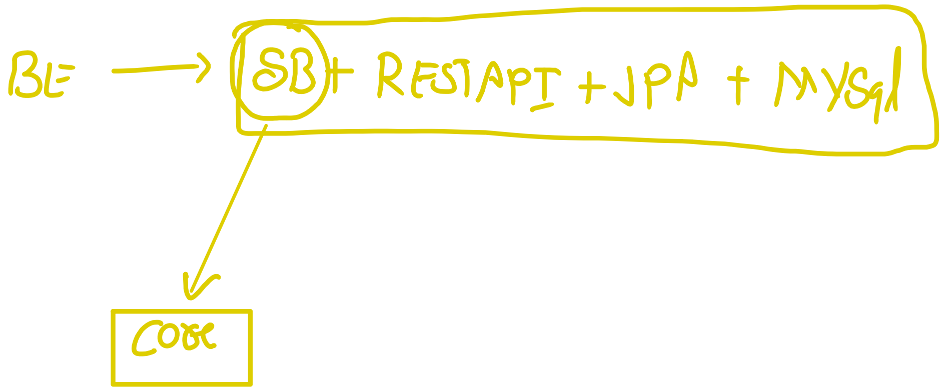
Spring Core

01 March 2025 09:03



What is programming lang ?

1. set of instructions
2. Each prog lang follow set syntax/rule

Using prog lang we develop ?

- Applications
- Tools
- Software
- Technologies
- Frameworks

What is technology ?

1. Technology is s/w which is developed by using prog lang
    a. Servlets, JDBC, JSP
2. Using technology we can develop many application like
    a. Project = BL + Common Logic

What is Framework ?

1. To overcome the problems in technologies framework came in market
2. Framework are semi developed software
3. Framework provides common logic required to development in application
    a. Fetch data
    b. Validate input
    c. Create connection pool

d. Crud operations
4. F/w provides re-usable component

For example, in java if we are working with jdbc

- Class.forName("………")
- getConn("……");
- createStatement()
- executeQuery()
- Process resultSet
- Close connection

For normal crud operations we have follow above steps again and again

Note : if we are writing same code for multiple times then it is called boiler plate code / reducntant code

To avoid duplicate code / common logic in project framework came into picture

Framework provides common logic required for the project development so that developer can focus on business logic

Framework will improve developer productivity
1. Java -> struct, hibernate, spring
2. .Net -> WCF
3. Python -> Django, Flask
4. Salesforce -> Lightening

