NESNE YÖNELİMLİ PROGRAMLAMA 2(Object Oriented Programming 2/OOP)

Öğr. Gör. Celil ÖZTÜRK
Marmara Üniversitesi
Teknik Bilimler Meslek Yüksekokulu

İçerik

- Behavioral Patterns(Davranışsal Kalıplar)
- Interpreter Pattern
- Mediator Pattern
- Memento Pattern
- Visitor Pattern

Tasarım Kalıpları

Behavioral Patterns(Davranışsal Kalıplar)

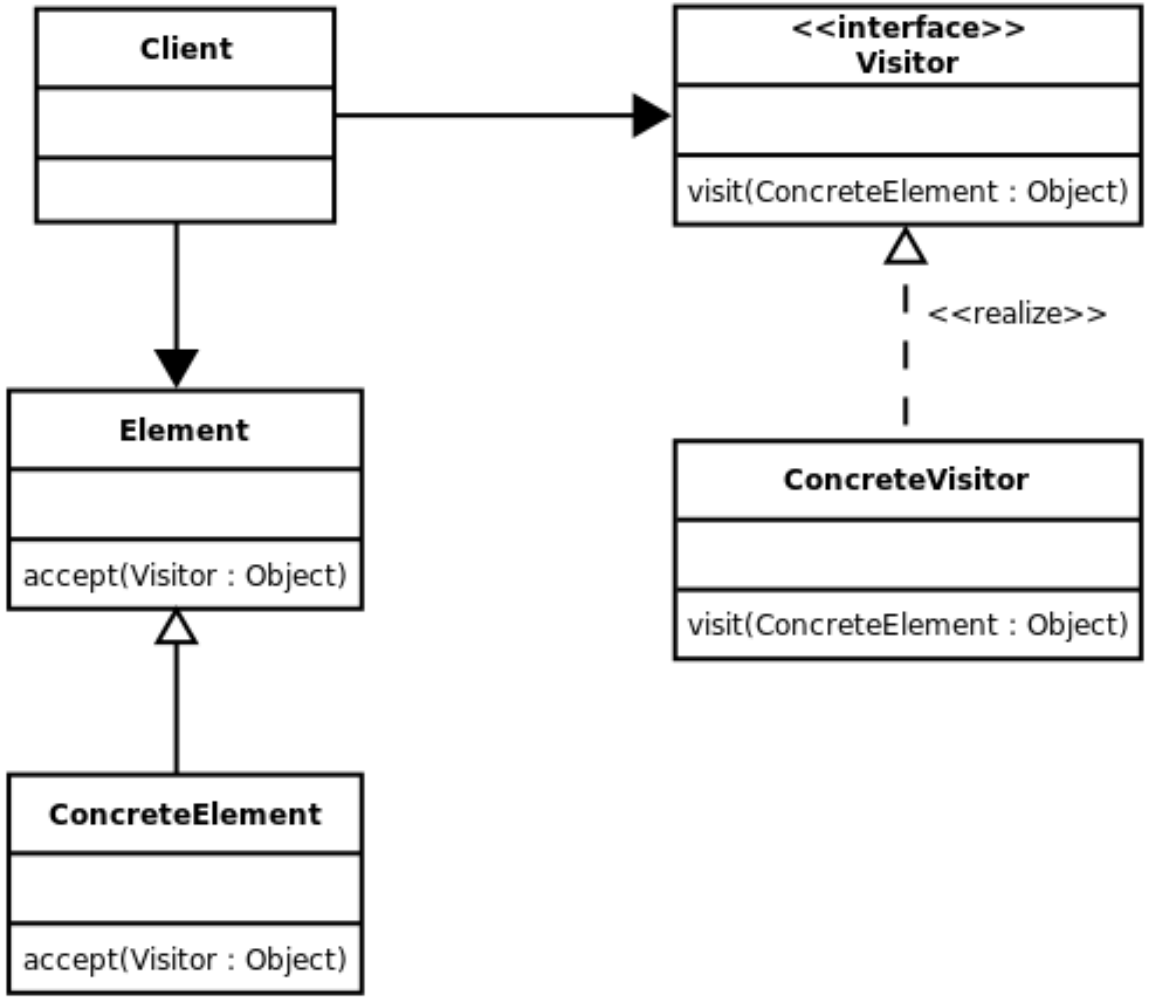
- Chain of Responsibility Pattern
- Command Pattern
- Interpreter Pattern
- Iterator Pattern
- Mediator Pattern
- Memento Pattern
- Observer Pattern
- Null Object Pattern
- Strategy Pattern
- State Pattern
- Visitor Pattern

Visitor(Ziyaretçi) Tasarım Kalıbı

- Ziyaretçi tasarım kalıbı, bir sınıf hiyerarşisinde yer alan sınıflar üzerinde değişiklik yapmadan, bu sınıflara yeni metotların eklenmesini kolaylaştırır. (bidb.itu.edu.tr)
- İstenilen metot bir ziyaretçi sınıfında tanımlanabilir.
- Bu tasarım deseni, çok sayıda ve farklı tipteki nesneler üzerinde işlem yapabilmek amacıyla kullanılır.
- Sisteme yeni nesneler eklenmiyor, fakat sık sık yeni işlemlerin eklenmesi gerekiyorsa bu tasarım deseni kullanılabilir.

