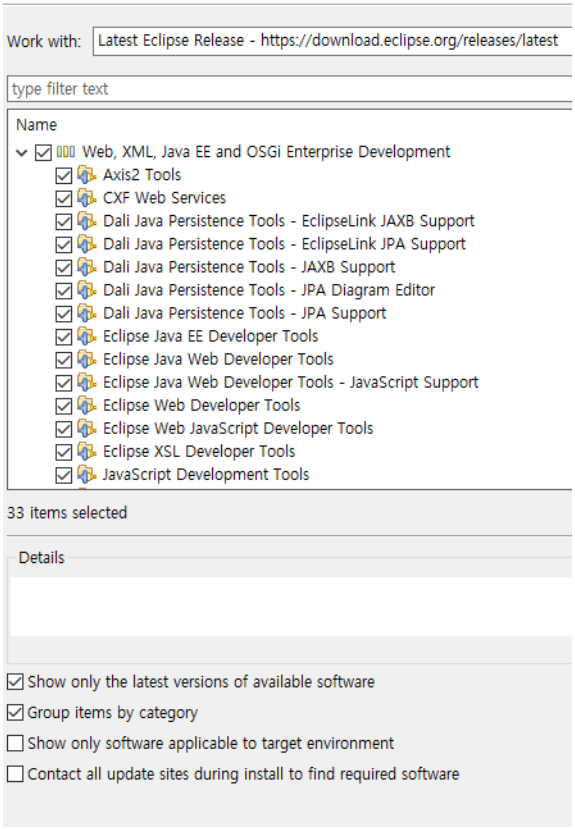
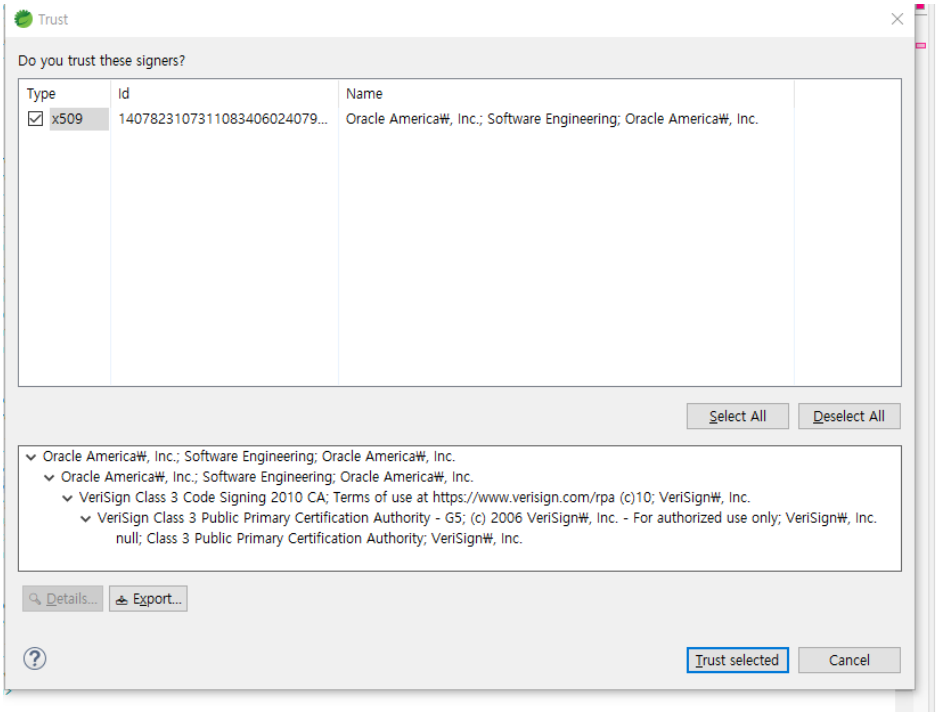
- spring framework

