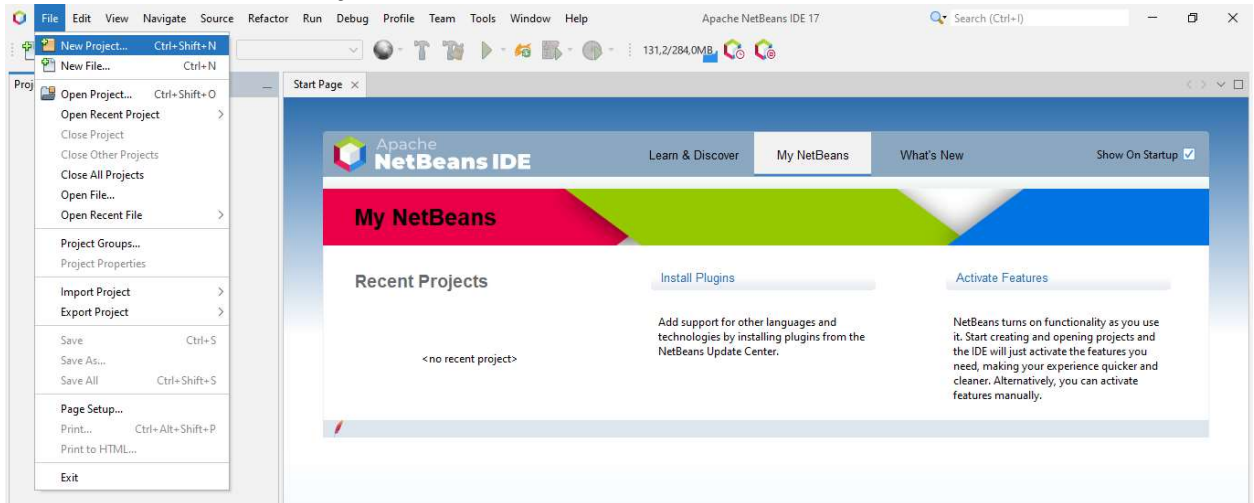


SERVLET TRONG JAVA

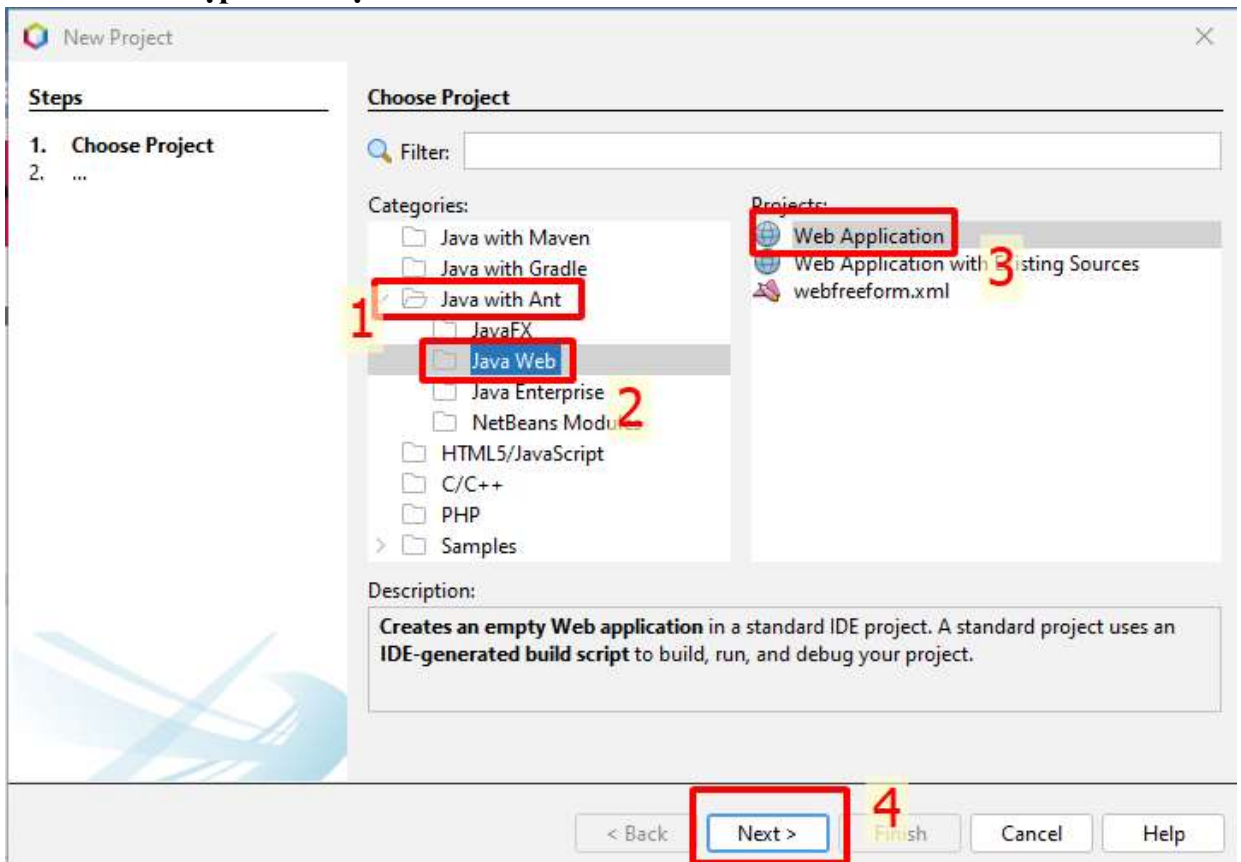
1. Tạo dự án trong Apache NET Bean

Vào File, chọn New Project:



Bước 1: Thiết lập Choose Project

Cửa sổ thiết lập xuất hiện:



Bước 2: Thiết lập Name and Location

Đặt tên cho Project và đường dẫn chứa Project ở bước tiếp theo:

Nên chủ động tạo đường dẫn ở mục Browse để quản lý các Project

New Web Application

Steps

1. Choose Project
2. **Name and Location**
3. Server and Settings
4. Frameworks

Name and Location

Project Name: Slot1

Project Location: ry\2.Materials\1.FPTU\17.PRJ301\1.For Lecturer\Demo-SE18103

Project Folder: laterials\1.FPTU\17.PRJ301\1.For Lecturer\Demo-SE18103\Slot1

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

< Back Next > Finish Cancel Help

Bước 3: Thiết lập Sever and Settings:

Sửa tên đường dẫn ở mục Context Path sang chữ thường vì sau này sẽ xuất hiện trong đường dẫn trên trình duyệt

