# Mục tiêu:

Kết thúc bài thực hành này bạn có khả năng củng cố lại kiến thức

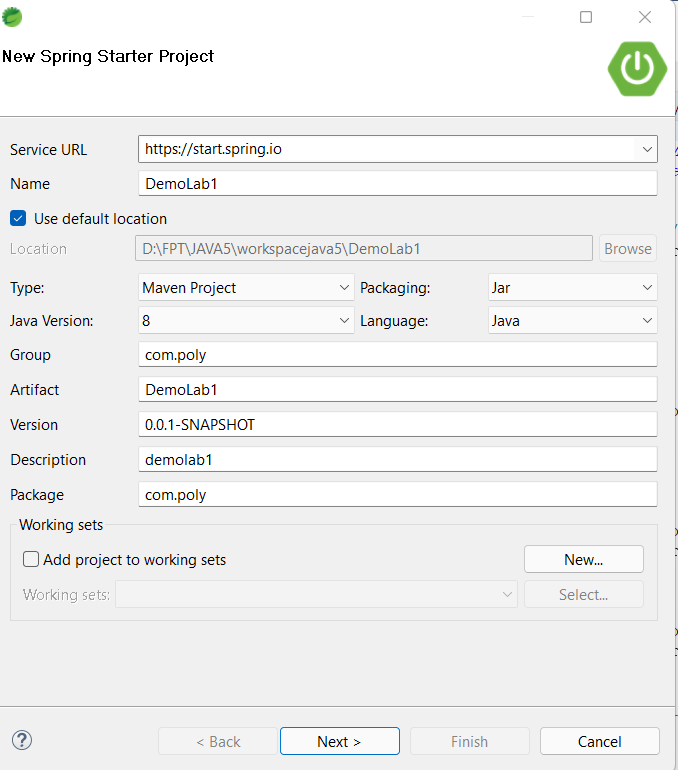
* Map request (Method, URL, Parametters) với phương thức
* Đọc được dữ liệu từ người sử dụng (Parameter, Cookie) bằng nhiều phương pháp khác nhau
* Sử dụng Model để chia sẻ dữ liệu
* Phân biệt kết quả trả về từ mapping method

