BÀI 1

Câu 1: Trình bày khái niệm và ý nghĩa của Framework ?

* Khái niệm :

Framework (bộ khung) là cấu trúc được dung để xây dựng phần mềm . Framework bao gồm các đoạn code được viết sẵn cùng với các thư viện, tệp hình ảnh và tài liệu tham khảo được đóng thành một gói. Gói này có thể sửa đổi để phù hợp với nhu cầu cụ thể của từng dự án. Các framework sẽ cung cấp nhưng chức năng API (Application Programing Interface) các trình biên dịch điễn dịch, để việc phát triển phần mềm ứng dụng được đơn giản hơn.

* Ý nghĩa :

+ dễ dàng kiểm thử

+ dễ bảo trì và cập nhật

+ tiết kiệm thời gian và chi phí

Câu 2: Phân biệt Framework và library

|  |  |  |
| --- | --- | --- |
|  | **Framework (Bộ khung)** | **Library (Thư viện)** |
| **Định nghĩa** | Là nền tảng để các developer dựa vào đó phát triển website/ ứng dụng cụ thể | Cung cấp cho developers các chức năng (function) và lớp (class) được xác định trước để họ dễ dàng thúc đẩy quá trình xây dựng website/ ứng dụng |
| **Nguyên lý hoạt động** | Framework sẽ điều khiển luồng ứng dụng.  Các khối mã lệnh trong Framework sẽ gửi yêu cầu đến mã lệnh của lập trình viên | Lập trình viên có thể kiểm soát luồng của ứng dụng.  Các khối mã của lập trình viên sẽ gửi yêu cầu đến mã lệnh của library |
| **Thành phần** | Framework sẽ bao gồm nhiều API, trình biên dịch, công cụ, các chương trình hỗ trợ, thư viện,… | Library tập hợp nhiều mô-đun trợ giúp, các đối tượng, lớp, chức năng, mã viết sẵn,… |
| **Khả năng sửa đổi mã** | Những mã trong framework không thể thay đổi/ chính sửa. | Các mã trong library hướng đến một chương trình cụ thể, vì thế library cho phép sửa đổi mã để đáp ứng nhu cầu đó. |
| **Khả năng mở rộng** | Các framework cung cấp chức năng chung. Do đó, Framework có thể mở rộng, cho phép developer kết hợp các tính năng dành riêng cho ứng dụng mà không cần sửa đổi mã nguồn | Library không thể mở rộng. Chúng chỉ được thiết kế để thực hiện một mục đích cụ thể |
| **Khả năng thay thế** | Framework rất khó thay thế. Nếu thay đổi developer phải viết lại toàn bộ codebase | Dễ dàng thay thế bằng một thư viện khác |
| **Performance** | Để phát triển ra một Framework đòi hỏi nhiều mã hóa. Điều này làm tăng thời gian tải và giảm hiệu suất | Cần ít mã để xây dựng nên thời gian tải nhanh hơn và hiệu suất tốt hơn |
| **Một số ví dụ** | Spring, NodeJS, AngularJS, Vue JS,… | JQuery, React JS,… |

Câu 3: Trình bày các thành phần của Spring Framework:

**Test**

Tầng này cung cấp khả năng hỗ trợ kiểm thử với JUnit và TestNG.

**Spring Core Container**

Bao gồm các module spring core, beans, context và expression languate (EL)

* Spring core, bean cung cấp tính năng IOC và Dependency Injection.
* Spring Context hỗ trợ đa ngôn ngữ (internationalization), các tính năng Java EE như EJB, JMX.
* Expression Language được mở rộng từ Expresion Language trong JSP. Nó cung cấp hỗ trợ việc setting/getting giá trị, các method cải tiến cho phép truy cập collections, index, các toán tử logic…

**AOP, Aspects and Instrumentation**

Những module này hỗ trợ cài đặt lập trình hướng khía cạnh (Aspect Oriented Programming), hỗ trợ tích hợp với AspectJ.

**Data Access / Integration**

Nhóm này bao gồm JDBC, ORM, OXM, JMS và module Transaction. Những module này cung cấp khả năng giao tiếp với database

**Web**

Hay còn gọi là Spring MVC Nhóm này gồm Web, Web-Servlet… hỗ trợ việc tạo ứng dụng web.

Câu 4: Trình bày luồng thực thi của Spring MVC

1. **Yêu Cầu từ Client:** Người dùng gửi một yêu cầu từ trình duyệt hoặc ứng dụng di động đến máy chủ thông qua URL.
2. **DispatcherServlet:** DispatcherServlet là một thành phần quan trọng của Spring MVC, nó nhận yêu cầu từ client và quyết định Controller nào sẽ xử lý yêu cầu này.
3. **Controller Handling:** Controller nhận yêu cầu và xử lý nó. Controller thực hiện các thao tác liên quan đến dữ liệu như truy xuất cơ sở dữ liệu, xử lý nghiệp vụ, và chuẩn bị dữ liệu cần thiết để hiển thị.
4. **ModelAndView:** Controller trả về một đối tượng ModelAndView, chứa dữ liệu cần hiển thị và tên của View.
5. **View Resolution:** DispatcherServlet sử dụng tên của View từ ModelAndView để tìm View tương ứng. View có thể là một trang JSP, Thymeleaf, hoặc một đối tượng hiển thị JSON/XML.
6. **Hiển Thị Kết Quả:** View nhận dữ liệu từ ModelAndView và hiển thị chúng theo định dạng thích hợp. Sau đó, kết quả được gửi trả lại cho client.
7. **Phản Hồi cho Client:** Kết quả được gửi trả lại client và hiển thị trên trình duyệt hoặc ứng dụng di động của họ.