

1- MVC kavramını açıklar mısınız ? Neden ihtiyaç duyuluyor. Java'da nasıl kurgulanıyor. Object Oriented katmanları nelerdir ?

Cevap:

MVC, Model-View-Controller kelimelerinin kısaltılmış hali olup, yazılımdaki katmanları belirten bir tasarım desendir. Bu tasarım deseni, uygulamanın farklı katmanlarını ayrı ayrı ele alıp bağımsız bir şekilde geliştirilmesini sağlar. Böylece uygulamanın geliştirilmesi, genişletilmesi, bakımı ve test edilmesi kolaylaşır.

Desendeki katmanlar şu şekilde açıklanabilir:

Model: Bu katman veritabanına erişim ve iş mantığının (business logic) geliştirildiği katmandır.

View: Bu katman, kullanıcının işlemler yapmasını sağlayan ve kullanıcıya yaptığı işlemler sonucunda bir web sayfasının gösterildiği katmandır. Kullanıcı arayüzü, HTML, CSS, JS gibi diller ile sağlanabilmektedir. Görsel öğeler bu katmanda oluşturulur ve kullanıcıya sunulur.

Controller: Bu katman, Model ile View arasındaki bağlantıyı sağlar. Kullanıcının isteklerini işleyen, Model katmanından veri çeken ve kullanıcıya gösterilen View'ın ayarlanması bu katmanda gerçekleşmektedir.

Java'da MVC'nin kurgulanması için her katman ayrı bir pakette ve sınıfta tanımlanmalıdır. Böylece her katmanın sadece kendi işinden sorumlu olması sağlanabilmektedir. Örneğin, sunucuya gelen bir GET isteği Controller tarafından karşılanır, Model katmanı ile veritabanına erişim sağlanır , gerekli filtrelemeler yapılabilir ve kullanıcıya isteğin cevabını dönen bir View oluşturulur. Bu katmanlar birbirleri ile doğrudan haberleşebilir ancak birbirlerinin işlerini kesinlikle yapmamalıdır.

OOP katmanları da aşağıdaki gibi sıralanabilir:

Kullanıcı Arayüzü Katmanı (User Interface Layer): Kullanıcı arayüzü katmanı, kullanıcının bir programla etkileşim kurduğu noktadır. Bu katman, kullanıcının girdilerini(isteklerini) alır ve sonuçları kullanıcıya gösterir.

İş Mantığı Katmanı (Business Logic Layer): İş mantığı katmanı, programın gerçek işlevselliğini sağlar. Bu katman, kullanıcı girdilerini işler, veritabanı ile iletişim kurar ve uygulamanın ana işlevselliğini sağlayan işlemleri gerçekleştirir.

Veri Erişim Katmanı (Data Access Layer): Veri erişim katmanı, veritabanı veya dosya sistemine erişim sağlar ve iş mantığı katmanı ile verilerin iletimini sağlar. Bu katman, verilerin güvenli ve doğru bir şekilde saklanması ve erişilmesini sağlar.

Nesne Erişim Katmanı (Object Access Layer): Nesne erişim katmanı, nesne yönelimli programlamada bir nesnenin özelliklerine ve davranışlarına erişimi sağlar. Bu katman, nesnelerin oluşturulmasını, değiştirilmesini ve silinmesini sağlar.

Bu katmanlar ile MVC katmanlarının çok benzer yapıda olduğunu görmekteyiz. Böylece uygulamada aslında birbirlerinden çok da farklı olmayan yapılar kullanılmaktadır.

2- Birbirinden bağımsız iki platformun birbiriyle haberleşmesi nasıl sağlanabilir. Örneğin, X platformu Java ile yazılmış olsun, Y platform u C# ile. Bu iki platformun bir biri ile iletişim halinde request-response ilişkisi kurması gerekiyor. Bu yapıyı nasıl sağlarız.

Cevap: Farklı platformlarda yazılmış uygulamalar arasında haberleşme sağlamak için gereken yöntemlerden bazıları şunlardır:

Web servisleri: REST ve SOAP gibi request-response yapısı kullanılarak farklı platformlarda yazılmış uygulamalar birbiri ile haberleşebilir. Haberleşmede kullanılan veri yapısı SOAP için yalnızca XML formatı iken REST servislerde JSON ve XML gibi veri formatları kullanılabilir.

