J2EE ARCHITECTURE
And
DESIGN PATTERN

Introduction

The J2EE platform provides runtime environment for developing and running large-scale, multi-tiered online/web and internet applications.

Design patterns give readymade explanations/solutions or templates to commonly occurring problems in the programming language.

There are many design patterns - MVC, Repository, Singleton, Factory Design patterns, etc.

WEB SERVER and WEB CONTAINER

Web Server is a server software that handles HTTP requests and responses to deliver web pages or web content to clients (i.e. web browser) using HTTP protocol. An example of web server is Apache HTTP Server.

Web Browser communicates with web server using the Hypertext Transfer Protocol (HTTP).

Hypertext Transfer Protocol (HTTP) is specially meant to communicate between Client and Server using Web (or Internet).

Web Container is a web server component that handles Servlets, Java Server Pages (JSP) files, and other Web-tier components. Web container is also called a Servlet Container or Servlet Engine.

It provides the run time environment to web applications. The most common web containers are Glassfish, Eclipse, JBOSS, Apache Tomcat, WebSphere and Web Logic.

The J2EE Platform

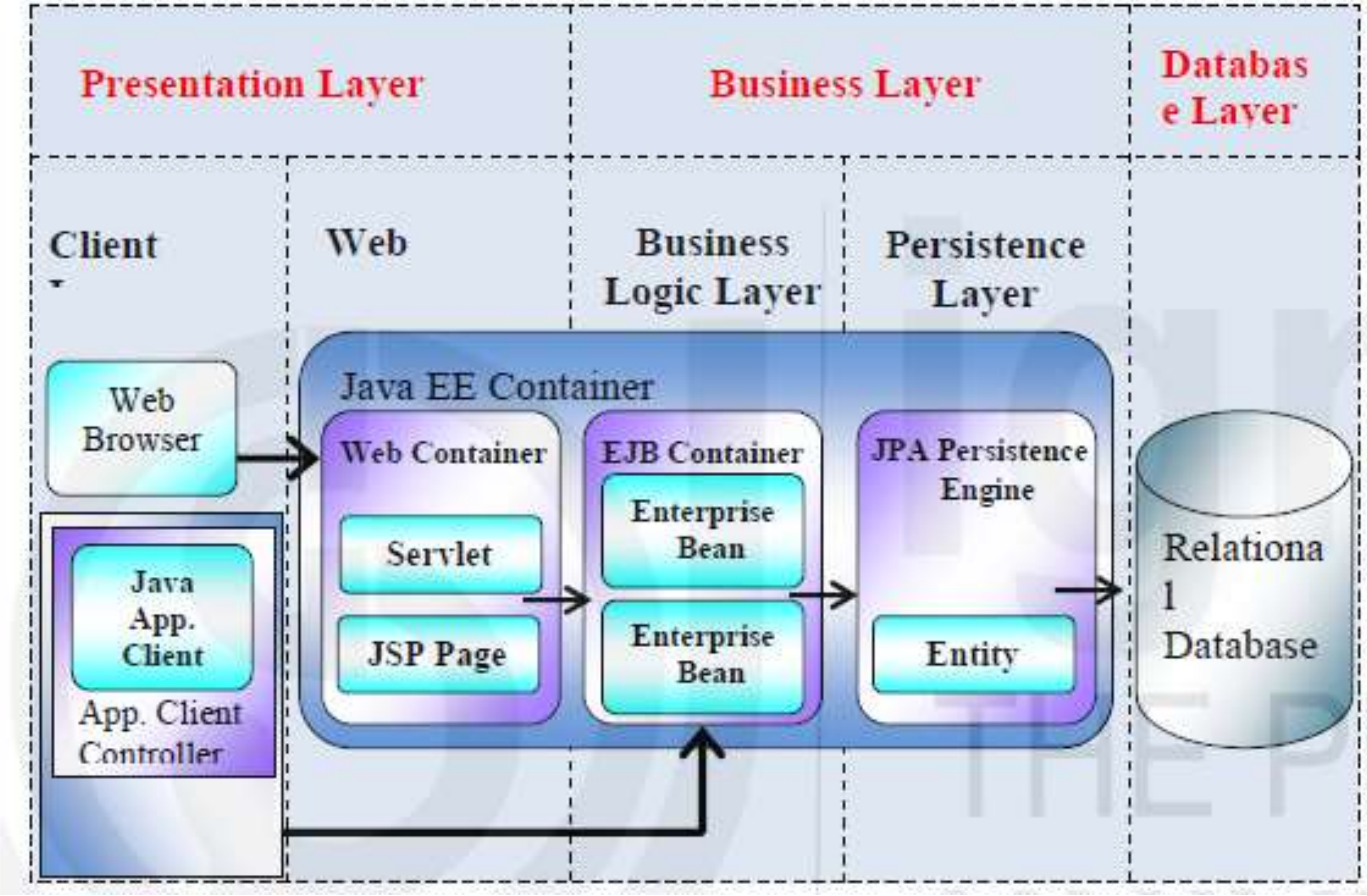
The J2EE platform is a set of services, application programming interfaces (APIs) and protocols.

