

## ĐỀ THI KẾT THÚC HỌC PHẦN - ĐỀ SỐ 05

Đề thi môn: **CHUYÊN ĐỀ WEB**

Mã môn học: **214374**

Thời gian làm bài: 90 phút

Được sử dụng tài liệu khi làm bài

---

**Câu 1:** Phương thức nào của jQuery có được dùng để thực hiện một request bất đồng bộ?

- a) `$.ajaxAsync()`
- b) `$.ajax()`
- c) `$.load()`
- d) Tất cả câu trên đúng

**Câu 2:** Chọn các phát biểu đúng về jQuery

- a) jQuery là một thư viện javascript nhanh và ngắn gọn được tạo bởi John Resig vào năm 2006 với khẩu hiệu "Write less, do more".
- b) jQuery làm đơn giản hóa việc duyệt tài liệu HTML, điều khiển sự kiện, tạo chuyển động và tương tác Ajax để phát triển web nhanh.
- c) jQuery hỗ trợ nhiều công cụ để phát triển web site đáp ứng (responsive) và giàu hình thức dùng kỹ thuật AJAX.
- d) Tất cả câu trên đúng.

**Câu 3:** Đối tượng `XMLHttpRequest` thực hiện gì trong Ajax?

- a) Nó là ngôn ngữ lập trình dùng để phát triển ứng dụng Ajax.
- b) Nó cung cấp khả năng đánh dấu và định kiểu cho việc hiển thị văn bản trên trang web.
- c) Nó cung cấp phương tiện để trao đổi dữ liệu có cấu trúc giữa client và web server.
- d) Nó cung cấp khả năng trao đổi bất đồng bộ dữ liệu giữa trình duyệt và web server.

**Câu 4:** Hàm nào của **jQuery** được dùng để ngăn việc chạy các mã javascript trước khi tài liệu được tải xong?

- a) `$(document).ready()`
- b) `$(body).onload()`
- c) `$(document).load()`

**Câu 5:** Với **jQuery**, cho phép toán chọn sau: `$("div.intro")`. Nó chọn gì?

- a) Tất cả các thành phần div với css class="intro"
- b) Tất cả các thành phần div với id="intro"
- c) Thành phần div đầu tiên với class="intro"
- d) Thành phần div đầu tiên với id="intro"

**Câu 6:** Mã **jQuery** dùng để đặt màu nền **blue** cho tất cả thành phần **span**?

- a) `$("span").manipulate("background-color","blue");`
- b) `$("span").layout("background-color","blue");`
- c) `$("span").css("background-color","blue");`
- d) `$("span").style("background-color","blue");`

**Câu 7:** Trong Spring, làm thế nào để tự động cắm vào một trường của một bean bằng một bean khác dùng tên của nó?

- a) Dùng thuộc tính name của **@Autowired** annotation
- b) Bằng cách sử dụng duy nhất một **@Qualifier** annotation
- c) Bằng cách sử dụng cả hai **@Autowired** và **@Qualifier** annotation của Spring
- d) Bằng cách sử dụng các **@Autowired** annotation và đặt tên trường trùng với tên bean

**Câu 8:** Spring MVC framework là gì?

- a) Spring MVC framework là kiến trúc Model-Value-Class được dùng để liên kết dữ liệu model với cá giá trị.
- b) Spring MVC framework cung cấp kiến trúc Model-View-Controller và các thành phần xây dựng sẵn có thể dùng phát triển ứng dụng web linh động và phụ thuộc yếu
- c) Spring MVC framework được dùng để quản lý giao tác (Transaction management) cho ứng dụng Web.
- d) Spring MVC framework được dùng để lập trình hướng lĩnh vực cho ứng dụng Web.

