

# Design Pattern #2

Peter Levinsky, IT Roskilde

20.03.2020

# Design Pattern – GRASP (General Responsibility Assignment Software Patterns)

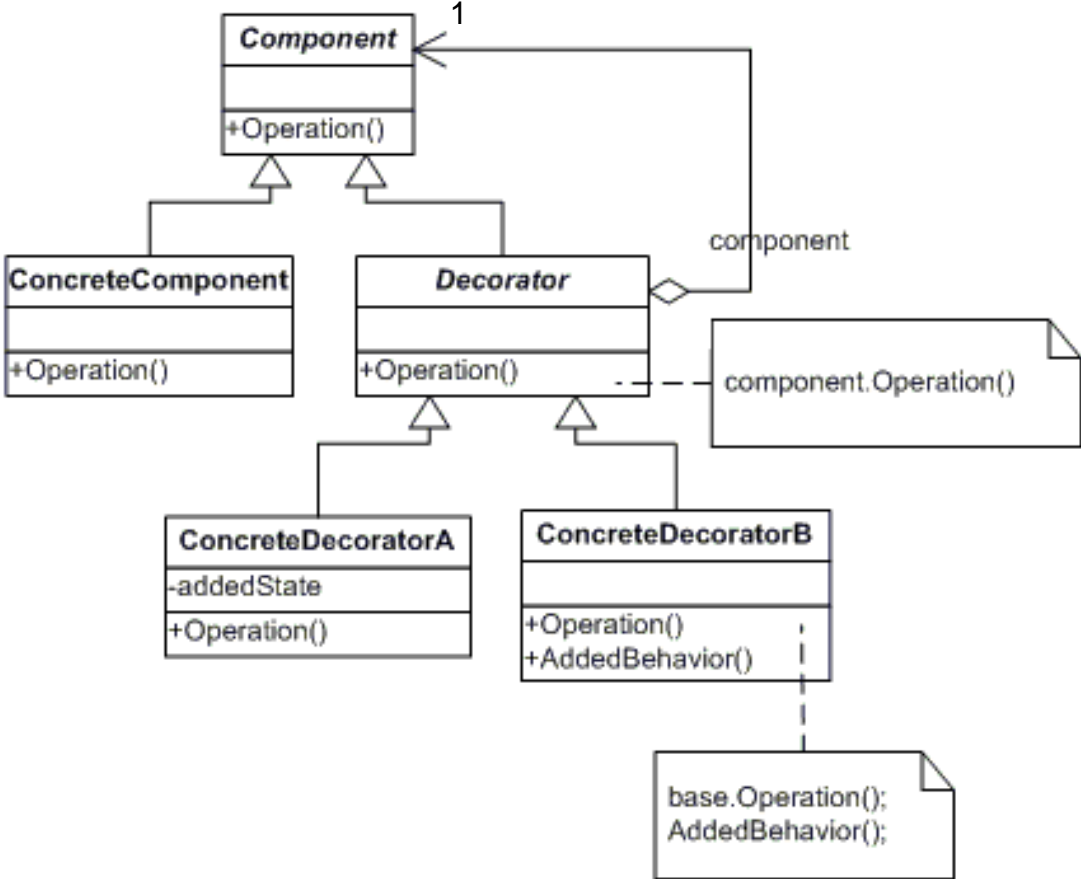
- Information Expert
- Creator Pattern
- Controller
- Low Coupling
- High Cohesion

# Design Pattern – Kategorier

- **Creational Patterns**
  - Factory, Abstract Factory, Singleton ...
- **Structural Patterns**
  - Adaptor, Proxy, Facade, Decorator ...
- **Behavioral Patterns**
  - Observer, Template, Chain of Responsibility, Strategi, State ...

