

# Design Pattern

Peter Levinsky, IT Roskilde

16.03.2020

# Design Pattern - Beskrivelse

**Navn** – fælles betegnelsen – fag ord

**Problem** – beskrivelse af problemet

**Løsning** – KUN! Design løsning (UML diagrammer)

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

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

# Design Pattern – Flere Pattern fra 1.semester

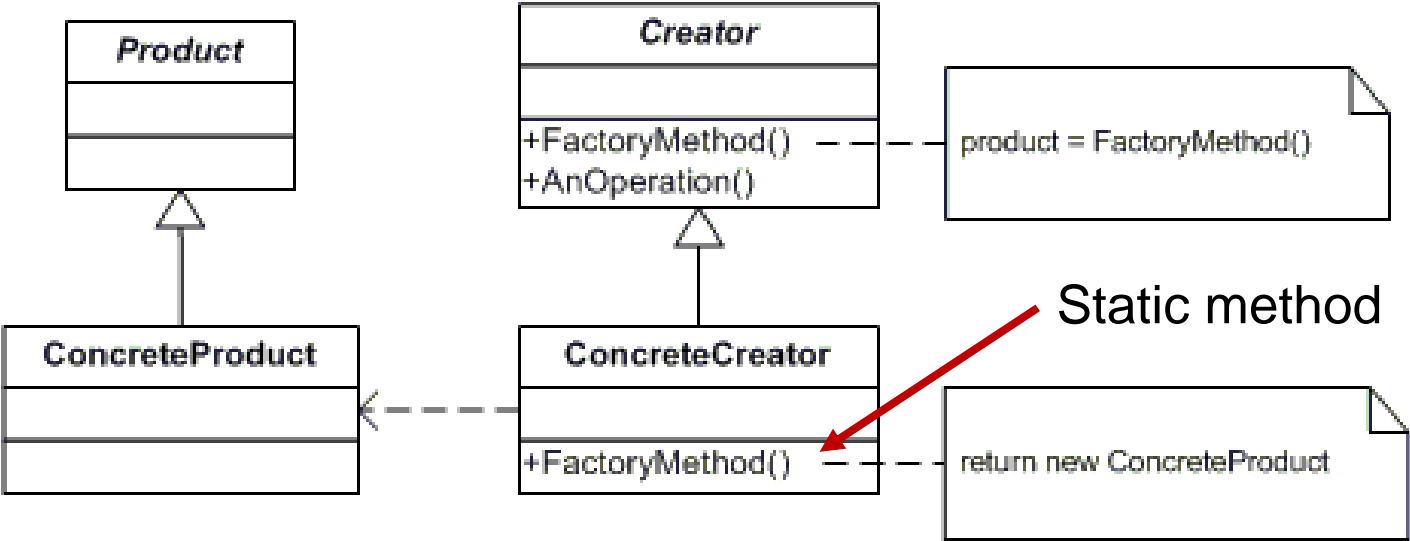
- Singleton
  - kun ét objekt
- Observer
  - sikre lav kobling