New Web Application

Steps

1. Choose Project
2. Name and Location
3. **Server and Settings**
4. Frameworks

Server and Settings

Add to Enterprise Application: <None>

Server: Apache Tomcat or TomEE

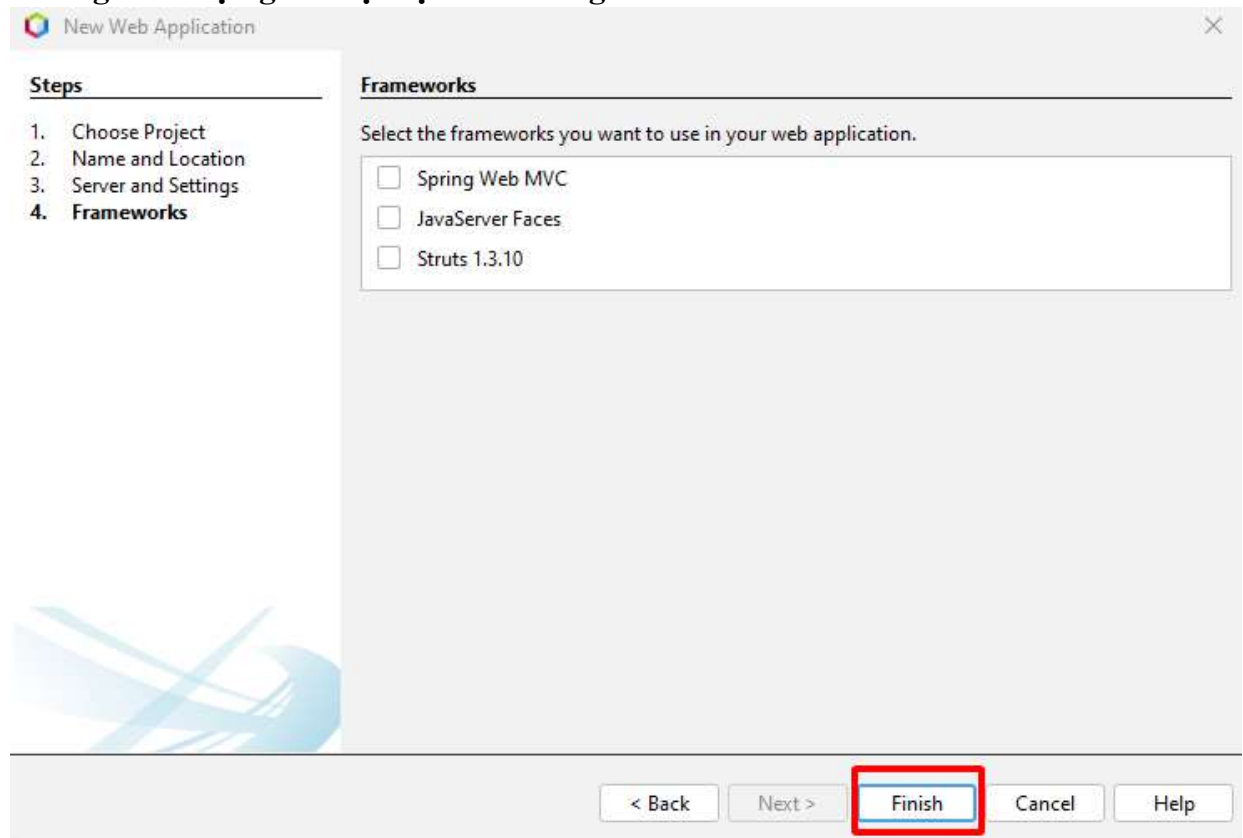
Java EE Version: Jakarta EE 10 Web

Context Path: /slot1

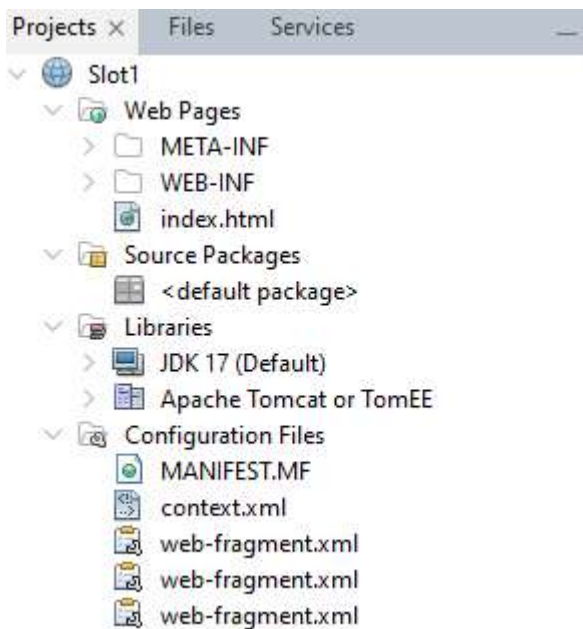
< Back Next > Finish Cancel Help

Bước 4: Thiết lập Frameworks

Không cần chọn gì vì hiện tại chưa dùng Framework



2. Cấu trúc Project trong Apache Net Bean:



Thư mục Web Pages:

Thư mục này chứa tất cả các tài nguyên giao diện web, bao gồm các trang JSP, HTML, CSS, JavaScript, và hình ảnh.

- JSP Pages: Các tệp JSP để xử lý giao diện và logic động.
- HTML Files: Các tệp HTML tĩnh.
- CSS, JavaScript, Images: Các thư mục con chứa tệp CSS, JavaScript và hình ảnh

Trong Web Pages có 2 thư mục con là **META-INF** và **WEB-INF**, trong đó:

+ **WEB-INF**: Thư mục này là nơi chứa các tệp cấu hình và các tài nguyên mà không được truy cập trực tiếp từ trình

duyet.

+ **META-INF** Thư mục này có thể chứa file MANIFEST.MF, cung cấp thông tin về metadata của ứng dụng.

- **web.xml**: File cấu hình của ứng dụng web, nơi bạn định nghĩa các servlet, bộ lọc, và các mapping.
- **Lib**: Thư mục con chứa các thư viện JAR mà ứng dụng cần.

Source Packages: Đây là nơi chứa mã nguồn của ứng dụng Java.

- Package chứa Servlet và Java Classes: Các lớp Java, bao gồm các Servlet, Beans và các logic nghiệp vụ khác của ứng dụng.

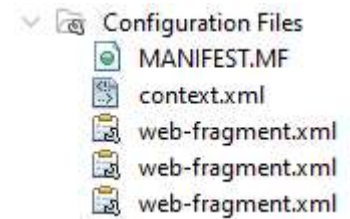
Libraries

Đây là thư mục chứa các thư viện JAR mà dự án sử dụng. Thư viện này bao gồm các thư viện được thêm vào dự án từ bên ngoài, chẳng hạn như JDBC driver hoặc các thư viện phụ trợ.

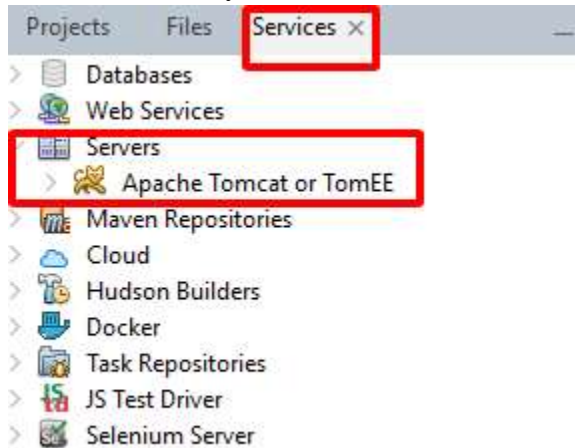


Configuration Files

Thư mục này chứa các file cấu hình khác cho dự án, ví dụ như các file cấu hình cho framework (Spring, Hibernate) hoặc các tệp cấu hình tài nguyên như context.xml cho cơ sở dữ liệu.



