

## MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Gọi phương thức action phân biệt theo
  - Phương thức web (POST|GET)
  - Tham số
- ✓ Hiểu các phương pháp nhận dữ liệu từ người dùng
  - Tham số
    - HttpServletRequest
    - @RequestParam
    - JavaBean
  - @PathVariable
  - @CookieValue

## PHẦN I

### Bài 1 (2 điểm)

Bài này hướng dẫn các bạn gọi các phương thức action theo phương thức web POST hay GET. Để hoàn thành bài tập này, các bạn hiệu chỉnh lại mã của bài 3 và bài 4 của Lab1 sẽ rất nhanh.

- ✓ Tạo StudentController có gồm 2 phương thức action như sau

```
@RequestMapping(value="/student", method=RequestMethod.GET)
public String showForm() {
    return "student/form";
}
```

```
@RequestMapping(value="/student", method=RequestMethod.POST)
public String saveData(HttpServletRequest request) {
    String name = request.getParameter("name");
    String mark = request.getParameter("mark");
    String major = request.getParameter("major");

    request.setAttribute("name", name);
    request.setAttribute("mark", mark);
    request.setAttribute("major", major);

    return "student/success";
}
```

Với hai phương thức action này khi bạn đưa ra yêu cầu student.htm thì phương thức nào sẽ chạy. Rõ ràng là chưa xác định được mà còn phụ thuộc vào phương thức web là POST hay GET.

- ✓ Tạo 2 file form.jsp và success.jsp trong thư mục WEB-INF/views/student
  - Mã của form.jsp

```
<form action="student.htm" method="post">
    <div>Họ và tên</div>
    <input name="name"/>

    <div>Điểm trung bình</div>
    <input name="mark"/>

    <div>Chuyên ngành</div>
    <label>
        <input name="major" type="radio" value="APP"/>
```

Ứng dụng phần mềm

</label>

<label>

<input name="major" type="radio" value="WEB"/>

Thiết kế trang web

</label>

<hr>

<button>Lưu</button>

</form>

○ Mã của success.jsp

<h3>THÔNG TIN SINH VIÊN POLY</h3>

<ul>

<li>HỌ VÀ TÊN: \${name}</li>

<li>ĐIỂM TB: \${mark}</li>

<li>CHUYÊN NGÀNH: \${major}</li>

</ul>

✓ Chạy: student.htm

## Bài 2 (2 điểm)

Bài thực hành này sẽ giúp các em một cách khác để gọi phương thức action bằng cách phân biệt tham số

✓ Tạo StudentMgrController với các action như sau

```
@Controller
@RequestMapping("/student-mgr")
public class StudentController {
    @RequestMapping()
    public String index(ModelMap model) {
        model.addAttribute("message", "Bạn gọi index()");
        return "student-mgr";
    }

    @RequestMapping(params="btnInsert")
```

```

public String insert(ModelMap model) {
    model.addAttribute("message", "Bạn gọi insert()");
    return "student-mgr";
}

@RequestMapping(params="btnUpdate")
public String update(ModelMap model) {
    model.addAttribute("message", "Bạn gọi update()");
    return "student-mgr";
}

@RequestMapping(params="btnDelete")
public String delete(ModelMap model) {
    model.addAttribute("message", "Bạn gọi delete()");
    return "student-mgr";
}

@RequestMapping(params="lnkEdit")
public String edit(ModelMap model) {
    model.addAttribute("message", "Bạn gọi edit()");
    return "student-mgr";
}
}

```

Với cách ánh xạ action này thì khi bạn gọi student-mgr.htm thì phương thức action này sẽ được thực hiện hoàn toàn phụ thuộc vào tham số truyền theo

- Nếu có tham số btnInsert thì insert() sẽ thực hiện
- Nếu có tham số btnUpdate thì update() sẽ thực hiện
- Nếu có tham số btnDelete thì delete() sẽ thực hiện
- Nếu có tham số lnkEdit thì edit() sẽ thực hiện
- Nếu có không có các tham số trên thì index() sẽ thực hiện
- Nếu có nhiều hơn một tham số trên sẽ gây lỗi

