

공통과제를 위한 JUnit 세팅 가이드

Java 진영에서 가장 많이 사용되는 Unit Test Framework

- 켄트백(TDD 창시자)과 에릭감마(GoF 중 한명)가 만든 Java Test Framework
- 최신버전 JUnit5이며, Java 8 이상 버전이 요구됨

JUnit 5 사용 가이드 (Github & Document)

- <https://junit.org/junit5/docs/current/user-guide/>
- <https://github.com/junit-team/junit5/>

JUnit 버전 차이

JUnit 4

- Legacy 시스템에서 사용하기 있기에, 여전히 사용되는 버전
- 마지막 release : 21년 2월

JUnit 5



- 2017년 9월 공식 버전 출시 이후 가장 많이 사용되는 버전
- 람다, 스트림을 적극적으로 지원한다.
- @DisplayName으로 가독성 좋은 TC 이름 작성 가능

IntelliJ IDEA - **Community Edition** (2025년 이후 버전)

- Java 진영에서 가장 많이 사용되는 IDE
- 다운로드 : <https://www.jetbrains.com/idea/>

Java 버전 - **Java 17**

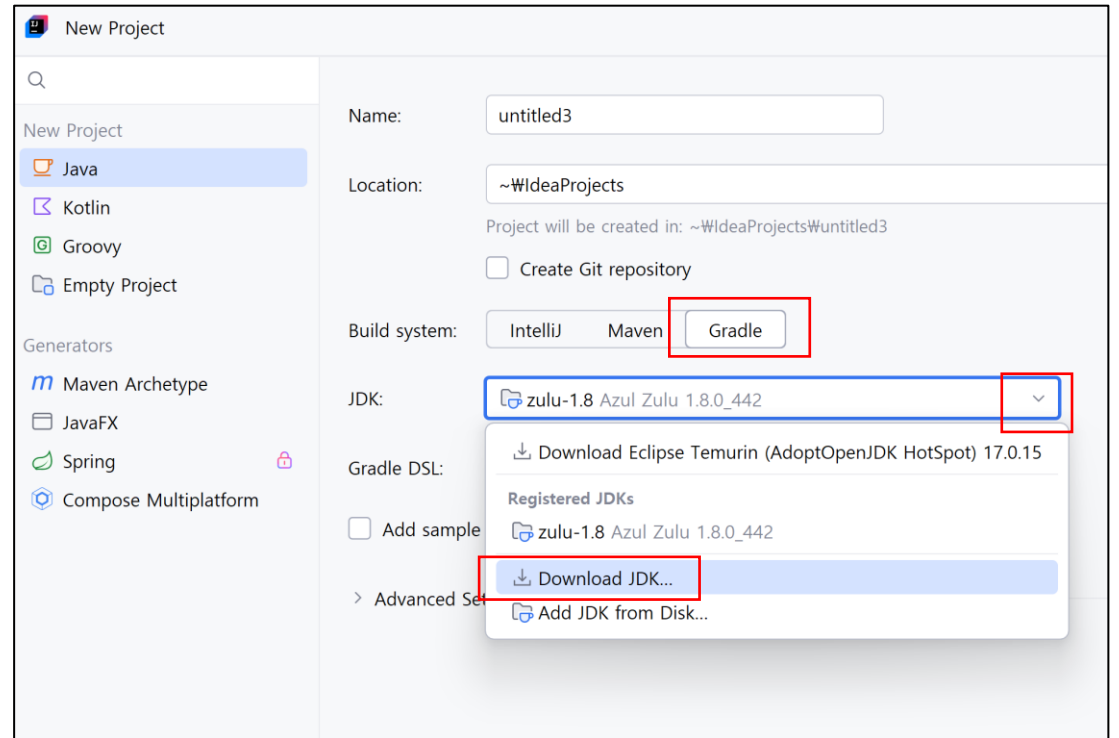
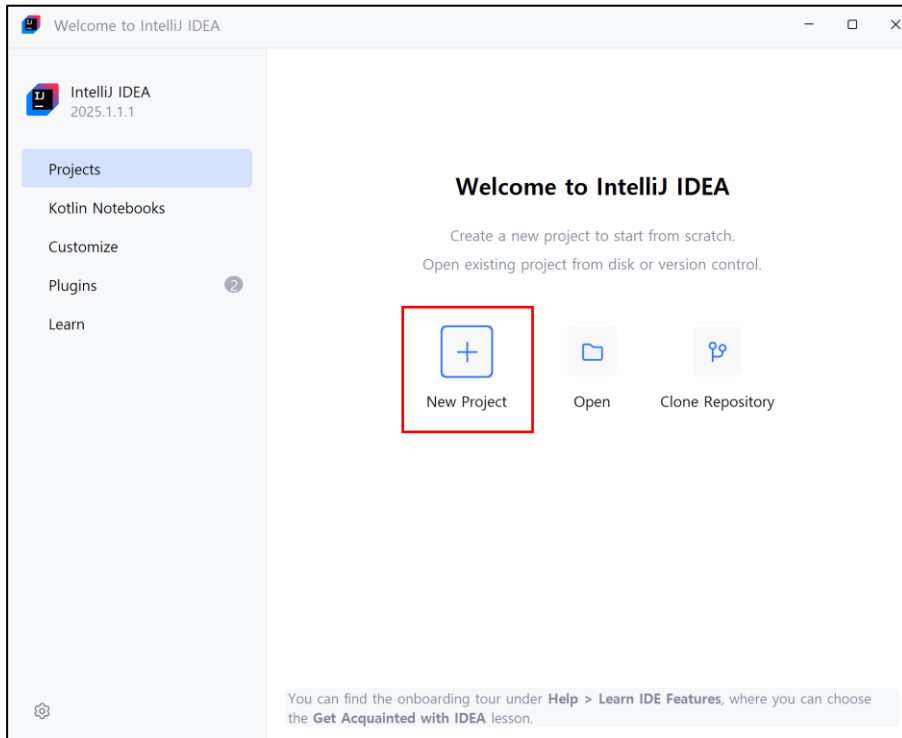
- Spring Boot 3.x 버전을 지원하는 최소버전 (Java 11 지원안됨)

