**SPRING CORE**

Programming Language (ex: Java) we can develop desktop apps

Technologies ( Ex: Servlets & JSP, JDBC) we use to develop web application

Framework (Struts,Spring) we use to develop web/standalone/distributed applications , Provide comman logic.

==========================

What is Framework

=========================

=> Framework is a semi developed software

=> Frameworks providing common logics for application development

=> If we use frameworks then we need to focus only on business logic

=> Using frameworks we can do more work in less time (productivity)

=> We have several frameworks in java community

=> Reduce Boilerplate code

1) JSF -----------> outdated

2) Hibernate -----> data jpa

3) Struts -------> out dated

4) Spring ---> SpringBoot ----> trending

=> Hibernate is an ORM framework.

=> ORM means Object Relational Mapping

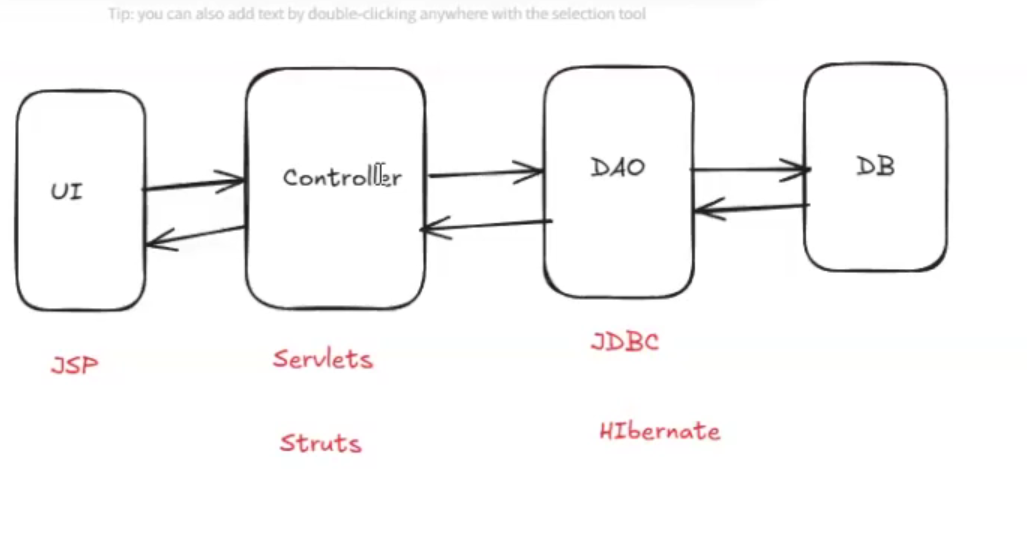
=> Using Hibernate we can develop persistence layer

=> Hibernate developed on top of JDBC.

=> Struts framework developed by Apache

=> **Struts is used to develop only web layer or controller**

in Struts Framework , There is not any provision for Database connectivity, So we used hibernate for this purpose.



=> Spring is called as **application development framework**

=> Using spring we can develop entire application including DB connectivity

=> Spring is free of cost & open source

=> Spring framework provides common logics required for application development.

Note: SpringBoot is an extension for Spring Framework.

===============

What is Spring

===============

=> It is java based framework developed by interface21 company.

Note: Now spring is under license of VMWare.

=> Spring is free and open source framework.

=> Spring is called as Application development framework.

Note: We can develop all the layers of application using spring framework.

=> Spring Framework developed by "Rod Johnson"

=> Spring 1.x released in 2004

=> The current version of spring is 6.x version

=================================

What we can develop using Spring

==================================

1) Stand-alone apps (desktop)

2) Web Applications (C2B)

3) Distributed Applications (B2B)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Spring Architecture

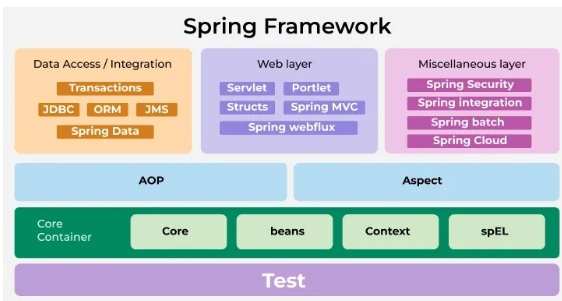
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

=> Spring developed in modular fashion

Spring 1.x => 7 modules

Spring 2.x => 6 modules (combined web mvc +mvc ) module

Spring 3.x to 6.x => 20+ modules....



