

1. 상황

가. 자바기반 어플리케이션은 최종 사용자에게 결과를 보여주기 위해 상호 협력하는 몇 개의 객체를 가짐

나. 복잡한 자바 어플리케이션은 가능한 다른 자바클래스들과 독립적으로 실행해야,

- 재사용성이 높고
- 독립적인 단위 테스트 가능

2. 의존성 주입(Dependency Injection or wiring)

가. 각각의 클래스를 독립적인 상태로 유지하면서 연결하는데 도움

3. 예

public class TextEditor { => 표준코드

private **SpellChecker** spellChecker;

public TextEditor() {

spellChecker = new SpellChecker();

}

}

public class TextEditor { => DI 코드

private **SpellChecker** spellChecker;

public TextEditor(SpellChecker spellChecker) {

this.spellChecker = spellChecker;

}

}

4. 의존성 주입 타입

4.1 생성자 기반 의존성 주입

가. 프로젝트 생성: spring_console_constructDI

나. 패키지 생성: com.kh.kht

다. 클래스 생성: TextEditor/SpellCheker/MainApp

라. 빈즈 설정파일 생성: Beans.xml

4.2 세터 기반 의존성 주입

가. 프로젝트 생성: spring_console_setterDI

나. 패키지 생성: com.kh.kht

다. 클래스 생성: TextEditor/SpellCheker/MainApp

라. 빈즈 설정파일 생성: Beans.xml

4.3 스프링 내부 빈 주입(Spring Injecting Inner Beans)

- 가. 프로젝트 생성: spring_console_innerbeanDI
- 나. 패키지 생성: com.kh.kht
- 다. 클래스 생성: TextEditor/SpellCheker/MainApp
- 라. 빈즈 설정파일 생성: Beans.xml

4.4 스프링 컬렉션 주입(Spring Injecting Collection)

4.4.1 개요

- 가. 빈에 단일값을 넘기기
 1. 빈 설정 파일에서 프로퍼티 태그 **<property>**를 사용해서 **value** 속성을 사용한 원시데이터 타입, **ref** 를 사용한 객체 참조 설정
spring_console_beanfactory: value 예제
spring_console_setterDI: ref 예제
 2. 컬렉션 값 넘기기: List -> 실습

4.4.2 실습

- 가. 프로젝트 생성: spring_console_collectionDI
- 나. 패키지 생성: com.kh.kht
- 다. 클래스 생성: TextEditor/SpellCheker/MainApp
- 라. 빈즈 설정파일 생성: Beans.xml