

Spring: Java için geliştirilmiş bir uygulama geliştirme framework'ü. Hibernate gibi frameworkleri içinde barındırır. Barındırdığı kütüphaneler sayesinde uygulama yazmayı ciddi bir şekilde kolaylaştırır falan.

Dependency injection: Bağımlılık oluşturacak parçaların ayrılıp bunların dışardan verilmesiyle sistem içerisindeki bağımlılığı minimize etme işlemi.

Spring IOC container: Spring frameworkün özü. Objelerin yaratılışı, konfigürasyonu ve dependencylerinin birleşimi sağlayan yani bütün life cycle'ını kontrol eden bir şey. Dependency injection kullanıyor.

Inversion of control: Bir yazılım tasarım prensibi. Obje instancelerin yönetimi sağlanarak bağımlılık en aza indirgenir. Bağımlılıkların yönetilmesinin yazılımcı yerine framework'e bırakılması.

Hibernate: Java için geliştirilmiş ORM kütüphanesi. Veritabanı ile olan ilişkiyi sağlayarak, veritabanı üzerinde yapılan işlemleri kolaylaştırmakla birlikte kurulan yapıyı da sağlamlaştırmaktadır.

Autowired: Bu annotation olduğunda spring o obje type'ındaki classı bulup çekiyor ve objeyi direkt instantiate ediyor.

Bean: Spring IOC container tarafından yönetilen objeler

Entity: bir databasedeki tabloya karşılık gelen java classı.

Dto: Bir tablodan çektiğimiz veriyi istediğimiz formatta (?) çekmeye yarayan java objesi. Yani database ile uygulamamız arasından veri aktarımını sağladığımız obje.

Converter: databaseden gelen objeyi entity'den dto'ya mapleyerek çeviren şey.

JPA: Verileri kalıcı olarak saklamak, almak, bulmak, silmek ve güncellemek için kullanılan Java orm kuralları.

ORM: Veritabanı ile programlama dili arasında ilişki kurarak, her bir alanı programlama diline uygun veri yapısıyla eşleştirmek için kullanılan bir yapı.

Servlet: Web sunucusu üzerinde çalışan ve gönderilen veriye göre sonuç üreten java sınıfları. http istemlerine cevap vermek için kullanılır. Server ve clientin haberleşmesini yönetmek için kullanılan sınıf ve arayüz topluluğu.

FİX MÜLAKAT SORULARI:

Polimorfizm: Bir nesnenin farklı bir nesne gibi davranması. Hayvan ana objem olsun, kedi ve köpek bunun child objeleri olsun. Ben sesçikar metodunu çağırdığımda kedi ve köpek birbirinden farklı sonuç verir?

Inheritance: Ortak özelliklere sahip objeleri bir ana (parent) objede toplayıp, modifikasyonları child objelere yansıtırız. Anne-baba-çocuk örneği ver

Interface vs abstract class: Abstract class dediğimiz şey subclassların implement veya override edebildiği fonksiyonalliteyi yazmamıza yarıyor. Interface fonksiyonalliteyi belirtmemize yarıyor fakat o fonksiyonalliteyi implement ettirmiyor. Abstract dediğimiz bu bir x'tir. Interface dediğimiz bu x'i yapabilir. Interfacede sadece boş metod var ve her şey public. Abstractta constructor olabilir, metodlar abstract değilse statik olabilir.

Checked vs unchecked exception: Checked compile time sırasında filenotfoundexception unchecked compoilerin kendi hallettiği exceptionlar null pointer exception. Compiler dosya açmaya çalışırken filenotfoundexception için try catch bloğuna almamızı ister.

Overriding: kalıtımda aldığımız metodu ezmek

Overloading: bir sınıf içinde metodu aynı isimle aynı dönüş değeriyle fakat farklı parametrelerle kullanmak.

Serialization: serializable'yi implement ederek bi java objesini byte stream olarak kaydetmemize yarar. Sonra okurken direkt java objesine castlayabiliriz

Error vs exception: Erroru handle edemeyiz outofmemoryerror, exceptionları try catchle handle edebiliriz.

Abstraction: gereksiz ayrıntıları göz ardı ederek bi nesnenin sadece gerekli özelliklerini belirleme işlevi.

Coupling: iki sınıfın birbirine bağlılığı.

Cohesion: bir sınıfın oluşturulma amacını ne kadar temsil ettiği.

SOLID prensipleri:

S->single responsibility

O->open/closed, her class geliştirmeye açık fakat değişime kapalı olmalı. Mesela bi savetofile(typeoffile) fonksiyonu olan file classı var ve typeoffile'ye göre farklı işlemler yapıyor. Bunun yerine file abstract class olarak tanımlanıp bunu extend eden file isimleriyle classlar oluşturulabilir.

L->liskov substitution: alt sınıflardan oluşturulan nesneler üst sınıfların nesneleriyle yer değiştirdiklerinde aynı davranışı göstermek zorunda. Türetilen sınıf türeyen sınıfın bütün özelliklerini kullanmak zorunda.

I->interface segregation: Her interfacenin belirli bir amacı olmalı. Tüm metodları kapsayan tek bir interface yerine farklı metod gruplarına hizmet veren birkaç interface tercih edilmeli.

