

## -jsp

- 1> 자바 문장 - 서블릿 api + 자바 api 사용
- 2> 응답 출력 간결 쉽게 = 웹디자이너 공유
- 3> 자바 문장 대신 수행 태그들
- 4> <jsp:xxx
- 5> el \${ el }
- 6> jstl 액션태그 없는 기능들 추가 라이브러리

<ul style="list-style-type: none"> <li>- 데이터 타입</li> <li>논리값- true, false</li> <li>정수</li> <li>실수</li> <li>문자열 - 'aaaa', "aaaa"</li> <li>null - null</li> </ul>	<ul style="list-style-type: none"> <li>- 연산자</li> <li>문자열 결합 : +=</li> <li>/ , div</li> <li>%, mod</li> <li>&gt; &lt; : gt, lt</li> <li>null -&gt; 공백 출력</li> <li>+ 사용 -&gt; 0으로 변경</li> </ul>
---	--

jsp내장객체	el내장객체
request.getParamemter	param.파라미터명 paramValues.파라미터명[i]
pageContext	pageScope
request	requestScope
session	sessionScope
application	applicationScope

## jstl

- 1> 현재 프로젝트\webcontent\web-inf\lib\jstl-xxx.jar 파일 위치
  - 2> 현재 jsp 파일
- <%@ taglib prefix="c" uri=<http://java.sun.com/jsp/jstl/core> %>

<c:set	
<c:out	

<c:remove	
<c:if	<c:if test="{10<20}" > 바디 </c:if>
<c:choose	<c:choose> <c:when test="{10<20}" > 조건 만족시 수행 </c:when> <c:when test="{10>20}" > 조건 만족시 수행 </c:when> ... <c:otherwise > 위 조건 만족하지 않는 경우 수행 </c:otherwise> </c:choose>
<c:forEach	<c:forEach begin="1" end="100" step="1"  <c:forEach items="배열 리스트 " var = "vo" > \${vo.XXX} </

	<pre> &lt;c:forEach items="맵" var = "vo" &gt; \${vo.key} \${vo.value.memberid} &lt;/ &lt;c:forEach items="....." varStatus="vo" &gt; \${vo.current.key} \${vo.current.value.memberid} &lt;/ </pre>
--	--

=====

## 18장 스프링 프레임워크 시작하기

### - 스프링 환경 설정

1> sts4 툴 새로 설치 – spring3추가, web, server 기능 사용 추가

==> \*.zip

2>spring.io

sts3 툴 다운로드 – spring 4 추가,

3> eclipse 툴 내부 – spring3 추가(eclipse web, server 기능 이미 존재)

java project

dynamic web project

spring project=> 자바클래스 + 서블릿 + jsp + html...+스프링

spring boot project

- 프레임워크 -

1> 일정 규격 맞추어 프로그래밍. 강제화

2> 뼈대 공통 설계+몸통 (사용자) 구현

3> 반 이미 구현 제공 + 나머지 반 구현 완성품-> 간결

4>

```
class A {  
    A( B b1 )    {    }  
    a1() 반드시  
    a2() x  
}
```

```
interface B{    m() ; }  
class B1 implements B {m1() {....}    }  
class B2 implements B {m1() {....}    }  
class C1 {m1() {....}    }
```

```
A a1 = new A(new B1());  
A a2 = new A(new B2());  
A a3 = new A(new C1());--.ERROR
```

struts 프레임워크 / ejb 기술 대신

==>

spring 프레임워크- 자바 구현 대표적 프레임워크 툴

ioc, di, aop, mvc, 다른 프레임워크 연동

<b>interface TV</b> <b>powerOn()</b> <b>powerOff()</b> <b>soundUp()</b> <b>soundDown()</b>	
<b>SamsungTV implments TV</b>	<b>LgTV implments TV</b>
<b>TVFactory</b> <b>TV getTV(String name)</b>	<b>TVMain</b> 1> TV 객체 변경시에도 기능 메소드 수정 필요없다 2> TVFactgory 전달 TV객체 사용 가능하고 의존하다 3> 소스 코드 변경시 최소화

