

Het Builder Pattern

Analysis & design 2

Builder Pattern

Doel

- Het Builder Pattern wordt gebruikt bij de **creatie** en configuratie van **complexe objecten**.
- Het Builder Pattern **scheidt het creatieproces** van een object **af** van het object zelf.
- Het Builder Pattern laat toe om **met een zelfde creatieproces verschillende objectvariëteiten** te maken.

Builder Pattern

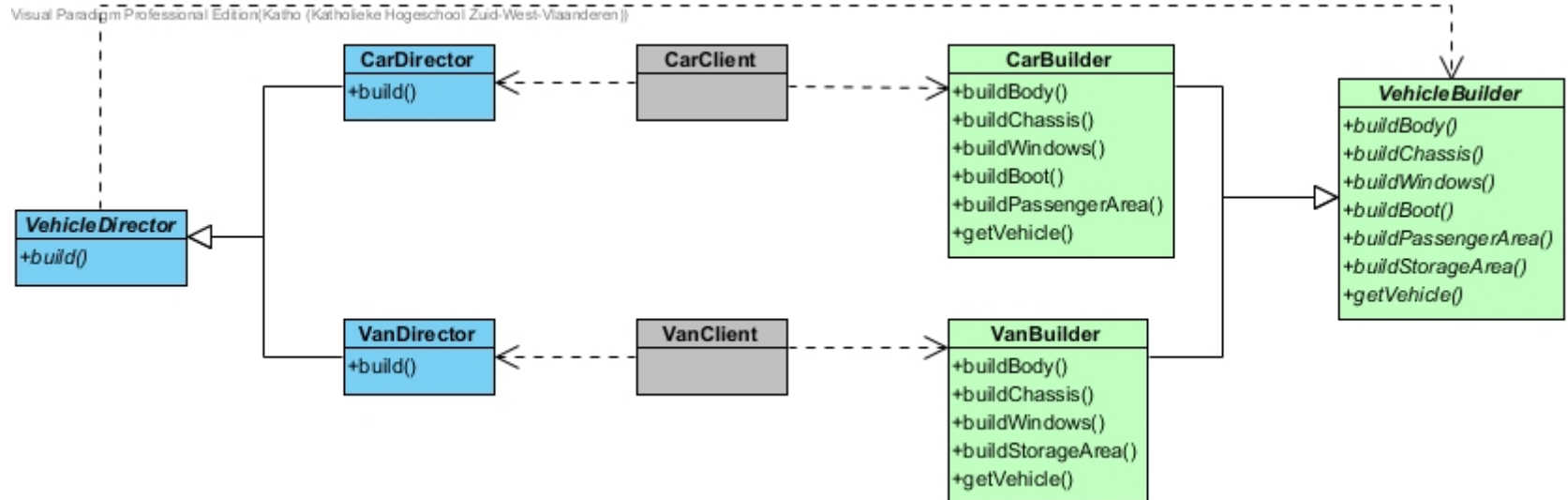
Nut

Telescoping constructor anti-pattern: soms heeft een klasse heel veel constructors. Dat is een aanwijzing dat de constructie van een object niet eenvoudig is.

Met het Builder Pattern kun je het constructieproces voor een client vereenvoudigen.

Builder Pattern

Deelnemende klassen



Builder Pattern

Builder-klasse(n)

- Verantwoordelijk voor de constructie van een ingewikkeld objecttype
- Heeft methodes voor de verschillende stappen in het constructieproces
- Heeft een accessormethode voor het afgewerkte object
- Moet niet per se een hiërarchie van klassen zijn

Builder Pattern

Director-klasse(n)

- Verantwoordelijk voor het aansturen van de Builder-klassen
 - Methode(s) die de constructie van een object voorstellen als een geheel – high-level constructie.
-
- Moet niet per se een hiërarchie van klassen zijn
 - Kun je eventueel samensmelten met de Builder-klassen

Builder Pattern

Client

- Instantieert een Builder
- Instantieert een Director
- Koppelt de Builder aan de Director
- Gebruikt de Director om de Builder aan te sturen, en zo op een eenvoudige manier een ingewikkeld object te construeren
- Krijgt het geconstrueerde object van de Builder

Builder Pattern

Product

- Datgene wat de Builder maakt
- Iets met een ingewikkelde constructie
- Staat niet in het schema

Builder Pattern

Variant zonder director

- Deze variant kan een alternatief zijn voor een constructor met veel parameters.

```
Acme acme = new Acme("wizbang", "automatic", 3, 8.2)
```

- Voor klasse Acme bestaat een builder-klasse AcmeBuilder (dat kan een geneste klasse zijn).

AcmeBuilder heeft methodes die elk een aspect van de constructie uitvoeren. Elk van die methodes retourneert de builder (this).

AcmeBuilder heeft ook een accessor-methode (build()) voor de geconstrueerde Acme.

Builder Pattern (variant)

Een nieuwe Acme construeren met de builder

```
// eerste versie
```

```
Acme acme = new Acme("wizbang", "automatic", 3, 8.2)
```

```
// met builder
```

```
Acme acme = Acme.builder()  
    .setGizmo("wizbang")  
    .setFlyType("automatic")  
    .setFoo(3)  
    .setBaz(8.2)  
    .build();
```

Builder Pattern (variant)

Acme

```
public class Acme
{

    public static AcmeBuilder builder()
    {
        return new AcmeBuilder();
    }

    // rest van de klasse
}
```

Builder Pattern (variant)

AcmeBuilder

```
public class AcmeBuilder
{
    // te construeren object
    private Acme newAcme;
    private string gizmo, flyType;
    private int foo;
    private double baz;

    // constructor
    public AcmeBuilder() { // evtl hier al constructie van Acme }

    // accessor voor product
    public Acme build() { return new Acme(gizmo, flyType, foo, baz); }
}

// hulpmethode voor constructie
public AcmeBuilder setGizmo(string gizmoParm) {
    gizmo = gizmoParm; return this;
}
```