웹프로젝트실습 중간고사 정리

3장

1. JSP란

1) JSP는 Java Server Page의 약자로서, 자바를 기반으로 만들어진 웹 프로그래밍 기술이다.

2) JSP는 HTML처럼 화면 위주의 프로그래밍을 지원하며, 스크립트릿, 빈즈 액션 등 자바 프로그램과 연동할 수 있는 구조로 되어 있다.

3) JSP를 학습하려면 HTML, 자바스크립트, CSS와 같은 기본 기술들과 다양한 자바 관련 응용 기술이 요구된다.

2. JSP 내부 동작 구조

1) JSP는 자바 서블릿 기반의 웹 프로그래밍 기술로서, JSP 소스 자체로는 실행할 수 없는 상태다.

2) 톰캣은 서블릿 컨테이너로서, JSP를 서블릿 형태의 자바 소스로 변환하고 자바 클래스로 컴파일 하고 서블릿으로 로딩한 다음 사용자 요청에 응답하는 구조다.

3) 이때 한번 컴파일된 JSP는 소스 수정 전까지 다시 컴파일되지 않고 메모리상에 적재된 서블릿으로 처리된다.

4) 개발자가 작성한 JSP 소스는 변환된 소스의 \_jspService() 메서드에 들어간다.

3. JSP 프로그래밍 기술 변천

1) 초기 자바 기반 웹프로그램은 서블릿 중심이었으나, 웹이 화면의 중심이라는 특성으로 인해 프로그램 내에서 화면을 처리하는 데 여러 문제를 야기한다.

2) 이후 화면을 중심으로 자바 프로그램을 연동할 수 있는 JSP가 등장하게 되었으나 JSP 내에서 과도한 스크립트릿을 사용하는 문제로 프로그램 개발 구조의 변화가 요구된다.

3) MVC 패턴은 웹 프로그램을 역할에 따라 기술 요소를 구분해서 구현하는 프로그래밍 모델로 이후 웹 프로그램 개발의 기본 모델로 정착되었다.

4장

1. 서블릿이란

1) 서블릿은 자바 플랫폼에서 컴포넌트 기반 웹 애플리케이션을 개발하기 위한 핵심 기술이다.

2) JSP의 고급 활용을 위해서는 서블릿 기술에 대한 이해가 필요하다.

3) 서블릿은 JSP와 달리 처음부터 순수 자바 코드로 개발하기 때문에 자바가 제공하는 거의 모든 기능을 활용할 수 있다.

4) 서블릿은 컨테이너(톰캣)에 의해 관리되는 웹 어플리케이션으로 기본적인 생명주기 관리가 컨테이너에 의해 이루어진다.

5) 개발자는 해당 생명주기 메서드를 오버라이딩하거나 doGet, doPost와 같은 사용자 요청 처리 메서드만 구현할 수도 있다.

2. 서블릿 개발과정

1) 서블릿은 javax.servlet.GenericServlet을 상속받는 javax.servlet.HttpServlet 클래스를 상속받아 구현할 수 있다.

2) 서블릿의 생명주기는 생성자 -> init -> service -> doPost/doGet -> destroy인데, 각각의 메서드가 컨테이너에 의해 호출되어 실행된다.

3) 기본 생명주기 외에 HTTP 프로토콜 처리를 위한 메서드도 사용할 수 있다.

4) 서블릿 실행은 컨테이너에 의해 관리되므로 서블릿 실행관련 정보를 컨테이너에 설정해야 한다. 기존에는 web.xml 파일에 서블릿 실행 정보를 기술했으나 서블릿 3.0 스펙 이상에서 부터는 애너테이션 기반으로 클래스 파일에 직접 기술한다.

3. GET 방식과 POST 방식

1) 서블릿 프로그램을 작성하려면 생명주기 메서드는 doGet 혹은 doPost 메서드를 구현해야 한다.

2) GET과 POST는 HTTP 프로토콜의 대표적인 명령으로 GET은 서버에 있는 정보를 가지고 오기 위한 요청이고, POST는 클라이언트 데이터를 서버에 전송하기 위한 요청을 정의한다.

3) 따라서 처리하고자 하는 요청에 따라 doGet 혹은 doPost 메서드를 개발자가 오버라이딩해서 구현하면 된다.

4) GET은 URL에 매개변수 형식으로 데이터를 전달한다. (e.f. http://www.xxx.co.kr/servlet/login?id=아이디&name=이름)

5) POST 방식은 HTML 폼에서 작성한 내용 등을 전달할 때 사용하며 URL에 보이지 않고 데이터 전송 크기에 제한이 없다.

5장

1. 주석

1) 주석은 프로그램 소스에 이해를 도우려는 설명문이다.

2) JSP는 자바, HTML, JSP 주석을 혼용해서 사용할 수 있다.

2. 지시어

1) 지시어는 JSP 페이지에 대한 전반적인 설정 정보를 기술하기 위한 부분이다.

2) page, include, taglib 등의 지시어가 있다.

3) page 지시어는 문서형식, 캐릭터셋, 자바 패키지 불러오기, 오류 페이지 설정 등을 담당한다.

4) include 지시어는 현재 위치에 다른 페이지를 포함하기 위한 지시어로, 화면의 모듈화 및 JSP 파일을 공유할 수 있게 한다.

4) taglib 지시어는 커스텀 태그 사용을 위한 설정으로 JSTL을 사용하기 위해서도 필요하다.

3. 액션

1) 액션은 JSP 주요 구성요소 중 하나로 JSP 페이지 간 흐름제어, 자바 애플릿지원, 자바 빈즈 컴포넌트와 JSP 상호작용을 지원한다.

