

〈6주차 실습〉

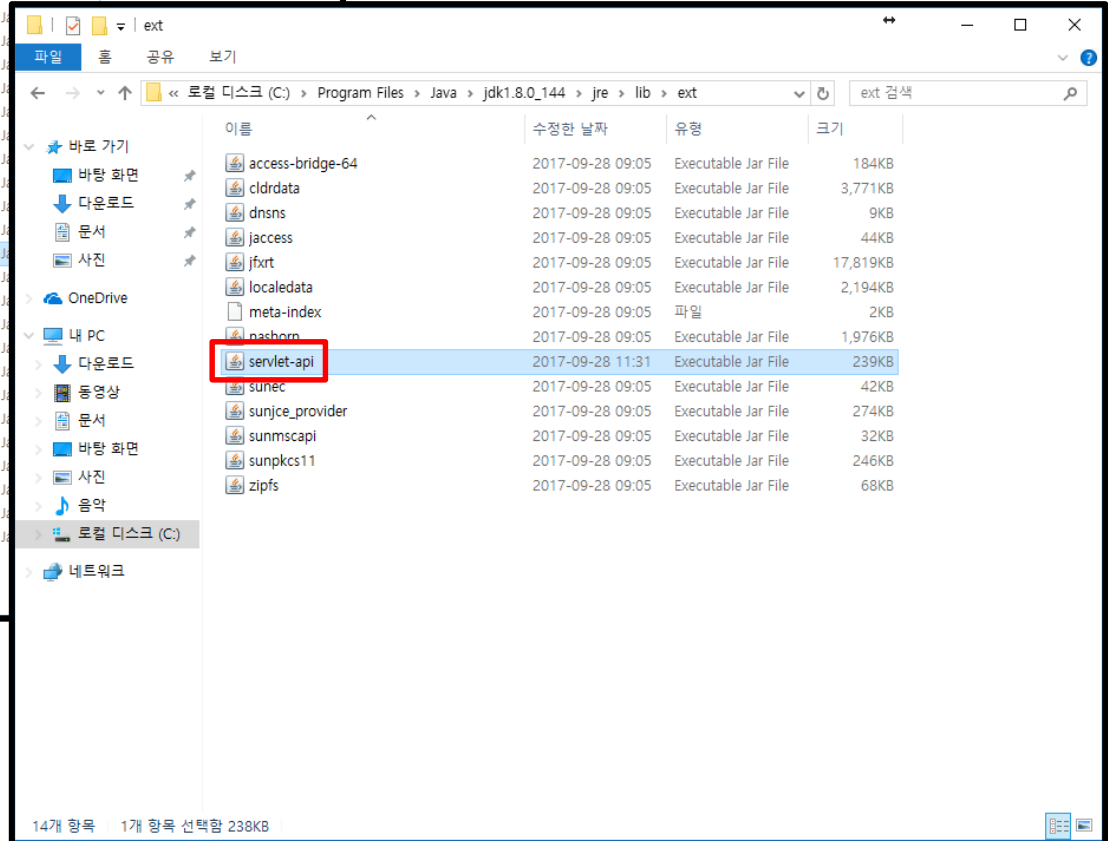
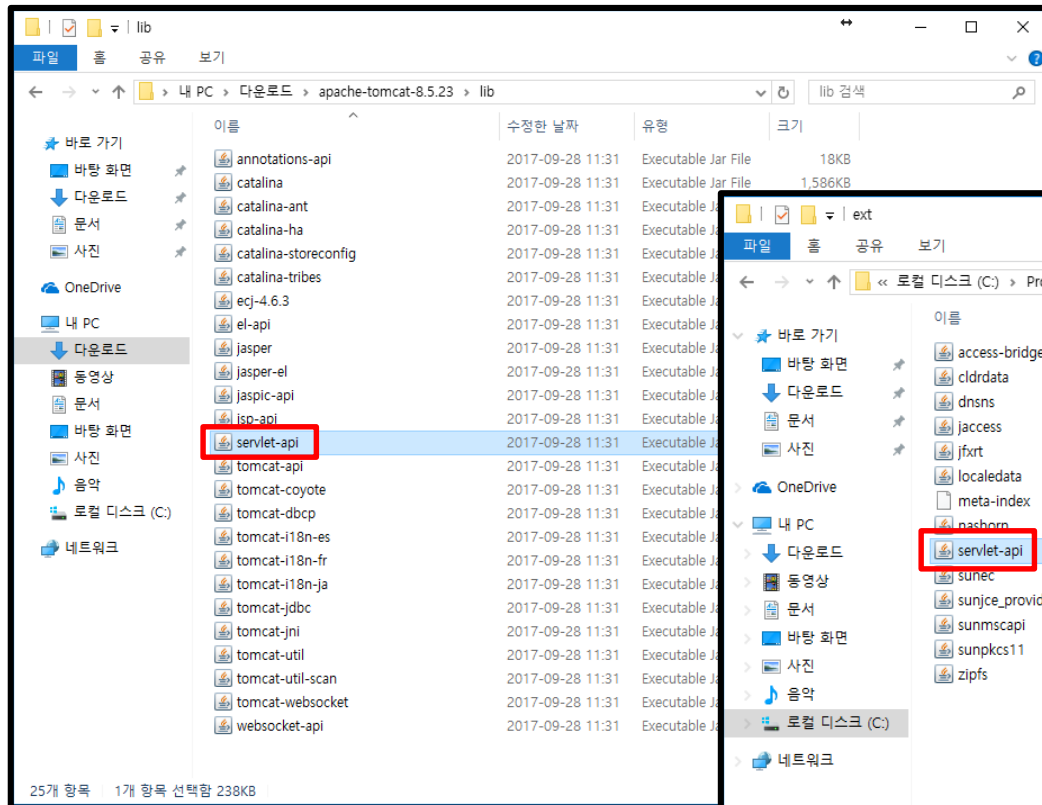
웹 프로그래밍