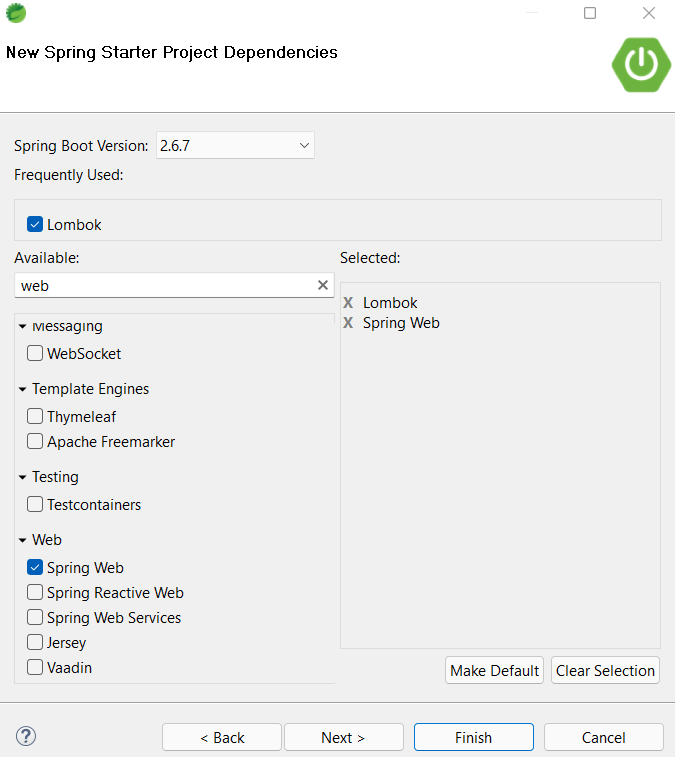
# PHẦN I

## Bài 1 (2 điểm)

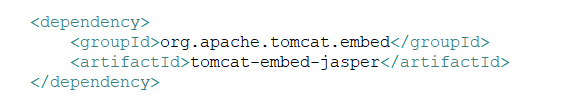
Tạo một project và thêm thư viện

Vào file 🡺 new spring starter project

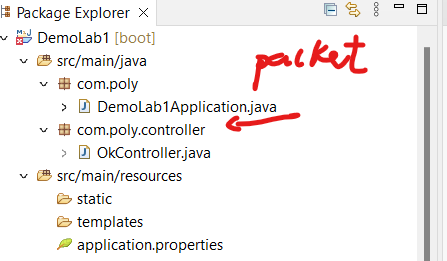




Tạo thêm thư viện để tomcat hiển thị giao diện jsp bỏ vào file pom.xml



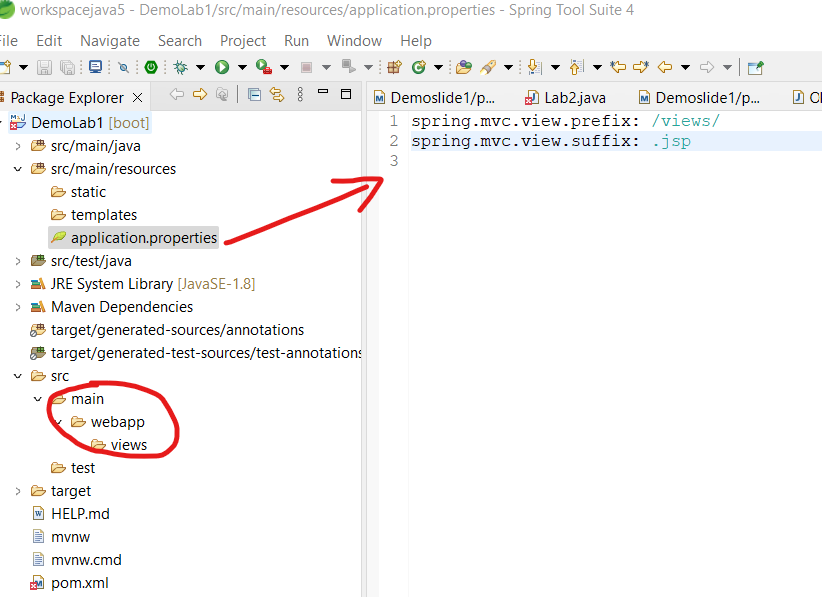
Sau khi tạo xong ta tạo packet



Tạo class:

|  |
| --- |
| @Controller  public class OkController {  @RequestMapping("/ok")  public String ok() {  return "ok"; // GỌI FORM OK.JSP  }  public String **m1**() {  return "ok";  }  public String **m2**() {  return "ok";  }  public String **m3**() {  return "ok";  }  } |

