# SPRING FRAMEWORK

## Spring  Framework Nedir?

Java ile geliştirme yapmayı kolaylaştıran Core Container, AOP, Data Access, Web gibi modüllerden oluşan bir framework-kütüphanedir.

Spring platformunda yer alan Spring Boot, Spring Data, Spring MVC, Spring Batch, Spring Security gibi projelerin temelinde Spring framework yer alır.

Spring framework temelinde ise Spring Core modülünde yer alan IoC Container vardır.

## IoC Container Nedir?

Sınıflardan nesne oluşturmak, oluşan nesnelerin ihtiyaç duyduğu nesneleri bağlamak ve yönetmek için kullanılan prensip ve desenlerinin bir araya geldiği araçlardır.

Ioc Container araçları genellikle aşağıdaki prensip ve desenleri kullanarak işlem yapar.

### IoC Container Framework

IoC – Inversion of Control – Tasarım Prensibi

Factory Pattern

Dependency Injection

Constructor Injection

Setter Injection

Interface Injection

Strategy Pattern

Service Locator Pattern

Template Method Pattern

DIP --- Dependency Inversion Principle – Tasarım Prensibi

Spring framewok **I**nversion **o**f **C**ontrol prensibini uygulamak için genellikle **D**ependency **I**njection tasarım desenini kullanır.

## IoC Inversion of Control nedir?

Bir sınıf içerisinde(A) kullanılan sınıfların(B, C vb.) başka bir yöntemle(Constructor, Factory vb.) oluşturulması, dahil edilmesi, kontrol edilmesine Inversion of Control denir.

OOP-NYP ile geliştirilen uygulamalarda sınıflar bir işlemi yapmak için diğer sınıfları kullanır.

Bir sınıfın(A) bir başka sınıfı(B) kullanabilmesi için **new** anahtar kelimesi ile oluşturulması gerekir.

```
public class A {

private B b;

   public A(){

      b = new B();

   }

   public void calistir(){

         b.yazdir();

   }

}

public class B{

         public void yazdir(){

         System.out.println(“Alperen Mutlu”);

   }

}
```

Sınıf(A) içerisinde **new** anahtar kelimesi ile başka bir sınıfı(B) kullanması, sınıfın(A) diğer sınıfa(B) bağımlı olmasına neden olur.

Bu bağımlılığın sınıftan(A) alınarak başka bir sınıfa verilmesi Kontrolün Tersine Çevrilmesi-Inversion of Control olarak adlandırılır.

Bu işlemin yapılması için Factory, Dependency Injection, Strategy, Service Locator, Template Method gibi çeşitli tasarım desenleri kullanılır.

# HIBERNATE

## Hibernate Nedir?

Hibernate Java geliştiriciler için geliştirilmiş bir ORM kütüphanesidir. Nesne yönelimli modellere göre veritabanı ile olan ilişkiyi sağlayarak, veritabanı üzerinde yapılan işlemleri kolaylaştırmakla birlikte kurulan yapıyı da sağlamlaştırmaktadır.

Hibernate bir nesne//ilişkisel eşleme (Object/Relational Mapping) aracıdır.

Hibernate yalnızca Java sınıflarından veritabanı tablolarına veya Java veri tiplerinde SQL veri tiplerine dönüşümü yapmaz. Hibernate veri sorgulama ve veri çekme işlemlerini de kullanıcı için sağlar.

Bu özellikleriyle Hibernate geliştirme kolaylığı ve zamandan kazanç sağlar.

**Hibernate** kullanmadan **JDBC** ile veri tabanına erişmek mümkündür. Ancak veri tabanındaki tablo sayısı arttığında buna bağlı olarak tablolar arası ilişkiler de artacaktır. Uygulama büyüdükçe bu ilişkiler çok karmaşık bir hal alabilir.

**Veritabanı** işlemleri için connection açma kapama, ilişkili tablolar için çok karmaşık **SQL** sorguları yazma, aynı fonksiyon içinde birden fazla connection açmama gibi dikkat etmemiz gereken işler artacaktır.

Bu işlemleri yaparken yapacağımız en ufak hata uygulamanın tümünü etkileyecektir. Uygulamamızın mimarisi ne kadar düzgün olursa yapısı da bir o kadar karmaşık olacaktır.

Hibernate, hemen hemen yaygın tüm **veritabanı** sistemleri ile uyumludur. Bu özelliği ile çok fazla firma tarafından tercih edilmektedir. Aynı zamanda **veritabanı** bağımlılığını da ortadan kaldırdığı için tercih edilmektedir.

**ORM** yapısının kullanılmasının avantajları **Hibernate** içinde geçerlidir. **ORM'nin** sağladığı avantajları aşağıda listesi mevcuttur..

