**Spring 필기**

**1 스프링 개요**

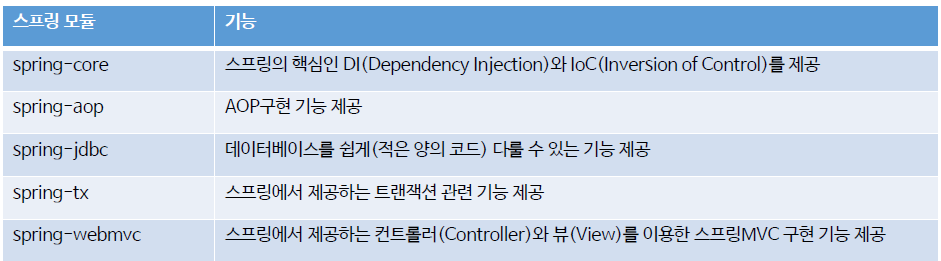
* 1. **스프링 프레임워크**

스프링 프레임워크는 주요 기능으로 DI, AOP, MVC, JDBC 등을 제공한다.

프레임워크는 개발자들의 업무를 도와주는 틀. (안전 운전만 할 수 있도록 해주는 네비게이션)

프레임워크가 없다면 모든 기능을 개발자가 고민, 개발해야 하는 번거로움이 있다.

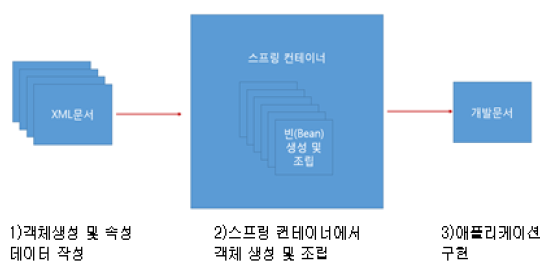
**1.2 스프링 프레임워크 모듈**



모듈을 사용하려면, 모듈에 대한 의존설정을 개발 프로젝트의 XML 파일 등을 이용해서 개발자가 직접 하면 된다.

* 1. **스프링 컨테이너(IoC)**

스프링에서 객체를 생성하고 조립하는 컨테이너(Container)로, 컨테이너를 통해 생성된 객체를 빈(Bean)이라고 한다.



**3 스프링 프로젝트 생성**

**3.1 프로젝트 생성**

Maven Project를 생성한다.

**3.2 pom.xml 작성**

**3.3 폴더 및 pom.xml 파일의 이해**

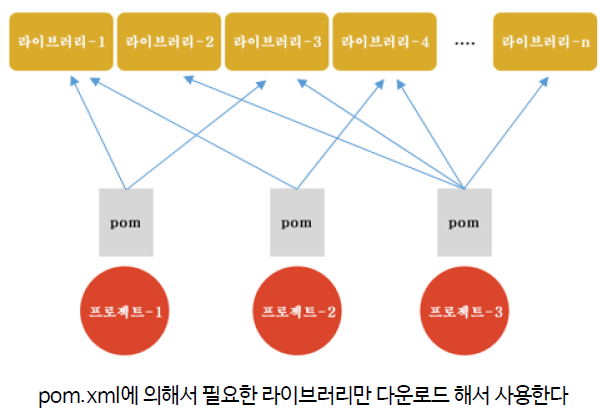
ch\_xx\_project/src/main/java 폴더 : 개발자가 자바 언어를 이용해 실제로 개발하는 부분

ch\_xx\_project/src/main/resources 폴더 : 개발함에 있어서 여러 보조적인 역할을 하는 파일들이 모인 부분. 스프링 설정 파일(XML) 또는 프로퍼티 파일 등이 관리되는 부분이다.

src/main 폴더가 java와 resources 폴더로 나뉜다는 것을 주의하자. 스프링 프레임워크의 기본 구조를 이루는 폴더로, 개발자는 이대로 폴더를 유지해야 한다.

pom.xml 파일은 메이븐 설정파일로, 메이븐은 라이브러리를 연결해주고, 빌드를 위한 플랫폼이다.

내 프로젝트에 필요한 라이브러리를 포함시켜주는 파일.



**4 처음 해보는 스프링 프로젝트**

**4.1 Java 파일을 이용한 프로젝트 실행**

평범한 java 파일을 생성해서 실행한다.

**4.2 우선 따라해보는 스프링 프로젝트**

resources에 xml 파일로 컨테이너와 bean을 설정해준다. 이렇게 해주면 스프링 콘테이너에 bean 객체가 생성이 되기 때문에 코드에서 일일이 객체를 new로 생성하지 않아도 된다.

**5 또 다른 프로젝트 생성 방법**

**5.1 폴더(java, resources)와 파일(pom.xml) 만들기**

직접 ch\_05\_ownproject-src-main-java, resources 폴더 만들기

pom.xml도 에디터를 이용해 만든다.

**5.2 이클립스에서 import하기**

이클립스에서 ch\_05를 import해서 java파일 생성 후 테스트 해보기

**6 DI (Dependency Injection)**

**6.1 DI(Dependency Injection) 란?**

DI, 의존 주입 : 객체를 만들어서 외부에서 주입하는 방식.