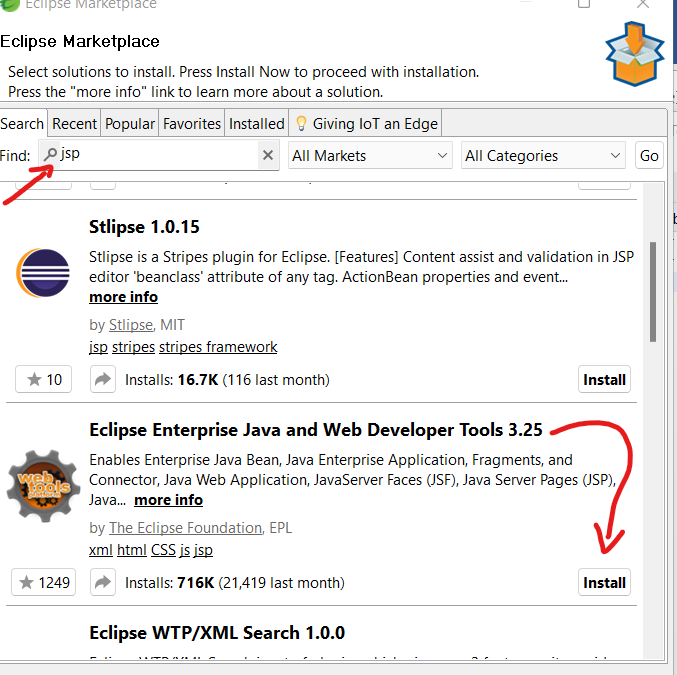
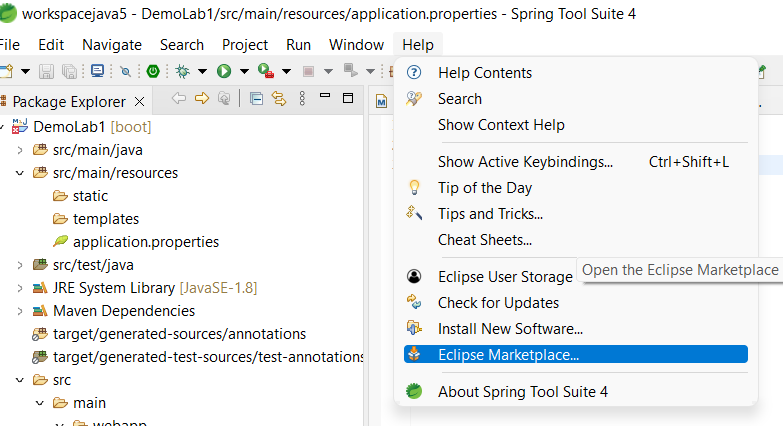
Bước 2: cài đặt tiện ích hỗ trợ tạo các trang jsp như sau:

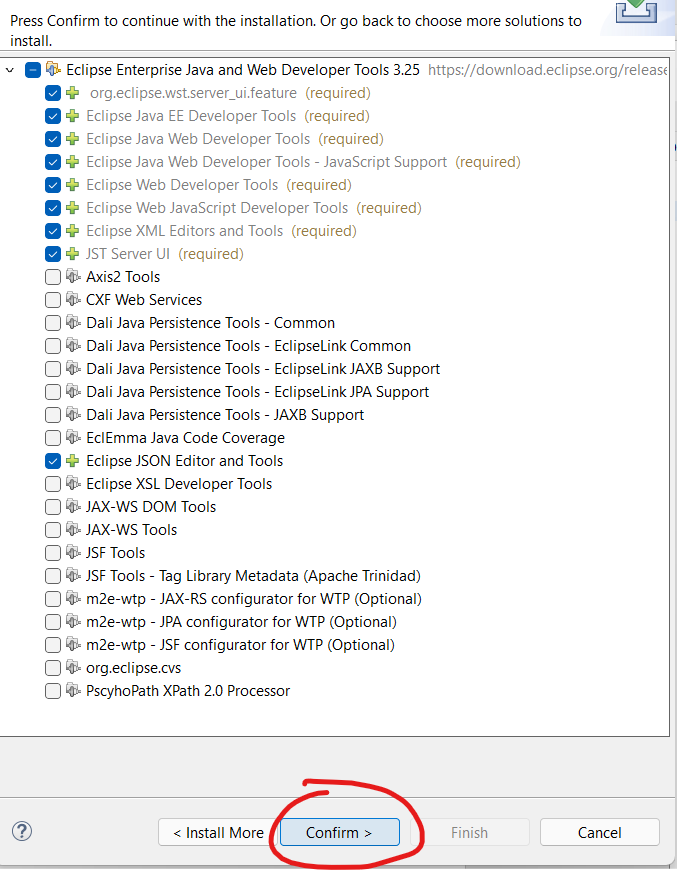


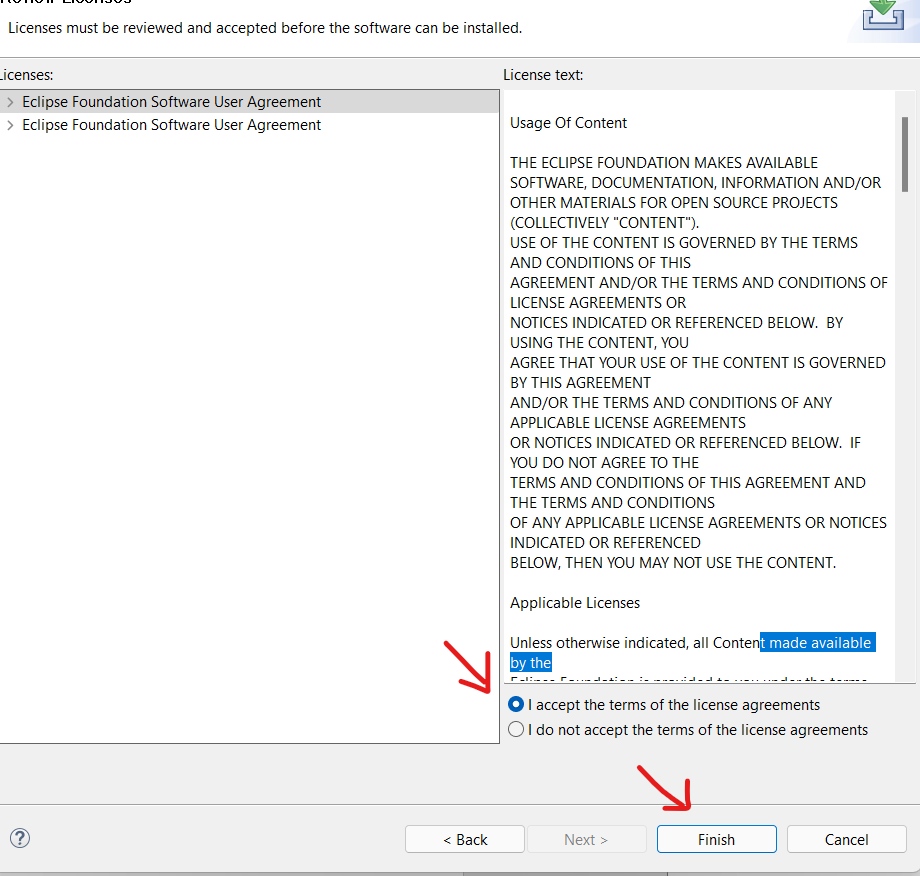
Muốn đổi port thêm dòng

server.port=8089 (ở dòng trên cùng trong file application.properties)

Cài tiện ích để tạo các file .jsp

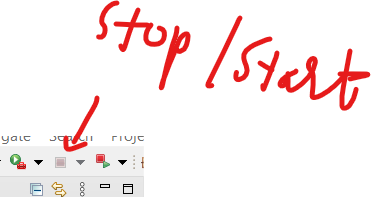




* Chọn cài đặt YES
* 
* Tạo file ok.jsp có chứa form HTML như sau:

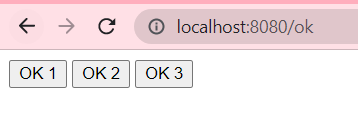
|  |
| --- |
| <form action="/ctrl/ok" method="post">  <button>OK 1</button>  <button formmethod="get">OK 2</button>  <button formaction="/ctrl/ok?x">OK 3</button>  </form>  Kết quả: ? |
|  |
|  |