– **Veritabanınız** ile modelleriniz arasındaki bağlantıyı sağlamak için geliştirilmiş en iyi sistemdir.
– Veri ekleme, silme, güncelleme, okuma gibi işlemleri kolaylaştırmaktadır.
– Veri aktarma sürecini yönetmenizi sağlayarak, alabileceğiniz veya çıkabilecek hatalara karşı yönetimi de sağlamaktadır.
– **SQL** yazmaktan kurtarmaktadır.

# JSF – Java Server Faces

JSF veya **J**ava **S**erver **F**aces sunucu taraflı olaya dayalı, bileşen tabanlı web uygulamaları geliştirmek için kullanılan Java EE kütüphanesidir.

Kütüphanede yer alan bileşenlere ait özellikler kullanılarak sunucuda yer alan veri ve olay arasında bağlantı kurularak işlem yapılır.

Örnek olarak kullanıcı veri giriş yaptıktan sonra sayfa içerisinde yer alan butona basarak veri girişinin doğrulanması, verinin kayıt edilmesi işlemi verilebilir.

## JSF Nasıl Çalışır?

JSF bir çok web uygulamasında kullanılan ve MVC yani **M**odel **V**iew **C**ontroller mimarisini kullanır.

### Controller

Gelen istekler Controller katmanında yer alan ve **Servlet** arayüzünü uygulayan-implement eden **javax.faces.webapp.FacesServlet** sınıfı tarafından yönetilir.

```
<servlet>
 <servlet-name>Faces Servlet</servlet-name>
 <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
 <load-on-startup>1</load-on-startup>
</servlet>
```

JSF sayfaları genellikle **.xhtml** ve **.jsf** uzantısı ile hazırlanır.

JSF sayfalarının işlenmesi için **web.xml** ayarı ile **FacesServlet** tarafından yönetileceği belirtilir.

```
<servlet-mapping>
 <servlet-name>Faces Servlet</servlet-name>
 <url-pattern>*.jsf</url-pattern>
</servlet-mapping>

<servlet-mapping>
 <servlet-name>Faces Servlet</servlet-name>
 <url-pattern>*.xhtml</url-pattern>
</servlet-mapping>
```

Gelen istekler eşleştikten sonra JSF sayfalarının yer aldığı view katmanı derlenerek istemciye gönderilir.

Derleme işlemi JSF yaşam döngüsü olarak adlandırılan adımlar kullanılarak yapılır.

**View**

View katmanında web sayfalarını oluşturmak için çeşitli JSF bileşenleri-etiketleri kullanılır.

JSF etiketlerini HTML etiketlerinden ayıran en önemli özellik Model katmanında yer alan ve Managed Bean olarak adlandırılan Java komutları ile bağlantı kurmayı sağlamasıdır.

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:h="http://xmlns.jcp.org/jsf/html">
  <h:head>
    <title>Merhaba JSF</title>
  </h:head>
  <h:body>
    Merhaba JSF
  </h:body>
</html>
```

Burada dikkat edilmesi gereken en önemli kısım aşağıdaki gibi etiket tanımlarının JSF sayfasına dahil edilmesidir.

```
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:h="http://xmlns.jcp.org/jsf/html">
```

Model katmanında yer alan alanlara erişim için **JSF Expression Language** kullanılır.

```
<h:form>
  <p>First name: <h:inputText id="firstName" value="#{student.firstName}" /></p>
  <p>Last name: <h:inputText id="lastName" value="#{student.lastName}" /></p>
  <h:commandButton value="Submit" action="#{student.Save}" />
</h:form>
```

Model katmanındaki değerler **FacesServlet** tarafından JSF yaşam döngüsü sırasında yerleştirilir.

**Model**

Verilerin bulunduğu ve veriler üzerinde işlem yapıldığı katmandır.

Bu katmandaki sınıflar bazı kuralları olan sıradan Java sınıflarıdır.

1. Parametresiz kurucusu olmalıdır.
2. Serializable arayüzünü uygulamalıdır.
3. Her bir sınıf özelliğine ait get ve set metodu olmalıdır.
4. Viw katmanı ile bağlantı için @Named Annotation veya XML ayarları kullanılmalıdır.

## JSF Yaşam Döngüsü

JSF, Servlet ve JSP gibi yaşam döngüsü olarak adlandırılan aşağıdaki çalışma adımlarını takip eder.

- Restore View Phase
  - View katmandaki bileşenler oluşturulur.
  - Olaylar ve doğrulama işlemleri bileşenlere bağlanır.
  - İşlemler FacesContext'e kaydedilerek sonraki adıma geçilir.
- Apply Request Values Phase
  - İstemciden gelen değerler View katmanıdaki bileşenlere aktarılır.
- Process Validations Phase
  - İstemciden gelen değerler belirlenen kurallara göre doğrulanır.
- Update Model Values Phase
  - Doğrulama adımında hata ile karşılanırsa Model katmanı güncellenir.
- Invoke Application Phase
  - Form gönderimi, JSF sayfaları arası geçiş bu adımda çalıştırılır.
- Render Response Phase
  - Oluşan sonuç istemciye gönderilir.