**웹기반정보시스템및실험 (2 분반) 기말고사 시험지**

**2017. 06. 15**

**[A4 2장(앞뒤로 허용] [단, 자필로 쓴 것만 허용함]**

**※ 유형 1. 객관식 [2\*9=27, 틀리면 –1 감점]**

**1. 다음 중에서 알맞은 것을 선택하시오.**

**1) 다음 표현 언어(Expression Language)에 대한 설명 중 잘못된 것은 무엇인가?**

①표현 언어는 ‘$’ 문자로 시작한다. ②표현식 member.id와 member[“id”]는 동일하다.

③표현식에는 boolean, null과 같은 상수 값은 올 수 없다.

④표현 언어는 빈즈 객체에 보다 쉽게 접근할 수 있으며, 표현식의 사용을 줄여준다.

**2) 다음 JSP 1.2 커스텀 태그를 구현하기 위한 구성 요소 중 불필요한 것은 무엇인가?**

①태그 파일 ②태그 라이브러리 기술자 ③태그 핸들러 클래스 ④JSP 파일

**3) JSTL 라이브러리별 URI와 prefix가 바르게 연결된 것은 무엇인가? (‘http://’는 생략)**

①코어 – java.sun.com/jsp/jstl/cores – c ②XML – java.sun.com/jsp/jstl/xml – xml

③데이터베이스 – java.sun.com/jsp/jstl/database – db ④함수 – java.sun.com/jsp/jstl/functions – fn

**4) 다음 커넥션 풀에 대한 설명 중 틀린 것은 무엇인가?**

①웹 서버가 시작될 때 일정 수의 커넥션을 미리 생성한다. ②요청에 따라 생성된 커넥션 객체를 전달한다.

③일정 이상의 커넥션이 사용되면 새로운 커넥션을 생성한다. ④사용하지 않는 커넥션은 종료하고 최대한의 커넥션을 유지한다.

**5) 트랜잭션 처리에 대한 설명 중 틀린 것은 무엇인가?**

①데이터의 무결성을 위하여 DB에서 제공하는 기능 ②데이터베이스에서 일련의 작업을 하나로 묶어 처리하는 기능

③JDBC2.0은 일괄 갱신 기능을 지원 ④JDBC에서는 자동 commit이 기본이지만 오라클은 부하를 줄이기 위하여 수동 commit이 기본

**6) 리스너와 서블릿 필터에 대한 설명 중 틀린 것은 무엇인가?**

①필터는 JSP가 실행되기 전에 내용의 첨삭을 지원 ②ServletFilter 인터페이스를 implements 하여 필터 클래스 구현

③필터는 doFilter 메서드를 구현해야 함 ④하나의 서블릿은 그것에 연관된 여러개의 필터들을 가질 수 있음

|  |
| --- |
| 웹 어플리케이션뿐만 아니라 일반 어플리케이션까지 범용으로 이용할 수 있는 로깅 프레임워크이다. 여러 로깅 API들을 단일화된 접근 방식으로 사용할 수 있도록 해주는 것으로 보통의 경우 내부적으로는 여러 로깅 API를 사용하면서 애플리케이션에서는 이 로깅 프레임워크의 API를 통하여 처리한다. |

**7) 다음 내용이 설명하고 있는 로깅 구현 형태는 무엇인가?**

①Log4j 프레임워크를 이용한 로깅 ②자카르타 Commons 로깅 ③서블릿 컨테이너 제공 ④AOP를 이용한 로깅

**8) JSP 2.0부터 에러처리를 위하여 ErrorData 클래스의 인스턴스가 추가되었다. 다음 중 이 클래스의 속성의 데이터 유형과 설명이 잘못된 것은 무엇인가?**

①requestURI – String – 요청이 실패한 URI ②ServletName – String – 실패한 JSP나 서블릿 이름

③statusValue – String – 실패 상태 값 ④throwable – Throwable – 오류 페이지를 불러들인 예외

|  |
| --- |
| MVC 패턴의 중심이 되는 부분으로, 사용자의 요청에 따라 처리해야 할 대상을 연결하고 처리 결과의 전달 대상을 지정한다. 이것은 직접 구현하거나 구현된 솔루션을 이용할 수 있다. 대표적으로 스트럿츠 프레임워크가 있다. |

