**Vũ Đức Thiệp-2022602296-M119**

**PHẦN 1: SOA và Webservice**

Bài 1: Nghiên cứu thư viện JAX-WS

| **Mục** | **Chi tiết** |
| --- | --- |
| **Nội dung** | - **Annotation của JAX-WS**: + @WebService: Định nghĩa một lớp là WebService.  + @WebMethod: Đánh dấu phương thức là một phần của dịch vụ.  + @WebParam: Xác định tham số của phương thức.  + @WebResult: Xác định giá trị trả về của phương thức.  + @SOAPBinding: Xác định kiểu binding (Document hoặc RPC).  - **Các lớp/giao diện quan trọng**:  + javax.xml.ws.Service: Đại diện cho dịch vụ WebService.  + javax.xml.ws.BindingProvider: Cung cấp cấu hình cho client.  + javax.xml.ws.Endpoint: Điểm cuối để triển khai dịch vụ.  - **Kỹ thuật lập trình WebService SOAP**:  + Xây dựng dịch vụ SOAP sử dụng JAX-WS.  + Tạo WSDL (Web Service Description Language) tự động.  + Triển khai và gọi dịch vụ từ client. |
| **Công cụ/Thư viện** | JAX-WS (Java API for XML Web Services) |
| **Kỹ thuật/Quy trình** | - Bước 1: Tạo interface dịch vụ với @WebService.  - Bước 2: Triển khai lớp dịch vụ với các phương thức sử dụng @WebMethod.  - Bước 3: Triển khai dịch vụ bằng Endpoint.publish().  - Bước 4: Tạo client gọi dịch vụ sử dụng các công cụ như wsimport để sinh mã từ WSDL. |
| **Điểm nổi bật** | - JAX-WS hỗ trợ xây dựng WebService dựa trên giao thức **SOAP**.  - Tích hợp tốt với Java EE, tự động sinh WSDL.  - Dễ dàng triển khai dịch vụ và client với các annotation. |

Bài 2: Nghiên cứu thư viện JAX-RS

| **Mục** | **Chi tiết** |
| --- | --- |
| **Nội dung** | - **Annotation của JAX-RS**:  + @Path: Xác định URL endpoint cho dịch vụ.  + @GET, @POST, @PUT, @DELETE: Xác định phương thức HTTP.  + @Produces: Chỉ định định dạng dữ liệu trả về (JSON, XML,...).  + @Consumes: Chỉ định định dạng dữ liệu đầu vào.  + @PathParam, @QueryParam: Trích xuất tham số từ URL.  - **Các lớp/giao diện quan trọng**: + javax.ws.rs.core.Application: Cấu hình ứng dụng REST.  + javax.ws.rs.core.Response: Xử lý phản hồi HTTP.  + javax.ws.rs.client.Client: Gọi API từ phía client.  - **Kỹ thuật lập trình RESTful**:  + Xây dựng REST API với các endpoint.  + Xử lý request/response với các phương thức HTTP.  - **Kiểm tra với Postman**: Gửi các yêu cầu HTTP và kiểm tra phản hồi. |
| **Công cụ/Thư viện** | JAX-RS (Java API for RESTful Web Services), Postman |
| **Kỹ thuật/Quy trình** | - Bước 1: Tạo lớp REST resource với @Path và các annotation HTTP.  - Bước 2: Triển khai logic xử lý request/response.  - Bước 3: Cấu hình ứng dụng REST với Application.  - Bước 4: Triển khai API trên server (như Tomcat, GlassFish).  - Bước 5: Sử dụng Postman để gửi các yêu cầu GET, POST, PUT, DELETE và kiểm tra kết quả. |
| **Điểm nổi bật** | - JAX-RS hỗ trợ xây dựng **RESTful API** đơn giản và linh hoạt. - Tích hợp Postman giúp kiểm tra API dễ dàng. - Hỗ trợ nhiều định dạng dữ liệu (JSON, XML). |