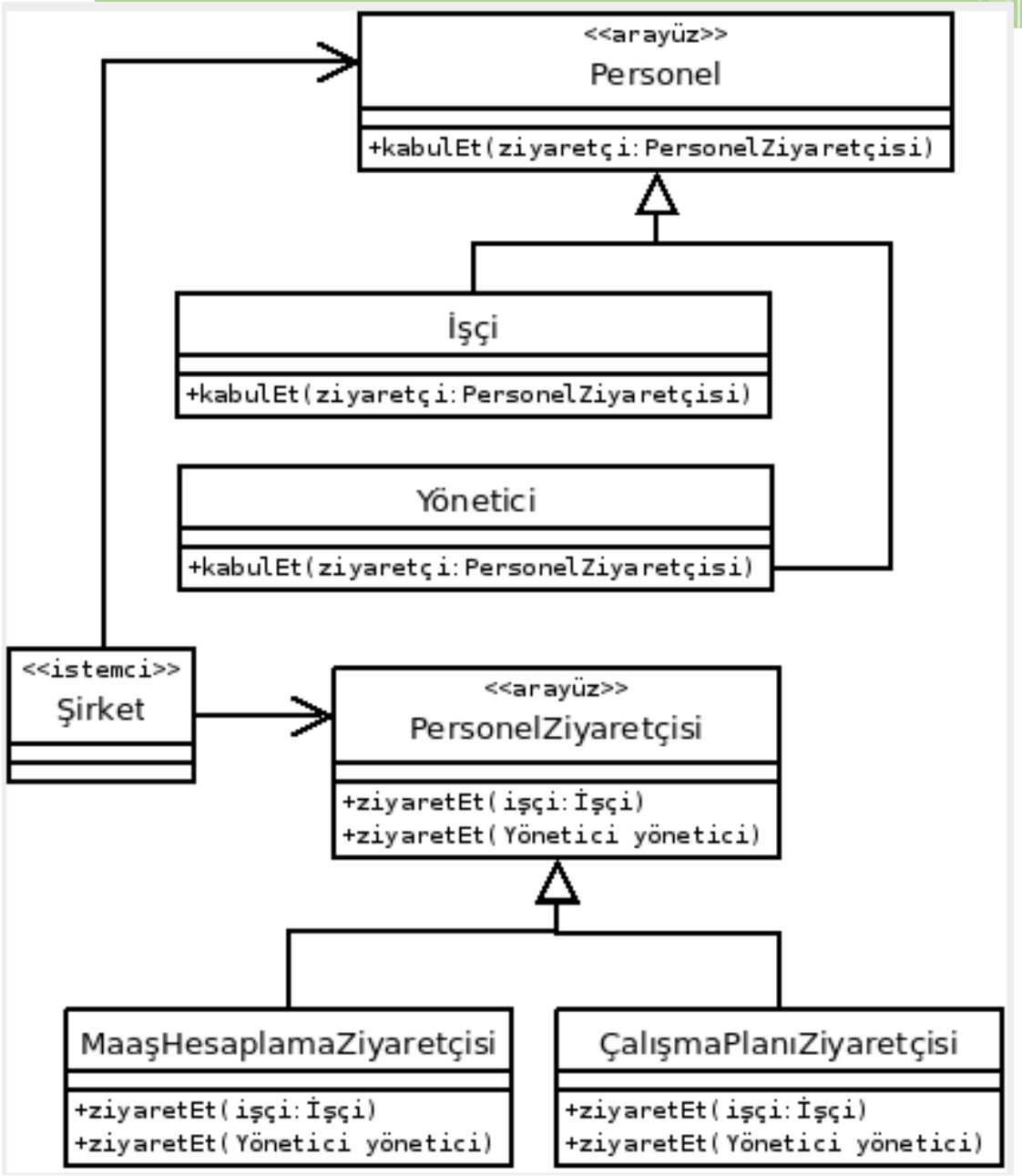
Visitor(Ziyaretçi) Tasarım I

- Visitor tasarımkalıbı sınıflara sonradan modüler olarak metodlar eklememizi sağlar.
- Sınıfın olmayan özelliğini sonradan eklemeyi kolaylaştırır.



https://kenanatmaca.com/visitor-pattern/

Visitor(Ziyaretçi) Tasarım Kalıbı



http://www.tasarimdesenleri.com/jsp/tasdesincele/visitor.jsp

Visitor(Ziyaretçi) Tasarım Kalıbı

Personel.java

```
public interface Personel {

  void kabulEt( PersonelZiyaretçisi ziyaretçi );
}
```

İsçi.java

```
public class İşçi implements Personel {

  @Override
  public void kabulEt( final PersonelZiyaretçisi ziyaretçi ) {
    ziyaretçi.ziyaretEt( this );
  }
}
```

http://www.tasarimdesenleri.com/jsp/tasdesincele/visitor.jsp

Visitor(Ziyaretçi) Tasarım Kalıbı

Yonetici.java

```
public class Yönetici implements Personel {

  @Override
  public void kabulEt( final PersonelZiyaretçisi ziyaretçi ) {
    ziyaretçi.ziyaretEt( this );
  }
}
```

http://www.tasarimdesenleri.com/jsp/tasdesincele/visitor.jsp

Visitor(Ziyaretçi) Tasarım Kalıbı

MaaşHesaplamaZiyaretcisi.java

```
public class MaaşHesaplamaZiyaretçisi implements PersonelZiyaretçisi {

  @Override
  public void ziyaretEt( final İşçi işçi ) {
    System.out.println( "İşçi maaşı hesaplandı." );
  }

  @Override
  public void ziyaretEt( final Yönetici yönetici ) {
    System.out.println( "Yönetici maaşı hesaplandı." );
  }
}
```

