



TA-102

Introductie UML

Docentenhandleiding



TA102 Introductie Ontwikkelaar

Topicus Academy

Inhoudsopgave

Cursusomschrijving	1
Inleiding	1
Inhoud	1
1. UML: de diagrammen	2
1.1. Class diagram	2
1.2. Sequence diagram	4
1.3. Deployment diagram	4

Cursusomschrijving

UML is een diagramspecificatietaal om software te kunnen ontwerpen en deze ontwerpen met anderen te kunnen delen. Deze cursus introduceert deze taal zodat trainees de diagrammen kunnen volgen in discussies bij het whiteboard.

Inleiding

De *Universal Modeling Language* (UML) wordt sinds het eind van de jaren 90 van de vorige eeuw gebruikt en is de standaard in de software industrie. Daarnaast geeft UML de handvatten om wanneer men een stuk software analyseert deze te visualiseren, wat het begrip van dat stuk software vergroot. Een goed begrip van deze notatietechniek is dan ook essentieel om effectief te kunnen communiceren met andere ontwikkelaars.

Doelen

Het doel van deze introductie is om trainees net gevaarlijk genoeg te laten zijn bij een whiteboard en hen te laten weten welke technieken er zijn om software grafisch weer te geven.

Na deze cursus kan de trainee:

- De meest essentiële diagrammen lezen
- Zelf deze schetsen op een whiteboard

Het is niet de bedoeling dat dit een formele training wordt om tot correcte UML te komen.

Inhoud

In deze modules komen de volgende diagrammen aan bod:

- class diagram
- sequence diagram
- deployment diagram

Deze zijn gekozen omdat ze het beste werken op een whiteboard (class en deployment diagrams), of helpen bij het nadenken over een stuk code in logische volgorde (sequence diagram). De overige UML diagrammen zijn natuurlijk nuttig, maar voor deze setting niet noodzakelijk.

Hierbij maken we gebruik van het onderstaande artikel van Laurie Williams.

Bronnen

[] Laurie Williams. [An Introduction to the Unified Modeling Language](#). 2004

Chapter 1. UML: de diagrammen

In een whiteboard sessie leg je de meest gebruikte diagrammen van UML uit.

Je kan een korte introductie geven over UML:

- veel verschillende O.O. diagram technieken → OMG → UML
- Unified Process
- Mogelijkheid om van diagrammen code te genereren (slecht idee)

1.1. Class diagram

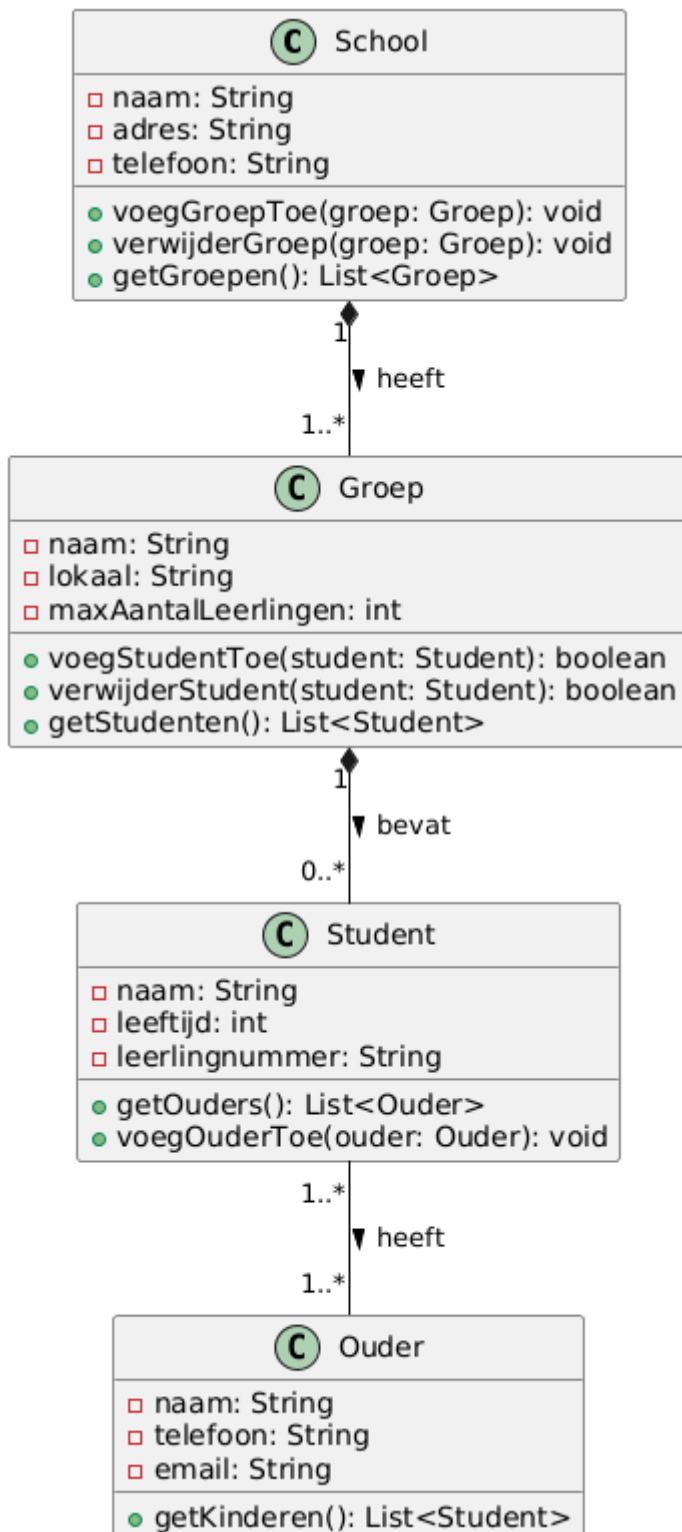
Laat zien hoe een class diagram eruit ziet.