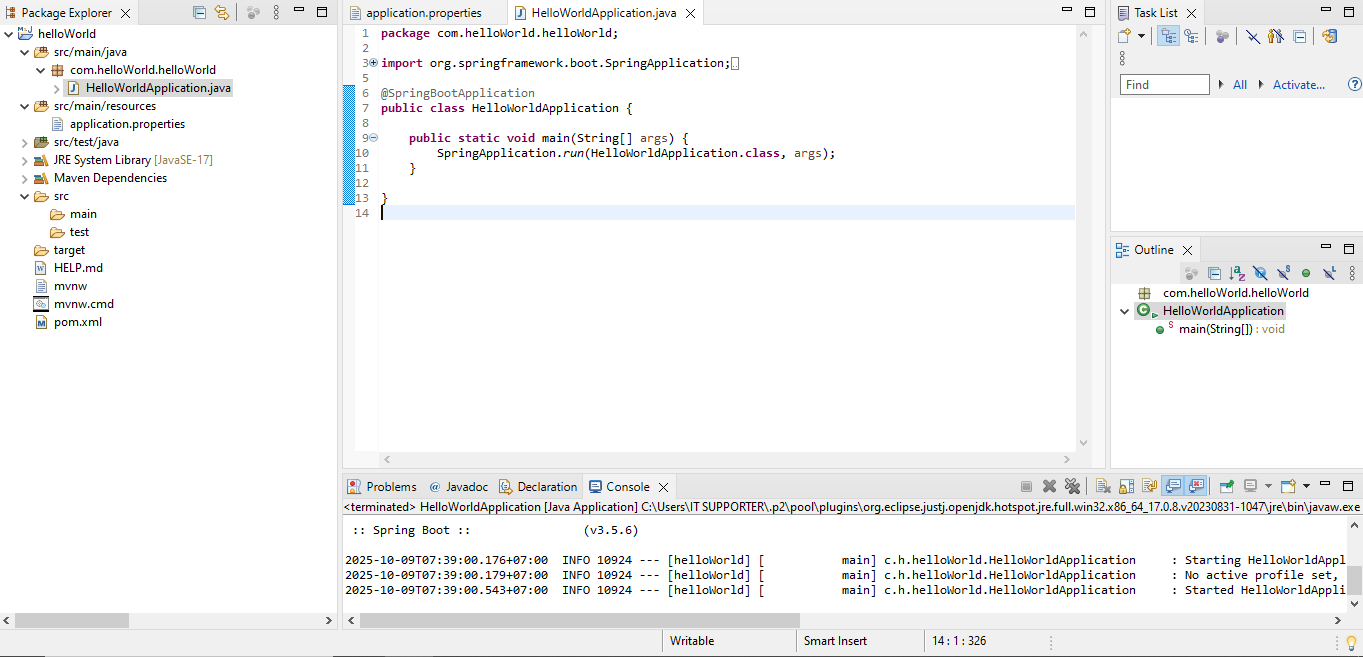
Bài 3: Lập trình với Spring Boot (Hello World)

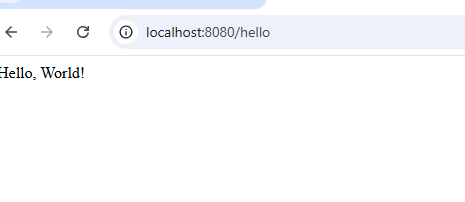
- Bước 1: Tạo dự án Spring Boot qua Spring Initializr hoặc IDE (IntelliJ, Eclipse).

- Bước 2: Tạo Controller với @RestController và phương thức trả về "Hello World".

- Bước 3: Chạy ứng dụng với SpringApplication.run().

- Bước 4: Kiểm tra ứng dụng qua trình duyệt hoặc Postman tại endpoint /helloWorld.





Bài 4: Lập trình WebService với SOAP sử dụng Spring Boot

