

# **Tasarım Örüntüleri**

## **Model View Controller**

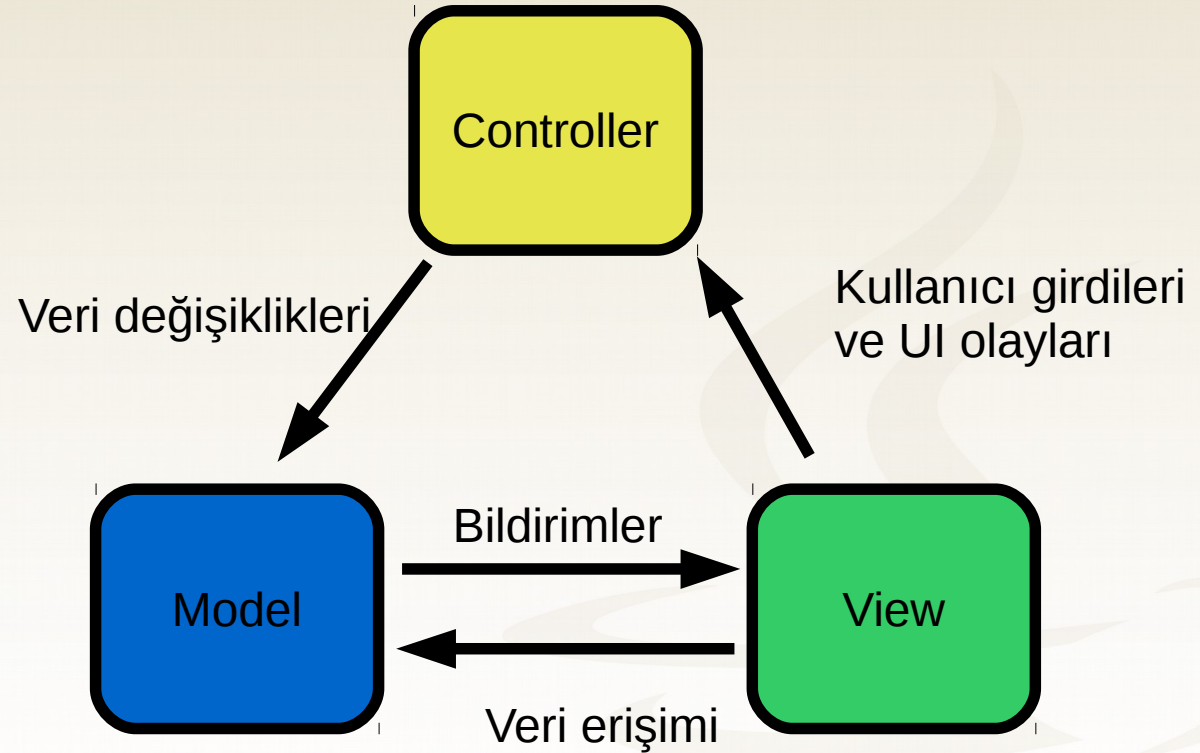
# Örüntülerin Temel Prensipleri

- GoF tasarım örüntülerinin altında yatan temel prensipler
  - Encapsulation
  - Composition
  - Abstract Data Types

# MVC Nedir?

- **Mimarisel** bir örüntüdür
- Uygulama içerisindeki bileşenlerin her biri **belirli bir takım sorumluluklar** üstlenir
- Her bileşenin veya bölümün **kendine özgü** ve diğerlerinden bağımsız bir **sorumluluğu** vardır
- Bu sorumluluklar çerçevesinde de **birbirleri ile etkileşime girerek** fonksiyonallite sergilerler