Build System - **Gradle**

- 2025년 기준, 가장 많이 사용되는 빌드시스템

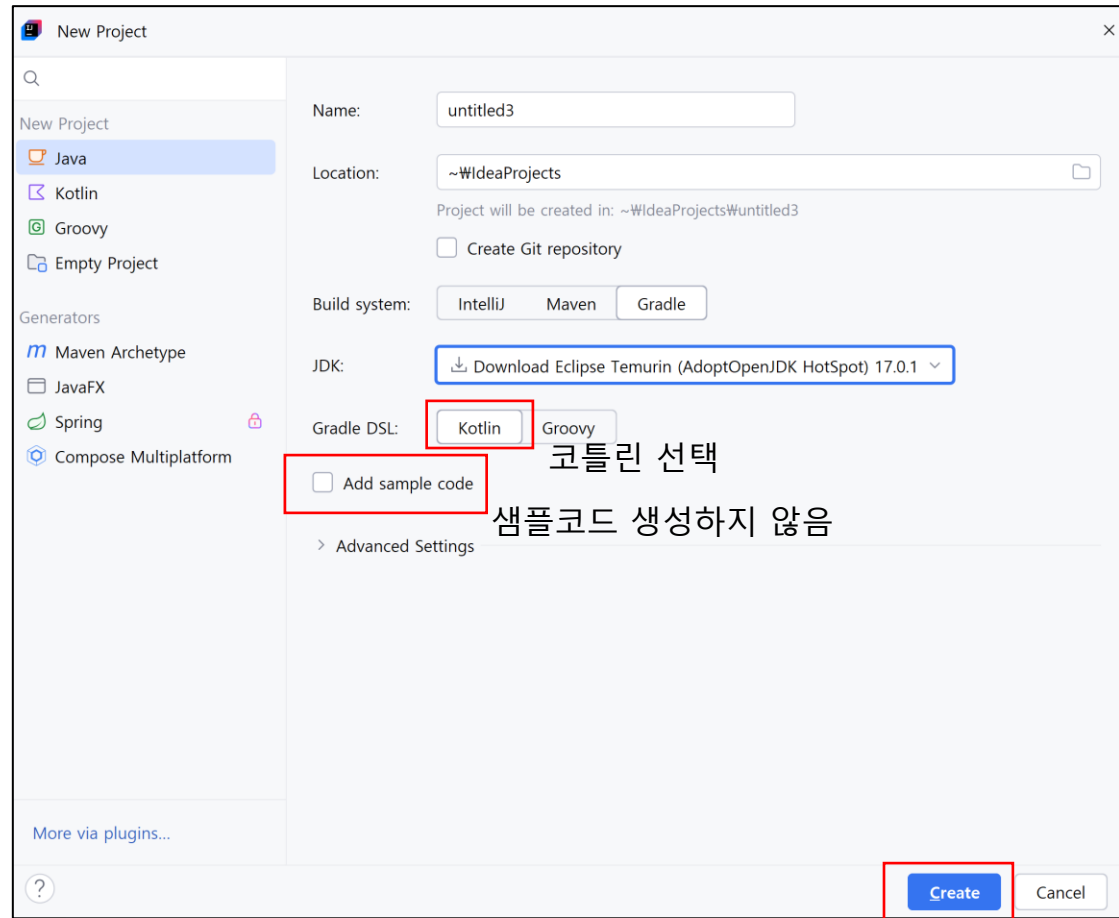
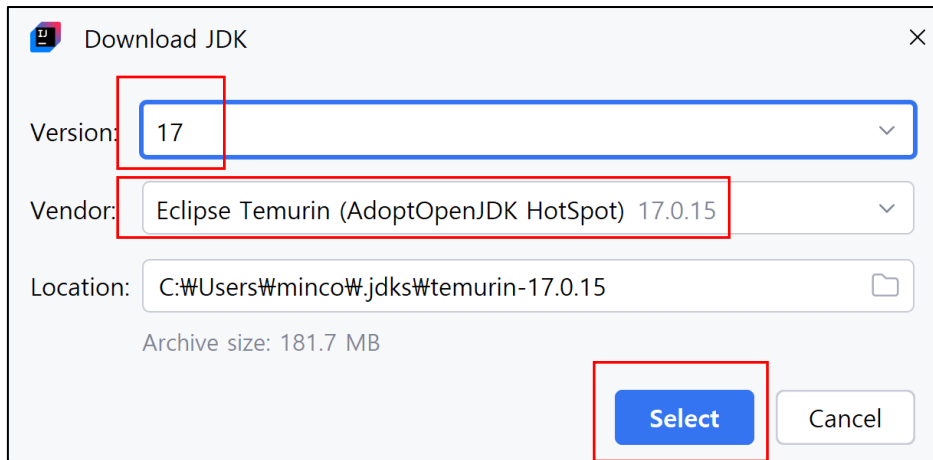
IntelliJ IDEA 실행 후 프로젝트 생성

1. New Project 클릭 후 Gradle 선택
2. Download JDK 누르기



JDK 선택 후, 프로젝트 생성하기

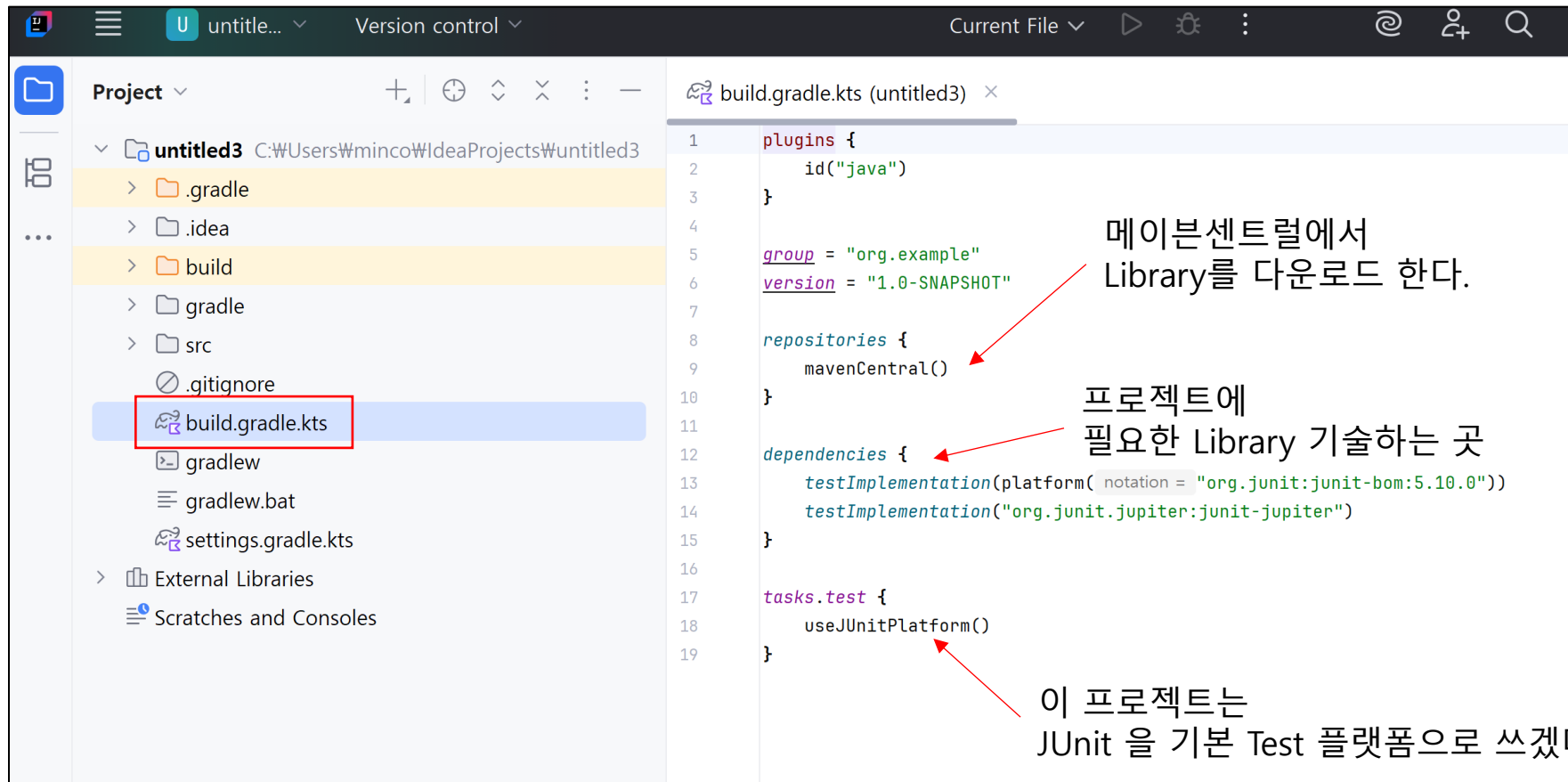
1. Java 17 버전, Eclipse Temurin (테무린) 선택
2. 프로젝트 생성하기