**Câu 9:** Trong Spring MVC, **@Controller** annotation là gì?

- a) Đăng ký một lớp đặc biệt có vai trò của một bộ điều khiển xử lý các HTTP request.
- b) Chỉ ra làm thế nào để điều khiển việc quản lý giao dịch.
- c) Chỉ ra làm thế nào để điều khiển việc cắm phụ thuộc (dependency injection).
- d) Chỉ ra làm thế nào để điều khiển việc quản lý cơ sở dữ liệu

**Câu 10:** Tên của tham số (phần chứa ?????) xác định vị trí của file cấu hình Spring MVC trong file web.xml?

```
<servlet>
  <servlet-name>mvc-dispatcher</servlet-name>
  <servlet-class>
    org.springframework.web.servlet.DispatcherServlet
  </servlet-class>
  <init-param>
    <param-name>????</param-name>
    <param-value>
      /WEB-INF/spring/mvc-config.xml
    </param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>mvc-dispatcher</servlet-name>
  <url-pattern>/</url-pattern>
</servlet-mapping>
```

- a) **contextListener**
- b) **configurationLocation**
- c) **contextConfigLocation**

**Câu 11:** Trong **web.xml**, tên của **servlet** được cấu hình là **mvc-dispatcher**. Khi đó tên mặc nhiên của file cấu hình Spring MVC là gì?

- a) **mvc-config.xml**
- b) **mvc-dispatcher.xml**
- c) **mvc-dispatcher-servlet.xml**
- d) **servlet-mvc-dispatcher.xml**

**Câu 12:** Điều nào sau đây là phát biểu đúng về cấm phụ thuộc (Dependency Injection)?

- a) Là mẫu thiết kế để cài đặt nguyên lý Inversion of Control cho phát triển phần mềm.
- b) Nó là một module của Spring.
- c) Nó là kỹ thuật để lấy phụ thuộc cho bất cứ dự án phần mềm nào..
- d) Nó giúp giảm sự phụ thuộc giữa các đối tượng trong ứng dụng.

**Câu 13:** Phát biểu đúng về spring bean?

- a) Spring bean được quản lý bởi Spring IoC container.
- b) Spring bean được khởi tạo, được lắp ráp, và được quản lý bởi Spring IoC container.
- c) Spring bean là các lớp Java bình thường (POJO).
- d) Tất cả điều trên đúng.

**Câu 14:** Phạm vi mặc nhiên của một bean trong Spring framework?

- a) **singleton**
- b) **prototype**
- c) **request**
- d) **session**

**Câu 15:** Trong Spring MVC, có thể định nghĩa bean bằng các annotation kiểu mở rộng sau?

- a) **@Component, @Entity, @Service, @Controller**
- b) **@Component, @Controller, @Service, @Repository**
- c) **@Repository, @Service, @Controller, @Entity**
- d) Không câu nào đúng

**Câu 16:** Các ORM (Object-Relation Mapping) nào mà Spring hỗ trợ?

- a) Hibernate
- b) iBatis
- c) JPA
- d) JDBC
- e) Tất cả điều trên đúng.

**Câu 17:** Kiểu cấm phụ thuộc nào Spring hỗ trợ?

- a) Constructor based, Setter based
- b) Constructor based, Setter based, Getter Based
- c) Setter based, Getter based, Properties based
- d) Constructor based, Setter based, Properties based

**Câu 18:** Trong Spring, **bean autowiring** là gì?

- a) Autowiring cho phép Spring phân giải các bean cộng tác với một bean được đăng ký, bằng cách duyệt tìm bean trong nội dung của **BeanFactory**.
- b) Autowiring cắm giá trị vào các spring bean.
- c) Autowiring cắm một bean vào một bean khác.
- d) Autowiring giúp nối kết một danh sách các giá trị, cho phép lặp lại.

**Câu 19:** Trong Spring MVC, **DispatcherServlet** là gì?

- a) **DispatcherServlet** được dùng để quản lý giao tác (transaction management).
- b) **DispatcherServlet** được dùng để lập trình hướng lĩnh vực (AOP).
- c) **DispatcherServlet** điều khiển tất cả các HTTP request và HTTP response.
- d) **DispatcherServlet** được dùng để cấm phụ thuộc (Dependency injection).

