
In java Domain, there are 3 categories of subjects to learn

a) Java Programming Language (Core java course)

b) Java Technologies (adv.java course)

c) Java Frameworks (Spring/spring boot & Micro Services course)

Frameworks

=====

Def1::

Framework is a special installable software that is built on top of one or more technologies having the ability to generate the common logics of the App dynamically to let the programmers to generate only application specific

Def2 :: Framework is a special software that provides abstraction on one more technologies to simplify the application development process

eg:: struts, jsf (java server faces), hibernate, spring,spring boot and etc.. (Java Based Frameworks) eg:: asp.net mvc, .net Core and etc.. (.net Based Frameworks) eg:: angular, ReactJs and etc.. (Java scrit Based Frameworks) eg:: Django, flask and etc.. (Python Based Frameworks)

eg:: WordPress, drupal and etc.. (PHP Based Frameworks)

=> Abstraction means exposing the required details to developer/enduser and hiding unwanted details from developer /end user

=> Framework internally uses technologies support to generate the common logics but does not make the programmers to worry about and lets programmers to concentrate only

on application specific logics ..This is nothing but frameworks providing abstraction on technologies.

呆

Developer

Develops

(spring/spring App)

Software App

uses

(spring/Spring boot framework)

Framework

uses

JDBC)

Technology1

Technology2 Technology3

33 other technology) Technologyn

uses

(JAVA) Language

=> JDBC is a java technology using which we can make Java Apps connecting to SQL DB s/w (oracle,mysql

and etc..) and to manipulate DB table records by performing CURD Operations

Plain JDBC App (Technology Based App Development)

=> Load JDBC driver class to activate JDBC driver s/w => Establish the connection with DB s/w

=> create JDBC Statement object

Common logics

=>JDBC driver s/w acts as bridge b/w java App and DB s/w. It converts java calls to DB calls and Vice-Versa

eg: JDBC driver s/w for oracle

eg: JDBC driver s/w for MySQL

eg: JDBC driver s/w for PostgreSQL

=> use JDBC Statement obj to send and execute SQL Queries in DB s/w => use JDBC Statement object to gather and process results given by DB s/w

=> Exception handling

=> Close connection with DB s/w

Common logics

=>While developing the technology based Apps,the Programmers need to take care of both Common logics and App specific logics.. this improves burden on the Programmers

Application specific

Logics

=> JDBC Connection acts as road b/w java app and DB s/w

=> JDBC Statement obj acts as vehicle or courier b/w Java App and DB s/w to send and execute SQL Queries (inputs) in DB s/w and to gather SQL Query results (outputs) back to java app from DB s/w

=>In technology based app development, there is a boilerplate code problem note:: The code that repeats in multiple parts of the Project either with no changes or with minor changes is called boilerplate code problem

=> spring/spring boot framework is given in the form of modules (25+)

eg:: JDBC, AOP (Aspect Oriented Programming), Web MVC, mail, scheduling,...

Spring JDBC App (Framework Based App Development)

Common for all the apps

of the same category

=> create JdbcTemplate class obj by supplying JDBC driver details (driver class name, url,db user,db pwd)

Taking care of Common logics

=>Send and execute SQL Queries in DB s/w

=> Gather and process DB s/w supplied SQL Query results

Application specific logics (will vary app to app)

note:: While developing F/w based Apps, the programmers need to write only App specific logics becoz the common logics of the App will be generated internally and dynamically by the Framework using technologies support. This indicates there is no Boiler plate code problem.

Advantages of working with Frameworks

=>provides abstraction on one or more technologies to simplify the app development process => Generates the common logics of the App dynamically to reduce the burden on the Programmers => Gives good Productivity (Doing more in less time with good accuracy)

=> The APIS (App Programming Interfaces) /Libraries of the Frameworks are designed after Real world use-cases and scenarios, So using them in real projects is going to be very useful

=> Frameworks have become industry's defacto standard in Apps/Projects development and etc..

=> if the project is very very small then prefer language based development eg: Deskop Game, calculator App

=> if the project is small then prefer technology based development

eg: Super market, small website

=> if the project is medium scale then prefer Framework (Spring) based development eg: E-commerce App, Banking Apps

=> if the Projects is large scale complex project then prefer Framework (Spring boot) based Development by using plan jdbc technology

internally

eg:: Phone Server App(UPI Payment), Papal (payment broker Apps), VISA App and etc..