**9) 다음 내용이 설명하고 있는 MVC 패턴의 구성 요소는 무엇인가?**

①Controller ②Model ③View ④DAO Class

2. Framework 관점에서 다음의 용어들과 Framework의 차이점을 설명하시오.[20]

① Library vs Framework ② Component vs Framework ③ Design Pattern vs Framework

④ Architecture vs Framework ⑤ Platform vs Framework

3. 다음 각각에 대하여 간략하게 답하시오.[2\*5=10]

(1) 트랜잭션(Transaction)을 예(저축통장)를 들어 설명하시오.

(2) 커넥션 풀에 대하여 설명하고 필요한 이유를 예를 들어 설명하시오.

4. 웹 화면과 프로그램이 복잡해질수록 커스텀 태그 라이브러리를 사용했을 때의 장점 3가지를 적으시오.[9]

5. 각각의 프로그램에 대하여 답안을 작성하시오.[12]

1) 다음의 출력 값을 적으시오. <urltest는 localhost 아래의 폴더임>

|  |
| --- |
| <c:url value="/urltest/choose.jsp" var="target">  <c:param name="sel">a</c:param>  </c:url>  단순출력 : ${target}<BR> |

2) 다음 코드의 출력값은 0522592214 이다. 괄호 안에 알맞은 키워드를 적으시오.

|  |
| --- |
| <c:forTokens items="052-259-2214" ( ① )="-" var="sel">  ( ② )  </c:forTokens> |

3) 다음 스크립트릿을 표현 언어로 수정하시오.

|  |
| --- |
| <form name=form1 method=POST action=ProductSel.jsp>  <jsp:useBean id="product" class="jspbook.ch10.Product" scope="session"/>  <select name="sel">  **<%**  **for(String item : product.getProductList()) {**  **out.println("<option>"+item+"</option>");**  **}**  **%>**  </select>  <input type="submit" value="선택"/>  </form> |

|  |  |
| --- | --- |
| <form name=form1 method=POST action=ProductSel.jsp>  <jsp:useBean id="product" class="jspbook.ch10.Product" scope="session"/>  <select name="sel">     |  | | --- | |  |   </select>  <input type="submit" value="선택"/>  </form> |

6. “제가 사는 곳은 울산.”과 같이 사용할 문구들을 태그파일 기반의 커스텀 태그를 사용하여 처리하려고 한다. 다음 조건에 따라 빈칸에 들어갈 내용을 채우시오[8]

|  |
| --- |
| 태그파일명 : address.tag  속성 : addr  출력 형태 : “안녕하세요. addr 속성값 ” |

[address.tag]

|  |  |  |
| --- | --- | --- |
| <%@ tag body-content="empty" pageEncoding="UTF-8" %>   |  | | --- | | ① |   제가 사는 곳은   |  | | --- | | ② | |

[Hello.jsp]

|  |  |  |
| --- | --- | --- |
| <%@ page contentType="text/html; charset=UTF-8"%>  <%@ taglib   |  | | --- | | ③ |   prefix=“home” %>  <HTML>  <HEAD><TITLE></TITLE></HEAD>  <BODY>  <div align=center>  <H2><   |  | | --- | | ④ |   addr=“울산” />  </div>  </BODY>  </HTML> |

7. jsp에서 빈즈를 선언하고 출력하여 나타내면 다음과 같다. [9]

|  |
| --- |
| <H2>  <jsp:useBean id=“member” class=“mysns.people.Member” />  <jsp:getProperty name=“member” property=“uid” /> |

① 표현식을 사용하여 위의 같은 결과 값을 가지도록 만드시오.

|  |
| --- |
| <H2>  <jsp:useBean id=“member” class=“mysns.people.Member” />  < ① >  </H2> |

② 표현 언어를 사용하여 위의 같은 결과 값을 가지도록 만드시오.

|  |  |
| --- | --- |
| <H2>   |  | | --- | | ② |   </H2> |

③ JSTL <c:out>태그를 사용하여 위와 같은 결과 값을 만드시오.

