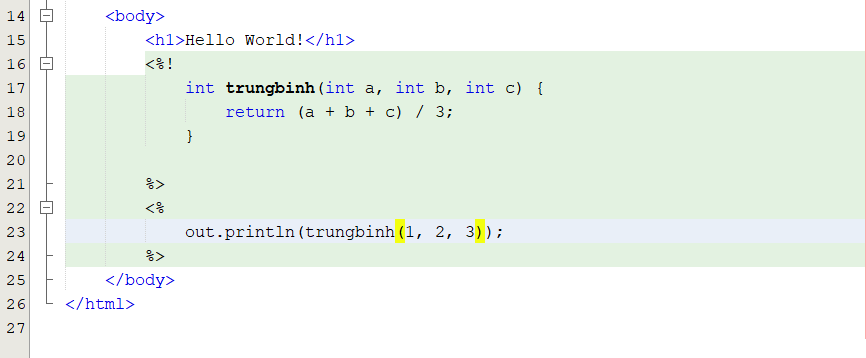
II. Phần thực hành: (đánh giá của mình là khá dễ miễn là code được)

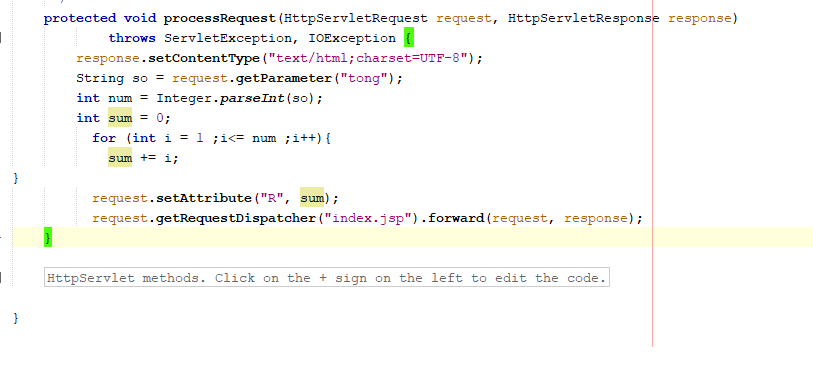
Đề mẫu:

1. Viết hàm tính trung bình cộng 3 số:

- Do là hàm nên dùng Delaration để code phương thức.

- Gọi hàm bằng Scriptlet( truyền thẳng đối số cho mình, kinh nghiệm)



2. Tính tổng dãy n số ( chắc từ 1 đến 10 )

+Một RequestDispatcher là một class vô cùng quan trọng trong Java. Nó giúp cho trang web chuyển request từ servlet này sang một servlet khác bằng cách cho phép gọi một servlet khác từ trong một servlet khác. Nó tạo ra một đối tượng nhận request từ client và gửi chúng đến bất kỳ resource nào như một servlet, HTML hay JSP file.

forward(), servlet container sẽ kích hoạt bất cứ resource nào (servlet, html file hoặc jsp file) mà mapping với url của RequestDispatcher . Giả sử resource là một servlet khác, thì servlet được gọi tới có thể truy xuất request như servlet gọi nó, và sẽ tạo response như servlet hiện tại.

1. Máy chủ Web nhận HTTP request

2. Máy chủ Web chuyển tiếp yêu cầu tới servlet container

3. Servlet được lấy ra một cách tự động và tải lên địa chỉ không gian của container, nếu nó nằm trong container.

4. Container gọi init () method của servlet để khởi tạo (được gọi một lần khi servlet được load lần đầu tiên)

5. Container gọi service () method của servlet để xử lý HTTP request, tức là đọc dữ liệu trong yêu cầu và hình thành một response

6. Web server trả lại kết quả động đúng với vị trí yêu cầu

**Phan Biet Forward va send redirect**

+Forward phương pháp chuyển tiếp yêu cầu là chuyển đến tài nguyên khác **trong cùng một máy chủ** để tiếp tục xử lý.

Trong trường hợp yêu cầu là  sendRedirect là chuyển hướng đến tài nguyên khác như tên miền khác hoặc các đường dẫn là các máy chủ khác nhau để xử lý.