D->dependency inversion: Üst seviye sınıflar alt seviye sınıflara bağlı olmamalı. İlişki abstraction veya interface kullanarak sağlanmalı. Alt sınıf interface sayesinde soyutlaştırılarak üst seviye sınıfımızda alt seviye sınıfına dair olan bağımlılığı tersine çeviriyoruz.

Integer vs int: int-> primitive, null alamaz, sınıf değildir, metodu yoktur. Integer sınıf, parseInt var

Upcasting: child objeyi parent objeye castlemek

Downcasting: parenti childa castlemek

Join: ?????????? bakıyım ben buna bi iyice

Reflection: bir uygulama geliştirme arayüzü, o anki jvmdeki sınıfları ve arayüzleri ve nesneleri yansıtan kütüphane

Singleton design pattern: bir classtan sadece 1 instance yaratılmasını sağlar

Factory design pattern: yazdığımız programda birbirine benzeyen birden fazla sınıf olabilir. Her seferinde new'le yeni sınıf yazmayalım diye oluşturulmuş bir pattern.

Observer design pattern: notification sistemi gibi. Subject, observer ve client var. Subjectte observer listesi var, clientta bi değişiklik olduğunda gerekli observerları updateliyor.

JVM vs JDK: JVM java dosyası derlendiği zaman java byte kodları oluşturur, jvm de makine kodlarına çevirir.

JRE: java uygulamasını çalıştırmak için gerekli kütüphanelerini ve jvmyi içerir.

JDK: java ile geliştirme yapmamız için yazdığımız java kodlarını compile edip .class uzantılı dosyaları oluşturmamızı sağlayan tool.

JDK = jre + derleyici + compiler

Trigger: bir veri tabanında bir tabloda belirli olaylar meydana geldiği zaman ekleme güncelleme silme fln biri gerçekleşmeden önce veya sonra çalışan yordam.

Autoboxing vs boxing: boxing -> intten integrera. Autoboxing de compilerin bunu otomatik halletmesi.

Process vs thread: Process dediğimiz şu an execute edilen herhangi bir program. Thread ise processin bi segmenti. Processler izole threadler memoryi paylaşır.

RMI nedir: farklı sunucularda olan metodların başka bir sunucudan çağırılarak kullanılması. Java nesnelerini kullanmak için farklı bir sunucuya veya sanal makinaya bağlanıyorsa buna dağıtık programlama denir.

JIT: bilgisayar kodunu çalıştırmanın bir yolu. Çalışma zamanında bir derleyici içerir. Çalıştırılacak bytecedun sadece kullanılacak kısmı derlenir.

Stringbuilder vs stringbuffer: builder bir tık daha hızlı. Hız farkının kaynağı senkronizasyon için harcanan vakit.

Instance nedir: test b = new test() -> b = instance

Constructor: class ilk çağırıldığında çalışan metod. Parametre alabilir fakat değer döndürmez, overload edilir override edilmez.

Finalize: obje garbage collector tarafından yok edilmeden önce çağırılan son metod.

Native: bir java programı içerisinde java harici bir programlama dilinde yazılmış bir program çağırabilmek.

Super ve this: süper parent class içindeki metotlara fieldlara ve constructorlara erişmek için kullanılır. This objenin kendisine erişmek için

Annotation: bir bileşene özellik katma veya konfigürasyon amaçlı kullanılan şey

XML vs JSON: json okunması ve kullanımı kolay bir kategorileme dili. Xml veri alış verişi sırasında büyük ve yavaş.

Varchar vs nvarchar: karakter bilgisini saklama biçimi. Nvarchar Unicode karakter serisi

Div vs span: div ekranda gidebildiği kadar gider. Spanda ise inline etiket, yani etiketin kapsamı içindeki elemanın kapsadığı yer kadar.

Comparable ve comparator: listlerde int ve string gibi değerleri değil de java sınıfını istenen veriye göre sıralar.

İnner sınıf: çoklu kalıtım yok javada. Bunun yerine dahili sınıf ve interfacerler yapar bunu. Kod okunurluğunu sağlar. Bir sürü sınıf yerine, her sınıfın ilgili olduğu alt sınıfları oluştururuz.

Readonly vs const: constta tanımlandığı anda değer verilmesi gerekir ama read only buna ek olarak constructor içinde de değer verebilir.

Immutable: değişmeyen demek. Primitive olan int decimal bool gibi tipler dışında kalan tüm değerler immutable. Değeri değişmez. Primitive olan değişkenlerin değerleri bellekte tutulduğunda başka bir değer atandığında eski değerın üstüne yazılır, fakat diğer tipler değerlerin adresleri heapta tutulduğu için değerleri değil de adresleri değişiyor.

Framework vs library: library bir kütüphane. Framework ise tüm altyapıyı oluşturabilen mimari çatı.

Ajax: tüm sayfayı yenilemeden kısmi yineme sağlayan yapı.

Web servis: http protokolü ile xml/json gönderip alarak iki uzak cihaz arasındaki iletişimi sağlayan bir haberleşme yöntemi. Xml/json olduğu için değişik platform ve programlama dilleri arasında haberleşme sağlanabilir.