**Câu 20:** Các kỹ thuật view nào không được hỗ trợ bởi Spring MVC?

- a) JSP
- b) PHP
- c) Velocity templates
- d) XSLT
- e) ASP
- f) Tiles

**Câu 21:** Cài đặt **ViewResolver** mặc nhiên được cấu hình cho Spring MVC là?

- a) InternalResourceViewResolver
- b) JspResourceViewResolver
- c) UrlBasedViewResolver
- d) BeanNameViewResolver

**Câu 22:** Phát biểu đúng về **HTTP Accept header**?

- a) Nó được dùng trong ứng dụng Spring MVC để quyết định kiểu dữ liệu được yêu cầu bởi client là một trình duyệt.
- b) Nó được dùng cho REST web services.
- c) Nó được client là trình duyệt tạo ra trong request để xác định nội dung dữ liệu được gửi cho server.
- d) Nó được dùng để đặt tham số truyền cho phương thức controller của ứng dụng Spring MVC.

**Câu 23:** Chọn các phát biểu đúng về **Data Binding**?

- a) Là kỹ thuật mà ứng dụng Spring MVC dùng để liên kết dữ liệu trên web form với thuộc tính của đối tượng Java Bean.
- b) Là kỹ thuật ánh xạ các đối tượng vào các bảng của cơ sở dữ liệu quan hệ.
- c) Nó chuyển đổi dữ liệu dạng chuỗi trên web form thành kiểu dữ liệu java tương ứng với thuộc tính của đối tượng Java liên kết với nó, và thực hiện việc kiểm tra tính hợp lệ của dữ liệu.
- d) Nó chỉ được dùng cho web services.

**Câu 24:** Dữ liệu gì mà một phương thức điều khiển của Spring MVC không được trả về?

- a) Một chuỗi **string**
- b) Một đối tượng **Model**
- c) Một đối tượng **ModelAndView**
- d) Một đối tượng **JstlView**
- e) Giá trị **null**

**Câu 25:** Cho controller sau, khai báo phương thức **show** có đúng không?

```
@Controller
@RequestMapping("/persons")
public class PersonsController {
    @RequestMapping("/{id}")
    public String show(@PathVariable String number, Model model) {
        ...
    }
}
```

- a) Yes
- b) No

**Câu 26:** Các annotation nào sau đây dùng để kiểm tra hợp lệ dữ liệu trên các thuộc tính của đối tượng form dùng JSR 303 Bean Validator API?

- a) @NotNull
- b) @Size
- c) @Valid
- d) @NotEmpty
- e) @Constraint
- f) @Required
- g) @Pattern

**Câu 27:** Phát biểu nào đúng về cấu hình dùng chú giải (annotation-based configuration)?

- a) Chú giải một lớp là **Controller** không đủ để cho lớp điều khiển request, lớp này cần phải mở rộng lớp **AbstractController** của Spring.
- b) @RequestMapping annotation được dùng cho cả mức lớp và phương thức.
- c) Để Spring MVC tự động tìm các lớp controller, chúng ta phải cho phép quét các thành phần trong cấu hình.
- d) @ModelAttribute annotation có thể dùng để chú giải đối số của phương thức controller.

**Câu 28:** Điều gì sai với định nghĩa của một Spring form sau?

```
<%@ taglib prefix="sf" uri="http://www.springframework.org/tags/form" %>
...
<sf:form action="${personsUrl}" method="GET">
    ...
</sf:form>
```

- a) Phương thức của form không thể là GET.
- b) Thiếu thuộc tính `modelAttribute` của form.
- c) Thiếu khai báo

```
<%@ taglib prefix="spring" uri="http://www.springframework.org/tags" %>
```

**Câu 29:** Spring MVC có hỗ trợ kiểm tra hợp lệ (validation) dữ liệu của form không?

- a) Yes
- b) No

**Câu 30:** Cho Spring MVC controller sau, request nào sẽ gọi phương thức **call**?

```
@Controller
@RequestMapping("/persons")
public class PersonsController {
    @RequestMapping("/list")
    public String call(Model model, HttpServletRequest rq) {
        ...
    }
}
```

- a) `http://localhost:8080/persons`
- b) `http://localhost:8080/persons/list`
- c) `http://localhost:8080/persons/call`

**Câu 31:** Điều gì là đặc trưng của JSF managed bean?

- a) Nó nên có các phương thức `getter` và `setter` cho các biến.
- b) Nó nên có một constructor rỗng.
- c) Chúng buộc ứng xử của web form người dùng với nhau.
- d) Tất cả điều trên đều đúng.