PersonelZiyaretcisi.java

```
public interface PersonelZiyaretçisi {

  void ziyaretEt( İşçi işçi );

  void ziyaretEt( Yönetici yönetici );
}
```

http://www.tasarimdesenleri.com/jsp/tasdesincele/visitor.jsp

Visitor(Ziyaretçi) Tasarım Kalıbı

CalışmaPlanıZiyaretcisi.java

```
public class ÇalışmaPlanıZiyaretçisi implements PersonelZiyaretçisi {

  @Override
  public void ziyaretEt( final İşçi işçi ) {
    System.out.println( "İşçi çalışma saatleri hesaplandı." );
  }

  @Override
  public void ziyaretEt( final Yönetici yönetici ) {
    System.out.println( "Yönetici çalışma saatleri hesaplandı." );
  }
}
```

PersonelZiyaretcisi.java

```
public interface PersonelZiyaretçisi {

  void ziyaretEt( İşçi işçi );

  void ziyaretEt( Yönetici yönetici );
}
```

http://www.tasarimdesenleri.com/jsp/tasdesincele/visitor.jsp

Visitor(Ziyaretçi) Tasarım Kalıbı

Main.java

```
public class Şirket {

   public static void main( final String[] args ) {

      // Personel nesnelerini oluştur.
      final İşçi işçi = new İşçi();
      final Yönetici yönetici = new Yönetici();

      // Ziyaretçi nesnelerini oluştur.
      final PersonelZiyaretçisi maaşHesaplamaZiyaretçisi = new MaaşHesaplamaZiyaretçisi();
      final PersonelZiyaretçisi çalışmaPlanıZiyaretçisi = new ÇalışmaPlanıZiyaretçisi();

      // Maaş Hesaplama işlemlerini yap.
      işçi.kabulEt( maaşHesaplamaZiyaretçisi );
      yönetici.kabulEt( maaşHesaplamaZiyaretçisi );

      // Çalışma Planı işlemlerini yap.
      işçi.kabulEt( çalışmaPlanıZiyaretçisi );
      yönetici.kabulEt( çalışmaPlanıZiyaretçisi );
   }
}
```

Ekran Çıktısı

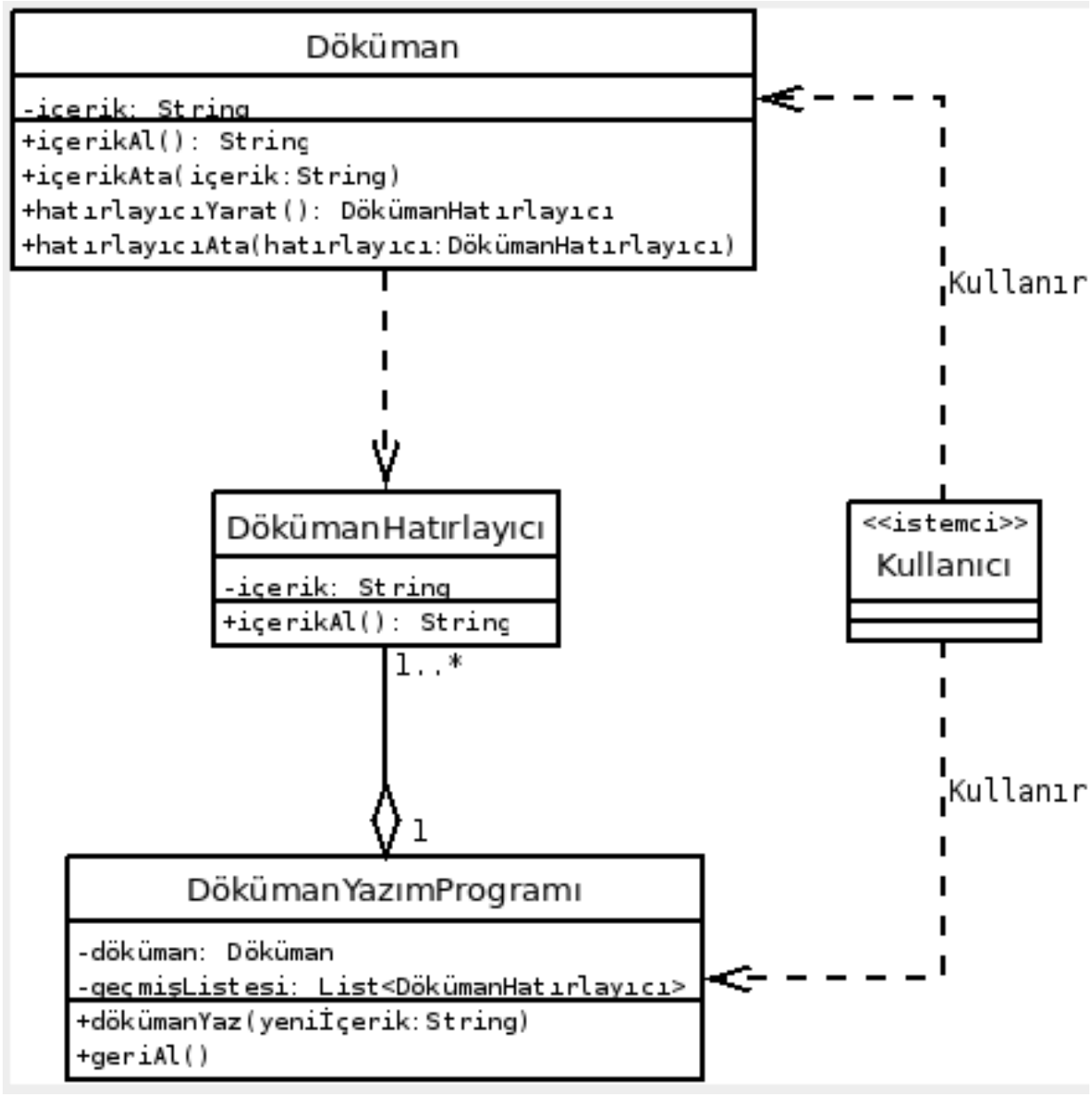
```
İşçi maaşı hesaplandı.
Yönetici maaşı hesaplandı.
İşçi çalışma saatleri hesaplandı.
Yönetici çalışma saatleri hesaplandı.
```