Web Programming

Tomcat 라이브러리 설정

1/21

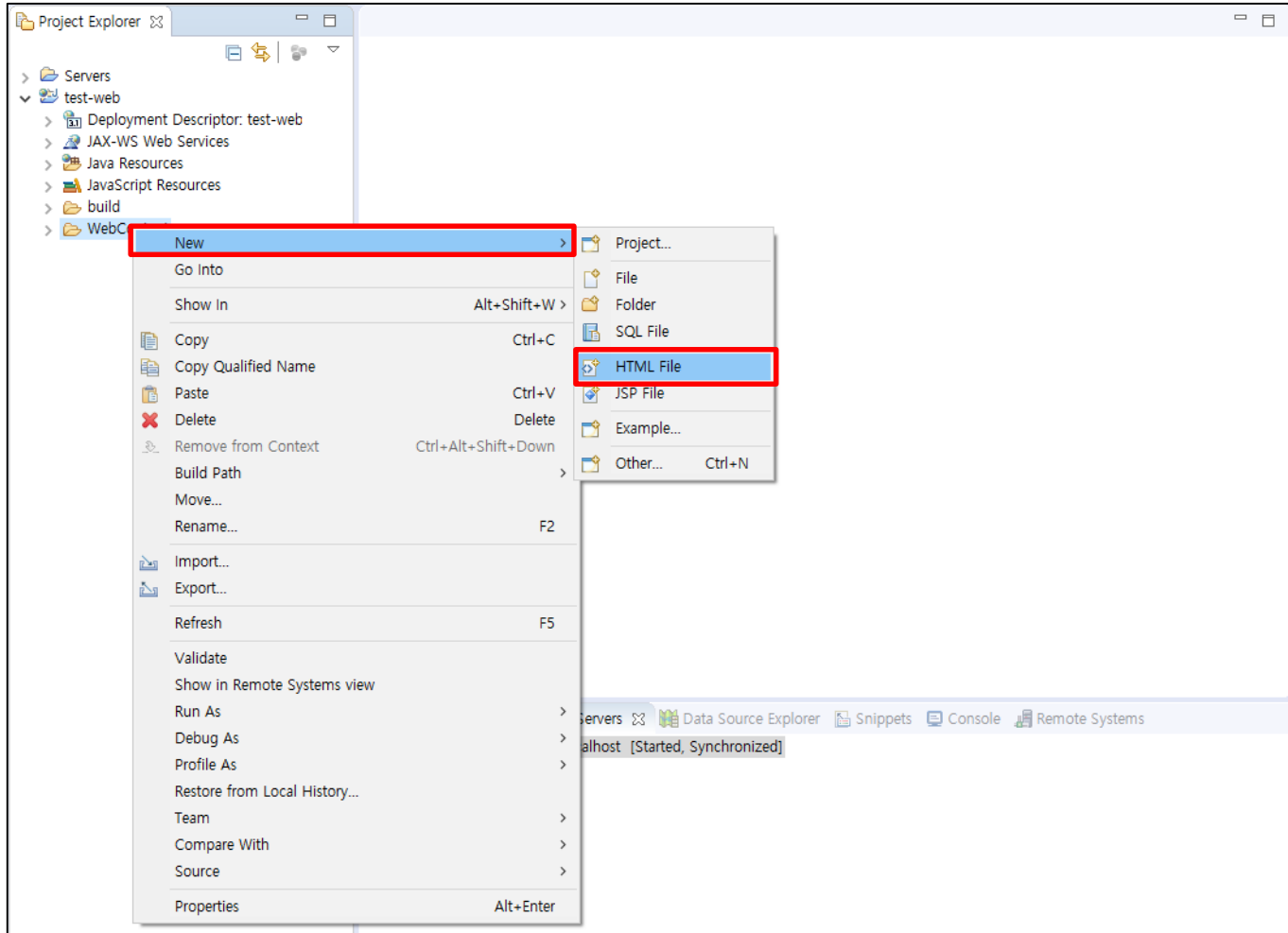
Tocat 설치 경로/lib 안의 servlet-api.jar을 jdk 설치 폴더/jre/lib/ext로 복사



Eclipse 를 이용한 Servlet 실행 (1/6)

2/21

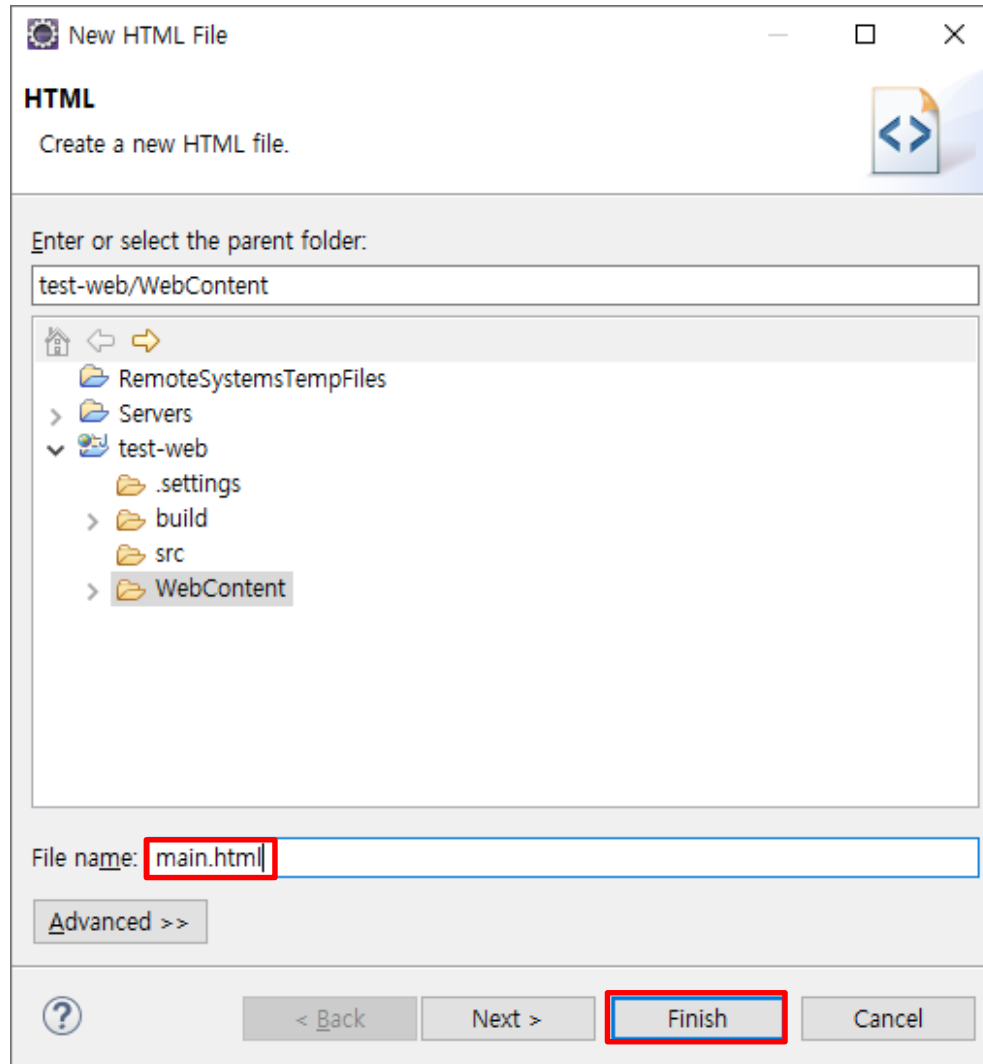
- 1) test-web 프로젝트에서 WebContent 폴더 오른쪽 마우스 클릭
- 2) New → HTML File



