1장. 프로그램 설치 및 셋팅

1. 설치 파일

① JDK
   => http://www.oracle.com
   => jdk-8u101-windows-x64.exe
② sts (spring-tool-suite)
   => http://spring.io/
   => spring-tool-suite-3.8.3.RELEASE-e4.6.2-win32-x86_64.zip
③ apache-tomcat-8.0
   => http://tomcat.apache.org
   => apache-tomcat-8.0.36-windows-x64.zip
④ OracleXE 11g
   => https://www.oracle.com
   => OracleXE112_Win64.zip
⑤ sqldeveloper
   => https://www.oracle.com
   => sqldeveloper-4.1.3.20.78-x64.zip

2. 파일 설치 및 설정

2-1. JDK 설치

① http://www.oracle.com

- jdk-8u101-windows-x64.exe

② JDK 설치

C:\C:\Program Files\Java\jdk1.8.0_144 <---jdk1.8.0_144는 버전에 따라 틀림

C:\Program Files\Java\jre1.8.0_144

③ 제어판 -> 시스템 -> 고급 시스템 설정 -> 환경변수 -> 시스템변수

- 새로만들기

변수이름 : JAVA_HOME

변수값 : C:\Program Files\Java\jdk1.8.0_144

<-- (jdk1.8.0_144 은 버전따라 틀림,때문에 탐색창에서 복사해서 붙이는것이 좋음)

- Path 를 찾아 추가할것 :

변수이름 : Path

변수값 : ;%JAVA_HOME%\bin

<--- 맨뒷줄에 추가할것,나머지 앞의 내용은 지우면 안됨

④ Dos창(테스트하기) : 시작 -> cmd

> javac -version <-- 버전확인

2-2. OracleXE112_Win64(오라클) 설치

① https://www.oracle.com

- OracleXE112_Win64.zip

② 설치시 패스워드는 oracle로 할것

③ 기본포트가 8080이므로 톰켓과 충돌나기 때문에 9000으로 변경

- 포트변경

시작 -> cmd 창에서 -> sqlplus system/oracle

sql> select dbms_xdb.getHttpport() from dual;

--> 현재사용중인 port확인 이때 8080이라고 나옴

sql> exec dbms_xdb.sethttpport(9000);

--> port를 9000으로 변경

sql> select dbms_xdb.getHttpport() from dual;

--> 현재사용중인 port확인 이때 9000이라고 나옴

2-3. sqldeveloper 설치

① https://www.oracle.com

- sqldeveloper-4.1.3.20.78-x64.zip

② sqldeveloper 설치

- 원하는 폴더에 압축풀기

- 경로 :

D:/Spring/sqldeveloper

2-4. sts 설치

① http://spring.io/

- spring-tool-suite-3.9.1.RELEASE-e4.7.1a-win32-x86_64.zip

② sts 설치

- 원하는 폴더에 압축풀기
- 경로 :

  D:/spring/sts-bundle

2-5. 웹서버 설치

① http://tomcat.apache.org

- apache-tomcat-8.0.45-windows-x64.zip

② apache-tomcat 설치

- 원하는 폴더에 압축풀기
- 경로 :

  D:/Spring/apache-tomcat-8.0.45

2장. 스프링 시작하기

1. 스프링

- 스프링은 struts와 마찬가지로 오픈소스이다.
- 스프링을 사용하는 이유는 "필요한 인스턴스를 스프링에서 미리 생성해준다" 라는 장점을 얻을 수 있기 때문이다.
- 필요한 인스턴스를 생성하는 방법은 new를 사용하는 방법도 있지만, 디자인패턴인 Factory패턴이나 JNDI(Java Naming and Directory Interface)를 이용하여 생성할 수도 있다

(1) 스프링의 특징

① 어플리케이션 프레임워크로 불리며, 웹 어플리케이션은 물론, 콘솔 어플리케이션이나 스윙과 같은 GUI 어플리케이션 등 어떤 어플리케이션에도 적용 가능한 프레임워크이다.

② 스프링은 EJB와 같이 복잡한 순서를 거치지 않아도 간단하게 이용할 수 있기 때문에 '경량(Lightweight) 컨테이너'라고도 부른다.