# Design Pattern – Kategorier

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

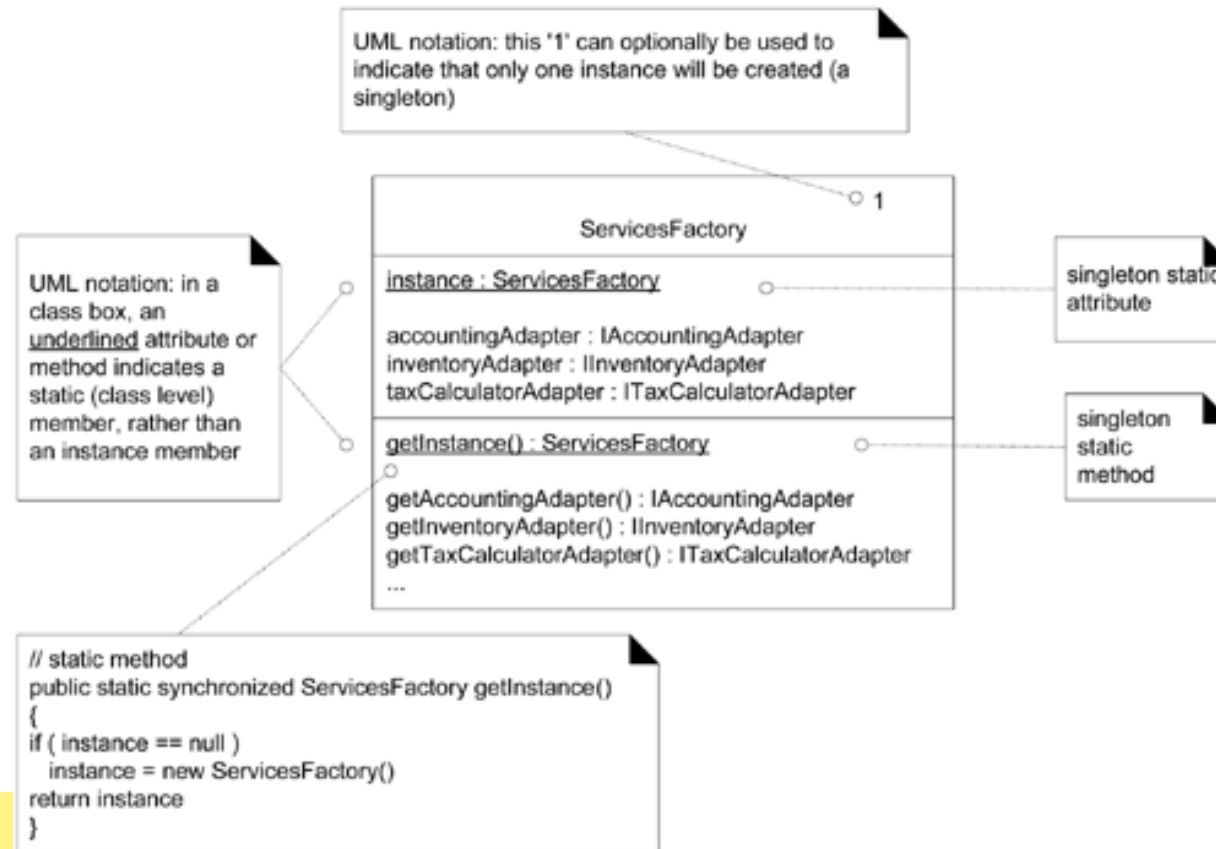
# Design Pattern – Creational Patterns

- Factory
  - Problem: Who should be responsible for creating objects when there are special considerations, such as complex creation logic, a desire to separate the creation responsibilities for better cohesion, and so forth?
  - Løsning:



# Design Pattern – Creational Patterns

- Singleton
  - Problem: Exactly one instance of a class is allowed.
  - Løsning:



# Demo og opgaver

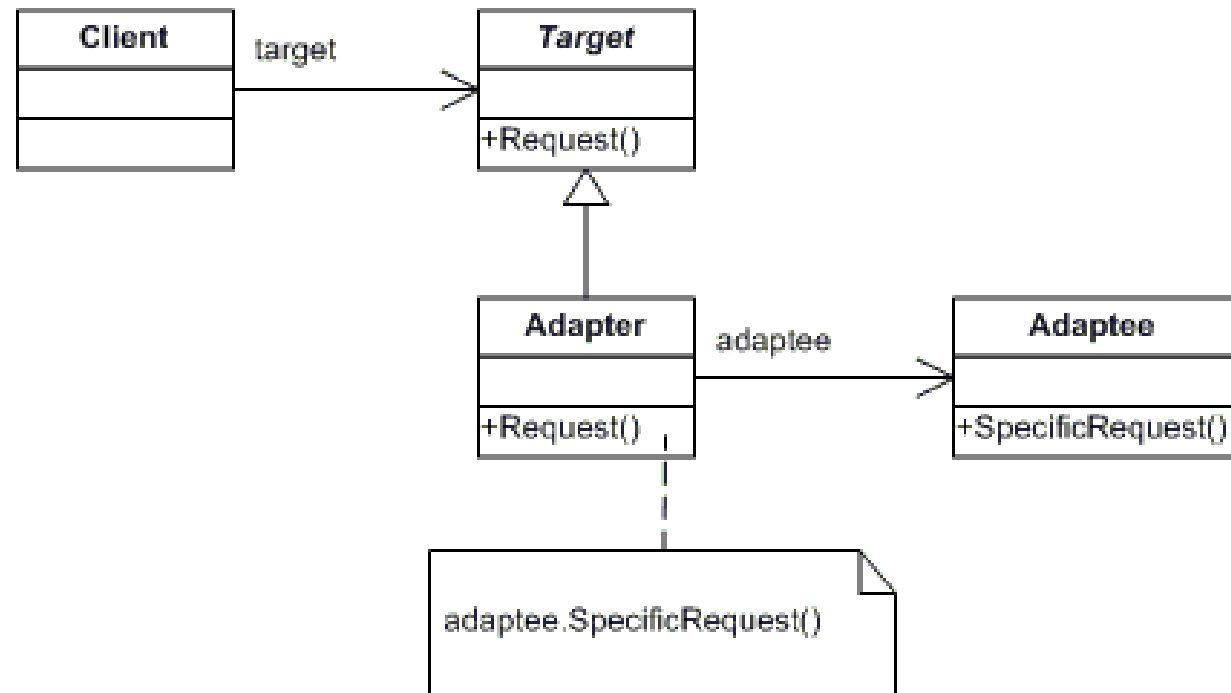
- Demo af Factory, Singleton og Abstract Factory
- Opgaverne Patterns.1 + Patterns.1
- Fortsætter med Structural Patterns kl 12:15



# Design Pattern – Structural Patterns

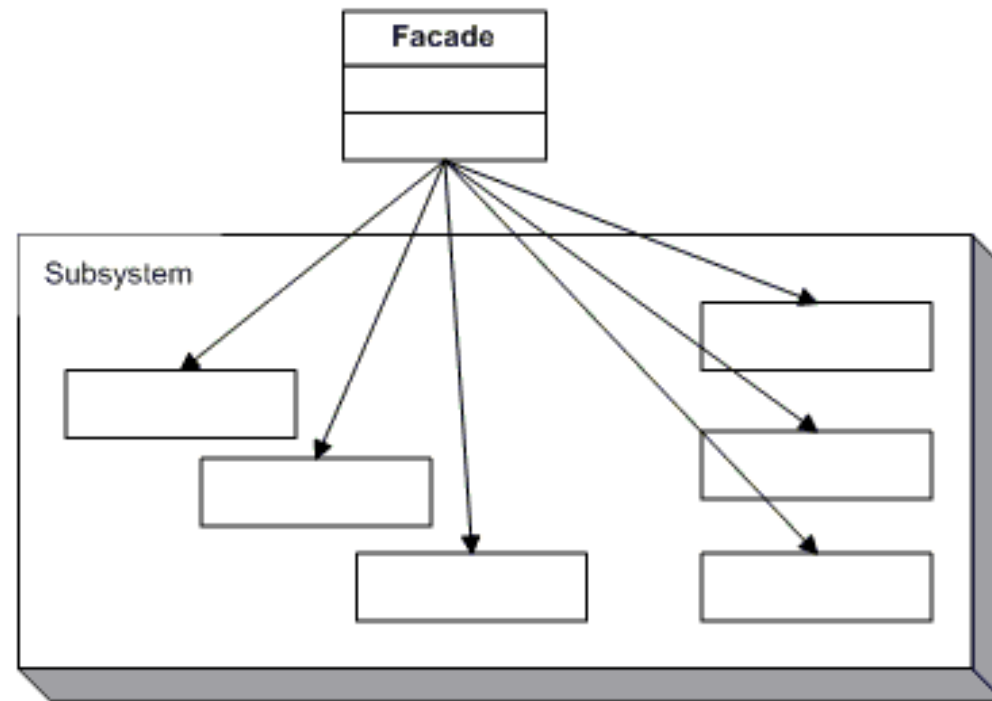
- Adaptor
- Problem: How to resolve incompatible interfaces, or provide a stable interface to similar components with different interfaces?