Bij het uitleggen van een class diagram aan iemand zonder kennis van UML of objectoriëntatie, volg je best deze stappen:

1. Begin met het vergelijken van een class met een blauwdruk of recept: "Een class is als een recept voor het maken van objecten. Net zoals een recept je vertelt hoe je een taart maakt, vertelt een class de computer hoe een object eruit moet zien."
2. Teken een eenvoudig voorbeeld van een rechthoek met drie secties:
 - Bovenste deel: "Dit is de naam van je 'ding', bijvoorbeeld 'Auto'"
 - Middelste deel: "Hier schrijven we op wat een Auto 'heeft', zoals 'kleur', 'merk', 'kenteken'"
 - Onderste deel: "Hier schrijven we op wat een Auto 'kan doen', zoals 'rijden()', 'parkeren()', 'tanken()'"
3. Maak duidelijk dat attributen eigenschappen zijn en methods acties: "De middelste sectie bevat de eigenschappen die beschrijven hoe het ding is. De onderste sectie bevat de acties die het ding kan uitvoeren."
4. Introduceer relaties met pijlen tussen klassen:
 - "Als twee dingen met elkaar te maken hebben, tekenen we een lijn ertussen."
 - "Een Auto heeft bijvoorbeeld een Motor, dus tekenen we een lijn tussen deze twee."
 - Gebruik voorbeelden uit hun dagelijks leven: "Een Bedrijf heeft Medewerkers" of "Een Winkel verkoopt Producten"
5. Voor overerving, gebruik een familievergelijking:
 - "Stel je voor dat 'Voertuig' een algemeen concept is. 'Auto', 'Fiets', en 'Bus' zijn allemaal voertuigen, maar met eigen specifieke eigenschappen."
 - "In UML tekenen we een pijl met een holle driehoekige punt van de specifieke dingen (Auto, Fiets) naar het algemene ding (Voertuig)."
 - "Dit betekent dat een Auto alles heeft en kan wat een Voertuig heeft en kan, plus zijn eigen extra dingen."
6. Benadruk dat class diagrammen bedoeld zijn om structuur en relaties te laten zien: "Dit diagram helpt ons om te zien hoe verschillende onderdelen van een systeem met elkaar

verbonden zijn, zonder ons te verliezen in details."

Hier is een voorbeeld van een class diagram met PlantUML dat je kunt gebruiken om de concepten te illustreren:



In dit diagram zien we:

1. Classes met attributen en methoden:

- Elke rechthoek stelt een class voor, met naam, attributen en methods
- Het "-" teken betekent private, "+" betekent public

2. **Compositie relatie** (gevulde ruit):

- Een School bestaat uit meerdere Groepen
- Een Groep bestaat uit meerdere Studenten
- Als de School of Groep verdwijnt, verdwijnen ook de onderliggende elementen

3. **Associatie relatie** (lijn):

- Student en Ouder hebben een many-to-many relatie (meerdere Studenten kunnen dezelfde Ouders hebben)

4. **Multipliciteit:**

- "1" betekent precies één
- "1..*" betekent één of meer
- "0..*" betekent nul of meer

Dit diagram kun je gebruiken om verschillende concepten uit te leggen zoals compositie, associatie, multipliciteit, en attributen/methoden.

Denk eraan dat je visuele voorbeelden gebruikt en de trainees laat meedoen door hen eenvoudige voorbeelden te laten bedenken uit hun eigen werkomgeving. Begin met 2-3 klassen en voeg geleidelijk meer complexiteit toe naarmate ze comfortabel worden met het concept.

1.2. Sequence diagram

1.3. Deployment diagram