TOOLS

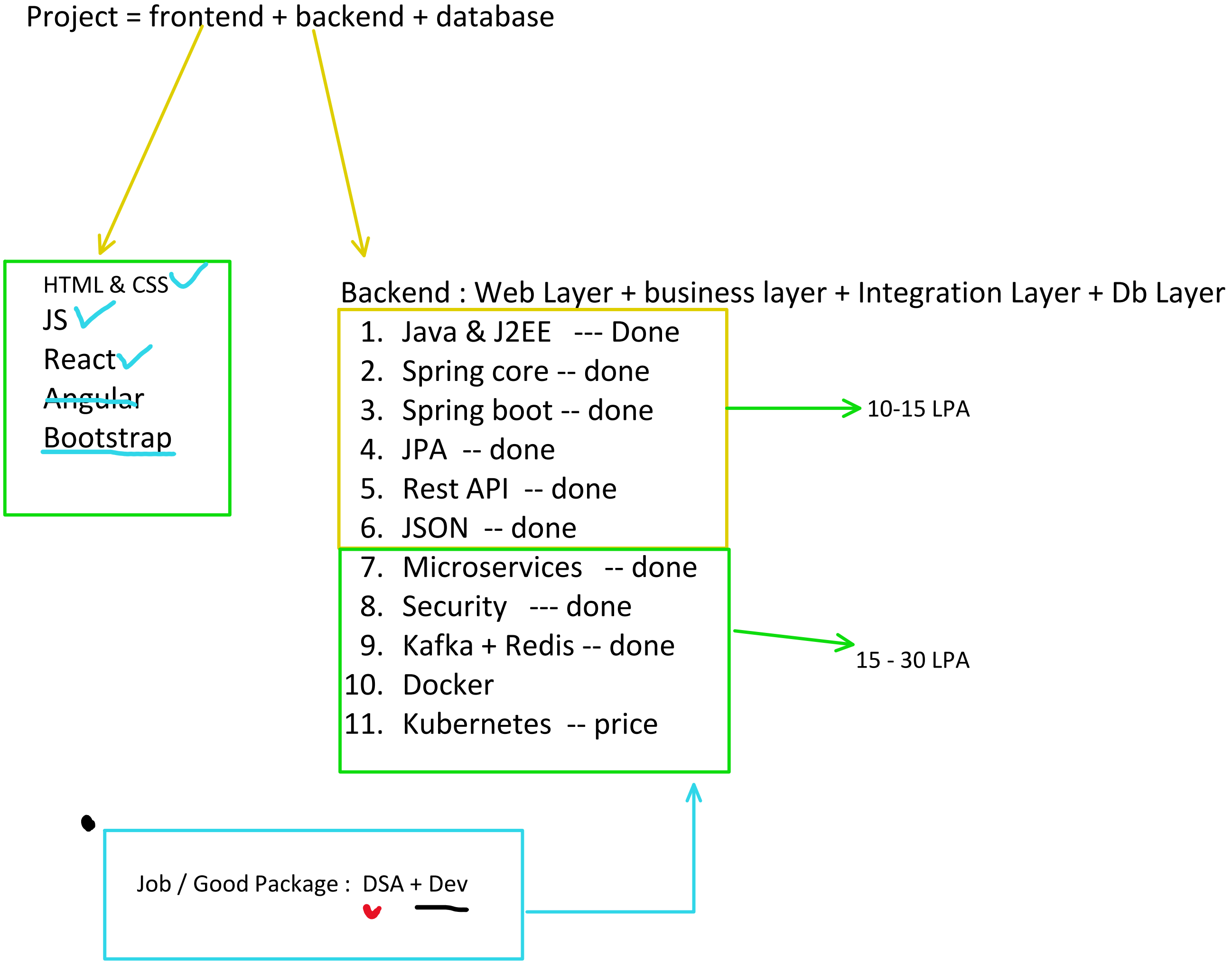
Tools are used to automate manual work
1. Maven
2. Jira
3. Jenkins
4. Jmeter
5. Postman
6. Sonarqube

Note : framework will be developed by using prog lang

Core java ===> prog lang
JBDC + Servlets===> technologies
Hibernate + Structs + Spring +Spring Boot ===> Framework

Application Architecture

Project = frontend + backend + database



HTML & CSS ✓
JS ✓
React ✓
~~Angular~~
Bootstrap

Backend : Web Layer + business layer + Integration Layer + Db Layer

1. Java & J2EE --- Done
2. Spring core -- done
3. Spring boot -- done
4. JPA -- done
5. Rest API -- done
6. JSON -- done
7. Microservices -- done
8. Security --- done
9. Kafka + Redis -- done
10. Docker
11. Kubernetes -- price