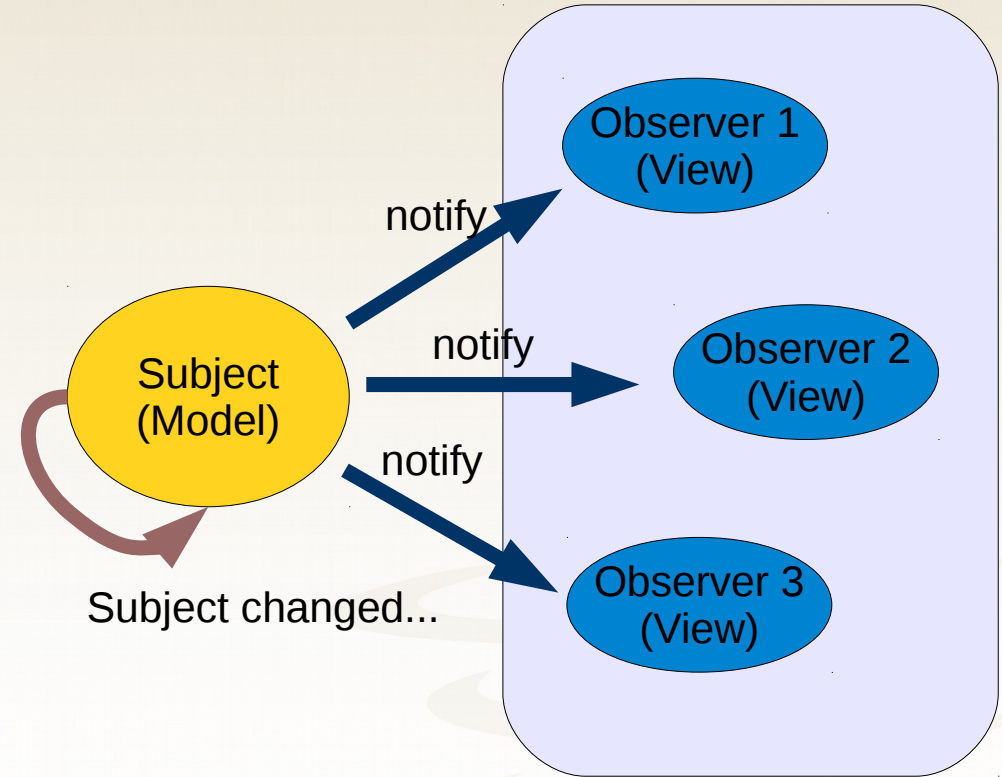
# MVC Bileşenleri



# MVC Bileşenleri Arasındaki Etkileşim

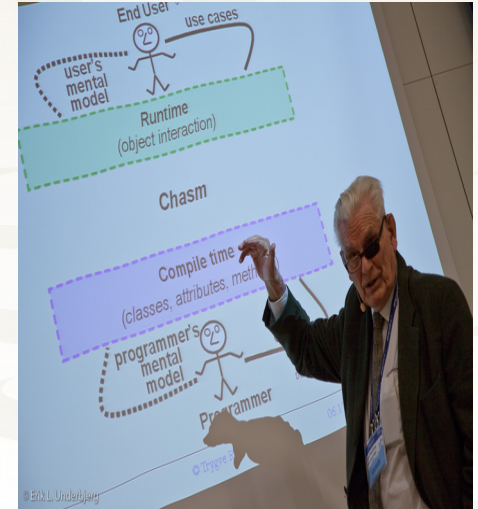
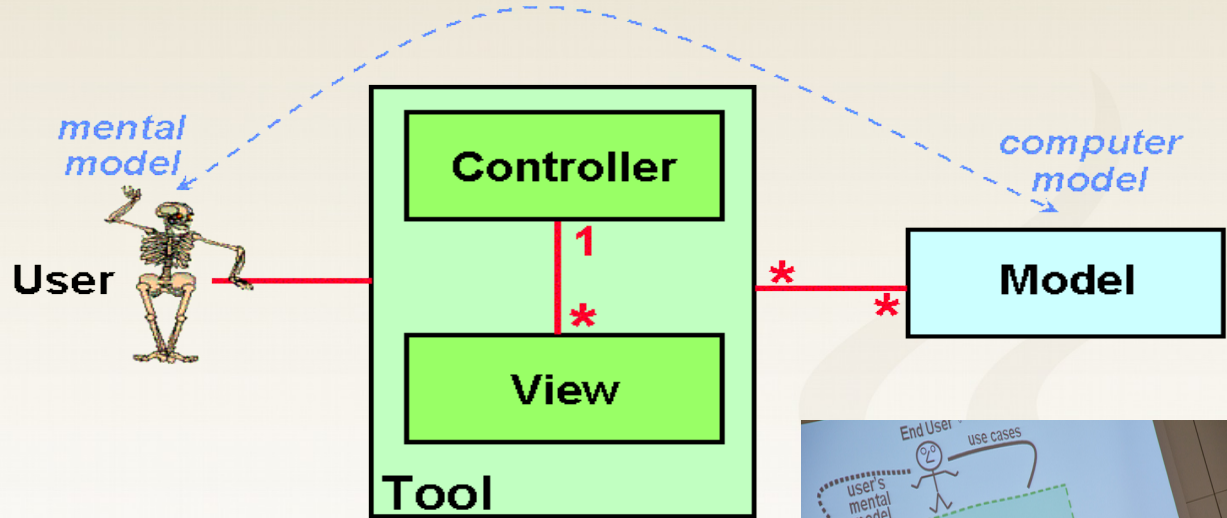
- **Model**, view tarafından görüntülenen **veriyi ifade eder**
- **View** ise kullanıcıya göstereceği veriye model üzerinden erişir ve **GUI render** işlemini gerçekleştirir
- **Controller**'da kullanıcı inputunun ele alınıp, **iş mantığının çalıştırılmasını** ve **model üzerinde değişikliğe gidilmesini** sağlar