Eclipse 를 이용한 Servlet 실행 (2/6)

3/21

3) File name 에 main.html 입력 → Finish

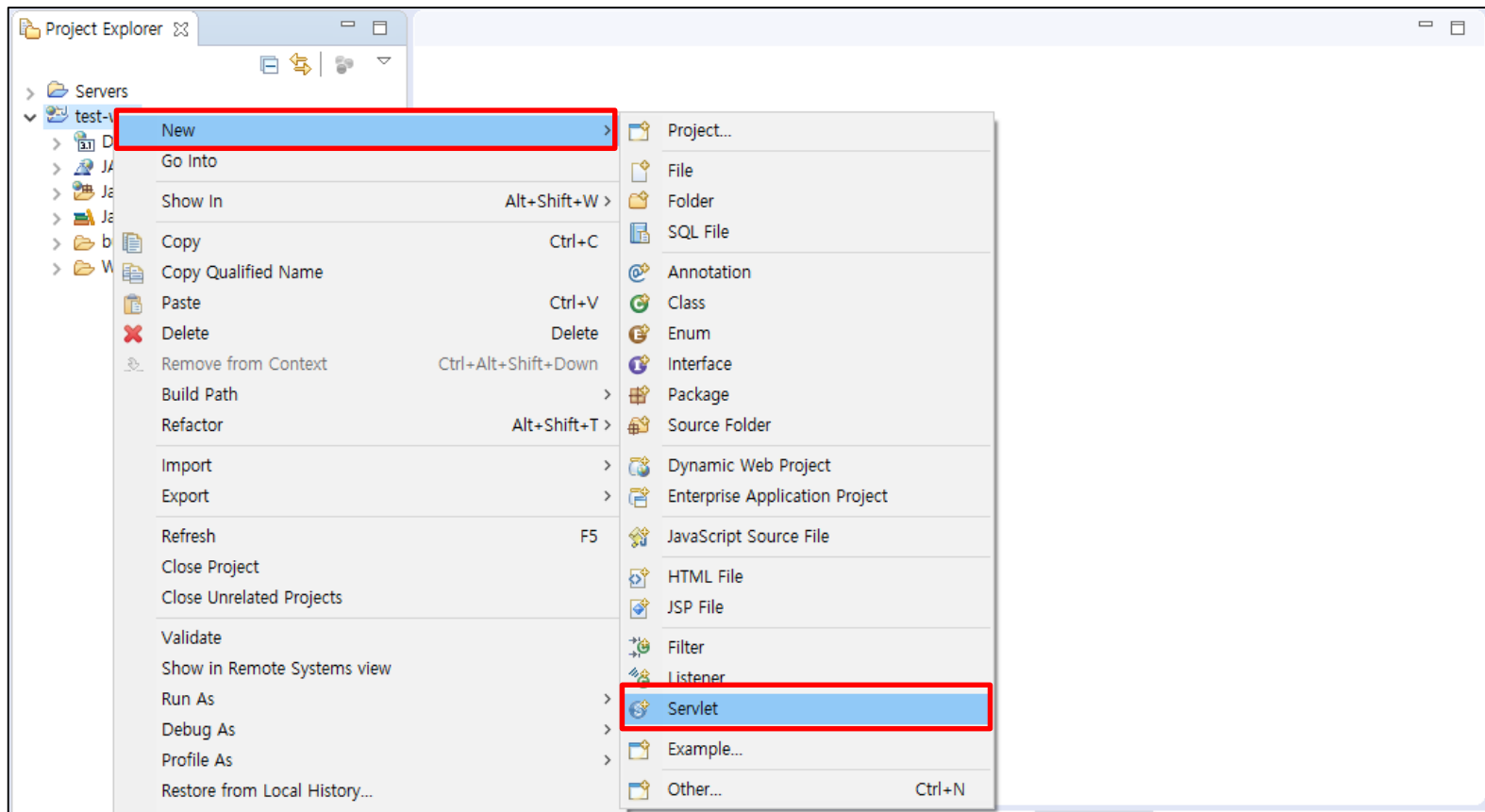


Eclipse 를 이용한 Servlet 실행 (3/6)

