



Spring Framework

spring

Content



Spring Core

Spring Data
Integration

Spring WEB

Spring Other Important Modules

Prerequisite of course



Core java

JDBC

Servlet &
JSP

Important Web and Database related terms

Why Learncodewith Durgesh



1. To the point content.
2. Clear explanation with practical approach
3. Ask your doubt in comment section or Instagram(@Durgesh_k_t)
4. Industry approach
5. Source code of every practical video is available to website.
6. Much more benefits from this channel

Support us



1. **Subscribe to my channel(Learncodewith Durgesh)**
2. **Follow on Instagram(@Durgesh_k_t)**
3. **Follow on facebook (@learncodewithdurgesh)**
4. **If you want to help me just donate some money follow the link**

<https://courses.onlyjavatech.com/about>



What ^{leaf}spring ng?

What is Spring?



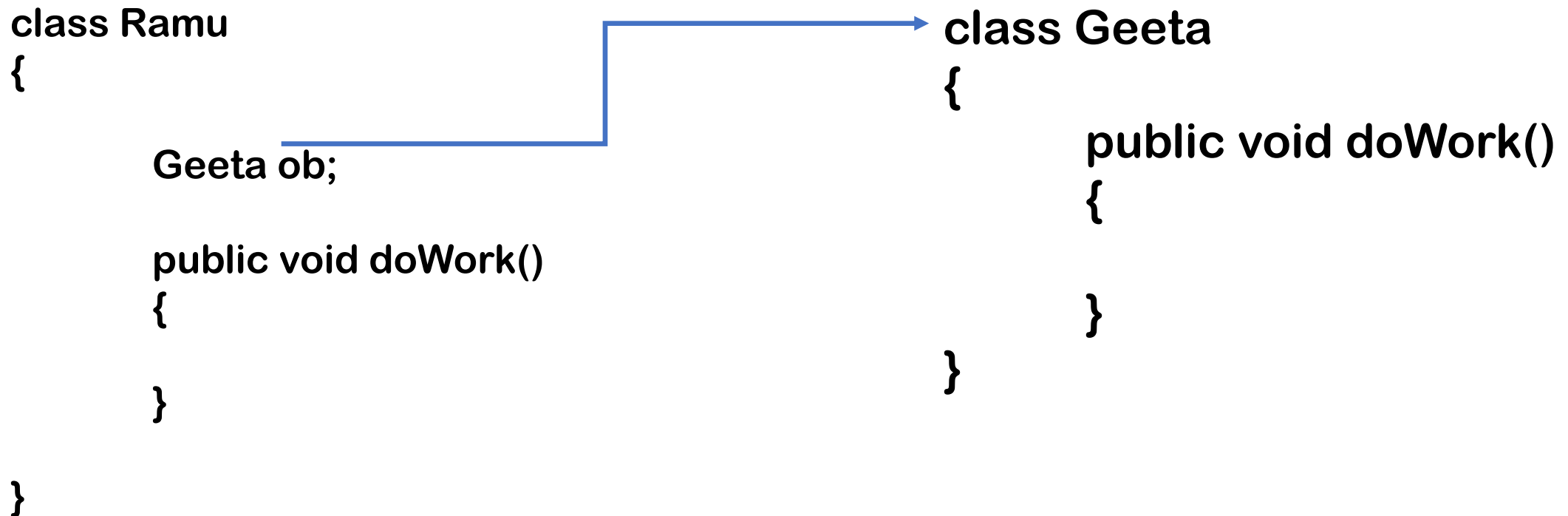
Spring is a Dependency Injection framework to make java application loosely coupled.

Spring framework makes the easy development of JavaEE application.

