

1-Bu soru çok detaylı bir şekilde incelenebilir ama ben biraz özet geçmeye çalışacağım. Öncelikle MVC'nin açılımı Model-View-Controller'dır, ve bu bir tasarım desendir. Uygulamanın modüler olmasında ve gelecekte yapılacak değişikliklerde kolaylık sağlar, yani sürdürülebilirlik konusunda yardımcı olur. Kısacası yapılacak işi katmanlara ayırarak her bir parçayı bağımsız hale getirmeyi, bundan dolayı hata çözümleri ve geliştirmelerin daha hızlı olmasına izin verir.

Model Katmanı - uygulamadaki veri ve iş mantığını temsil eder, verilerin yönetimi ve verilere erişim bu katman üzerinden yapılır.

View Katmanı - Arayüz bu katman üzerinden oluşturulur, modelin o anki halini görsel olarak kullanıcıya sunar.

Controller Katmanı - Kullanıcı ile iletişim bu katman üzerinden olur, girdiler çıktılar vs. bu katman üzerinden yapılır. Model ve view katmanı arasındaki köprü gibi düşünülebilir.

Java'da MVC mimarisi çeşitli yollarla kurgulanabilir ama en popülerleri Java Swing, Java Spring ve JavaFx'dir. Örnek olarak Swing'i ele alacak olursak, model katmanında Java'nın sınıflarını kullanarak entity'ler oluştururuz. View katmanı için Swing'in bize sunmuş olduğu bileşenleri kullanarak oluştururuz.

Object Oriented Programming bir yazılım paradigmasıdır, izlenmesinde fayda göreceğimiz kurallar gibi düşünebiliriz.

- Sınıflar(classes) birbirine benzeyen nesnelerin taslağını tanımlar ve nesneler(objects) ise bu sınıflardan üretilen örneklerdir.
- Kapsülleme(encapsulation) ise veri ve methodları bir araya getirmeye verilen isimdir, sınıfların içindeki verileri private(özel) olarak tanımlayıp bu verilere sadece sınıf içindeki methodlara erişimini sağlamaktır. Veri gizliliğini sağlar.

- Kalıtım(inheritance) bir sınıfın başka bir sınıfın özelliklerini miras alabilmesini belirten prensiptir. Örnek olarak Öğrenci sınıfının İnsan sınıfından miras almasını verebiliriz.
- Çok biçimcilik(polymorphism) ise aynı sınıfların ve methodların farklı durumlarda farklı davranışlar vermesini sağlayan prensiptir.
- Soyutlama(abstraction) ise karmaşık yapıları daha basite indirgeyerek gerekli özelliklere odaklanmayı sağlayan prensiptir.

2- Bunun için aslında birçok yöntem var ama ben örnek olarak RESTful API sistemini anlatacağım. Örneğin java Spring ile yazılmış bir API ile JSON formatında veri döneriz ve C#'da aldığımız bu veriyi kullanabiliriz.

3- Bunu yapabilmek için de birçok yöntem vardır fakat benim bildiğim iki tanesi var. Biri AJAX diğeri ise WebSockets. Örneğin AJAX bir sayfanın belirli bölümlerini veya verileri güncellemek için kullanılabilir. Sunucu tarafında bir değişiklik olduğunda, javascript kullanarak o kısmı sayfayı yenilemeye gerek olmadan güncelleyebiliriz.

```
4- public class StarPattern {  
    public static void main(String[] args) {  
        for (int i = 1; i <= 5; i++) {  
            for (int j = 1; j <= 2 * i - 1; j++) {  
                System.out.print("*");  
            }  
            System.out.println();  
        }  
    }  
}
```

5- Bunun için de aynı şekilde birçok yöntem var fakat en popüler SSH'dir. Örneğin console'a şu kodu yazarak giriş yapabiliriz.

```
ssh -p port_numarasi kullanıcı_adi@ipadres  
ssh -p 22 fettah@127.0.0.1
```

Bunu yazdıktan sonra şifre isteyecektir onu girmeliyiz. Erişimin testi için basit şekilde "ls" dosyalara erişip erişemediğimize bakabiliriz.

Localden remota'a şu şekilde veri çekebiliriz.

```
scp -P 22 ~/Belgeler/metin.txt fettah@127.0.0.1:/home/fettah/
```

Bu örnekte Belgeler'deki metin.txt dosyasını server'daki home/fettah yoluna kopyalayacaktır. Benzer şekilde de şöyle dosya çekebilir.

```
scp -P 22 fettah@127.0.0.1:/home/fettah/metin.txt ~/Belgeler/
```

Burdaki kod ise remote serverdaki home/fettah klasöründeki metin.txt belgesini local bilgisayarındaki belgeler klasörüne kopyalayacaktır.

6- Burada basit bir Java Spring uygulaması yazdım. RESTful API ile basit CRUD işlemleri yapmakta. Daha detaylı bir örnek yapabilirdim ama örnekte istenmediğinden ve hali hazırda yazmış olduğum bir Spring uygulaması bulunmamasından böylesi basit bir şeyi uygun gördüm.

<https://github.com/FettahB/EmployeeCompany>

7- Pek yetkin olmadığım bir konu bu, fakat yaptığım araştırmalar sonucunda böyle bir sonuca vardım. Küçük hataları olması muhtemeldir.

[http://example/solr/collection/select?q=updatedAt:\[2020-01-01T00:00:00Z%20TO%20*\]&rows=10&sort=updatedAt%20desc](http://example/solr/collection/select?q=updatedAt:[2020-01-01T00:00:00Z%20TO%20*]&rows=10&sort=updatedAt%20desc)