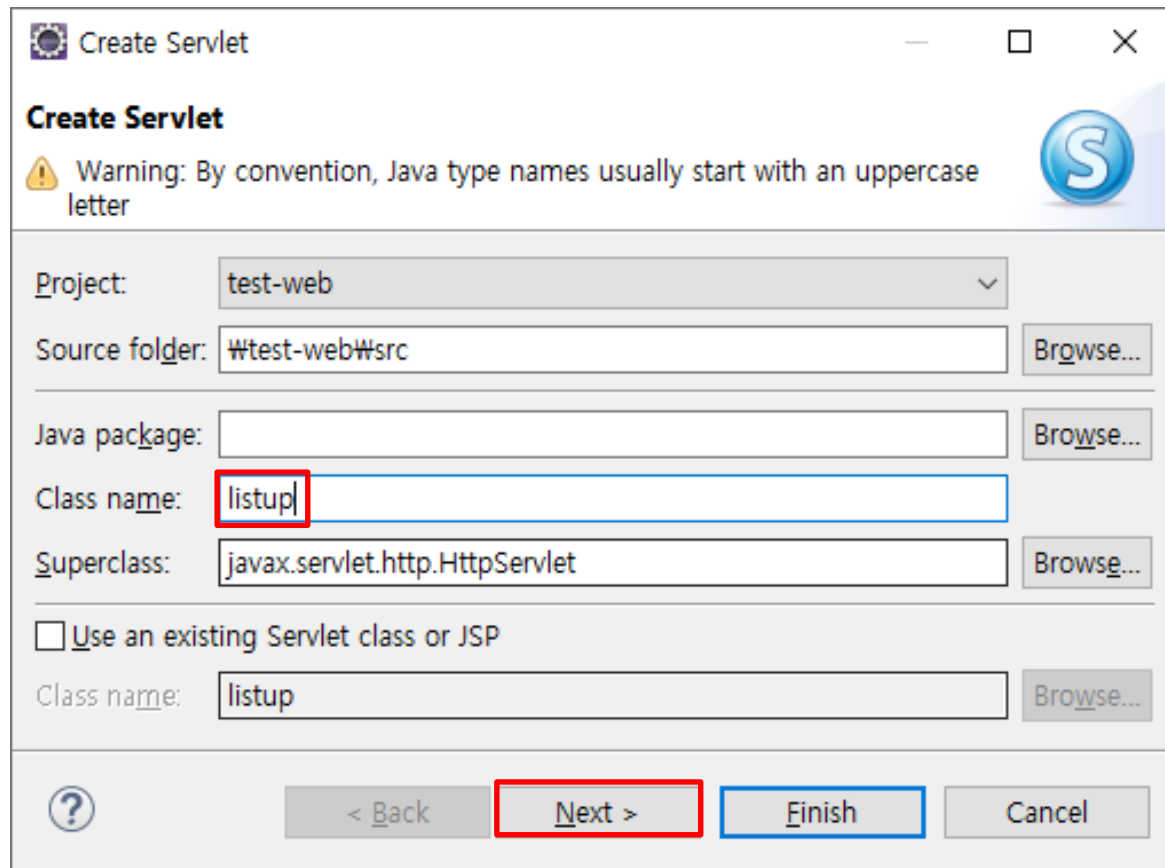
4/21

4) test-web 프로젝트를 마우스 오른쪽 클릭

5) New → Servlet 선택



6) Class name 부분에 listup 입력 -> Next



The image shows the 'Create Servlet' dialog box in Eclipse. The 'Class name' field is highlighted with a red box and contains the text 'listup'. The 'Next >' button at the bottom is also highlighted with a red box. Other fields include 'Project' (test-web), 'Source folder' (#test-web#src), 'Java package' (empty), 'Superclass' (javax.servlet.http.HttpServlet), and 'Class name' (listup). A warning message at the top states: 'Warning: By convention, Java type names usually start with an uppercase letter'.

Create Servlet

Warning: By convention, Java type names usually start with an uppercase letter

Project: test-web

Source folder: #test-web#src

Java package:

Class name: listup

Superclass: javax.servlet.http.HttpServlet

☐ Use an existing Servlet class or JSP

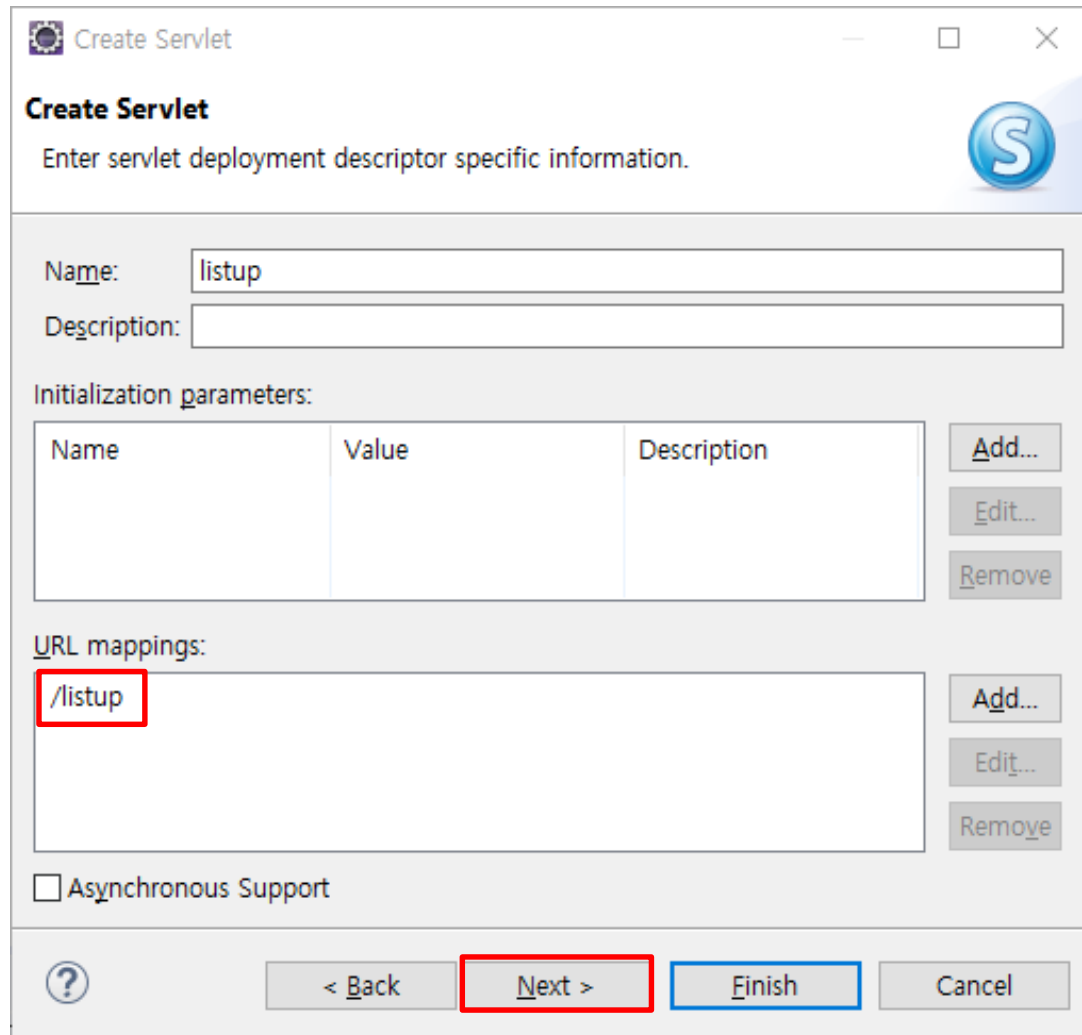
Class name: listup

< Back Next > Finish Cancel

Eclipse 를 이용한 Servlet 실행 (5/6)

6/21

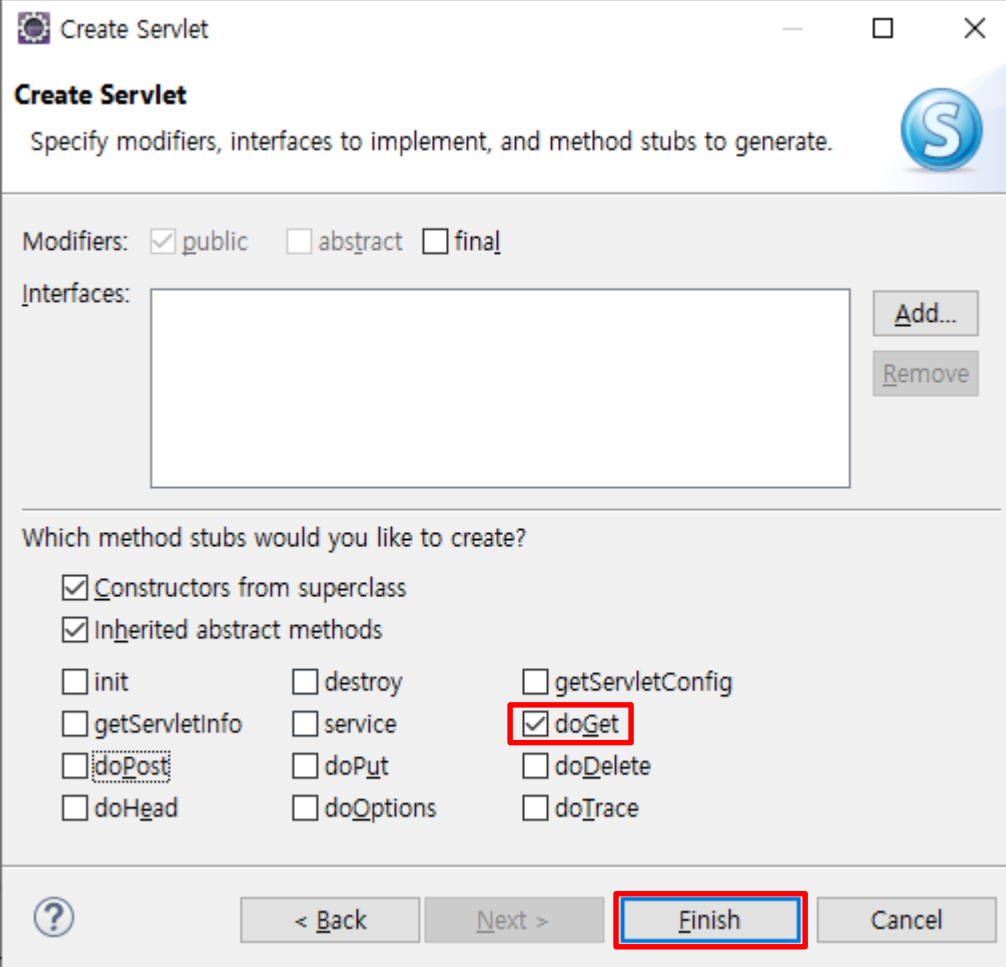
7) URL mappings: 확인 후 Next 클릭



The image shows the 'Create Servlet' dialog box in the Eclipse IDE. The dialog has a title bar with the Eclipse logo and the text 'Create Servlet'. Below the title bar, there is a section titled 'Create Servlet' with a description: 'Enter servlet deployment descriptor specific information.' and a blue 'S' icon. The main area contains several fields and buttons:

- Name:** A text field containing 'listup'.
- Description:** An empty text field.
- Initialization parameters:** A table with three columns: 'Name', 'Value', and 'Description'. To the right of the table are three buttons: 'Add...', 'Edit...', and 'Remove'.
- URL mappings:** A text area containing '/listup', which is highlighted with a red rectangle. To the right of the text area are three buttons: 'Add...', 'Edit...', and 'Remove'.
- Asynchronous Support:** A checkbox that is currently unchecked.
- Navigation buttons:** At the bottom, there are four buttons: a help button (question mark icon), '< Back', 'Next >' (highlighted with a red rectangle), and 'Finish' (highlighted with a blue rectangle). A 'Cancel' button is also present.

8) doGet 체크 -> Finish (doGet/doPost 원하는 방식을 선택)



The image shows the 'Create Servlet' dialog box in the Eclipse IDE. The dialog has a title bar with the Eclipse logo and the text 'Create Servlet'. Below the title bar, there is a section titled 'Create Servlet' with a subtitle 'Specify modifiers, interfaces to implement, and method stubs to generate.' and a blue 'S' icon. The 'Modifiers' section has three checkboxes: 'public' (checked), 'abstract' (unchecked), and 'final' (unchecked). The 'Interfaces' section has a text box and two buttons: 'Add...' and 'Remove'. The 'Which method stubs would you like to create?' section has a list of checkboxes: 'Constructors from superclass' (checked), 'Inherited abstract methods' (checked), 'init' (unchecked), 'destroy' (unchecked), 'getServletConfig' (unchecked), 'getServletInfo' (unchecked), 'service' (unchecked), 'doGet' (checked and highlighted with a red box), 'doPost' (unchecked and highlighted with a red box), 'doPut' (unchecked), 'doDelete' (unchecked), 'doHead' (unchecked), 'doOptions' (unchecked), and 'doTrace' (unchecked). At the bottom, there is a row of buttons: a help button (question mark icon), '< Back' (disabled), 'Next >' (disabled), 'Finish' (highlighted with a red box), and 'Cancel' (disabled).

