**JAVA BEAN and POJO**

POJO – Plain Old Java Object; it has private properties, getters and setters for respective properties. It may also override Obejct.toString() and Object.equals().

Java Bean – same with POJO but must implement either Serializable or Externalizable and must have a no-arg constructor

**WEB SERVICE**

Definition:

• Web-Service can be considered as a set of methods that enables communication amongst applications irrespective of the application's coding language or framework.

• Web services are really nothing more than a request/ response mechanism that allows a web application to interact with other applications

• Web Service is any piece of software that provides some functionality that is available over the internet/network (via HTTP, SMTP, FTP, etc.) using open standards (XML, SOAP, HTML, etc.).

Example:

I can go to maps.google.com, and type in my home address, and see a map of where I live in my browser.

But what if you were writing a computer program where you wanted to take an address and show a pretty map, just like Google maps?

Well, you could write a whole new mapping program from scratch, OR you could call a web service that Google maps provides, send it the address, and it will return a graphical map of the location, which you can display in your program.

There is a lot more to it, as some of the other posts go into, but the upshot is that it allows your application to either retrieve information FROM, or submit information TO some resource. Some other examples:

**1.**You can use a web service to retrieve information about books at Amazon.com

**2.**You can use a similar web service to submit an order to Amazon.com

**3.**You could CREATE a web service to allow outside applications to find out about product information within your company

**4**.you could create a web service to allow outside applications to submit orders to your company.

Benefits:

• Exposing the Existing Function on the network - Web services allows you to expose the functionality of your existing code over the network. Once it is exposed on the network, other application can use the functionality of your program.

• Interoperability - Web services allow various applications to talk to each other and share data and services among themselves.

• Standardized Protocol - Web services use standardized industry standard protocol for the communication. All the four layers (Service Transport, XML Messaging, Service Description, and Service Discovery layers) use well-defined protocols in the web services protocol stack.

• Low Cost of Communication - Web services use SOAP over HTTP protocol, so you can use your existing low-cost internet for implementing web services.

SOAP (Simple Object Access Protocol)

Web Service takes the help of SOAP to transfer message.

RESTFUL

**MUTABLE AND IMMUTABLE OBJECT**

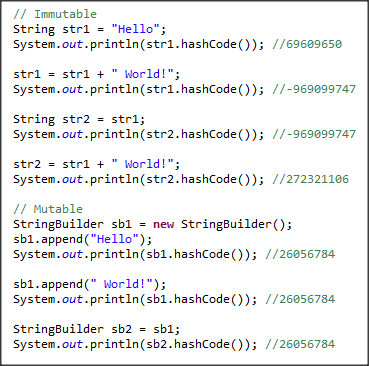
Definition:

• Mutable – changeable; object whose state CAN be updated directly after it is created

• Immutable – unchangeable; object whose state CANNOT be altered after it is created

Example:

**String is immutable**, when you are trying to modify a String, what you are really doing is creating a new one. However, when you use a **StringBuilder, which is mutable object**, you are actually modifying the contents, instead of creating a new one



\* If the memory allocated in heap is immutable type, then the value hold by that memory allocation can’t be changed thus creating a new memory space in heap.

**PASS BY REFERECE and PASS BY VALUE**

UML

**SDLC**

- stands for Software Development Life Cycle

- is a methodology that defines the steps to follow, by a software industry, in developing a software or application.

It is made up of phases that is used to ensure the effective and efficient delivery of work or product, aiming to produce a high quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates.

It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software.

\**methodology*: a system of ways/methods of doing, teaching, or studying a particular area of study or activity

**Benefits**:

● The intent of an SDLC process is to help produce a product that is cost-efficient, effective and of high quality.

● The SDLC process provides Information Technology (IT) project managers with the tools to help ensure successful implementation of systems or applications that satisfy strategic and business objectives.

**Phases**:

Phases of SDLC can vary from project to project or again depends on the client’s aptitude towards the solution one looking for.

**1. Planning / Requirement Gathering and Analysis**

Basically this is the step where client come to you with own idea and will ask you to provide the estimate for the same. It produces a high level overview document of the project which relates to the project requirements and scope.

In this stage, A developer need to gather complete requirement. Once requirement gathering is done, developer needs to perform the analysis i.e. how the solution will be implemented, what will be the solution, which all technologies will be used etc. Once analysis is done, Estimation will be prepared and provided to client.

