MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Nắm vững cơ chế buộc dữ liệu lên form
- ✓ Sử dụng điều khiển của Spring form
- ✓ Đổ dữ liệu vào các List Control

PHẦN I

Bài 1 (2 điểm)

Buộc dữ liệu vào form bằng phương pháp truyền thống sử dụng các điều khiển HTML

- ✓ Tạo StudentController như sau

```
@Controller
@RequestMapping("/student/")
public class StudentController {
        @RequestMapping("index")
        public String index(ModelMap model) {
                Student student = new Student("Nguyễn Văn Tèo", 9.5, "WEB");
                model.addAttribute("student", student);
                return "student";
        }
}
```

- ✓ Tạo student.jsp có mã như sau

```
<form action="student/update.htm" method="post">
        <div>Họ và tên</div>
        <input name="name" value="${student.name}"/>

        <div>Điểm trung bình</div>
        <input name="mark" value="${student.mark}"/>

        <div>Chuyên ngành</div>
        <label>
```

```
            <input name="major" type="radio" value="APP"
                ${student.major=='APP'?'checked':''}/>
            Ứng dụng phần mềm
        </label>
        <label>
            <input name="major" type="radio" value="WEB"
                ${student.major=='WEB'?'checked':''}/>
            Thiết kế trang web
        </label>
        <hr>
        <button>Cập nhật</button>
</form>
```

✓ Chạy: student/index.jsp bạn sẽ thấy dữ liệu của bean student đặt trong model được buộc lên các điều khiển của form

Bài 2 (2 điểm)

Sử dụng Spring form để buộc dữ liệu một cách dễ dàng và đơn giản hơn rất nhiều. Hãy giữ nguyên StudentController của bài 1 bạn chỉ đổi lại view (return "student2") và thiết kế student2.jsp có mã như sau

```
<%@ page pageEncoding="utf-8"%>
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8"/>
    <title>Spring MVC - Databinding</title>
    <base href="${pageContext.servletContext.contextPath}/">
</head>
<body>
    <h3>SINH VIÊN POLY</h3>
    <form:form action="student/update.htm" modelAttribute="student">
        <div>Họ và tên</div>
        <form:input path="name"/>

        <div>Điểm</div>
```

```
            <form:input path="mark"/>

            <div>Chuyên ngành</div>
            <form:radiobutton path="major"  value="APP"
                                        label="Ứng dụng phần mềm"/>
            <form:radiobutton path="major"  value="WEB"
                                        label="Thiết kế trang web"/>
            <div>
                <button>Cập nhật</button>
            </div>
        </form:form>
</body>
</html>
```

- ✓ Chạy: student/index.jsp bạn sẽ thấy dữ liệu của bean student đặt trong model được buộc lên các điều khiển của form

PHẦN II: ĐỔ DỮ LIỆU VÀO LIST CONTROL

Form được sử dụng ở bài 1 và bài 2 có điều khiển chuyên ngành. Ở phần này được thiết kế sẵn 2 chuyên ngành là APP (Ứng dụng phần mềm) và WEB (Thiết kế trang web).

```
<form:radiobutton path="major"  value="APP" label="Ứng dụng phần mềm"/>
<form:radiobutton path="major"  value="WEB" label="Thiết kế trang web"/>
```

Chúng ta sẽ phải thay đổi mã như thế nào nếu chuyên ngành là dữ liệu động (đọc từ CSDL chẳn hạn).

Phần bài tập này sẽ hướng dẫn bạn cách đổ dữ liệu vào các List Control

Bài 3 (2 điểm)

Sử dụng map để đổ danh sách chuyên ngành vào ListControl

- ✓ Bổ sung vào StudentController một phương thức như sau:

```
    @ModelAttribute("majors")
    public Map<String, String> getMajors() {
        Map<String, String> majors = new HashMap<>();
```

```
        majors.put("APP", "Ứng dụng phần mềm");
        majors.put("WEB", "Thiết kế trang web");
        return majors;
    }
```

*Với cách định nghĩa này thì trong model sẽ có một attribute có tên là **majors** và giá trị là kết quả của phương thức getMajors() là Map<String, String>*

✓ Thay đổi giao diện sử dụng ListControl để hiển thị chuyên ngành

```
<div>Chuyên ngành</div>
<form:radiobuttons path="major" items="${majors}"/>
```

Attribute majors được sử dụng để tạo ra các radio button với value là key và label là value

✓ Bạn hoàn toàn có thể thay thế <form:radiobuttons> bằng <form:select> để tạo ra ComboBox.

✓ Chạy: student/index.jsp bạn sẽ thấy dữ liệu của Map được đổ vào ListControl

Bài 4 (2 điểm)

Sử dụng List<JavaBean> để đổ danh sách chuyên ngành vào ListControl

✓ Tạo lớp JavaBean có 2 thuộc tính là id và name như sau

```
package poly.bean;

public class Major {
    private String id;
    private String name;

    public Major() {}
    public Major(String id, String name) {
        this.id = id;
        this.name = name;
    }

    public String getName() {
```

```
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
}
```

✓ Bổ sung vào lớp StudentController phương thức

```
    @ModelAttribute("majors")
    public List<Major> getMajors() {
        List<Major> majors = new ArrayList<>();
        majors.add(new Major("APP", "Ứng dụng phần mềm"));
        majors.add(new Major("WEB", "Thiết kế trang web"));
        return majors;
    }
```

*Với cách định nghĩa này thì trong model sẽ có một attribute có tên là **majors** và giá trị là kết quả của phương thức getMajors() là List<Major>*

✓ Để đổ dữ liệu từ majors (List<Major>) trong model vào ListControl bạn cần viết mã như sau

```
<div>Chuyên ngành</div>
<form:radiobuttons path="major" items="${majors}"
    itemValue="id" itemLabel="name"/>
```

Attribute majors được sử dụng để tạo ra các radio button với value là thuộc tính id và label là thuộc tính name của mỗi đối tượng Major trong List.

✓ Bạn hoàn toàn có thể thay thế <form:radiobuttons> bằng <form:select> để tạo ra ComboBox.

✓ Chạy: student/index.jsp bạn sẽ thấy dữ liệu của List<Major> được đổ vào ListControl

Bài 5 (2 điểm)

Giảng viên cho thêm

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.*