## Thử chạy và gọi form ok: (nhớ khi chạy lần 2 là phải stop server và chạy lại)

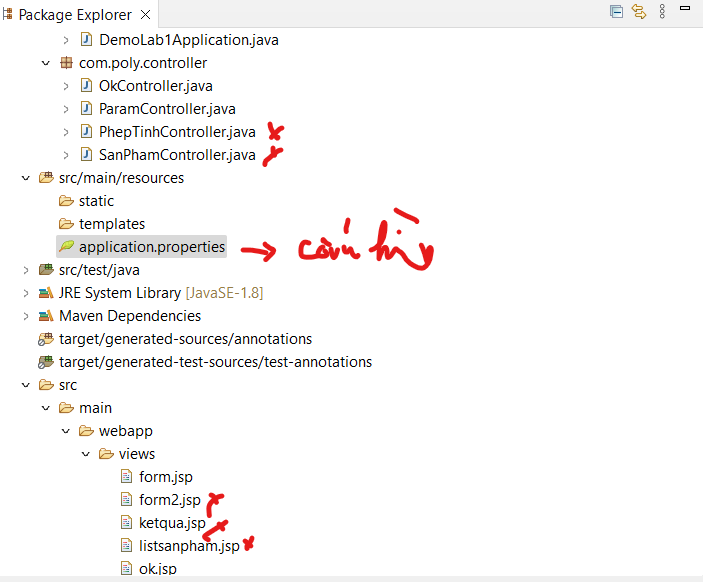


1. TA CLICK CHUỘT PHẢI TRÊN ỨNG DỤNG 🡺 RUN AS 🡺 SPRING BOOT APP.
2. KHI SERVER START XONG TA LÊN TRÌNH DUYỆT GÕ: <http://localhost:8080/ok>

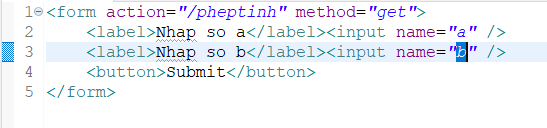
RA HÌNH SAU LÀ THÀNH CÔNG:



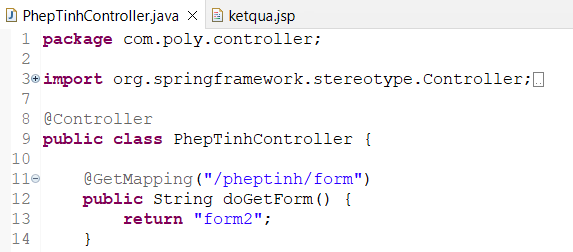
## Bài 2 (2 điểm)



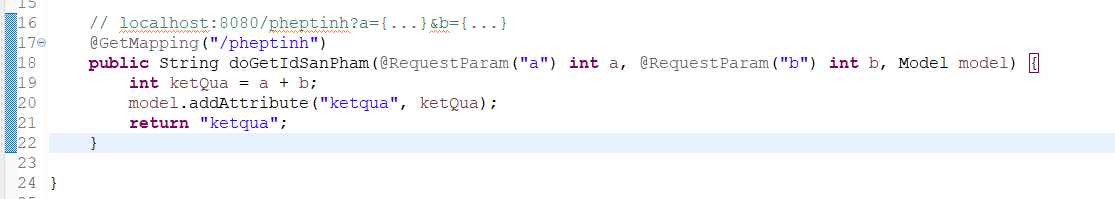
* 1. Cách sử dụng @RequestParam để truyền dữ liệu từ form xuống Controller
* Tạo form phép tính để nhập giá trị a và b:

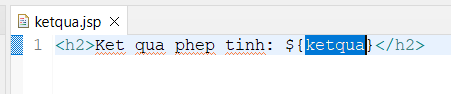


* Tạo PhepTinhController, trong đó có hàm hiển thị form:

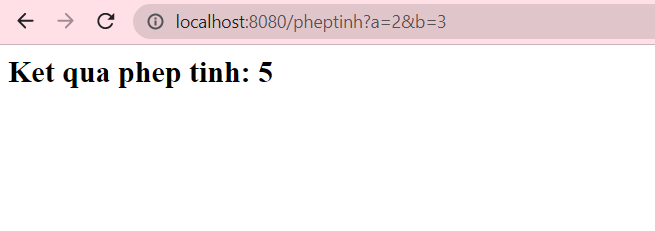


* Tạo tiếp hàm nhận sự kiện xử lý form, dùng @RequestParam để nhận giá trị a và b, thực hiện phép tính và xuất ra trang ketqua.jsp:

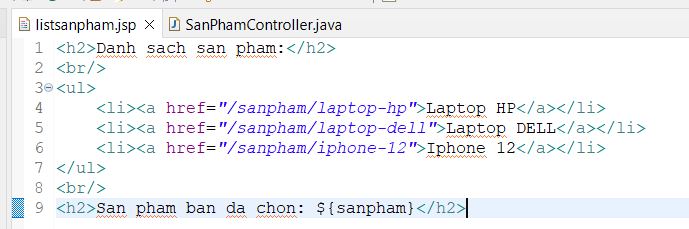




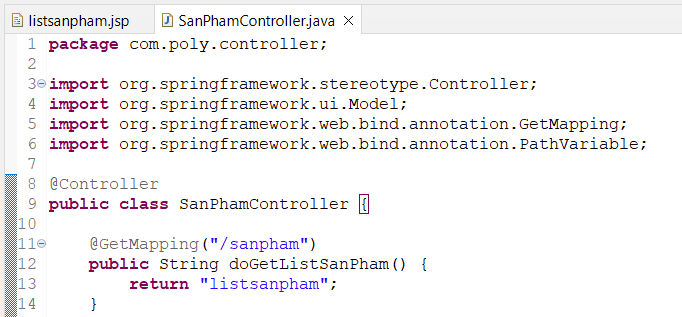
* Kết quả xử lý form sẽ có dạng như bên dưới:



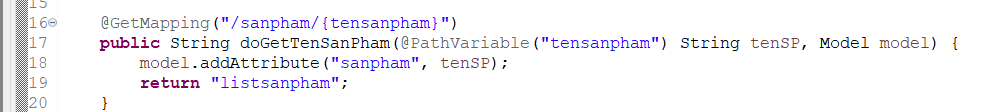
* 1. Cách sử dụng @PathVariable để truyền dữ liệu thông qua URL xuống Controller
* Tạo trang listsanpham.jsp:



* Tạo SanPhamController để hiển thị list sản phẩm:



* Tạo hàm sử dụng @PathVariable để nhận tên sản phẩm gửi đi thông qua URL:

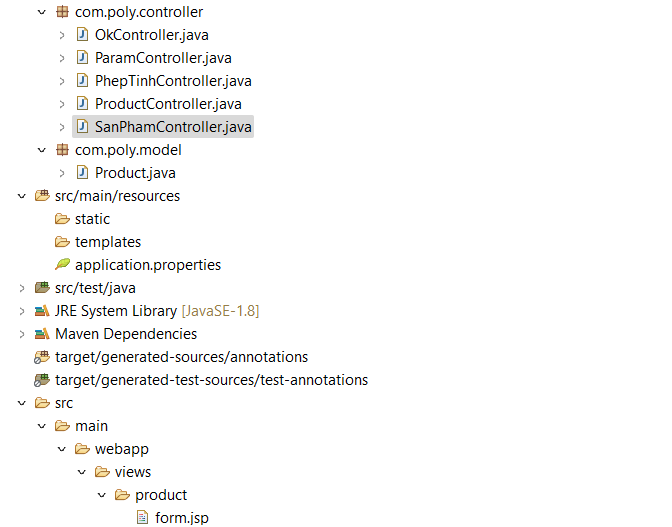


* Kết quả khi click vào các sản phẩm trong list:

Cấu trúc



## Bài 3 (1 điểm)



Tạo class Product với sự hỗ trợ của **Lombok** như sau:

|  |
| --- |
| @Data  @NoArgsConstructor  @AllArgsConstructor  public class Product {  String name;  Double price;  } |

