**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 file ok.html 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: ?

Hãy ánh xạ các request được tạo ra khi click vào các button và OK 1, OK 2 và OK 3 với các phương thức tương ứng m1(), m2() và m3() của OkController sau đây. Sau đó hiển thị tên của phương thức được gọi tại vị trí ? của ok.html

@Controller **public** **class** OkController {

@RequestMapping("/ok")

**public** String ok() { **return** "ok";

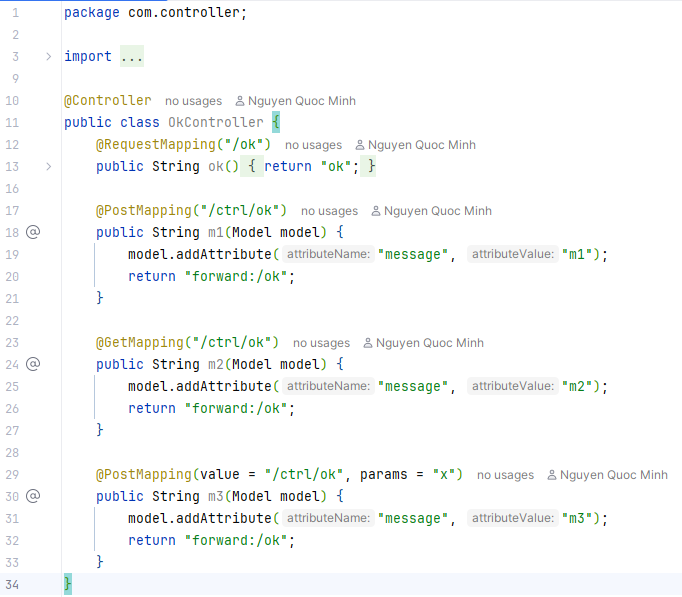
}

**public** String m1() { **return** "ok";

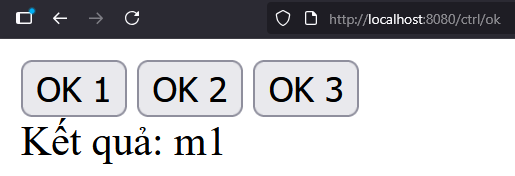
}

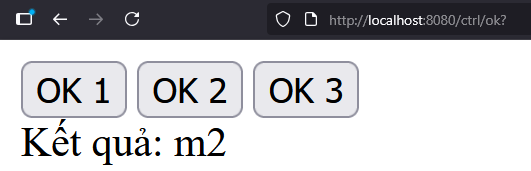
**public** String m2() { **return** "ok";

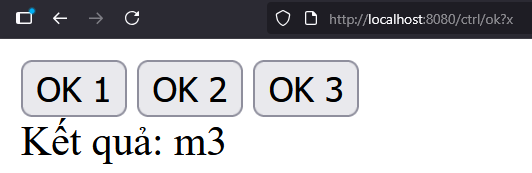
|  |  |
| --- | --- |
| } | }  **public** String m3() {  **return** "ok";  } |











## Bài 2 (2 điểm)

Tạo ParamController có cấu trúc như sau

|  |
| --- |
| @Controller  **public** **class** ParamController {  @RequestMapping("/param/form")  **public** String form() { **return** "form";  }  @RequestMapping("/param/save/{x}")  **public** String save(/\*?1\*/ String x, /\*?2\*/ String y) {  **return** "form";  }  } |