http://www.tasarimdesenleri.com/jsp/tasdesincele/visitor.jsp

Memory(Hatıra)/Memento Tasarım Kalıbı

- Bir nesnenin, daha önce sahip olduğu durumlardan birine tekrar dönüştürülebilmesi için hatıra tasarım kalıbı kullanılır. Bu sayede belirli durumlardan sonra nesnelerin aldıkları çeşitli durumların kolaylıkla değiştirilebilmesi ve bu durumların farklı sınıflarda tanımlanarak kullanılabilirliğinin arttırılması amaçlanmaktadır.(bidb.itu.edu.tr)

Memory(Hatıra)/Memento Tasarım Kalıbı



http://www.tasarimdesenleri.com/jsp/tasdesincele/memento.jsp

Memory(Hatıra)/Memento Tasarım Kalıbı

- Dokuman.java

```
public class Döküman {

  private String  içerik;

  public String içerikAl() {
    return içerik;
  }

  public void içerikAta( final String içerik ) {
    this.içerik = içerik;
  }

  public DökümanHatırlayıcı hatırlayıcıYarat() {
    return new DökümanHatırlayıcı( this );
  }

  public void hatırlayıcıAta( final DökümanHatırlayıcı hatırlayıcı ) {
    this.içerik = hatırlayıcı.içerikAl();
  }
}
```

http://www.tasarimdesenleri.com/jsp/tasdesincele/memento.jsp

Memory(Hatıra)/Memento Tasarım Kalıbı

Dokumanhatırlayıcı.java

```
public class DökümanHatırlayıcı {

  private final String  içerik;

  public DökümanHatırlayıcı( final Döküman döküman ) {
    this.içerik = döküman.içerikAl();
  }

  public String içerikAl() {
    return içerik;
  }
}
```

http://www.tasarimdesenleri.com/jsp/tasdesincele/memento.jsp

Memory(Hatıra)/Memento Tasarım Kalıbı

DokumanYazımProgramı.java

```
public class DökümanYazımProgramı {

  private final Döküman            döküman;
  private final List< DökümanHatırlayıcı >  geçmişListesi  = new ArrayList< DökümanHatırlayıcı >();

  public DökümanYazımProgramı( final Döküman döküman ) {
    this.döküman = döküman;
  }

  public void dökümanYaz( final String yeniİçerik ) {
    final DökümanHatırlayıcı hatırlayıcı = döküman.hatırlayıcıYarat();
    geçmişListesi.add( hatırlayıcı );

    döküman.içerikAta( yeniİçerik );
  }

  public void geriAl() {
    final DökümanHatırlayıcı hatırlayıcı = geçmişListesi.remove( geçmişListesi.size() - 1 );
    döküman.hatırlayıcıAta( hatırlayıcı );
  }
}
```

http://www.tasarimdesenleri.com/jsp/tasdesincele/memento.jsp

Memory(Hatıra)/Memento Tasarım Kalıbı

Kullanıcı.java

```java
public class Kullanıcı {

  public static void main( final String[] args ) {
    final Döküman döküman = new Döküman();
    döküman.içerikAta( "" );

    final DökümanYazımProgramı program = new DökümanYazımProgramı( döküman );
    program.dökümanYaz( "1" );
    program.dökümanYaz( "2" );
    program.dökümanYaz( "3" );
    program.dökümanYaz( "4" );

    // En son içerik "4" yazılır.
    // Fakat önceki içerikler saklanır.
    System.out.println( "En son içerik : " + döküman.içerikAl() );

    // Son üç işlem geri alınır.
    program.geriAl();
    program.geriAl();
    program.geriAl();

    // Son üç işlem geri alındığı için, "1" içeriği yazılır.
    System.out.println( "Üç işlem geri alındıktan sonra içerik : " + döküman.içerikAl() );
  }
}
```