**Câu 32:** Trong JSF, chọn các phát biểu đúng

- a) Sự khác nhau giữa một java bean bình thường và một managed bean là managed bean cũng là một java bean bình thường nhưng vòng đời được quản lý bởi JSF.
- b) Mục đích duy nhất của managed bean là lưu giữ trạng thái của trang web.
- c) Các managed bean không nên có biến thể hiện khai báo là public.
- d) Một trang JSF không được phép truy xuất nhiều hơn một managed bean.
- e) Thuộc tính của managed bean nên được truy xuất thông qua các phương thức getXxx và setXxx.

**Câu 33:** Trong JSF, servlet mapping cho Faces servlet trong **web.xml** là

```
<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>*.faces</url-pattern>
</servlet-mapping>
```

Giả sử ngữ cảnh ứng dụng web là **/mywebapp**, khi đó có thể truy xuất trang **index.xhtml** :

- a) `http://localhost:8080/mywebapp/index.xhtml`
- b) `http://localhost:8080/mywebapp/faces.index`
- c) `http://localhost:8080/mywebapp/faces/index.xhtml`
- d) `http://localhost:8080/mywebapp/index.faces`

**Câu 34:** Trong JSF, cho backing bean được đăng ký là thể hiện của lớp sau::

```
01: @ManagedBean
02: public class TestBean {
03:     public void action() {
04:         System.out.println("ACTION");
05:     }
06:
07:     public void action(String message) {
08:         System.out.println(message);
09:     }
10:
11:     public void action(int id) {
12:         System.out.println(id);
13:     }
14: }
```

và có nút lệnh trên một trang như sau:

```
<h:commandButton id="button" action="#{testBean.action}" />
```

Điều gì xảy ra khi click vào nút lệnh?

- a) Không có gì xảy ra
- b) Console hiển thị chuỗi "ACTION"
- c) Console hiển thị chuỗi id của nút lệnh là "button"
- d) Ném một ngoại lệ

**Câu 35:** Trong JSF, mã để lấy thiết lập locale hiện thời (thiết lập về ngôn ngữ, định dạng số ngày giờ, ...) trong backing bean là gì?

- a) `FacesContext context = FacesContext.getCurrentInstance();`  
`Locale locale = context.getViewRoot().getLocale();`
- b) `FacesContext context = FacesContext.getCurrentInstance();`  
`Locale locale = context.getLocale();`
- c) `FacesContext context = FacesContext.getCurrentInstance();`  
`Locale locale = context.getApplication().getDefaultLocale();`

**Câu 36:** Trong JSF, nếu thuộc tính *immediate* của một thành phần **UIInput** được đặt là **true**, thì xử lý kiểm tra tính hợp lệ dữ liệu và chuyển đổi dữ liệu được tiến hành ở tầng nào trong vòng đời trang JSF

- a) Apply Request values
- b) Process Validation
- c) Update Model values
- d) Invoke Application
- e) Render Response

**Câu 37:** Trong JSF, giả sử muốn cung cấp một input mà sẽ kiểm tra dữ liệu nhập của người dùng là một số nguyên có giá trị từ 1 tới 5. Mã nào sau đây tương ứng với yêu cầu:

- a) `<h:inputText id="rating" value="#{someBean.rating}" required="true" type="long" minimum="1" maximum="5"/>`
- b) `<h:inputText id="rating" value="#{someBean.rating}" required="true">  
    <f:validateLongRange minimum="1" maximum="5"/>  
</h:inputText>`
- c) `<h:inputLong id="rating" value="#{someBean.rating}" required="true" minimum="1" maximum="5"/>`
- d) `<h:inputText id="rating" value="#{someBean.rating}" required="true">  
    <f:validate type="long" range="1..5"/>  
</h:inputText>`

**Câu 38:** Giả sử có lớp User với 2 thuộc tính: *firstName* có kiểu String và *dateOfBirth* có kiểu Date. Bean user được đăng ký với *firstName* được đặt là John và *dateOfBirth* được đặt ngày 1 April 2000.

