spring framework
"용어" "개념" "여전히 프로그래밍(강제적 규칙 틀)"
- DB연동기능, WEB기능 등의 다양한 프로그래밍 개발 규칙을 포함한 통합 기능 자바 프레임워크

framework
0> 어떤 것을 구성하는 뼈대 – 틀 내부 범위 작업
틀(=자바클래스/인터페이스 제공) – 동일하다
작업 – 다르다
예)
class A implements 인터페이스{메소드오버라이딩}

class 프레임워크클래스 {
메소드(A a1) {}
}

class A{}
class B extends A{}
1> 기능을 미리 만들어 제공하는 반제품
2> 어느 정도 완성된 상태의 라이브러리 제공-사용자마다 동일하다
3> 2번의 라이 브러리에 포함된 기능은 사용자가 구현할 필요가 없다
4> 사용자 구현 프로그램-사용자마다 다르다
5> "new" 객체생성 최소화
6> POJO 클래스 사용 가능 – PLAIN OLD JAVA OBJECT
---> 이전 자바 클래스들 스프링 내부 사용 가능(단 규칙)
DTO, DAO
class A{}--> main()/servlet,jsp/spring
class B extends HttpServlet{doGet | doPost }--> servlet,jsp

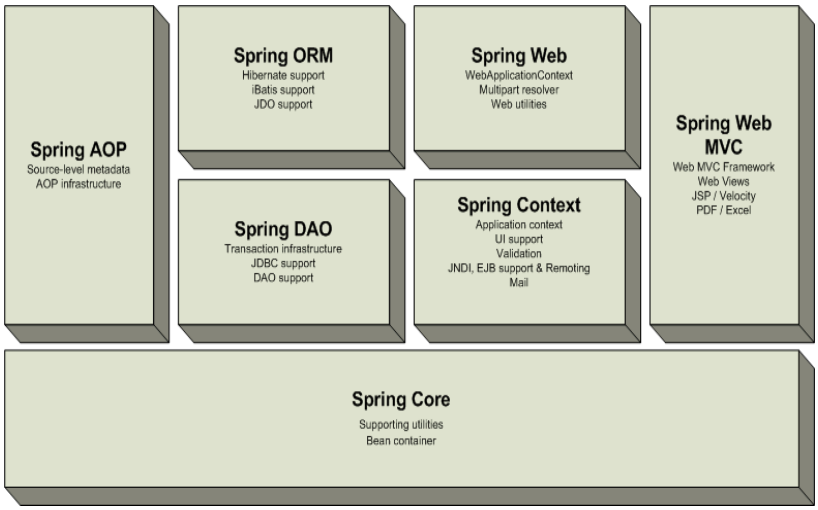
프레임워크 규칙 + 연결 고리 + 규칙내부 자유롭게 작성
일정 틀 내부 프로그램 작성 스프링 연동
스프링 규칙 알아보자
SPRING IOC
SPRING DI
SPRING MVC
SPRING JDBC
spring aop
MYBatis – DB 연동 기능 프레임워크
자바언어 구현

스프링 규칙 라이브러리(상속, 매개변수 xxx 요구) + 사용자 클래스
MemberDTO, MemberDAO – 상속 메소드 매개변수 규칙 없었다
+ SPRING 내부 사용
POJO 사용 가능
= PLAIN OLD JAVA OBJECT

자유롭게 프로그램 작성
class A { 리턴타입 m(매개변수) {} }
서블릿 규칙 작성
class A extends HttpServlet {
  public void doGet(HttpServletRequest request, HttpServletResponse response){}
}

- 자바 framework
spring -> 통합 기능 제공 프레임워크 (MVC, JDBC, ... ,)
mybatis-> jdbc 개선 제공 프레임워크(DB FRAMEWORK)