10-15 LPA

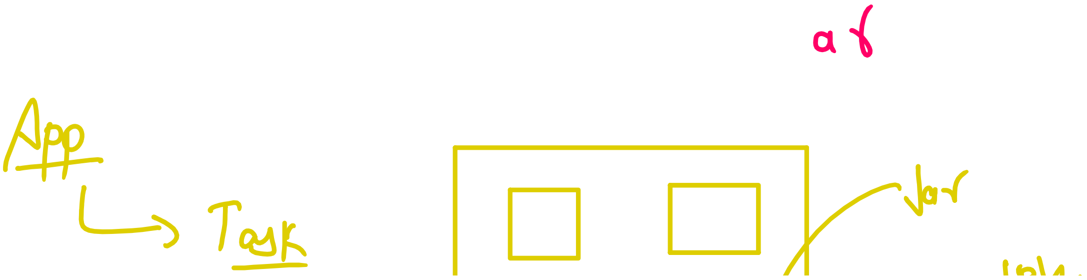
15 - 30 LPA

Job / Good Package : DSA + Dev

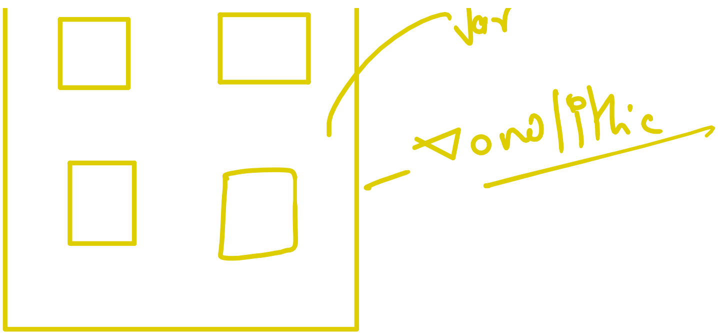
Type of applications

----

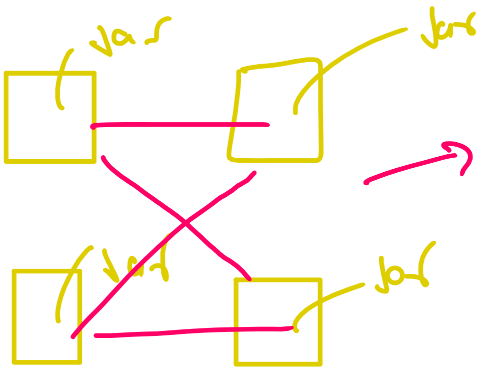
1. Monolithic app
2. Microservices

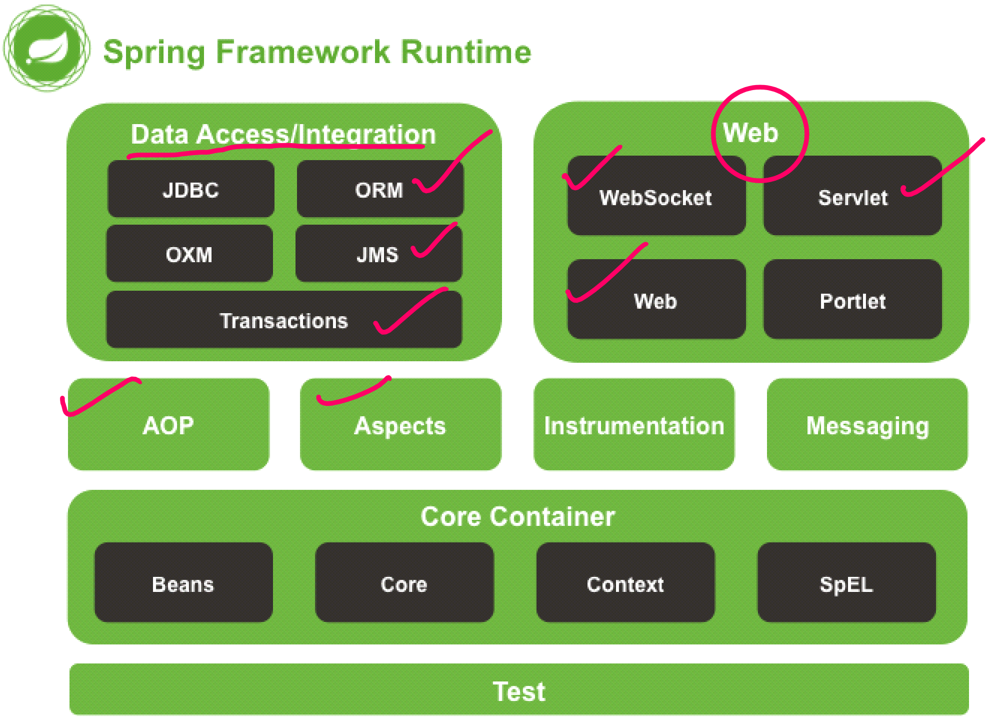


↳ Task



* Microservices




Spring Framework Runtime

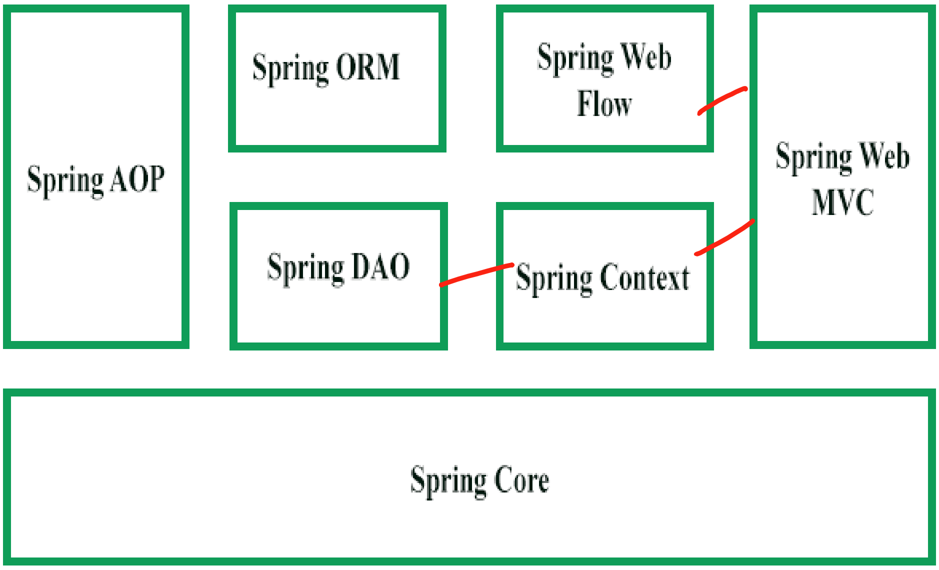
Data Access/Integration
JDBC
ORM
OXM
JMS
Transactions

Web
WebSocket
Servlet
Web
Portlet

AOP
Aspects
Instrumentation
Messaging

Core Container
Beans
Core
Context
SpEL

Test



Spring modules

1. Spring core
    a. It is base module in spring framework
    b. Spring core module provide fundamentals of spring framework
        i. IOC container
        ii. Dependency injection
        iii. Bean life cycle
        iv. Bean Scope
        v. Autowiring etc.....

1. Spring context : deal with config required for spring app
2. Spring jdbc : simplify database communication
3. Spring aop : Aspect oriented prog
    i. Aop is separate business logic & secondary logic
4. Spring mvc : used in web app
5. Spring security :
6. Spring batch : used in bulk processing
7. Spring data jpa : simplify database communication
8. Spring rest : used in making api
9. Spring cloud : used in config
10. Spring ai : used to integrate AI logic in spring boot

Spring Core

SC it is all about managing dependencies among the classes with loosely coupling

In project we will develop several classes all those classes are categorize into 3 type

1. POJO
2. Java Bean
3. Component

What is POJO ?
----

1. Any java class which can be compiled by using only JDK s/w is called pojo

Examples

```
Class Demo {

Int id;
String name;

}
```

POJO

Valid pojo

Example 2

```
Class Demo2 implements Runnable
{
       // run method
}
```

POJO

Valid pojo

Example 3

```
Class Demo3 implements Servlet
{
    ///get()
   //post()
}
```

~~POJO~~

Not a Valid pojo

What is Java Bean ?
----

1. Any java object which follow bean specification rule is called a java bean

Rules:

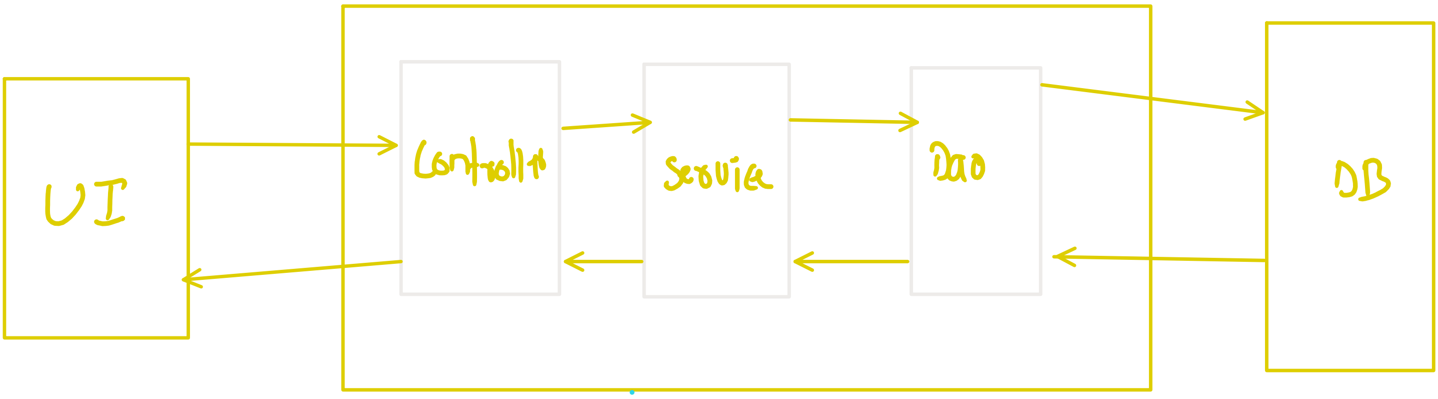
1. Class should implement serializable interface
2. Class should have private data members
3. Every private member should have public getter & setter methods
4. Class Should have Zero-Param constructor

Note : Bean classes are used to write business logic, store and retrieve data from db

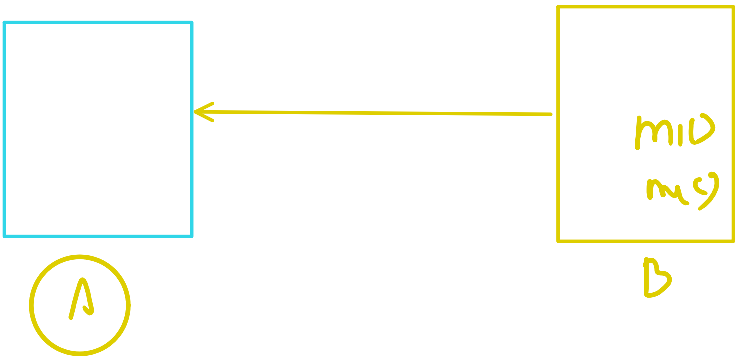
What is component

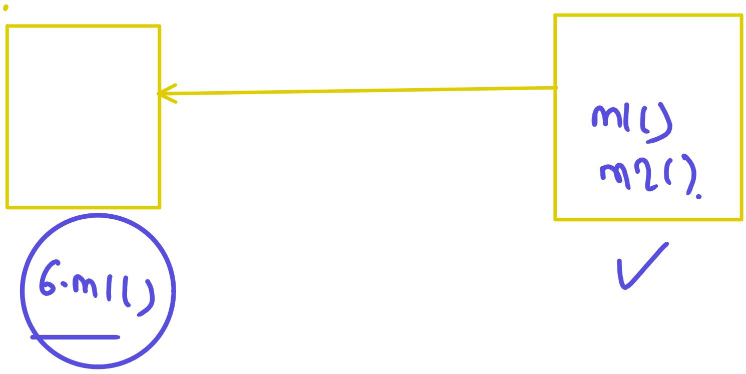
The java class which contains business logic is called a component

