

Command Örüntüsü

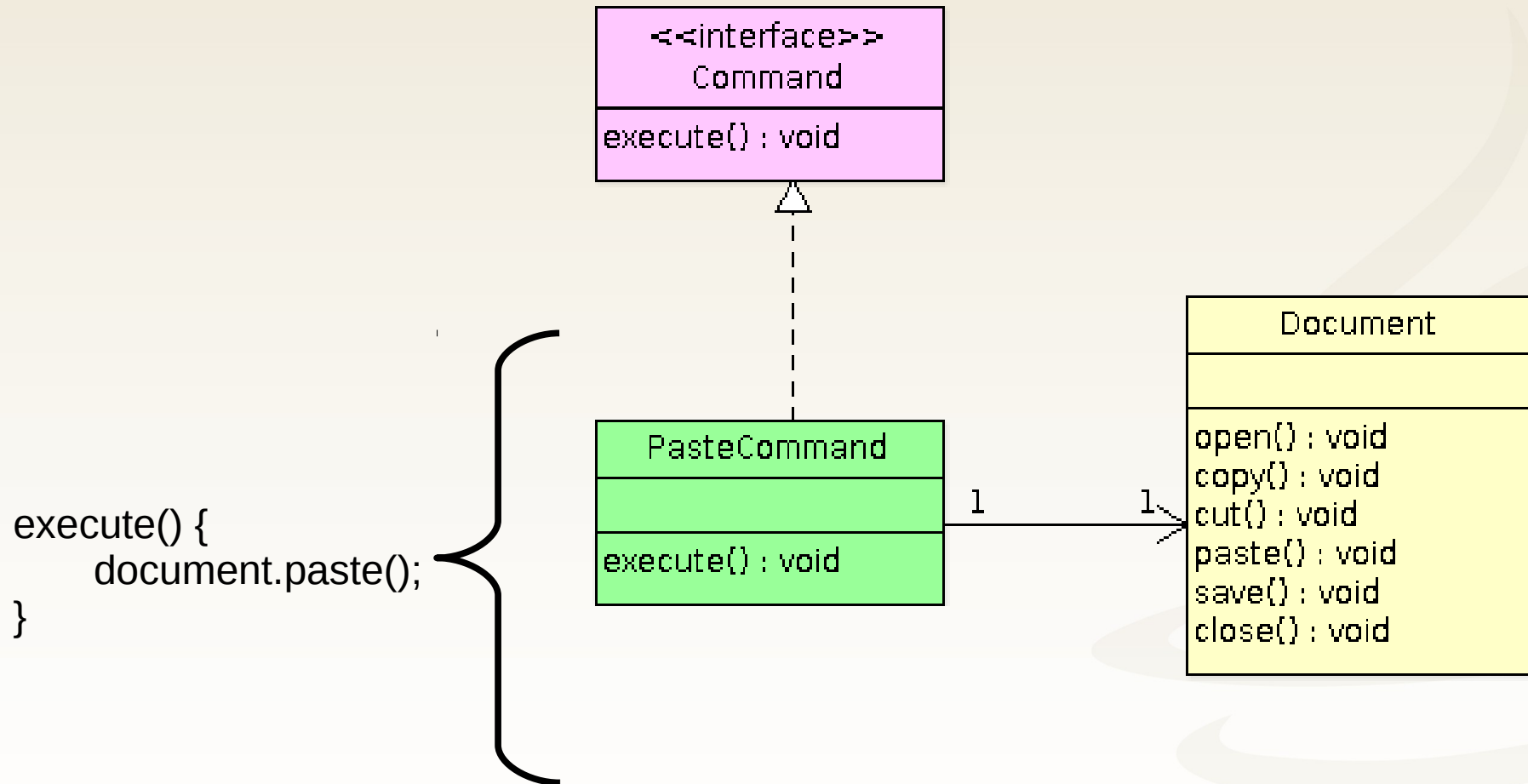
Metot Encapsulation

Metot invokasyonu da encapsule edilebilir...

Metot Encapsulation

- Bir request'in veya **metot çağırısının nesne olarak ifade edilmesidir**
- GUI bileşenleri kullanıcı inputu sonucu (örneğin bir button click) belirli bir metot çağırısı tetikler. Ancak **GUI framework** bu metot çağırısının tam olarak ne yapacağına karar veremez
- Dolayısı ile **metot çağırısının implemantasyonunu** uygulama tarafı yapmalıdır
- Uygulama metot çağırısını **bir nesne olarak** GUI framework'e iletir, GUI framework'de gereken yerden bu nesne üzerinden metot çağırısını işletir