J2EE is used to develop and deploy multi-tier web-based enterprise applications.

J2EE contains several APIs such as Java Servlets, Java Server Pages (JSP), Enterprise Java Beans (EJB), Java Database Connectivity (JDBC), Java Message Service (JMS), Java Naming and Directory Interface (JNDI), etc.

For creating complex applications, J2EE frameworks like Struts, Spring and Hibernate can be used.

J2EE Architecture



Design Patterns

Design patterns are the best solutions that are tested, verified developed prototypes that can speed up the development process.

There are 23 classic design patterns and they are categorized into three groups:

Creational, Structural and Behavioral.

The Creational design patterns deal with the creation of an object. Structural design patterns deal with the class structure, and the behavioral design patterns define the interaction between objects.

Model View Controller (MVC)

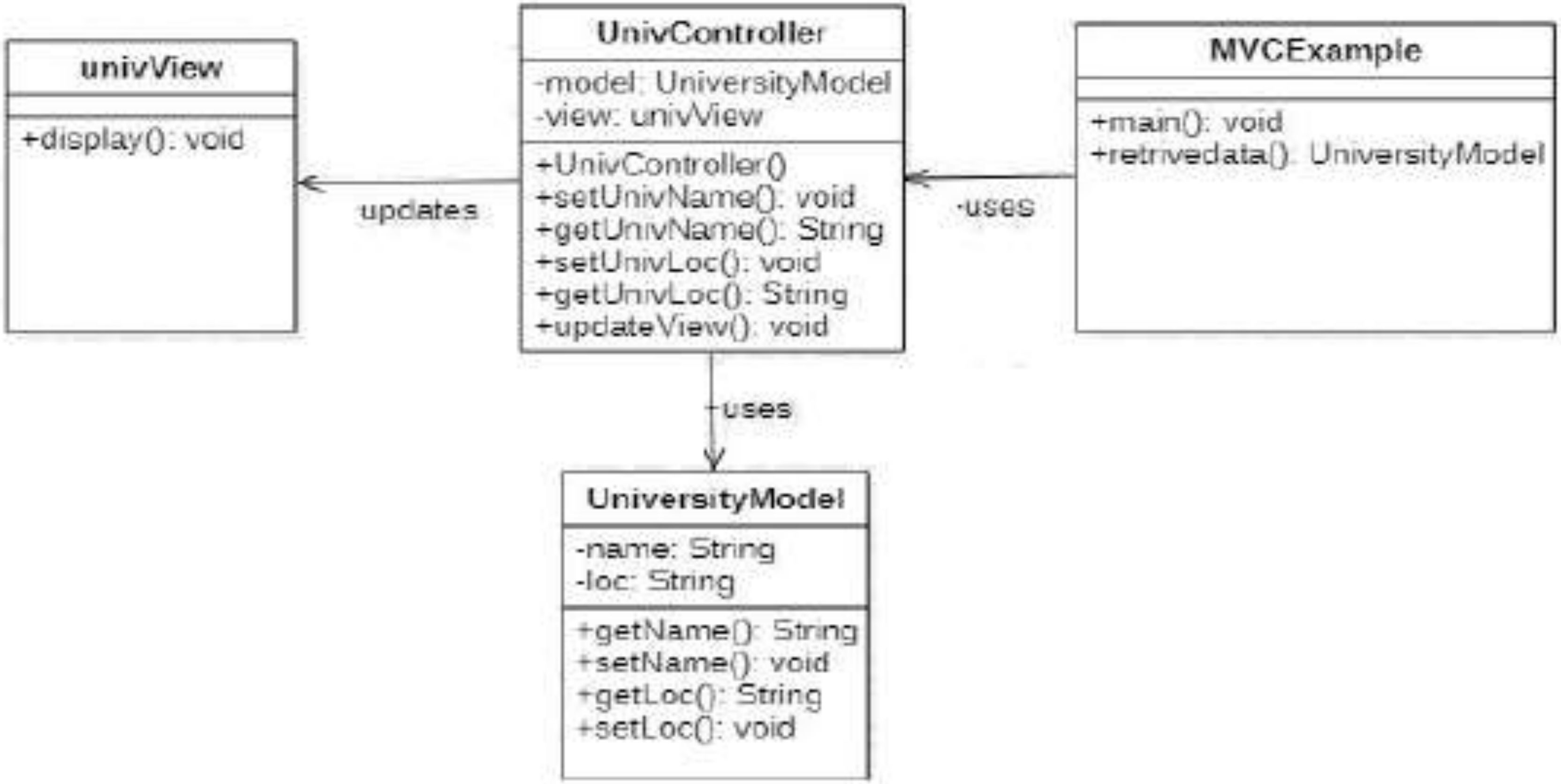
MVC design pattern is an architectural pattern for web applications. It can be used across many frameworks with many programming languages such as Java, PHP, Python, Ruby, etc.

