

Spring Framework 테스트 총 20문항

소속 :

이름:

시험일시: 2018 년 월 일

1. Servlet이나 JSP를 여러 번 호출해도 한번만 객체를 생성해서 사용할 수 있도록 하는 패턴은?
 - ① 팩토리
 - ② 싱글턴
 - ③ MVC
 - ④ Iterator
 - ⑤ Dispatcher
2. Spring MVC에서 컨텍스트가 실행되기전 가장 먼저 스프링에 필요한 환경변수를 읽어들이는 것은?
 - ① ContextLoaderListener
 - ② ApplicationContextListener
 - ③ LoaderContextListener
 - ④ ContextLoaderFilter
 - ⑤ Dispatcher
3. Spring MVC에서 원하는 컨트롤러로 보내는 것은?
 - ① ServletDispatcher
 - ② DispatcherServlet
 - ③ ContextListener
 - ④ AccessLoggerFilter
 - ⑤ CharacterEncodingFilter
4. Spring MVC에서 이동시 가로채서 원하는 인코딩으로 바꿔주는 것은?
 - ① ServletDispatcher
 - ② DispatcherServlet
 - ③ ContextListener
 - ④ AccessLoggerFilter
 - ⑤ CharacterEncodingFilter
5. OOP의 5대 원리(Principle)가 아닌 것은?
 - ① SIP or ISP (interface segregation principle)
 - ② OCP
 - ③ LSP
 - ④ DIP

- ⑤ IOC
- ⑥ SRP

6. 스프링 기본 원리가 아닌 것은?

- ① DI
- ② OCP
- ③ AOP
- ④ POJO
- ⑤ IOC

7. Spring MVC에서 컨트롤러에서 화면(VIEW)으로 이동할 때 원하는 화면으로 갈수 있도록 해주는 것은?

- ① Servlet
- ② Handler
- ③ Listener
- ④ Filter
- ⑤ Resolver

8. Spring MVC에서 Model이나 ModelAndView에 저장하는 객체는 기본으로 어느 스코프에 저장되는가?

- ① Page
- ② Request
- ③ Session
- ④ Application
- ⑤ Config

9. JSP에서 스크립트 엘리먼트를 사용을 줄이기 위해서 사용하는 것이 아닌 것은?

- ① ActionTag
- ② EL
- ③ JSTL
- ④ CustomTag
- ⑤ NameSpace

10. XML을 사용할 때 이름간에 충돌을 방지하기 위해서 사용하는 것은?

- ① XSLT
- ② XPATH

- ③ XSI
- ④ NameSpace

11. Spring MVC에서 XML환경 설정을 할 때 <bean> 엘리먼트의 p 어트리뷰트를 <bean> 엘리먼트의 자식 엘리먼트로 선언할 때 사용하는 것은?

- ① page
- ② pid
- ③ property
- ④ attribute
- ⑤ field

12. 다음중 다른 레이어에 속하는 것은?

- ① Tiles
- ② Sitemesh
- ③ JSP
- ④ MyBatis
- ⑤ Velocity

13. ORM 계열의 프레임워크는 EJB의 어느 Bean과 기능이 같은 것인가?

- ① Session
- ② Entity
- ③ MDB

14. 객체가 가질 수 없었던 DB의 특성을 갖게 하는 ORM의 핵심이 되는 단어는 무엇인가?

- ① 영속성
- ② 원자성
- ③ 독립성
- ④ 고립성
- ⑤ 중복성

15. Spring MVC에서 컨트롤러, 서비스, CRUD 전문객체에 붙이는 어노테이션을 순서대로 쓰시오.

(예 ①②③)

- ① @Controller
- ② @Autowired
- ③ @RequestMapping
- ④ @PathVariable

- ⑤ @Repository
- ⑥ @Service
- ⑦ @Override
- ⑧ @Transactional

16. Spring MVC 3.X에서 모든 등록된(XML 등록, 자동)된 객체를 사용할 때 DI하기 위해서 사용하는 어노테이션은?

- ① @Controller
- ② @Autowired
- ③ @RequestMapping
- ④ @PathVariable
- ⑤ @Repository
- ⑥ @Service
- ⑦ @Override
- ⑧ @Transactional

17. Spring MVC에서 서비스 메서드에 붙여서 여러 행위(메서드)를 묶어 한 행위처럼 만드는 어노테이션은?

- ① @Controller
- ② @Autowired
- ③ @RequestMapping
- ④ @PathVariable
- ⑤ @Repository
- ⑥ @Service
- ⑦ @ResponseBody
- ⑧ @Transactional

18. Spring MVC에서 XML이나 JSON형태로 결과를 출력하기 위해서 컨트롤러 메서드에 붙이는 어노테이션은(AJAX를 사용할 때 많이 사용한다)?

- ① @Controller
- ② @Autowired
- ③ @RequestMapping
- ④ @PathVariable
- ⑤ @Repository
- ⑥ @Service
- ⑦ @ResponseBody

Ⓢ @Transactional

19. Spring MVC에서 중요 비즈니스 로직과 보조 로직으로 나누어 작성하지만, 실행할 때는 비즈니스 로직과 보조 로직이 섞여서 실행되도록 하는 원리는 다음 중 무엇인가?

- ① DI
- ② OCP
- ③ AOP
- ④ LSP
- ⑤ IOC

답 안

- 1. 2
- 2. 1
- 3. 2
- 4. 5
- 5. 5
- 6. 4
- 7. 5
- 8. 2
- 9. 5
- 10. 4
- 11. 3
- 12. 4
- 13. 2
- 14. 1
- 15. 165
- 16. 2

17. 8

18. 7

19. 3

20. 다음과 같은 어노테이션 대신 사용할 수 있는 어노테이션은?