2) include 액션은 include 지시어와 유사한 페이지 포함 기능을 제공한다.

3) forward 액션은 사용자 요청을 다른 페이지로 전달한다.

4) useBean 액션은 빈즈 클래스를 사용하고 속성을 관리한다.

4. 선언식과 표현식

1) 선언은 <%! %> 사이에 기술하는데, 메서드, 멤버변수의 선언 등에서 사용할 수 있다.

2) JSP에서 자바 프로그램적인 선언을 가능하게 하는 문법으로 MVC 패턴을 적용할 때에는 사용을 권하지 않는다.

3) 표현식은 <%= %> 사이에 기술하며 간단한 변수나 메서드 실행의 결과를 출력할 수 있다. 유용한 구문이지만 최근에는 표현 언어를 주로 사용하고 있다.

5. 스크립트릿

1) JSP의 가장 대표적인 요소로 자바 프로그램을 HTML 문서와 혼용해서 작성할 수 있다.

2) 초기 JSP의 대표 장점으로 인식되었으나 MVC 패턴의 사용으로 최근에는 스크립트릿 사용을 권장하지 않는다.

연습문제

1. HTML 소스에는 보이지 않는 JSP 주석 표기법은 무엇인가?

<%-- --%>

2. page 지시어의 기본적인 사용 예를 들어 설명하시오.

현재 페이지에 다른 페이지를 포함하는 지시어로 네이버의 main 화면에 다양한 jsp 파일을 포함하여 모듈화 한 것이 그 예이다.

3. 현재 페이지의 특정 위치에 다른 jsp를 포함할 수 있는 JSP 문법 두 가지를 쓰고 차이를 설명하시오.

<jsp:include page=”” flush=””>, <%@ include file=”” %>

include 액션은 별도의 파일로 요청 처리 흐름을 이동하고, include 지시어는 다른 파일의 내용을 현재 위치에 삽입시킨 후에 JSP 파일을 자바파일로 변환하고 컴파일하는 방식이다.

메서드

3) request.getParameter(“”);

4) response.sendRedirect(“”);

6장

1. 내장객체

1) 내장객체란 JSP 내에서 선언하지 않고 사용할 수 있는 자바 객체를 말한다.

2) 보통 스크립트릿에서 사용하며 내장객체를 통하면 사용자 요청을 컨트롤하거나 페이지 간의 데이터를 공유할 수 있다.

3) 구조적으로는 JSP가 변환된 서블릿 클래스에 선언되거나 메서드 매개변수로 전달된 객체들로 컨테이너에 의해 초기화되고 다양한 기능을 제공하는 메서드가 있다.

4) 내장객체 중 page, request, response, session, application은 빈즈 액션의 scope에서도 활용하며 각각 정해진 범위 안에서 유지되고 공유된다.

2. request, response 내장객체

1) request는 사용자 요청과 관련된 기능을 제공하는 내장객체다. 주로 HTML 폼으로부터 전달되는 값을 읽어오거나 캐릭터셋 변환을 수행한다.

2) response는 사용자 응답과 관련된 기능을 제공하는 내장객체로 사용자 요청에 대한 응답을 다른 페이지로 전달하는 기능을 제공한다.

3. out 내장객체

1) out은 출력 스트림으로써, 사용자 웹브라우저로의 출력을 위한 채널이다.

2) out을 이용해서 출력한 내용은 서버의 콘솔이 아닌 사용자에게 전달된다.

4. session, application 내장객체

1) session은 각각의 클라이언트마다 생성되는 객체로 각각 개별적인 정보를 유지하고 컨트롤할 수 있다.

2) 대표적으로 일정시간 유지되는 로그인 구현과 쇼핑몰의 장바구니 구현에 사용한다.

3) application은 애플리케이션 전체 즉, 한 프로젝트 내 전체 jsp, servlet 등에서 공유할 수 있는 객체로써, 이를 통해 애플리케이션 정보와 공유 데이터를 활용할 수 있다.

5. 내장객체의 속성 관리

1) request, session, application 내장객체는 각각의 고유 기능뿐 아니라 맵 형태의 속성을 관리하는 기능을 제공한다.

2) 이를 통해 jsp 페이지 간의 데이터를 공유할 수 있고 MVC 패턴 구현과 JSTL, 표현 언어를 활용할 수 있다.

3) 각 내장객체는 주어진 범위에서만 속성을 공유할 수 있다.

연습문제

1. 구조적 특징을 중심으로 JSP 내장객체를 설명하고 대표적인 내장객체를 나열하시오.

jsp가 서블릿 형태로 자동 변환된 코드 내에 포함되어 있는 멤버변수, 메서드, 매개변수 등의 각종 참조 변수를 말한다.

request, response, out, session, application 등

2. request 내장객체의 대표적인 기능은 무엇인지 설명하시오.

사용자의 요청에 대한 처리를 담당한다.

3. 내장객체의 속성 관리 기능에 대해 설명하고 session과 application 내장객체 속성 관리의 유효 범위를 쓰시오.

4개의 영역으로 나누어 변수의 사용 범위를 제안한다.

page는 하나의 jsp 파일 내에서 사용할 수 있다.

request는 하나의 요청된 페이지 내에서 사용할 수 있다.

session은 하나의 웹브라우저 내에서 사용할 수 있다.

application은 하나의 어플리케이션에서 사용할 수 있다.

메서드

1) session.setAttribute(“”, “”);

2) (String)session.getAttribute(“”);

3) session.removeAttribute(“”);