1> 여러개 클래스에 규칙 강제화

2> TV 객체 변경시에도 기능별 메소드 수정 필요없다

TV tv = xxxxTV();

**IOC ==> INVERSION OF CONTROL**

1>제어의 역전

2>

원하는 객체 생성하고자 한다면 클래스 내부 new XXXX()

원하는 객체 수정하고자 한다면 클래스 내부 new XXXX()

==>MAIN 객체 생성 제어권 기본 있다(제어 역전 아니다)

3>

==>TVFactory 객체 생성 제어권 가지고 main 은 제어 역전

==> 객체 타입 한정하자 -인터페이스

4>

**DI ==> dependency injection**

====> IOC 개념 구현 방법

main 전달받아야 한다 = 주입받아야 한다

tvfactory 의존하여 생성한 객체만 주입받는다

spring은 ioc 기능을 지원하며 di 구현 방법 사용한다

==> java구현==> spring

interface TV powerOn() powerOff() soundUp() soundDown()	
SamsungTV implments TV	LgTV implments TV
spring factory 역할 api	TVMain spring 프레임워크에서 di 전달 객체 사용
spring bean configuraion xml  jsp bean – jsp 자바객체 spring bean -spring 자바객체	스프링 설정 파일

DI: 의존성 주입

게시판 –

BoardService.java

BoardDAO.JAVA

BoardVO.java

- DAO와 VO 이용하기( 자바 APPLICATION, SERVLET,JSP, SPRING)

1>프레임워크다(반 완성 + XML + 나머지 반 구현)

2>경량의 컨테이너다

3>독립적 실행

4> pojo 클래스 – plain old java object

5> ioc, di

===>

SERVLET,JSP --> TOMCAT(...), JDK

class A extends HttpServlet{}

JAVA APP – JDK

MemberDAO

selectMember()

insertMember()

MemberMain

.....

