Структурни шаблони за проектиране

(Structural Design Patterns)

Учителски екип

Обучение за ИТ кариера

https://it-kariera.mon.bg/e-learning/

Съдържание

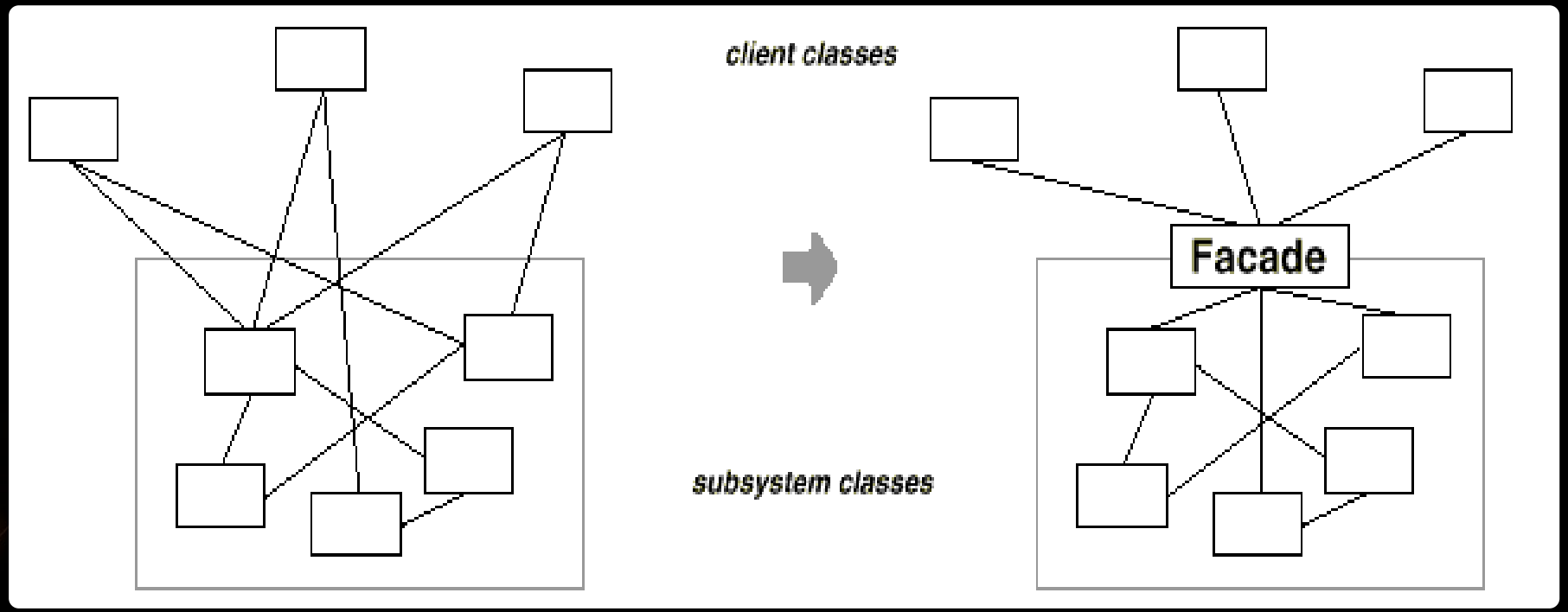
1. Шаблони за дизайн в структурата на приложението
   - Шаблон Façade
   - Шаблон Composite
   - Шаблон Decorator
   - Шаблон Adapter

Шаблони в структурата

- Шаблоните в структурата описват начини за групиране на обекти за реализиране на нова функционалност
  - Или как класове и обекти се групират в по-големи структури
  - Структурните шаблони на класове използват наследяване за съставяне на интерфейси или имплементации
  - Структурните шаблони на обекти съставят обекти за новата функционалност
- Примери за шаблони в проектирането на структурата:
  - Composite, Decorator, Façade, Adapter, Bridge, Proxy

Шаблон Façade

- Façade осигурява опростен интерфейс към по-голям програмен код
  - Интерфейс от по-висок ред скрива сложността на подсистемите