Create Servlet

Create Servlet

Specify modifiers, interfaces to implement, and method stubs to generate.

Modifiers: ☒ public ☐ abstract ☐ final

Interfaces:

Add...

Remove

Which method stubs would you like to create?

☒ Constructors from superclass

☒ Inherited abstract methods

☐ init ☐ destroy ☐ getServletConfig

☐ getServletInfo ☐ service ☒ doGet

☐ doPost ☐ doPut ☐ doDelete

☐ doHead ☐ doOptions ☐ doTrace

? < Back Next > Finish Cancel

1) main.html 코드

```
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <form action="listup" method="get">
      <h1>색상 선택</h1>
      <select name="color">
        <option value="red">붉은색</option>
        <option value="black" selected="selected">검은색</option>
        <option value="blue">파란색</option>
        <optgroup label="기타 색상">
          <option value="purple">보라색</option>
          <option value="pink">분홍색</option>
        </optgroup>
      </select>
      <input type="submit" value="선택 넘기기">
    </form>
  </body>
</html>
```

2) listup.java 코드

```
import java.io.PrintWriter;
```

일부분 생략

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)  
throws ServletException, IOException {
```

```
response.setContentType("text/html;charset=UTF-8");
```

```
String color = request.getParameter("color");
```

```
PrintWriter out = response.getWriter();
```

```
out.println("<HTML>");
```

```
out.println("<HEAD></HEAD>");
```

```
out.println("<BODY>");
```

```
out.println("단일 선택 항목 색상 = " + color + "<br/><br/>");
```

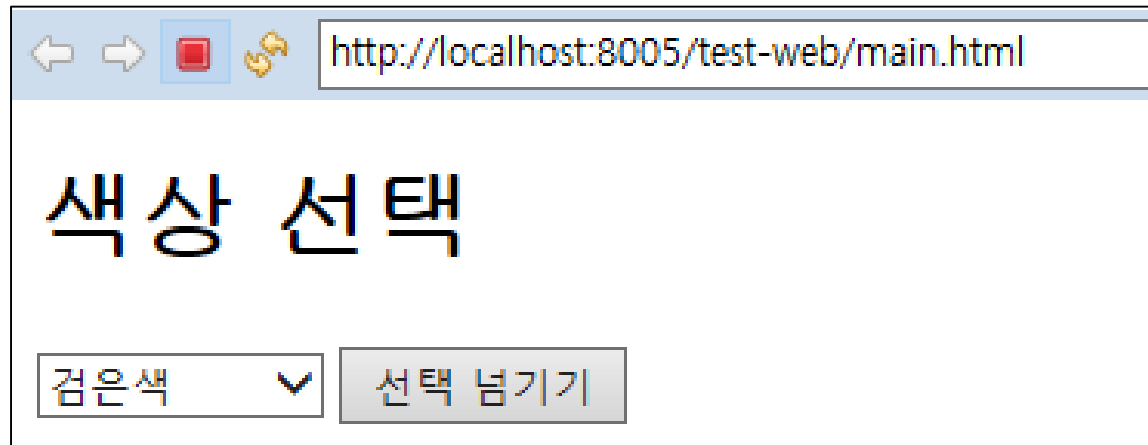
```
out.println("</BODY>");
```