- Løsning:



# Design Pattern – Structural Patterns

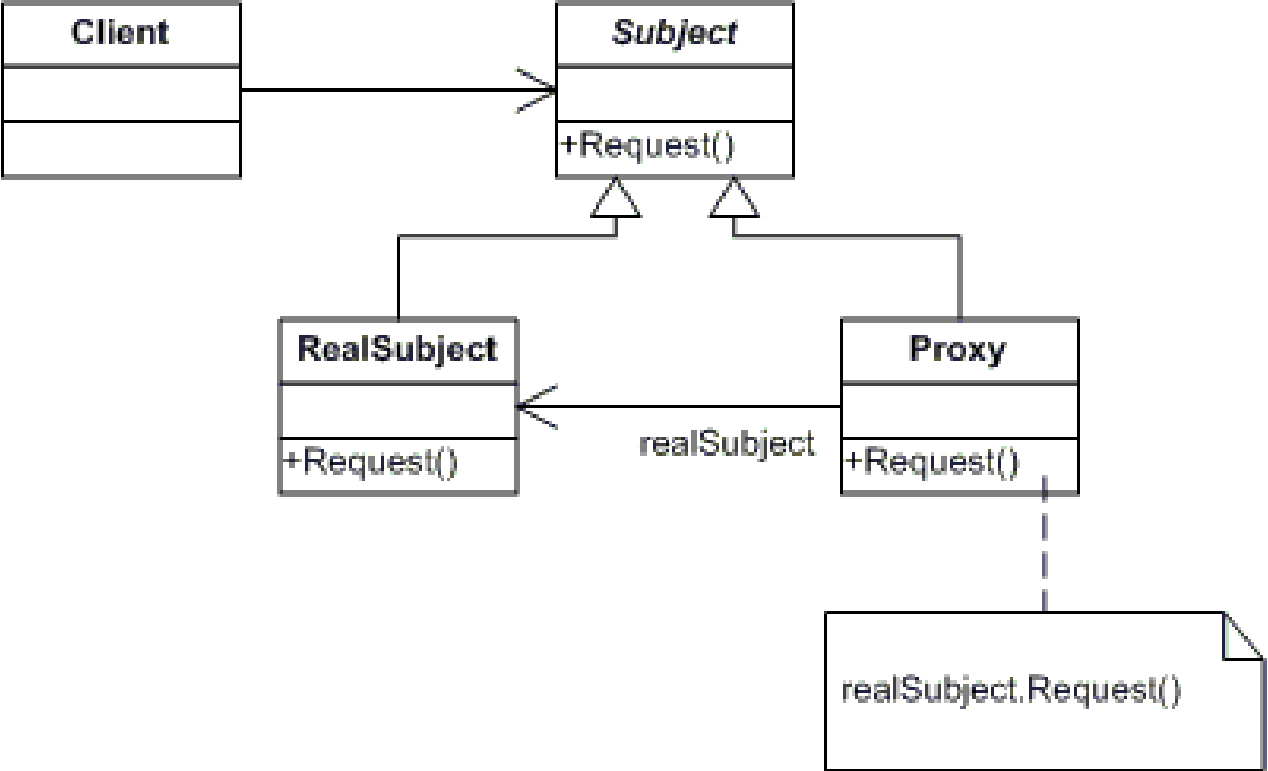
- Facade
  - Problem: A common, unified interface to a disparate set of implementations or Interfaces such as within a subsystem is required.
  - Løsning:



# Design Pattern – Structural Patterns

- Proxy
  - Problem: How to provide a placeholder for another object to control access to it.

• Løsning:



# Det kommende 2 semester system