✓ Tạo student-mgr.jsp có mã như sau

```
<h1>STUDENT MANAGER</h1>
```

```
<form action="student-mgr.htm" method="post">
    ${message}

    <hr>
    <button name="btnInsert">Thêm</button>
    <button name="btnUpdate">Cập nhật</button>
    <button name="btnDelete">Xóa</button>
    <button name="btnReset">Nhập lại</button>
</form>
<hr>
<table border="1" style="width:100%">
    <tr>
        <th>Họ và tên</th>
        <th>Điểm TB</th>
        <th>Chuyên ngành</th>
        <th></th>
    </tr>
    <tr>
        <td>Lê Phạm Tuấn Kiệt</td>
        <td>8.5</td>
        <td>CNTT</td>
        <td><a href="student-mgr.htm?lnkEdit">Sửa</a></td>
    </tr>
</table>
```

- ✓ Chạy: student-mgr.htm, nhấp vào các nút để xem kết quả

## PHẦN II

Hãy nâng cấp bài 2 theo các bài tập nhỏ sau đây

### Bài 3 (2 điểm)

Thêm các điều khiển vào form để có giao diện đầy đủ

```
<form action="student-mgr.htm" method="post">
    <div>Họ và tên</div>
```

```

<input name="name"/>

<div>Điểm trung bình</div>
<input name="mark"/>

<div>Chuyên ngành</div>
<label>
    <input name="major" type="radio" value="APP"/>
    Ứng dụng phần mềm
</label>
<label>
    <input name="major" type="radio" value="WEB"/>
    Thiết kế trang web
</label>
<hr>
<button name="btnInsert">Thêm</button>
<button name="btnUpdate">Cập nhật</button>
<button name="btnDelete">Xóa</button>
<button name="btnReset">Nhập lại</button>
</form>
<hr>
<table border="1" style="width:100%">
    <tr>
        <th>Họ và tên</th>
        <th>Điểm TB</th>
        <th>Chuyên ngành</th>
        <th></th>
    </tr>
    <tr>
        <td>Lê Phạm Tuấn Kiệt</td>
        <td>8.5</td>
        <td>CNTT</td>
        <td><a href="student-mgr.htm?lnkEdit">Sửa</a> </td>
    </tr>

```

```

        <tr>
            <td>Bùi Minh Nhật</td>
            <td>7.5</td>
            <td>MUL</td>
            <td><a href="student-mgr.htm?lnkEdit">Sửa</a></td>
        </tr>
    </table>

```

#### Bài 4 (2 điểm)

- ✓ Sử dụng @RequestParam để nhận các tham số name, mark và major sau đó bổ sung các đối số này vào model

```

@RequestMapping(params="btnInsert")
public String insert(ModelMap model,
    @RequestParam("name") String name,
    @RequestParam("mark") Double mark,
    @RequestParam("major") String major) {
    model.addAttribute("name", name);
    ...
    return "success";
}

```

- ✓ Chuyển sang success.jsp để hiển thị thông tin sinh viên đã nhập

```

<h3>THÔNG TIN SINH VIÊN POLY</h3>
<ul>
    <li>HỌ VÀ TÊN: ${name}</li>
    <li>ĐIỂM TB: ${mark}</li>
    <li>CHUYÊN NGÀNH: ${major}</li>
</ul>

```

#### Bài 5 (2 điểm)

- ✓ Định nghĩa lớp JavaBean Student như sau

```

package poly.bean;

public class Student {

```

```

private String name;
private Double mark;
private String major;

public String getName() {
    return name;
}
public void setName(String name) {
    this.name = name;
}
public Double getMark() {
    return mark;
}
public void setMark(Double mark) {
    this.mark = mark;
}
public String getMajor() {
    return major;
}
public void setMajor(String major) {
    this.major = major;
}
}

```

- ✓ Sử dụng lớp này khai báo đối số cho phương thức action update() để nhận dữ liệu form nhập bằng phương pháp sử dụng JavaBean

```

@RequestMapping(params="btnUpdate")
public String update(ModelMap model, Student student) {
    model.addAttribute("student", student);
    return "success2";
}

```

- ✓ Tạo trang success2.jsp như sau để hiển thị thông tin nhập vào form



```
<h3>THÔNG TIN SINH VIÊN POLY</h3>
<ul>
  <li>HỌ VÀ TÊN: ${student.name}</li>
  <li>ĐIỂM TB: ${student.mark}</li>
  <li>CHUYÊN NGÀNH: ${student.major}</li>
</ul>
```

**Chú ý:**

- ✓ Phần I và Phần II chỉ áp dụng cho dạy tích hợp. Sinh viên làm phần 1 và phần 2 theo 2 bài khác nhau tương ứng với 2 phần lý thuyết đã dạy trong bài học.
- ✓ Nếu giảng dạy theo phương pháp truyền thống thì sinh viên phải thực hiện tất cả các bài trong một buổi.