

# 롬복

## 개요

- 롬복(Lombok)
  - Java Class에 반복되는 Getter, Setter, toString등의 메소드를 자동 생성해주는 Java 라이브러리이다.
  - 예를 들어, Person 객체에 name, age 필드가 존재할 시에 두 필드에 해당하는 getName, getAge, setName, setAge, 등을 롬복에서는 자동으로 생성해주는 것이다.
  - Java 클래스나 필드 등등에 어노테이션을 붙여서 자동화한다.

## 예시

### 1. 롬복을 적용하지 않은 Person Class

```
public class Person {  
    private String name;  
    private int age;  
  
    public Person(){  
  
    }  
  
    public Person(String name, int age){  
        this.name = name;  
        this.age = age;  
    }  
  
    public String getName(){  
        return this.name;  
    }  
  
    public int getAge(){  
        return this.age;  
    }  
  
    public void setName(String name){  
        this.name = name;  
    }  
  
    public void setAge(String age){  
        this.age = age;  
    }  
}
```

## 2. 롬복을 적용한 Person Class

```
import ... (중략)

@Getter // getName(), getAge() 자동생성 어노테이션
@Setter // setName(), setAge() 자동생성 어노테이션
@NoArgsConstructor // Person() 자동생성 어노테이션
@AllArgsConstructor // Person(name, age) 자동생성 어노테이션

public class Person {
    private String name;
    private int age;
}
```

- ✓ [1.]과 [2.]의 클래스 내용은 같은 내용이 된다.  
[1.]과 [2.]를 비교해보면 확연히 코드가 간결해진 것을 확인할 수 있다.  
클래스의 필드가 더 많아진다면 롬복의 필요성을 더 느낄 것이다.

## 실습) IntelliJ에 Lombok 설치 및 활용해보기

※ 새로운 프로젝트를 생성하였다는 가정하에 실습을 진행한다.

### 1. build.gradle에 Lombok 의존성 코드 추가 및 build 진행



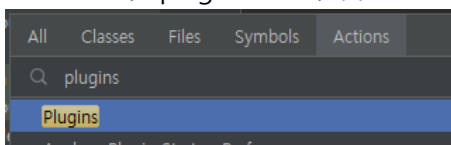
```
plugins {
    id 'java'
}

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

repositories {
    mavenCentral()
}

dependencies {
    compileOnly 'org.projectlombok:lombok:1.18.12'
    annotationProcessor 'org.projectlombok:lombok:1.18.12'
    testCompile group: 'junit', name: 'junit', version: '4.12'
}
```