Mesajlaşma sistemleri: RabbitMQ, ActiveMQ gibi mesajlaşma sistemleri farklı platformlarda çalışan uygulamalar arasında iletişimi sağlamak için kullanılabilir. Bu sistemler ile bu uygulamalar arasında asenkron haberleşme sağlanarak sistemlerin birbirinden bağımsız çalışması sağlanabilir.

Veritabanı paylaşımı: Veri tabanı kullanarak farklı platformlar arasında veri paylaşımı yapmak mümkündür. Bir platform, veritabanına veri ekler veya değiştirir, diğer platform da aynı veritabanından veriyi alır ve kullanır.

Socket programlama: Farklı platformlar arasında soket programlama kullanarak, low-level bir ağ protokolü üzerinden haberleşme sağlanabilir. Bu yöntemle, uygulamaların daha doğrudan ve esnek bir şekilde haberleşmesi mümkündür.

3- Bir web sayfasında ekran sürekli Backend' den veya bir başka yapı tarafından güncelleniyor. Siz, web sayfasını refresh etmeden bu güncel bilgiyi anlık ekrana nasıl yansıtırsınız.

Cevap: Bu işlemi gerçekleştirmek için yapılabilecekler şunlardır:

AJAX: AJAX, Asynchronous JavaScript and XML'in kısaltmasıdır. Bu teknoloji, web sayfasına bir istek göndererek sunucudan yeni veriler alır ve web sayfasını günceller. Bu işlem, sayfayı yenilemek zorunda kalmadan gerçekleşir.

Web Soketleri: Web soketleri, tarayıcı ile sunucu arasında gerçek zamanlı iletişim kurmak için kullanılan bir teknolojidir. Sunucu, herhangi bir değişiklik olduğunda bir mesaj gönderir ve tarayıcı hemen bu mesajı alır ve içeriği günceller. Bu yöntem, dinamik bir web sayfası oluşturmak için popüler bir seçenektir.

4-Bir for döngüsü ile aşağıdaki çıktıyı yazar mısınız.

*
**

Cevap:

```
public class Main {  
    public static void main(String[] args) {  
        final int ROW_COUNT = 6;  
        for (int i = 0; i < ROW_COUNT; i++) {  
            if (i == 0) {  
                System.out.print("*");  
            }  
            for (int j = 0; j < i; j++) {  
                System.out.print("***");  
            }  
            System.out.println();  
        }  
    }  
}
```

5- Firmada çalışan için sana remote bir linux server verildi. Elinde ip adresi port bilgisi kullanıcı adı ve şifren var. Server a erişimi nasıl test edersin, Server a nasıl erişirsin, Server a nasıl dosya atarsın, Serverdan nasıl dosya çekersin.

Cevap: Öncelikle ip adresine ping atılıp atılmadığını denerim. Ardından SSH komutu ile komut satırından ya da terminal aracılığıyla **ssh kullanıcıadı@ip_adresi -p port** komutu ile bağlantıyı test ederim. Server'a eriştikten sonra da SCP komutunu kullanarak dosya aktarımını gerçekleştiririm. Örnek olarak:

scp /local/makine/dosya/yolu kullaniciadi@ip_adresi:port /uzak/makine/dosya/yolu

7- Apache Solr servisine yazılacak bir query örneği Apache Solr kullanılan sql programlarından daha farklı runtime bir database. Solr a hali hazırda kayıtlı bir alan olduğunu düşünelim. Alanın ismi “updatedAt” long tipinde tutulan bir alan. Ben 2020 Ocak ayından sonraki verileri getir dediğimde solr a nasıl bir query yazılmalı. <http://example?query=> kısmını nasıl doldurmalıyım?

Cevap: Solr'da belirtilen sorguyu atacak olan query şöyledir:

[http://example?fq=updatedAt:\[1577836800000 TO *\]](http://example?fq=updatedAt:[1577836800000 TO *])

Burada fq 'filtered query' anlamına gelip eğer kullanılması mümkünse daha performanslı çalıştığı kullanan kişilerce[1] belirtilmiştir. Bu query cache'leme yapısını da kullanabildiği için bu avantaja sahiptir. **1577836800000** sayısı unix timestamp olarak 2020 Ocak ayına denk gelmektedir. * kısmı ise erişilebilecek en son tarih anlamına gelmektedir.

[1]-<https://stackoverflow.com/questions/11627427/solr-query-q-or-filter-query-fq>