이클립스+spring 사용 추가

sts4 = 이클립스+spring (새로운 버전)툴+ 추가 설치(이전 버전)
1> jdk 11 설치
2> srpin...ini 파일 수정 – 저장
-vm
c:₩kdt-venture₩jdk11₩bin₩javaw.exe
3> spring...exe 실행
4> help-eclipse market...
sts3 add on ... 설치-restart
5> file-new-other....-spring 보이는지 확인
6> sts4 = 이클립스 dynamic web project(servlet, jsp, html, js, css+tomcat server) 기능 사용 화면 제공 툴 추가 설치



7> restart한후에
file-new-other...-server, web 항목 보이는지 확인
8> tomcat server 연동
9>브라우저에서 실행



spring framework- DB연동, WEB기능, di,aop,,,,통합 기능 프레임워크

framework – 어플리케이션 규칙 + 연결 고리 + 규칙내부 자유롭게 작성

일정 틀 내부 프로그램 작성 스프링 연동

스프링 규칙 알아보자

SPRING IOC

SPRING DI

SPRING MVC

SPRING JDBC

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){}
}
```

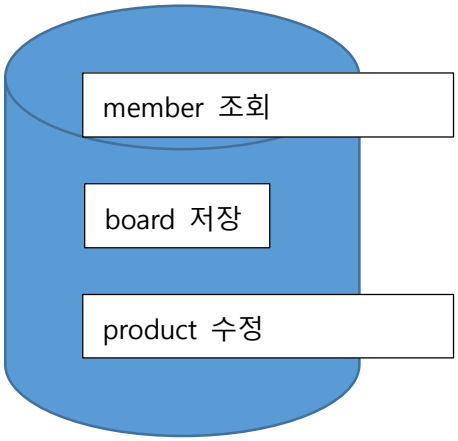
- framework = 틀 , 뼈대

1>전체 완성 프로그램 = 이미 완성하여 제공하는 반제품 + 나머지 반 구현 집중 => 조립

2> 게시판, 회원관리 공통 반복 코드 뼈대 제공

3> 나머지 구현

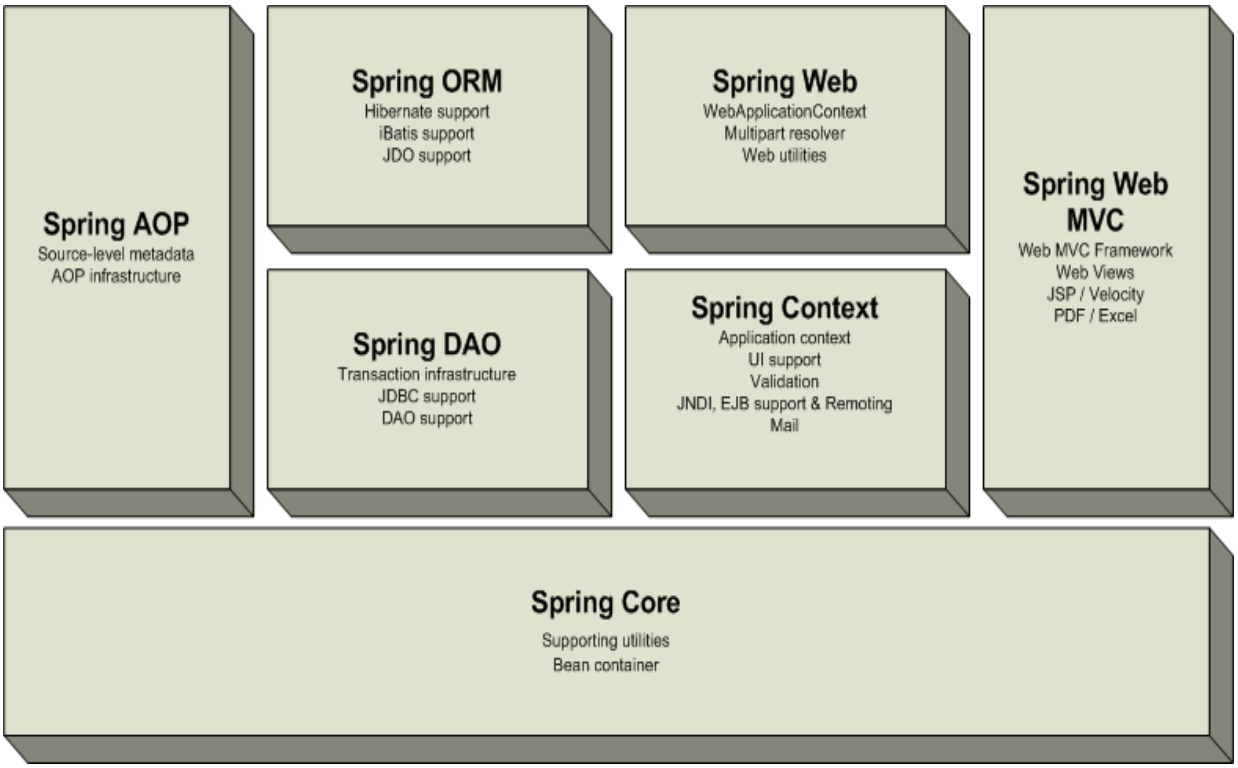
```
try{