### 2. IntelliJ IDEA에서 [Ctrl + Shift + A] 단축키를 눌러서 Actions 활성화 : Actions에 "plugins" 검색 및 실행



All Classes Files Symbols Actions

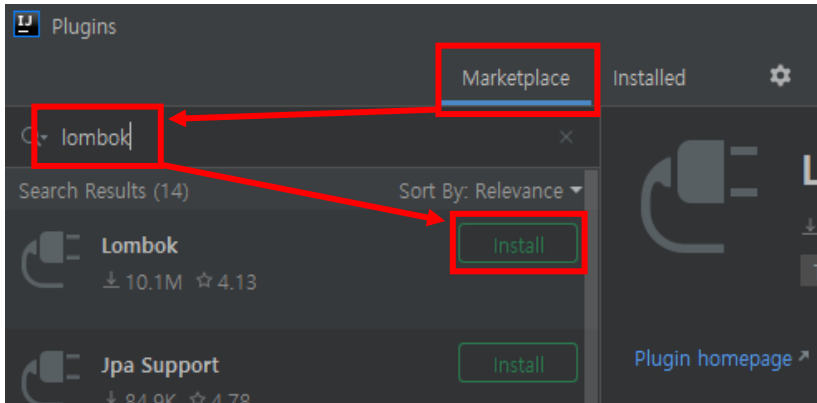
plugins

Plugins

Analyze Plugin Startup Performance

3. Plugins에서 Lombok 설치

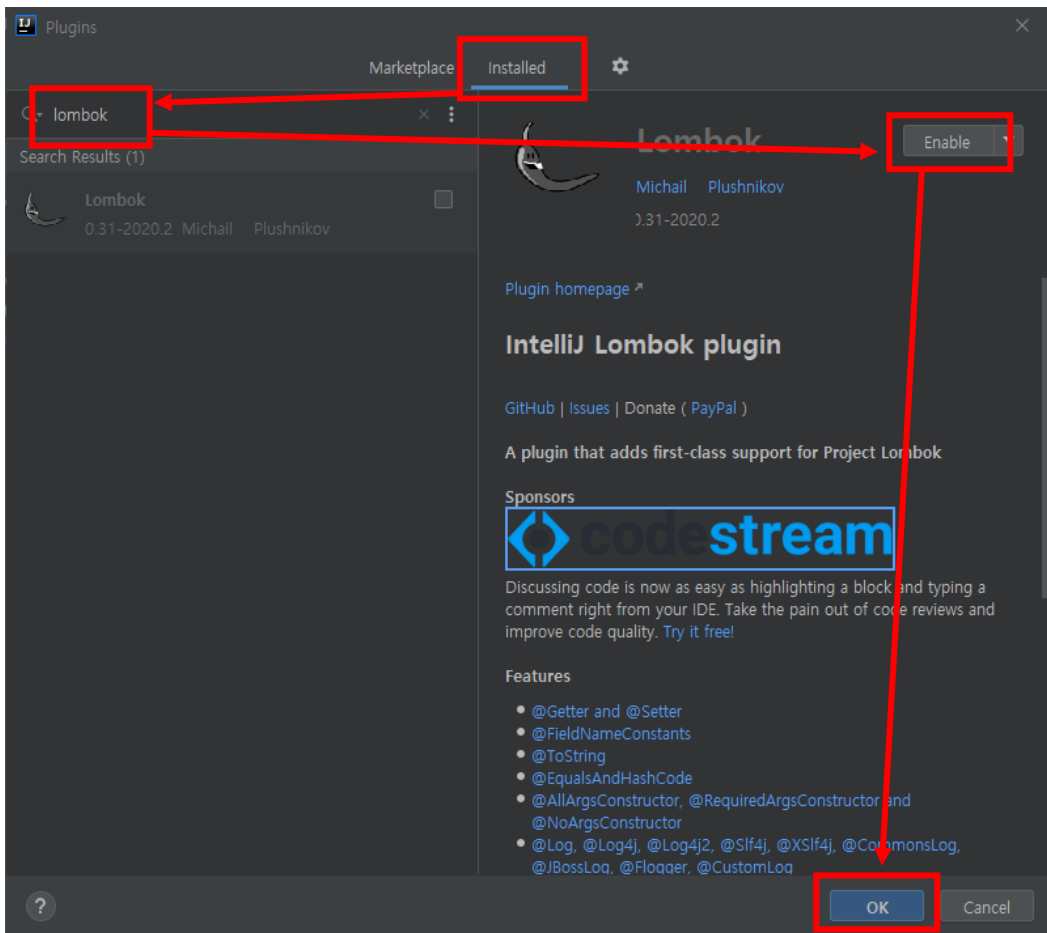
: Marketplace -> Lombok 검색 -> Install 클릭



4. Plugins에서 Lombok Enable

: Installed -> Lombok 검색 -> Enable 클릭 -> Ok 클릭

( ! 기존에 Lombok이 Enable되어 있다면 따라하지 않아도 된다. )



5. 클래스를 생성하기 위한 Package 생성

: src/main/java 오른쪽클릭 -> New -> Package 왼쪽클릭

6. 패키지 이름은 자신이 원하는 이름으로 작성

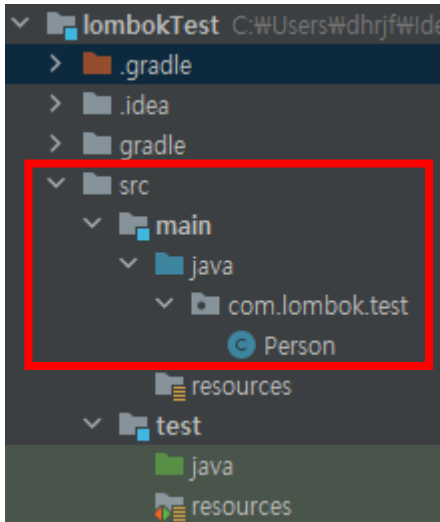
: 본 실습에서는 "com.lombok.test"로 작성하였다.

7. 생성된 package에 새로운 클래스 생성

: src/main/java/com.lombok.test 오른쪽클릭 -> New -> Java Class 왼쪽클릭

8. 클래스 이름을 **"Person"**으로 작성

9. [8.]까지 진행되었다면 프로젝트가 아래와 같이 구성되었을 것이다.



10. Person 클래스의 내용을 아래 박스와 같이 입력

```
package com.lombok.test;

import lombok.*;

@Getter // getName(), getAge() 자동생성 어노테이션
@Setter // setName(), setAge() 자동생성 어노테이션
@NoArgsConstructor // Person() 자동생성 어노테이션
@AllArgsConstructor // Person(name, age) 자동생성 어노테이션
public class Person {
    private String name;
    private int age;
}
```