프로젝트 생성 완료

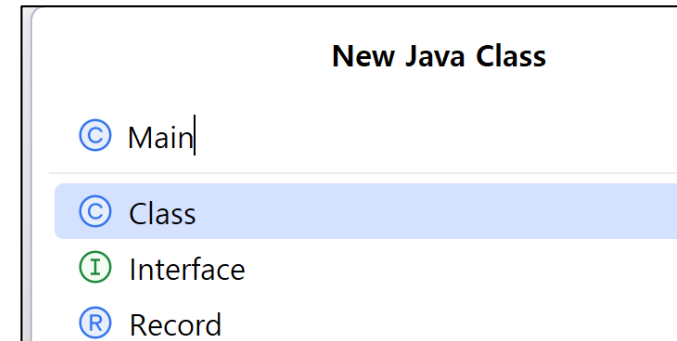
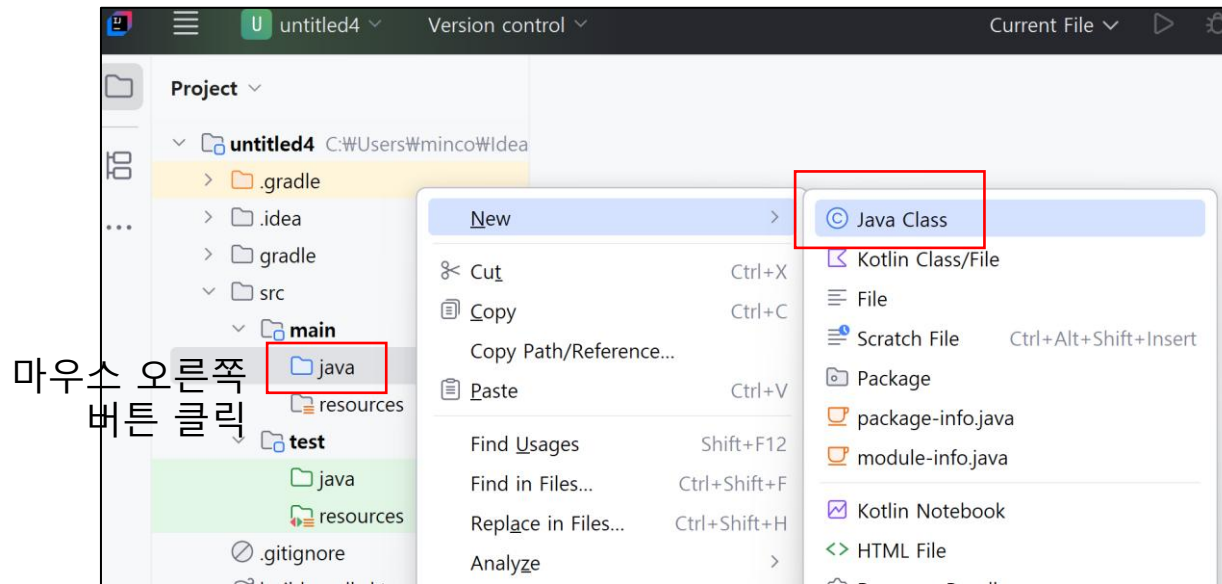
프로젝트 의존성 관리 파일

- Maven에서 pom.xml 과 같은 역할을 하는 파일
- 이 파일에서 필요한 Library를 적고, 저장하면 자동으로 다운로드 후 설치된다.



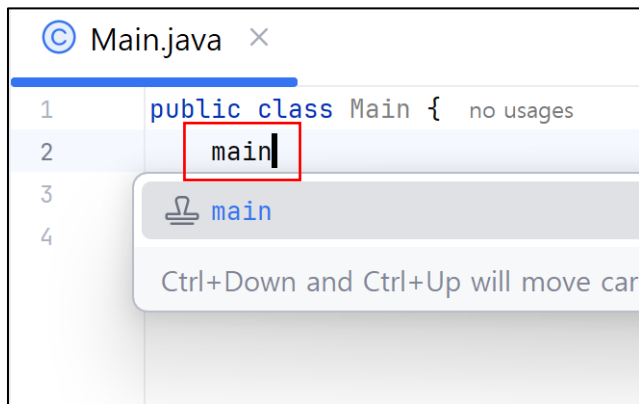
Main.java 생성 / main 함수 생성

Main.java 파일을 src/main/java/ 에 생성한다.



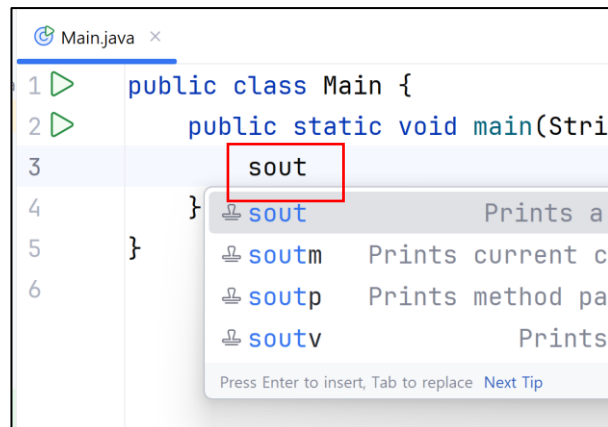
Main.java 파일 생성

빌드 테스트



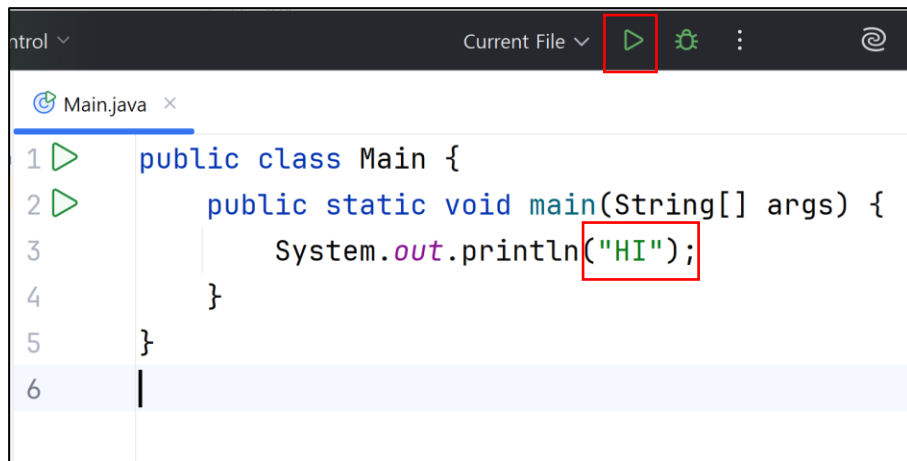
```
1 public class Main { no usages
2     main
3
4
```