This design pattern is extensively used to design enterprise web applications and mobile applications.

M (Model) denotes only the pure application data and does not contain any logic for representing data to a user, whereas V (View) is responsible for presenting data to the user.

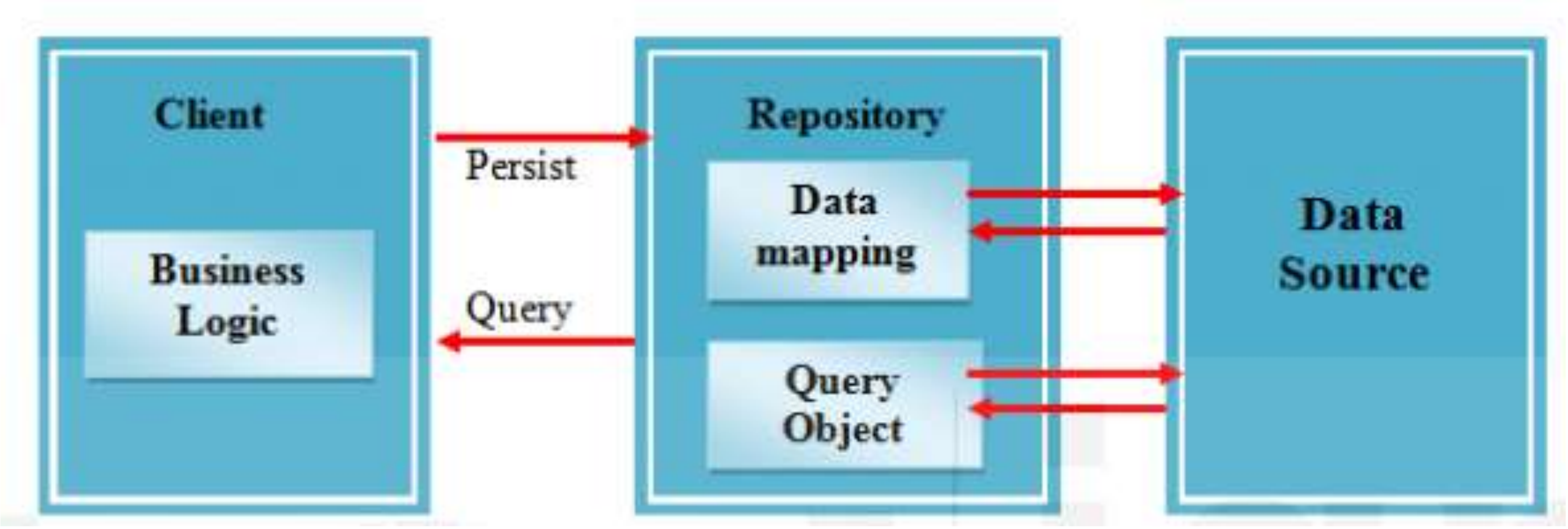
The C (Controller) comes between the view and the model layer and it is responsible for accepting a request from the user, performs interactions on the data model objects, and sends it back to the view layer.