It was developed by Rod Johnson in 2003

Dependency Injection

It is design pattern

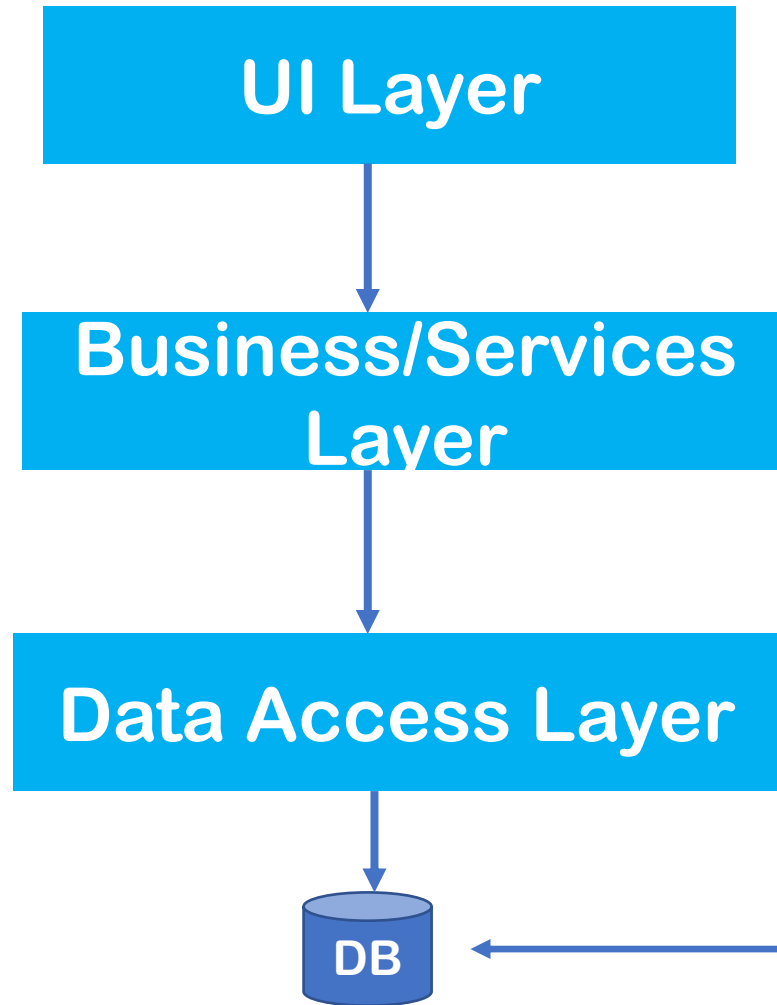


Inversion Of Control (IOC)