SPRING MVC + MYBATIS



sts 3 – spring tools suite 3

스프링 개발 가능 이클립스

자바, servlet, jsp, html...+spring

1> 컴퓨터 jdk 버전 여러개 설치 가능
2> 기본 버전
도스 –
java version
3>
4> sts3 --> jdk 11버전 필요

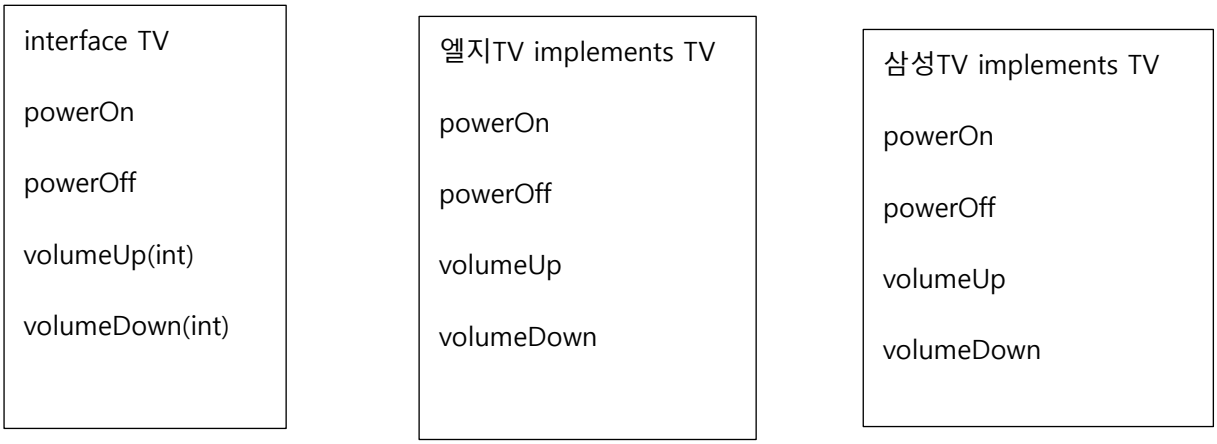
sts

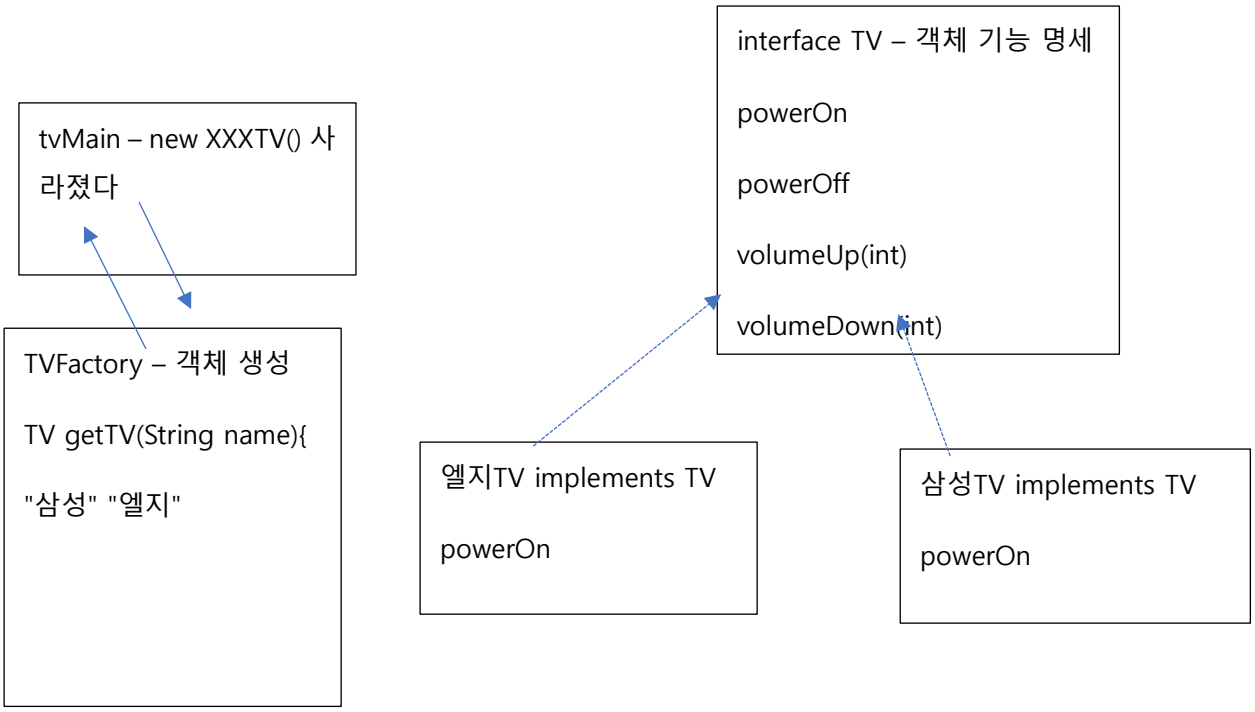
eclipse

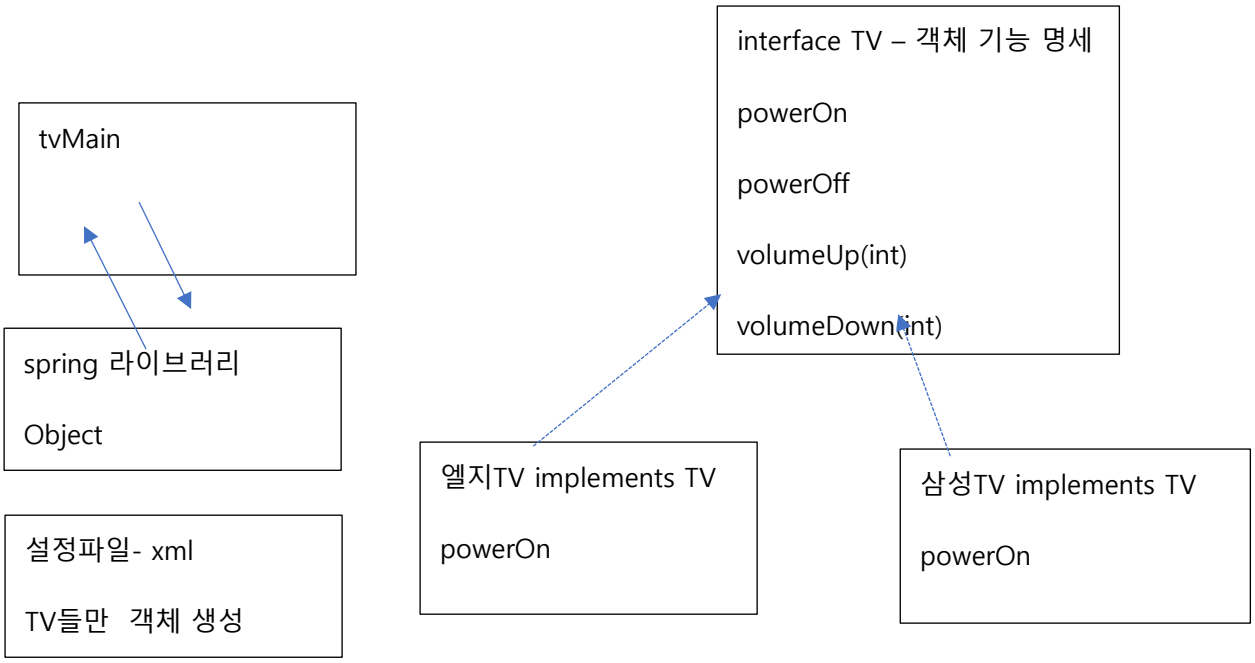
- 자바 클래스

19장 스프링 의존성 주입과 제어 역전 기능

스프링 의존성 주입 – spring DEPENDENCY INJECTION(DI)
SPRING DI 기능




interface TV – 객체 기능 명세
powerOn
powerOff
volumeUp(int)
volumeDown(int)
tvMain – new XXXTV() 사라졌다
TVFactory – 객체 생성
TV getTV(String name){
"삼성" "엘지"
엘지TV implements TV
powerOn
삼성TV implements TV
powerOn



1> 객체생성 필요한 클래스에서 "직접객체생성x"
2> 스프링- xml -내가만든클래스객체
3> dependency injection – 객체 관리 (생성-소멸..)

MemberDTO / MemberDAO==> 사용자 객체 생성 필요

/ MamberMain ==> 사용 클래스

/ + spring 라이브러리 / + *.xml

1> spring bean configuration file

<bean id="" class=""/>	**< bean id="" class="" > <property name="" value나 ref중 하나 /> </bean>**	**<bean id="" class=""> <constructor-arg name="" value나 ref 중 하나 /> </bean>**
객체생성-setter값 없다 외부로부터 전달값 없다	**객체생성 외부로부터 전달값 있다 setter 메소드 통해**	**객체생성 외부로부터 전달값 있다 생성자 통해**
DEPENDENCY INJECTION	**SETTER INJECTION**	**CONSTRUCTOR INJECTION**

DEPENDENCY INJECTION

1> SETTER INJECTION

```
class A{
  String id;
  int pw;
  Cart cart; //객체 변수 포함 선언
  setCart(Cart c){  cart = c; }

