**What is final in java?**

Final is keyword that is applicable only to variables, methods, or a class.

When a variable is declared as final, after we initialize the value, we cannot reassign value for it.

When a method is declared as final, we cannot override that method.

When a class is declared as final, we cannot extend that class.

**Interface and Abstract Class**

|  |  |
| --- | --- |
| **Interface** | **Abstract Class** |
| In an interface we can have only abstract methods. | In an abstract class we can define both abstract and non-abstract methods. |
| In an interface by default methods are public and abstract. |  |
| A class can implement any number of interfaces. | A class can extend only one abstract class. |
| The interface keyword is used to declare interface. | The abstract keyword is used to declare abstract class. |
| Abstract class can provide the implementation of interface. | Interface cannot provide the implementation of abstract class. |
| Abstract class doesn’t support multiple inheritance. | Interface supports multiple inheritance. |
|  |  |

**Servlet and Jsps**: provides support for dynamic response and data persistence

**JSP** is also server-side technology. It is used for creating dynamic web content. Basically, it’s like HTML with additional features to add dynamic content where we need it. It is advanced version of Servlet technology. JSP helps developers to insert java code in HTML pages by making use of special JSP tags. JSP provides additional functionality such as expression language, custom tags etc.

**WebServer**: A machine which can take in client request over the web(internet) and give back a static response back to the client (static HTML page, jar file etc)

**Web Container:** Servlets/JSPs have the capability to develop dynamic pages (pages whose content/layouts changes based on the request). Servlets/JSPs live within the Web Container (i.e. Web Container manages the Servlets/JSPs). The WebServer communicates with the Web Container to generate dynamic web pages.

**Web Container**

Tomcat is a web container, when a request is made from Client to web server, it passes the request to web container and it’s web container job to find the correct resource to handle the request (servlet or JSP) and then use the response from the resource to generate the response and provide it to web server. Then the webserver sends the response back to the client.

When web container gets the request and if it’s for servlet then container creates two Objects HTTPServletRequest and HTTPServletResponse. Then it finds the correct servlet based on the URL and creates a thread for the request. Then it invokes the servlet service() method and based on the HTTP method service() method invokes doGet() or doPost() methods. Servlet methods generate the dynamic page and write it to the response. Once servlet thread is complete, the container converts the response to HTTP response and send it back to the client.

**JDBC API:**

JDBC is an API that has the collection of classes and interfaces written in the java programming language. JDBC API is used to create a connection between Java programming language and wide range of databases. Basically, it is used to create SQL statements, execute SQL queries in the database and view and modify the resulting records. So, we can say it is used to preform CRUD operations in database. Some of the classes and interfaces of JDBC API are : Connection, Prepared Statement, Statement, ResultSet,

**Servlet API:**

Servlet is an API that belongs to J2EE Architecture. Servlet resides on a server side and generates a dynamic web page. Servlets are used to handle the request obtained from the web server, process the request, produce the response and send the response back to the web server. Servlet has its different life cycle which defines the whole process of its creation till the destruction. It has different methods to handle different HTTP request such as: doPost(), doGet(). The life cycle of servlet is managed by web container such as Tomcat.

Web container(tomcat) will search for classes under classes folder and inside jars under lib folder,

**Life Cycle of Servlet:**

A life cycle of a servlet can be defined as the entire process from its creation till the destruction. Servlet follows a certain life cycle, and the life cycle of servlet is managed by the servlet container. Life cycle phases of servlet are:

1.) instantiation

2.) initialization

3.) servicing

4.) destruction

There are three states of servlet i.e., new, ready, and end.

Life cycle methods of servlet:

1. init() method
2. service() method
3. destroy() method

- The servlet is in new state if servlet instance is created. Servlet class is loaded by class loader when application is deployed and instantiated by the container.

- After invoking the init() method servlet comes to ready state. The init() method is called only once. The init() method simply creates or loads some data that will be used through the life cycle of servlet. This method is a good place to create db connection.

- Servlet performs all the task in ready state. The web container calls the service method each time when request for the servlet is received. The service() method is the main method to perform the actual task. The servlet container (i.e. web server) calls the service() method to handle requests coming from the client( browsers) and to write the formatted response back to the client. Each time the server receives a request for a servlet, the server creates a new thread and calls service. The service() method checks the HTTP request type (GET, POST, PUT, DELETE, etc.) and calls doGet, doPost, doPut, doDelete, etc. methods as appropriate.

- When the web container invokes the destroy() method, it reaches to the end state. The destroy() method is called only once at the end-of-life cycle of servlet. The destroy() method gives opportunity for servlet to clean up any resources for example memory, thread, close connection etc. After destroy method is called, the servlet object is marked for garbage collection.