MemberVO

spring

factory api

member.xml

,<bean

<property

<constructor-arg

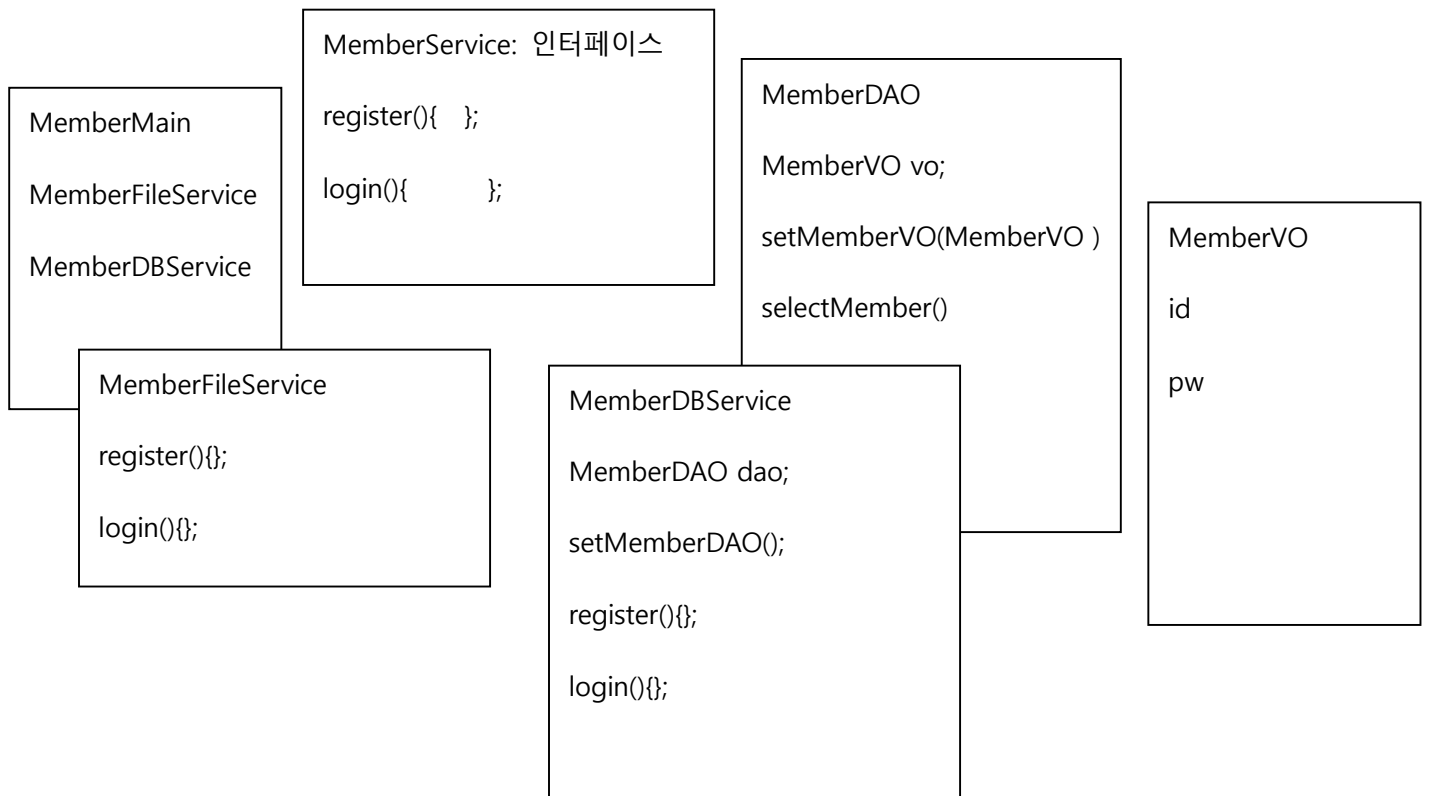
- SERVICE 추가하기

dao- 메소드 1개 단위 - sql 1개 실행단위 구체적

기능=서비스 1개 단위- 추상적

login(){ selectMember() }

register(){ selectMember() 후 insertMember() }







- 1> spring pojo 클래스(dao, vo, service) 구현한다
- 2> 서로 다른 클래스 공통 메소드 필요 - 인터페이스
- 3> spring 연결 - 객체 생성, 객체 관계 설정

