Spring...? :

Spring is modularised framework developed by *Rod Johnson* in*2003*. It is a light weight &open source framework.

light weight: as it supports POJO model

open source:as there is no need to purchase any License

Spring is a complete &modular framework developed to design any real-time application.

complete: as it can be used to developed all layer of application

modular : as it can be used to developer particular layer of application

Advantages of Spring :

* Spring is non-invasive : as it does not forces us to extend any class or implement any interface
* Spring is light weight :as it supports POJO model
* Loose coupling : as we do not need to change java code
* Spring has its own container, so it doesn’t need server to run application
* Spring supports all (Web, Mobile based, Desktop based) applications
* Spring can integrate another framework/tool easily

Why Spring...? :

Simplicity : Spring is simple as it is non-invasive & light weight

Testability : Testing of the application is easy

Loose coupling : In Spring objects are loosely coupled as xml file comes into picture

Spring Modules :

Spring framework is devided into 6 modules as –

1)Spring Core Module 2)Spring Context - J2EE

3)Spring DAO Module - Spring JDBC 4)Spring ORM module

5)Spring AOP - Aspect Oriented Programming 6)Spring WEB-MVC Module

for database interaction

**M**

**V**

**C**

**A**

**O**

**P**

**JDBC**

logic/service

code separation

for all types of apploications

for web applications

**J**

**2**

**E**

**E**

**ORM**

**CORE**

Spring Core Module - Spring IOC :

IOC : Inversion of Control

Core module is of Spring. It is about tight & loose coupling between java objects.

Tight-coupling :Tight coupling is when two or more classes are highly dependent on each other. Changes in one class forces us to change other class i.e two classes changes together as they are depending on each other. When we make some changes in java code,

re-compilation of java file occurs ,which is not recommended.

Loose-coupling : Loose coupling is designed to reduce the interdependencies of two classes.

As two classes will not dependent on each other ,even if we make some changes in one class, we do not need to change other class.

In Spring, loose coupling is achieved through Dependancy Injection.

In Spring there are 2 containers :-

1. Core container : BeanFactory (used to develop Desktop based applications)
2. J2EE container : ApplicationContext (used to develop Enterprise/Web based applications)

Functions of containers –

* Create instance of POJO class
* Manage life-cycle of POJO class
* Dependancy injection of POJO class

Resources (i)

BeanFactory(i)

FileSystemResource (c)

ClassPathResource (c)

XMLBeanFactory(c)

ApplicationContext (i)

FileSystemXMLApplicationContext (c)

ClassPathXMLApplicationContext (c)

AnnotationConfigApplicationContext (c)

WebApplicationContext(i)

WebXMLApplicationContext(c) AnnotationConfigWebApplicationContext

***Bydefault Spring followsSingleton Design Pattern.***

BeanFactory container :

It is used for desktop based application. It is light weighted .

Resource rs = **new** ClassPathResource(spring configuration file path);

BeanFactory ap = **new** XmlBeanFactory(rs);

ApplicationContext container :

It extends BeanFactory container so, ApplicationContext container has all the properties of BeanFactory container.It adds some extra functionality than BeanFactory such as simple integration with Spring's AOP, message resource handling, event propagation, application layer specific context (e.g. WebApplicationContext) for web application. So it is better to use ApplicationContext than BeanFactory.

ApplicationContext ap = new ClassPathXmlApplicationContext(spring configuration file path);

Difference between beanFactory and Application Context :

|  |  |  |
| --- | --- | --- |
|  | BeanFactory | ApplicationContext |
| 1 | it is also called as core container | it is also called as J2EE container |
| 2 | Implemented class is XMLbeanFactory | Implemented class is  1.classpathXMLApplicationContext  2.fileSystemXMLAppliactionContext  3.AnnotationConfigAppliactionContext |
| 3 | Post resource should add manually  By using  Resource r=new classpathResource(“ bean.xml”);  BeanFactory bf= new XMLBeanFactory(r); | Post resource will add automatically |
| 4 | It does not support any annotation | It support annotation |
| 5 | It does not support internationalization | It supports internationalization |