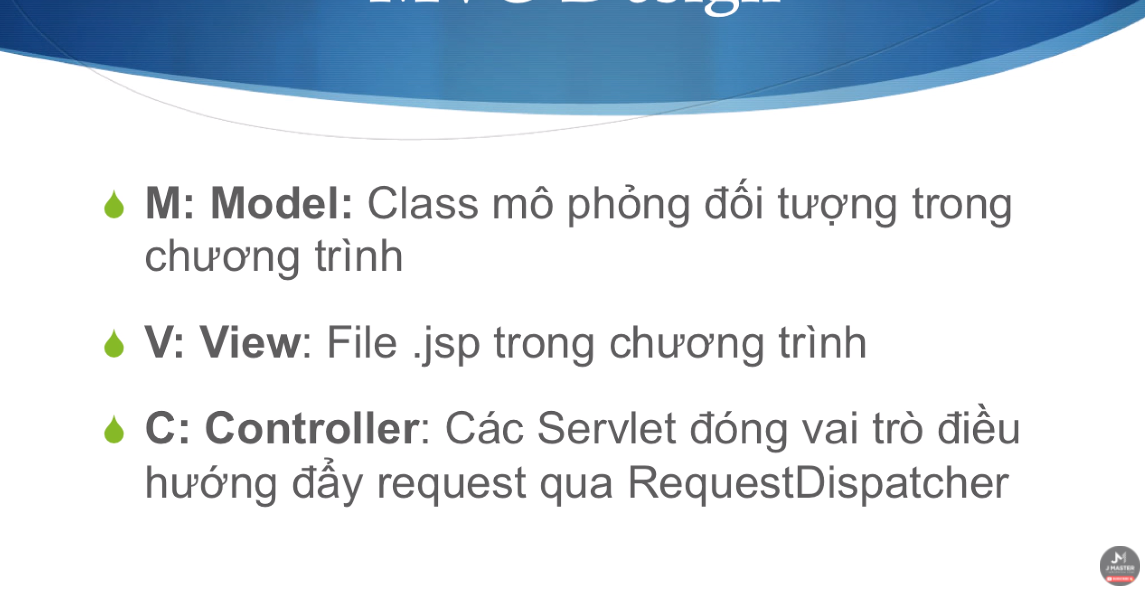
Foward Nhìn bề ngoài chúng ta không thể thấy địa chỉ chuyển tiếp như vậy sẽ bảo mật hơn.

SendRedirect Trong thanh địa chỉ chúng tôi có thể thấy địa chỉ chuyển hướng mới.

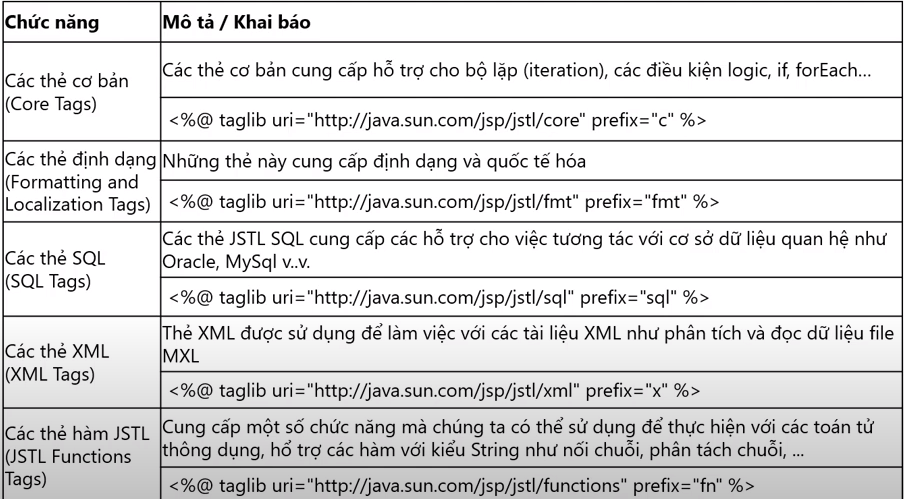
Sử dụng phương pháp chuyển tiếp *forward()*  thì tốc độ chuyên tiếp đến trang xử lý là nhanh hơn.

SendRedirect lnó chuyển hướng  chậm hơn bởi vì một chuyến đi vòng thêm là cần thiết . yêu cầu mới được tạo ra cho nên tốc độ xử lý sẽ chậm hơn so với *forward*

* getParameter()trả về tham số yêu cầu http. Những người được truyền từ máy khách đến máy chủ. Ví dụ http://example.com/servlet?parameter=1. Chỉ có thể trở lạiString
* getAttribute()chỉ dành cho sử dụng phía máy chủ - bạn điền yêu cầu với các thuộc tính mà bạn có thể sử dụng trong cùng một yêu cầu. Ví dụ: bạn đặt một thuộc tính trong một servlet và đọc nó từ một tệp JSP. Có thể được sử dụng cho bất kỳ đối tượng, không chỉ chuỗi.