main 기입 후 [TAB]키 누르기



```
1 public class Main {
2     public static void main(Stri
3         sout
4     }
5
6
```

sout 기입 후 [TAB]키 누르기



```
1 public class Main {
2     public static void main(String[] args) {
3         System.out.println("HI");
4     }
5 }
6
```

"HI" 출력 코드 입력 후 Run 버튼 클릭

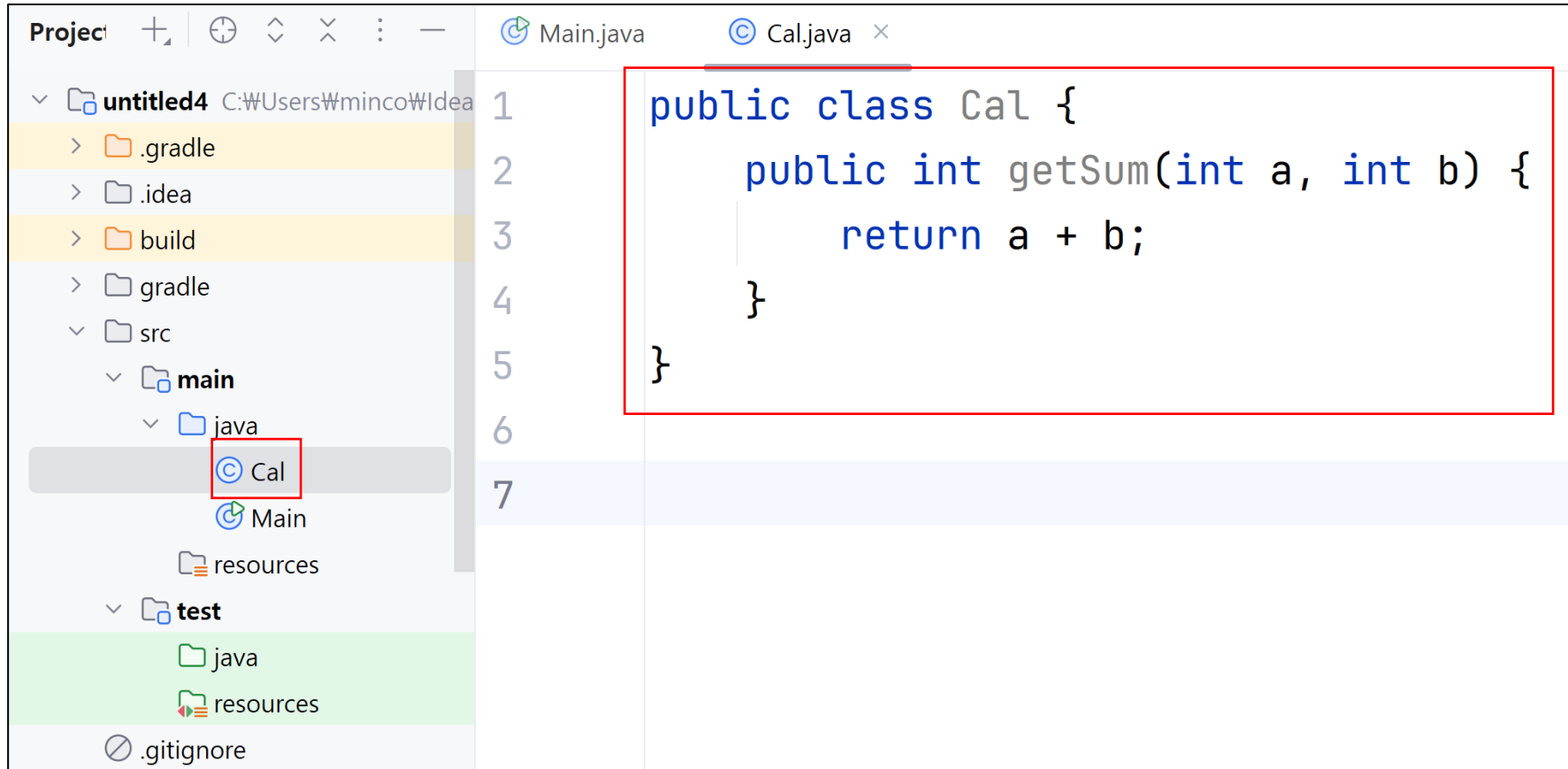


```
untitled4 [:Main.main()]
> Task :compileJava
> Task :processResources NO-SOURCE
> Task :classes
> Task :Main.main()
HI
BUILD SUCCESSFUL in 2s
2 actionable tasks: 2 executed
오후 7:30:55: Execution finished ':Main.main()'.
```

동작 확인하기

테스트 대상이 되는 파일 추가하기

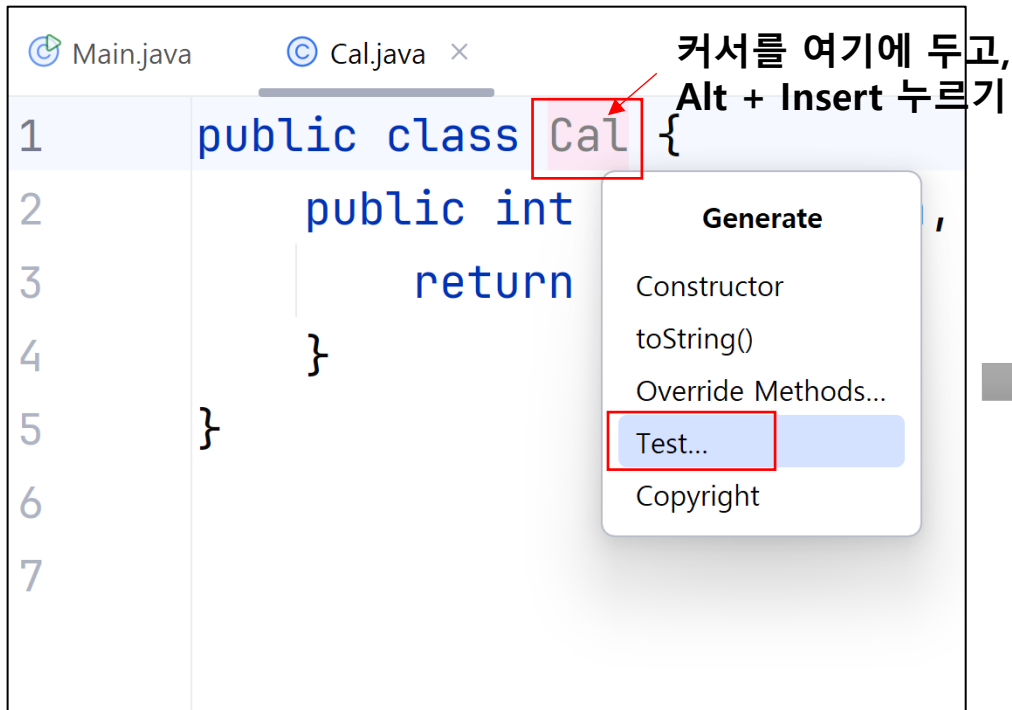
1. Cal.java 파일을 추가한다.
2. 테스트를 위한 기본 코드를 작성한다.



