

## MỤC TIÊU:


Kết thúc bài thực hành này bạn có khả năng:

- ✓ Sử dụng biểu thức
- ✓ Sử dụng chỉ thị
- ✓ Định dạng dữ liệu
- ✓ Hiển thị danh sách dữ liệu lên bảng và list control

## PHẦN I

### Bài 1 (2 điểm)

Xây dựng trang web hiển thị dữ liệu từ bean theo yêu cầu sau đây

Bean Staff	Kết quả hiển thị
<pre>@NoArgsConstructor @AllArgsConstructor @Builder @Data public class Staff {     private String id;     private String fullname;     @Default     private String photo = "photo.jpg";     @Default     private Boolean gender = true;     @Default     private Date birthday = new Date();     @Default     private Double salary = 12345.6789;     @Default     private Integer level = 0; }</pre>	 <hr/> <p>Email: <b>user@gmail.com</b></p> <hr/> <p>Họ và tên: <b>Nguyễn Văn User</b></p> <hr/> <p>Giới tính: <b>Nam</b></p> <hr/> <p>Ngày sinh: <b>23-12-2024</b></p> <hr/> <p>Cấp bậc: <b>Tướng</b></p> <hr/> <p>Salary: <b>\$12,345.68</b></p>

Yêu cầu về trình bày dữ liệu cụ thể:

- Họ và tên: chữ in hoa đầu mỗi từ
- Hiển thị hình thay vì tên hình
- Giới tính hiển thị là Nam nếu true và ngược lại là Nữ
- Ngày sinh hiển thị theo định dạng ngày – tháng – năm
- Lương hiển thị 2 số lẻ và phân nhóm bằng dấu phẩy
- Cấp bậc:
  - 0: Ủy
  - 1: Tá
  - 2: Tướng

Hướng dẫn:

Bước 1: Tạo bean class Staff như trên

Bước 2: Tạo trang staff-detail.html

```
<div>
  <hr>
  Email:
  <b th:text="${staff.id}"/></b><hr>
  Họ và tên:
  <b th:text="${#strings.capitalizeWords(staff.fullname)}"/></b><hr>
  Giới tính:
  <b th:text="${staff.gender ? 'Nam' : 'Nữ'}"/></b><hr>
  Ngày sinh:
  <b th:text="${#dates.format(staff.birthday, 'dd-MM-yyyy')}/></b><hr>
  Cấp bậc:
  <th:block th:switch="${staff.level}">
    <b th:case="0">Ủy</b>
    <b th:case="1">Tá</b>
    <b th:case="2">Tướng</b>
    <b th:case="*">Chưa có</b>
  </th:block><hr>
  Salary:
  <b th:text="${#numbers.formatDecimal(staff.salary, 0, 'COMMA', 2, 'POINT')}/></b><hr>
</div>
```

Chú ý:

- Biến staff là bean Staff được chia sẻ bởi controller (trong bước 3)
- #strings, #numbers và #dates là các tiện ích được Thymeleaf cung cấp sẵn hỗ trợ xử lý chuỗi, số và thời gian
- Các chỉ thị được sử dụng trong bài này gồm th:src, th:text, th:switch...th:case
- Các biểu thức gồm: \${}, @{}

Bước 3: Tạo controller

```
@RequestMapping("/staff/detail")
public String detail(Model model) {
    Staff staff = Staff.builder()
        .id("user@gmail.com")
        .fullname("nguyễn văn user")
        .level(2)
        .build();
    model.addAttribute("staff", staff);
    return "/demo/staff-detail";
}
```

## Bài 2 (2 điểm)

Xây dựng trang web hiển thị List<Staff> như hình sau.