11. Person 클래스를 테스트 하기위한 테스트 클래스 작성

12. 테스트 클래스를 생성하기 위한 package 생성

: src/**test**/java 오른쪽클릭 -> New -> package 왼쪽클릭

13. package 이름은 위 src/main/java에 작성한 패키지 이름과 동일하게 작성

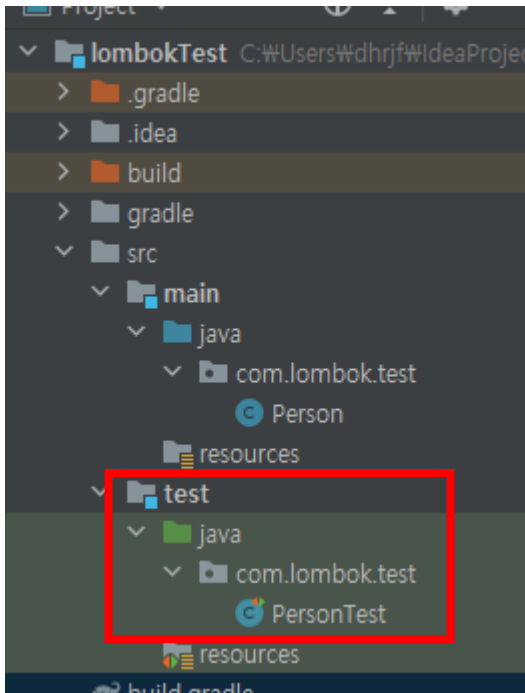
: 본 실습에서도 src/main/java와 동일하게 "com.lombok.test"로 작성하였다.

14. 생성된 package에 테스트 클래스 생성

: src/test/java/com.lombok.test 오른쪽클릭 -> New -> Java Class 왼쪽클릭

15. 클래스 이름은 **"PersonTest"**로 작성

16. [15.]까지 진행되었다면 프로젝트가 아래와 같이 구성되었을 것이다.



17. PersonTest 클래스의 내용을 아래 박스와 같이 입력

```
package com.lombok.test;

import org.junit.Assert;
import org.junit.Test;

public class PersonTest {

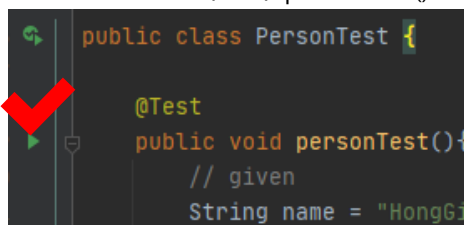
    @Test
    public void personTest(){
        // given
        String name = "HongGilDong";
        int age = 20;

        // when
        Person person = new Person(name, age);

        // then
        Assert.assertEquals(person.getName(), name); // given에서 name과 생성된 객체의 name이 동일한지 확인
        Assert.assertEquals(person.getAge(), age); // given에서 age와 생성된 객체의 age가 동일한지 확인
    }
}
```

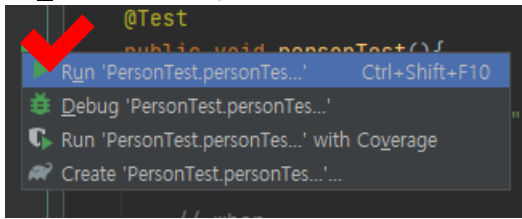
18. 테스트 코드 실행 (1)

: PersonTest 클래스의 personTest() 메소드 왼쪽의 초록색 화살표 클릭

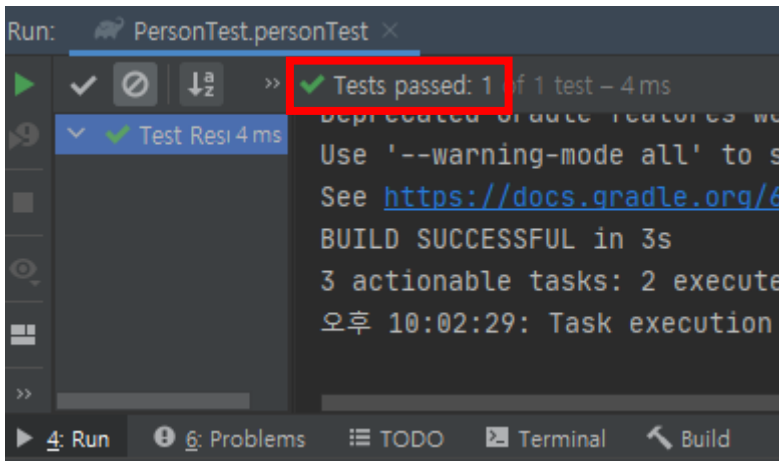


## 19. 테스트 코드 실행 (2)

: Run 'PersonTest.personTes...' 왼쪽클릭



## 20. 테스트 코드 결과 확인



: Tests passed가 뜬다면 정상적으로 테스트가 완료되었음을 의미한다.

: 즉, 롬복의 @Getter, @Setter등의 어노테이션이 정상적으로 Person 클래스에 등록되었단 의미이다.

## 참고문헌

- 버터필드 (2020). <https://atoz-develop.tistory.com/entry/IntelliJ%EC%97%90%EC%84%9C-Lombok-%EB%9D%BC%EC%9D%B4%EB%B8%8C%EB%9F%AC%EB%A6%AC%EB%A5%BC-%EC%82%AC%EC%9A%A9%ED%95%98%EA%B8%B0-%EC%9C%84%ED%95%9C-%EC%84%B8%ED%8C%85-%EB%B0%A9%EB%B2%95>