

Strateji(Strategy) Tasarım Deseni

Nedir?

Strateji(Strategy) tasarım deseni, **behavioral** tasarım desenlerinden biridir. Bir dizi **algoritma tanımlar**, bu algoritmaların erişimi için arayüz sağlar ve algoritmaları tanımlarken **birbirlerinin yerine** kullanılabilir özellikte tasarlar. Ayrıca client runtime zamanında **dilediği** algoritmayı seçebilir.

Ne zaman Kullanılır?

Belirli bir iş için birden çok algoritmaya(yönteme) sahip olduğumuzda kullanılır.

Nasıl Kullanılır?

Öncelikle bir algoritma **interface**'i oluşturulur .Daha sonra oluşturulan **interface**'i implement edecek algoritma sınıfları yaratılır. **Context** sınıfı yaratılır. Bu sınıf içerisinde algoritma **interface** türünden algoritmaların set edilmesi için bir metod bulunur. Bu metod sayesinde **Context** sınıfı algoritma nesnesini tutmuş olur. Son olarak bir **Client** sınıfı oluşturulur. Bu sınıf **Context** sınıfından ve algoritma sınıflardan nesnelere üreterek işlemlerin yapılmasını sağlar

Faydaları Nedir?

Runtime zamanında **özel** bir işlem için tanımlanmış algoritmalarından dilediğimizi seçebiliriz.

Gerekenler

Türü **interface** olan **Strategy** sınıfı

En az iki tane **somut Strategy** sınıfı

Context sınıfı

Client sınıfı

Örnek Kullanım Alanları

1. `java.util.Comparator#compare()`
2. `javax.servlet.http.HttpServlet, service()` ve tüm `doXXX()` metodları
3. `javax.servlet.Filter#doFilter()`

Örnek Uygulama

```
1  /**
2   * Strateji interface
3   */
4  public interface Communicate {
5      public void send();
6  }
```

```
1  /**
2   * Somut algoritma sınıfı
3   */
4  public class Email implements Communicate {
5      @Override
6      public void send() {
7          System.out.println("Email is send");
8      }
9  }
```

```
1  /**
2   * Somut algoritma sınıfı
3   */
4  public class Sms implements Communicate {
5      @Override
6      public void send() {
7          System.out.println("Sms is send");
8      }
9  }
```

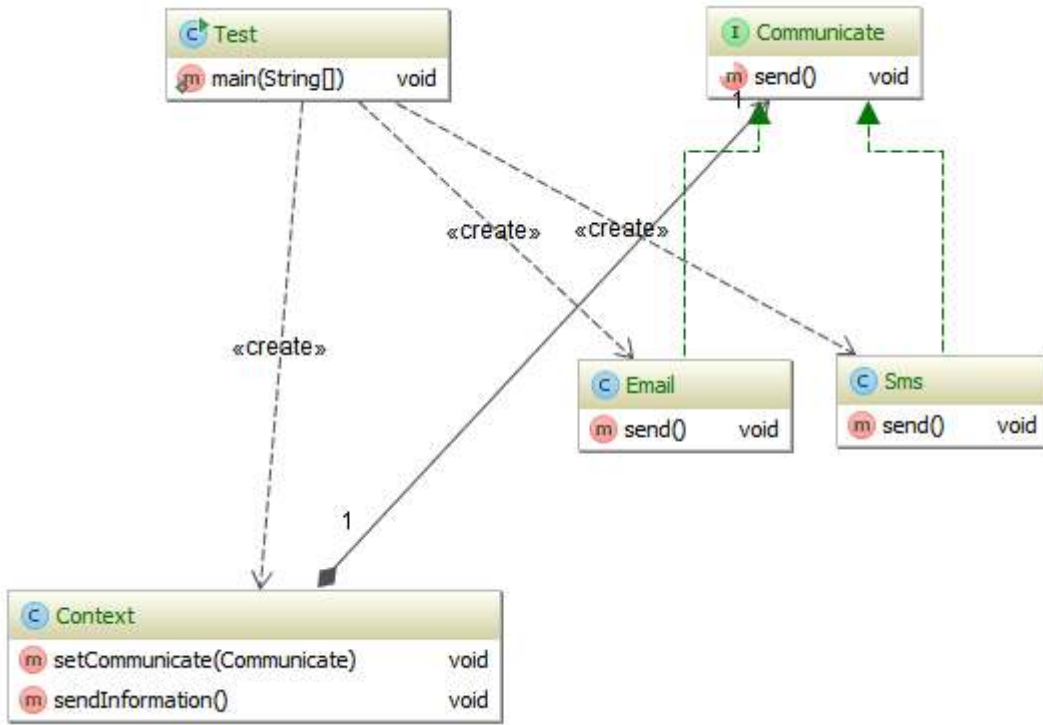
```
1  /**
2   * Context sınıfı
3   */
4  public class Context {
5      private Communicate communicate;
6
7      public void setCommunicate(Communicate communicate) {
8          this.communicate = communicate;
9      }
10     public void sendInformation(){
11         communicate.send();
12     }
13 }
```

```

1  /**
2  * Test sinifi
3  */
4  public class Test {
5      public static void main(String[] args) {
6          Context context=new Context();
7          context.setCommunicate(new Email());
8          context.sendInformation();
9          context.setCommunicate(new Sms());
10         context.sendInformation();
11     }
12 }

```

Uygulamamızın UML Diagramı



Strateji Tasarım Deseni'nin Şematik Gösterimi

