2. Yeni bir nesne oluştururken sürekli new keywordü ile oluşturmak yerine bir model ile nesne oluşturma yapısıdır.

Factory Method = Üst sınıflara nesneler üretmek amacı ile bir subclass yapısı olmasıdır. Mesela şekiller adında bir class sahibiyiz ve şekillerden oluşan subclasslar var. Yeni bir class içerisinde şekil üreten bir method yazıyoruz. Bu method sayesinde new kullanmadan classlar üretebiliyoruz.

Abstract Factory Pattern = Factory methodun gelişmiş hali diyebiliriz. Örneğin şekillerin renkli veye renksiz olma durumu var bir abstract class yapıyoruz ve factory method gibi yapılan bir class yapıyoruz. Abstract classın görevi renkli mi renksiz mi olacağına karar vermek diğer class ise şeklin tipini belirleyip yeni bir şekil üretiyor.

Singleton Pattern = Her nesnenin sadece bir örneği olması ve kod çalıştığında sadece bu nesnenin çağırılmasıdır. Bir class içerisinde private static bir new obje oluşturuyoruz. Daha sonra constructoru oluşturuyoruz. Main methodda çağırdığımızda her seferinde aynı obje geliyor.

Builder = Karışık yapısı olan bir objeyi adım adım yöntemi ile bir tek objeden oluşturmaktır. Class içinde static bir class oluşturup içine her bir değişkenin constructorunu ayrı ayrı yapmak.

Prototype = Birden fazla nesne oluşturmamız gerektiğinde new kullanmak yerine oluşturduğumuz nesnenin klonunu oluşturduğumuz bir paterndir. Mesela bir araba markası classı yaratıyoruz içerisinde özellikleri var bu classı Cloneable classına extend ediyoruz class içinde de clone methodunu override ediyoruz. Main methodda clone yapısını kullanarak yeni araba markaları üretebiliyoruz.

4.

Spring = Nesnelerin bağımlılıkları ve yaratılması, yazdığımız kodun yaşam döngüsü, paket kullanımı gibi şeyleri yöneten bir framework. Aşağıda ki gibi bir obje oluşturacağımızda spring bizim için örneğin userServicei kendi üretiyor.

@Configuration

**public** **class** BeanConfig {

@Bean

**public** User user() {

**return** **new** User();

}

}

@Service

**public** **class** UserService {

@Autowired

**private** UserService userService;

**public** **void** getUser(){

**return** userService.getAll();

}

}

Hibernate = Yazdığımız java programı ile database yönetim sistemi arasında bağlantıyı sağlayan, veri çekme, sorgulama işlerine yardımcı olan bir framework.

Struts = Doğrulama desteği , entegrasyon desteği gibi konularda yardımcı olan bir framework.

JavaServer Faces = Back end developerların işini kolaylaştırmak için yazılmış bir framework. Daha az kod kullanarak front end dizayn yapmayı sağlıyor.

Google Web Tookit = Karmaşık JavaScript front end uygulamalarını yazabilmemizi sağlıyor.

Blade = Verimli ve etkili bir şekilde hızlı web uygulamaları yapmamızı sağlamaktadır. Çok fazla katman gerektirmeyen bir yapıya sahip olduğu için anlaşılması kolay bir tasarımı var.

Apache Wicket = HTML ve Java kodunu kullanarak kullanıcı dostu uygulamalar yazmamızı sağlıyor. Yerleşik Ajax işlevi sayesinde JavaScript kodu yazmak zorunda kalmadan gerçek zamanlı olarak sayfa bölümlerini güncellememizi sağlar.

5.

Singleton, Factory, Proxy ve Template Design patternleri

Model View ve Front Control patternleri

View Helper pattern

Prototype pattern

Dependency Injection/Inversion of Control patterni

6.Service Oriented Architecture (SOE) = Yazılım bileşenlerini hizmet arabirimleriyle yeniden kullanılabilir ve çalışabilir hale getiren, ortak arabirim standartları ve mimari modeller kullanarak yapan bir yaklaşımdır. Örneğin bir banka sisteminin parayla ilgili, güvenlikle ilgili, müşteri yönetimi ile ilgili sistemleri vardır. Bu sistemleri güncellerken tüm uygulamayı etkilemeyecek bir sistem oluşturmak için SOE kullanılır. Farklı sistemler kendi işlerini yaparlar.

Web Service = Bir aplikasyonun başka bir aplikasyondan data alışverişi yapması için kullanılan open protocol ve standartları olan bir sistem. Bir mobil aplikasyonun web service kullanarak databaseden bilgileri isteyip gelen bilgileri kullanıcıya yansıtır bunun içinde web service kullanılır.

Restful Service = Representational State Transfer denen web servis yaratırken kullanılan kuralları belirten bir mimari yapı var. Okuma, güncelleme, yaratma ve silme gibi işlem isteklerini yapabildiğimiz bir uygulama programı arayüzüdür(API). Mesela gene bir mobile app kullanıcı adını databaseden istemesi ve göstermesi buna örnek olarak gösterilerbilir.

HTTP methods = Hypertext Transfer Protocol metodu client ile server arasında istek cevap protokolünü yapan bir methoddur. Restful servicede kullanılır. Connect, delete, get gibi işlemleri yapar. Bir internet sitesine giriş yapmak istediğimizde servera gönderilen get komutu buna bir örnek olabilir.