- Подобен шаблон: Adapter – преобразувател на интерфейси

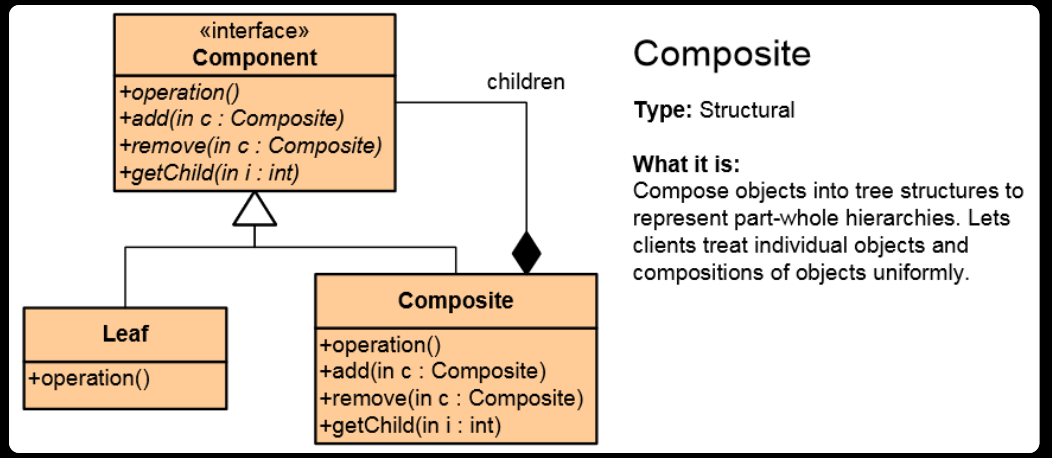
Façade – пример

```
interface IAESFacade
{
  string AESEncrypt(string message, string password);
  byte[] AESEncrypt(byte[] bytesToBeEncrypted, string password);
  byte[] AESDecrypt(byte[] bytesToBeDecrypted, string password);
  string AESDecrypt(string encryptedMessage, string password);
}
```

```
class AESFacade : IAESFacade
{
  public string AESEncrypt(string message, string password) { … }
  public byte[] AESEncrypt(byte[] bytes, string password) { … }
  public byte[] AESDecrypt(byte[] bytes, string password) { … }
  public string AESDecrypt(string msg, string password) { … }
}
```

Шаблон Composite

- Шаблонът Composite позволява групирането на различни типове обекти в дървовидни структури
  - Третира по един и същ начин отделните обекти и групите от обекти
- Пример:
  - Система за документооборот
- Използва се когато
  - Имате различни обекти и искате да ги третирате еднакво
  - Искате да представите йерархия от обекти



Composite – пример

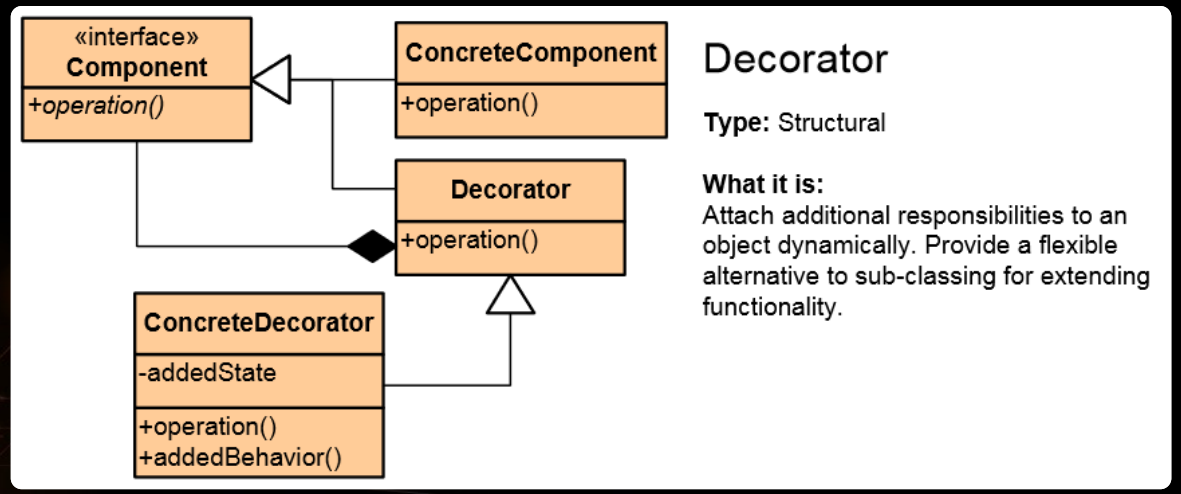
```
public interface IComponent { … }

public interface ICompositeComponent : IComponent
{
  void Add(Component page);
  void Remove(Component page);
}

public class Commander : ICompositeComponent
{
  private ICollection<Component> childComponents =
    new List<Component>();

  public override void Add(Component component)
  { this.childComponents.Add(component); }

  public override void Remove(Component component)
  { this.childComponents.Remove(component); }
}
```