3. Kiểm tra cài đặt Server:



Mở vào tab Services, chọn Servers, nếu đã cài Apache Tomcat thì sẽ xuất hiện như hình

SAU KHI TẠO PROJECT XONG, bước tiếp theo:

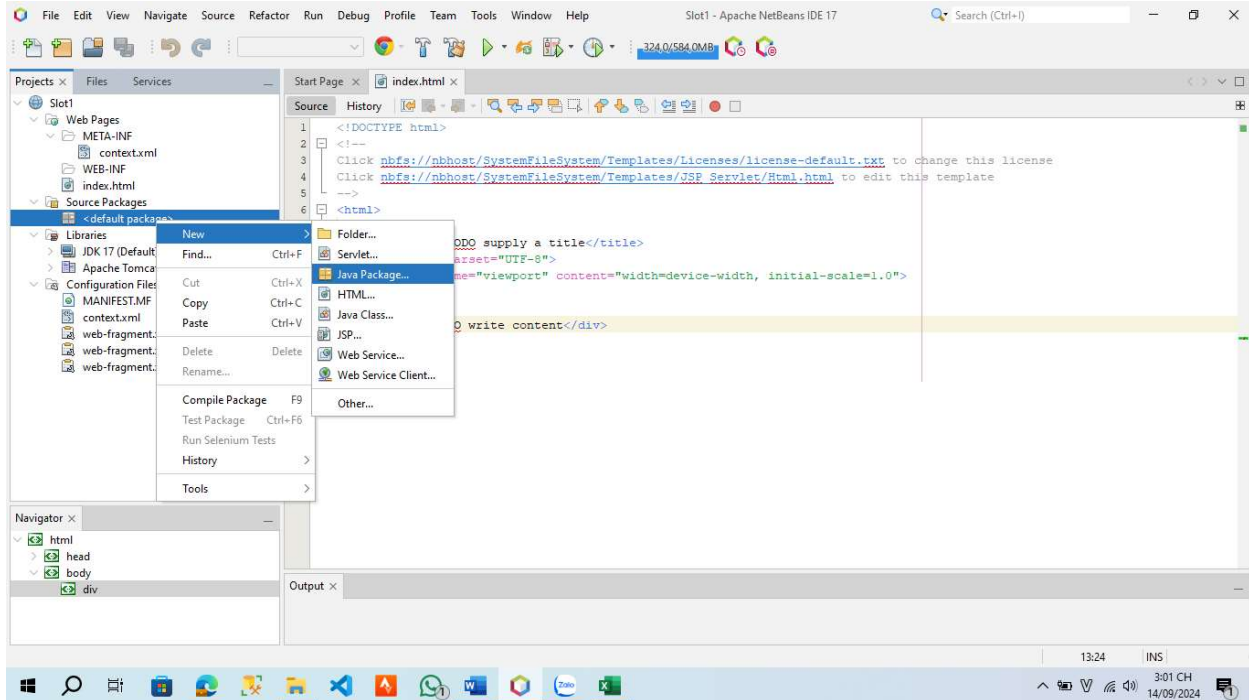
- ✓ *Xây dựng giao diện (thiết kế các form, xây dựng website theo yêu cầu của từng bài)*
- ✓ *Tạo các servlet để xử lý logic ở phía back-end.*

Ở đây sẽ trình bày chi tiết phân tạo servlet, còn phần giao diện và nội dung servlet thì sẽ tùy thuộc yêu cầu từng bài

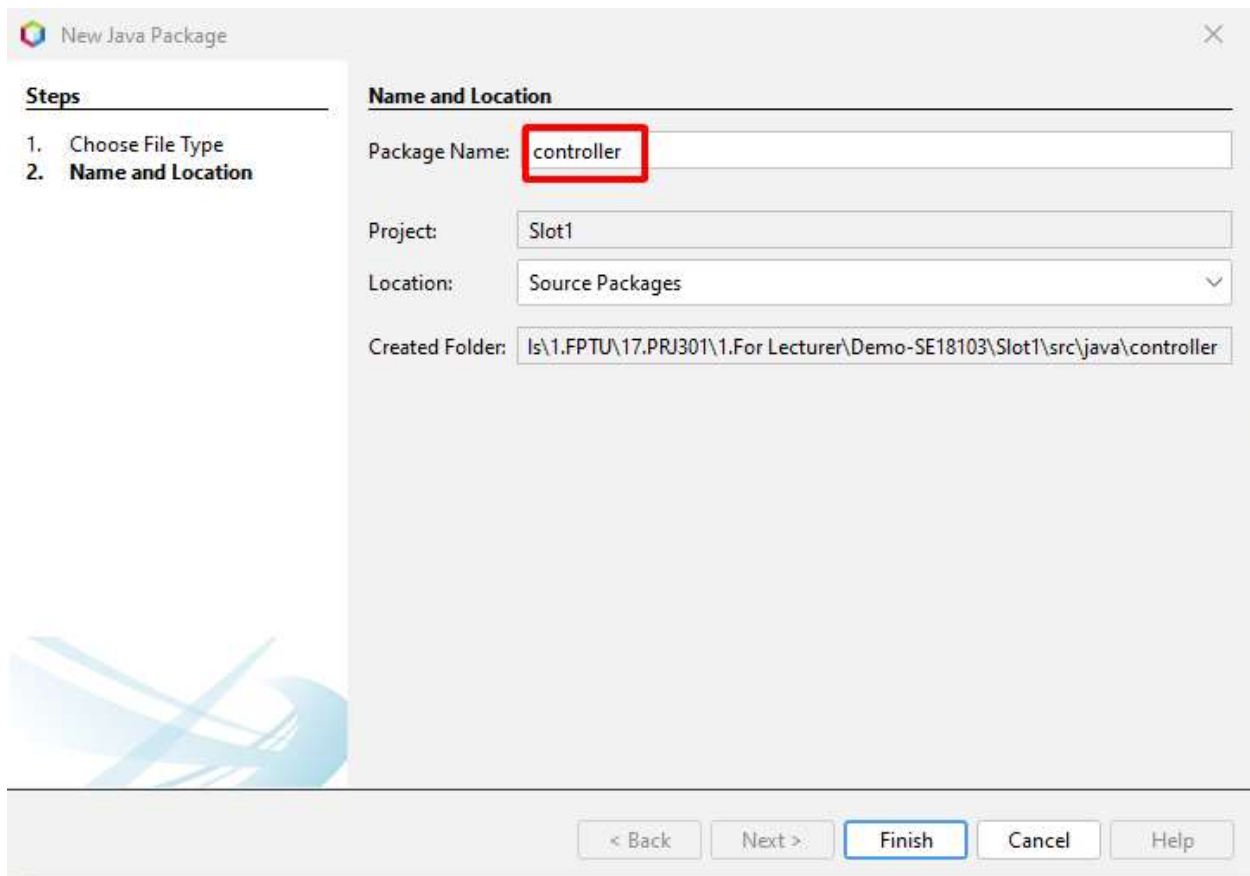
TẠO SERVLET

Tạo Package (nếu Project) chưa có Package:

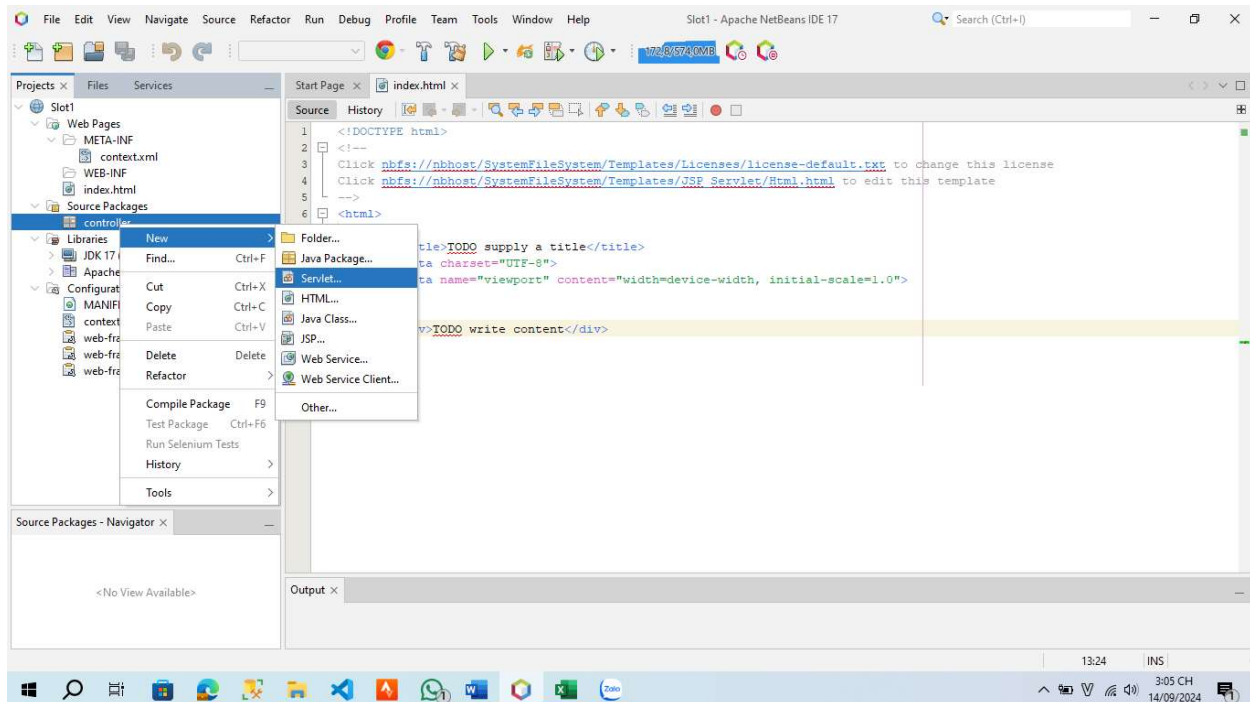
Click chuột phải vào Default Package, chọn New, Chọn Java Package...



Đặt tên cho Package ở hộp thoại tiếp theo, sau đó click Finish:



Click chuột phải vào Package vừa tạo, chọn New, chọn Servlet...



Thiết lập cho Servlet:

+ Đặt tên file Java chứa Servlet

+ chọn đường dẫn:

+ Tên Package chứa Servlet:

New Servlet

Steps

1. Choose File Type
2. **Name and Location**
3. Configure Servlet Deployment

Name and Location

Class Name: ChuviServlet

Project: Slot1

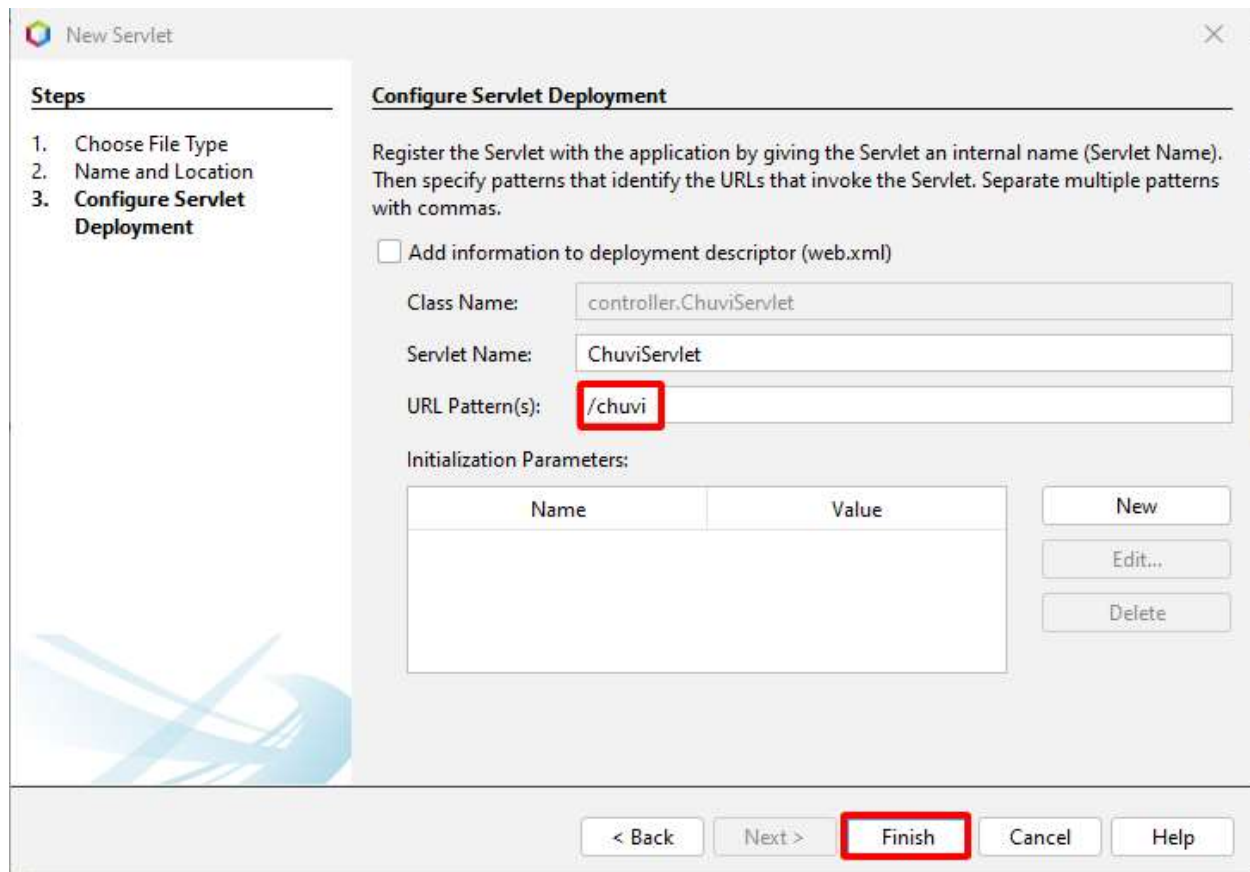
Location: Source Packages

Package: controller

Created File: J301\1.For Lecturer\Demo-SE18103\Slot1\src\java\controller\ChuviServlet.java

< Back Next > Finish Cancel Help

Click Next:



New Servlet

Steps

1. Choose File Type
2. Name and Location
3. **Configure Servlet Deployment**

Configure Servlet Deployment

Register the Servlet with the application by giving the Servlet an internal name (Servlet Name). Then specify patterns that identify the URLs that invoke the Servlet. Separate multiple patterns with commas.

