Lombok은 자바 개발자들이 간결하고 보일러플레이트 코드를 줄이기 위해 사용하는 라이브러리입니다. Lombok을 사용하면 일반적으로 반복적으로 작성해야하는 메서드, 생성자, getter, setter 등을 어노테이션을 통해 자동으로 생성할 수 있습니다. 아래는 Lombok의 주요 기능과 사용법에 대한 간단한 설명입니다.

롬복 설치하기: 프로젝트에서 롬복을 사용하기 위해 먼저 해당 라이브러리를 설치해야 합니다. 롬복은 주로 빌드 도구인 Maven이나 Gradle을 사용하는 프로젝트에서 의존성을 추가하여 사용합니다.

클래스 레벨 어노테이션 사용:

@Data: 모든 필드에 대한 Getter, Setter, EqualsAndHashCode, ToString 등을 생성합니다.

@Getter와 @Setter: 필드별로 Getter와 Setter를 생성합니다.

@NoArgsConstructor: 파라미터 없는 기본 생성자를 생성합니다.

@AllArgsConstructor: 모든 필드를 인자로 받는 생성자를 생성합니다.

@RequiredArgsConstructor: final이나 @NonNull인 필드만 인자로 받는 생성자를 생성합니다.

메서드 레벨 어노테이션 사용:

@Builder: 빌더 패턴을 사용하여 객체를 생성하는 빌더 클래스를 생성합니다.

@SneakyThrows: checked exception을 catch하지 않아도 되도록 합니다.

예시:

java

Copy code

import lombok.\*;

@Data

@NoArgsConstructor

@AllArgsConstructor

public class Person {

private String name;

private int age;

}

// @Data 어노테이션은 모든 필드에 대해 Getter, Setter, EqualsAndHashCode, ToString 등을 자동으로 생성합니다.

// @NoArgsConstructor는 파라미터 없는 기본 생성자를 생성합니다.

// @AllArgsConstructor는 모든 필드를 인자로 받는 생성자를 생성합니다.

// 사용 예시:

public class Main {

public static void main(String[] args) {

Person person = new Person("John", 30);

System.out.println(person.getName()); // "John" 출력

System.out.println(person.getAge()); // 30 출력

}

}

위와 같이 Lombok 어노테이션을 사용하면 Getter, Setter, 생성자 등을 명시적으로 작성하지 않아도 됩니다. 이를 통해 코드의 가독성을 높이고 개발 시간을 단축시킬 수 있습니다. 하지만 프로젝트를 진행할 때, Lombok을 사용한다면 개발 환경에서 Lombok이 정상적으로 동작하도록 롬복 플러그인을 설치해야 합니다. IDE에 따라 Lombok 플러그인 설치 방법이 다를 수 있으니 해당 IDE에 맞는 방법으로 설치해주시기 바랍니다.