Spring and JEE



Strut / JSF
Spring MVC
Security
Transaction Management
Spring JDBC
Spring ORM



ProductController

ProductService

ProductDao

inject

inject

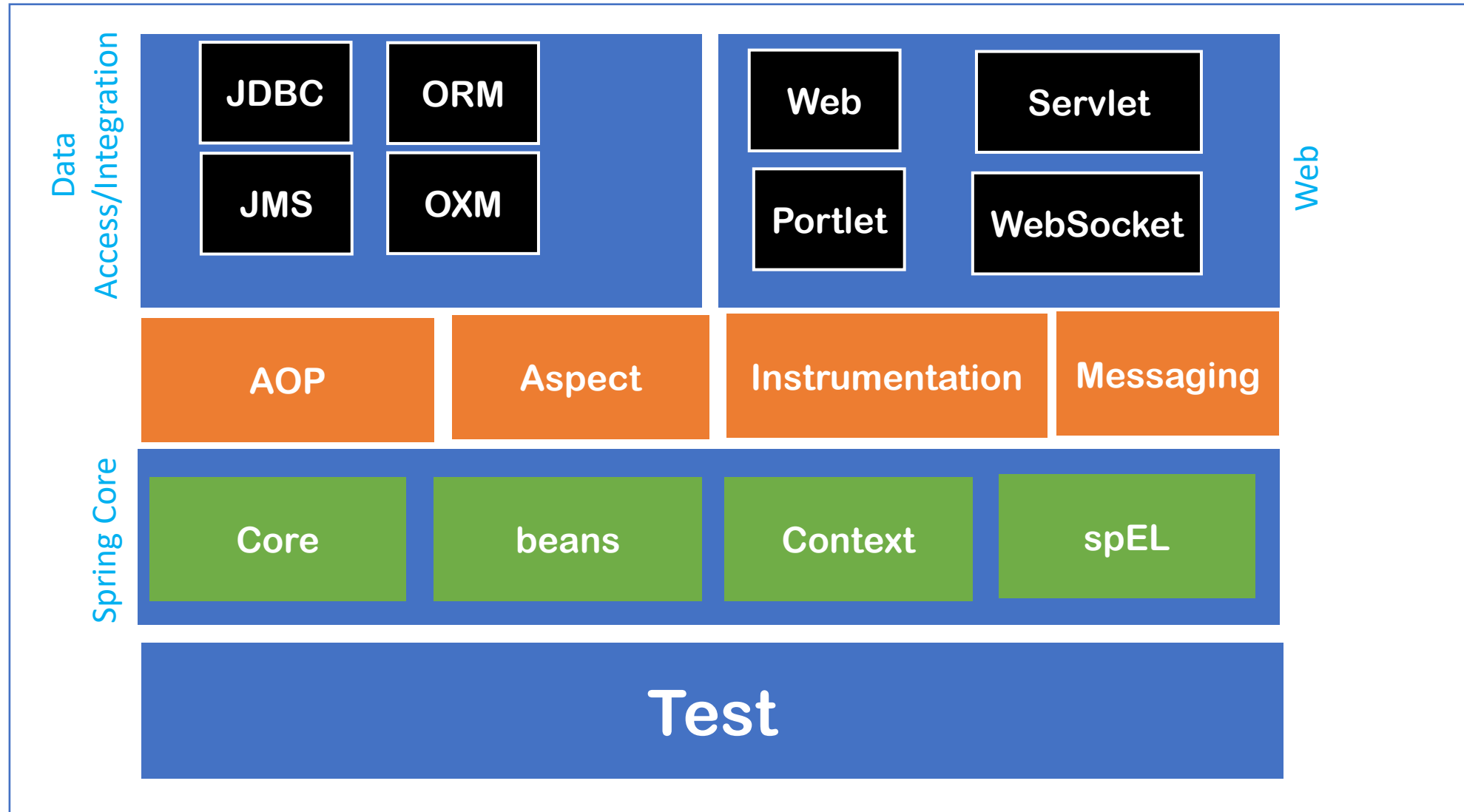


Spr^{ing} rules

Spring Modules



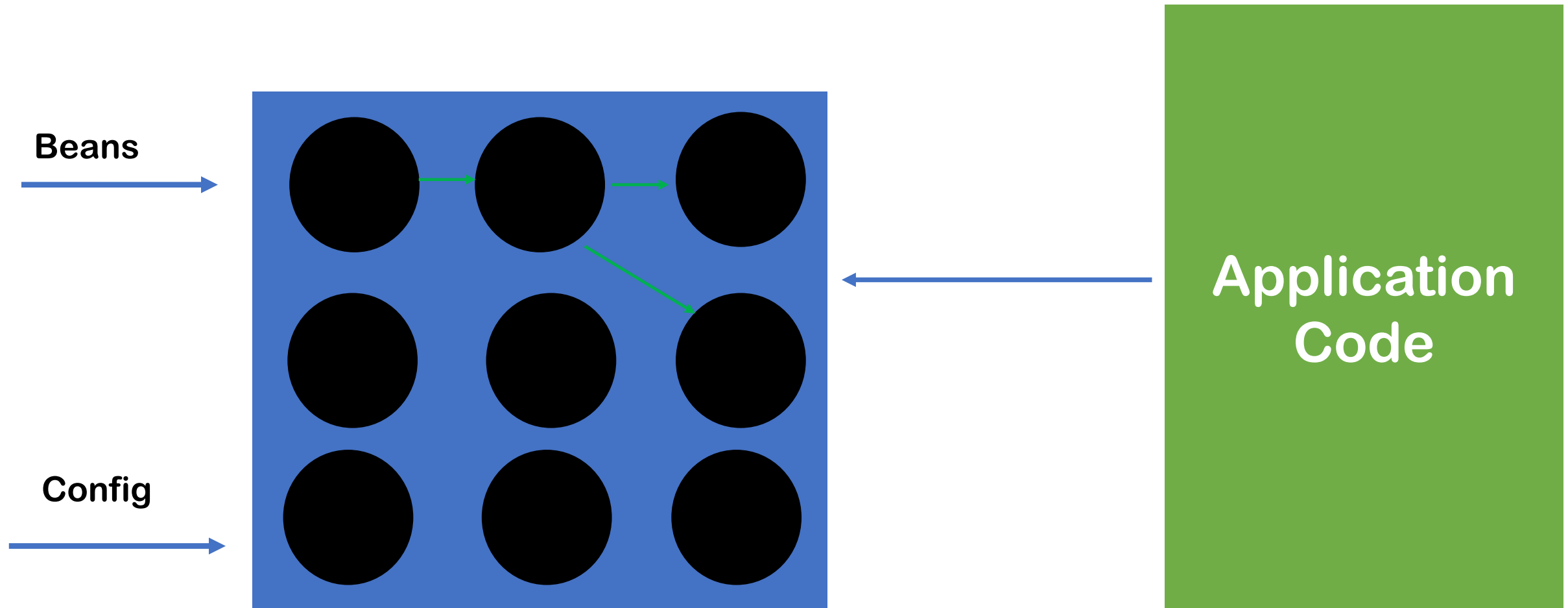
Spring Framework



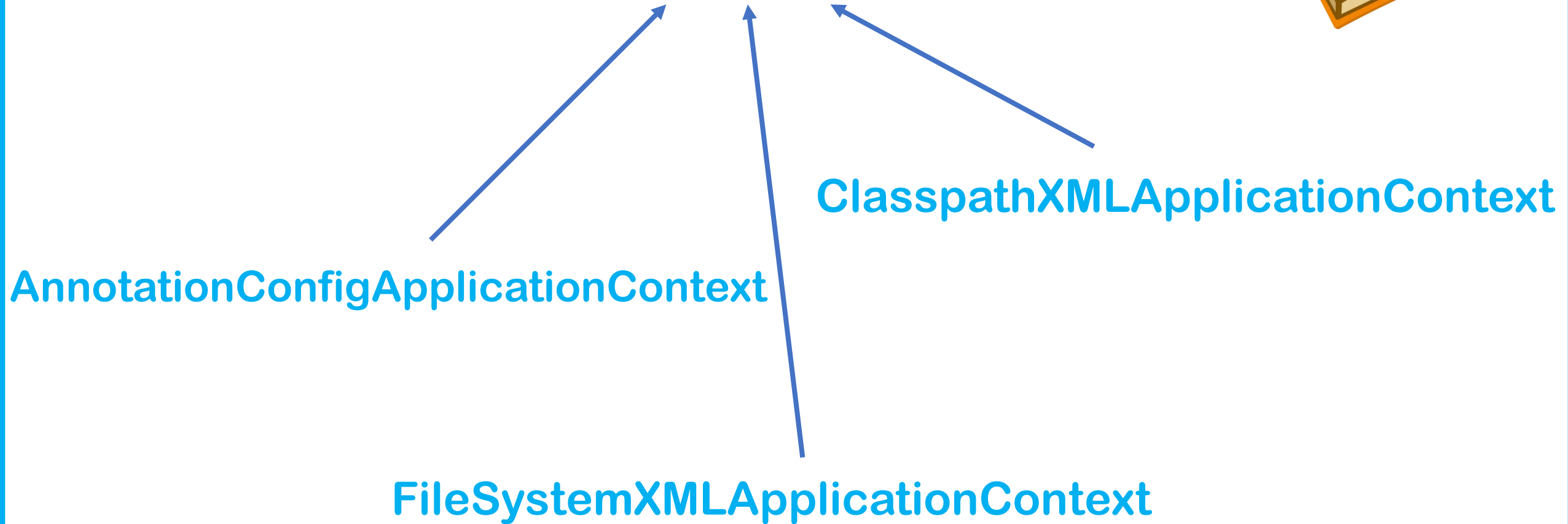


Spring  spring ntainer

Spring IoC Container



ApplicationContext





Deper spring njection

Dependency Injection



```
class Student  
{
```

```
    → int id;  
    → String name;  
    Address address;  
}
```

```
class Address  
{
```

```
    String street; ←  
    String city; ←  
    String state; ←  
    String country; ←
```

```
}
```

Dependency Injection Can be done in 2 ways



Using Setter Injection

Using Constructor Injection



Setter Injection



```
class Student
{
    id,name,address

    setId(id){ }

    setName(name){ }

    setAddress(address){ }

}
```



```
class Address
{
    street,city,state,country
    setStreet(street)
    setCity(city)
    setState(state)
    setCountry(country)

}
```

Constructor Injection



```
class Student
{
    String id,name,address;

    Student(id,name,address)
    {

    }

}
```



```
class Address
{
    String street,city,state,country;