**What is HttpServlet?**

HttpServlet is a java class that extends the GenericServlet class. HttpServlet class is slightly more advanced class than the GenericServlet class. HTTP servlet class handles http request and provides an HTTP response, usually in the form of HTML page.

Differences between get and post method:

**GET**

Doget method is used for getting the information from the server. Doget method is designed to get response context from the web resource by sending limited amount of input data, this response contains response header and response body.

Meant for getting data/retrieving/fetching data

No Body

Query String

No Sensitive Data

Restriction on Data

Idempotent: It doesn't change the application state

No matter how many times we execute the get method

**POST**

doPost method is used for sending the information to the server. doPost method is designed to send unlimited amount of data along with the request to web resource.

Meant for posting data

Body Present

Payload/Body

Sensitive data can be sent

No Restriction

Non-Idempotent: It changes the application state, if we update/or multiple times there is chance of duplication

**How do we create Servlet?**

There are three ways to create Servlet:

1.) By implementing Servlet interface

2.) By inheriting Generic class

3.) By inheriting HttpServlet class

HttpServlet is the most widely used servlet nowadays as it provides methods such as doPost(), doGet(), doHead(), doPut, doDelete etc.

The important methods of Servlet are:

**1.) public void init(ServletConfig config)**

initializes the servlet. It is the life cycle method of servlet and invoked by the web container only once.

**2.) public void service(ServletRequest request,ServletResponse response)**

provides response for the incoming request. It is invoked at each request by the web container.

3.) **public void destroy()**

Is invoked only once and indicates that servlet is being destroyed

What is the differences between sendRedirect() and fordward() method?

The forward() method is executed in the server side whereas sendRedirect() method is executed in the client side.

**Spring Framework**: Spring is a powerful lightweight, loosely coupled framework that is used to develop Java Enterprise application. Spring follows Dependency Injection design pattern that makes code loosely coupled. Spring provides IoC container, which is responsible for creating the objects, wire them together, configure them, and manage their complete life cycle from creation till destruction.

It is also called as frameworks of frameworks because it provides supports for various other frameworks like hibernate, structs etc. Spring framework comprises several modules such as, AOP, ORM, WEB MVC, Security etc.

What are the advantages of Spring framework?

* Spring promotes loose coupling by providing support for Dependency injection.
* Spring is highly modular in nature. We have an option to use the entire spring framework, or we can just use the necessary spring module for our project.
* Spring eliminates the creation of singleton and factory classes.
* Spring provides flexible configuration. Developer has the option to choose either XML or java-based annotation for configuration purposes.
* Testing is easier on application designed using spring framework.

**What are features of Spring?**

The main features of Spring Framework are :

* Lightweight
  + Spring Framework is very lightweight due to its POJO implementation. It does not force to inherit any class or implement any interfaces.
* Aspect-Oriented Programming
  + Spring support aspect-oriented programming which is used for separating cross-cutting concern. For example, logging, security etc. from the business logic of the application.
* Transaction Management
  + Helps in handling transaction management of an application without affecting its code.
* IoC Container
  + Spring provides IoC container, which is responsible for creating the objects, wire them together, configure them, and manage their complete life cycle from creation till destruction.
* Dependency Injection
  + We can design loosely coupled application with the help of Spring.
* Integration with Other Frameworks
  + Another beautiful thing about Spring framework is it does not try to solve the problem that is already solved.  It just tries to integrate them with it’s framework.
* **Flexible Mapping** - It provides the specific annotations that easily redirect the page.

**Strategies of Spring**:

Lightweight and minimally invasive development with POJOs

Loose coupling through Dependency Injection and interface orientation

Eliminating boilerplate code with aspects and templates

**Spring Beans**: Beans is an object whose life cycle is managed by IOC container.

What does Application Context do?

The application context bean is built on top of the Bean Factory interface. They read the configuration metadata and uses it to create a fully configured application. It will load the container, scans the beans in the package.

What is the difference between Bean Factory and Application Context ?

They are the interfaces that acts as the IoC container. The application context bean is built on top of the BeanFactory interface. They read the configuration metadata and uses it to create a fully configured application.

1) BeanFactory Container is basic container, it can only create objects and inject Dependencies. But we cannot attach other services like security, transaction ,messaging etc. To provide all the services we have to use ApplicationContext Container.

2) BeanFactory Container doesn't support the feature of AutoScanning , but ApplicationContext Container supports.

3) Beanfactory Container will not create a bean object upto the request time. It means Beanfactory Container loads beans lazily. While ApplicationContext Container creates objects of Singleton bean at the time of loading only. It means there is early loading.