Tạo View form.html chứa mã HTML như sau

<form action="/param/save/2021" method="post">

<input name="y" value="2031">

<button>Save</button>

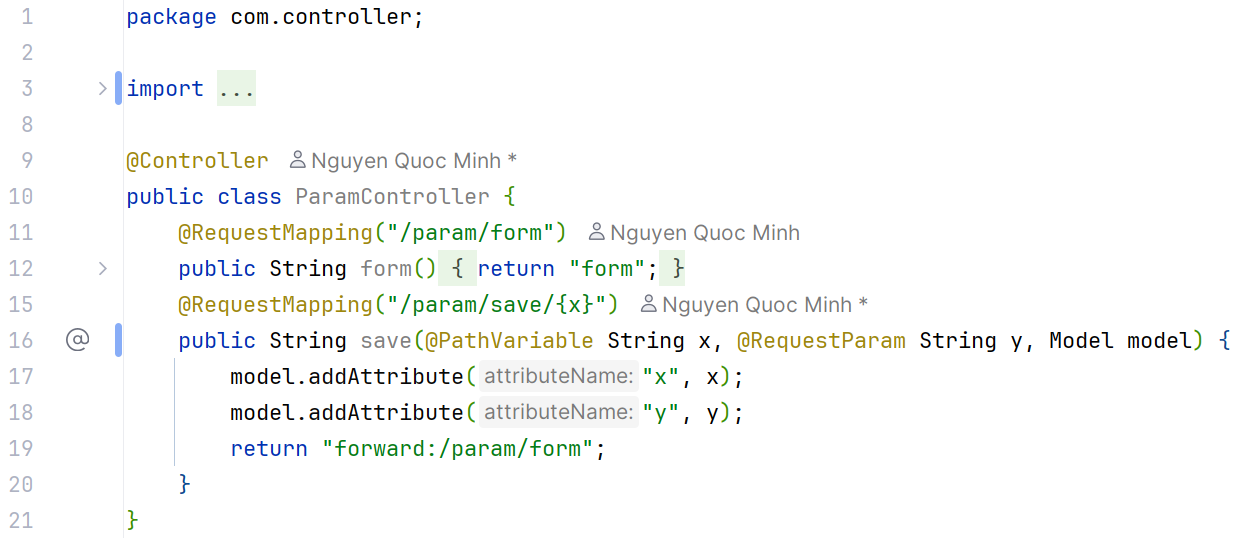
</form>

<ul>

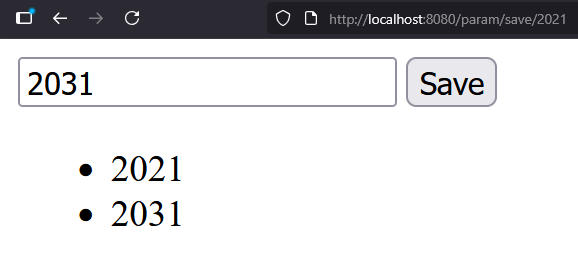
<li>?x</li>

<li>?y</li> </ul>

Sử dụng @RequestParam và @PathVariable thay vào các vị trí /\*?1\*/ và /\*?2\*/ để đọc các giá trị của tham số x, biến đường dẫn y. Sau đó xuất giá trị của y, y tại vị trí ?x và ?y của form.html







## 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.html

<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.html.

