**1) Class nedir? ne işe yarar? Fonksiyonla farkı nedir? Abstract ve Interface kavramları ile ilgili 1-2 cümle yorumlarınız nedir?**

**Class nedir?**

Class bir nesneyi oluşturacak benzer özelliklere sahip metot ya da değişkenleri bir araya topladığımız yapıdır.

**Ne işe yarar?**

Class ile proje içerisinde aynı metotları, değişkenleri tekrarlamaktan, teker teker düzeltmekten kurtuluruz, başka projelerde de referans alıp kullanabiliriz.

**Fonksiyonla farkı nedir?**

Class temelde içine fonksiyonda eklenebilen nesneyi oluşturan şablondur diyebiliriz.

Fonksiyonlar ise içinde class barındıramayan istediğimizde tekrar çağırabildiğimiz kod parçalarıdır.

**Abstract ve Interface Kavramları**

Abstract sınıflar, soyutlama sağlayan diğer sınıflar için genellikle temeli sağlayan sınıflardır.

Abstract sınıfların kendilerini miras alan sınıflar ile arasında “is-a” ilişkisi mevcuttur.

Interface, kendisinden türeyecek olan sınıfların doldurması için boş metot tanımlarının bulunduğu ve soyutlama yapmamıza olanak sağlayan yapıdır.

Interface ile kendini miras alacak sınıf arasında “can-do” ilişkisi mevcuttur.

**2) C# Entity framework ile Group by nasıl yapılır? Join nasıl yapılır?**

**Group by nasıl yapılır?**

using (var db = new DBEntities()) //db entities tanımlandı, db nesnesi türetildi.

{

var degerler = db.TABLO.OrderBy(x=>x.stnx).GroupBy(y => y.stnx).

//tablodaki stnx = sütunx değerlerini OrderBy ile sıralayıp GroupBy ile gruplandırdık

Select (z => new {stnx = z.Key, Toplam=z.Count });

//seçme işlemini z.Key gruplandırma işlemindeki değerler,z.Count ise tekrar sayısını stnx ve Toplam sütununa yazdırarak yaptık.

dataGridView1.DataSource=degerler. ToList ();} //yeni tabloyu listeledik.

**Join nasıl yapılır?**

using (var db = new DBEntities()) //db entities tanımlandı, db nesnesi türetildi.

{

var sorgu = from x in db.Tablox // Tablox çağırıldı

join y in db.Tabloy // tablo y çağırılıp eklendi

on x.Idx equals y.Idy // Id karşılaştırması yapıldı

select new { x.stnx,y.stny }; // Tablo1 densütunx,Tablo2 den sütuny seçildi.

dataGridView1.DataSource=sorgu. ToList ();} // yeni tablo listelendi.

**3) Java JHipster ile Group by nasıl yapılır? Join nasıl yapılır?**

entity Driver

entity Car

relationship ManyToMany {

Car{driver} to Driver{car}

}

entity Citizen

entity Passport

relationship OneToOne {

Citizen{passport} to Passport

}

JHipster’ da Group by ve join için yukarıdaki gibi farklı ilişkiler mevcuttur.

**4) Java ya da C#'ta fibonacci sayılarını hesaplayan bir program nasıl yazılır? (recursive nedir?)**

Recursive türkçede ‘özyinelemeli’ demektir.

Bundan yola çıkarak Recursive fonksiyon içinde kendini tekrar çağıran(özyineleyen) fonksiyondur.

import java.util.Scanner;

class fibonacci{

//Recursive Fonksiyonumuz

static int fibo(int n) {

if (n <= 1) {

return n;

}

else {

return fibo(n-1) + fibo(n-2); // return ile recursive ediyoruz.

}

}

// main(ana) fonksiyonumuz

public static void main(String[] args) {

Scanner oku = new Scanner(System.in);

System.out.println("Fibonacci uzunlugunu giriniz");

int fib=oku.nextInt();

int sonuc=fibo(fib);

System.out.println("Fibonacci sonucu = "+sonuc);

}

}

**5) JSON nedir? XML nedir? Birbirlerinden farkları, avantaj ve dezavantajları nelerdir?**

**JSON nedir?**

JSON (JavaScript Object Notation), farklı sistemlerin veri alışverişi yapabilmesi için verileri basit şekilde metinler olarak kaydederek oluşturulmuş metin biçimidir.

**XML nedir?**

XML (Extensible Markup Language ya da Türkçesiyle Genişletilebilir İşaretleme Dili), insanlar tarafından kolayca okunabilen veri iletişimini standart hale getirmek için tasarlanan bir işaretleme dilidir.