Note: Spring is loosely coupled framework.

==================

Spring Modules

==================

1) Spring Core (IOC & DI)

2) Spring Context : Deals with configuration management

3) Spring AOP (Aspect Oriented Programming) : used to separate primary and secondary logic in the application

4) Spring JDBC / Spring DAO : used to communicate with Database with the help of text data.

5) Spring ORM (Object relational mapping) : convert database text format in java objects format, hibernate is first ORM framework

Spring ORM deals with Object data where as Spring JDBC deals with text data.

6) Spring Web MVC (C 2 B) :used to develop web application by using spring framework with C2B communication.

7) Spring REST (B 2 B) : used to develop distributed application with B2B communication

8) Spring Security (Authentication & Authorization)

9) Spring Cloud (Microservices)

10) Spring Batch (bulk operation) : Top process bulk operation : invoice generation, statement generation

11) Spring Schedular : Schedule batch operation

12) Spring Data JPA

Project Development

--Collection of programs

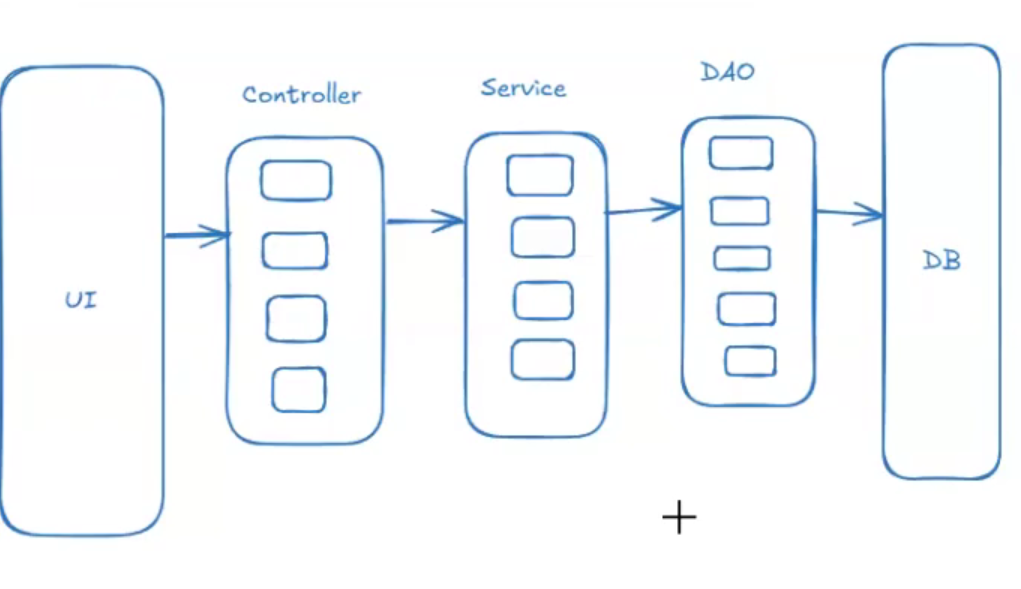
> We will develop several classes in the project

Pojo classes : transfer the data from one layer to another layer Plain old java object

Controller classes : to handle the request and response, Take the request from UI and provide the response back to UI.