**JSTL**

****

**I. Lý thuyết.**

1. Phân loại Web động và Web tĩnh:

Web động

=> Để tạo ra nội dung HTML động tùy thuộc vào sự kiện hoặc yêu cầu từ phía client.

Trang web có khả năng thay đổi được nội dung hiển thị: ví dụ điều kiện tìm kiếm khác nhau hiện được kết quả dữ liệu khác nhau, qua thời gian khác nhau thì dữ liệu hiển thị khác nhau, …

Web tĩnh

=>Hiển thị cố định nội dung HTML, CSS

Nội dung trang web là cố định, không có khả năng them hoặc thay đổi nội dung

2. Web server:

Là máy chủ cài đặt các chương trình phục vụ ứng dụng web, thực thi code của trang web.

Web server có khả năng nhận request từ browser, gửi và phản hồi đến client thông qua giao thức HTTP( hypertext transfer protocol).

3. Servlet Container:

Là nơi chứa tất cả servlet, là 1 phần của web server, nó dùng để sử lí web động, giúp server nhận request từ client và tính toán và sử lí những request đó.

Gọi servlet đáp ứng như cầu của người dùng.

\*Là thùng chứa servlet là nơi chứa servlet

Là môi trường là ứng dụng web server để chứa ứng dụng web

Khi viết xong web java sẽ deploy lên Scontainer (nó là tomcat ) để thực thi web viết bằng java

Những code java trong công nghệ web jsp là gọi servlet khi deploy chạy trên servlet container

***(nó là tomcat )***

*Servlet container chịu trách nhiệm cho việc tạo, thực hiện và hủy của servlet.*

4. Web.xml

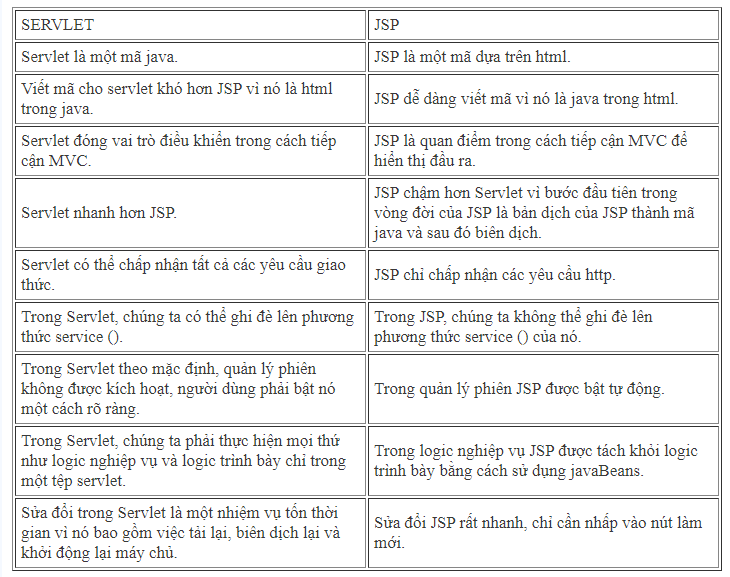
Ánh xạ đường dẫn url và servlet xử lí yêu cầu với các đường dẫn đó.

Cung cấp thông tin triển khai và cấu hình cho các thành phần của web như bộ lọc, tham số khởi tạo, tài nguyên, trang chào mừng.

~~Url~~mapping : <servlet-mapping> là phần tử con của web.xml và ánh xạ cho servlet

<url-pattern> là phần tử con của <servlet-mapping> và sử dụng cho phía máy khách để gọi servlet, hay nói cách khác là url tương ứng cho servlet.

5. Phân biệt JSP VÀ SERVLET



6. WEB CONTAINER

Là thành phần của máy chủ web tương tác với các servlet Java. Một vùng chứa web quản lí vòng đời của các servlet. Nó ánh xạ một URL tới một servlet cụ thể trong khi đảm bảo người yêu cầu có quyền truy cập liên quan.

7. Vòng đời của servlet

Servlet: là công nghệ tạo ra ứng dụng web, 1 thành phần của web được triển khai trên server để tạo một trang web động. Đóng vai trò trung gian giữa request từ client và ứng dụng hoặc cơ sở dl.

Vòng đời Servler gồm 3 bước:

- Khởi tạo init()

- Thực thi service() - Xử lí yêu cầu của người dùng.

- Hủy destroy() - Khi undeploy web app hoặc shutdown server.

=> init() thực thi một lần duy nhất khi servlet được nạp đầu tiên.

=> Thực thi phương thức service() để xử lí yêu cầu của người dùng. Có thể thực thi nhiều lần.