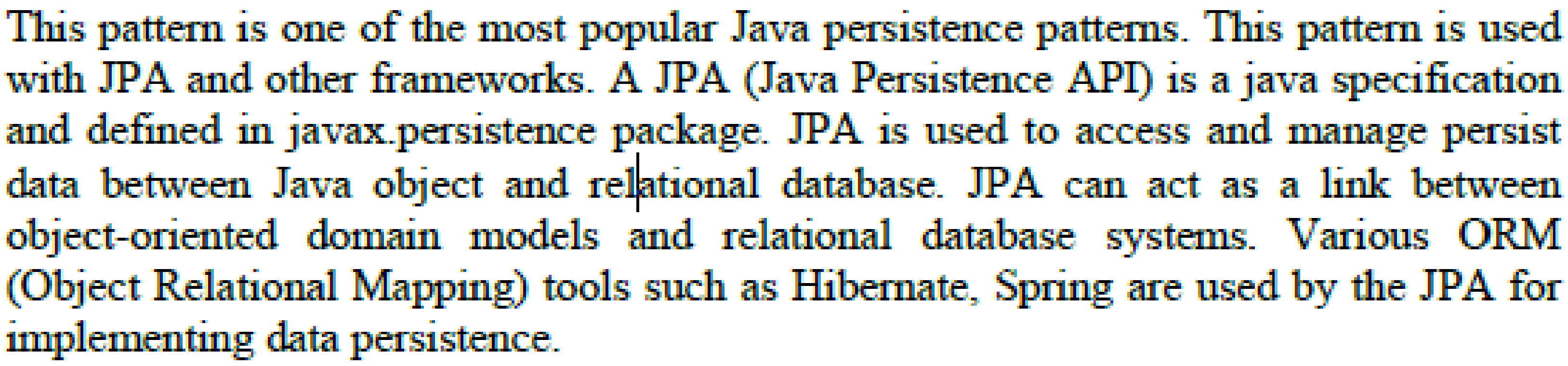
UML class diagram of MVC Design Pattern



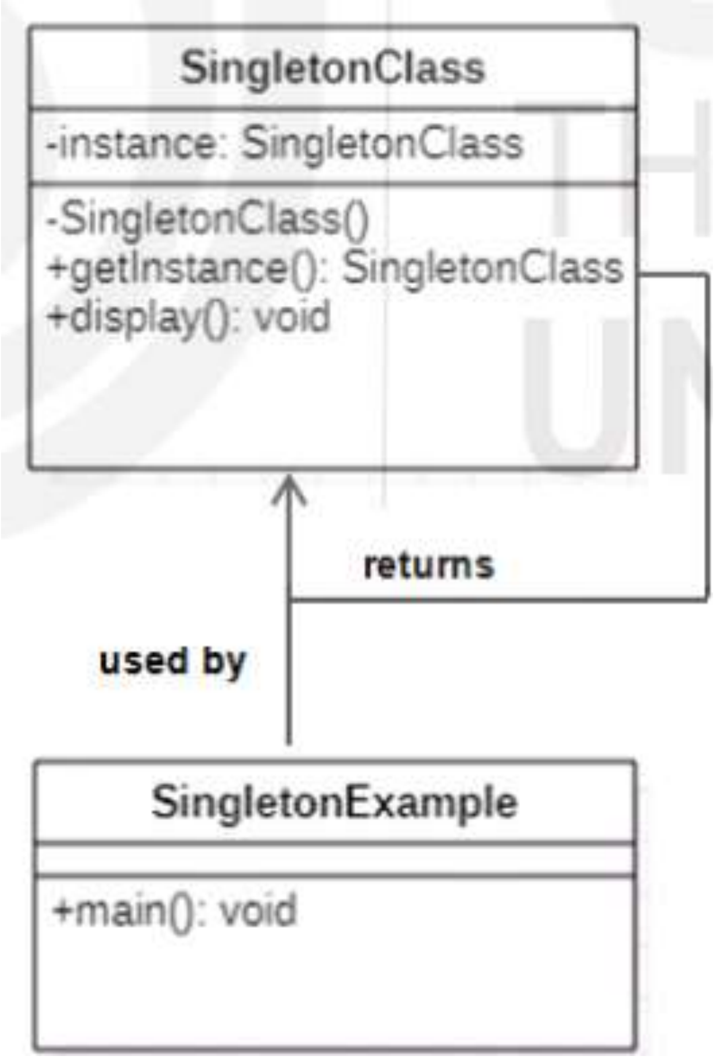
Repository Design Pattern

It is used in data-centric applications. It isolates the Business Logic Layer and Data Source Layer in the application.




This pattern is one of the most popular Java persistence patterns. This pattern is used with JPA and other frameworks. A JPA (Java Persistence API) is a java specification and defined in javax.persistence package. JPA is used to access and manage persist data between Java object and relational database. JPA can act as a link between object-oriented domain models and relational database systems. Various ORM (Object Relational Mapping) tools such as Hibernate, Spring are used by the JPA for implementing data persistence.