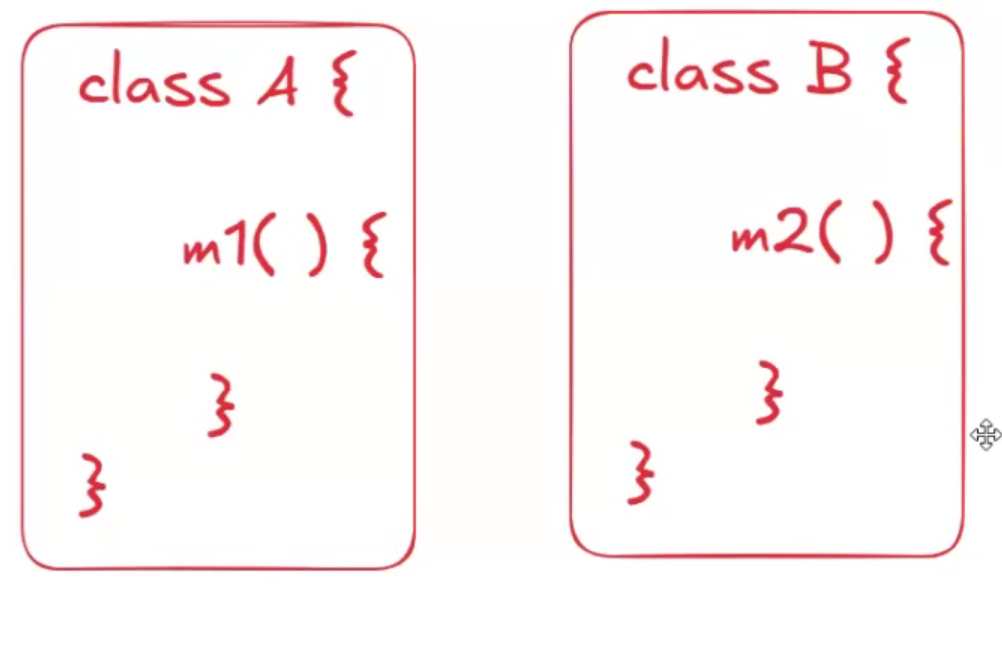
Service Classes : to handle business logic /operations email service,pwdService,

DAO classes : responsible to communicate with Data base table, for each table there is one DAO, Data Access Object



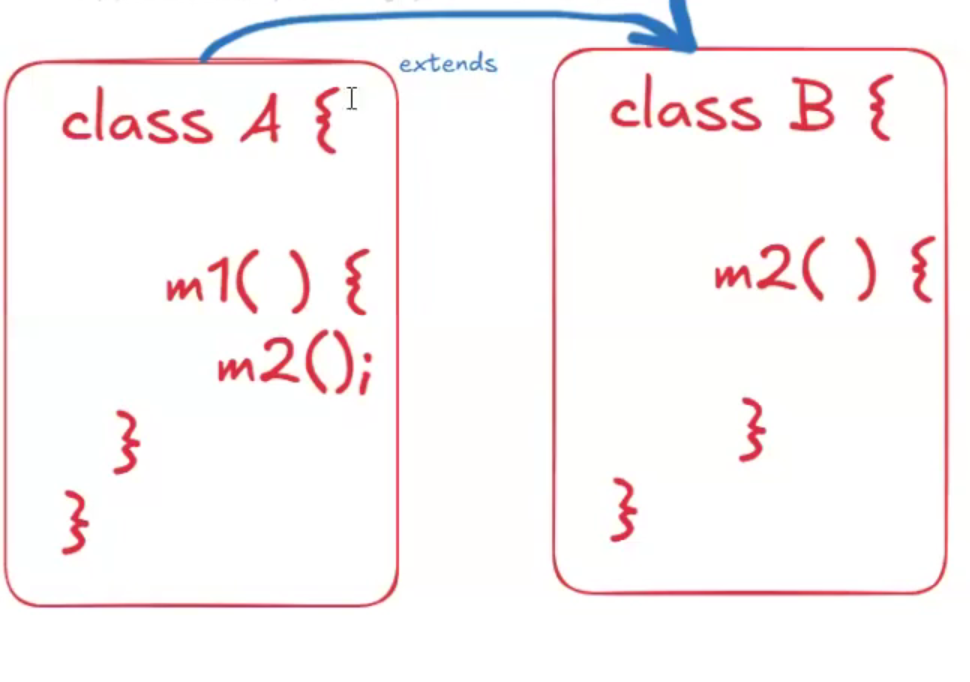
**How one java class method calls , another java class method ?**

How m1 method of class A calls m2 method of class B.