객체지향 프로그램의 유연성

1) 배터리 일체형 장난감 : 배터리가 떨어지면 장난감을 새로 구해야 한다.

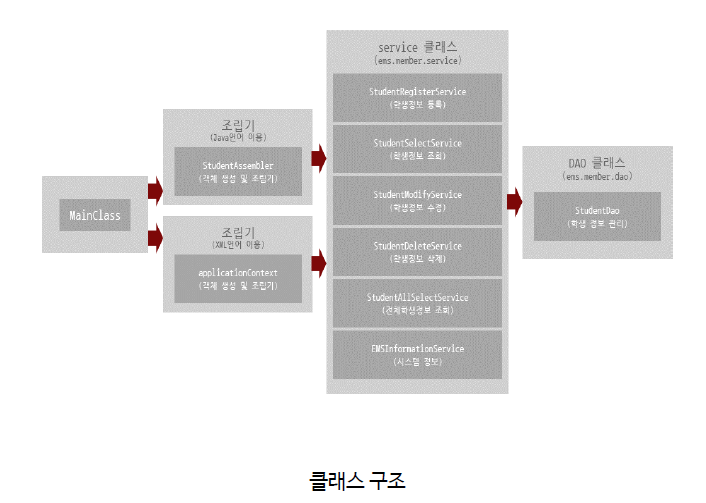
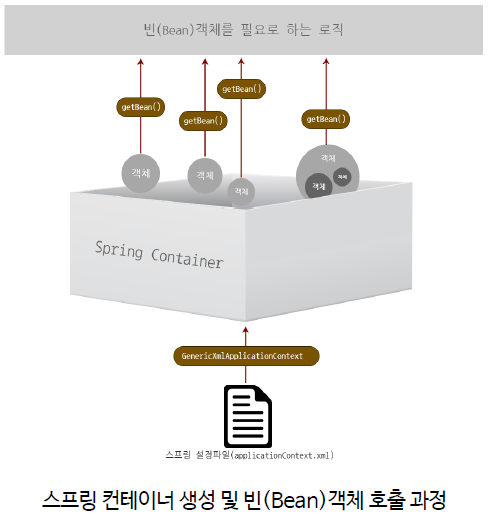
생성자에서 배터리를 생성한 후 변경 불가능

2) 배터리 분리형 장난감 : 배터리가 떨어지면 배터리만 교체하면 된다.

2-1) 생성자에서 장난감만 만들고 배터리를 setter로 설정

2-2) 생성자에서 배터리도 설정한 후 setter로 설정 -> 가장 유연하다.

**6.2 스프링 DI 설정 방법**



스프링 컨테이너에 빈(bean) 객체가 다 모여있다. 따라서 컨테이너에서 객체끼리 의존 주입이 되어있어야 한다.

**7 다양한 의존 객체 주입**

**7.1 생성자를 이용한 의존 객체 주입**

<constructor-arg> 태그 이용해서 매개변수 설정.

**7.2 setter를 이용한 의존 객체 주입**

property 태그에 멤버 변수를 name, 값을 value로 해서 설정

**7.3 List 타입 의존 객체 주입**

property와 name은 그대로 쓰고 <list>태그를 이용해 그 안에 <value>를 여러개 쓴다.

**7.4 Map 타입 의존 객체 주입**

<map>으로 묶고 <entry>로 묶어서 key와 value를 설정한다.

**8 스프링 설정 파일 분리**

**8.1 스프링 설정 파일 분리**

설정 파일이 너무 길어지면 비효율적이므로 효율적으로 관리하기 위해 설정 파일을 분리한다.

applicationContext.xml에 모두 설정해둔 것을 기능별로 나눠 appCtx1.xml, appCtx2.xml, appCtx3.xml으로 분리한다. 이름도 기능에따라 설정하는 게 좋다.

main 함수에서는 string 배열을 이용해 각 xml 파일을 로드한다.

import 태그를 이용해 Ctx1.xml Ctx2.xml을 import해서 한 파일로 xml을 합칠수도있다. 보통 배열로 사용하는 방법을 더 선호한다.

**8.2 빈(Bean)의 범위**

싱글톤(Singleton) : 스프링 컨테이너에 생성된 빈(Bean) 객체의 경우 동일한 타입에 대해서는 기본적으로 **한 개만 생성**이 되며, getBean() 메소드로 호출될 때 동일한 객체가 반환된다.

-> 컨테이너가 이미 객체를 한 개씩 생성했기 때문에, 생성된 동일한 객체를 반환하게 된다.

프로토타입(Prototype) : 싱글톤 범위와 반대의 개념. 프로토타입의 경우 개발자는 별도로 설정을 해 줘야 하는데, 스프링 설정 파일에서 빈(Bean)객체를 정의할 때 scope 속성을 명시해 줘야한다.

ex) <bean id="classA" class="com.ClassA" **scope="prototype"**>

getBean으로 객체를 참조할 때마다 새로운 객체를 생성하게 된다.

싱글톤이 디폴트이므로 필요할 때만 프로토타입을 설정한다.