Ekran Çıktısı

```
En son içerik :4
Üç işlem geri alındıktan sonra içerik :1
```

http://www.tasarimdesenleri.com/jsp/tasdesincele/memento.jsp

Mediator(Aracı) Tasarım Kalıbı

- Aracı tasarım kalıbı nesnelerin yönetimi ve aralarındaki iletişimin merkezi bir noktadan sağlanması ve yönetilmesi için kullanılmaktadır. Bu nesneler arasındaki bağı azaltmakta ve sadece bir sınıfı, yönetici sınıf olarak diğer sınıfların koordine edilmesinden sorumlu kılar.(bidb.itu.edu.tr)

Mediator(Aracı) Tasarım Kalıbı

- Kalıp genellikle yazılan programın sınıf sayısı kadar kullanılır.
- Mantık (logic) ve hesaplama (compotation) bu sınıflar arasında dağıtılır. Ayrıca çok gelişmiş programlarda sınıflar arasında iletişim gittikçe daha da karmaşık bir hal alır ve kodun ilerlemesi ile okunaklılığı da bundan etkilenir.
- Aracı kalıbı kullanıldığında ise iletişim sınıflar arasında doğrudan sağlanmaz. Sınıflar arasında aracı kalıbı bulunur ve sınıfları yönetir.

https://bidb.itu.edu.tr/seyir-defteri/blog/2013/09/08/arac%C4%B1-kal%C4%B1b%C4%B1-(mediator-pattern)

Mediator(Aracı) Tasarım Kalıbı

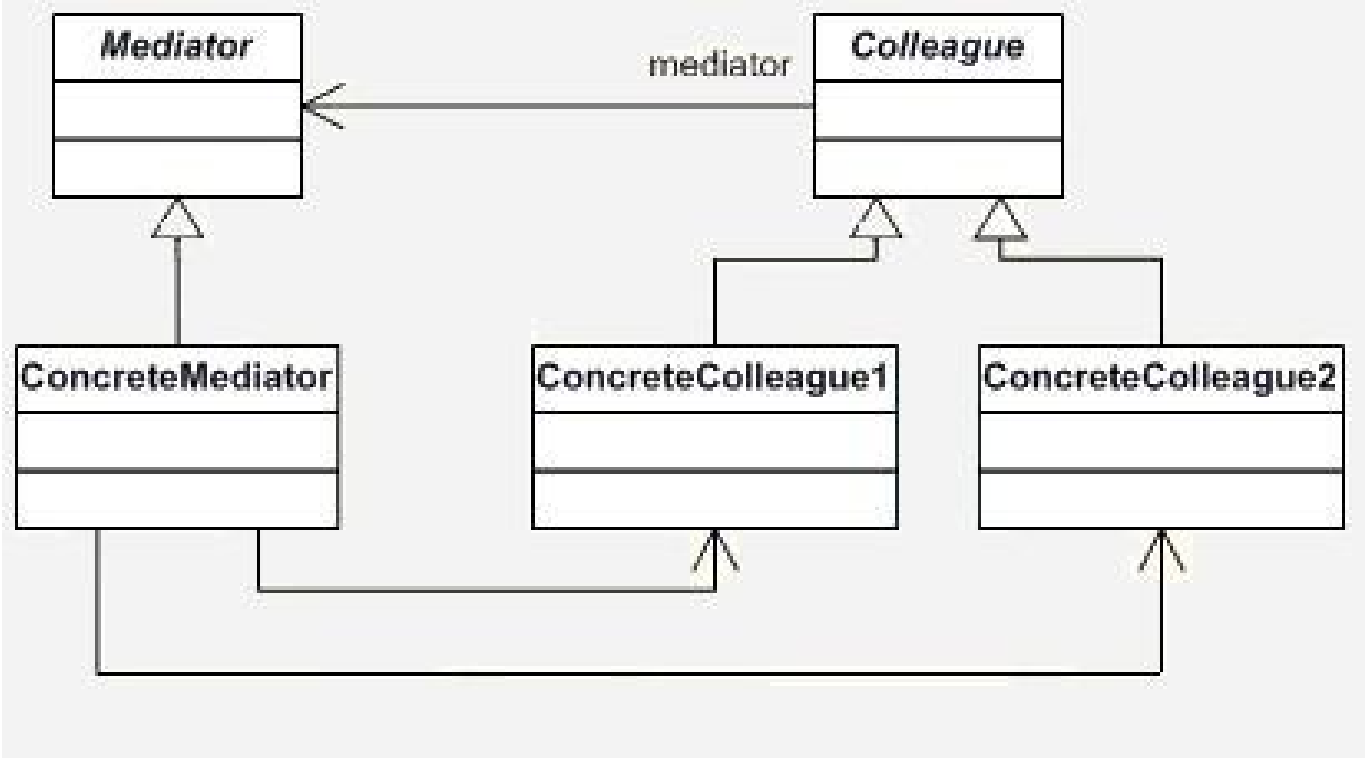
Kullanım Alanları

- Anlaşılması zor ve karmaşık şekilde yapılandırılmış nesneler için kullanılır.
- Gelişmiş nesnel yapılarda bir nesnede yapılan değişikliğin diğer nesneleri etkilemesini engellemek için kullanılır.
- Yeni sınıflandırmalara gerek kalmadan nesneler arasında dağıtımı sağlamak için kullanılır.