```
out.println("</HTML>");
```

```
}
```

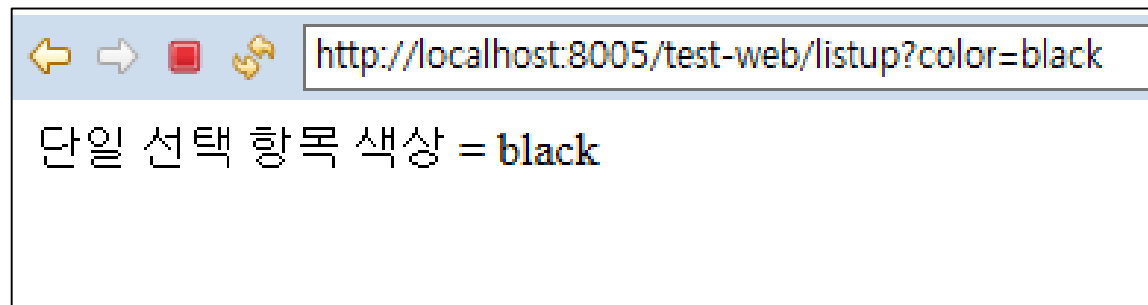
일부분 생략

<HTML 파일 실행>



분홍색 항목 선택 후 버튼 클릭

<Java 파일 호출>



실습 과제 <실험·실습 제출자료>

11/21

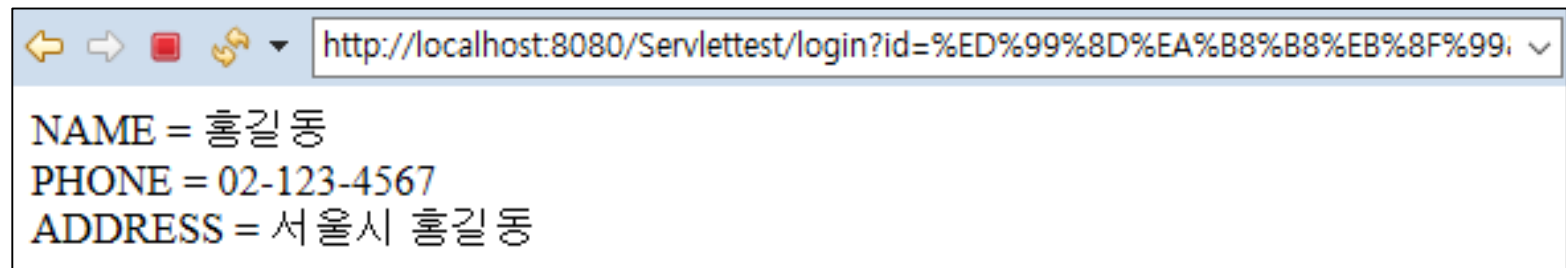
Q1) 아래와 같은 결과가 나오도록 HTML 코드와 Servlet 코드를 작성하시오.

<HTML 파일 실행>



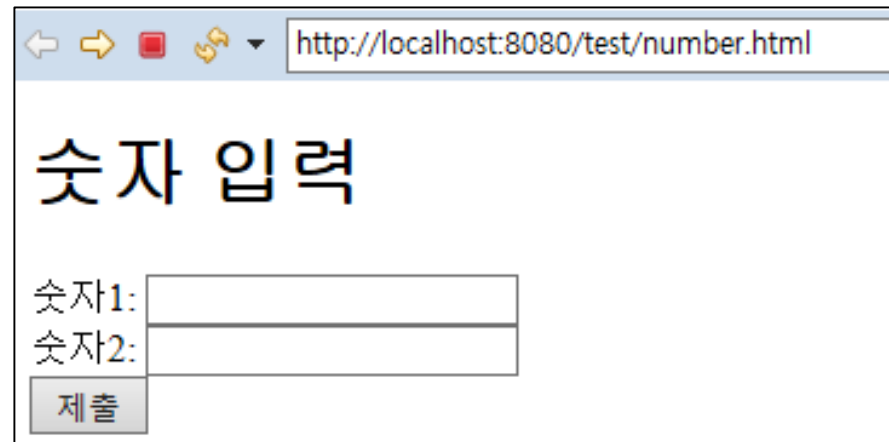
정보 입력 후 “제출”

<Java 파일 호출>



Q2) 아래와 같은 결과가 나오도록 HTML 코드와 Servlet 코드를 작성하시오.

<HTML 파일 실행>



숫자 입력

숫자1:

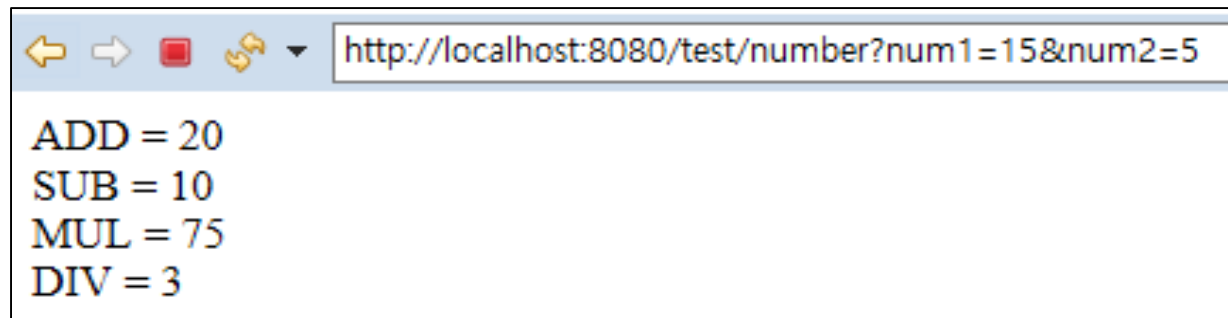
숫자2:

제출



값 입력 후 “제출”

<Java 파일 호출>

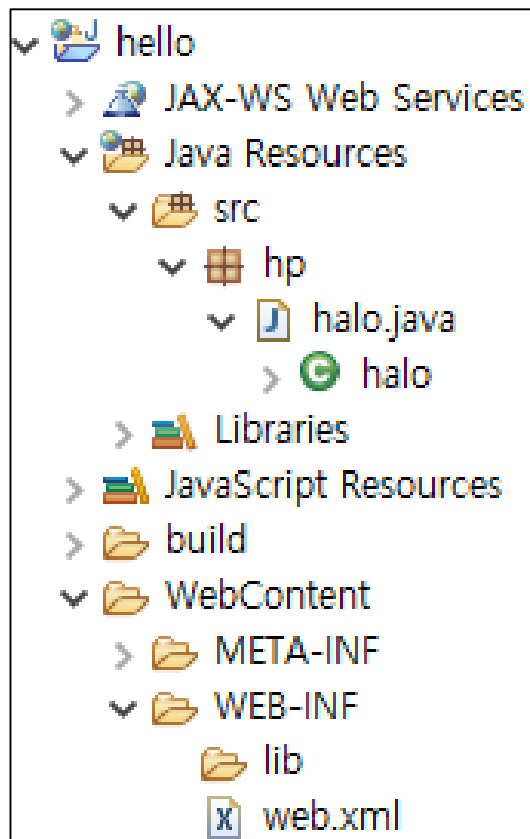


http://localhost:8080/test/number?num1=15&num2=5

ADD = 20
SUB = 10
MUL = 75
DIV = 3

Q3) 아래와 같은 구조로 프로젝트를 생성하고, 아래와 같이 실행되도록 Servlet 코드와 XML 코드를 작성하시오.

<프로젝트 구조>



<halo.java 파일 실행 결과>



Q3) 아래와 같은 구조로 프로젝트를 생성하고, 아래와 같이 실행되도록 Servlet 코드와 XML 코드를 작성하시오.

<halo.java 코드>

```
package   
  
import java.io.IOException;  
import javax.servlet.ServletException;  
import javax.servlet.annotation.WebServlet;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import java.io.PrintWriter;  
  
@WebServlet("/halo")  
public class halo extends HttpServlet {  
    private static final long serialVersionUID = 1L;  
  
    public halo() {  
        super();  
    }  
  
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
        response.setContentType("text/html;charset=UTF-8");  
  
        PrintWriter out =   
  
        out.println("<html>" + "<body>" + "<h1>XML에서 Servlet URL mapping 하기</h1>" + "</body>" + "</html>");  
    }  
  