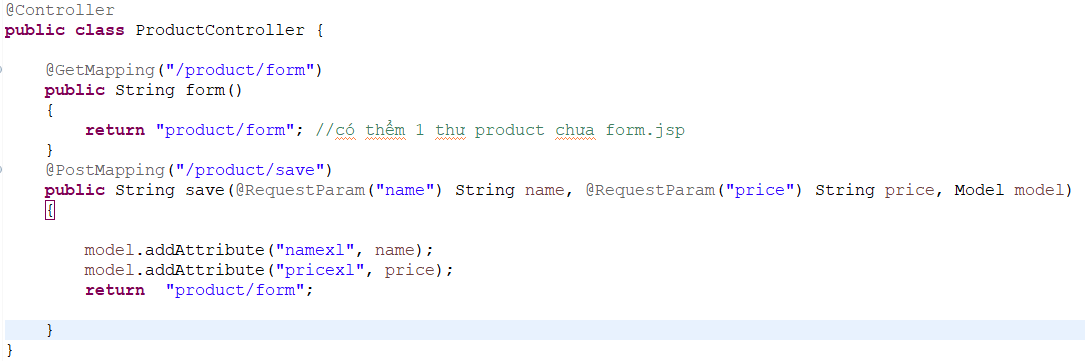
Tạo View form.jsp

|  |
| --- |
| <form action="/product/save" method="post">  <input name="name">  <input name="price">  <button>Save</button>  </form>  <ul>  <li>?name</li>  <li>?price</li>  </ul> |

Tạo ProductController như cấu trúc sau

|  |
| --- |
| @Controller  public class ProductController {  @GetMapping("/product/form")  public String form() {  return "product/form";  }  @PostMapping("/product/save")  public String save(**/\*?1\*/**) {  return "product/form";  }  } |

Hãy viết thêm mã sử dụng Product để nhận dữ liệu form. Sau đó chia sẻ và hiển thị giá trị các thuộc tính name, price của Student tại các vị trí ?name và ?price của form.jsp.



ở trang hiển thị:

# 

# PHẦN II

## Bài 4 (2 điểm)

Tạo ProductController có cấu trúc như sau

|  |
| --- |
| @Controller  public class ProductController {  @GetMapping("/product/form")  public String form() {  Product p = new Product();  p.setName("iPhone 30");  p.setPrice(5000.0);  **/\*?1\*/**  return "product/form";  }  @PostMapping("/product/save")  public String save(**/\*?2\*/** Product p) {  return "product/form";  }  **/\*?3\*/**  public List<Product> getItems() {  return Arrays.asList(new Product(“A”, 1), new Product(“B”, 12);  }  } |

Tạo View form.jsp có mã như sau

|  |
| --- |
| <form action="/product/save" method="post">  <input name="name">  <input name="price">  <button>Save</button>  </form>  <ul>  <li>**?1**name</li>  <li>**?1**price</li>  </ul>  <ul>  <li>**?2**name</li>  <li>**?2**price</li>  </ul>  <ul>  <c:forEach var=”item” items=”${**?3**}”>  <li>item.name</li>  <li>item.price</li>  </c:forEach>  </ul> |

Hãy sử dụng Model, @ModelAttribute bằng cách bổ sung mã vào các vị trí /\*?1\*/, /\*?2\*/ và /\*?3\*/ để chia sẻ dữ liệu controller với giao diện form.jsp. Hãy hiển thị các thuộc tính name, price của các sản phẩm và danh sách sản phẩm tại các vị trí tương ứng trên form.jsp

## Bài 5 (2 điểm)

Tạo ResultController như sau

|  |
| --- |
| @Controller  public class ResultController {  @RequestMapping("/a")  public String m1() {  return "a";  }  @RequestMapping("/b")  public String m2(Model model) {  model.addAttribute("message", "I come from b");  return **"?1"**;  }  @RequestMapping("/c")  public String m3(RedirectAttributes params) {  params.addAttribute("message", "I come from c");  return **"?2"**;  }  **?3**  @RequestMapping("/d")  public String m4() {  return "I come from d";  }  } |

Tạo View a.jsp có mã như sau

|  |
| --- |
| <h1>${message}</h1>  <h1>${param.message}</h1> |

* Hãy viết thêm mã vào các vị trí ?1 để khi request đến /b thì chuyển về /a hiển thị biến message trong Model
* Hãy viết thêm mã vào các vị trí ?2 để khi request đến /c thì chuyển về /a hiển thị biến message trong RedirectAttributes
* Hãy viết thêm mã vào các vị trí ?3 để khi request đến /d thì hiển thị kết quả trả về của m4()

## Bài 6 (1 điểm)

Giảng viên cho thêm