Singleton Design Pattern



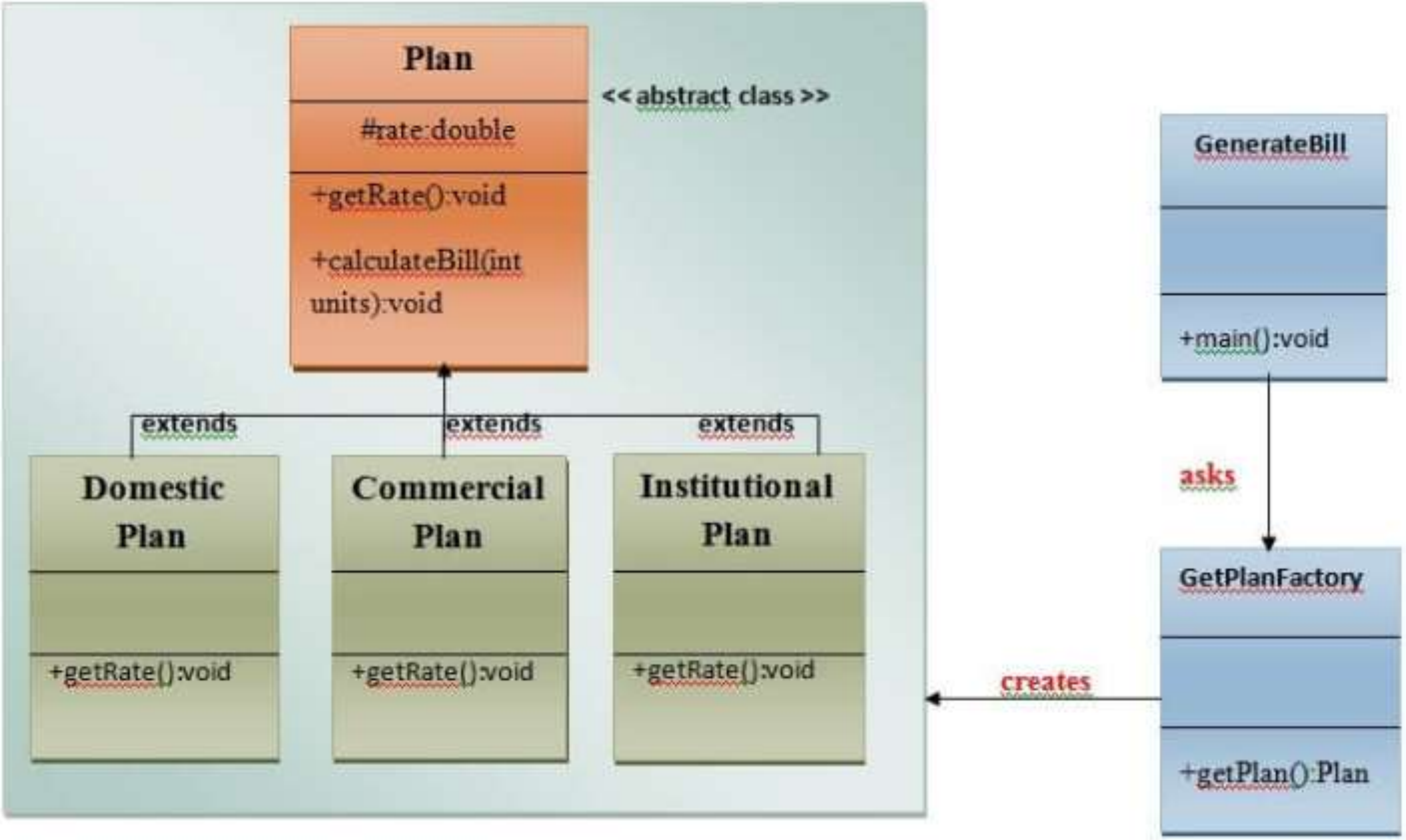
This pattern involves a single class which is responsible to create an object while making sure that only single object gets created.

This class provides a way to access its only object which can be accessed directly without need to instantiate the object of the class.

Mostly used in DB Connection & Multi-threaded Applications.

Factory Design Pattern

