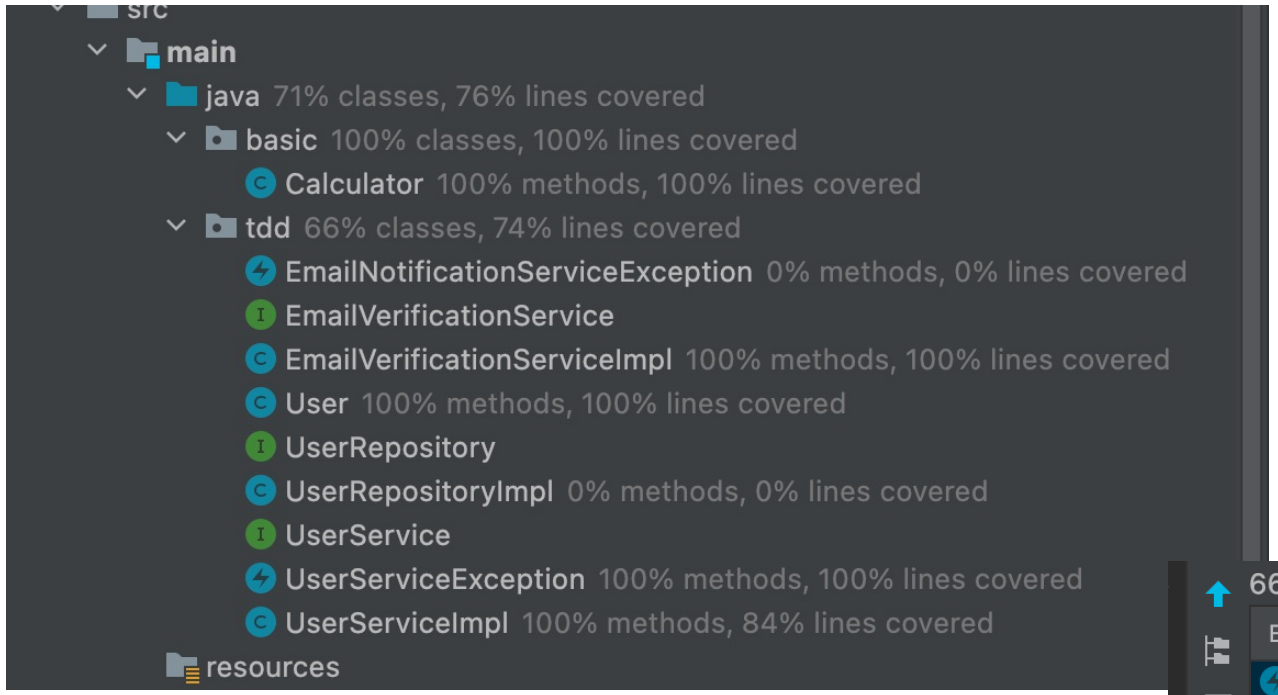


06
테스트 커버리지와 Jacoco

테스트 커버리지와 Jacoco



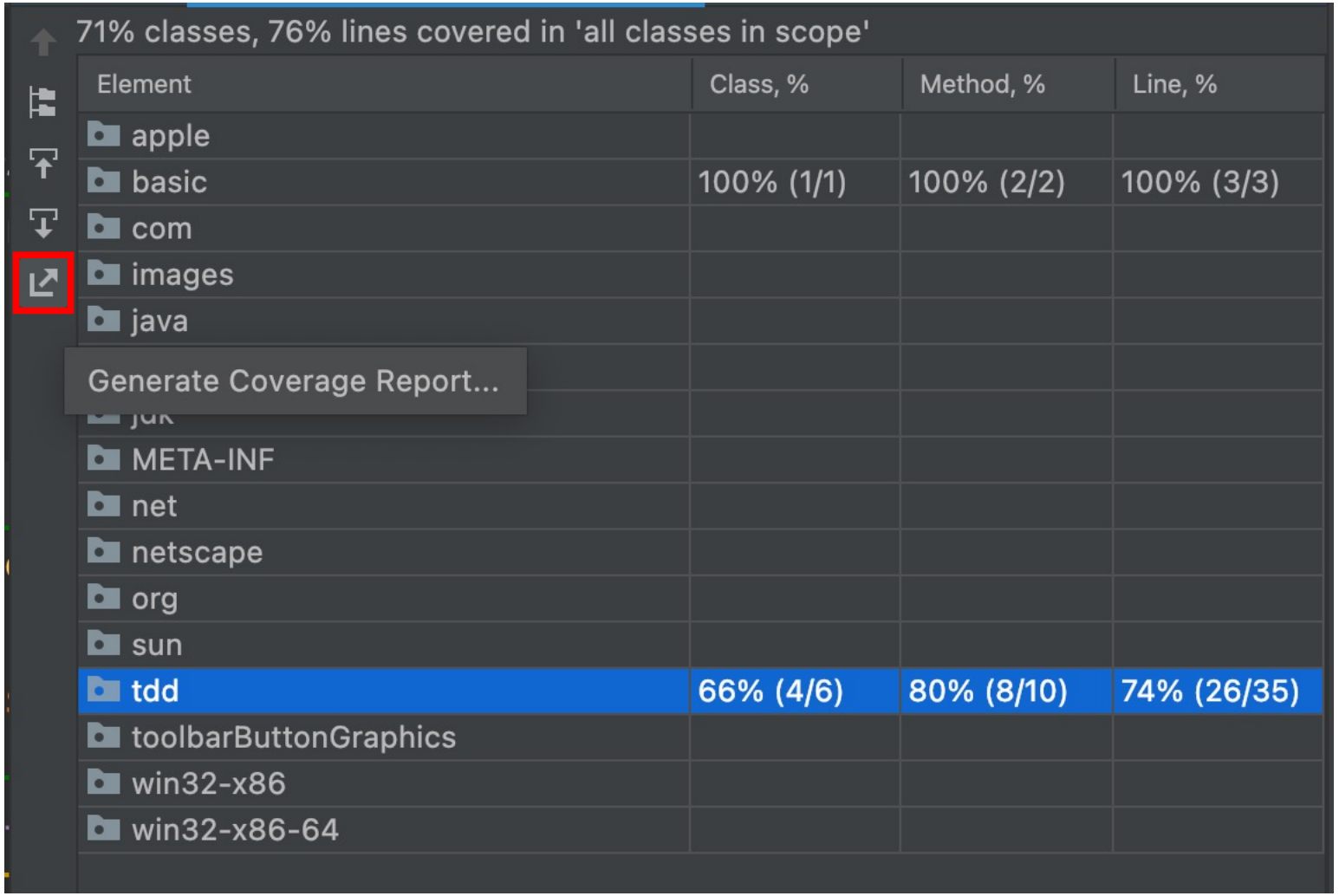
테스트 커버리지와 Jacoco

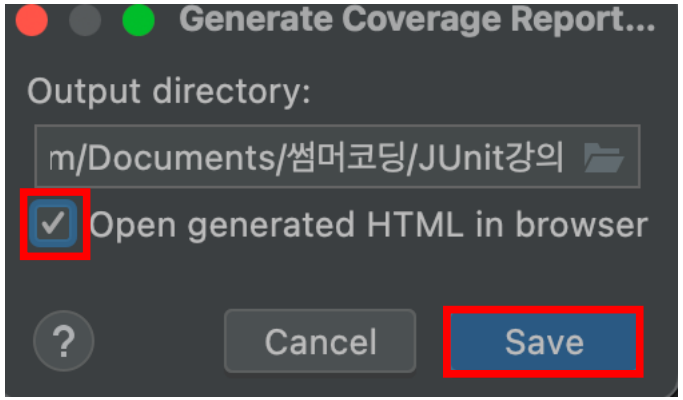


66% classes, 74% lines covered in package 'tdd'