@Controller

@Repository

@Service

21. 어노테이션을 사용할 수 있도록 환경설정에 꼭 있어야 할 엘리먼트는?

22. com.ssafy.edu 패키지부터 그 아래 패키지의 어노테이션을 자동으로 찾으려고 한다. 환경설정에 꼭 있어야 할 엘리먼트는?

23. MyBatis를 사용하는 이점은 무엇인가? 특징은 무엇인가? 주의할 점은 무엇인가?

24. CRUD를 전문적으로 담당하는 것은 무엇이며, 어떤 어노테이션을 사용하는가?

25. DS 컨트롤러와 직접적으로 연결된 것이 아닌 것은 무엇인가? 또는 DS 컨트롤러가 부르는 것이 아닌 것 ?

1) 서비스 2) 핸들러 3) 리졸버, 4) 다오, 5) 컨트롤러

26. 목록로 시작하여 목록을 출력하는 books.jsp까지 흐름을 적으시오.

27. Restful 서비스 메서드에 대하여 ?

20. @Component

21. <annotation-driven />

22. <context:component-scan base-package="com.ssafy.edu" />

23. 개발자가 직접 개발해야할 DAO에서 JDBC 6단계를 이용하여 직접 구현해야했는데 마이바티스를 사용하면 추상화 프로그래밍을 할 수 있다. 다시 말해서 마이바티스를 사용만 하면 JDBC 단계를 몰라도 된다. 마이바티스는 데이터 베이스에 관련된 CRUD 작업을 추상화 및 단순화(select, insert, update, delete 만 사용)하여 사용을 편하게 하였다. 객체를 SQL과 매핑하면 객체를 주고 받는 객체 지향 프로그래밍을 할 수 있으면, ORM 매핑으로 객체가 디비 특징인 영속성을 얻는다. 사용은 객체로 주고 받으면 뒤에서 매핑된 SQL이 자동으로 실행된다. 마이바티스는 어플리케이션, 웹 등 모든 곳에서 사용할 수 있다. 예로 SpringMVC와 SpringBoot, Restful 서비스를 만들 때 CRUD 관련은 모두 DAO가 마이바티스를 이용하여 사용했다. 마이바티스는 동적 쿼리를 실행할 수 있다. 서치 기능에서 제목으로, 이름으로, 내용으로 찾기를 할 때 <if test 를 이용하여 여러 쿼리를 조립할 수 있다. XML 환경이므로 >,<을 >, <으로 치환하여 사용한다. Book..xml형식으로 파일로 저장할수도 자바로 설정할 수도 있다.

```
<select id="getBookPageList"

parameterType="com.ssafy.edu.model.BookRequest"

resultType="com.ssafy.edu.model.Book">
```

```

SELECT  bs.ISBN, bs.TITLE, bs.CATALOGUE, bs.AUTHOR  from

(select * from chi_book where 1=1

<if test="s_category != " and s_category != null and s_keyword != " and s_keyword != null">

<if test="s_category == 'title'"> AND TITLE like concat('%',{s_keyword},'%')</if>

<if test="s_category == 'publisher'"> AND publisher like concat('%',{s_keyword},'%') </if>

</if>

ORDER BY ISBN ) bs

limit   #{start} ,   #{recordCountPerPage}

</select>

```

```

<select id="getBookGT"

resultType="com.ssafy.edu.model.Book">

        select * from chi_book where price > 3000

</select>

```

```

<select id="getBookGT"

resultType="com.ssafy.edu.model.Book">

<![CDATA[

        select * from chi_book where price > 3000

]]>

</select>

```


24. DAO, @Repository

25. DAO는 서비스와 DI(의존)

26. books.do ->

@Controller -> HandlerMappfer -> @RequestMapping

```
@RequestMapping(value = "books.do", method = RequestMethod.GET)

public String books(Model model) {

    logger.debug("Welcome MainController books! " + new Date());

    model.addAttribute("books", bookservice.getBookList());

    return "books";

} //
```

➔ Resolver books.jsp

```
<c:if test='${empty books}'>

<tr>

<td colspan="5">목록이 없습니다.</td>

</tr>

</c:if>

<c:forEach items="${books}" var="m" varStatus="vs" >

<tr>

<td onclick="bookdetail('${m.isbn }')">[${vs.count}]${m.isbn }</td>

<td onclick="bookdetail('${m.isbn }')">${m.title }</td>
```

```
<td> ${m.catalogue } </td>

<td> ${m.author } </td>

<td> <button onclick="bookdelete('${m.isbn }')"> 삭제 </button> </td>

</tr>

</c:forEach>
```

27. Get -> select, Post -> insert, Put , Patch -> update (Put 전체업데이트, path 부분업데이트)