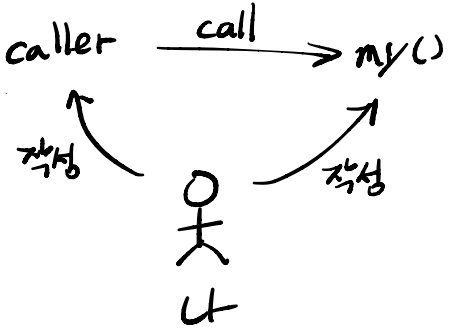
* Inversion of Control (I.O.C)

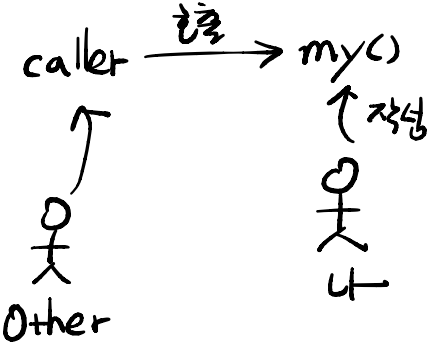
↳ 제어의 역행

① 메서드 호출

일반



역행



예) (Servlet의 메서드
Listener의 메서드)

* Inversion of Control (I.O.C) 의존성

↳ 제어의 역전

@ 객체 사용

일반

```
class Car {
    Engine e = new Engine();
    void run(){
        e.start();
    }
}
```

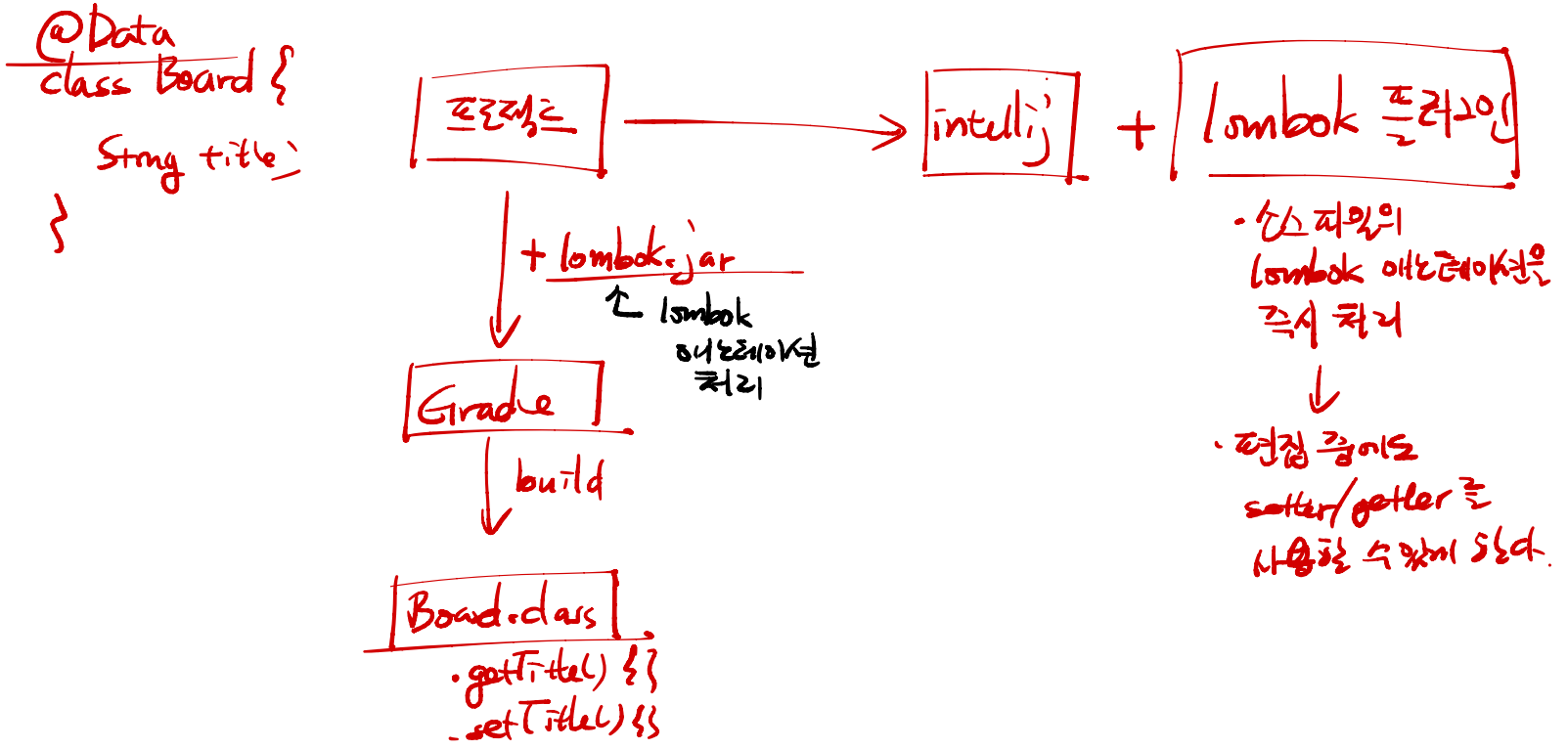
사용할 객체를 직접 생성한다

"DI"