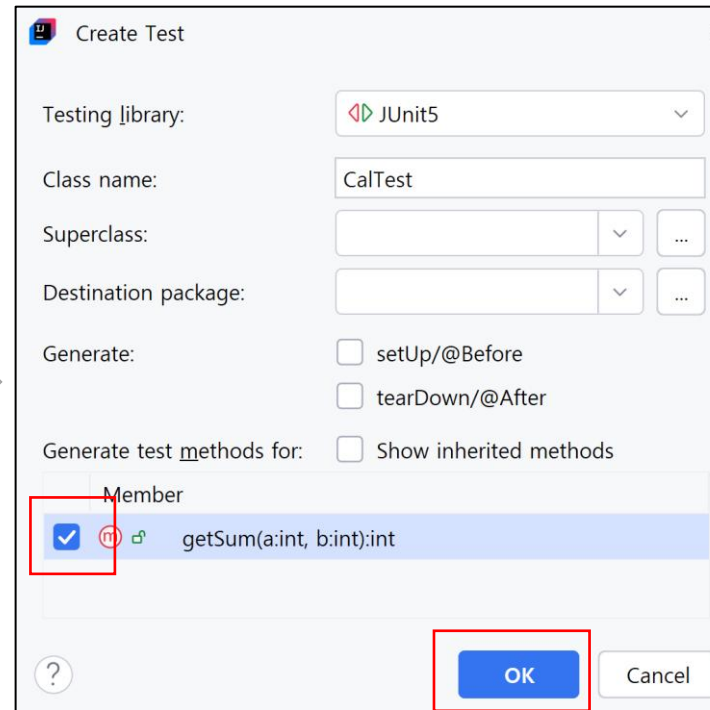
테스트 파일 생성

Cal 클래스를 클릭 후 Alt + Insert 키를 누른다.
메뉴에서 Code > Generate를 눌러도 된다.

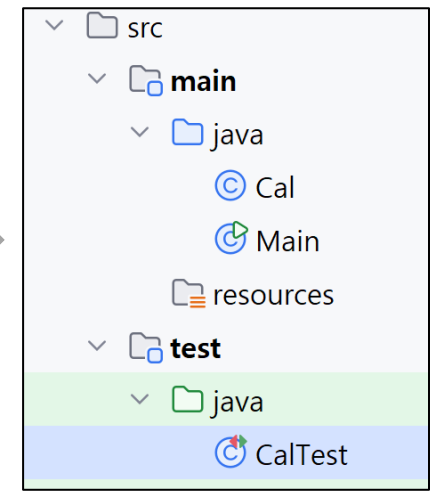
Test 파일이 자동 생성된다.



Generate > Test... 클릭



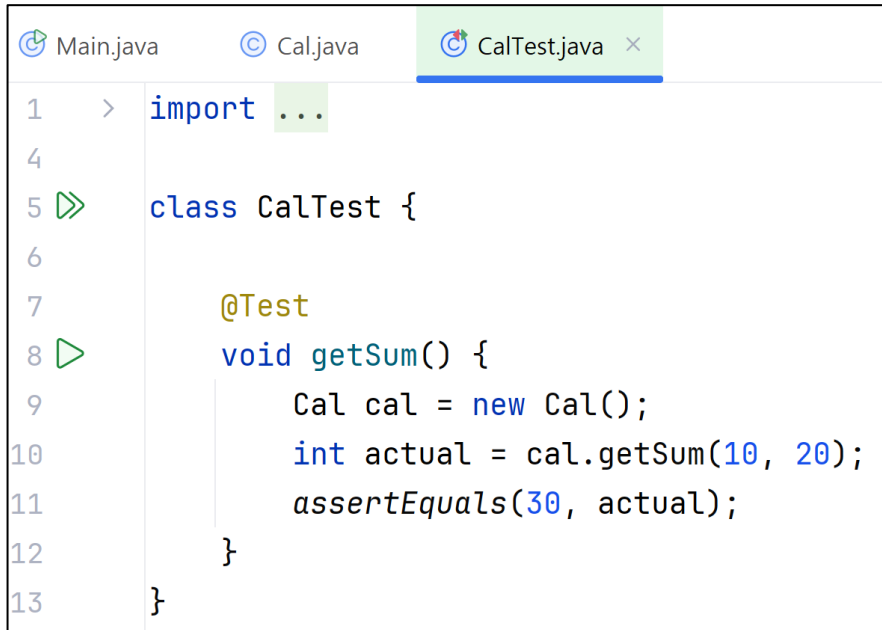
테스트할 대상을 선택한다.



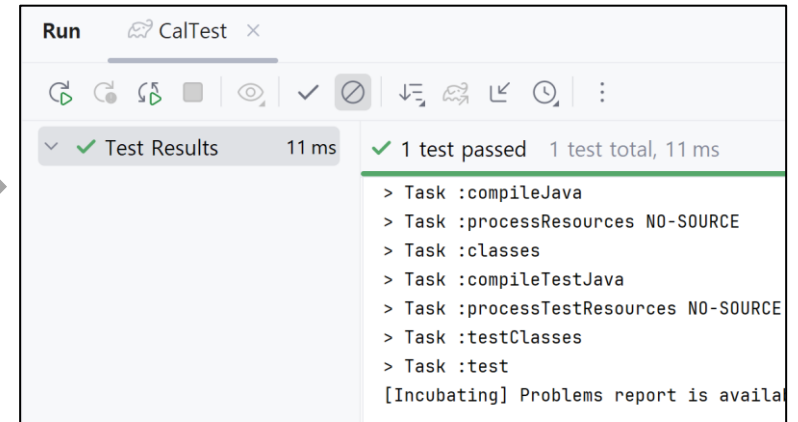
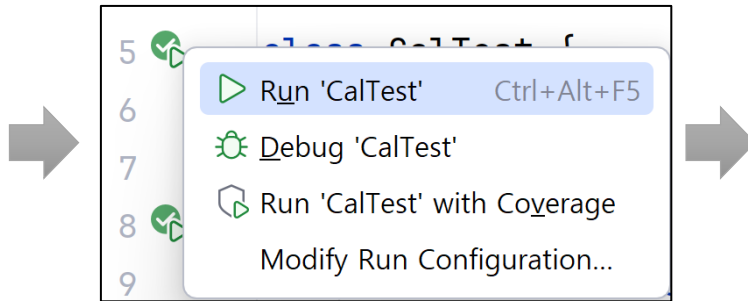
CalTest.java 파일
자동생성 된다.