Kết xuất của đoạn mã sau trong trang JSF là gì?

```
<h:outputFormat value="Hello {0}. I know you were born on {1, date}.">  
    <f:param value="#{user.firstName}"/>  
    <f:param value="#{user.dateOfBirth}"/>  
</h:outputFormat>
```

- a) Hello John. I know you were born on 01/04/00
- b) Hello John. I know you were born on date
- c) Hello John. I know you were born on xxxxxx

Với việc hiển thị ngày xxxxxx phụ thuộc vào thiết lập local hiện thời.

- d) Không câu nào trên đúng

**Câu 39:** Trong JSF, các sự kiện Value Change và Action được thực hiện sau tầng Process Validation. Để bỏ qua xử lý kiểm tra hợp lệ dữ liệu và đi thẳng tới tầng Render Response, có thể đặt một thuộc tính cho thành phần command như thế nào?

- a) `<h:commandLink actionListener="#{bean.someMethod}" immediate="true" />`
- b) `<h:commandLink actionListener="#{bean.someMethod}" immediate="false" />`
- c) `<h:commandLink actionListener="#{bean.someMethod}" validate="true" />`
- d) `<h:commandLink actionListener="#{bean.someMethod}" validate="false" />`

**Câu 40:** Trong JSF, để lấy một biến có tên "myVar" trong phạm vi **request** dùng mã gì?

- a) `FacesContext context = FacesContext.getCurrentInstance();  
Object o = context.getExternalContext().getRequest().get("myVar");`
- b) `FacesContext context = FacesContext.getCurrentInstance();  
Object o = context.getExternalContext().getRequestMap().get("myVar");`
- c) `FacesContext context = FacesContext.getCurrentInstance();  
Object o = context.getApplication().getRequest().get("myVar");`



**Câu 41:** Trong JSF, cho các khai báo chuyển trang sau:

```
<navigation-rule>
  <from-view-id>hello.xhtml</from-view-id>
  <navigation-case>
    <from-outcome>Hello</from-outcome>
    <to-view-id>login.xhtml</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>turtle</from-outcome>
    <to-view-id>powerRanger.xhtml</to-view-id>
  </navigation-case>
</navigation-rule>
```

Và liên kết action trong trang JSP:

```
<h:commandLink value="Click" action="hugo" />
```

Điều gì xảy ra nếu chọn liên kết?

- a) Sẽ có một lỗi server.
- b) Sẽ hiển thị trang hiện thời.
- c) Sẽ hiển thị trang login.xhtml.
- d) Sẽ hiển thị trang powerRangers.xhtml.

**Câu 42:** Để khai báo nhiều file cấu hình cho ứng dụng JSF, thêm khai báo sau trong file `/WEB-INF/web.xml`

- a) 

```
<context-param>
  <param-name>javax.faces.CONFIG_FILES</param-name>
  <param-value>/WEB-INF/navigation.xml,
                /WEB-INF/beans.xml
</param-value>
</context-param>
```
- b) 

```
<init-param>
  <param-name>javax.faces.CONFIG_FILES</param-name>
  <param-value>/WEB-INF/navigation.xml,
                /WEB-INF/beans.xml
</param-value>
</init-param>
```
- c) Không thành phần nào ở trên

**Câu 43:** Kiểu dữ liệu gì có thể buộc cho thuộc tính value của thẻ JSF

```
<f:selectItems value="#{_____}" />
```

- a) SelectItem
- b) SelectItem[]
- c) Collection<SelectItem>
- d) Map với các mục biểu diễn các giá trị value và label của SelectItem
- e) Null

**Câu 44:** Đoạn mã của trang JSF có hợp lệ không ?

```
<html xmlns=http://www.w3.org/1999/xhtml
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:h="http://java.sun.com/jsf/html">
  <h:body>
    <h:commandButton action="#{bean.dosomething}" value="Do something"/>
  </h:body>
</html>
```

