**BẢNG ĐÁNH GIÁ ĐIỂM THUYẾT TRÌNH**

**MÔN LẬP TRÌNH JAVA NÂNG CAO**

**LỚP CTK44-PM**

|  |  |
| --- | --- |
| **Thông tin đề tài thuyết trình** | **Thông tin sinh viên thực hiện đánh giá** |
| Tên đề tài: Tìm hiểu về công cụ và cách tổ chức chương trình trong Java Servlet | MSSV của sinh viên đánh giá: 2012353 |
| Điểm tổng kết: Ngày trình bày:04/04/2023 | Họ tên sinh viên & ký xác nhận: Đoàn Cao Nhật Hạ |
| ***Thông tin sinh viên thuyết trình*** |
| MSSV: 2015749 - 2011382  Họ tên: Nguyễn Hoàng Nhật Tiến – Trần Trung Hiếu |

1. **ĐÁNH GIÁ THUYẾT TRÌNH**

**Thang điểm: Kém 0-5; Trung bình: 5-7; Khá 7-8; Tốt 8-9; Xuất sắc: 9-10**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Bảng chấm điểm nhóm thuyết trình** | | | | | | |
| Các chủ đề chính có được xác định? (A) | Slide thuyết trình có thiết kế tốt (B) | Giao tiếp với người nghe bằng ánh mắt (C) | Giao tiếp với người nghe bằng giọng nói (D) | Khả năng trình bày nội dung (E) | Thảo luận & Hỏi đáp (F) | Điểm trung bình = (A+B+C+D+E+F)/6 |
| **7** | **8.5** | **6** | **6** | **6** | **6** | **6.6** |

1. **BÀI TẬP NẮM KIẾN THỨC**

Dựa vào nội dung vừa trình bày, trả lời các câu hỏi sau:

1. Trình bày tóm tắt nội dung anh/ chị đã nắm được về chủ đề vừa trình bày?
2. Khái niệm:

* Là một công nghệ đươc sử dụng để tạo ra ứng dụng web (nằm ở phía máy chủ và tạo ra trang web động).
* Là một API cung cấp các interface và các lớp bao gồm các tài liệu
* Là một interface được triển khai để tạo bất kỳ Servlet

1. Công dụng:

* Nhận request từ client và lấy các thông tin từ request đã nhận
* Truy cập database để xử lý các nghiệp vụ và những phát sinh
* Thực hiện việc tạo, sau đó gửi request đến client hay tạo một request mới đến Servlet và JSP mới

1. Nhiệm vụ :

* Đọc dữ liệu rõ ràng do client gửi
* Đọc dữ liệu yêu cầu HTTP ẩn được gửi bởi client
* Gửi dữ liệu rõ ràng tới client
* Gửi phản hồi HTTP ẩn cho client
* Xử lý dữ liệu và tạo ra các kết quả

1. Servlet Container

* Là sử dụng Java để tự động tạo ra trang web ở phía máy chủ , xử lý các yêu cầu tính toán từ phía client và trả lại các kết quả theo request của client đó. Thực chất nó là một phần của một web server tương tác với các servlet. Đúng như tên gọi servlet container giống như một cái thùng chứa, chứa tất cả các servlet ở bên trong.

1. Môi trường làm việc Servlet:

* Là 1 lớp Java nên nó cần được thực thi trên máy ảo Java(gọi là JVM) thông qua một dịch vụ tên là Servlet engine. Servlet engine sẽ thực hiện tải lớp Servlet đầu tiên ma nó được yêu cầu hoặc tại thời điểm Servlet engine bắt đầu. Tiếp đến, servlet sẽ ngừng tải để tập trung nguồn lực xử lý các yêu cầu khi Servlet engine bị dừng hoặc tắt.
* Vai trò: Sử dụng Servlet cho phép JVM có thể tiến hành xử lý từng yêu cầu riêng lẻ trong mỗi chuỗi Java riêng biệt

1. Cách Servlet container web server xử lý một request: có 6 bước

* Máy chủ Web nhận HTTP request
* Máy chủ Web chuyển tiếp yêu cầu tới servlet container
* Servlet được lấy ra một cách tự động và tải lên địa chỉ không gian của container, nếu nó nằm trong container.
* Container gọi init () method của servlet để khởi tạo (được gọi một lần khi servlet được load lần đầu tiên)
* Container gọi service () method của servlet để xử lý HTTP request, tức là đọc dữ liệu trong yêu cầu và hình thành một response
* Web server trả lại kết quả động đúng với vị trí yêu cầu

1. Các công cụ hỗ trợ:

* Eclipse: Tích hợp Git và Apache Maven, đa nền tảng;
* NetBeans: Thêm các công cụ tái cấu trúc mã,đa nền tảng;
* Intellij: Hỗ trợ nhiều ngôn ngữ khác nhau, có tính năng chạy thử, đa nền tảng;
* Công cụ khác: Emmet: tối giản thời gian viết code

1. Môi trường:

* ApacheTomcat: Ứng dụng mã nguồn mở thực hiện các công nghệ Java Servlet và Java Server Pages, hoạt động như máy chủ độc lập, tính ổn định cao;
* GlassFish: có 1 giao diện đơn giản

1. Mô hình MVC: Model ⬄ Controller -> View -> User

* Model: Tương tác và truy xuất dữ liệu đến database
* Controller: Có nhiệm vụ điều khiển tương tác giữa Model và View cũng như xử lý logic nghiệp vụ(Business)
* View: Giao diện mà người dùng có thể nhìn thấy, thường View chỉ có một nhiện vụ là hiển thị dữ liệu.
* Yêu cầu được gửi đến Controller(Servlet). Controller tiếp nhận, xử lý và làm việc với dữ liệu thông qua Model. Cuối cùng chuyển tiếp đến View(JSP) để tạo giao diện người dùng
* View lấy dữ liệu từ Model (đã được tạo ra từ Controller trước đó) để tạo giao diện theo yêu cầu
* Model đóng vài trò cung cấp, cập nhật dữ liệu và chia sẻ giữa Controller và View

1. Anh/ chị hãy đưa ra các nhận xét và góp ý cho sinh viên vừa trình bày?

* Trình bày chậm hơn chút, nói rõ nội dung bài học hơn, to hơn
* Demo: Cần trình bãy rõ phần nào hiển thị View, phần nào chạy Model, phần nào thực hiện Controller, phần nào người dùng tương tác, khi người dùng tương tác thì sẽ hiện ra những gì, cách thức thực hiện của code