

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng:

- ✓ Buộc dữ liệu giữa bean với form
- ✓ Đổ dữ liệu vào List Controls (select, radios, checkboxes)
- ✓ Upload file lên server
- ✓ Kiểm và hiển thị lỗi dữ liệu nhập
- ✓ Tùy biến thông báo lỗi đa ngôn ngữ

PHẦN I

Bài 1 (2 điểm)

Thực hiện theo hướng dẫn sau đây để buộc dữ liệu bean vào form

Tạo bean class Student

```
@Data
@AllArgsConstructor
@NoArgsConstructor
public class Student {
    String name;
    String email;
    Double marks;
    Boolean gender;
    String faculty;
}
```

Tạo controller StudentController

```
@Controller
public class StudentController {
    @RequestMapping("/student/form")
    public String form(@ModelAttribute("sv") Student sv) {
        return "student/form";
    }
    @RequestMapping("/student/save")
    public String save(@ModelAttribute("sv") Student sv) {
```

```
        return "student/form";  
    }  
}
```

Tạo View form.jsp chứa Spring Form để buộc với bean sv. Trong đó sử dụng radio cho gender và select box cho faculty (khoa).

Hướng dẫn:

Sử dụng bộ thư viện thẻ Spring Form để tạo form và buộc dữ liệu với bean sv.

```
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form"  
%>  
<form:form action="" modelAttribute="">  
    <form:input path=""/>  
    <form:radiobutton path=""/>  
    <form:select path="">  
        <form:option value=""></form:option>  
    </form:select>  
</form:form>
```

Bài 2 (2 điểm)

Hãy thêm vào class Student field hobbies: List<String> để chứa sở thích của sinh viên.

```
List<String> hobbies;
```

Bổ sung vào StudentController các phương thức sau để chuẩn bị nguồn dữ liệu đổ vào các List Control.

```
@ModelAttribute("genders")  
public Map<Boolean, String> getGenders(){  
    Map<Boolean, String> map = new HashMap<>();  
    map.put(true, "Male");  
    map.put(false, "Female");  
    return map;  
}  
  
@ModelAttribute("faculties")  
public List<String> getFaculties(){  
    return Arrays.asList("CNTT", "DLNHKS", "QTDN");  
}
```

```
@ModelAttribute("hobbies")
public Map<String, String> getHobbies(){
    Map<String, String> map = new HashMap<>();
    map.put("T", "Travelling");
    map.put("M", "Music");
    map.put("F", "Food");
    map.put("O", "Other");
    return map;
}
```

Hiệu chỉnh form.jsp đổ các nguồn dữ liệu vào List Control và buộc dữ liệu các field của bean.

```
<form:radiobuttons items="" path=""/>
<form:checkboxes items="" path=""/>
<form:select path="">
    <form:option value=""></form:option>
</form:select>
```

Bài 3 (1 điểm)

Giảng viên cho thêm về **phần upload file**

PHẦN II

Bài 4 (2 điểm)

Yêu cầu kiểm lỗi cho form nhập trong bài 2 như sau

FIELD	REQUIREMENT	MESSAGES
Name: String	@NotBlank	Vui lòng nhập họ và tên
Email: String	@NotBlank @Email	Vui lòng nhập địa chỉ email Vui lòng nhập đúng định dạng email
Marks: Double	@Min(0) @Max(10) @NotNull	Điểm phải lớn hơn hoặc bằng 0 Điểm phải nhỏ hơn hoặc bằng 10 Vui lòng nhập điểm
Gender: Boolean	@NotNull	Vui lòng chọn giới tính

Faculty: String	@NotBlank	Vui lòng chọn khoa
Hobbies: List<String>	@NotEmpty	Vui lòng chọn sở thích

Hướng dẫn:

- Thư viện cần thiết

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-validation</artifactId>
</dependency>
```

- Bổ sung các annotation kiểm lỗi vào các field của Student

```
@NotBlank(message = "")
@email(message = "")
String email;
```

- Bổ sung mã cho phương thức StudentController.save()
 - Thêm @Validated vào trước đối số Student
 - Thêm đối số Errors vào ngay sau đối số Student
 - Thêm mã vào phương thức save() như sau

```
if(errors.hasErrors()) {
    model.addAttribute("message", "Vui lòng sửa các lỗi sau:");
} else {
    model.addAttribute("message", "Chúc mừng, bạn đã nhập đúng");
}
```

- Hiển thị message và thông báo lỗi lên form.jsp

```
${message}
<form:errors path=""/>
```

Bài 5 (2 điểm)

Hãy tách thông báo lỗi và tổ chức dạng file tài nguyên (*.properties) để dễ dàng quản lý, thay đổi, nâng cấp cũng như đa ngôn ngữ sau này.

Hướng dẫn:

- Tạo file tài nguyên (student.properties) trong thư mục resources của dự án

NotBlank.sv.email=Vui lòng nhập nhập địa chỉ email
Email.sv.email=Vui lòng nhập đúng định dạng email
...

Chú ý: key phải có dạng <Annotation>.<bean>.<property>

- Cấu hình bean MessageSource để nạp file tài nguyên vào ứng dụng

```
@Configuration
public class MessagesConfig {
    @Bean("messageSource")
    public MessageSource getMessageSource() {
        ReloadableResourceBundleMessageSource ms
            = new ReloadableResourceBundleMessageSource();
        ms.setBasenames("classpath:messages/student.properties");
        ms.setDefaultEncoding("utf-8");
        return ms;
    }
}
```

Bài 6 (1 điểm)

Giảng viên cho thêm