Command Sınıf Diagramı



Java ve Command

- JDK'da Command örüntüsünün kullanıldığı yaygın bilinen iki örnek **javax.swing.Action** ve **java.lang.Runnable** arayüzleridir
- Action arayüzü **ActionListener**'dan türemiş olup birden fazla UI nesnesi üzerinden erişilebilecek kullanıcı komutlarını encapsule etmek için kullanılır
- Pek çok Swing bileşeni **action property**'sine sahiptir
- Bu property üzerinden UI bileşenlerine ilgili **komut nesneleri** tanımlanabilir

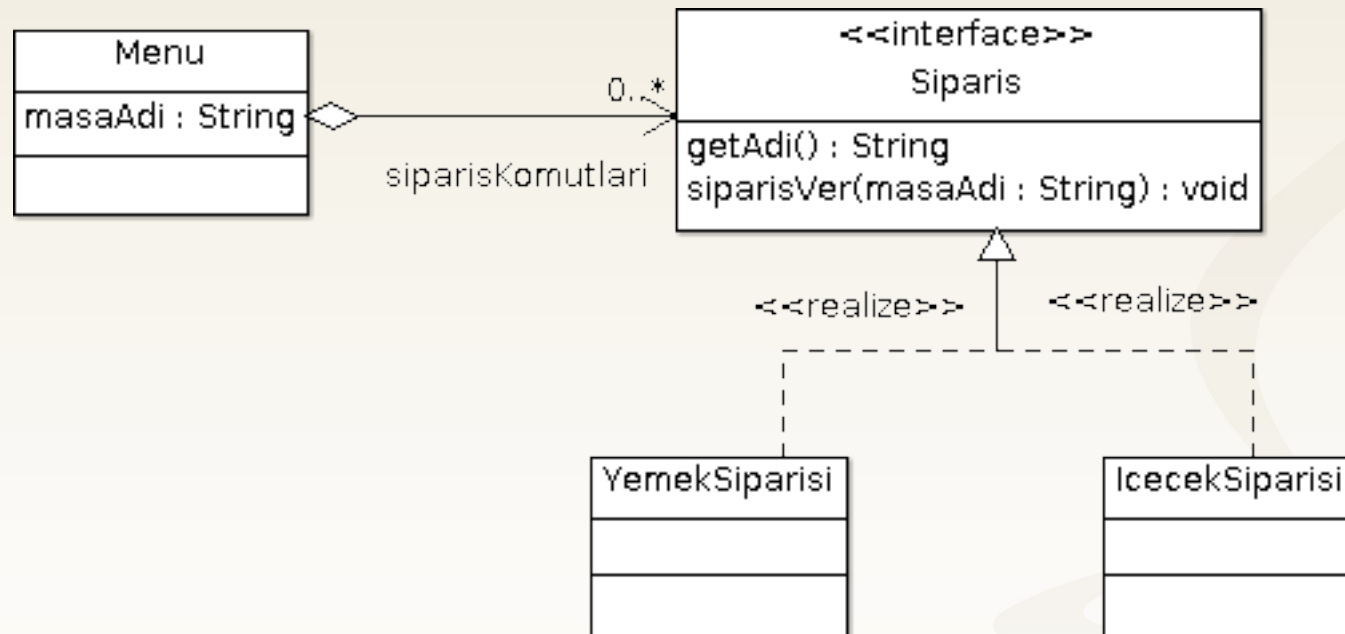
- **java.lang.Runnable** arayüzü ise ayrı bir Thread tarafından çalıştırılacak görevleri nesne olarak encapsule etmeyi sağlar

```
Runnable task = new Runnable() {  
  
    @Override  
    public void run() {  
        System.out.println("Hello world!");  
        try {  
            Thread.sleep(1000);  
        } catch (InterruptedException e) {}  
    }  
};  
  
Thread t = new Thread(task);  
t.start();
```

LAB ÇALIŞMASI: Command

- Kafeye gelen müşterilerin siparişlerini masalarından online olarak verebilmeleri istenmektedir
- Bu sistemde her bir masaya kurulacak olan elektronik bir menü içerisinde yiyecek ve içecek seçenekleri sıralanacak, herhangi bir sipariş doğrudan mutfağa iletilecektir
- Kafe sahibi elektronik menü içerisindeki yiyecek ve içecek listesini zaman zaman değiştirebilmelidir
- Hatta her bir masaya özel ayrı içerikte menüler de hazırlamak istemektedir

LAB ÇALIŞMASI: Command



Örüntüsünün Sonuçları

- Command örüntüsü sayesinde metot çağrıları **tekrar kullanılabilir, saklanabilir, bir yerden başka bir yere transfer edilebilir, loglanabilir veya undo yapılabilir**
- Tek tek command nesneleri bir araya getirilerek birden fazla işlemi gerçekleştiren **composite command nesneleri** oluşturulabilir

İletişim



www.harezmi.com.tr

www.java-egitimleri.com



info@harezmi.com.tr

info@java-egitimleri.com



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

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