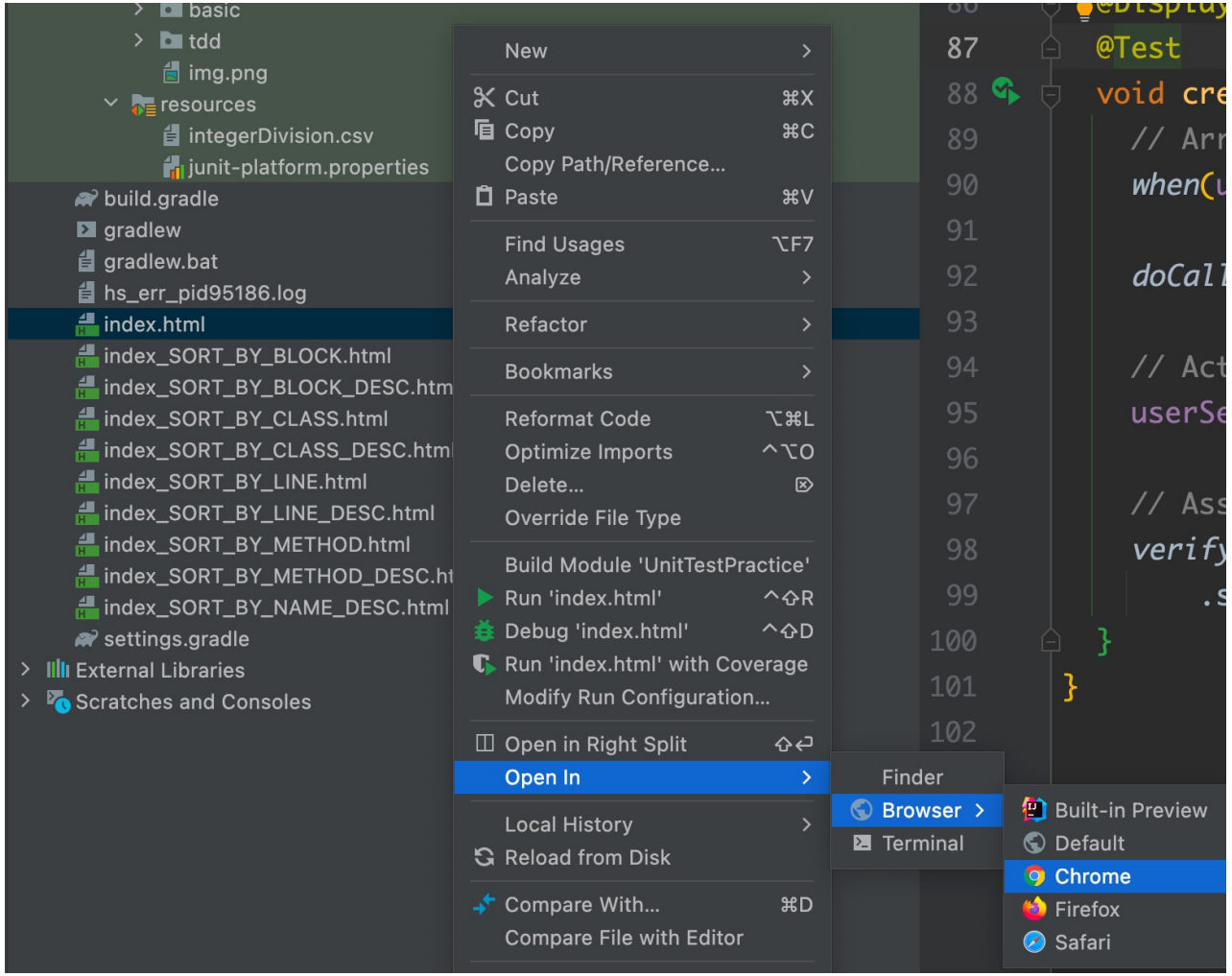
Element	Class, %	Method, %	Line, %
EmailNotificationServiceException	0% (0/1)	0% (0/1)	0% (0/1)
EmailVerificationService	100% (0/0)	100% (0/0)	100% (0/0)
EmailVerificationServiceImpl	100% (1/1)	100% (1/1)	100% (1/1)
User	100% (1/1)	100% (4/4)	100% (8/8)
UserRepository	100% (0/0)	100% (0/0)	100% (0/0)
UserRepositoryImpl	0% (0/1)	0% (0/1)	0% (0/5)
UserService	100% (0/0)	100% (0/0)	100% (0/0)
UserServiceException	100% (1/1)	100% (1/1)	100% (1/1)
UserServiceImpl	100% (1/1)	100% (2/2)	84% (16/19)

테스트 커버리지와 Jacoco





테스트 커버리지와 Jacoco



Overall Coverage Summary

Package	Class, %	Method, %	Line, %
all classes	71.4% (5/7)	73.3% (11/15)	70.7% (29/41)

Coverage Breakdown

Package	Class, %	Method, %	Line, %
basic	100% (1/1)	100% (3/3)	100% (3/3)
tdd	66.7% (4/6)	66.7% (8/12)	68.4% (26/38)

generated on 2023-07-11 10:11

테스트 커버리지와 Jacoco

```
plugins {
    id 'java'
    id 'jacoco'
}

jacocoTestReport{
    reports{
        html.enabled true
        xml.enabled false
        csv.enabled false
        html.destination file("${buildDir}/jacocoHtml")
    }
}
```

JUnit ㄱ ㅏoo - `./gradlew test jacocoTestReport`

테스트 커버리지와 Jacoco