**Birbirlerinden farkları**

|  |  |
| --- | --- |
| **JSON** | **XML** |
| JSON verileri nesne türündedir | XML verileri türsüzdür |
| JSON türleri: string, number, array, Boolean | Tüm XML verileri string olmalıdır |
| Verilere kolayca JSON nesneleri olarak erişilebilir | XML verilerinin ayrıştırılması gerekir |
| JSON çoğu tarayıcı tarafından desteklenir. | XML ayrıştırması zor olabilir |
| JSON'un görüntüleme özelliği yoktur. | XML, bir biçimlendirme dili olduğu için verileri görüntüleme yeteneği sunar. |
| JSON, yalnızca metin ve sayı veri türünü destekler. | XML, sayı, metin, resim, çizelge, grafik vb. gibi çeşitli veri türlerini destekler. |
| Birçok Ajax araçseti tarafından desteklenir | Ajax araç seti tarafından tam olarak desteklenmiyor |
| JavaScript ile sıkıştırılıp çıkartılması kolaydır | Sıkıştırıp çıkartılması için JavaScript kodu yazılmalıdır |
| Nesne için doğal destek. | Nesne, kurallarla ifade edilmelidir |
| Yalnızca UTF-8 kodlamasını destekler. | Çeşitli kodlamayı destekler. |
| Açıklamaları desteklemez | Açıklamaları destrekler |
| JSON dosyalarının okunması, XML'e kıyasla daha kolaydır. | XML belgelerinin okunması ve yorumlanması nispeten daha zordur. |
| Ad alanları için herhangi bir destek sağlamaz. | Ad alanlarını destekler. |
| Daha az güvenlidir. | JSON'dan daha güvenlidir. |

**JSON kullanmanın avantajları**

* Tüm tarayıcıları destekler
* Okuması ve yazması kolay
* Basit sözdizimi
* eval() işlevini kullanarak JavaScript'te ayrıştırabilirsiniz.
* Oluşturulması ve işlemesi kolay
* Tüm büyük JavaScript çerçeveleri tarafından desteklenir
* Çoğu arka uç teknolojisi tarafından desteklenir
* JSON, JavaScript tarafından yerel olarak tanınır
* Bir ağ bağlantısı kullanarak yapılandırılmış verileri iletmenize ve serileştirmenize olanak tanır.
* Modern programlama dilleri ile kullanabilirsiniz.
* JSON, herhangi bir JavaScript nesnesini JSON'a dönüştürülebilen ve bu JSON'u sunucuya gönderebilen metindir.

**JSON kullanmanın dezavantajları**

* Ad alanı desteği yok, dolayısıyla zayıf genişletilebilirlik
* Sınırlı [geliştirme araçları](https://www.guru99.com/software-development-tools.html) desteği
* Resmi dilbilgisi tanımı için destek sunar

**XML kullanmanın avantajları**

XML kullanmanın önemli yararları/eksileri şunlardır:

* Belgeleri sistemler ve uygulamalar arasında taşınabilir hale getirir. XML yardımıyla farklı platformlar arasında hızlı bir şekilde veri alışverişi yapabilirsiniz.
* XML, verileri HTML'den ayırır
* XML, platform değiştirme sürecini basitleştirir
* Kullanıcı tanımlı etiketler oluşturmaya izin verir.

**XML kullanmanın dezavantajları**

* XML bir işleme uygulaması gerektirir
* XML sözdizimi, bazen kafa karıştırıcı olan diğer alternatif 'metin tabanlı' veri iletim biçimlerine çok benzer.
* İç veri türü desteği yok
* XML sözdizimi gereksiz

**6) React'ta component nedir? ne işe yarar?**

**React'ta component nedir?**

React Component, kullanıcı arayüzünü şekillendiren, belirli görevleri olan, class veya fonksiyon olarak tanımlanmış, geriye react elementleri döndüren, opsiyonel olarak parametre alan yapılardır.

**ne işe yarar?**

Componenetler bağımsız ve yeniden kullanılabilir kod parçalarıdır. Uygulamayı parçalar halinde yöneterek kodlamamıza olanak sağlar.

**7) React'ta props nedir? state nedir?**

**Props nedir?**

Props componentlere yine farklı component içerisinden veri gönderilebilmesini sağlayan kısacası componentler arası iletişimi gerçekleştiren yapıdır, fonksiyon, string, integer, array, object olabilir ve gönderildiği componentde sadece okunabilir.

**State nedir?**

React State, bileşenlerdeki dinamik verileri tutmaya yarayan ve bileşenin alt veya üst bileşenlerle ilişkisinde yardımcı olan, bileşenin projenin tamamı içinde nasıl davranacağını belirten bir javascript objesidir. Bu tutulan değerler, fonksiyon, string, integer, array, object olabilir.