# Design Pattern – Structural Patterns

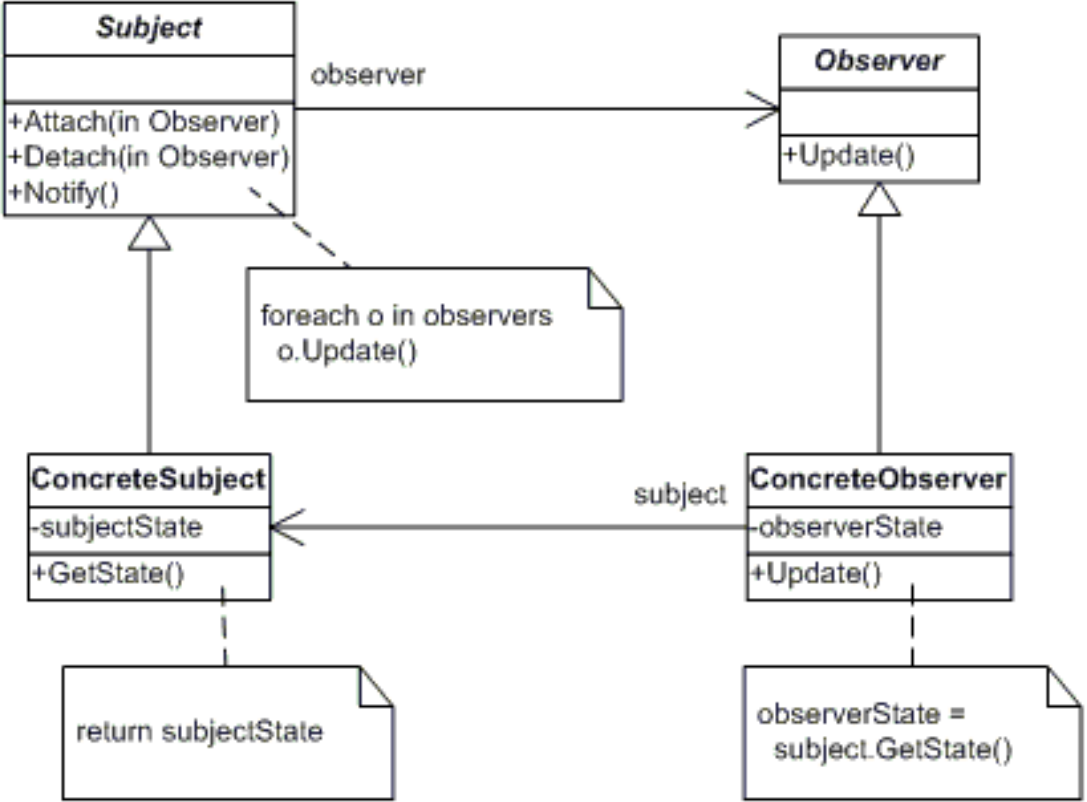
- **Decorator**
  - Problem: How to Attach additional responsibilities to an object dynamically
  - Løsning:



# Design Pattern – Behavioral Patterns

- **Observer** (*kender fra 1 semester*)  
Problem: How to handle different kinds of subscriber objects are interested in the state changes or events of a publisher object

• Løsning:



# Design Pattern – Behavioral Patterns

- **Observer** (*kender fra 1 semester*)
  - C# form

## Den der observerer

```
...
XX x = new XX();

// tilmeld som observatør
x.PropertyChanged += Update;

....
protected void Update(object sender,
                        PropertyChangedEventArgs arg)
{
    ...
}
```