☐ Add information to deployment descriptor (web.xml)

Class Name:

Servlet Name:

URL Pattern(s):

Initialization Parameters:

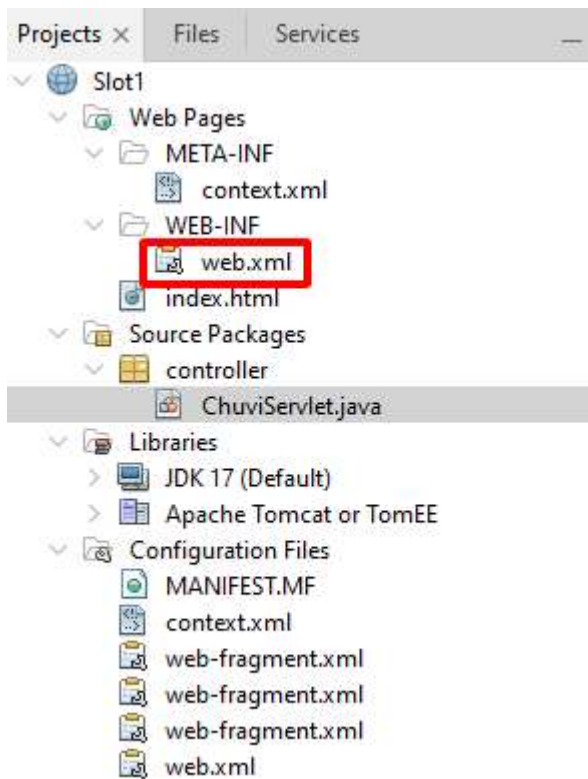
Name	Value
------	-------

New Edit... Delete

< Back Next > **Finish** Cancel Help

Ở bước Configure Servlet Deployment này có 2 lựa chọn:

Nếu tích vào mục **Add information to deployment descriptor (web.xml)** thì thông tin của Servlet sẽ được add vào file web.xml (nếu là lần đầu tiên tạo Servlet cho project và tích chọn mục này thì file web.xml sẽ được sinh ra, các lần tiếp theo tạo Servlet cho project này và tiếp tục chọn mục này thì thông tin của Servlet đó sẽ được add thêm vào file web.xml)



Nếu không tích vào mục **Add information to deployment descriptor (web.xml)** thì thông tin của servlet sẽ xuất hiện trong file Java.

Ở đây mình tạo thêm 1 servlet có tên là ChuviDientichServlet và không tích vào Add infor... Thì thông tin của servlet này xuất hiện trong file Java:



Các mục còn lại:

Server Name: để nguyên

URL Pattern: nên đổi thành chữ thường hết (vì sẽ xuất hiện trên thanh địa chỉ của page)

Ngoài ra còn có cách tạo Servlet khác (chỉ tham khảo cách này):

New 1 file java, ví dụ MyServlet.java, kế thừa từ HttpServlet (extends HttpServlet)

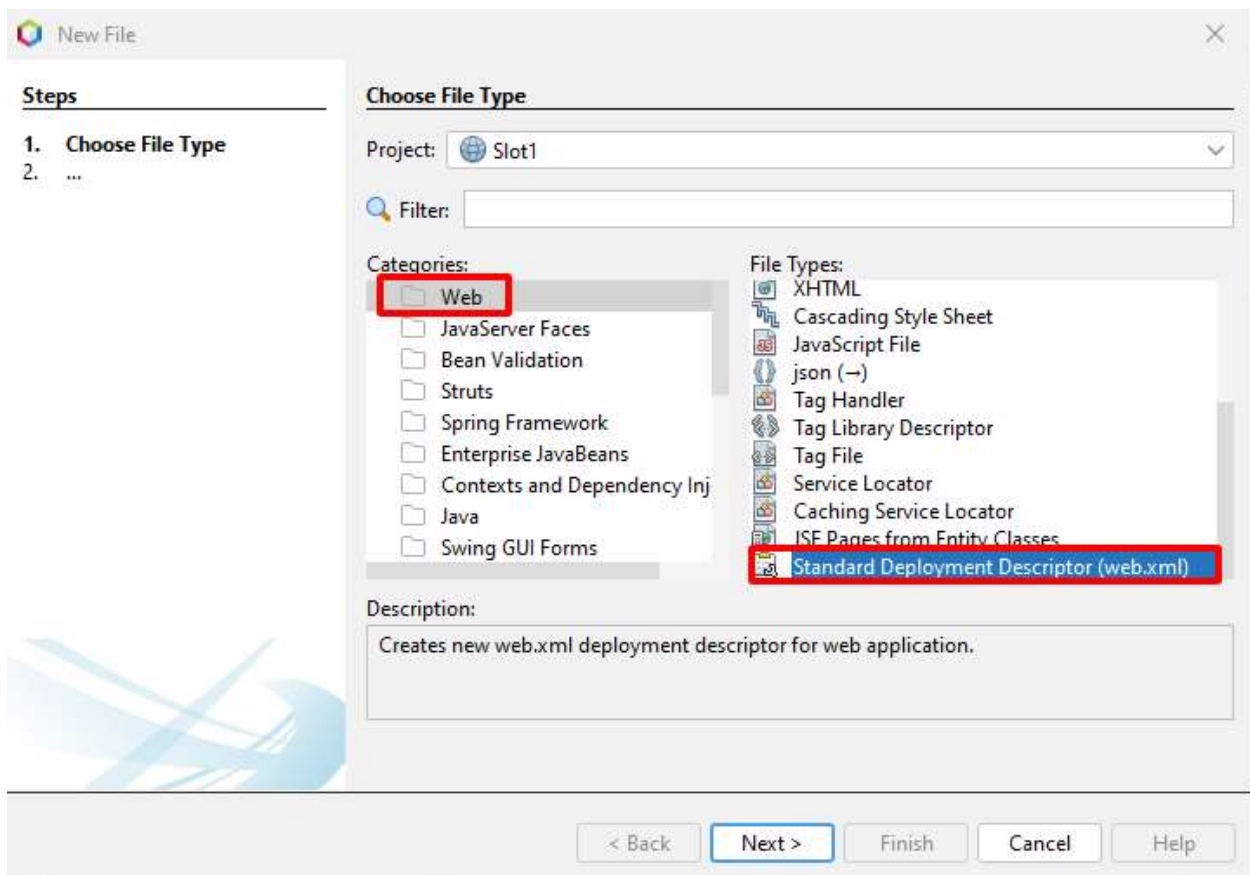
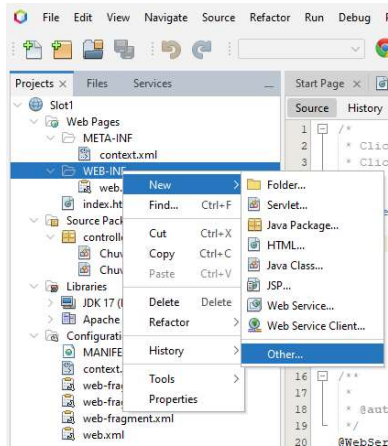
Tạo 2 phương thức ghi đề doGet và doPost

```
public class MyServlet extends HttpServlet{  
    @Override  
        protected void doPost(HttpServletRequest req, HttpServletResponse resp)  
        throws ServletException, IOException {  
  
    }  
  
    @Override  
        protected void doGet(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
  
        PrintWriter out=response.getWriter();  
        out.println("Hello!!!");  
  
    }  
}
```

Sau đó thêm mô tả vào file đó:

```
@WebServlet(name = "MyServlet",urlPatterns = {"/test"})
```

Hoặc tạo file web.xml bằng cách: click chuột phải vào WEB-INF, chọn New, chọn Other....