1. Nguyễn Văn User1, \$12,345.68



2. Nguyễn Văn User2, \$12,345.68



3. Nguyễn Văn User3, \$12,345.68



4. Nguyễn Văn User4, \$12,345.68



5. Nguyễn Văn User5, \$12,345.68



6. Nguyễn Văn User6, \$12,345.68

Yêu cầu trình bày:

- Mỗi hàng 1 Staff
- Thông tin mỗi Staff gồm hình, số thứ tự, họ tên, lương và đường kẻ dưới (hàng cuối không có)

Hướng dẫn:

Bước 1: Trong controller tạo List<Staff> và chia sẻ với view staff-list.html

```
@RequestMapping("/staff/list")
public String list(Model model) {
    List<Staff> list = List.of(
        Staff.builder().id("user1@gmail.com").fullname("nguyễn văn user1").level(0).build(),
        Staff.builder().id("user2@gmail.com").fullname("nguyễn văn user2").level(1).build(),
        Staff.builder().id("user3@gmail.com").fullname("nguyễn văn user3").level(2).build(),
        Staff.builder().id("user4@gmail.com").fullname("nguyễn văn user4").level(2).build(),
        Staff.builder().id("user5@gmail.com").fullname("nguyễn văn user5").level(1).build(),
        Staff.builder().id("user6@gmail.com").fullname("nguyễn văn user6").level(0).build());
    model.addAttribute("list", list);
    return "/demo/staff-list";
}
```

Bước 2: Tạo view staff-list.html hiển thị List<Staff>

```
<div th:each="staff, st : ${list}">
    
    <b th:text="@{${st.count}. }"></b>
    <b th:text="${#strings.capitalizeWords(staff.fullname)}"></b>,
    $<b th:text="${#numbers.formatDecimal(staff.salary, 0, 'COMMA', 2,
'POINT')}"></b>
    <th:block th:if="${!st.last}"><hr></th:block>
</div>
```

Chú ý:

- Sử dụng th:each để duyệt list lấy từ Model
- Biến st chứa thông tin trạng thái của mỗi phần tử.
  - st.count là số thứ tự của phần tử hiện tại
  - st.last có giá trị true cho biết đó là phần tử cuối cùng

- Biểu thức `@{|${st.count}. |}` dùng để ghép số thứ tự và chuỗi “.”
- Dòng code `<th:block th:if="${!st.last}"><hr></th:block>` chỉ hiển thị `<hr>` nếu đó không phải là phần tử cuối cùng.

### Bài 3 (1 điểm)

Giảng viên cho thêm

## PHẦN II

### Bài 4 (1 điểm)

Hiển thị thông tin trạng thái của mỗi phần tử trong danh sách `List<String>` lên bảng như sau:

HỌ VÀ TÊN	GIÁ TRỊ CÁC THUỘC TÍNH TRẠNG THÁI					
	Index	Count	First	Last	Odd	Even
Nguyễn Văn User1	0	1	true	false	true	false
Nguyễn Văn User2	1	2	false	false	false	true
Nguyễn Văn User3	2	3	false	false	true	false
Nguyễn Văn User4	3	4	false	false	false	true
Nguyễn Văn User5	4	5	false	false	true	false
Nguyễn Văn User6	5	6	false	true	false	true

Bước 1: Trong controller tạo `List<Staff>` và chia sẻ với view `list-status.html`

```
@RequestMapping("/staff/list")
public String list(Model model) {
    List<Staff> list = List.of(
        Staff.builder().id("user1@gmail.com").fullname("nguyễn văn user1").level(0).build(),
        Staff.builder().id("user2@gmail.com").fullname("nguyễn văn user2").level(1).build(),
        Staff.builder().id("user3@gmail.com").fullname("nguyễn văn user3").level(2).build(),
        Staff.builder().id("user4@gmail.com").fullname("nguyễn văn user4").level(2).build(),
        Staff.builder().id("user5@gmail.com").fullname("nguyễn văn user5").level(1).build(),
        Staff.builder().id("user6@gmail.com").fullname("nguyễn văn user6").level(0).build());
    model.addAttribute("list", list);
    return "/demo/list-status";
}
```