Ex: Controller, Services, Dao



1. Inheritance (IS-A)
2. Composition (Has-A)





✓ Is-A Relation

```
package org.example;

public class Sparrow extends Bird{

    public void flyHigh()
```

```
package org.example;
package org.example;

public class Bird {
    public int fly()
    {
        System.out.println("bird flying.....");
        return 1;
    }
}
```

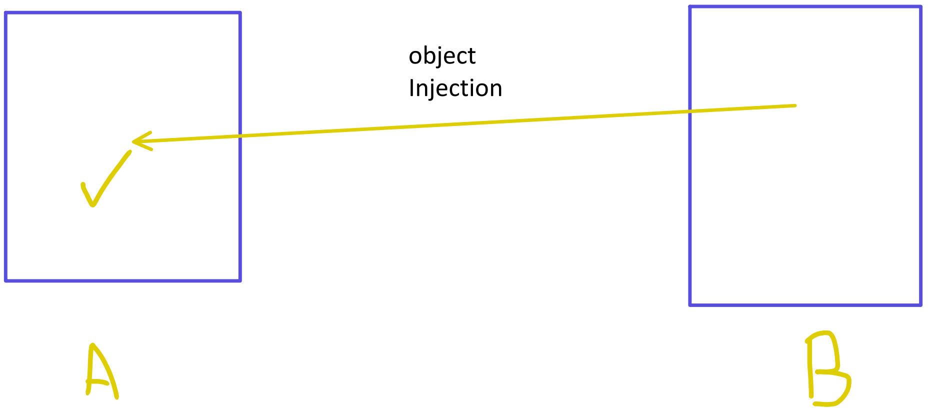
```
public class Sparrow extends Bird{

    public void flyHigh()
    {
        // first need to fly
        int fly=super.fly();
        if(fly>=1)
        {
            System.out.println("start fly high....");
        }
    }
}
```

✓

Has-A relation

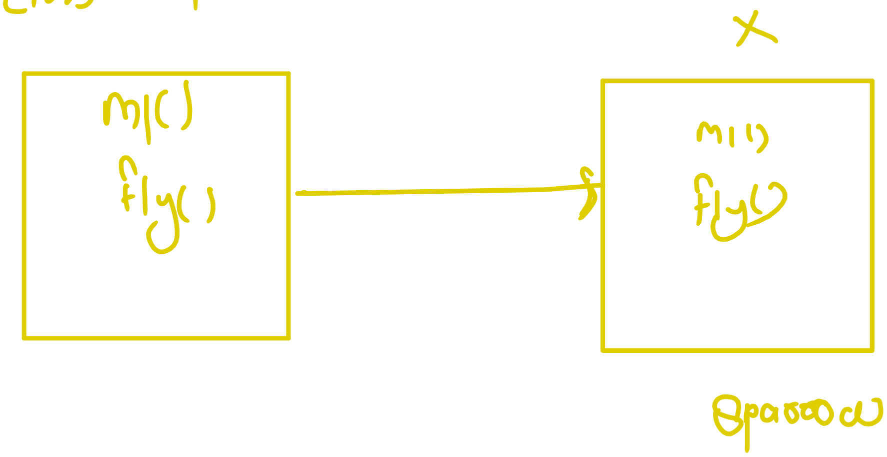
```
package org.example;

public class Bird2 {
    public int fly()
    {
        System.out.println("bird flying.....");
        return 1;
    }
}
```

```
package org.example;

public class Sparrow2 extends Bird{

    public void flyHigh()
    {
        // first need to fly
        int fly=super.fly();
        if(fly>=1)
        {
            System.out.println("start fly high....");
        }
    }
}
```