A Factory Pattern says that just **define an interface or abstract class for creating an object but let the subclasses decide which class to instantiate.** In other words, subclasses are responsible to create the instance of the class.

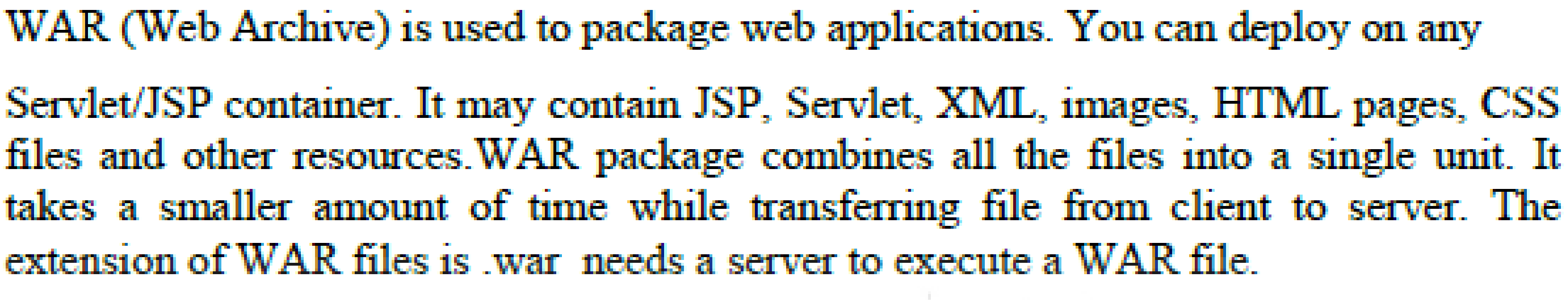


JAR and WAR Packaging

The JAR (Java Archive) is a package file format. The file extension of JAR files is .jar and may contain libraries, resources, and metadata files**.**