## Observeret

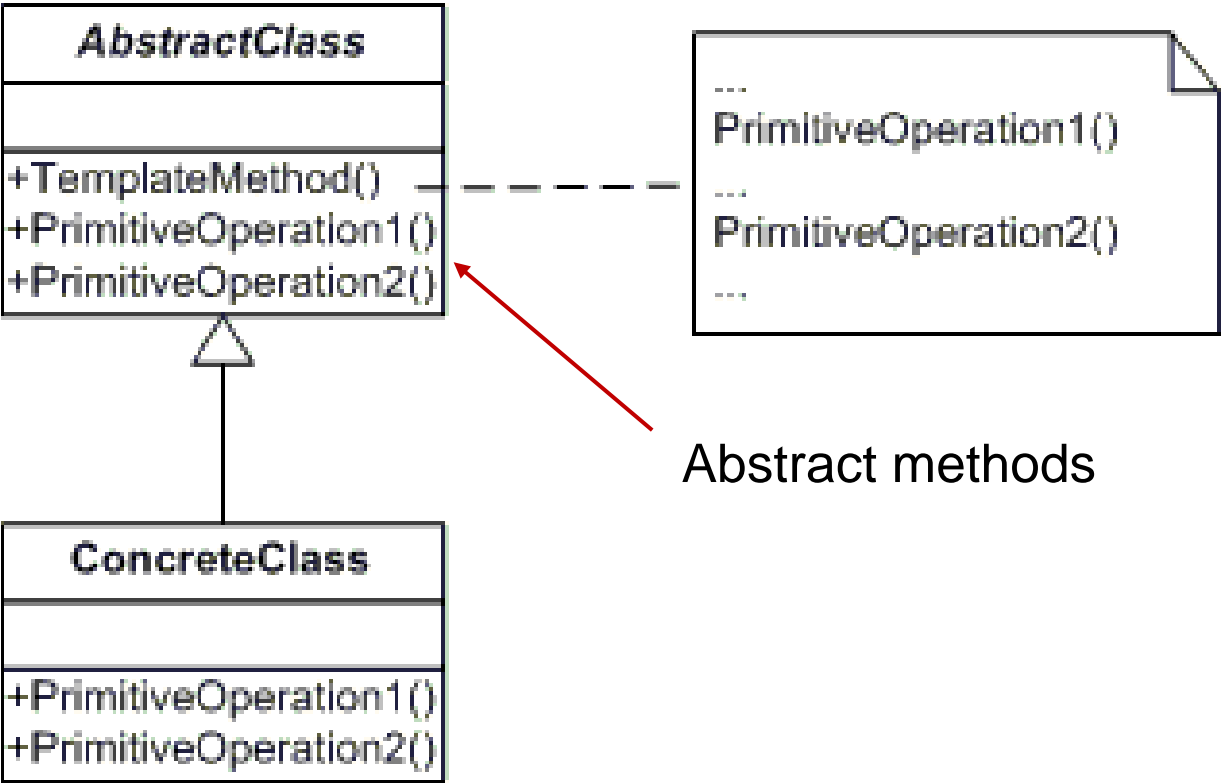
```
Class XX : INotifyPropertyChanged
{
    ...
    // Attach, Deattach
    public event PropertyChangedEventHandler PropertyChanged;

    // notify
    protected virtual void OnPropertyChanged(string propertyName)
    {
        PropertyChanged?.Invoke(this,
                                new PropertyChangedEventArgs(propertyName));
    }
}
```

# Design Pattern – Behavioral Patterns

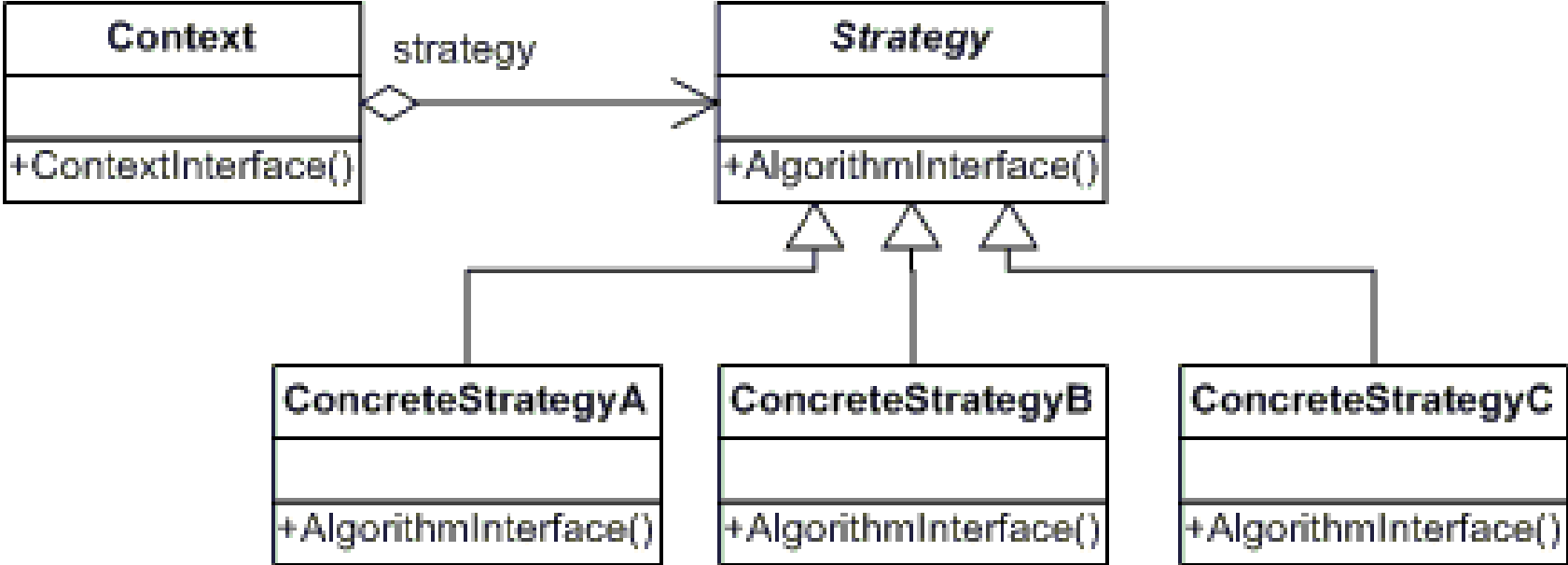
- **Template**  
Problem: How to reuse a skeleton of an algorithm in an operation