https://bidb.itu.edu.tr/seyir-defteri/blog/2013/09/08/arac%C4%B1-kal%C4%B1b%C4%B1-(mediator-pattern)

Mediator(Aracı) Tasarım Kalıbı

- UML Diagramı



https://bidb.itu.edu.tr/seyir-defteri/blog/2013/09/08/arac%C4%B1-kal%C4%B1b%C4%B1-(mediator-pattern)

Mediator(Aracı) Tasarım Kalıbı

- Örnekte, mediator(aracı) türünden bir de yönetici sınıf tanımlanır. Böylece mediator(aracı) kalıbı diğer sınıflar arasında yönetici durumuna gelerek diğer sınıflardaki katılımcıların koordine edilmesini sağlar.

https://bidb.itu.edu.tr/seyir-defteri/blog/2013/09/08/arac%C4%B1-kal%C4%B1b%C4%B1-(mediator-pattern)

Mediator(Aracı) Tasarım Kalıbı

```
abstract class Araci
{
   public abstract void Gonder(string mesaj,Katilimci
                               katilimci);
}
```

```
class SomutAraci : Araci
{
   private SomutKatilimci1 _katilimci1;
   private SomutKatilimci2 _katilimci2;
   public SomutKatilimci1 Katilimci1
   {
      set { _katilimci1 = value; }
   }
   public SomutKatilimci2 Katilimci2
   {
      set { _katilimci2 = value; }
   }
   public override void Gonder(string mesaj, Katilimci
                               katilimci)
   {
      if (katilimci == _katilimci1)
      {
         _katilimci2.Al(mesaj);
      }
      else
      {
         _katilimci1.Al(mesaj);
      }
   }
}
```

https://bidb.itu.edu.tr/seyir-defteri/blog/2013/09/08/arac%C4%B1-kal%C4%B1b%C4%B1-(mediator-pattern)

Mediator(Aracı) Tasarım Kalıbı

```
abstract class Katilimci
{
    protected Araci araci;

    public Katilimci(Araci araci)
    {
        this.araci = araci;
    }
}
```

```
class SomutKatilimci1 : Katilimci
{
    public SomutKatilimci1(Araci araci)
        : base(araci)
    {
    }
    public void Gonder(string mesaj)
    {
        araci.Gonder(mesaj, this);
    }
    public void Al(string mesaj)
    {
        Console.WriteLine("Katılımcı 1 tarafından alınan
                    mesaj: " + mesaj);
    }
}
```

```
class SomutKatilimci2 : Katilimci
{
    public SomutKatilimci2(Araci araci)
        : base(araci)
    {
    }
    public void Gonder(string mesaj)
    {
        araci.Gonder(mesaj, this);
    }
    public void Al(string mesaj)
    {
        Console.WriteLine("Katılımcı 2 tarafından alınan
                    mesaj: " + mesaj);
    }
}
}
```

https://bidb.itu.edu.tr/seyir-defteri/blog/2013/09/08/arac%C4%B1-kal%C4%B1b%C4%B1-(mediator-pattern)

Mediator(Aracı) Tasarım Kalıbı

```
class MainApp
{
    static void Main()
    {
        SomutAraci sAraci = new SomutAraci();
        SomutKatilimci1 sKatilimci1 =
                new SomutKatilimci1(sAraci);
        SomutKatilimci2 sKatilimci2 =
                new SomutKatilimci2(sAraci);
        sAraci.Katilimci1 = sKatilimci1;
        sAraci.Katilimci2 = sKatilimci2;
        sKatilimci1.Gonder("Test Verisi 1");
        sKatilimci2.Gonder("Test Verisi 2");
        Console.ReadKey();
    }
}
```

Ekran Çıktısı

```
Katılımcı 2 tarafından alınan mesaj: Test Verisi 1
Katılımcı 1 tarafından alınan mesaj: Test Verisi 2
kPress any key to continue . . .
```

https://bidb.itu.edu.tr/seyir-defteri/blog/2013/09/08/arac%C4%B1-kal%C4%B1b%C4%B1-(mediator-pattern)