|  |  |
| --- | --- |
| <H2>   |  | | --- | | ③ |   </H2> |

8. DBCP를 이용한 커넥션 풀을 사용하려면 먼저 톰캣 서버에 DataSource 등록해야 한다. 톰캣 서버 등록은 server.xml이나 context.xml에서 할 수 있다. context.xml 파일이 다음과 같이 설정하고자 한다. <web.xml>을 참고하여 알맞은 내용을 적으시오.[6]

<context.xml>

|  |
| --- |
| <Resource ① ② >  type= ③  driverClassName="com.mysql.jdbc.Driver"  url="jdbc:mysql://127.0.0.1/jspdb"  username="jspbook" password="1234"  maxActive="5" maxIdle="3" maxWait="-1" /> |

<web.xml>

|  |
| --- |
| <resource-ref>  <res-ref-name>jdbc/mysql</res-ref-name>  <res-auth>Container</res-auth>  </resource-ref> |

9. 다음 그림과 같이 출력하도록 알맞은 내용을 적으시오.[12]

application scope에 members, member 객체의 속성(name, email)이 저장되어 있다고 가정한다.

|  |  |
| --- | --- |
| 텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진  자동 생성된 설명 | <c:forEach var="i" items="${members}" begin="5" varStatus="status" end="7">  index: ( ① ) <BR>  count: ( ② ) <BR>  name: ( ③ ) <BR>  email: ( ④ ) <BR>  <HR>  </c:forEach>  <c:forTokens ( ⑤ ) delims="," var="sel">  ( ⑥ ) <BR>  </c:forTokens> |

10. 다음 빈칸을 채우시오 [12]

(DBCP와 트랜잭션) 톰캣에 DataSource를 설정하여 데이터베이스 커넥션 풀을 이용할 때 server.xml 파일에 몇 가지 작업을 해주어야 한다. 그중에서 (①) 태그는 사용할 커넥션 풀 자원을 등록하는 부분으로 JNDI에서 사용할 이름, 데이터 유형, 데이터베이스 접속 계정, 비밀번호 등을 해당하는 속성에 알맞게 설정해주어야 한다. 특히, (②) 속성에는 데이터베이스 최대 연결 개수를 (③) 속성에는 데이터베이스 최대 유휴 연결 개수를 설정해주어야 한다.

트랜잭션은 데이터베이스에서 일련의 작업을 하나로 묶어 처리하는 것을 의미하는데, 데이터베이스에 트랜잭션이 완료되었음을 알리는 명령은 (④)이고, 트랜잭션을 취소하는 명령은 (⑤)이다. JDBC 2.0에서 일괄갱신기능이 추가되었는데 일괄 갱신 작업에 SQL 문을 추가하는 것은 Statement의 (⑥) 메서드를 사용하고, 일괄 작업을 수행하고자 할 때에는 executeBatch() 메서드를 사용하면 된다.

정답 :

11. 웹 애플리케이션 초기화 매개변수는 웹 애플리케이션이 컨테이너에 의해 구동될 때 로딩되는 정보로서, 웹 애플리케이션 전반에 걸쳐 공통적으로 참조하는 값을 설정하는 용도로 사용한다. 다음 내용을 채우시오.[12]

서블릿 초기화 매개변수는

|  |
| --- |
| ① |

를 통해 접근할 수 있다. 이것은 서블릿 단위로 설정하기 때문에 해당 서블릿에서만 참조할 수 있다. 서블릿 동작에 필요한 정보를 제공하려는 목적으로 사용한다. 웹 애플리케이션 초기화 매개변수는

|  |
| --- |
| ② |

를 통해 접근 할 수 있다. 초기 값의 설정은 web.xml에서 한다.