    Address(street,city,state,country)
    {

    }

}
```

Configuration File



Where we declare beans
and its dependency



Data Types(Dependencies)



1) Primitive DataTypes



Byte , short , char , int , float , double , long , boolean

Data Types



2)Collection Type



List , Set , Map and Properties

Life Cycle Methods



Spring provide two important methods to every bean

Initialization
code
Loading
config,
Connecting db,
Webservice etc

```
public void init()
```

```
public void destroy()
```

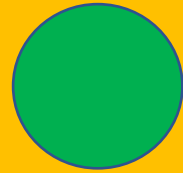
Clean up code

We can change the
name of these
method
But signature must
be
same

Life Cycle



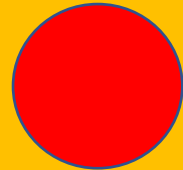
Spring bean



init()

Then we Read and use the bean

destroy()



Configuration
Xml File



Configure Technique



Xml

Spring Interface

Annotation



Using Interfaces



InitializingBean

DisposableBean



Using Annotations



@PostConstruct

@PreDestroy



Autowiring in Spring

- Feature of spring Framework in which spring container inject the dependencies automatically .
- Autowiring can't be used to inject primitive and string values. It works with reference only.



A



B



Manually

`<ref bean="" />`



Programmer

Automatically



Spring container

Autowiring

```
graph TD; A[Autowiring] --> B[XML]; A --> C[Annotations];
```

XML

Annotations

Autowiring Modes

no

byName

byType

constructor

autodetect

It is deprecated since Spring 3.

@Autowired

Autowiring Advantages

- **Automatic**
- **less code**

Autowiring Disadvantages

- **No control of programmer.**
- **It can't be used for primitive and string values.**

Stereotype Annotations

- XML

<bean />

@Component

Class Student

{

}



<context:component-scan base-package=" " />



Stereotype Annotation

@Component

Class Student

{

}

Student student=new Student()



Bean Scope

Singleton

prototype

request

session

globalsession



Configure bean scope

```
<bean class=" " name=" " scope=" " />
```

```
@Component  
@Scope( " ")  
Class Student  
{  
  
}
```



SpEL Spring Expression Language

Supports Parsing and executing expression with the help of @Value annotations

Expression

@Value

Classes, Variable, Methods, Constructors and Objects

and symbols

char, numerics, operators, keywords and special symbols which return a value



SpEL

```
@value("#{11+22}")
```

```
@value("#{ 8>6 ? 88 : 55"}")
```

static methods

object methods

variables



How to invoke static method and variable?

T(class).method(param)

T(class).variable



How to create objects?

new Object(value)