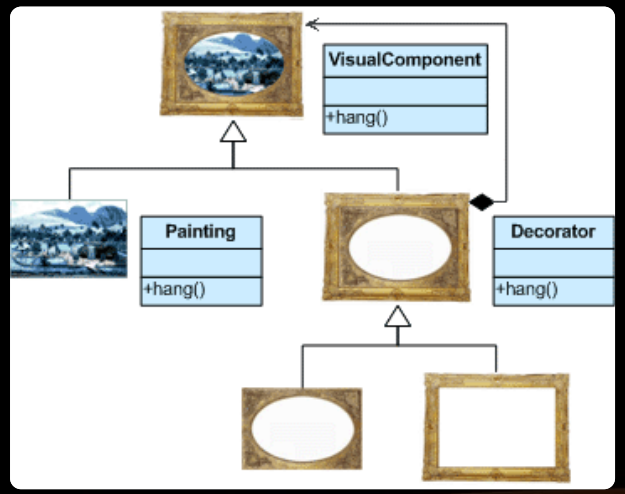
Composite – примери за реална употреба

- Контролите от Windows Forms
  - Класът `System.Windows.Forms.Control` има дъщерни контроли
  - Свойства `Controls`, `HasChildren`, …
- Контролите в ASP.NET Web Forms
  - Класът в `System.Web.UI.Control` пак е с дъщерни контроли
  - Свойство `Controls`
- Контролите в AWT / Java Swing
  - Класовете `java.awt.Component`, `java.awt.Container`

Шаблон Decorator

- Decorator добавя динамично нови отговорности на обектите
  - Обвива оригиналния компонент
  - Алтернатива е на наследяването (class explosion)
  - Поддържа Open-Closed принципа





Decorator – пример

```
public abstract class Pizza
{
    public abstract string GetDescription();
    public abstract decimal GetPrice();
}

public class TomatoSaucePizza : Pizza
{
  private Pizza basePizza;

  public TomatoSaucePizza(Pizza pizza)
  {  this.basePizza = pizza; }

  public override string GetDescription()
  { return this.basePizza.GetDescription() + " + Tomato Sauce"; }

  public override decimal GetPrice()
  { return basePizza.GetPrice() + 0.60m; }
}
```

Decorator – примери за реална употреба

- **BufferedStream** в .NET декорира **Stream**
  - **CryptoStream** декорира **Stream**

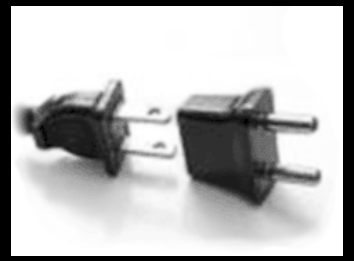
```
CryptoStream crStream = new CryptoStream(stream,
    encryptor, CryptoStreamMode.Write);
```

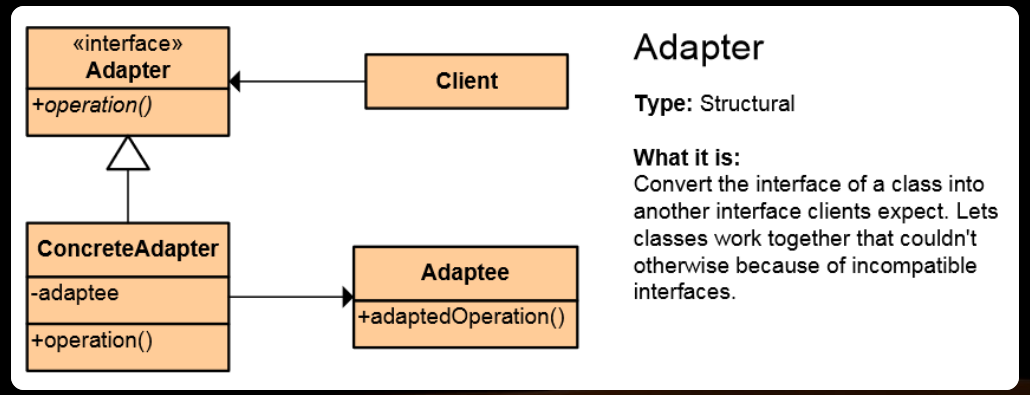
- **BufferedReader** в Java

```
BufferedReader bufferedReader =
  new BufferedReader(
    new InputStreamReader(
      new FileInputStream(
        new File("file_name.txt"))));
```

Шаблон Adapter

- Adapter преобразува интерфейса на даден клас в друг, изискван от клиента
  - Обгражда съществуващ клас с нов интерфейс
  - Преходник за напасване на стар компонент в нова система
- Позволява на класове да работят заедно, когато това е невъзможно заради различни интерфейси





Adapter – пример

Старият клас

```
class ChemicalDatabank
{
  public float GetMolecularStructure(string compound) {…}
  …
}
```

```
interface ICompound
{
  void Display();
}
```

Нужния ни нов интерфейс

```
public RichCompound : ICompound
{
  public RichCompound(string compound) {
    var chemicalBank = new ChemicalDatabank();
  }

  public void Display() {…}
}
```

Adapter клас

Обобщение

- Шаблони за дизайн в структурата на приложенията
  - Façade, Composite, Decorator, Adapter

Структурни шаблони за проектиране

Въпроси?

Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист"

- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под свободен лиценз **CC-BY-NC-SA**