Sau khi tạo xong file thì viết thêm nội dung vào file web.xml vừa tạo:

```

3  <servlet>
4      <servlet-name>ChuviServlet</servlet-name>
5      <servlet-class>controller.ChuviServlet</servlet-class>
6  </servlet>
7  <servlet-mapping>
8      <servlet-name>ChuviServlet</servlet-name>
9      <url-pattern>/chuvi</url-pattern>
10 </servlet-mapping>

```

Vậy là tổng hợp lại có 3 cách tạo servlet cho 1 project như trên.

Cấu trúc của 1 file Java Servlet lúc khởi tạo sẽ gồm các phần:

- Khai báo Package
- Khai báo các thư viện

```
package controller;

import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
```

- Khai báo lớp servlet được kế thừa từ lớp HttpServlet

```
public class ChuviServlet extends HttpServlet
```

- Trong thân lớp sẽ có 3 hàm:

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    try (PrintWriter out = response.getWriter()) {
        /* TODO output your page here. You may use following sample code. */
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet ChuviServlet</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Servlet ChuviServlet at " + request.getContextPath () + "</h1>");
        out.println("</body>");
        out.println("</html>");
    }
}
```

Hàm doGet xử lý các request có phương thức là get (nó chính là các url chứa tham số) - mặc định

```
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException { ...3 lines }

/**
 * Handles the HTTP <code>POST</code> method.
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
```

Hàm doPost: để xử lý các request có phương thức là post (nó chính là request đến từ form)

```
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 * @return a String containing servlet description
 */
```

Exercise:

Bài 1: Tạo servlet để thực hiện xử lý form sau:

Example Tính chu vi của hình chữ nhật

Nhập vào chiều dài:	<input type="text" value="Nhập chiều dài"/>
Nhập vào chiều rộng:	<input type="text" value="Nhập chiều dài"/>
<input type="button" value="Kết quả"/>	

Thực hiện theo các bước sau:

Tạo form trong file index.html

Viết code xử lý form trong file servlet:

Chạy chương trình

Tạo form trong file index.html:

```
<div> <h1> Example Tính chu vi của hình chữ nhật</h1></div>
<form action="chuvi" method="post">
  <table>
    <tr>
      <td> Nhập vào chiều dài: </td>
      <td><input type="text" name="dai" required placeholder="Nhập chiều dài"/></td>
    </tr>
    <tr>
      <td> Nhập vào chiều rộng: </td>
      <td><input type="text" name="rong" required placeholder="Nhập chiều rộng"/></td>
    </tr>
    <tr>
      <td>
        <input type="submit" value="Kết quả"/>
      </td>
    </tr>
  </table>
</form>
```

Chú ý thiết lập các attributes của thẻ form:

action: tên servlet xử lý form

method: post

Viết code xử lý form trong file servlet:


```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    //processRequest(request, response);
    String dai=request.getParameter(string: "dai");
    String rong=request.getParameter(string: "rong");
    double a, b;
    try{
        a=Double.parseDouble(s: dai);
        b=Double.parseDouble(s: rong);
        response.getWriter().println("<hl> Chu vi: "+ chuvi(x: a, y: b) + "</hl>");
    }
    catch(Exception e){
        response.getWriter().println();
    }
}