③ 스프링은
- 의존주입(DI : Dependency Injection) 지원
- AOP(Aspect Oriented Programming) 지원
- MVC 웹 프레임워크 제공
- JDBC 연동 기술 및 선언적 트랜잭션 처리 등 DB 연동 지원

등이 가장 중점적인 기술로 사용되지만, 이외에도 여러가지 기능을 제공하고 있다.

④ 스프링은 자주변경이 되거나 컴포넌트의 재활용이 높은 유연한 어플리케이션에 작성할수 있다는 장점을 가지고 있다

(2) 스프링의 7가지 모듈

① Spring Core (core container)
- Spring 프레임워크의 핵심 기능을 제공한다.
- 코어 컨테이너의 주요 컴포넌트는 Bean-Factory(Factory 패턴의 구현)이다.
- BeanFactory는 Inversion of Control (IoC) 패턴을 사용하여 애플리케이션의 설정/의존성 스팩을 실제 애플리케이션 코드에서 분리시킨다.
- Spring Core 바로위에 있으면서 Spring Core에서 지원하는 기능 외에 추가적인 기능들과 좀 더 쉬운 개발이 가능하도록 지원하고 있다.
- 유저 인터페이스 및 타당성 검증이라는 어플리케이션의 기반 성능, JNDI 및 EJB의 지원, 메일 송.수신 기능 등을 지원한다.

② Spring Context
- Spring을 컨테이너로 만든 것이 핵심 모듈의 BeanFactory라면, Spring을 프레임워크로 만든 것은 컨텍스트 모듈이다.
- 이 모듈은 국제화된 메시지, 애플리케이션 생명주기 이벤트, 유효성 검증 등을 지원함으로써 BeanFactory의 개념을 확장한다.
- 이 모듈은 이메일, JNDI 접근, EJB 연계, 리모팅, 스케쥴링 등과 같은 다수의 엔터프라이즈 서비스를 추가로 제공한다
- 템플릿 프레임워크와의 통합도 지원한다.

③ Spring AOP 모듈(Spring AOP)
- 설정 관리 기능을 통해 aspect 지향 프로그래밍 기능을 Spring 프레임워크와 직접 통합시킨다.
  따라서 Spring 프레임워크에서 관리되는 모든 객체에서 AOP가 가능하다.
- Spring AOP 모듈은 Spring 기반 애플리케이션에서 객체에 트랜잭션 관리 서비스를 제공한다.
- Spring AOP에서는 EJB 컴포넌트에 의존하지 않고도 선언적 트랜잭션 관리를 애플리케이션과
  결합할 수 있다.

④ Spring DAO(Data Access Object)
- Spring JDBC DAO 추상 레이어는 다른 데이터베이스 벤더들의 예외 핸들링과 오류 메시지를
  관리하는 중요한 예외 계층을 제공한다.
  이 예외 계층은 오류 핸들링을 간소화하고, 예외 코드의 양도 줄여준다.
- Spring DAO의 JDBC 예외는 일반 DAO 예외 계층에 순응한다.
- JDBC에 의한 데이터베이스 액세스를 지원하고 트렌젝션 관리의 기반이 된다

⑤ Spring ORM
- 프레임워크는 여러 ORM(Object/Relation Mapping) 프레임워크에 플러그인 되어,
  Object Relational 툴 (JDO, Hibernate, iBatis SQL Map)을 제공한다.
- 이 모든 것은 Spring의 일반 트랜잭션과 DAO 예외 계층에 순응한다.

⑥ Spring Web module
- 웹 컨텍스트 모듈은 애플리케이션 컨텍스트 모듈의 상단에 구현되어, 웹 기반 애플리케이션에
  컨텍스트를 제공한다.
- Spring 프레임워크는 Jakarta Struts와의 통합을 지원한다.
- 웹 모듈은 다중 요청을 핸들링하고, 요청 매개변수를 도메인 객체로 바인딩하는 작업을 수월하게 한다.

⑦ Spring MVC framework
- MVC 프레임워크는 완전한 기능을 갖춘 MVC 구현이다.
- MVC 프레임워크는 전략 인터페이스를 통해 설정할 수 있으며, JSP, Velocity, Tiles, iText, POI 같은
  다양한 뷰 기술을 허용한다.