Inheritence : ( IS -A)

Claas A extends B. then



Limitation :

In future , if we want to use to access another class method (C) then multiple inheritance is not supported.

If some one make class B as final, then inheritance can’t be possible.

If any modification happen in class B, it will impact on class A, so we can say class B and Class A are tighly coupled.

So it is not recommended.

**Association ( HAS -A)**

Create the object of classes and call the methods.

If someone change in the constructor of class B.

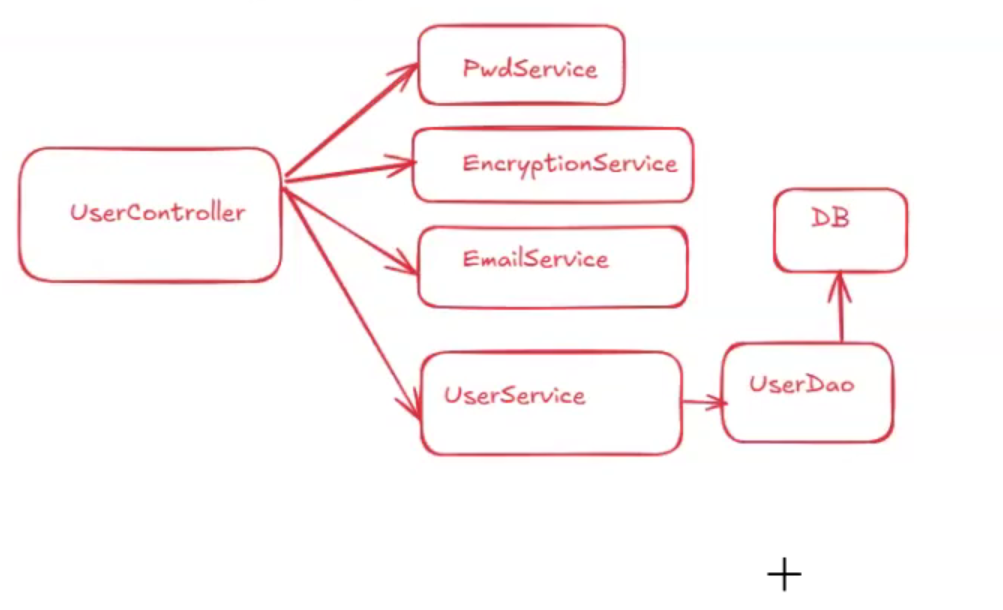
Changes as parameter constructor .