Vòng đời JSP gồm 4 bước:

- Biên dịch JSP: Phân tích cú pháp, Chuyển đổi JSP thành Servlet, Biên dịch Servlet

- Khỏi tạo jspInit()

- Thực thi \_jspService()

- Hủy jspDestroy()

8. JSP LÀ GÌ

JAVA SERVER PAGES

Công nghệ phát triển trang web động, giúp chèn code java vào trang html bằng cách dùng các thẻ jsp đặc biệt.

JSP là một kiểu JAVA SERVLET được thiết kế để tạo ra giao diện người dùng cho ứng dụng java web.

Các loại thẻ trong jsp:

- Comment <%- comment -%>

- Declaration <%! declaration %>

Khai báo hàm, hằng số, biến.

- Expression <%= expression %>

In ra biểu hiện ngôn ngữ

- Directive <%@ directive %>

Chèn file, thư viện, language, session, errorPage

- Scriptlet <% scriptlet %>

Dung de code java tren jsp

9. Attribute

Là đối tượng để chia sẻ thông tin trong ứng dụng web.

Thuộc tính cho phép servlet chia sẻ thông tin với nhau có thể thiết lập và get trong phạm vi:

- Request

- Session

- Application

Truyền tải dữ liệu FORM (Handle data)

* Trên trang JSP có 1 form (ví dụ form login).
* Trong form khai báo thuộc tính action là serverlet sẽ nhận request.
* Form có các input khai báo name tương ứng (username, password).
* Khi người dùng chọn submit, trang sẽ điều hướng gửi request đến servlet được khai báo trên action của form, tạo danh sách parameters có tên và giá trị theo các thuộc tính name và value của input.
* Trên servlet đọc dữ liệu trong ds parameters bằng cách dùng lệnh request.GetParameters(name).
* Sự khác nhau giữa các phương thức (method) truyền tải dữ liệu: GET truyền trên URL của header, giới hạn ở những dữ liệu nguyên thủy (số, chuỗi, luận lý, …) và độ dài tham số, tham số truyền trên URL nên người dùng và bên ngoài nhìn thấy, có ý nghĩa SEO và giữ nội dung filter. POST truyền trong thẻ body, không giới hạn kiểu dữ liệu và độ dài cho nên không có giới hạn trong truyền tải dữ liệu, nhưng không sử dụng được trong SEO và filter.

10. SO SÁNH METHOD GET VÀ POST

Điểm chung là các HTTP method dùng để trao đổi dữ liệu giữa client và server.

Điểm khác nhau:

* POST: Bảo mật hơn GET vì dữ liệu được gửi ngầm, không xuất hiện trên URL
* GET: Dữ liệu được gửi tường minh, chúng ta có thể nhìn thấy trên URL, đây là lý do khiến nó không bảo mật so với POST.
* GET thực thi nhanh hơn POST vì những dữ liệu gủi đi luôn được webbrowser cached lại.
* Khi dùng phương thức POST thì server luôn thực thi và trả về kết quả cho client, còn phương thức GET ứng với cùng một yêu cầu đó webbrowser sẽ xem trong cached có kết quả tương ứng với yêu cầu đó không và trả về ngay không cần phải thực thi các yêu cầu đó ở phía server.
* Đối với những dữ liệu luôn được thay đổi thì chúng ta nên sử dụng phương thức POST, còn dữ liệu ít thay đổi chúng ta dùng phương thức GET để truy xuất và xử lý nhanh hơn.

//cc

If you are only concern against stored browser history, Proxy Logs, Proxy Cache, and people looking at your URLs, then use POST.

11. 4 LOẠI SCOPE TRONG JSP

-Page scope: có sẵn trong page

-Session scope: có khi người dùng kết nối tới trang web, data tồn tại cho đến khi session bị huỷ or do time out

-Request scope: có thể sử dụng và truy cập cho đến khi response được tạo ra.

-Application scope: có thể được truy cập từ request, session khác nhau khi client truy cập vào trang ứng dụng, và chỉ bị huỷ khi app bị shutdown.

\* Phần này có thể yêu cầu xác định thực thể trên IDE ( vd như page scope là những biến khai báo trong thẻ scriptlet nó chỉ tồn tại trên trong thẻ, khi ra ngoài nó sẽ mất,...)

12. LÀM SAO ĐỂ NHẬN BIẾT CONTROLLER

- Khái niệm servlet controller

- Xác định servlet controller trên IDE( ví dụ trong NETBEAN IDE là nằm trong phần server như tomcat….)

