Rick Aerts, Tom Eisermann, Harm van Veen

Presentatie

Ondezoek naar het gebruik van design patterns binnen software engineering

Design Patterns

Proxy Design Pattern

[Design Patterns 2](#_Toc56943160)

[Wat zijn software design patterns? 2](#_Toc56943161)

[Hoe hebben design patterns hun intrede gedaan in het vak van software engineering? 2](#_Toc56943162)

[Welke software design pattern categorieën onderkennen we? 3](#_Toc56943163)

[Wat zijn de voordelen van design patterns? 4](#_Toc56943164)

[Wat zijn de nadelen van design patterns? 4](#_Toc56943165)

[Proxy Structural Design Pattern 4](#_Toc56943166)

[Virtual Proxy 4](#_Toc56943167)

[… Proxy 4](#_Toc56943168)

[… Proxy 4](#_Toc56943169)

# Design Patterns

## Wat zijn software design patterns?

* Gestandaardiseerde oplossingen voor veel voorkomende problemen in software ontwerp.
* **WEL** een blauwdruk, template of mal. **GEEN** kant en klare code.
* **Opbouw:**
  + Intent  
    Een korte samenvatting van zowel het probleem, als de oplossing
    - Algemene termen
  + Motivation  
    Een uitgebreidere uitleg van het probleem, en hoe het ontwerp daar een oplossing voor bied.
  + Structure  
    Een gedetaileerdere omschrijving van de verschillende onderdelen/classes binnen het patroon, en hoe deze aan elkaar gerelateerd zijn.
    - Class diagram
    - Pseudocode
    - Etc.
  + Code Example  
    Een uitwerking van het patroon in code om een concreet voorbeeld te geven.

## Hoe hebben design patterns hun intrede gedaan in het vak van software engineering?

* Het gebruik van design patterns (in algemene termen) wordt buiten software engineering sinds jaar en dag toegepast.
  + Men heeft probleem
  + Men vind verschillende mogelijke oplossingen
  + Oplossingen worden breed toegepast, en verfijnd
  + Iemand besluit om de resulterende oplossing een naam te geven.
* Een vroege poging tot definiëren van design patterns:
  + A pattern Language: Towns, Buildings, Constructions (1977)
  + Christopher Alexander
* Een van de eerste publicaties van gebruik in context software engineering:
  + Design Patterns: Elements of Reusable Object-Oriented Software (1994)
  + Erich Gamma, John Vlissides, Ralph Johnson, Richard Helm
  + “Gang of Four”
  + 23 patterns omschreven binnen OO programming

## Welke software design pattern categorieën onderkennen we?

* 2 manieren van onderscheid:
  + Toepasbaarheid
  + Intent/Doel

### Toepasbaarheid

#### Idioms

* Basic / Low Level
* Toepasbaar op één taal

#### Architectural Patterns

* Universal / High Level
* Toepasbaar in nagenoeg alle talen.
* Kan gebruikt worden voor gehele applicatie ontwikkeling

### Intent

#### Creational Patterns

* Leveren mogelijkheden voor object creatie die flexibiliteit en herbruikbaarheid van code bevorderd.
  + Factory Method
  + Abstract Factory
  + Builder
  + Prototype
  + Singleton

#### Structural Patterns

* Leveren mogelijkheden om objecten/classes in grotere structuren te verbinden die toch flexibel en efficiënt blijven
  + Adapter
  + Bridge
  + Composite
  + Decorator
  + Facade
  + Flyweight
  + Proxy

#### Behavioral Patterns

* Leveren mogelijkheden voor effectieve communicatie en toekenning van verantwoordelijkheden tussen objecten.
  + Chain of Responsibility
  + Command
  + Itterator
  + Mediator
  + Memento
  + Observer
  + State
  + Strategy
  + Template Method
  + Visitor

## Wat zijn de voordelen van design patterns?

* Design patterns zijn beproefde oplossingen voor veel voorkomende problemen
  + Het wiel hoeft niet telkens opnieuw ontworpen te worden.
  + Minder kans op onverwachte verassingen.
* Design patterns bieden een gedeelde vocabulaire voor communicatie over ontwerp

## Wat zijn de nadelen van design patterns?

* Het is een steunpillaar voor zwakke programmeer talen die weinig ondersteuning voor abstractie bieden.
* Inefficiënte oplossingen voor problemen wanneer ze blindelings gevolgd worden zonder ze naar de situatie aan te passen.
* Ongerechtvaardigd gebruik / Toepassing op momenten waar het niet nodig is en een simpelere oplossing zou volstaan.

# Proxy Structural Design Pattern

## Virtual Proxy

### Probleem

### Oplossing

## … Proxy

### Probleem

### Oplossing

## … Proxy

### Probleem

### Oplossing