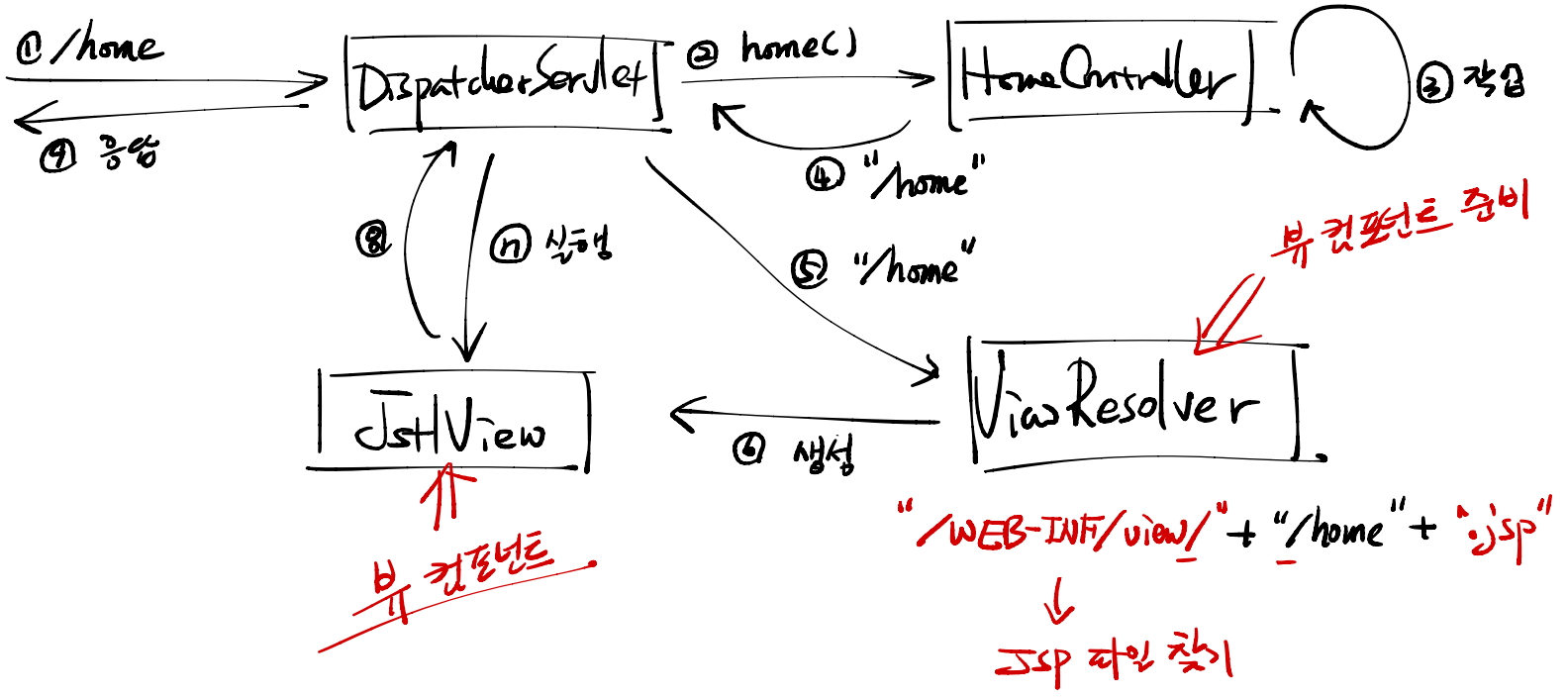
Dependency 의존객체 Injection

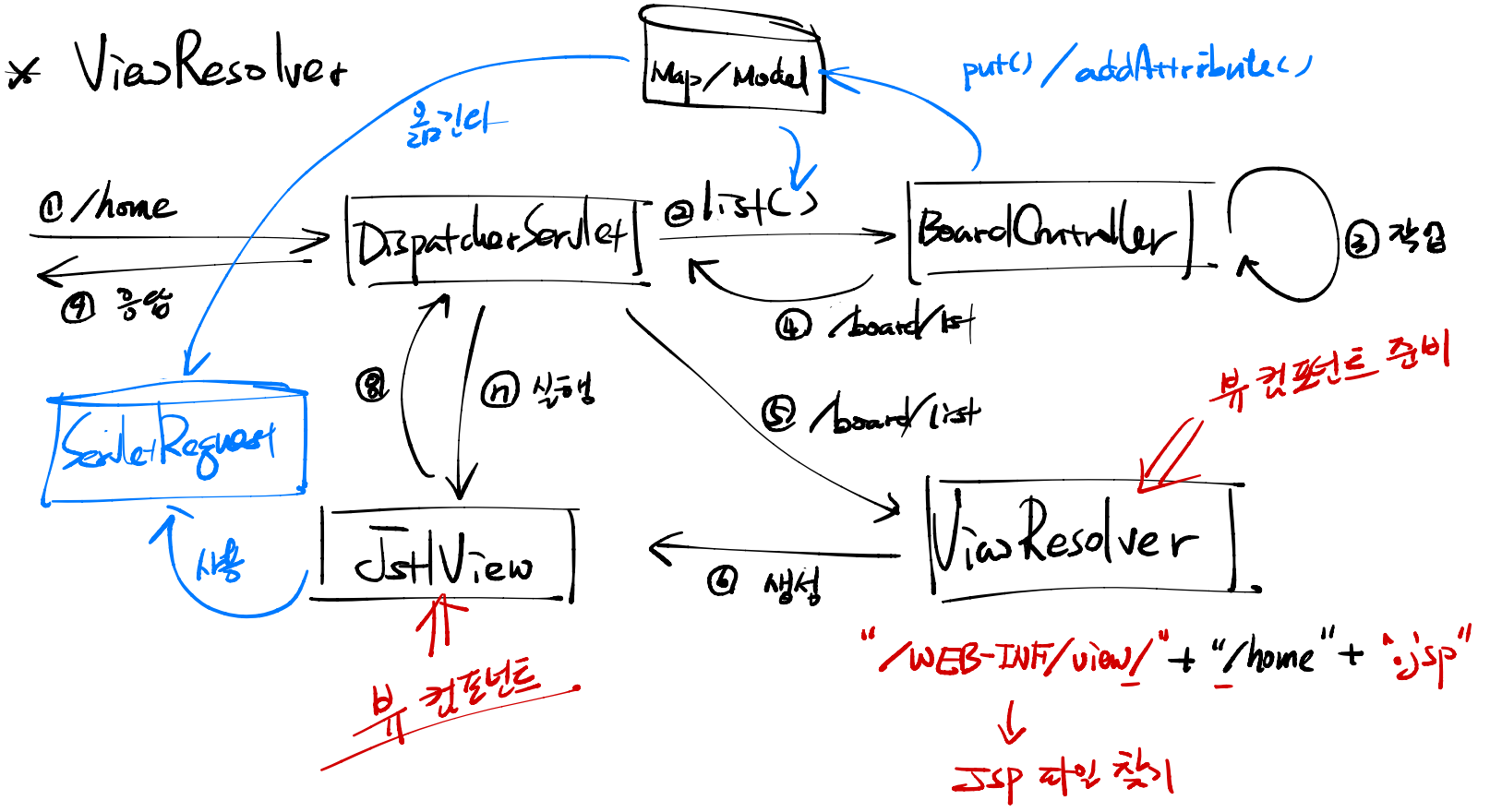
```
class Car {
    Engine e;
    public Car(Engine e){
        this.e = e;
    }
    void run(){
        e.start();
    }
}
Engine e1 = new Engine();
Car car = new Car(e1);
```

= Car가 사용할 객체를 외부에서 주입받음


@Data
class Board {
String title;
}
프로젝트
+ lombok.jar
lombok 애노테이션 처리
Gradle
build
Board.class
.getTitle() {}
.setTitle() {}
intellij
+
lombok 플러그인
• 소스 파일의 lombok 애노테이션을 즉시 처리
• 컴파일 할 때도 setter/getter를 사용할 수 있게 된다.

* ViewResolver




* ViewResolver
Map / Model
put() / addAttribute()
요청
① /home
DispatcherServlet
② list()
BoardController
③ 작업
④ /board/list
⑤ /board/list
⑥ 생성
⑦ 실행
⑧
⑨ 응답
ServletRequest
사용
JstlView
ViewResolver
뷰 컴포넌트
뷰 컴포넌트 준비
"/WEB-INF/view/" + "/home" + ".jsp"
JSP 파일 찾기