private String chuvi(double x, double y){
    return " "+2*(x+y);
}
```

Khai báo biến dai, rong lấy giá trị từ form yêu cầu: *request*

Khai báo biến a, b để chứa giá trị của dai và rong sau khi đổi sang kiểu double.

Viết hàm chuvi để tính chu vi.

Hiển thị ra ngoài trình duyệt: *response.getWriter().println()*

Phân biệt System.out.println() và response.getWriter().println()

Chúng đều là các phương thức dùng để in ra kết quả, nhưng chúng có sự khác biệt lớn về cách thức và mục đích hoạt động trong ứng dụng Java, đặc biệt là khi làm việc với các ứng dụng web.

1. System.out.println():

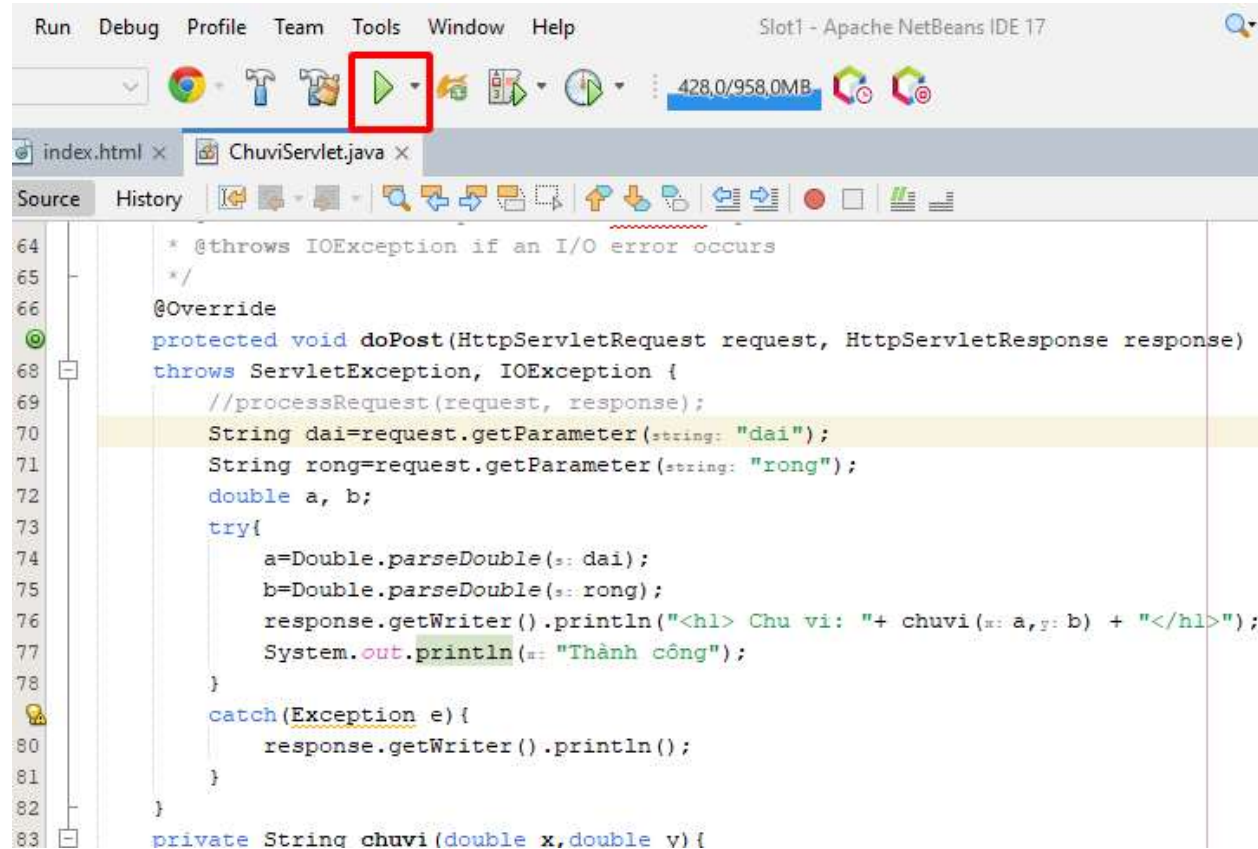
- **Mục đích:** Được sử dụng để ghi dữ liệu ra **console** hoặc **log**. Đây là phương thức tiêu chuẩn để in ra màn hình console trong các ứng dụng Java nói chung.
- **Phạm vi sử dụng:**
 - Dùng trong các ứng dụng Java thuần (ứng dụng console).
 - Trong các ứng dụng web (ví dụ: Servlet), nó ghi thông tin ra console hoặc log server (ví dụ: Tomcat log).
- **Kết quả:** Thông tin in ra **không** được gửi đến trình duyệt của người dùng, mà chỉ xuất hiện trong console của máy chủ hoặc trong file log của ứng dụng.

2. response.getWriter().println():

- **Mục đích:** Được sử dụng để ghi dữ liệu **trả về** cho **trình duyệt** của người dùng, là kết quả HTTP Response.
- **Phạm vi sử dụng:**

- Dùng trong các ứng dụng web, đặc biệt là khi bạn muốn gửi phản hồi từ Servlet đến trình duyệt.
- Kết quả được gửi về phía trình duyệt và hiển thị cho người dùng hoặc được xử lý bởi JavaScript trên trang.
- **Kết quả:** Nội dung in ra sẽ được gửi về **trình duyệt của người dùng** và hiển thị trên trang web, hoặc trở thành một phần của phản hồi HTTP.

Sau khi viết xong thì chạy Chương trình:



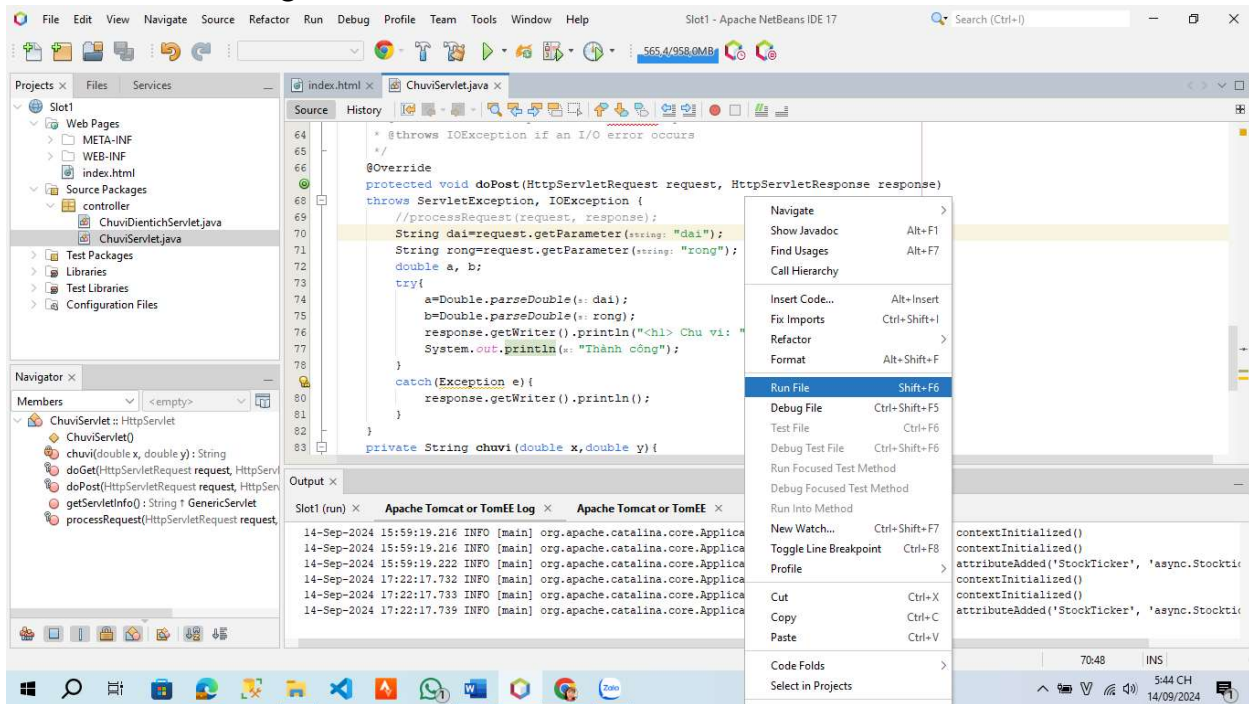
```

64      * @throws IOException if an I/O error occurs
65      */
66      @Override
67      protected void doPost(HttpServletRequest request, HttpServletResponse response)
68          throws ServletException, IOException {
69          //processRequest(request, response);
70          String dai=request.getParameter("dai");
71          String rong=request.getParameter("rong");
72          double a, b;
73          try{
74              a=Double.parseDouble(dai);
75              b=Double.parseDouble(rong);
76              response.getWriter().println("<hl> Chu vi: "+ chuvi(a, b) + "</hl>");
77              System.out.println("Thành công");
78          }
79          catch(Exception e){
80              response.getWriter().println();
81          }
82      }
83      private String chuvi(double x,double v){

```

Để chạy Project:

Nhấn vào biểu tượng Run hoặc nhấn F6



Bài tập 2: Viết servlet xử lý form sau:

Example Tính chu vi và diện tích của hình chữ nhật

Nhập vào chiều dài:

Nhập vào chiều rộng:

Tính Chu vi: ☐ Tính Diện tích: ☐

Tạo form:

```

<body>
<div> <h1> Example Tính chu vi và diện tích của hình chữ nhật</h1></div>
<form action="chuvidientich" method="post">
  <table>
    <tr>
      <td> Nhập vào chiều dài: </td>
      <td><input type="text" name="dai" required placeholder="Nhập chiều dài"/></td>
    </tr>
    <tr>
      <td> Nhập vào chiều rộng: </td>
      <td><input type="text" name="rong" required placeholder="Nhập chiều rộng"/></td>
    </tr>
    <tr>
      <td>
        Tính Chu vi: <input type="radio" name="chuvi"/>
      </td>
      <td>
        Tính Diện tích: <input type="radio" name="dientich"/>
      </td>
    </tr>
    <tr>
      <td>
        <input type="submit" value="Kết quả"/>
      </td>
    </tr>
  </table>
</form>
</body>

```

Chú ý: **action** của form bây giờ được set cho servlet **chuvidientich**

Viết servlet:

Để hiển thị nội dung tiếng Việt trong servlet lên trình duyệt, sử dụng **setContentType** và **setCharacterEncoding** để thiết lập mã hóa ký tự UTF-8 cho phản hồi từ Servlet.