class.forName("");

DriverManager.getConneciton

("jdbc:mysql:127.0.0.1:3306....", "", "")

// 변경될부분

con.close()

}
```



- 자바 framework

spring -> 통합 기능 제공 프레임워크

mybatis-> jdbc 개선 제공 프레임워크



19장 스프링 의존성 주입과 제어 역전 기능

스프링 의존성 주입 – spring DEPENDENCY INJECTION(DI)

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

1> 스프링 미사용 구현

```
interface TV   {
powerOn();
poweOff();
soundUp();
soundDown(); }
```

```
class TVFactory{
public TV getTV(String name){
 samsung--> return samsungtv()
lg--> return   lgtv()
}
```

```
class TVUser
TVFactory f = new TVFactory();
TV tv = f.getTV(args[0]);
tv.powerOn()
....
```

```
class 삼성tv implements TV
```

```
class LGtv implements TV
```

1> main TV 교체하면 수정 코드 많다- 줄이자

2> 인터페이스 공통 메소드 선언. 상속 TV 들 정의하자

3> SAMSUNGTV, LGTV들은 TV 인터페이스 메소드 포함

4> TV T1 = new SamsungTV()

TV T2 = new LgTV()

5> main메소드에서 생성 tv들 정하는 것이 아니고 main 외부(TVFactory)에서 정한 객체 전달받자.

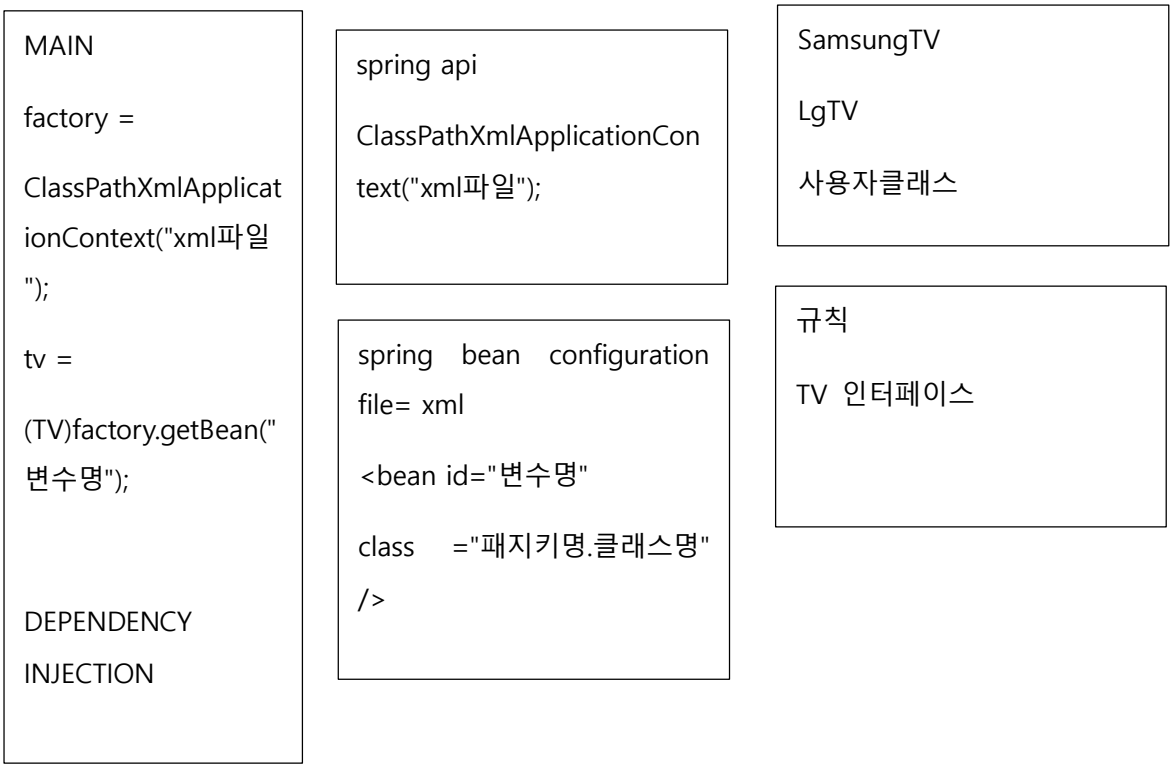
```
MAIN
tv = new Factory().getTV("");
스프링 프로그램
--> 직접 객체 생성 하지 않고
DEPENDENCY INJECTION
```