테스트 기본 코드 작성

1. 테스트 코드를 확인한다.
2. 테스트 실행시 즉시, 결과를 확인할 수 있다.



```
1  > import ...
4
5  >> class CalTest {
6
7      @Test
8  > void getSum() {
9      Cal cal = new Cal();
10     int actual = cal.getSum(10, 20);
11     assertEquals(30, actual);
12 }
13 }
```



Mockito Library 설치를 위한 준비

mvnrepository.com 에 접속하여 "Mockito"를 검색한다.

- <https://mvnrepository.com/search?q=Mockito>
- Mockito 5.x 버전은 Java 11 이상부터 모두 사용 가능

The screenshot shows the Maven Repository search results for "mockito". The search bar contains "mockito". The results section shows "Found 311 results". The first result is "1. Mockito Core" by "org.mockito", with a link to "mockito-core". The description is "Mockito mock objects library core" and the last release is "Apr 4, 2025".

Repository	Count
Central	264
Sonatype	44
Mulesoft	18
IBiblio	15
JCenter	15



The screenshot shows the Mockito Core page on the Maven Repository. The page title is "Mockito Core" and the description is "Mockito mock objects library core API and implementation". The license is "MIT" and the category is "Mocking". The tags are "quality", "mock", "mocking", "testing", and "mockito". The home page is "https://github.com/mockito/mockito". The ranking is "#8 in MvnRepository (See Top Artifacts)" and "#1 in Mocking". The number of artifacts used by is "34,333".

Indexed Artifacts (56.8M)

Popular Categories

- Testing Frameworks & Tools
- Android Packages
- Logging Frameworks
- JVM Languages
- Java Specifications
- JSON Libraries
- Core Utilities
- Mocking
- Web Assets
- Annotation Libraries

Version

Version
5.17.x
5.16.x
5.15.x

5.x 버전
최신버전
아무거나 선택

Mockito Library 설치하기

Mockito 설치를 위한 의존성 선언문을 복사하여
build.gradle.kts 파일에 붙여넣기를 한다.

Mockito Core » 5.17.0
Mockito mock objects library core API and implementation

License	MIT
Categories	Mocking
Tags	quality mock mocking testing mockito
HomePage	https://github.com/mockito/mockito
Date	Apr 04, 2025
Files	pom (2 KB) jar (692 KB) View All
Repositories	Central
Ranking	#8 in MvnRepository (See Top Artifacts) #1 in Mocking
Used By	34,333 artifacts

Maven **Gradle** SBT Mill Ivy Grape Leiningen Buildr

Scope: Test Format: Kotlin

```
// https://mvnrepository.com/artifact/org.mockito/mockito-core
testImplementation("org.mockito:mockito-core:5.17.0")
```

의존성 선언문
복사하기



Project ▾

- untitled4 C:\Users\Wminco\Idea
 - .gradle
 - .idea
 - build
 - gradle
 - src
 - .gitignore
 - build.gradle.kts**
 - gradlew
 - gradlew.bat
 - settings.gradle.kts
- External Libraries
- Scratches and Consoles

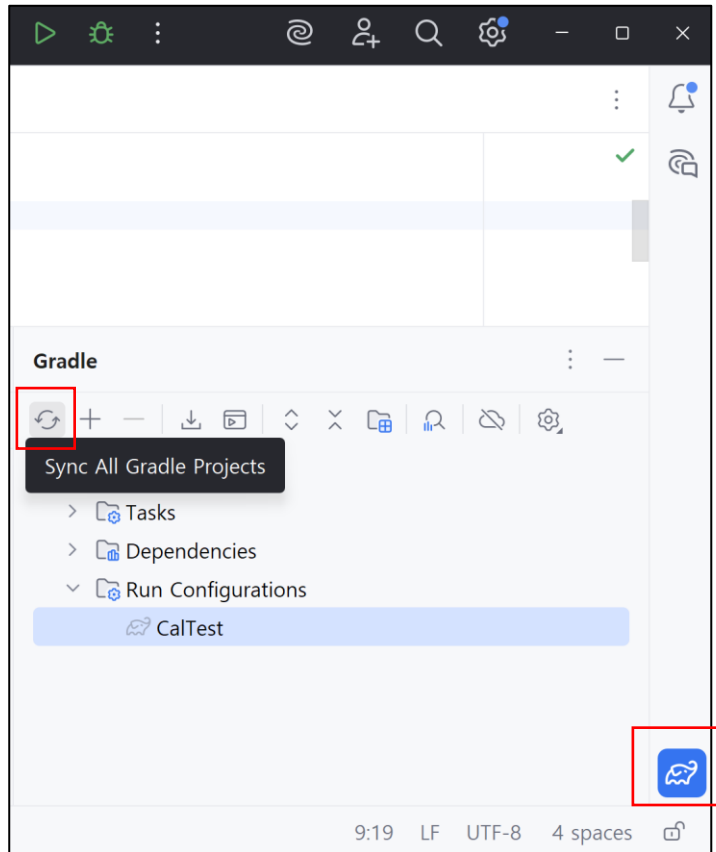
build.gradle.kts (untitled4) ×

```
1  plugins {
2      id("java")
3  }
4
5  group = "org.example"
6  version = "1.0-SNAPSHOT"
7
8  repositories {
9      mavenCentral()
10 }
11
12 dependencies {
13     testImplementation(platform("org.junit:junit-bom:5.10.0"))
14     testImplementation("org.junit.jupiter:junit-jupiter")
15     // https://mvnrepository.com/artifact/org.mockito/mockito-core
16     testImplementation("org.mockito:mockito-core:5.17.0")
17 }
18
19 tasks.test {
20     useJUnitPlatform()
21 }
```