### 3-1. xml

<bean <property <constructor-arg

### 3-2. 자바

annotation - 자바 실행 환경(jdk, tomcat, spring)으로 알려주는 설명

## spring annotation

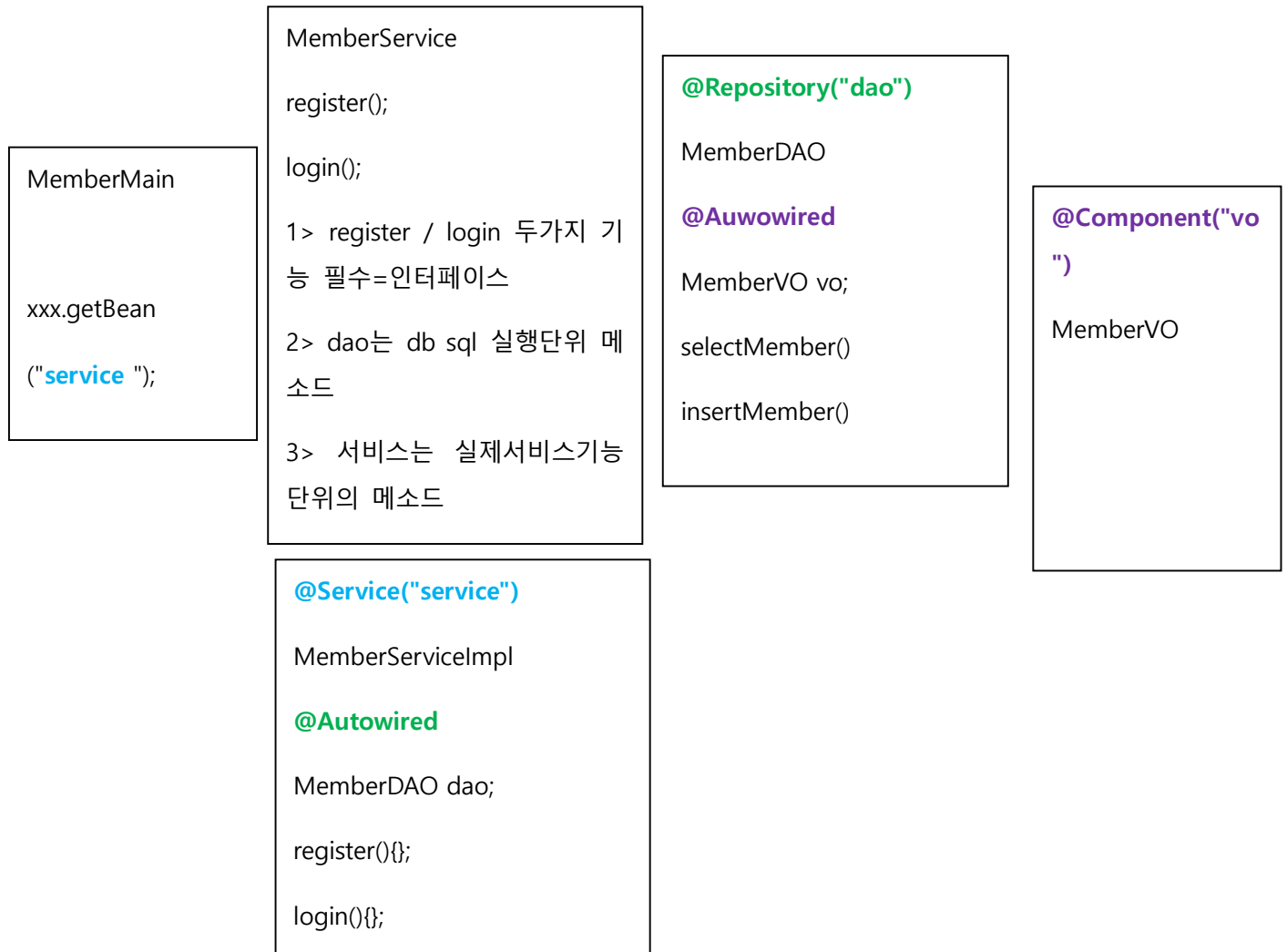
19장 ioc di xml방식	26장 annotation 방식
<bean id="a" class="ss.A"	<b>@Component("a")</b> class A {}
<bean id="a" class="ss.xxxServiceImpl"	<b>@Service("a")</b> class xxxServiceImpl implements xxxService {}
<bean id="a" class="ss.ADAO"	<b>@Repository("a")</b> class ADAO { <b>@Autowired</b> AVO vo; } <b>@Component("vo")</b> class AVO{}  <b>@Repository</b> class BDAO {}
	<b>@Autowired</b>
	<b>@Qualifier</b>

@WebServlet("/a") --> WebServlet annotation --> 클래스 선언부

web.xml <servlet.....

http://ip:port/context/a

## - xml + annotation



19 di ioc	필수
20.aop(aspect oriented programming)	x
21. mvc	가장 많이 활용 spring + web
22, 23 spring jdbc	

23 mybatis java.sql jdbc	
24 스프링과 mybatis	
26 annotation	일부 학습 + ....
27 sts 스프링 - sts설치설정파일.pdf	
28, 29 spring web 고급기술	ajax json rest
30 31	spring 게시판, 쇼핑몰
32	spring boot