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
        doGet(request, response);  
    }  
}
```

Q3) 아래와 같은 구조로 프로젝트를 생성하고, 아래와 같이 실행되도록 Servlet 코드와 XML 코드를 작성하시오.

<web.xml 코드>

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
  id="WebApp_IC" version="3.0">
  <display-name>hello</display-name>
  <servlet>
    <servlet-name>helloServlet</servlet-name>
    <servlet-class>[redacted]</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>[redacted]</servlet-name>
    <url-pattern>[redacted]</url-pattern>
  </servlet-mapping>

  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

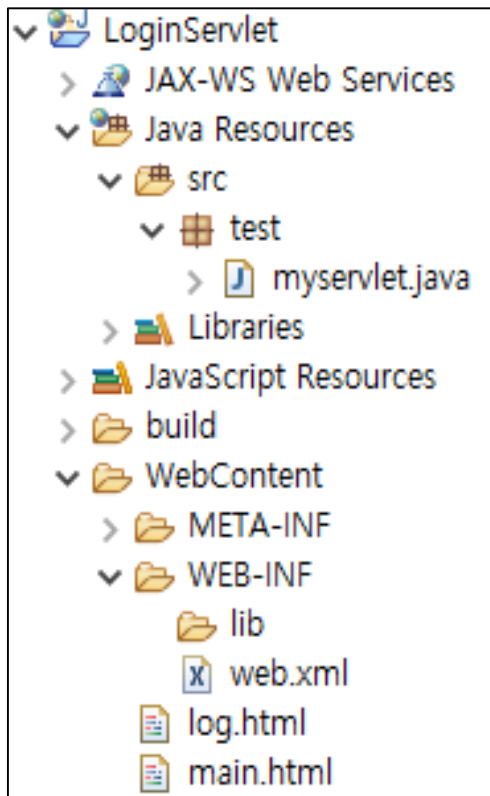

실습 과제 <실험·실습 제출자료>

16/21

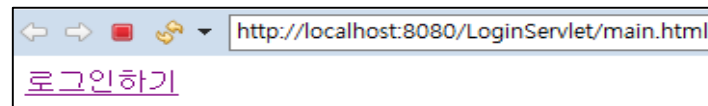
Q4) 아래와 같은 구조로 프로젝트를 생성하고, 아래와 같이 동작될 수 있도록 코드를 작성하시오.

- 1) html 코드 작성
- 2) servlet 코드 작성
- 3) xml 코드 작성
- 4) post 방식으로 동작

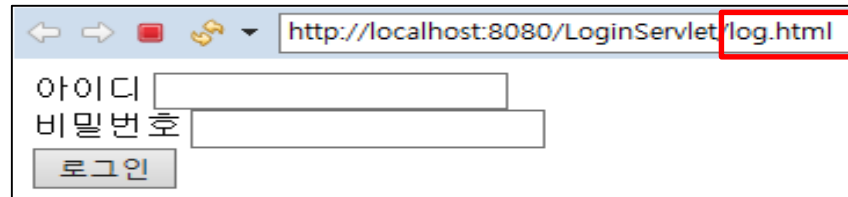
<프로젝트 구조>



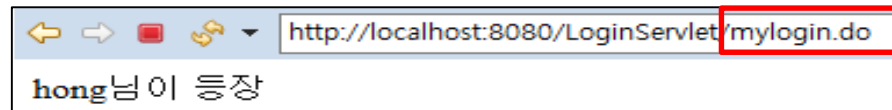
<main.html 파일 실행 화면>



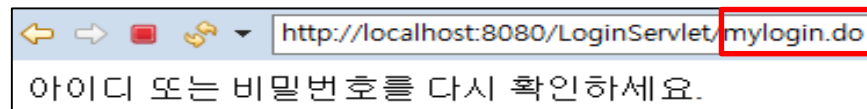
<“로그인하기” 클릭 후 실행 화면>



<로그인 성공 화면>



<로그인 실패 화면>



Q4) 아래와 같은 구조로 프로젝트를 생성하고, 아래와 같이 동작될 수 있도록 코드를 작성하시오.

<main.html 코드>

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
    <a href=" " >로그인하기</a>
</body>
</html>
```

Q4) 아래와 같은 구조로 프로젝트를 생성하고, 아래와 같이 동작될 수 있도록 코드를 작성하시오.

<log.html 코드>

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
  <form method=" " action=" ">
    아이디 <input type="text" name="id"/><br/>
    비밀번호 <input type="text" name="pwd"/><br/>
    <input type=" " value="로그인"/>
  </form>
</body>
</html>
```

Q4) 아래와 같은 구조로 프로젝트를 생성하고, 아래와 같이 동작될 수 있도록 코드를 작성하시오.

<myservlet.java 코드>

```
package test;

import java.io.IOException;

@WebServlet("/myservlet")
public class myservlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public myservlet() {
        super();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.sendRedirect(" ");
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        request.setCharacterEncoding(" ");
        response.setContentType(" ");

        String id = " ";
        String pwd = " ";
        PrintWriter out = " ";

        out.println("<html>");
        out.println("<head></head>");
        out.println("<body>");
        if(id.equals("hong")&&pwd.equals("1234")) {
            out.println(" + "님이 등장");
        }
        else {
            out.println("아이디 또는 비밀번호를 다시 확인하세요.");
        }
        out.println("</body>");
        out.println("</html>");
    }
}
```

Q4) 아래와 같은 구조로 프로젝트를 생성하고, 아래와 같이 동작될 수 있도록 코드를 작성하시오.

<web.xml 코드>

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
  id="WebApp_IC" version="3.0">
  <display-name>login</display-name>

  <servlet>
    [redacted]
  </servlet>
  <servlet-mapping>
    [redacted]
  </servlet-mapping>

  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

- 제출 방식 : E-Class를 통하여 제출
- 제출 내용 : 프로젝트 파일을 압축한 war 파일
 - 각 문제 별로 프로젝트 파일을 생성할 것
 - 각 프로젝트 명은 Q + 문제 번호로 할 것 (ex. 1번 문제의 경우 Q1)
- 제출 형식 : ZIP File(만들어지는 war 파일들을 압축)
- 제목 형식 : 학번_이름_주차
 - Ex) 학번_홍길동_6주차.zip
- 제출 기한 : 10월 9일(토) 23시 59분
 - 기한 엄수 지각 제출 불허