|  |  |  |
| --- | --- | --- |
| 구분 | 적용 범위 | web.xml 설정 |
| |  | | --- | | ① | | 해당 서블릿에만 사용가능하다. | ③   |  | | --- | |  |   ⑤   |  | | --- | |  |   ⑦   |  | | --- | |  |   ⑧   |  | | --- | |  |   ⑥   |  | | --- | |  |  |  | | --- | | ④ | |
| |  | | --- | | **②** | | 동일한 웹 애플리케이션 내 모든 서블릿(혹은 jsp)에서 사용가능하다. | ⑨   |  | | --- | |  |   ⑪   |  | | --- | |  |   ⑦   |  | | --- | |  |   ⑧   |  | | --- | |  |   ⑫   |  | | --- | |  |   ⑩   |  | | --- | |  | |

12. 웹 애플리케이션 오류관리에서 오류 처리 코드와 관련된 내용이다. 내용을 보고 알맞은 오류 처리 코드를 고르시오.[5]

|  |  |  |
| --- | --- | --- |
| **번 호** | **오류 처리 코드** | **내용** |
| ① |  | Not implemented(클라이언트에서 서버가 수행할 수 없는 행동을 요구함 |
| ② |  | Internal server error, 내부 서버 오류(잘못된 스크립트 실행 시) |
| ③ |  | Not found(문서를 찾을 수 없음) |
| ④ |  | Bad request(클라이언트의 잘못된 요청으로 처리할 수 없음) |
| ⑤ |  | Forbidden(접근이 거부된 문서를 요청함) |

13. jsp 프로그램을 개발할 때 폴더(디렉토리) 구조와 배포할 때 폴더(디렉토리) 구조를 각각 적으시오.[그림 참고] [12]

|  |
| --- |
| 단, 작업은 이클립스(Eclipse)에서 사용하여 개발하고 war파일로 배포한다고 가정한다.  프로젝트 명 : jspbook  작업공간(workspace) : c:\dev\workspace — 줄여서 workspace로 표시 가능  컨테이너(톰킷) : c:\program files\tomcat – 줄여서 tomcat로 표시 가능  배포서술자 : web.xml  배포할 때 소스코드(src)도 배포함 |

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

① 개발할 때 jsp 파일를 포함하는 **폴더의 경로**와

② 배포할 때 jsp파일을 포함하는 **폴더의 경로**를 적으시오.

③ 개발할 때 web.xml를 포함하는 폴더의 경로와

④ 배포할 때 web.xml를 포함하는 **폴더의 경로**를 적으시오.

⑤ 개발할 때 소스 코드를 포함하는 **폴더의 경로**와

⑥ 배포할 때 클래스파일과 소스코드를 포함하는 **폴더의 경로**를 적으시오.

⑦ 개발할 때 라이브러리 파일을 포함하는 **폴더의 경로**와

⑧ 배포할 때 라이브러리 파일을 포함하는 **폴더의 경로**를 적으시오.

⑨ 개발할 때 context.xml를 포함하는 **폴더의 경로**와

⑩ 배포할 때 context.xml를 포함하는 **폴더의 경로**를 적으시오.

⑪ 개발할 때 log4j.xml를 포함하는 **폴더의 경로**와

⑫ 배포할 때 log4j.xml를 포함하는 **폴더의 경로**를 적으시오.

14. 다음 조건을 만족하는 log4j.xml를 작성하시오(주석을 참고하여 작성).[12]

(답안 작성의 편의를 위하여 org.apache.log4j 로 시작하는 것은 log4j만 사용하면 됨.)