<bean id="a1" class="A" >
 <property name="cart" ref="Cart생성bean id값" />
```

2> CONSTRUCTOR INJECTION

```
class A{
  String id;
  int pw;
  Cart cart; //객체 변수 포함 선언
  A(String id, int pw, Cart cart){.....}

<bean id="a1" class="A" >
 <constructor-arg name="id" value="" />
<constructor-arg name="pw" value="" />
<constructor-arg name="cart" ref="Cart생성bean id값" />
```

DI – 현재 자바 클래스에서 객체 생성 주도권 없다

스프링 생성 전달 – 스프링 객체 전달 주도권 갖는다

스프링 제어 역전 기능 – SPRING INVERSION OF CONTROL(IOC)

ioc 개념 만족 스프링 프로그램 기법 di – setter / constructor

19 DI / IOC

프로그래밍 개발 패턴- spring "규칙"

스프링 비즈니스 로직 – service + dao +dto = pojo클래스

1> xml 태그 --> 2> @Service
1+2

26장 di annotation

interface
Service

MAIN

MemberService1

메소드 1개=1개 기능

(2개 이상 sql 실행)

DAO dao = new DAO();

회원가입(){

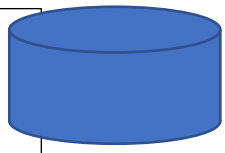
boolean r = dao.selectMember(...);

if(r == false){

dao. insertMember(.....);

dao.insertEmp();

}



MemberDAO

메소드1개단위=1개 SQL실행결과

boolean selectMember(id, pw){

1>SELECT ID FROM MEMBER

WHERE ID=?

}

void insertMember(.....){

2>INSERT INTO MEMBER VALUES(?,...)

}

void insertEmp(){

INSERT INTO EMP VALUES

}

SERVLET/JSP

(DataSource)

spring

MemberService2

메소드 1개=1개 기능

(2개 이상 sql 실행)

DAO dao = new DAO();

회원가입(){

boolean r = dao.selectMember(...);

if(r == false){

dao. insertMember(.....);

}

MemberDTO- 임시 저장소