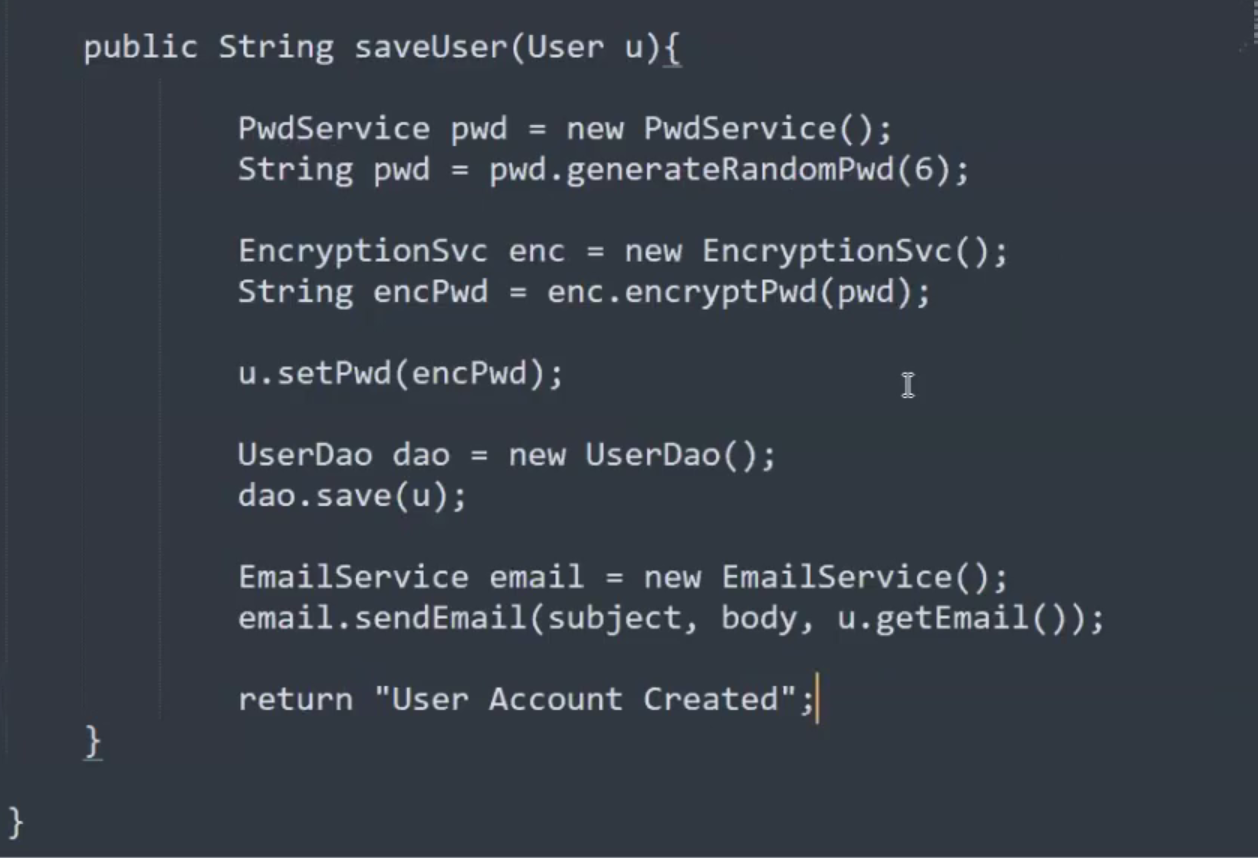
Make constructor as private

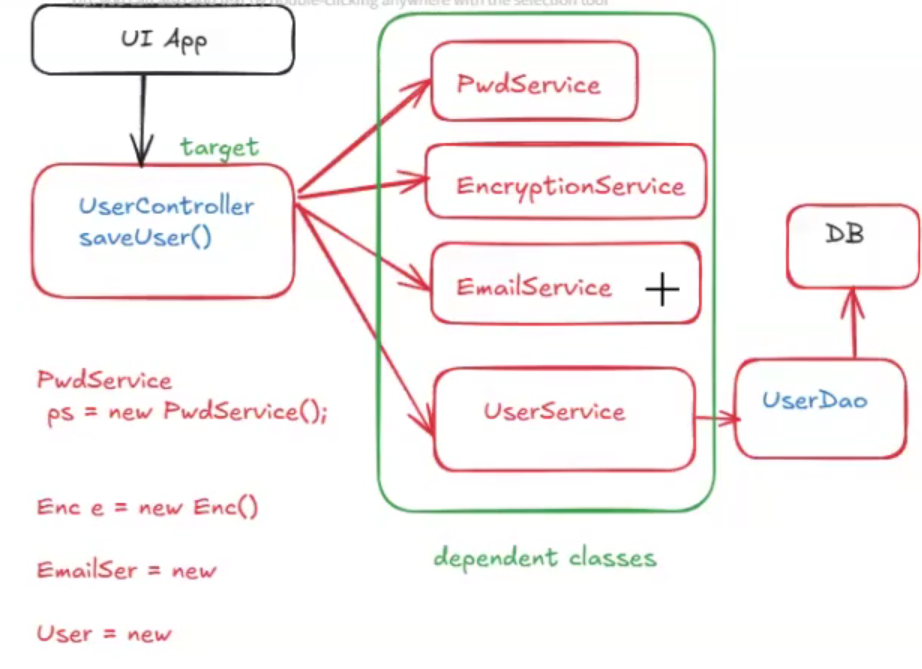
Classes will be tightly coupled.

So it is not recommended.

I want to call another class method without using inheritance and association , then spring core is come into picture. In loosely coupled manner. By using IOC and DI.