```

protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
  //processRequest(request, response);
  response.setContentType("text/html;charset=UTF-8");
  response.setCharacterEncoding("UTF-8");
  String dai=request.getParameter("dai");
  String rong=request.getParameter("rong");
  String chuvi=request.getParameter("chuvi");
  String dientich=request.getParameter("dientich");
  double a, b;
  try{
    a=Double.parseDouble(dai);
    b=Double.parseDouble(rong);
    if(chuvi!=null && dientich==null){
      response.getWriter().println("<h1> Chu vi: " + chuvi(x: a, y: b) + "</h1>");
    }
    if(chuvi==null && dientich!=null){
      response.getWriter().println("<h1> Diện tích: " + dientich(x: a, y: b) + "</h1>");
    }
    if(chuvi!=null && dientich!=null){
      response.getWriter().println("Chu vi: " + chuvi(x: a, y: b) + "<br>" + "Diện tích: " + dientich(x: a, y: b));
    }
  }
  catch(Exception e){
    response.getWriter().println();
  }
}

```

Viết 2 hàm chuvi và dientich:

```

private String chuvi(double x, double y) {
    return " "+2*(x+y);
}
private String dientich(double x, double y) {
    return " "+(x*y);
}

```

Bài 3:

Viết servlet xử lý form đăng nhập:



Login form

User name:

Password:

Code tạo form:

```

<div><h1> Login form</h1></div>
<form name="input" action="login" method="post">
<table>
    <tr>
        <td>User name:</td><td><input type="text" name="user" required="" placeholder="Nhập username"/></td>
    </tr>
    <tr>
        <td>Password:</td><td><input type="password" name="pass" required="" placeholder="Nhập pass"/></td>
    </tr>
    <tr>
        <td></td><td><input type="submit" value="Login"/></td>
    </tr>
</table>
</form>

```

Viết servlet có 2 cách:

- + Cách 1: là so sánh giá trị người dùng nhập vào với giá trị cài đặt trong servlet:
- + Cách 2: là so sánh giá trị người dùng nhập vào với giá trị được thiết lập trong web.xml

Servlet của cách 1:

```

protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    //processRequest(request, response);
    response.setContentType(string: "text/html;charset=UTF-8");
    response.setCharacterEncoding(string: "UTF-8");
    String name=request.getParameter(string: "user");
    String pass=request.getParameter(string: "pass");

    if(name.equals(anObject: "thuybt26") && pass.equals(anObject: "123@") ){
        response.getWriter().println("<hl> Xin chao ban "+name+"</hl>");
    }
    else{
        response.getWriter().println(x: "<hl> Nhập lại</hl>");
    }
}
}

```

Servlet cách 2:

Mở file web.xml, thẻ Source, tạo thêm 2 cặp thẻ như hình:



Viết servlet:


```

protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    //processRequest(request, response);
    response.setContentType("text/html;charset=UTF-8");
    response.setCharacterEncoding("UTF-8");
    String name=request.getParameter("user");
    String pass=request.getParameter("pass");
    String username=getServletContext().getInitParameter("username");
    String password=getServletContext().getInitParameter("password");
    if(name.equals(username) && pass.equals(password) ){
        response.getWriter().println("<h1> Xin chào bạn "+name+"</h1>");
    }
    else{
        response.getWriter().println("<h1> Nhập lại</h1>");
    }
}
}

```

Bài 4:

Viết servlet xử lý form thông tin sinh viên:

The screenshot shows a web browser window with the address bar displaying "localhost:8080/slot3/". The page title is "Thông tin sinh viên". The form contains the following fields and controls:

- Họ và Tên đệm:
- Tên:
- Giới tính: ☐ Nam ☐ Nữ
- Chọn môn học:
- Nhập Điểm (0 - 10):
-

Tạo form

```

13 <div><h1>Thông tin sinh viên</h1></div>
15 <form id="studentForm" action="student" method="post">
16   <table>
17     <tr>
18       <td><label for="name">Họ và Tên đệm:</label></td>
19       <td><input type="text" id="ho" name="ho" placeholder="Nhập họ và tên đệm" required></td>
20     </tr>
21     <tr>
22       <td><label for="name">Tên:</label></td>
23       <td><input type="text" id="name" name="name" placeholder="Nhập tên" required></td>
24     </tr>
25     <tr>
26       <td><label>Giới tính:</label> </td>
27       <td> Nam <input type="radio" id="male" name="gender" value="Nam" required></td>
28       <td> Nữ <input type="radio" id="female" name="gender" value="Nữ" required></td>
29     </tr>
30     <tr>
31       <td><label for="subject">Chọn môn học:</label></td>
32       <td>
33         <select id="subject" name="subject" required>
34           <option value="">-- Chọn Môn Học --</option>
35           <option value="PRO192">PRO192</option>
36           <option value="PRF192">PRF192</option>
37           <option value="PRJ301">PRJ301</option>
38           <option value="PRJ302">PRJ302</option>
39           <option value="PRN212">PRN212</option>
40           <option value="PRN222">PRN222</option>
41           <option value="PRN232">PRN232</option>
42           <option value="DBI202">DBI202</option>
43           <option value="SWP201">SWP201</option>
44           <option value="NWC204">NWC204</option>
45         </select>
46       </td>
47     </tr>
48     <tr>
49       <td><label for="score">Nhập Điểm (0 - 10):</label></td>
50       <td><input type="number" id="score" name="score" min="0" max="10" step="0.1" required></td>
51     </tr>
52     <tr>
53       <td><button type="submit">Gửi</button> </td>
54       <td></td>
55     </tr>
56   </table>

```

Viết code cho Servlet


```

68 protected void doPost(HttpServletRequest request, HttpServletResponse response)
69     throws ServletException, IOException {
70     response.setContentType("text/html;charset=UTF-8");
71     response.setCharacterEncoding("UTF-8");
72     // Lấy thông tin từ form
73     String ho = request.getParameter("ho");
74     String ten = request.getParameter("name");
75     String monHoc = request.getParameter("subject");
76     String diemStr = request.getParameter("score");
77     // Chuyển đổi điểm sang kiểu số (double)
78     double diem = Double.parseDouble(diemStr);
79     // Đặt kiểu mã hóa để hiển thị tiếng Việt
80     response.setContentType("text/html;charset=UTF-8");
81     response.setCharacterEncoding("UTF-8");
82     // Xây dựng chuỗi kết quả
83     StringBuilder result = new StringBuilder();
84     result.append("<html><body>");
85     // Hiển thị lời chào
86     result.append("<h1>Xin chào ").append(ho).append(" ").append(ten).append("</h1>");
87     // Kiểm tra điểm và hiển thị kết quả phù hợp
88     if (diem >= 5) {
89         result.append("<p>Chúc mừng bạn đã qua môn ").append(monHoc)
90             .append(" với số điểm ").append(d: diem).append("</p>");
91     } else {
92         result.append("<p>Rất tiếc bạn đã phải học lại môn ").append(monHoc)
93             .append(" vì điểm ").append(d: diem).append(" nhỏ hơn 5.</p>");
94     }
95     result.append("</body></html>");
96     // Hiển thị kết quả ra màn hình
97     response.getWriter().println(x: result.toString());
98 }

```

Bài 5: Viết servlet xử lý form sau:

Chương trình máy tính cơ bản

Nhập vào số thứ nhất:

Nhập vào số thứ hai:

+ ☐ - ☐ * ☐ / ☐