```
TVFactory
=>스프링 사용자클래스와
스프링 API 연결고리
```

```
SamsungTV
LgTV
사용자클래스
```

```
규칙
TV 인터페이스
```

2> 스프링 사용 구현



MAIN 객체 직접 생성 – NEW xxxxx();
-->객체 생성 권한 main 있다
--> control main 에게 있다

MAIN–스프링 도움받아서 스프링 제공 객체들만 사용
--> 객체 생성 권한 스프링(xml + A....) 있다
--> main 객체 사용 + 스프링에게 객체 생성, 관리 control 있다
--> inversion of control(IOC 제어역전)
--> DEPENDENCY INJECTION (DI)

- 스프링 특징
1> 사용자가 정의한 클래스 객체를 직접 MAIN에서 생성하지 않고 스프링 API를 통하여 전달받는다--> INVERSION OF CONTROL(IOC)
2> 전달받는 객체는 자바의 모든 객체일 수 있다.
POJO 클래스 – PLAIN OLD JAVA OBJECT - 스프링 제어 생성 BEAN 이다.
3> 스프링 관리 클래스는 이전의 non-spring 환경에서 사용했던 클래스 재사용
4> setDto(memberDTO d){ d전달객체(dto1, dto2) }
다른 객체 전달시는 setter 메소드 통해 전달받는다
==> DEPENDENCY INJECTION
4-1. SETTER INJECTION - SETTER메소드 통해 매개변수 타입

4-2. CONSTRUCTOR INJECTION 생성자 매개변수 타입

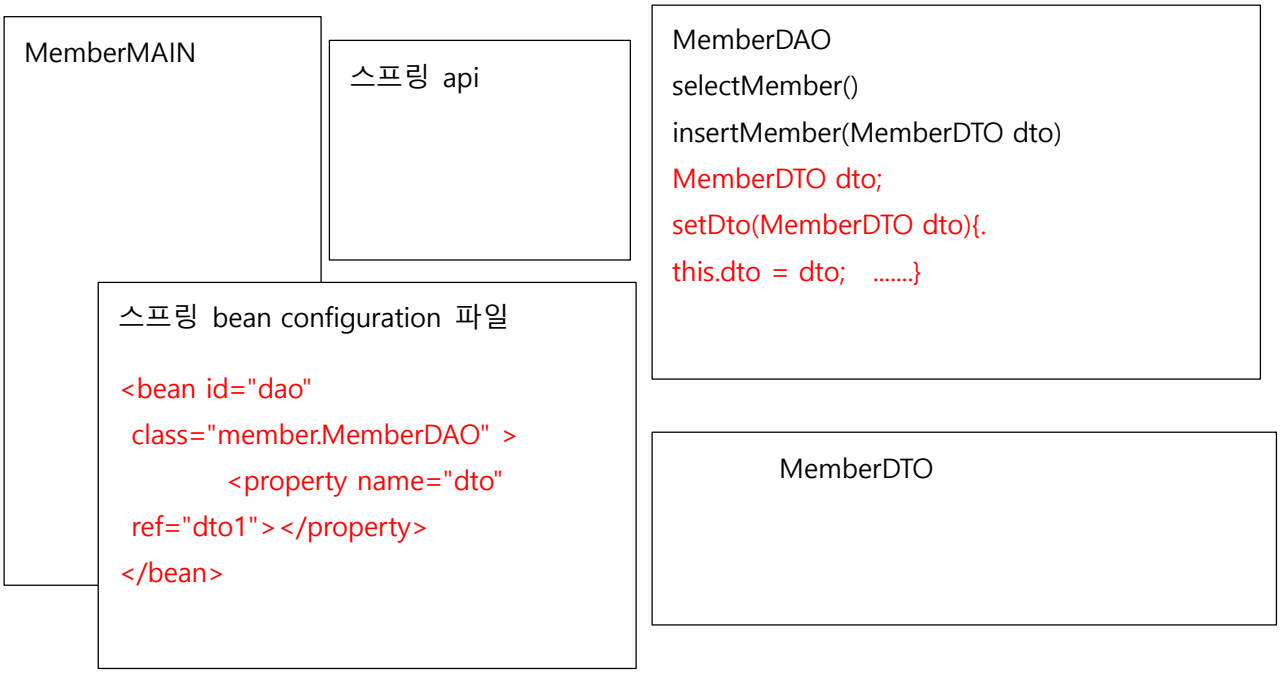
5> 스프링은 자바객체, pojo 클래스, 스프링 bean --> <bean태그정의>

생성-소멸 관리 – 스프링은 1개 bean 생성-singleton 타입

singleton- 같은 타입 객체 1개 생성 공유(기본)

prototype- 같은 타입 객체 여러개 생성

<bean scope=""


