

Week 1

1. Kaynaklara bakıldığından 30'u aşkın Java Framework'ü gözlemlenmektedir. Bunlardan en yaygın olan 3 tanesini ve çözdükleri problemlerden bahsedeceğim.

Spring: Java uygulamaları geliştirmemiz için altyapı sağlayan açık kaynak bir framework'tür. Geleneksel Java nesnelerini kullanarak yüksek performanslı uygulamalar oluşturmamızı sağlar. Dependency Injection, MVC web uygulamaları, RESTful hizmetlerine çözüm getirmektedir.

```
public class SimpleMovieLister {
    // SimpleMovieLister sınıfı MovieFinder'a bağımlıdır
    private final MovieFinder movieFinder;
```

```
// Spring container'ın MovieFinder objesini inject edebilmek için oluşturulan bir constructor
public SimpleMovieLister(MovieFinder movieFinder) {
    this.movieFinder = movieFinder;
```

```
} ...
}
```

Hibernate: İlişkisel veritabanı yönetimi için bir ORM framework'üdür. ORM nesne ile ilişkisel eşlemedir. ORM teknolojisi veritabanındaki tabloları kolonları, yazılım içindeki nesneler ve nesnelerin özellikleri ile bağlar. Hibernate ve ORM frameworkleri genel olarak veritabanlarındaki işlemleri yapmamızı kolaylaştırır.

```
@Entity
@Table(name = "products")
public class Product {
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String name;
    private double price; ...
}
```

Struts: Struts, Java web uygulamaları geliştirmek için bir framework'tür. Model-View-Controller (MVC) mimarisini kullanır ve web uygulamalarını parçalara ayırarak geliştirilmesini kolaylaştırır. Web uygulamalarını yapılandırmada kolaylık sağlar.

```
public class LoginAction extends ActionSupport {
    private String username;
    private String password;
    public String execute() {
        return SUCCESS;
    }
    ...}
```

SOA (Service-Oriented Architecture)

SOA, (Hizmet odaklı mimari) iş uygulamalarının modüler bir yapıya sahip olması için oluşturulan bir mimari yaklaşımdır. Hizmet odaklı mimari, işlevleri bağımsız hizmetler olarak tanımlar ve bu hizmetlerin birbirleriyle etkileşimini sağlar. Bu hizmetler, farklı sistemlerde yeniden kullanılabilir.

Örneğin, bir banka uygulaması, müşteri yönetimi, hesap yönetimi ve işlem yönetimi gibi farklı hizmetlere sahip olabilir.

Web Services

Web servisleri, farklı sistemler arasında bir ağ üzerinden iletişim kurmak için kullanılan teknolojilerdir. Genellikle XML veya JSON formatında veri alışverişi yaparlar ve SOAP (Simple Object Access Protocol) veya REST (Representational State Transfer) gibi iletişim protokolleri kullanarak erişilirler. Online alışveriş sistemleri, elektronik bankacılık, sosyal medya platformlarının hepsi web servislerine örnektir.

Restful Service

REST, web hizmetlerinde kullanılan bir mimari tarzıdır. Bu tarz, veri ve işlevleri web bağlantıları aracılığıyla erişilen kaynaklar olarak düşünür ve bu kaynaklara Uniform Resource Identifiers (URI'lar) kullanarak erişir. Bu yaklaşım, web hizmetlerinin performansını, ölçeklenebilirliğini ve değiştirilebilirliğini artırır. RESTful servisler, HTTP protokolünü kullanarak kaynaklara erişim sağlarlar. HTTP yöntemlerini kullanarak kaynaklar üzerinde işlemler yapılır.

HTTP Method

HTTP metotları client ve serverlar arasındaki iletişimi sağlar. Client server'a istek yollar (bu istek bir veriye erişmek için olabilir ya da bir veriyi değiştirmek/silmek için olabilir) ve server'dan cevap gelir (bu cevap verinin kendisi ya da başarı mesajı olabilir). En yaygın HTTP Metotları aşağıdaki gibidir.

GET: Belirtilen veriye ulaşmak için kullanılır.

POST: Yeni bir veri oluşturmak için kullanılır.

PUT: Belirli bir veriyi güncellemek için kullanılır.

DELETE: Belirli bir veriyi silmek için kullanılır.

Kaynaklar:

[10 Best Java Frameworks You Should Know | Simplilearn](#)

[Dependency Injection :: Spring Framework](#)

[SOA Nedir? - Hizmet Odaklı Mimariye Ayrıntılı Bakış - AWS \(amazon.com\)](#)

[What is a web service? - IBM Documentation](#)

[What Are RESTful Web Services? - The Java EE 6 Tutorial \(oracle.com\)](#)

[HTTP Methods GET vs POST \(w3schools.com\)](#)