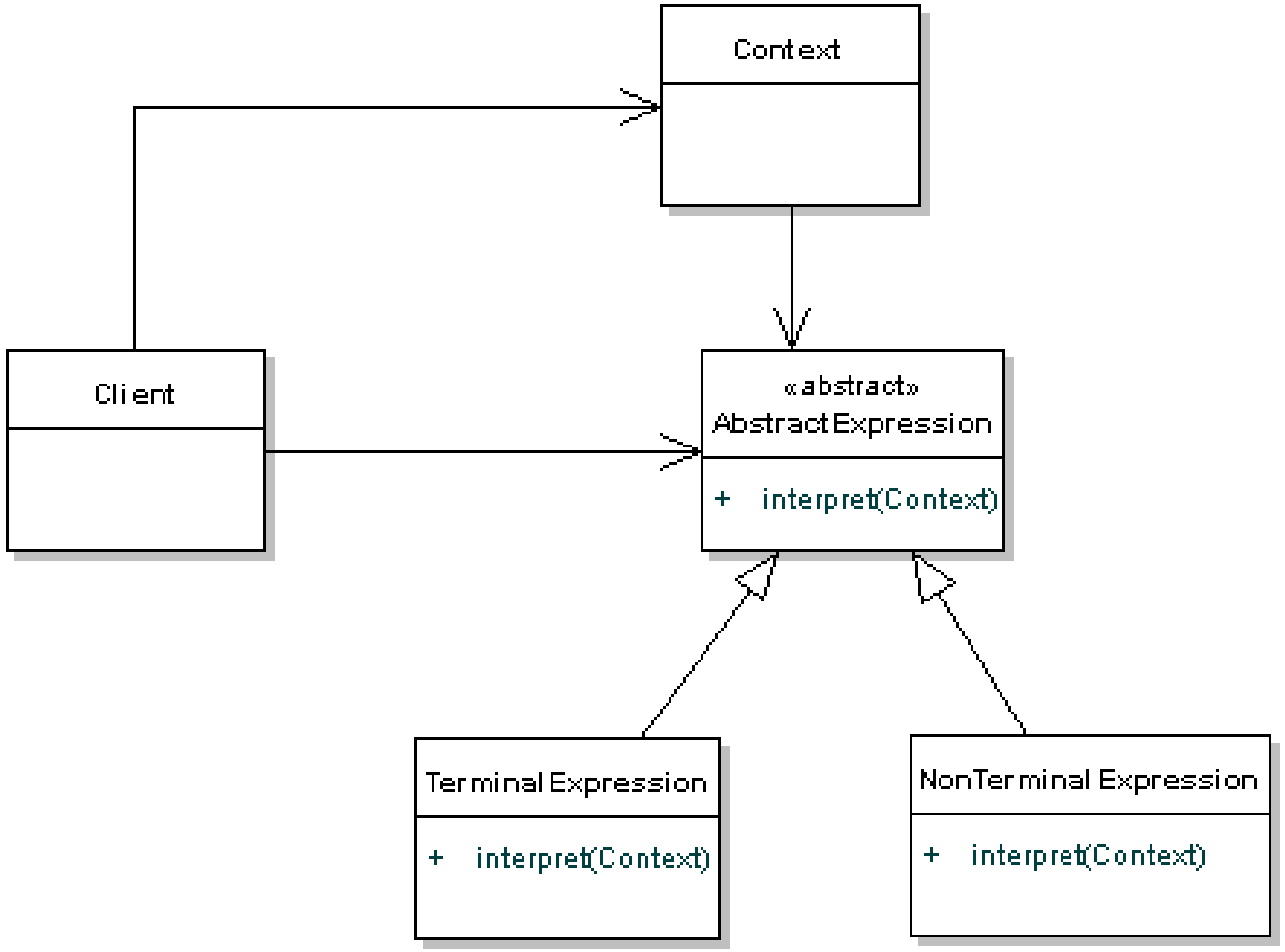
Interpreter(Yorumlayıcı) Tasarım Kalıbı

Sıklıkla karşılaşılan belli mantıksal kalıpların bir bütün içerisinde yer almasını sağlamak amacıyla kullanılmaktadır.

Dil bilgisi kuralları gibi kalıplar içerisinde yer alan ifadelerin yorumlanması amacıyla kullanılması tercih edilmektedir.(bidb.itu.edu.tr)