MemberMAIN
스프링 api
MemberDAO
selectMember()
insertMember(MemberDTO dto)
MemberDTO dto;
setDto(MemberDTO dto){.
this.dto = dto; .....}
스프링 bean configuration 파일
<bean id="dao"
class="member.MemberDAO" >
<property name="dto"
ref="dto1"></property>
</bean>
MemberDTO

-spring + non-spring 자바 프로그램 템플릿 (=자바 디자인 패턴)

DTO=VO=DO --> 여러 개의 데이터 저장 / 추출 역할 객체
DB –중간 임시 저장- 자바클래스
DAO= --> DB 접근- 조회 , 저장, 수정, 삭제 역할 객체
JDBC 문장
메소드 1개 – sql 1개 실행 단위

SERVICE --> 사용자 요청 기능 1개 단위

class MemberService{
? regiterMember(.....){
1. 회원가입정보가 이미 데이터베이스에 존재 여부 - select
2. 가입 - insert / 중복
}

MemberMain

```
MemberService ser =
new MemberService();
ser.registerMember();
```

MemberService

```
MemberDAO dao=
new MemberDAO();
regiterMember(){
boolean f =
dao.selectMember();
if(f==true){중복}
else
{dao.insertMember()}
}
```

MemberDAO –sql 1개 단위

```
selectMember()
insertMember()
```

MemberDTO

Main – service – dao – dto
servlet+jsp– service – dao - dto
Main – (spring api + xml) -service – dao – dto