**2. Design**

Once requirement is clear, Design document will be prepared. Design document generally known as Requirement Specification / Software design specification. It describes desired features and operations in detail, including screen layouts, business rules, process diagrams, pseudocode and other documentation.

Design doc will contain all the activity involved in conceptualizing, framing, implementing, commissioning, and ultimately modifying complex systems" or "the activity following requirements specification and before programming.

**3. Develop**

Development is a stage where idea/design is being converted in actual code.

Here all the coding is being performed, and Unit testing is also being performed. Unit testing is a testing which is performed by developer to confirm all the functionality working correctly.

**4. Testing**

Testing is being performed by tester. This is the stage where all the functionality will be tested. Application will be connected to actual backend and will be tested completely. If all went good then only code is ready to deploy.

**5. Deployment and Maintenance**

Once the customers start using the developed system and in future if the client faces some issues, maintenance is done by the organization who has developed it.

\*This process of taking care for the developed product is known as maintenance.

**CODE REVIEWS**

Meaning:

Benefits:

Steps:

HOW DOES SPRING WORK??

EAGER and LAZY

SNAPSHOT and RELEASE

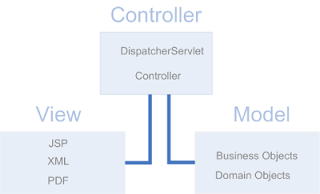
CRUD

CRUD is an acronym for Create, Read, Update, and Delete

POC

**EXTENDS THREAD or IMPLEMENTS RUNNABLE**

**SPRING MVC**

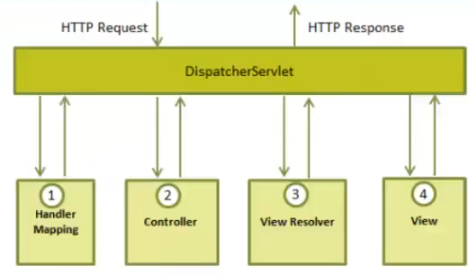


In Spring MVC DispatcherServlet and Controllers act as the Controller and they receive the requests and decide how the request will be served and also decide which view is used. Business objects and domain objects are the Model and the business objects are invoked by the controllers and the all data are in domain objects. The domain objects will be passed to some JSP which are the View part. The JSP will render the data in the domain objects.

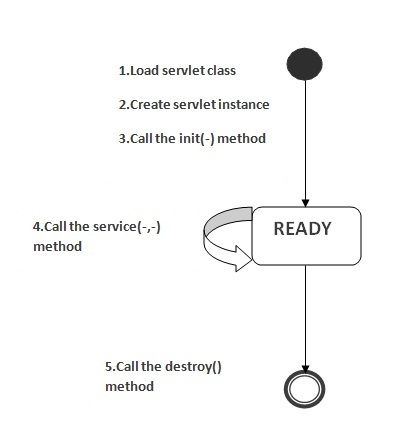
**DisptacherServlet**

DispatcherServlet is the front controller in Spring MVC.   It is a HttpServlet that will receive the request and return the response.  DispatcherServlet is the key player in SpringMVC.  From the below brief work flow of SpringMVC you can see DispatcherServlet is the driving force that make the request served with the right response sent back to the client.    

1. Receive the request from client
2. Consult HandlerMapping to decide which controller processes the request
3. Dispatch the request to the controller
4. Controller processes the request and returns the logical view name and model back to DispatcherServlet
5. Consult ViewResolver for appropriate View for the logical view name from Controller
6. Pass the model to View implementation for rendering
7. View renders the model and returns the result to DispatcherServlet
8. Return the rendered result from view to the client



**LIFE CYCLE OF SERVLET**



**Servlet class is loaded**

The classloader is responsible to load the servlet class. The servlet class is loaded when the first request for the servlet is received by the web container.

**Servlet instance is created**

The web container creates the instance of a servlet after loading the servlet class. The servlet instance is created only once in the servlet life cycle.

**init() method is invoked**

The init method is designed to be called only once. It is called when the servlet is first created, and not called again for each user request. So, it is used for one-time initializations, just as with the init method of applets.



**service() method is invoked**

The service() method is the main method to perform the actual task (client requests). The servlet container calls the service() method to handle requests coming from the client and to write the formatted response back to the client.

Each time the server receives a request for a servlet, the server spawns a new thread and calls service. The service() method checks the HTTP request type (GET, POST, etc.) and calls doGet, doPost, etc. methods as appropriate.