Interpreter(Yorumlayıcı) Tasarım Kalıbı

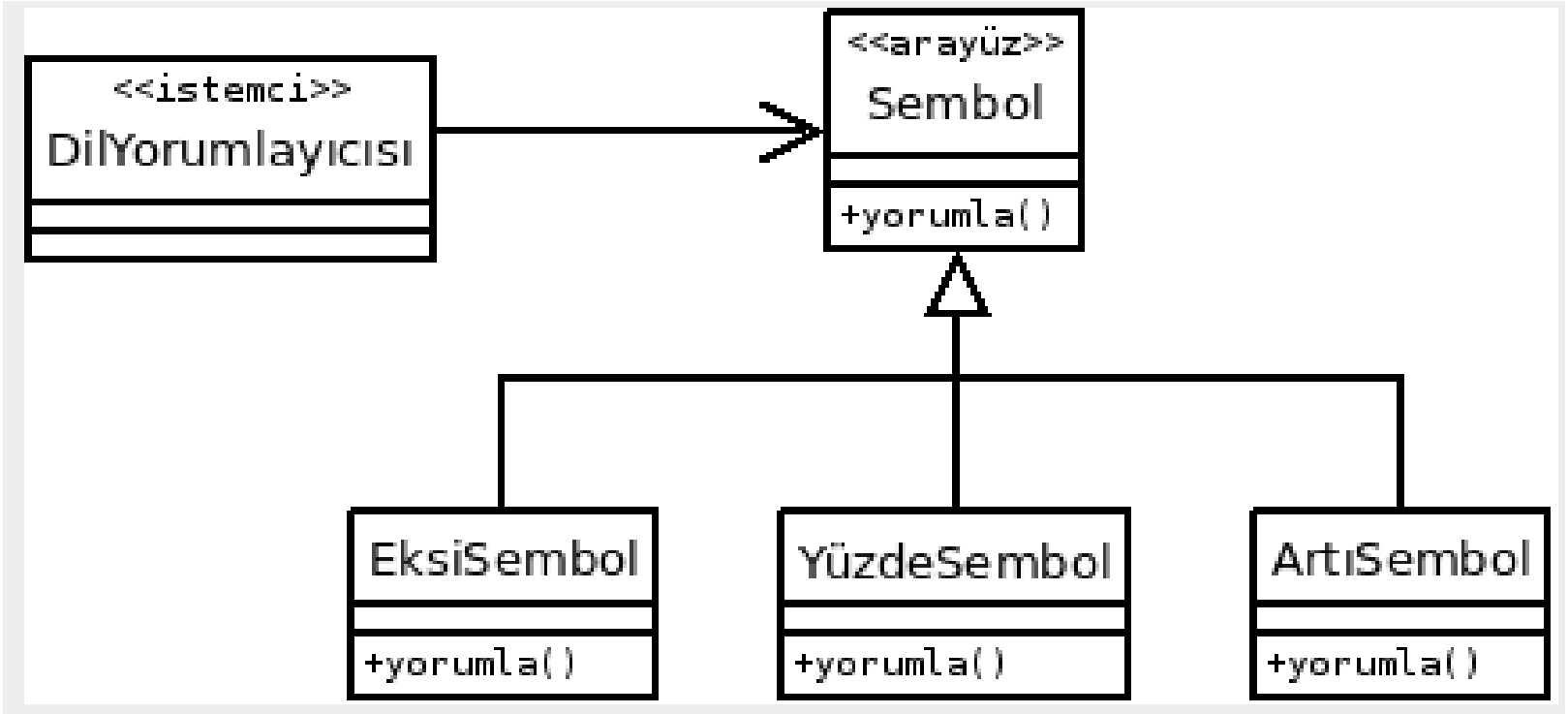
Interpreter (Yorumlayıcı) tasarım deseni **String** üzerinde yapılan yorumlama ve çevirme işlemleri için tasarlanmıştır.



https://kenanatmaca.com/interpreter-pattern/

Interpreter(Yorumlayıcı) Tasarım Kalıbı Örnek

Örnekte, '+' (artı), '-' (eksi) ve '%' (yüzde) karakterlerinden oluşan bir bilgisayar dilinin yorumlanması gösterilmiştir.



http://www.tasarimdesenleri.com/jsp/tasdesincele/interpreter.jsp

Interpreter(Yorumlayıcı) Tasarım Kalıbı Örnek

```
public interface Sembol {

  void yorumla();
}
```

```
public class ArtıSembol implements Sembol {

  @Override
  public void yorumla() {
    System.out.println( "+ sembolü yorumlandı." );
  }
}
```

```
public class YüzdeSembol implements Sembol {

  @Override
  public void yorumla() {
    System.out.println( "% sembolü yorumlandı." );
  }
}
```

```
public class EksiSembol implements Sembol {

  @Override
  public void yorumla() {
    System.out.println( "- sembolü yorumlandı." );
  }
}
```

http://www.tasarimdesenleri.com/jsp/tasdesincele/interpreter.jsp

Interpreter(Yorumlayıcı) Tasarım Kalıbı Örnek

Ekran Çıktısı

```
+ sembolü yorumlandı.
+ sembolü yorumlandı.
- sembolü yorumlandı.
% sembolü yorumlandı.
% sembolü yorumlandı.
+ sembolü yorumlandı.
- sembolü yorumlandı.
```

```java
public class DilYorumlayıcısı {

  public static void main( final String[] args ) {
    // Yorumlanacak dil.
    final String metin = "++-%%+-";
    final List< Sembol > sembolListesi = new ArrayList< Sembol >();

    // Metni ayrıştır ve sembol sınıflarını
    // listeye ekle.
    for ( final char c : metin.toCharArray() ) {
      if ( c == '+' ) {
        sembolListesi.add( new ArtıSembol() );
      } else if ( c == '-' ) {
        sembolListesi.add( new EksiSembol() );
      } else if ( c == '%' ) {
        sembolListesi.add( new YüzdeSembol() );
      }
    }

    // Yorumla.
    for ( final Sembol s : sembolListesi ) {
      s.yorumla();
    }
  }
}
```

http://www.tasarimdesenleri.com/jsp/tasdesincele/interpreter.jsp

Kaynaklar

- https://bidb.itu.edu.tr/seyir-defteri/blog/2013/09/08/davran%C4%B1%C5%9Fsal-kal%C4%B1plar-(behavioral-patterns)
- http://www.tasarimdesenleri.com/jsp/tasdesincele/visitor.jsp
- https://kenanatmaca.com/visitor-pattern/
- https://bidb.itu.edu.tr/seyir-defteri/blog/2013/09/08/davran%C4%B1%C5%9Fsal-kal%C4%B1plar-(behavioral-patterns)
- http://www.tasarimdesenleri.com/jsp/tasdesincele/memento.jsp
- https://bidb.itu.edu.tr/seyir-defteri/blog/2013/09/08/arac%C4%B1-kal%C4%B1b%C4%B1-(mediator-pattern)
- https://kenanatmaca.com/interpreter-pattern/