Any change in dependent classes will impact to target class, if we change the constructor default to parameterized then target class will not be compile. So IS-A and HAS-A relationship both are tightly coupled

I want my classes should be loosely coupled , if someone change in dependent classes there should not be change in target classes. So project development and maintenance will be easy.

IOC : Inversion of Control : Opposite, Instead of I am doing, some one is doing that task

Instead of programmer is creating the object , IOC container is creating the object and programmer will only use the object. Where programmer was creating the object there was issue of tight coupling and it was time consuming task. This creating of object is requirement of all, not for single one.

Third problem,Memory issue, like

PwdService pwd=new PwdService(); in our project , if I need to create the PwdService object 1000 times , programmer is creating the object then memory will be exhaust then you will receive out of memory issue. So let give this activity to IOC container which will perform this activity. So issue of Lossely coupled issue resolve and memory issue will be resolved.

The process of creating the object and giving the object by IOC container is known as dependency injection.

IOC Container : Inversion of Control :

* IOC will manage objects in the application with loosely coupling
* IOC will perform the dependency injection.

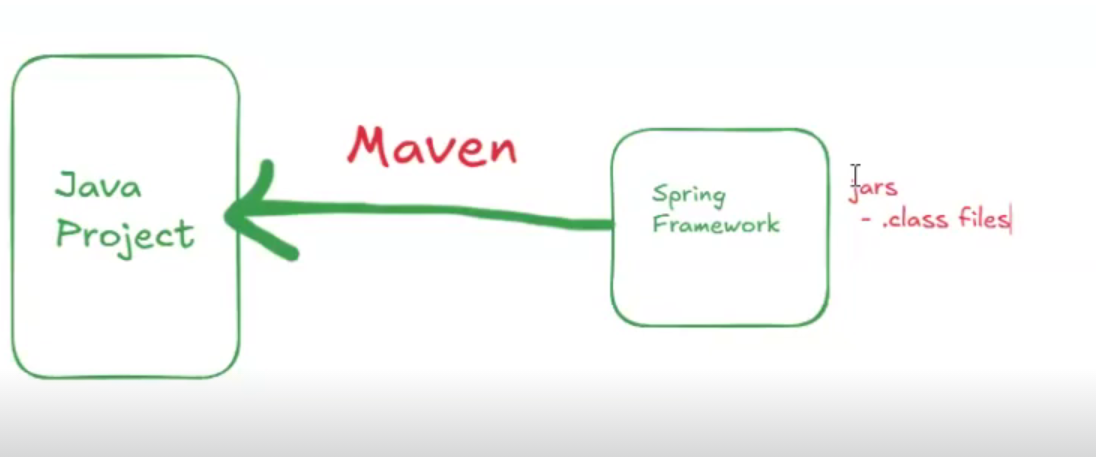
Dependency Injection :

* Create the dependent class object and inject it into target class object.

After this we don’t need to create the objects.

Download the spring JARs and add to your project class path and use it to develop business logics.

Currently we are using “Maven” or “Gradle” tool to download the JARs.



======

Maven

======

=> It is free and open source s/w.

=> It is developed by Apache org.

=> It is developed using java lang.

=> It is used as build tool for java projects.

=> Maven is used to automate "java projects build process".

===========================

What we can do using Maven

============================

1) We can create java projects skelton/structure.

2) We can download required libraries (jars)

Ex: spring, boot, hibernate, junit, poi, itext...

3) We can compile project source code ( At the time of deployment , eclipse is not there then such case maven works)

4) We can execute project unit test cases . (junits)

5) We can package application as jar or war file for execution.

==================

Maven Terminology

==================

Archetype : Represents type of project

quick-start : console app

web-app : web application

groupId : Represents company name

Ex : com.tcs

com.ibm

in.ashokit

ArtifactId : Represents project name

Version : Represents project version

SNAPSHOT : under development

RELEASE : Delivered to client

Dependencies : Represents libraries required for project

Ex : spring, spring-boot, hibernate, junit..

Maven Repositories : Represents the location of maven dependencies

1) Central (public) : anyone can access (mvnrepository.com)

2) Remote (private -> nexus/jfrog) access within company or project

3) Local (.m2) ( in our system) (C:\Users\<>\.m2

Our project look into local repository for dependencies if not found go into central. Then save into local, next time it will pick from local.