4) Beanfactory Container support only two scopes (singleton & prototype) of the beans. But ApplicationContext Container supports all the beans scope.

**Dependency Injection**: Dependency Injection is a design pattern that implements IoC principle. It is considered as the fundamental aspect of Spring framework. Dependency is an Object needed for another object to accomplish certain task or a goal and Dependency Injection is the process of providing/supplying the dependencies needed for an object. And the spring core module is responsible for injecting dependencies through either constructor or setter method for resolving dependencies of objects so that it will be easy to manage and test the application. In dependency injection, instead of object having to configure itself, they are configured by external entity/framework. Some of the benefits of IOC (dependency injection) are:

* Reduced Dependencies
* Code is loosely coupled
* Reusable and maintainable code
* More testable code
* Code is clean and readable

**Different types of IOC:**

* Constructor Injections: Dependencies are provided as constructor parameters.
* Setter Injection: Dependencies are provided through Java bean properties.

**IOC Container**: IOC Container can be defined as the framework for implementing dependency injection. The IOC container creates an object of the specified class and injects all the dependencies through a constructor, or method. Spring provides IoC container, which is responsible for creating the objects, wire them together, configure them, and manage their complete life cycle from creation till destruction.

**Loose coupling**: When two classes, modules, or components have low dependencies on each other, it is called loose coupling in Java. Loose coupling in Java means that the classes are independent of each other. Changing one class should not affect the other. So, loose coupling basically means they are mostly independent., they can interact but have very little knowledge of each other.

When the objects are loosely coupled there is a flexibility to switch between different dependencies.

Spring MVC

* **Model** - A model contains the data of the application. A data can be a single object or a collection of objects.
* **Controller** - A controller contains the business logic of an application. Here, the @Controller annotation is used to mark the class as the controller.
* **View** - A view represents the provided information in a particular format. Generally, JSP+JSTL is used to create a view page. Although spring also supports other view technologies such as Apache Velocity, Thymeleaf and FreeMarker.
* **Front Controller** - In Spring Web MVC, the DispatcherServlet class works as the front controller. It is responsible to manage the flow of the Spring MVC application.

HTML

<!DOCTYPE> : It is the tag that is the first line of code required in every HTML document. The DOCTYPE declaration is an instruction to the web browser about what version of HTML the page is written in.

The five most important tag in HTML are:

1. <div> tag

The div tag is known as division tag. The div tag is used in HTML to make divisions of content in the web page like (text, images, header, footer, navigation bar, etc) where we can put more than one HTML element group the together and can apply CSS. It is most usable tag especially in web development because it helps us to separate out data in web page and with the help of div tag we can create a particular section for particular data in web page. The use of the div tag in HTML elements increases the code readability and is used for organizing content. All the related items are placed in one div container.

2.)<a> tag

<a> tag defines the hyperlink, which is used to link from one page to another. The most important attribute of <a> tag is href that indicates the link’s destination. The links can be in the form of image, email address, phone number

3.)<img> tag

The image tag is also the most important tag in HTML which is used to fix an image in an HTML page. Images are not technically inserted into a web page, rather they are linked where this <img> tag creates a holding space for the referenced image. The <img> tag has two required attributes: i.e. src and alt.

Src: It specifies the path to the image , while specifying the image we should always specify the height and weight, it makes the image clear and

Alt: Specifies an alternated image if the image for some reason cannot be displayed.

4.) form tag

The form tag is used to create an HTML form for user input. The two attributes of form tag are action and method. The action attribute specifies where to send the form-data when a form is submitted and the method attribute specifies how to send form data either to get or post. A form can contain textfields, checkboxes, radio-buttons, label, email, date, select, drop down, password. Input element is the most important form element. The <input> tag specifies an input field where the user can enter data. And we should always use lable tag to define labels for <input type="text">, <input type="checkbox">, <input type="radio">, <input type="file">, and <input type="password">.

5.) table tag

The <table> tag defines a table. And the table in HTML consists of table cells inside rows and columns. We have <tr> tag in table that indicates the rows in the table where we can put <th> and <td> tag. We can have any number of rows we want but every time the number of cells should be same in each row. <th> is another important tag that in table tag that indicates the header of the cell. By default, the text in <th> elements are bold and centered but we can make changes using CSS. Each table cell is defined by <td> and we define the content of the cell in between <td> tags.

JavaScript

DOM: Each of the document of HTML tags are represented as an object, there is the hierarchy. DOM is the representation of same HTML document in a different format. The reason behind this is that JavaScript cannot understand the tags, it can easily understand the format, objects. JavaScript use this object to manipulate their attributes.