(3) 스프링을 이용한 자바 프로젝트의 진행과정
① 스프링 프로젝트 생성
② 클래스 파일 추가
③ 스프링에 맞는 자바코드와 설정 파일 작성
④ 실행

(4) 메이븐 프로젝트
1) 프로젝트 폴더 구조

```
workspace
    => sp4-chap2_maven
        => src
            => main
                => java
                => resources
```

2) 메이븐 프로젝트 설정파일
  => pom.xml 파일
  => 위치 : workspace/sp4-chap2_maven/pom.xml

(5) 용어

1) 아티팩트 파일 (artifact file)

- 메이븐은 한 개의 모듈을 아티팩트라는 단위로 관리한다.
- 보통 스프링은 결과물(Artifact)을 다음 네 곳에 배포한다.

▪커뮤니티 다운로드 사이트
http://www.springsource.org/downloads/community 다운로드받기 쉽도록 zip 파일로 묶인 모든 스프링 jar파일이 있다.
3.0 버전부터 이곳의 jar파일명은 org.springframework.*-<version>.jar 의 형식을 따른다.

▪메이븐 센트럴(Maven Central)
메이븐 센트럴은 메이븐의 기본 저장소이므로 사용하기 위해서 특별한 설정을 할 필요는 없다.
스프링이 의존하는 공통 라이브러리 중 다수는 메이븐 센트럴에서 이용할 수 있고 스프링 커뮤니티의 많은 부분은 의존성 관리에 메이븐을 사용하므로 편리하다.
메이븐 센트럴의 jar 파일명은 spring-*-<version>.jar의 형식을 따르고 메이븐 groupId는 org.springframework이다.

▪SpringSource에서 운영하는 Enterprise Bundle Repository (EBR)
EBR는 스프링에 통합하는 모든 라이브러리를 제공한다. 모든 스프링 jar와 의존성 라이브러리를 포함해서 그 외 스프링으로 어플리케이션을 개발할 때 사용하는 공통 라이브러리에 대한 메이븐 저장소와 아이비(Iby) 저장소를 모두 이용할 수 있다.
릴리즈 버전과 마일스톤을 포함해서 개발 스냅샷까지 이곳에 배포된다.
jar 파일명은 커뮤니티 다운로드와 같은 형식(org.springframework.*-<version>.jar)을 사용하고 외부 라이브러리(스프링소스가 만들지 않은)는 com.springsource 접두사가 있는 긴 형식이다.

▪개발 스냅샵과 마일스톤 릴리즈를 위해 아마존 S2에서 운영되는 공개 메이븐 저장소를 사용한다. (최종 릴리즈의 복사본도 배포된다.)
jar 파일명은 메이븐 센트럴과 같은 형식이므로 메이븐 센트럴에 배포된 다른 라이브러리와 함께 스프링의 개발 버전을 사용할 때 유용하다.

2) 메이븐 원격 리포지토리 (repository)

- 메이븐은 컴파일하거나 코드를 실행하기 위해 <dependency>로 설정한 artifact 파일을 메이븐 로컬 리포지토리에 같은 이름의 파일이 있는지 검사하고 존재하면 해당 파일을 사용하지만, 존재하지 않는다면 원격 중앙 리포지토리로부터 해당 파일을 다운 받아 로컬 리포지토리에 복사한 뒤 그 파일을 사용한다.
- pom.xml 파일에 지정한 아티팩트 파일을 메이븐 중앙 리포지토리에서 다운로드 받는다.
- 스프링을 비롯한 지비 개발에서 시용되는 많은 오픈소스 프로젝트들이 이미 메이븐 중앙 리포지토리에 아티팩트 파일을 등록하고 있다.
- <dependency> 설정만 알맞게 추가해주면 필요한 jar 파일을 손쉽게 메이븐 프로젝트에 추가할 수 있다.