Basically, it is a zipped file containing the compressed versions of .class files, compiled Java libraries and applications. You can create a JAR file using the following jar command:

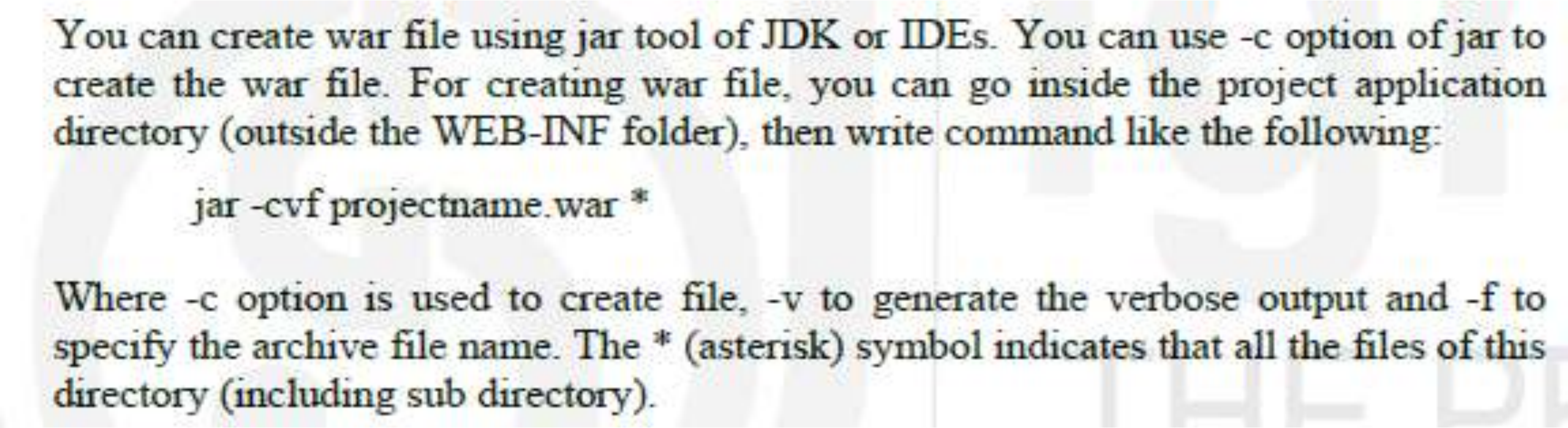
Jar cf name-jar-file input-file(s)



WAR (Web Archive) is used to package web applications. You can deploy on any

Servlet/JSP container. It may contain JSP, Servlet, XML, images, HTML pages, CSS files and other resources.WAR package combines all the files into a single unit. It takes a smaller amount of time while transferring file from client to server. The extension of WAR files is .war needs a server to execute a WAR file.

Creating and Deploying WAR File



You can create war file using jar tool of JDK or IDEs. You can use -c option of jar to create the war file. For creating war file, you can go inside the project application directory (outside the WEB-INF folder), then write command like the following:

```
jar -cvf projectname.war *
```

Where -c option is used to create file, -v to generate the verbose output and -f to specify the archive file name. The * (asterisk) symbol indicates that all the files of this directory (including sub directory).

you are using the tomcat server and want to deploy this file manually, go to the 'webapps' directory of apache tomcat and paste the war file. The server will extract the war file internally. Now, you are able to access the web project through a browser.

... Suppose, you are deploying project.war file. First go to tomcat webapps folder and paste it.

... Go to tomcat->bin folder and start tomcat by clicking startup.bat file

... Now, you can access your application from the browser. Open the browser and write in the address bar as localhost:port/projectname eg. localhost:8080/project

If you wish to extract the war file manually, then you need to use -x switch of jar tool of JDK. The following command is used to extract the war file.

```
jar -xvf projectname.war
```

Thank You....