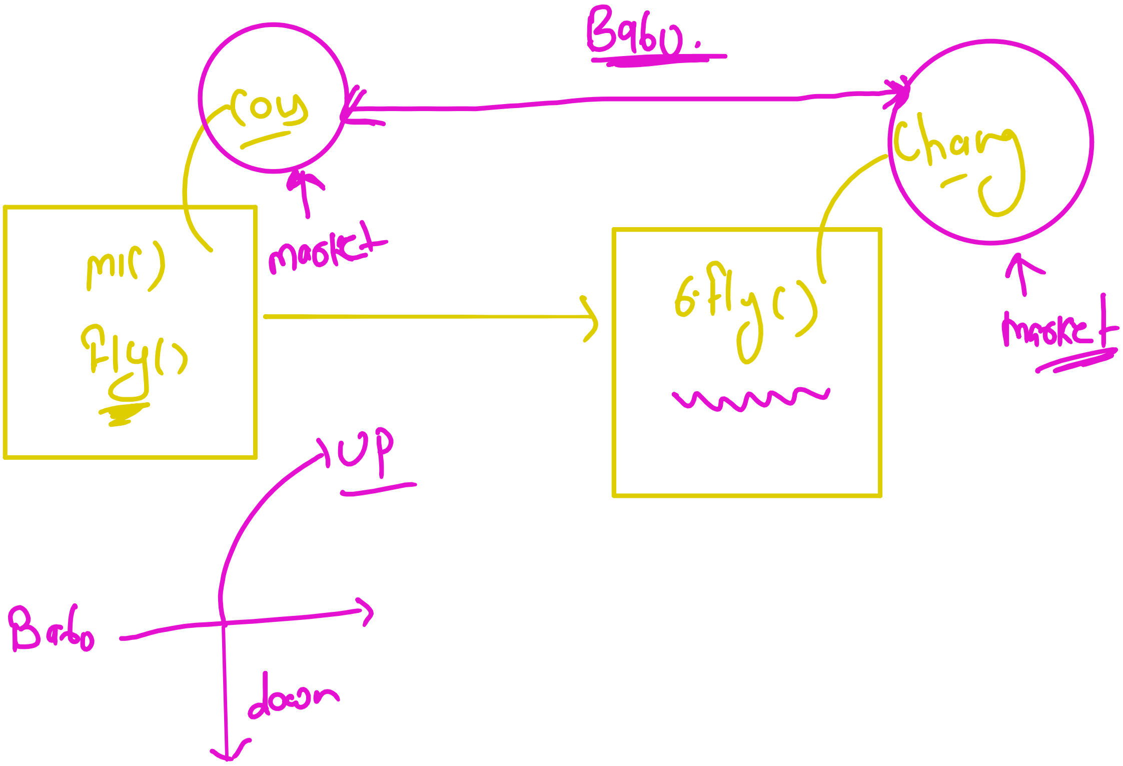


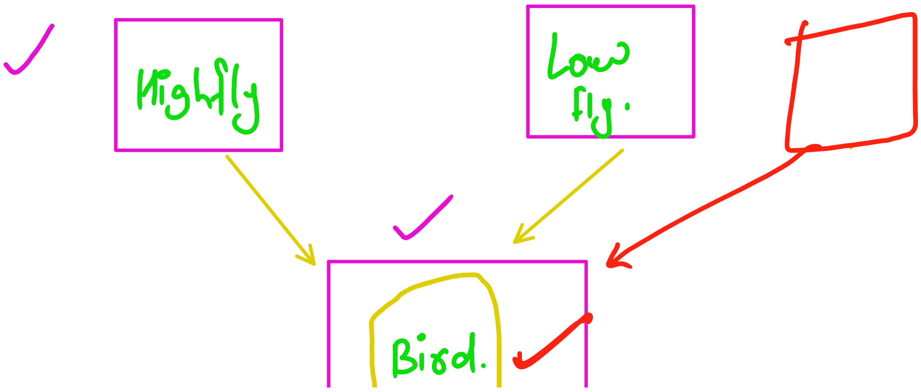
inter

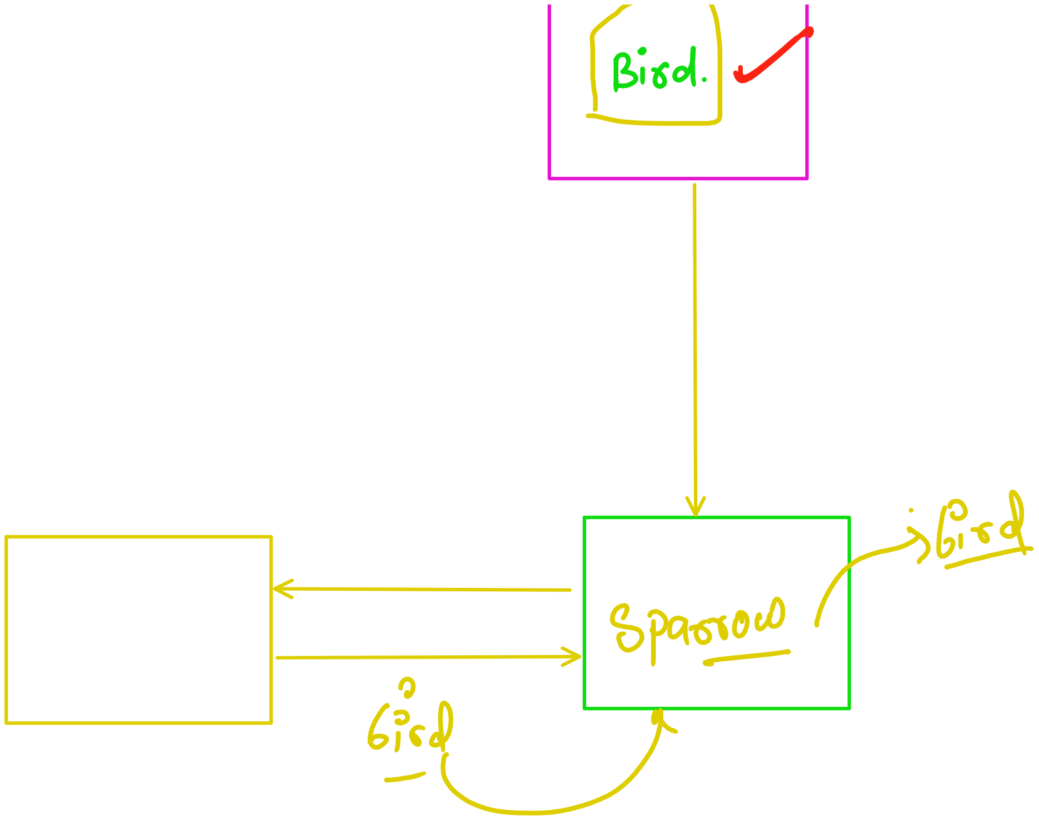
class Bird



has-A







What is dependency injection ?

1. The process of injecting one class object into another class is DI
2. We can perform DI in 3 ways
   a. Setter Injection
   b. Constructor Injection -- done
   c. Field Injection

Constructor Injection

Injecting dependent object into target object using target class constructor

Setter injection

Injecting dependent object into target object using target class setter method

Filed Injection

Injecting dependent object into target object using target class field/member/variable

Note : Can we perform both SI and CI for single variable ?

Yes, but setter injection override construction injection

IOC Container

----

1. IOC start for inversion of control
2. IOC is responsible for Dependency Injection in spring Application
3. Dependency Injection means creating bean and injecting bean into target bean classes

We have two IOC container

1. BeanFactory (outdated)
2. ApplicationContext

Note: IOC container will manage life cycle of spring beans
Note: we need to provide "java classes + bean configuration" as input to IOC then IOC will perform DI and provides spring beans which are ready to use.

What is Spring Bean ?
----

1. Any java class whose lifecycle(creation to destruction) is managed by IOC is called a spring bean
2. We can represent java class as spring bean in 2ways
   a. XML--bean configuration
      i. <bean id="id1" class="full package.classname"/>
   b. Annotation based
      i. @Component, @Service, @Repository.....

Note:

1. In spring we can use both xml & annotation approaches

Note:

1. In spring boot will only play with annotation (no xml)

How to make spring application