Bước 2: Hiển thị thông tin trạng thái của mỗi phần tử trong List<Staff>

```
<table border="1" style="width:100%">
  <thead>
    <tr>
      <th rowspan="2">HỌ VÀ TÊN</th>
      <th colspan="6">CÁC THUỘC TÍNH TRẠNG THÁI</th>
    </tr>
    <tr>
      <th>Index</th>
      <th>Count</th>
      <th>First</th>
      <th>Last</th>
      <th>Odd</th>
      <th>Even</th>
    </tr>
  </thead>
  <tbody>
    <th:block th:each="staff, st : ${list}">
      <tr th:style="@{background-color:${st.odd?'pink':'cyan'}}">
        <td th:text="${#strings.capitalizeWords(staff.fullname)}"></td>
        <td th:text="${st.index}"></td>
        <td th:text="${st.count}"></td>
        <td th:text="${st.first}"></td>
        <td th:text="${st.last}"></td>
        <td th:text="${st.odd}"></td>
        <td th:text="${st.even}"></td>
      </tr>
    </th:block>
  </tbody>
</table>
```

Chú ý:

- Thuộc tính count bắt đầu từ 1 trong khi đó index bắt đầu từ không
- Biểu thức `${st.odd?'pink':'cyan'}` cho kết quả 'pink' nếu phần tử lẻ và ngược lại (chẵn) là 'cyan'.
- Khi chạy cần view source để xem kết quả HTML được sinh ra.

## Bài 5 (1 điểm)

Viết mã để đổ dữ liệu và select box, radio group theo yêu cầu sau

### Fill to SelectBox

nguyễn văn user3 ▾

### Create Radio Button Group

- ☐ nguyễn văn user1
- ☐ nguyễn văn user2
- ☐ nguyễn văn user3
- ☒ nguyễn văn user4
- ☐ nguyễn văn user5
- ☐ nguyễn văn user6

Hướng dẫn:

Bước 1: Trong controller tạo List<Staff> và chia sẻ với view list-controls.html

```
@RequestMapping("/staff/list")
public String list(Model model) {
    List<Staff> list = List.of(
        Staff.builder().id("user1@gmail.com").fullname("nguyễn văn user1").level(0).build(),
        Staff.builder().id("user2@gmail.com").fullname("nguyễn văn user2").level(1).build(),
        Staff.builder().id("user3@gmail.com").fullname("nguyễn văn user3").level(2).build(),
        Staff.builder().id("user4@gmail.com").fullname("nguyễn văn user4").level(2).build(),
        Staff.builder().id("user5@gmail.com").fullname("nguyễn văn user5").level(1).build(),
        Staff.builder().id("user6@gmail.com").fullname("nguyễn văn user6").level(0).build());
    model.addAttribute("list", list);
    return "/demo/list-controls";
}
```

Bước 2: Đổ List<Staff> vào SelectBox và tạo radio button group

```
<h3>Fill to SelectBox</h3>
<select name="cbo">
<th:block th:each="staff: ${list}">
    <option th:value="${staff.id}" th:text="${staff.fullname}"></option>
</th:block>
</select>

<h3>Create Radio Button Group</h3>
<th:block th:each="staff: ${list}">
    <label>
        <input type="radio" name="rdo" th:value="${staff.id}">
        <th:block th:text="${staff.fullname}"></th:block>
    </label>
    <br>
</th:block>
```

### Bước 3: Phân tích kết quả HTML được render

```
<h3>Create Radio Button Group</h3>
<label>
    <input type="radio" name="rdo" value="user1@gmail.com">
    nguyên văn user1
</label>
<br>
<label>
    <input type="radio" name="rdo" value="user2@gmail.com">
    nguyên văn user2
</label>
<br>
<label>
    <input type="radio" name="rdo" value="user3@gmail.com">
    nguyên văn user3
</label>
<br>
<label>
    <input type="radio" name="rdo" value="user4@gmail.com">
    nguyên văn user4
</label>
<br>
```



```
<label>
    <input type="radio" name="rdo" value="user5@gmail.com">
    nguyên văn user5
</label>
<br>
<label>
    <input type="radio" name="rdo" value="user6@gmail.com">
    nguyên văn user6
</label>
<br>
```

**Bài 6 (1 điểm)**

Giảng viên cho thêm