13. MÔ HÌNH MVC

Viết tắt bởi Model – View – Controller. Mô hình thiết kế sử dụng kĩ thuật phần mềm. Mô hình source code chia thành 3 phần với chức năng và nhiệm vụ riêng biệt

Model chứa thông tin dữ liệu và nghiệp vụ logic, phương thức xử lý, truy xuất database

View: thành phần hiển thị, hiển thị dữ liệu ra màn hình, cho phép người dùng tương tác;

Controller: đón nhận và điều hướng yêu cầu từ người dùng, thực hiện xử lí, hỗ trợ kết nối model và view, giúp model xác định view tương ứng để trình bày.

MVC 1: JSP đóng vai trò là view và controller, sử dụng model là bean ( Có nhiều cách gọi( mô hình 1 vì nó không tuân theo mô hình mvc, không phân chia code làm 3 phần với nhiệm vụ và chức năng riêng biệt)

MVC 2: Servlet là controller và Jsp là view – cho phép phân chia code từ các logic điều khiển lường để thực hiện.

Trong phần này menter có thể yêu cầu xác định từng phần model – view – controller trên IDE và chỉ ra sự khác biệt giữa hai mô hình.)

14. JAVA BEAN

Một lớp java được xây dựng một cách đặc biệt trong java và được mã hóa theo javabeans api specifications

Cung cấp constructor mặc định không tham số

Có thể xếp thứ tự, triển khai serializable interface

Có đặc tính đọc và viết

Có phương thức getter và setter

\* Nên xem kĩ vì có thể dùng nó để nêu ra lợi ích của thẻ JSTL ( giúp tối ưu việc code, code nhanh hơn, giúp xem trang jsp với hoàn toàn mã html và thẻ giống html)

15. API

Là phương thức giao thức kết nối với các thư viện và ứng dụng khác. API cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó có thể trao đổi dl giữa các ứng dụng.

16. MÔ HÌNH 3 LAYER

Presentation layer(GUI)

Business logic layer(BLL)

Data Access layer(DAL)

\* Tìm hiểu thêm – chưa thấy mentor hỏi qua

17. JAVA BEAN

Là các lớp java thuần, dùng để thể hiện các thực thể trong chương trình java.

Đặc điểm:

- Là một public class

- Có các thuộc tính private

- Ứng với mỗi thuộc tính, lớp sẽ cung cấp một cặp phương thức getter/setter

19. JSTL

Là thẻ thư viện chuẩn cung cấp các thẻ để quản lí hành vi của trang như lặp đi lặp lại, các lệnh điều khiển, các thẻ quốc tế hóa, các thẻ SQL

- Core Tags:

Các thẻ cơ bản cung cấp hỗ trợ cho bộ lặp (iteration), các điều kiện logic, bắt ngoại lệ, url, chuyển tiếp (forward) hoặc chuyển hướng (redirect),.... Để sử dụng JSTL core tags, bạn cần đưa nó vào trang JSP giống dưới đây:

- Formatting and Locallzation tags:

Những thẻ này cung cấp định dạng cho các con số, ngày tháng và hỗ trợ i18n thông qua miền địa phương (locales) và bó tài nguyên (resource bundles).

- SQL tags:

Các thẻ JSTL SQL cung cấp các hỗ trợ cho việc tương tác với cơ sở dữ liệu quan hệ như Oracle, MySql v..v. Sử dụng thẻ SQL mà bạn có thể chạy các truy vấn cơ sở dữ liệu

- XML tags:

Thẻ XML được sử dụng để làm việc với các tài liệu XML như phân tích cú pháp XML, chuyển đổi dữ liệu XML và XPath đánh giá biểu thức.

- JSTL function tags:

Thẻ JSTL cung cấp một số chức năng mà chúng ta có thể sử dụng để thực hiện các toán tử thông dùng, hầu hết trong số đó là cho String như nối chuỗi, phân tách chuỗi, ...:

***\* Lợi ích:***

Cho phép xem chương trình hoàn toàn bằng mã HTML và mã gần giống HTML

Đơn giản hóa việc code, code nhanh hơn.

20. Cookie

Là các text file được lưu trữ trên máy tính phía client. Được sử dụng cho mục đích theo dõi các thông tin đa dạng. Server sử dụng cookie để nhận dạng người dùng trong lần truy cập tiếp theo.

\* Sẽ có câu hỏi nhận dạng web động kiểu web skype phải web động không? Đặc điểm nhận dạng?

Web động web tĩnh

Jsl là gì

So sáng jsp vs svl

Mô hình mvc

Web sẻver...container

So sánh post vs get

* jsp là gì
* sự khác nhau giữa servlet và servlet container
* jsp khác html ở đâu
* jsp servlet dựa trên mô hình gì
* model trong mvc là gì
* tính tổng các số từ 1 – 100 bằng jsp