- a) Đúng
- b) Sai



**Câu 45:** Cách đúng để đăng ký và sử dụng một bộ chuyển đổi kiểu (converter)

- a) `<converter>`  
    `<converter-name>EmailConverter</converter-name>`  
    `<converter-class>com.emailConverter</converter-class>`  
    `</converter>`  
    Và trong trang JSF: `<f:converter converter="EmailConverter" />`
- b) `<converter>`  
    `<converter-name>EmailConverter</converter-name>`  
    `<converter-class>com.emailConverter</converter-class>`  
    `</converter>`  
    Và trong trang JSF: `<h:converter converter="EmailConverter" />`
- c) `<converter>`  
    `<converter-id>EmailConverter</converter-id>`  
    `<converter-class>com.emailConverter</converter-class>`  
    `</converter>`  
    Và trong trang JSF: `<h:converter converterId="EmailConverter" />`
- d) `<converter>`  
    `<converter-id>EmailConverter</converter-id>`  
    `<converter-class>com.emailConverter</converter-class>`  
    `</converter>`  
    Và trong trang JSF: `<f:converter converterId="EmailConverter" />`

**Câu 46:** Cho 2 bean:

**Bean 1:**

```
@ManagedBean(name="personBean")
@RequestScope
public PersonBean1 {
    String name = "John";
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }
}
```

**Bean 2:**

```
@ManagedBean(name="personBean")
@SessionScope
public PersonBean2 {
    String name = "Smith";
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }
}
```

Và trang **test.xhtml**:

```
<h:body>
    name = <h:outputText value="#{personBean.name}" />
</h:body>
```

Phát biểu nào sau đây đúng?

- a) Ứng dụng bị lỗi khi tải và ném ngoại lệ có 2 managed bean cùng tên.  
b) Khi chạy test.faces, ném ngoại lệ có 2 managed bean cùng tên.  
c) Khi chạy test.faces, sẽ in ra chuỗi 'John'  
d) Khi chạy test.faces, sẽ in ra chuỗi 'Smith'.  
e) Không câu nào đúng.

**Câu 47:** Câu lệnh nào hiển thị kết quả 5 khi chạy trang JSF

- a) `${3+2}`
- b) `<h:outputText value="#{3+2}"></h:outputText>`
- c) `<h:outputText value="#{${3+2}}"></h:outputText>`
- d) Không câu nào đúng

**Câu 48:** Đoạn mã JSF nào tạo liên kết html như

- `<a href="http://www.mylink.com?projectId=123&userId=45">Go</a> ?`
- a) `<h:outputLink value="http://www.mylink.com";>  
    <f:param name="projectId" value="123" />  
    <f:param name="userId" value="45" />  
    <h:outputText value="Go" />  
</h:outputLink>`
  - b) `<h:outputFormat value="http://www.mylink.com";>  
    <f:param name="projectId" value="123" />  
    <f:param name="userId" value="45" />  
    <h:outputText value="Go" />  
</h:outputFormat>`
  - c) `<h:commandLink  
    value="http://www.mylink.com?projectId="123"&userId="45" />`

**Câu 49:** Cho một danh sách các chuỗi `java.util.List<String> shoppingItems` trong một bean `shoppingCart`. Cách đúng để hiển thị các mục vào một bảng dữ liệu trong JSF.

- a) `<h:panelGrid collection="#{shoppingCart.shoppingItems}" var="item">  
    <h:column>  
        <h:outputText value="#{item}" />  
    </h:column>  
</h:panelGrid>`
- b) `<h:dataTable collection="#{shoppingCart.shoppingItems}" var="item">  
    <h:column>  
        <h:outputText value="#{item}" />  
    </h:column>  
</h:dataTable>`
- c) `<h:dataTable var="item" value="#{shoppingCart.shoppingItems}">  
    <h:column>  
        <h:outputText value="#{item}" />  
    </h:column>  
</h:dataTable>`
- d) None of the above

**Câu 50:** Đoạn Mã nào sau đây truy xuất đúng thuộc tính `cistId` trong phạm vi `session`?

- a) `<h:outputText value="#{facesContext.externalContext.session.cistId}" />`
- b) `<h:outputText value="#{session.cistId}" />`
- c) `<h:outputText value="#{pageContext.session.cistId}" />`
- d) Không thể truy xuất nếu dùng mã JSF thuần túy.

Hết

---

**TRƯỞNG BỘ MÔN**  
(Ký, ghi rõ họ và tên)

---

**GIẢNG VIÊN RA ĐỀ**  
(Ký, ghi rõ họ và tên)

.....

.....