|  |
| --- |
| 로그 레벨 : warn  출력대상 : 콘솔, 파일  레이아웃 : 1) 콘솔 출력 - 패턴레이아웃  2) 파일 출력 - 심플레이아웃,  로그 파일명: c:/tmp/out.log (jspbook.ch14 패키지에 있는 파일이 실행될 때만 기록)  최대 파일 사이즈 : 20 kb  출력 패턴 : 주석 참고 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| <?xml version="1.0" encoding="UTF-8"  <!DOCTYPE log4j:configuration PUBLIC "-//APACHE//DTD LOG4J 1.2//EN" "http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/xml/doc-files/log4j.dtd">  <log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">  <appender name="console"   |  | | --- | | ① |   > <!-- 콘솔에 로크 메시지 출력 -->  <layout class=   |  | | --- | | ② |   > <!-- 패턴레이아웃 지정 -->  <param name="   |  | | --- | | ③ |   “   |  | | --- | | ④ |   /> **<!--시간(시:분:초)만 출력, 카테고리, 로깅 레벨, 로그 내용, 개행 문자출력 -->**  </layout>  </appender>  <appender name="logFile"   |  | | --- | | ⑤ |   > <!-- 파일에 로그 메시지를 기록하고, 파일 크기가 일정 수준 이상이 되면 다른 이름으로 저장하고 새롭게 기록한다. -->  <param name="file"   |  | | --- | | ⑥ |   /> <-- 저장할 로그 파일 지정 -->  <param name="Append" value="true"/>  <param name="MaxFileSize"   |  | | --- | | ⑦ |   /> <-- 저장할 파일의 최대크기 지정 -->  <param name="MaxBackupIndex" value="1"/>  <layout class=   |  | | --- | | ⑧ |   /> <!--심플레이아웃으로 지정 -->  </appender>  <   |  | | --- | | ⑨ |   “jspbook.ch14”>  ...........  <root>   |  | | --- | | ⑩ |   <!-- 로그 레벨 -->   |  | | --- | | ⑪ |   <!-- 표준 출력 -->   |  | | --- | | ⑫ |   <!-- log 파일에 출력 -->  </root>  </log4j:configuration> |

15. 다음 조건을 만족하는 log4j.xml를 작성하시오(주석을 참고하여 작성).[17]

(답안 작성의 편의를 위하여 org.apache.log4j 로 시작하는 것은 log4j만 사용하면 됨.)

|  |
| --- |
| 로그 레벨 : WARN  출력대상 : 파일  레이아웃 : 1) 첫 번째 파일 출력 - 패턴레이아웃  2) 두 번째 파일 출력 - HTML레이아웃,  로그 파일명: c:/temp/dailyout.log, c:/temp/rolling.log.html  최대 파일 사이즈 : 1024 kb  출력 패턴 : 주석 참고 |

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">  <log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">  <appender name="dailyout" ( **①** ) ><!--하루 단위로 파일에 로그 메시지를 기록해서 저장함 -->  <param name="file" ( **②** ) />  <param name="Append" value="true"/>  <!-- <param name="DatePattern" value="'.'yyMMdd"/> -->  <layout ( **③** ) > 패턴레이아웃 지정  <param ( **④** ) value="기말고사> [ **⑤** ] [ **⑥** ] [ **⑦** ] [ **⑧** ] [ **⑨** ] [ **⑩** ] [ **⑪** ] [ **⑫** ] **⑬** /> <-- 출력 값은 아래 내용 참고할 것>  </layout>  </appender>  <appender name="rolling" ( **⑭** )> <--파일에 로그 메시지를 기록하고, 파일 크기가 일정 수준 이상이 되면 다른 이름으로 저장하고 새롭게 기록한다. -->  <param name="file" ( **⑮** ) /> <-- 저장할 파일 지정 -->  <param name="Append" value="true"/>  <param ( ) value="1024kb"/> <!-- 저장할 파일의 최대크기 지정 -->  <param name="MaxBackupIndex" value="1"/>  <layout ( ) /> <!--Html레이아웃으로 지정 -->  </appender>  <root>  <level value="WARN" />  <appender-ref ref="dailyout" />  <appender-ref ref="rolling" />  </root>  </log4j:configuration> |

|  |
| --- |
| ⑤ ~ ⑬ 까지의 참고 사항이며, 출력 내용은 다음과 같다.  기말고사> [logtest\_jsp] [WARN] [2016-06-15 10:31:05] [http-nio-8080-exec-2] [org.apache.jsp.ch14.logtest\_jsp] [logtest\_jsp.java] [132] [warn-jsp 파일에서 처리한 로그]  [logtest\_jsp]는 카테고리, [http-nio-8080-exec-2]는 스레드 이름, [warn-jsp 파일에서 처리한 로그]는 로그 내용이다. |

[Bonus] 웹기반정보시스템및실험 과목을 수강하면서 아쉬웠던 점과 좋았던 점을 적으시오.[5]

**한 학기 동안 수고했습니다. 즐거운 여름 방학 되기를...**