Bài 3 đã làm chung bài 4

# 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.html có mã như sau

<form action="/product/save" method="post">

<input name="name">

<input name="price">

<button>Save</button>

</form>

<ul>

<li>?1name</li>

<li>?1price</li>

</ul>

<ul>

<li>?2name</li>

<li>?2price</li>

</ul>

<ul>

<th:block th:each="item: ${?3}">

<li>item.name</li>

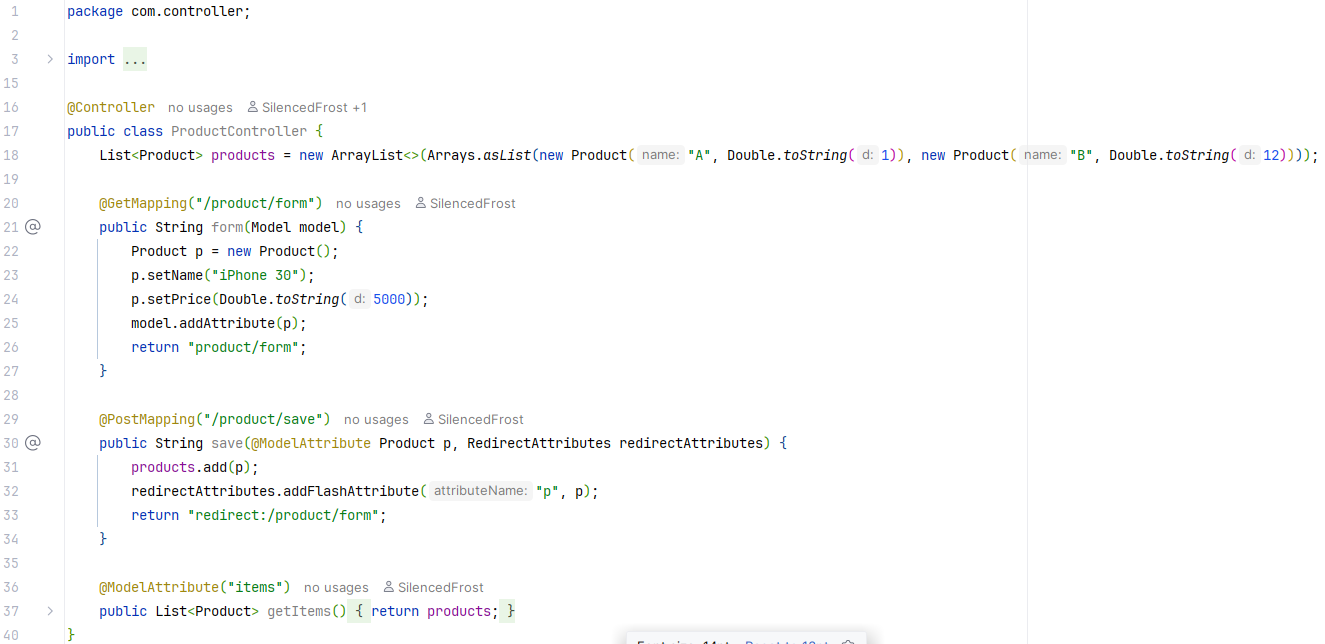
<li>item.price</li>

</th:block> </ul>

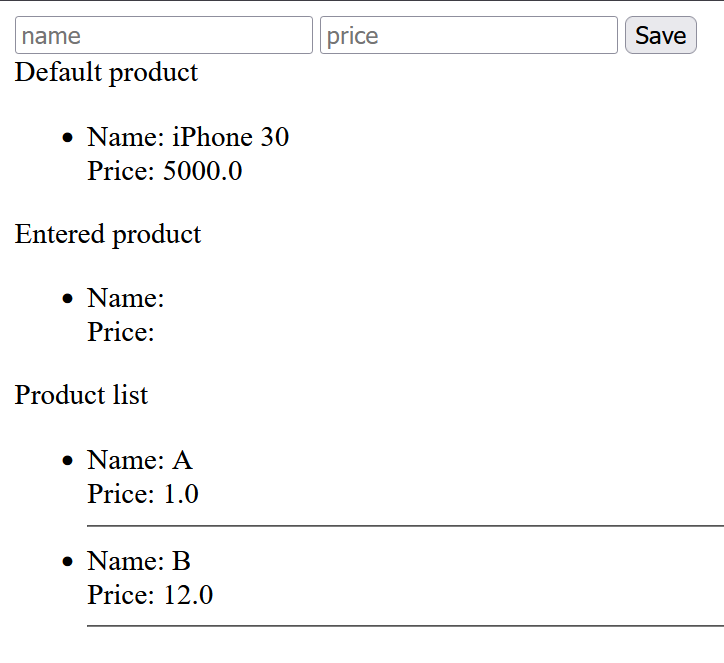
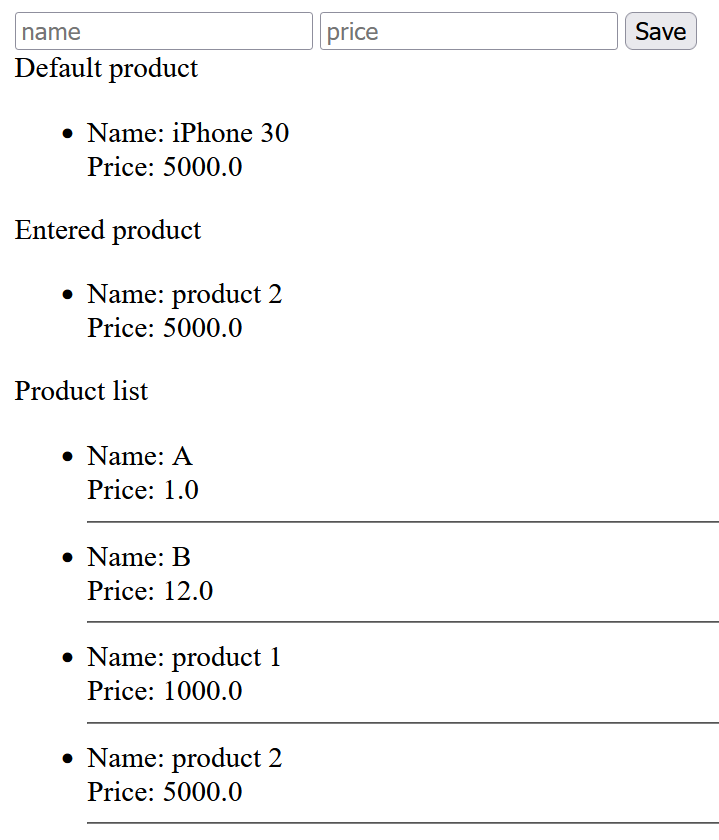
Hãy sử dụng Model, @ModelAttribute bằng cách bổ sung mã vào các vị trí /\*?1\*/, /\*?2\*/ và /\*?3\*/ để chia sẻ dữ liệu controller với giao diện form.html. Hãy hiển thị các thuộc tính name, price của các sản phẩm và danh sách sản phẩm tại các vị trí tương ứng trên form.html

## Bài 6 (1 điểm)

Giảng viên cho thêm việc thêm item mới nhập vào danh sách





## 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.html có mã như sau

<h1 th:text="${message}"></h1>

<h1 th:text="${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()