3) 지역 리포지토리

- 스프링 프레임워크에 필요한 jar 파일들을 프로젝트 폴더에 저장해서 사용

2. 스프링 세팅

2-1. 스프링 셋팅방법-1
- http://spring.io 에서 다운로드
- spring-tool-suite-3.9.1.RELEASE-e4.7.1a-win32-x86_64.zip 다운로드 저장

2-2. 스프링 셋팅방법-2
(1) www.eclipse.org 에서 Eclipse를 다운로드한후 설치
(2) 이클립스 실행한 후 Help -> Eclipse Marketplace -> Search탭에서 Find : STS -> go
-> Spring Tool Suite(STS) for Eclipse 3.9.1.RELEASE -> install

2-3. sts 설치 및 실행
- spring-tool-suite-3.9.1.RELEASE-e4.7.1a-win32-x86_64.zip 파일의 압축을 푼다.
- sts-bundle/sts-3.9.1.RELEASE/STS.exe를 실행한다.

2-4. springframework 라이브러리
- spring-aop-4.3.6.RELEASE.jar
- spring-beans-4.3.6.RELEASE.jar
- spring-context-4.3.6.RELEASE.jar
- spring-core-4.3.6.RELEASE.jar
- spring-expression-4.3.6.RELEASE.jar

2-5. logging 라이브러리
- commons-logging-1.2.jar

3장. spring_step01

예제1. 결합도가 높은 프로그램

Project Name : step01 (Simple Java)

package Name : sample1
class Name : src/sample1/MessageBeanEn.java
src/sample1/MessageBeanKr.java
src/sample1/HelloSpring.java (main)

<작업 순서>

1. 프로젝트 만들기
2. package와 class 파일 추가하기
3. 자바코드 작성
4. 실행

<실행 결과>

Hello,Spring!!

예제2. 다형성을 이용해서 결합도를 낮춤

Project Name : step01 (Simple Java)
package Name : sample2
interface Name : src/sample2/MessageBean.java (interface)
class Name : src/sample2/MessageBeanEn.java
src/sample2/MessageBeanKr.java
src/sample2/HelloSpring.java (main)

<작업 순서>

1. package와 class 파일 추가하기
2. 자바코드 작성
3. 실행

<실행 결과>

Hello, Spring!!

예제3. Factory 패턴을 이용해서 결합도를 낮춤

Project Name : step01 (Simple Java)
package Name : sample3
interface Name : src/sample3/MessageBean.java (interface)
class Name : src/sample3/MessageBeanEn.java
src/sample3/MessageBeanKr.java
src/sample3/BeanFactory.java
src/sample3/HelloSpring.java (main)

<작업 순서>

1. package와 class 파일 추가하기
2. 자바코드 작성
3. 실행

<실행 결과>

Hello, Spring !!

예제4. bean.xml을 이용해서 결합도를 낮춤 - Spring 이용

- 스프링 라이브러리를 등록해서 사용

Project Name : step01 (Simple Java)

Folder : step01/libs
Library File : libs/spring-aop-4.3.6.RELEASE.jar
libs/spring-beans-4.3.6.RELEASE.jar
libs/spring-context-4.3.6.RELEASE.jar
libs/spring-core-4.3.6.RELEASE.jar
libs/spring-expression-4.3.6.RELEASE.jar
libs/commons-logging-1.2.jar

package Name : sample4
class Name : src/sample4/MessageBean.java (interface)
src/sample4/MessageBeanEn.java
src/sample4/MessageBeanKr.java
src/sample4/HelloSpring.java (main)
XML File : src/sample4/bean.xml

<작업 순서>

1. 폴더 만들기
2. 라이브러리 저장 및 등록
3. package와 class 파일 추가하기
4. XML 파일 만들기
5. 스프링에 맞는 자바코드와 설정 파일 작성
6. 실행

<실행 결과>

Hello, Spring!!

4장. spring_step02_maven

예제1. maven을 이용 - Simple Spring Maven

Project Name : step02

Package : sample4
interface Name : src/main/java/sample4/MessageBean.java
class Name : src/main/java/sample4/MessageBeanEn.java
src/main/java/sample4/MessageBeanKr.java
src/main/java/sample4/HelloSpring.java (main)
XML File : src/main/resources/bean.xml

<작업 순서>

1. 프로젝트 만들기
2. JRE System Library 버전 변경하기
3. pom.xml 파일 수정하기
4. Class 파일 추가하기
5. XML 파일 만들기

<실행 결과>

Hello, Spring!!