

2. Creational Design Pattern'lar incelenmelidir. Örneklerle anlatınız.

Creational Design Pattern'lar, nesnelerin oluşturulma sürecine odaklanır ve bu süreci düzenler. Bu pattern'lar, nesneleri oluştururken kod tekrarlarını azaltarak, kodun daha anlaşılır ve daha kolay yönetilebilir hale gelmesini sağlar. Creational Design Pattern'ların bazı örnekleri şunlardır:

Factory Pattern: Bu pattern, bir sınıfın bir nesne oluşturma işlemini bir alt sınıfa devretmesini sağlar. Bu sayede, sınıfın içeriği değiştirilebilir hale gelir ve bu değişikliklerin etkilerini sınıfın kullanıldığı diğer yerlerde göstermesini engeller.

Singleton Pattern: Bu pattern, bir sınıfın sadece bir tane nesnesinin oluşturulabileceğini sağlar. Bu sayede, bir sınıfın nesnesi, programın her yerinden erişilebilir ve tek bir kaynaktan yönetilebilir hale gelir.

Prototype Pattern: Bu pattern, bir nesnenin bir kopyasını oluşturmak için kullanılır. Bu sayede, bir nesne üzerinde yapılan değişiklikler, kopyaladığımız nesneye de yansıtılır ve bu değişikliklerin etkilerini sınırlı tutabiliriz.

4. Java dünyasındaki framework'ler ve çözdükleri problemler nedir? Kod Örneklendirini de içermelidir.

Java dünyasında kullanılan birçok framework bulunmaktadır. Bu framework'ler, geliştiricilere çeşitli problemleri çözme imkanı sunarak, kodun yazımını kolaylaştırır ve tekrar kullanılabilirliğini artırır. Örnek olarak, Spring framework'ü, nesnelerin birbirleriyle etkileşimini kolaylaştıran bir yapı sunarak, nesne yönelimli tasarım kalıplarını uygulamanızı sağlar. Hibernate framework'ü ise veritabanı işlemlerini kolaylaştırarak, veritabanı tabanlı uygulamalar geliştirmenizi sağlar. Struts framework'ü ise web uygulamaları geliştirirken, kullanıcı arayüzlerini düzenlemenizi ve işlemleri yönetmenizi kolaylaştırır.

Javada nesne oluşturma

```
Kisi kisi = new Kisi(); kisi.setAd("Cem"); kisi.setSoyad("Yılmaz");
```

Spring Framework ile nesne oluşturma

```
ApplicationContext context = new AnnotationConfigApplicationContext(AppConfig.class);
```

```
Kisi kisi = context.getBean(Kisi.class);
```

```
kisi.setAd("Cem");
```

```
kisi.setSoyad("Yılmaz");
```

Spring Framework kullanıldıktan sonra yazılan kodun daha kapsamlı ve düzenli bir yapıya sahip olması beklenir.

5.Spring framework kullandığı design patternlar neler?

Spring Framework, birçok farklı design patterni kullanır. Örneğin, Spring Framework kullanıcı arayüzleri için Model-View-Controller (MVC) patternini, veritabanı erişim için Data Access Object (DAO) patternini ve nesne oluşturma ve yönetme için Inversion of Control (IoC) ve Dependency Injection (DI) patternlerini kullanır. Bu patternler, Spring Framework'ün esnek, genişletilebilir ve kolayca test edilebilir bir yapısını sağlar.

6.SOA - Web Service - Restful Service - HTTP methods kavramlarını örneklerle açıklayınız.

Service Oriented Architecture (SOA) bir yazılım mimarisi yöntemidir. Bu yöntem, bir sistemi, birden fazla hizmetten oluşan bir dizi küçük parçaya böler ve bu hizmetler arasında haberleşmeyi sağlar. Bu sayede, SOA yöntemi, bir sistemi daha esnek, daha kolay anlaşılır ve daha kolay değiştirilebilir hale getirir. SOA yöntemi ayrıca, sistemin bileşenlerini bağımsız olarak geliştirmeyi ve dağıtmayı da kolaylaştırır.

Web Service, bir yazılım sistemini, internet üzerinden diğer yazılım sistemleriyle etkileşim halinde bulunabilen bir hizmet olarak tanımlayan bir kavramdır. Web Servisleri, genellikle bir web sunucusu üzerinden çalışan ve ağ üzerinden erişilebilen bir API (Application Programming Interface) olarak gerçekleştirilir. Bu sayede, farklı platformlar ve programlama dilleri arasında haberleşme sağlanabilir ve bir yazılım sistemi, diğer sistemlerle etkileşim halinde bulunabilir.

RESTful (Representational State Transfer) Service, web servisleri için bir tasarım prensibi ve kalıbıdır. RESTful Service, bir web servisinin URL yapısını, veri modelini ve HTTP metodlarını kullanarak, bir web servisini diğer sistemlerle etkileşim halinde bulunabilir hale getirir. Bu sayede, RESTful Service, web servisleri için bir standart oluşturur ve web servislerinin, internet üzerinden erişilebilir, anlaşılır ve kolayca kullanılabilir hale gelmesini sağlar.

HTTP (Hypertext Transfer Protocol) bir internet protokolüdür. HTTP, web tarayıcıları ve web sunucuları arasında veri transferini sağlar ve bu verilerin nasıl gönderileceği, nasıl alınacağı ve nasıl işleneceği hakkında standartlar belirler. HTTP, ağ üzerinden veri transferi için bir kalıp oluşturur ve bu sayede, web tarayıcıları ve web sunucuları arasında anlaşılır bir iletişim kurulabilir.