Maven Goals : Used to perform project build process.

compile : convert .java files to .class files

test : execute junits

package : generate jar or war

clean : delete .class files

=============

Maven Setup

=============

@@ Ref Video : https://www.youtube.com/watch?v=hV1OWzYpzxo

1) Download and install jdk s/w

2) Setup JAVA\_HOME and Java Path in env variables

3) Download Maven software from apache website

4) Setup MAVEN\_HOME and maven path in env variables

5) Verify maven setup using cmd

mvn -version

==================================================

Create Maven Project in CLI

============================

=> Open CmD and execute below command to create maven project

mvn archetype:generate -DgroupId=in.ashokit -DartifactId=my-app -DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.5 -DinteractiveMode=false

=> We can see below folder structure in the maven project

My-App

- src/main/java (source code .java files)

- src/main/resources (config files)

- src/test/java (junit classes)

- src/test/resources (config files for testing)

- Maven Dependencies (jars)

- **pom.xml (project object model)**

=> Go inside maven project and execute maven goals

cd my-app

mvn compile

mvn test ( use to execute unit test of application)

mvn package

mvn clean package ( delete the target, compile the project,test the project, package the project)

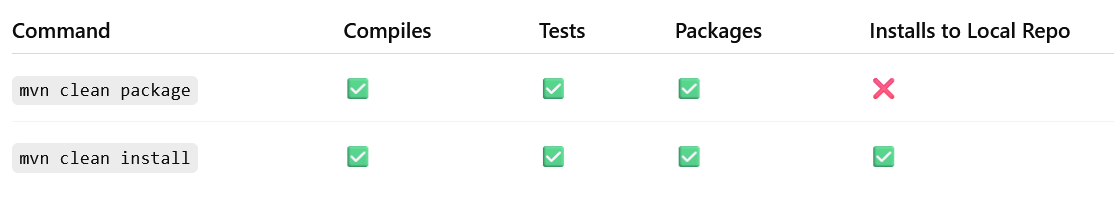
mvn clean complie ( two goals at a time)

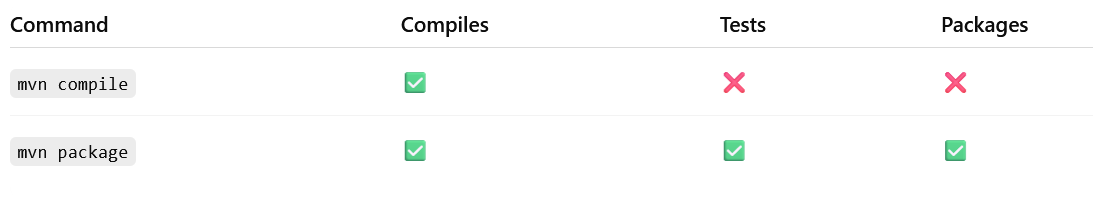
=> Byte code will be generated in project "target" directory.

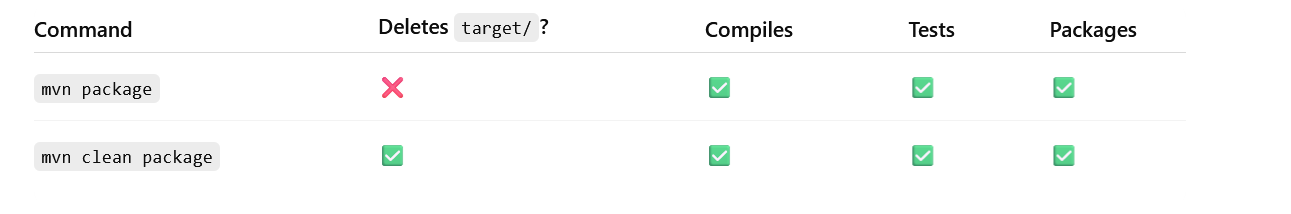
=> We can add maven dependencies in "pom.xml" file.

=> We can find maven dependencies in below website

URL : www.mvnrepository.com







Maven supports java but gradle supports other programming languages also.