

HBO-ICT richtlijnen m.b.t. UML

Versie: 1.2

Datum: 17-9-2023

Inhoud

1	C# Properties	2
2	Use case diagram.....	2
3	Sequentiediagram	2

1 C# Properties

In de programmeertaal Java werken we met zogenaamde getters en setters. De tool Visual Paradigm biedt daar ondersteuning voor. Voor attributen van een klasse kun je zelfs automatisch getters en setters genereren. In de taal C# werken we echter met properties. De tool Visual Paradigm kent daarvoor het stereotype <<property>>. Goed te gebruiken in een klassendiagram.

In een sequentiediagram kun je die echter niet aanroepen. De afspraak is om ook in het geval we in C# programmeren in de sequentiediagrammen met getters en setters te werken. In de code hanteren we echter properties.

2 Use case diagram

Een HBO-ICT Richtlijn voor het use case diagram is dat we geen gebruik maken van generalisatie van use cases. Dat is overigens een tip die je in meerdere boeken tegen komt.

Een andere richtlijn van HBO-ICT is dat we in principe geen extends gebruiken. Alleen bij het aanbieden van klantspecifieke functionaliteit in een standaardpakket is dit het overwegen waard.

3 Sequentiediagram

Activation bars zijn in UML optioneel. Ze geven echter veel inzicht in het gedrag van de objecten. Een HBO-ICT richtlijn is dat we in de documentatie altijd activation bars tonen. Als je een sequentiediagram gebruikt tijdens een overleg met je collega systeemontwikkelaars, bijvoorbeeld op een whiteboard, dan is het in principe toegestaan om alleen een life-line te tekenen.

C# werkt niet met getters en setters, maar met properties (zie hoofdstuk 1 C# Properties). Als het nodig is om in een sequentiediagram een property op te vragen of een waarde te geven, nemen we deze op als een operatie waarvan de naam begint met get respectievelijk set. Op de Java manier dus eigenlijk.