# MVC ve Observer Örüntüsü

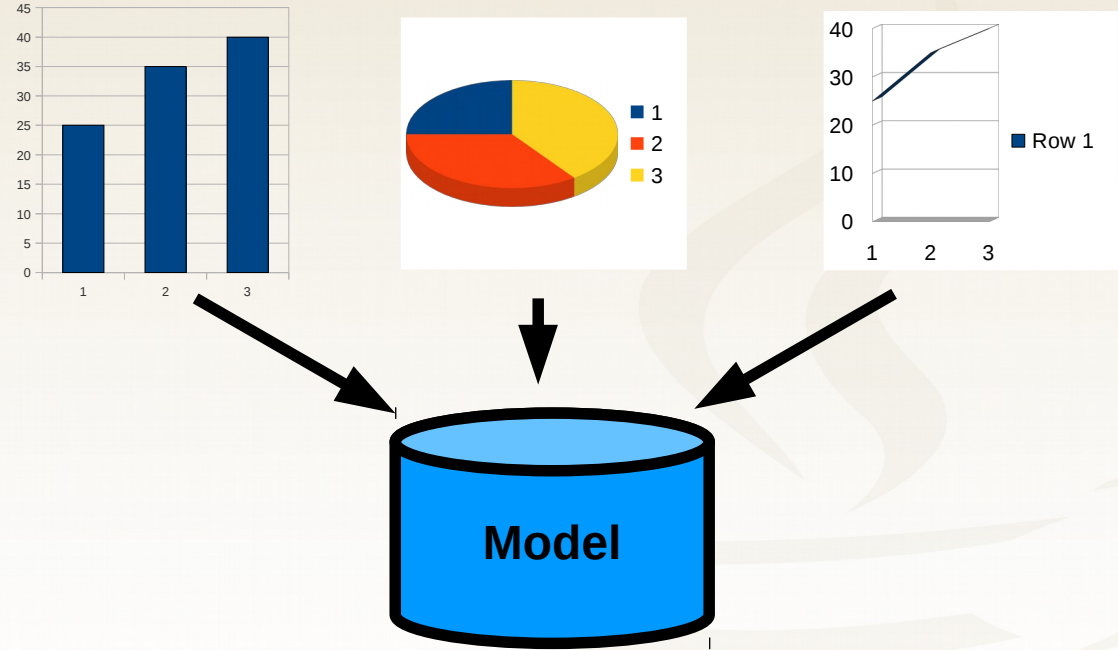




# MVC'nin Asıl Amacı?

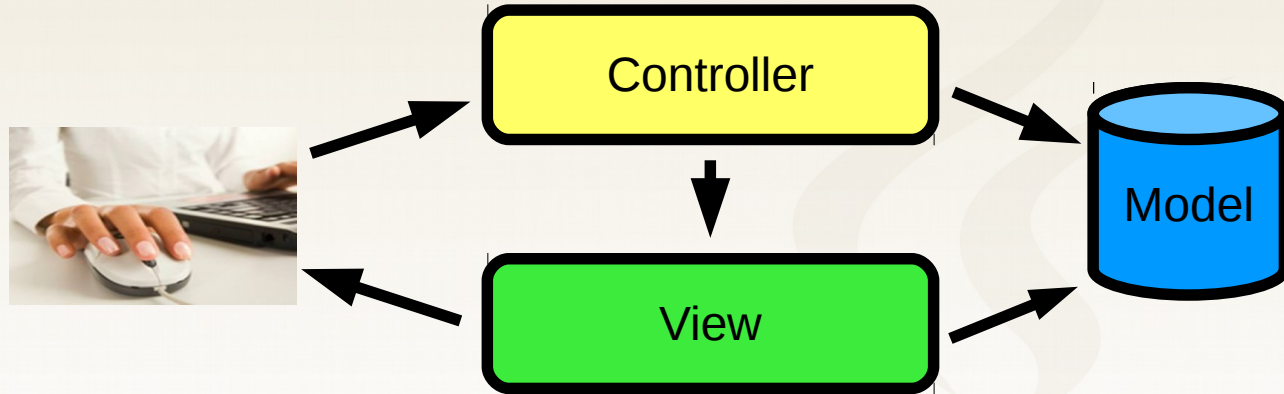


# MVC'nin Asıl Amacı?

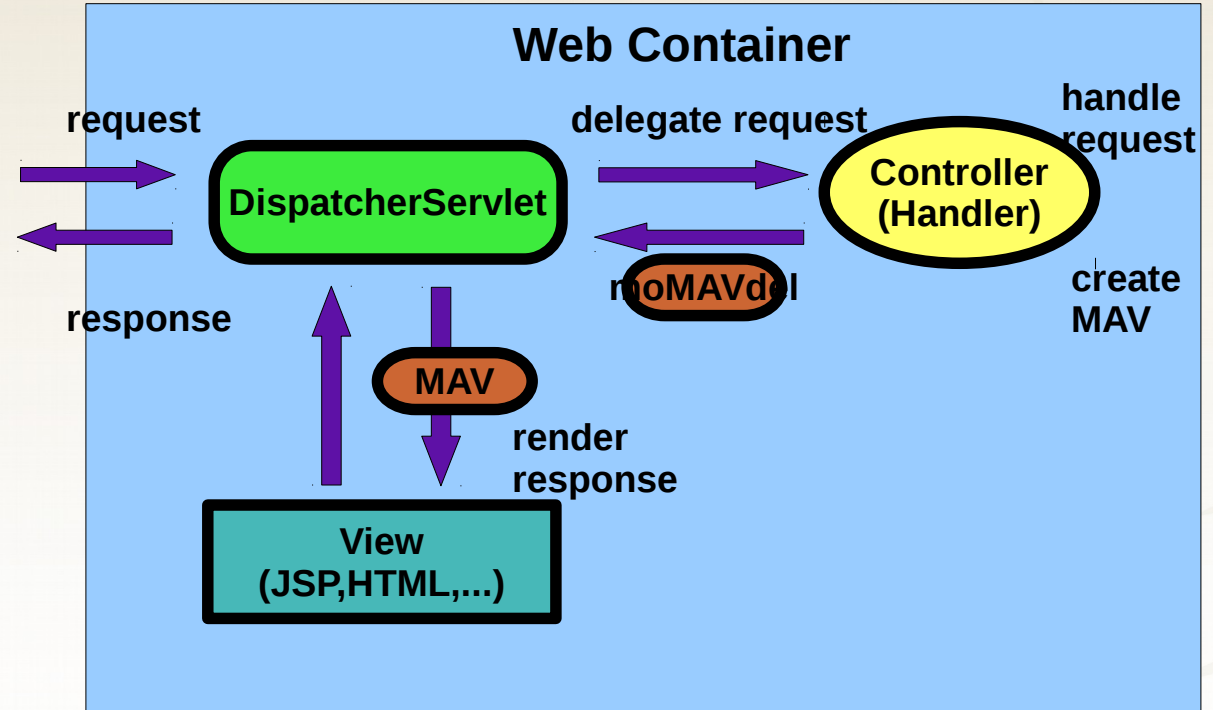




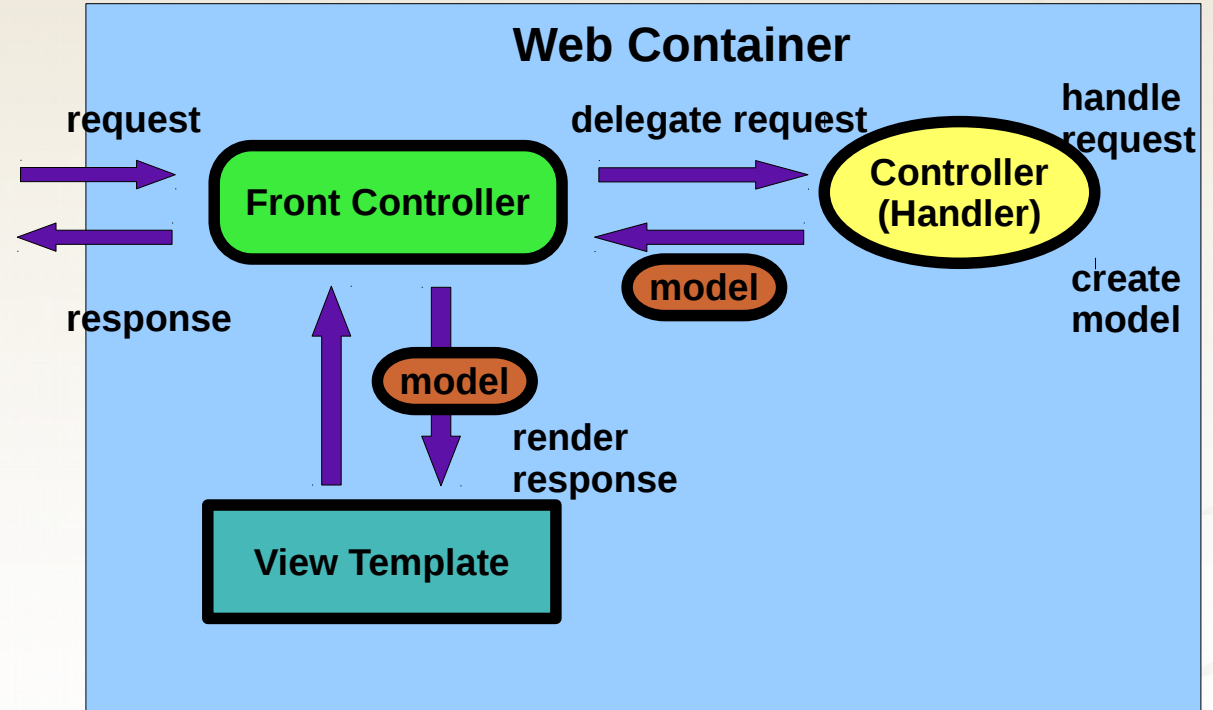
# MVC ve Web



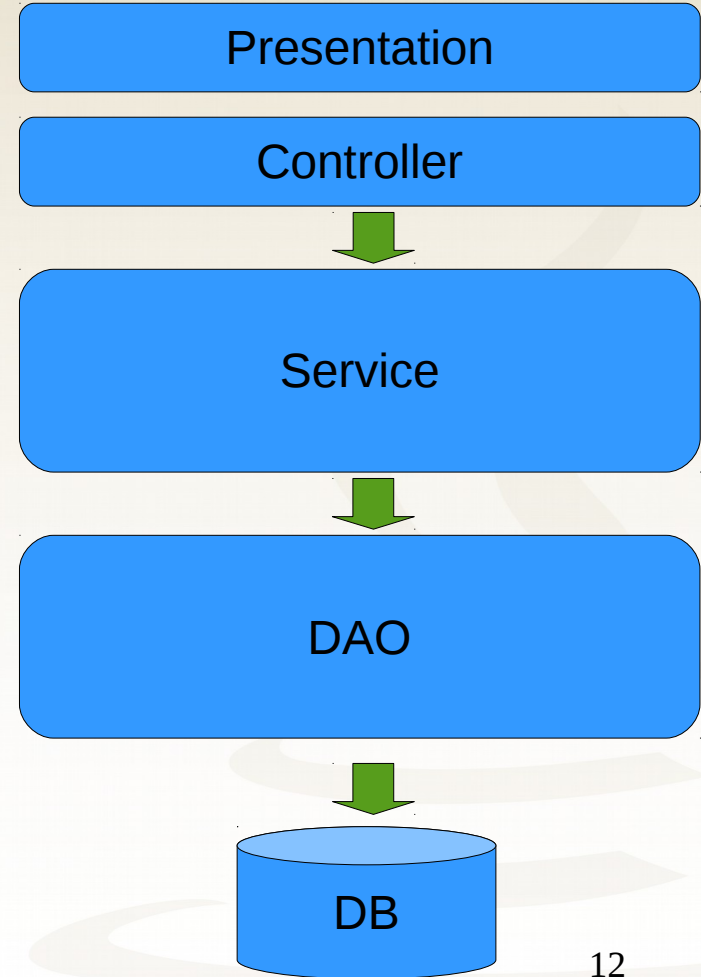