Three basic nodes of JavaScript are:

1.) Element Node

2.) Attribute Node

3.) Text Node

Diagram

Description automatically generated

DOM Document Object Model is a platform that allows programs and scripts to dynamically access and update the content, structure, and style of a document. With HTML DOM, javascript can access and change all the elements of an HTML document. With dom javascript can create a dynamic HTML, change the HTML and css elements, add, remove, elements and attributes. HTML DOM is a standard for how to get, change, add or delete HTML elements. In DOM, all the html elements are defined as objects.

Different methods to find HTML element:

1. Document.getElementById Method: this method is used to access and HTML element by the use of id of the element.
2. Document.getElementByTagName: this method is used to access HTML element by use of tag name
3. document.getElementsByClassName:is used to find the element by class name

Maven:

Maven is a powerful build tool that is used for Java software projects. Maven is centered around the concept of POM files (Project Object Model). A POM file is an XML representation of project resources like source code, test code, dependencies (external JARs used) etc. The POM contains references to all these resources. The POM file should be in the root directory of the project it belongs to.

Build tool is a tool that automates everything related to building the software project including generating a source code, generating documentation from the source code, compiling source code, packaging compiled code into JAR files or zip files, installing the packaged code on a server, in a repository somewhere. The main advantage of using this build tool is it is typically faster, and the other hand the use of build tool can minimize the risk of human errors while building the software manually.

War File:

War file is a file that contains files of a web project that may have servlet, jsp, xml, image, html, css etc.

**Maven:**

Maven is a build automation project management tool that will compile the source code, run the unit test cases and when all the test cases are passed it will bundle all the compiled classes into jar or war file (depending upon the type of application: if it is standalone application it will bundle into a jar, or if it is a web application it will bundle into a war file) and deploy the application. So, the four most important steps of Maven are: compile, run, package and deploy.

**Ant vs Maven**

In short, an ant script **tells the ant tool what to do** - "compile these files and then copy them to that folder. Then take the contents of this folder and create an archive."

While a maven pom declares **what we would like to have as the result** - "here are the names of the libraries the project depends upon, and we would like to generate a web archive". Maven knows how to fetch the libraries and where to find the source classes on it's own.

The ant scripts are not reusable while the Maven plugin are reusable.

Ant uses imperative approach, while Maven uses declarative approach.

While ant gives you more flexibility, it also forces you to constantly reinvent the wheel.

Maven on the other side requires less configuration, but may feel too constraining, especially if you are used to a different workflow.

In this simple Ant example, you can see how you have to tell Ant exactly what to do. There is a compile goal which includes the javac task that compiles the source in the src/main/java directory to the target/classes directory. You have to tell Ant exactly where your source is, where you want the resulting bytecode to be stored, and how to package this all into a JAR file. While there are some recent developments that help make Ant less procedural, a developer's experience with Ant is in coding a procedural language written in XML.

Contrast the previous Ant example with a Maven example. In Maven, to create a JAR file from some Java source, all you need to do is create a simple pom.xml, place your source code in ${basedir}/src/main/java and then run mvn install from the command line. The example Maven pom.xml that achieves the same results. That's all you need in your pom.xml. Running mvn install from the command line will process resources, compile source, execute unit tests, create a JAR, and install the JAR in a local repository for reuse in other projects. Without modification, you can run mvn site and then find an index.html file in target/site that contains links to JavaDoc and a few reports about your source code.

Ant

* Ant is a Java-based build tool. It is used in:
* Compiling the code
* Packaging the binaries
* Deploying the binaries to the test server
* Testing the changes
* Copying the code from one location to another

Ant scripts are written in XML file, where we should explain each and everything about the project like:

Hibernate:

Object Relational Mapping: is a technique where we map the java classes to entities or tables that helps hibernate to deal with data persistence and fetch all data automatically and transforms java object to records or records to java objects. Java are based on Object Oriented concepts and Database are based on relational concepts. Both these concepts are different and working with both is difficult, data representation is different (in java data are store in the form of object whereas in, DB {data, are stored in form of records/tuples) there is a paradigm mismatch the way data is represented. So, ORM concept can be used for converting data between relational databases and object-oriented programming language.

Hibernate:

Hibernate is a framework that follows ORM concept that acts as a bridge between java applications and database servers. Hibernate takes care of the mapping from java classes to database tables, vice versa. The limitations of JDBC API are addressed using Hibernate framework. Using Hibernate framework we don’t have to write queries, hibernate framework provides the query itself, so, we can say our application is database independent. Using Hibernate framework we can reduce redundant codes like establishing connection, creating prepared statement object, validating the results, writing the codes to transform into objects and so on. So, the use of hibernate framework simplifies the development of java application to interact with the database.