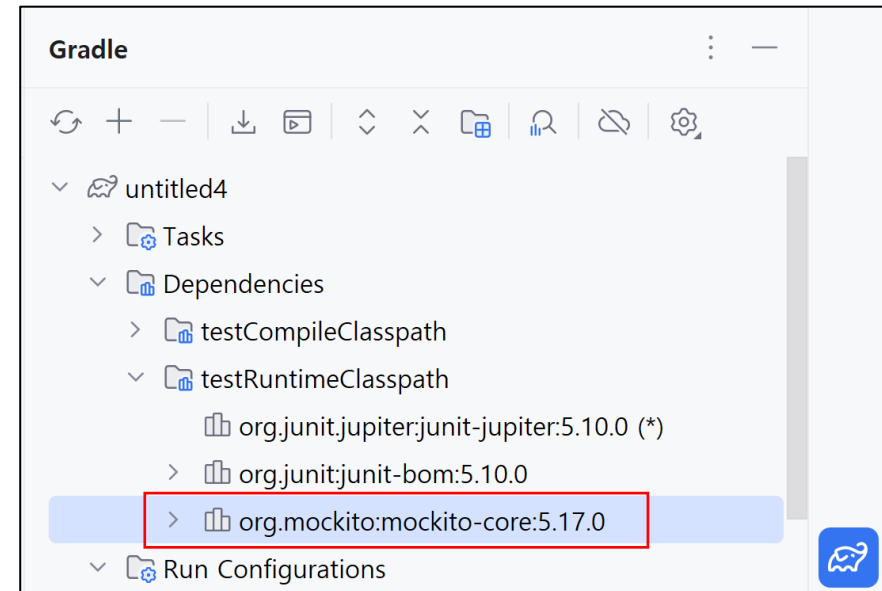
의존성 선언문
붙여넣기

Mockito 설치 / 설치 확인

build.gradle 파일을 저장하면 자동 설치가 시작된다. (Auto-Sync)



코끼리 버튼을 누르고
Sync 버튼을 누르면
수동으로 Library를 설치할 수 있다.

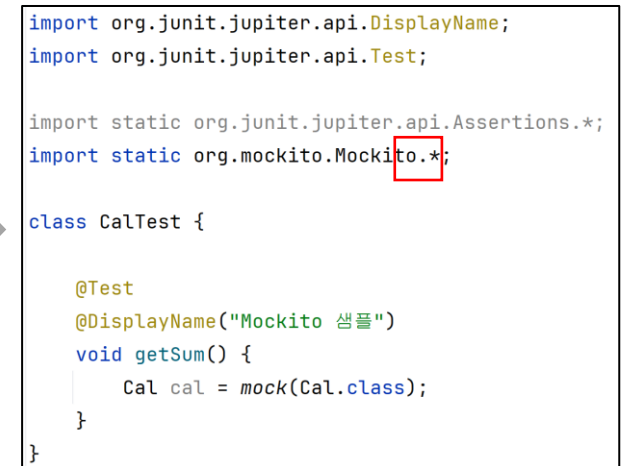
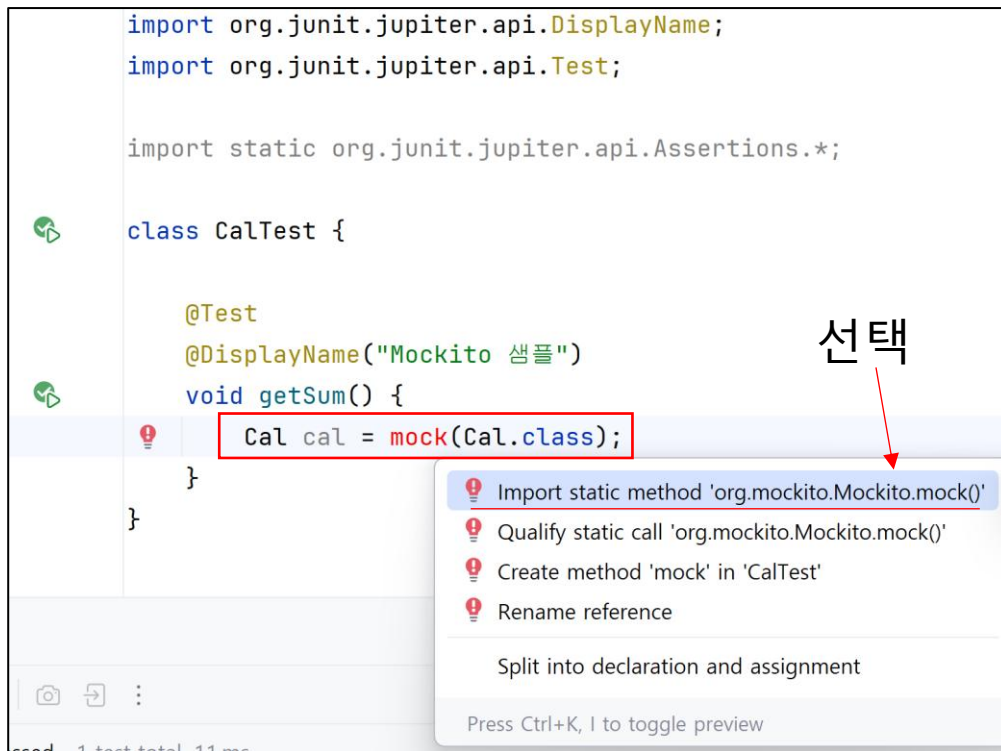


설치가 잘 되었다면
Dependencies 목록에
Mockito-core가 추가되어있다.

Mockito 동작 테스트 1

Mocking을 위한 import static 을 추가한다.

- 자동완성이 잘 안되면, 수동으로 import static을 기입해준다.



import static 코드가 추가되었다.
mock --> * 로 변경한다.

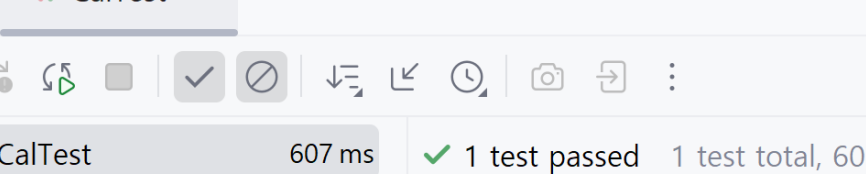
한 줄 코드 입력 후,
mock 클릭 후 Alt + Enter 누르기

Mockito 동작 테스트 2

Stubbing 이후, 결과를 확인해본다.

```
1  > import ...
6
7  class CalTest {
8
9      @Test
10     @DisplayName("Mockito 샘플")
11     void getSum() {
12         Cal cal = mock(Cal.class);
13         when(cal.getSum(1, 2)).thenReturn(999);
14         System.out.println(cal.getSum(1, 2));
15     }
16 }
17
18
```

getSum(1, 2) 호출시 999 리턴하도록 Stub 걸기



The screenshot shows the IntelliJ IDEA Run window. The top bar displays the 'Run' button and the test name 'CalTest'. Below the toolbar, the test results are listed in a table:

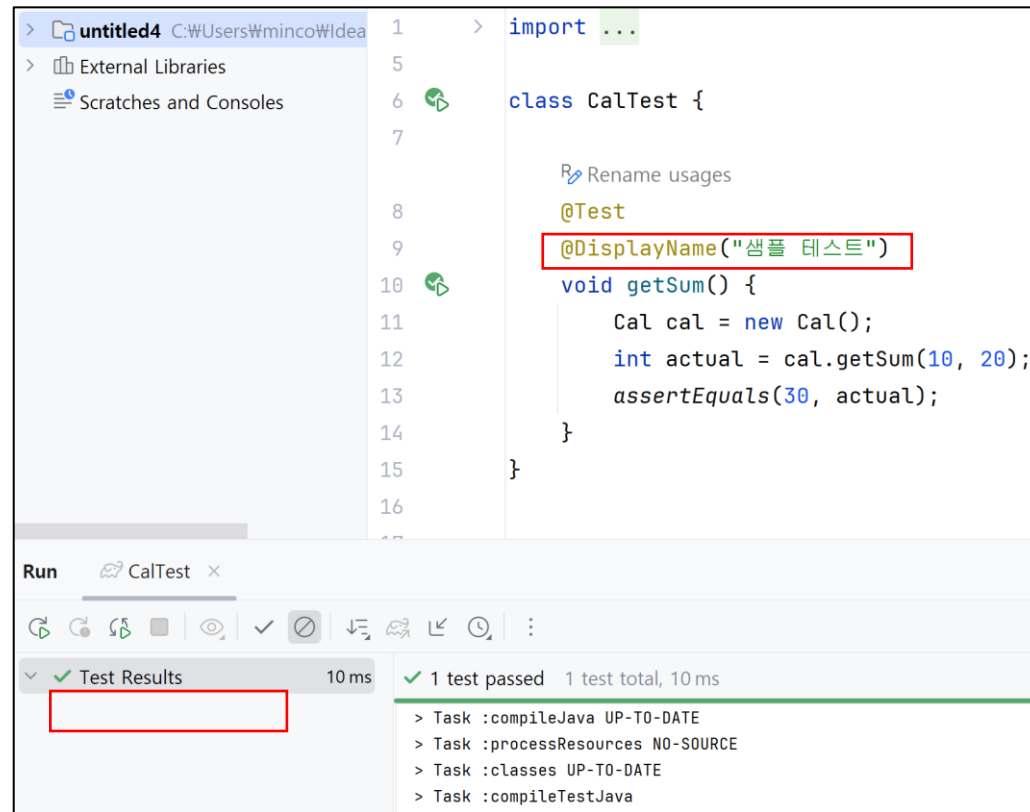
Test Name	Duration
✓ CalTest	607 ms
✓ Mockito 샘플	607 ms

Below the table, the test output is displayed. It shows a green checkmark and the text '1 test passed 1 test total, 607 ms'. Below this, the output path is shown: 'C:\Users\minco\.jdk\temurin-17.0.15\bin'. A red box highlights the output '999'. Below the red box, the text 'Process finished with exit code 0' is displayed.

999가 잘 출력됨을 확인할 수 있다.

[Trouble Shooting] Test Case 목록에 안뜰 경우

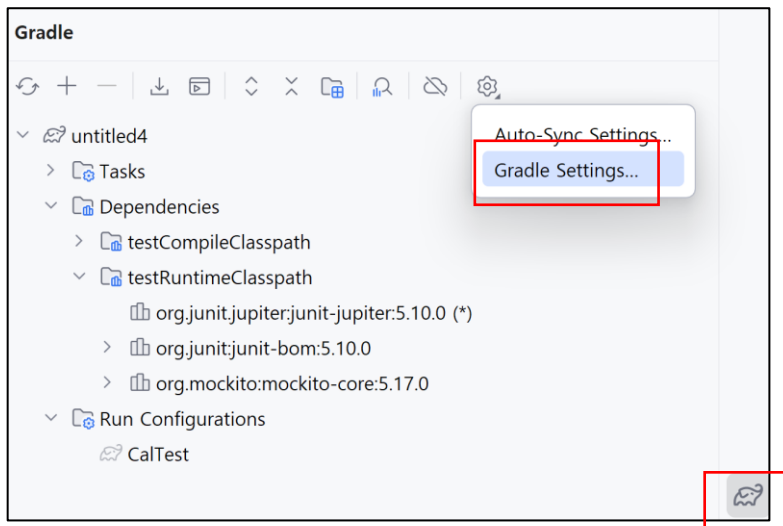
테스트탐색기에 Test Case들이 목록에 뜨지 않거나,
DisplayName을 명시했지만 뜨지 않는 경우 해결 방법



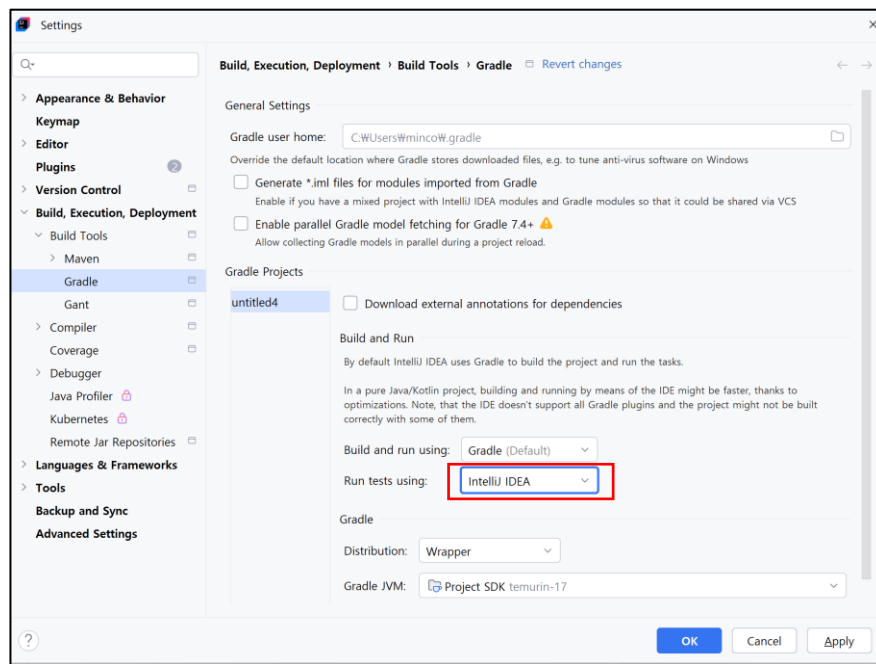
Test Case가 목록에 뜨지 않는다.

[Trouble Shooting] 해결방법

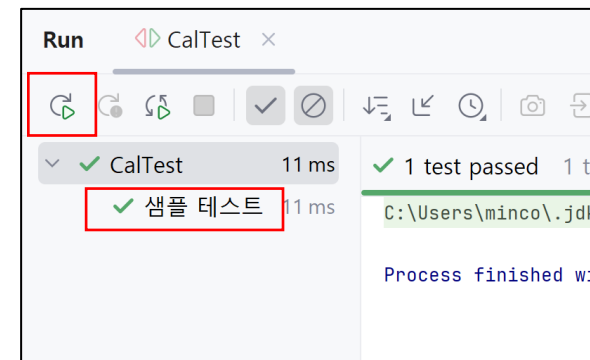
Test Case들이 목록에 뜨지 않거나,
DisplayName을 명시했지만 뜨지 않는 경우 해결 방법



Gradle 설정 진입하기



Gradle --> IntelliJ IDEA로 선택 변경



테스트를 다시 실행하면
Test Case가 목록에 뜬다.

감사합니다.