- Løsning:



# Design Pattern – Behavioral Patterns

- **Strategy**  
Problem: How to interchange part of algorithm dynamicly
- Løsning:



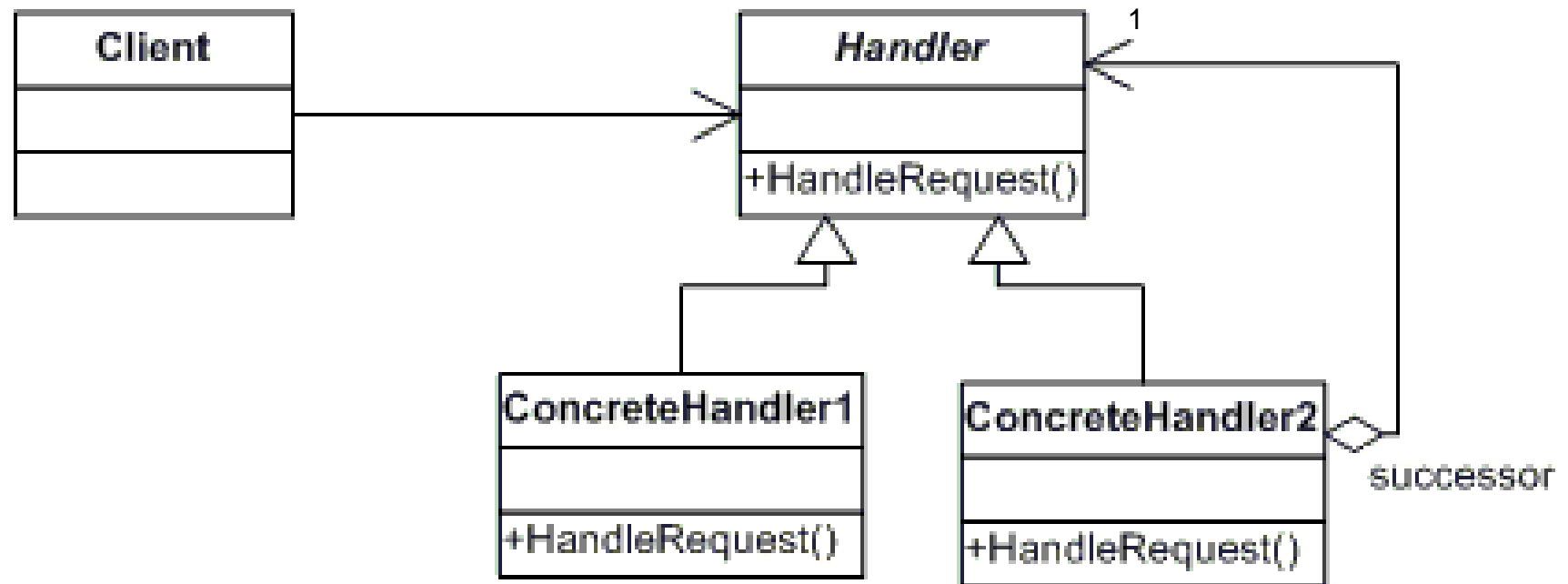


# Design Pattern – Behavioral Patterns

- **Chain of Responsibility**

Problem: How to avoid coupling the sender of a request to its receiver

- Løsning:



# Jeres Tur

- Først en kort demo af Decorator, Observer, Template og Strategy
- Opgaverne: Patterns.5, Patterns.6
- *Extra pizza opgave*
- *Til par programmering VS2019 Live Share*  
*(2017 ext. [Visual Studio Live Share](#))*
  - \* *Benyt GitHub 1) før start pull 2) programmer 3) push 4) skift*