



## Modelleren 3

Blok 5

# van Model naar Code

## Implementatie van 'complexe' relaties

In dit document staat een reeks opdrachten die gaan over het implementeren van fragmenten van klassendiagrammen.

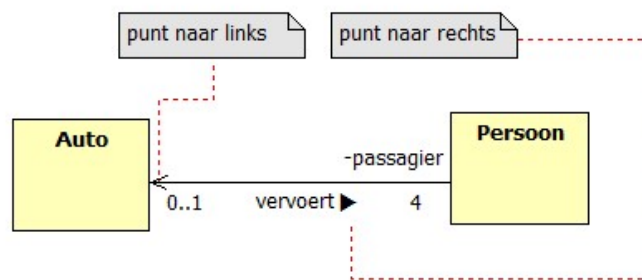
### opdracht

- Geef de implementatie van ieder fragment in C# code. (uitzonderingen staan erbij)
- Geef alle varianten als er meerdere mogelijkheden zijn.
- Maak een MS-Word document, plak die stukjes code erin op volgorde.
- Beantwoord ook de vragen die naast de fragmenten staan.

### 3.1

Er staan twee pijlpunten in dit fragment die verschillende richtingen in wijzen, mag dit?

Hoe kun je de multipliciteit van **precies 4** implementeren? Let op de gegeven navigatierichting?



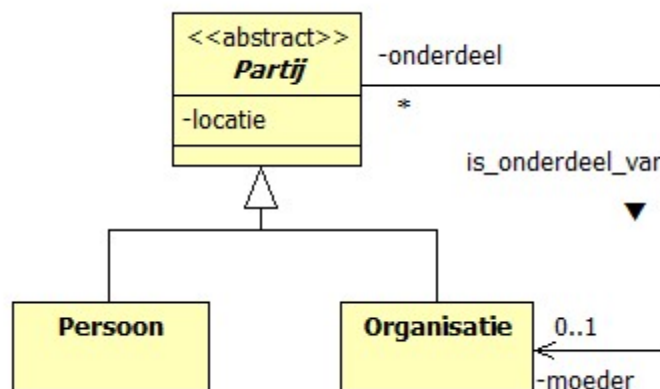
Hoe zou het model er uit zien als de bestuurder niet onder de noemer "passagier" zou vallen, wel als object `:Persoon` zou voorkomen en ook in hetzelfde object `:Auto` zit?

### 3.2

*<< abstract >> staat hier als een stereotype, dat is fout en een tekortkoming van StarUML. Het zou als een property {abstract} tussen accolades moeten staan volgens de UML standaard.*

Hoe kun je een abstracte klasse implementeren?

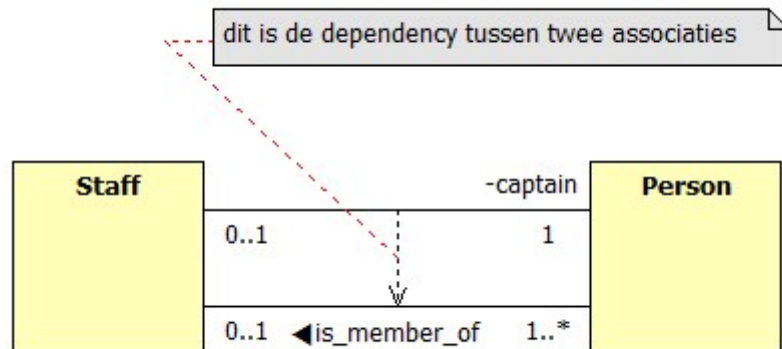
Hoe kun je de associatie tussen Partij en Organisatie implementeren?



### 3.3

Met een dependency tussen twee associaties kun je de onderlinge afhankelijkheid aangeven. Hier wordt bedoeld dat een captain ook member van de staff moet zijn.

Hoe implementeer je deze dependency?

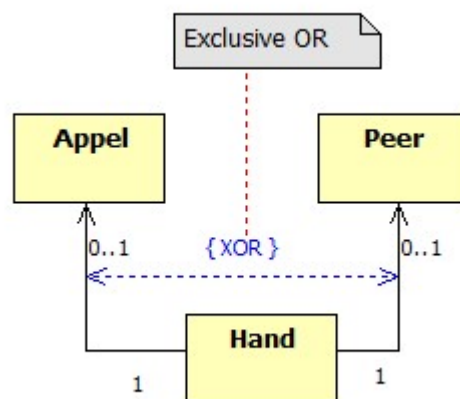


### 3.4

Een bijzondere dependency tussen associaties is de Exclusive-OR, dat wil zeggen "een-van-beide" en niet "allebei".

Hoe implementeer je deze dependency?

Hoe kun je dit met overerving oplossen zodat je geen XOR nodig hebt?



Een :Hand kan dus blijkbaar een :Appel of een :Peer hebben en niet allebei, maar hoe zit het met de multipliciteit aan de zijde van :Hand, die staat namelijk in het model als precies 1 bij allebei. Is dat correct?

Is het probleem met de multipliciteit bij :Hand met het introduceren van een :Fruitmand nu opgelost?