**destroy() method is invoked**

The destroy() method is called only once at the end of the life cycle of a servlet. This method gives you an opportunity to clean up any resource for example memory, thread etc.

After the destroy() method is called, the servlet object is marked for garbage collection.



**ENTERPRISE and WEB APPLICATION**

**APPLICATION and WEB SERVER**

**HTTP REQUEST METHODS**

**doGet() -** shall be used when small amount of data and insensitive data like a query has to be sent as a request

**doPost() -** shall be used when comparatively large amount of sensitive data has to be sent. Examples are sending data after filling up a form or sending login id and password.

In doGet(), the parameters are appended to the URL and sent along with header information. This does not happen in case of doPost(). In doPost(), the parameters are sent separately. Since most of the web servers support only a limited amount of information to be attached to the headers, the size of this header should not exceed 1024 bytes. doPost() does not have this constraint.

**final, finally, finalize**

**Final** is a keyword in java, its uses differ depending on where we use it. Final class can't be inherited, final method can't be overridden and final variable value can't be changed.

**Finally** is placed together with try catch, it will be executed whether exception is handled or not.

**Finalize** is used to perform clean up processing just before object is garbage collected.

**throw, throws, throwable**

**throw** is a keyword in java which is used to throw an exception manually. Using throw keyword, you can throw an exception from any method or block. But, that exception must be of type java.lang.Throwable class or its sub classes.

**throws** is also a keyword in java which is used in the method signature to indicate that this method may throw mentioned exceptions. The caller to such methods must handle the mentioned exceptions either using try-catch blocks or using throws keyword.

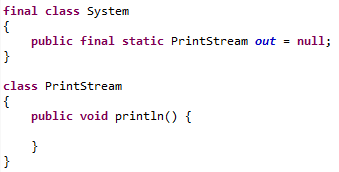
**Throwable** is a super class for all types of errors and exceptions in java. This class is a member of java.lang package. Only instances of this class or its sub classes are thrown by the java virtual machine or by the throw statement.

**System.out.println()**

**System** is a final class from the java.lang package.

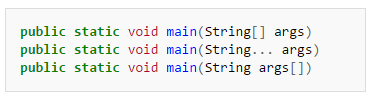
**out** is a class variable of type PrintStream declared in the System class.

**println** is a method of the PrintStream class.



**public static void main (String [] args) {}**

This is an entry point for program execution.



**public** : is an Access Modifier. Public means that this Method will be accessible by any Class.

Unless the method is declared public, Java runtime classes (inside rt.jar) wont be able to invoke the main method resulting in NoSuchMethodError:main at runtime

**static** : is a keyword which identifies the class related thing. This means the given Method or variable is not instance related but Class related. It can be accessed without creating the instance of a Class.

When the JVM makes call to the main method there is no object existing for the class being called therefore it has to have static method to allow invocation from class.

**void** : is used to define the Return Type of the Method. It defines what the method can return. Void means the Method will not return any value.

Java is platform independent language and hence it cannot return a value to the operating system and expect consistency across all operating systems.

**main**: is the name of the Method. This Method name is searched by JVM as a starting point for an application with a particular signature only.

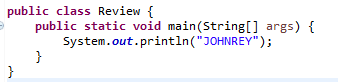
This name is fixed and as it's called by the JVM as entry point for an application.

**String args[]** : is the parameter to the main Method.

These are the arguments of type String that your Java application accepts when you run it.

While running the program if we want something to be passed on to the main method then this parameter is used an we can pass the parameters while calling the program. args is the name of an array of type String, so we can pass some strings while running the program if want.

Let’s say you have written following program in java language:



So now when you compile this using javac command it will create the class called Review.class. And when you run it using java command i.e. “java Review arg1 arg2 arg3”it will find the signature of main method. that is ‘public static void main(String[])’ in the Review.class and calls it i.e java Review.main(arg1,arg2,arg3).

In the end as the main method is the entry point of the program it will be called from outside of the class so it must be “public”. As the compiler calls it without creating its object it must be “static”. No return type so it is “void. “main” is the name. The last but not the least “String[] args” is the array of string if you want to take any user input at the time of running the program.

If you don’t specify the “public static void main(String[])” in your program, it can be compiled successfully but cannot be run directly ( unless you have plans to call its methods from other class that you can run.