# Spring ve MVC



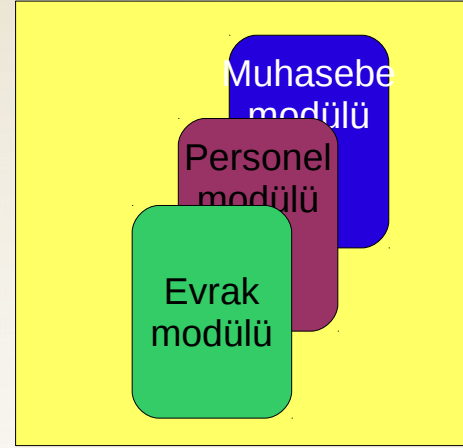
# MVC ve Front Controller



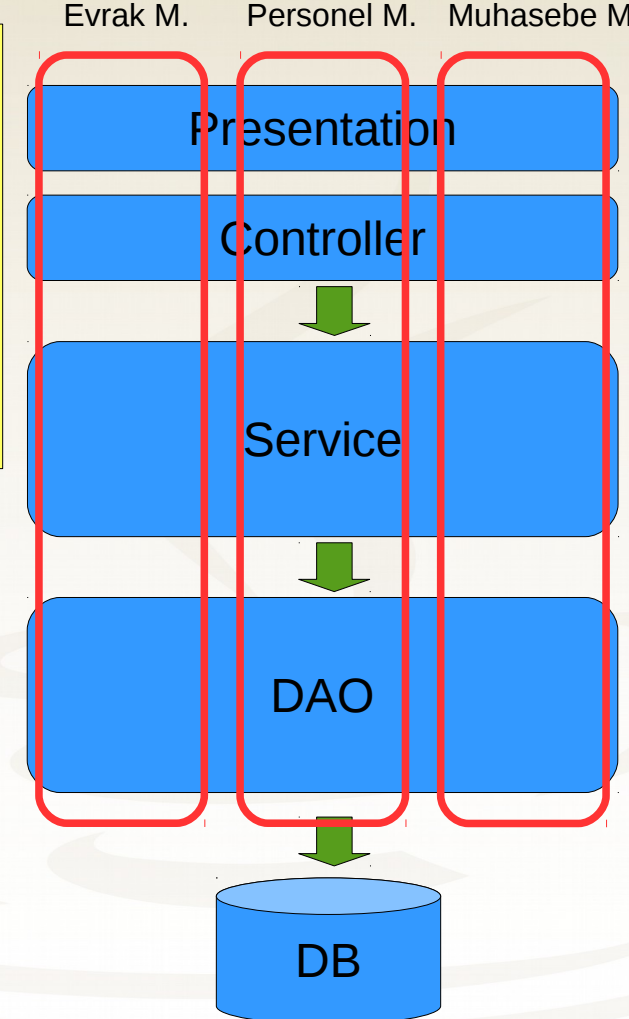
# MVC ve Katmanlı Mimari



# Katmanlı Mimari ve Modülerlik



Yazılım Sistemi



# MVC Mimarisel Örüntüsünün Sonuçları

- MVC'nin en büyük sonucu **“Seperation of Concern”**dir
- Bu sayede **modülerlik** sağlanır
- **Test edilebilirlik** ve **yeniden kullanım** imkanları artar
- Sistemin **farklı bölümlerinin farklı ekipler tarafından geliştirilmesini** mümkün kılar





## Kurumsal Java Eğitimleri



[www.java-egitimleri.com](http://www.java-egitimleri.com)



[info@java-egitimleri.com](mailto:info@java-egitimleri.com)



[@javaegitimleri](https://twitter.com/javaegitimleri)



[youtube.com/c/  
KurumsalJavaEğitimleri](https://youtube.com/c/KurumsalJavaEgitimleri)