

Klassendiagram - opdrachten

Versie: 1.0

Datum: 29-8-2022

Inhoud

1.	Klasse Vlucht.....	2
2.	Klassen Vlucht en Luchthaven.....	2
3.	Klasse Persoon.....	2
4.	Klassen Persoon en Bedrijf.....	3
5.	Voetbalteam: domeinmodel.....	3
6.	Schaaktoernooi: domeinmodel.....	3
7.	Speelstukken Stratego.....	5
8.	Klasse Order: attributen weergeven als associaties.....	5
9.	Associatie Order - OrderLine: aggregatie of compositie?.....	6
10.	Lid in rol van Koper en Verkoper.....	6
11.	Domeinmodel vliegtuigvluchten.....	7
12.	Theater reserveringssysteem: domeinmodel.....	8
13.	Theater reserveringssysteem uitbreiding: domeinmodel.....	8
14.	Poor man's console MVC applicatie: software klassen.....	9
15.	Tips.....	9

1. Klasse Vlucht

De klasse Vlucht is hieronder weergegeven in C#-code.

```
public class Vlucht
{
    public String Vluchtnummer { get; set; }
    public DateTime Vertrektijd { get; set; }
    public DateTime Aankomsttijd { get; set; }
}
```

Geef de klasse Vlucht weer in een klassendiagram, ervan uitgaande dat het gebruikt wordt als een domeinmodel.

2. Klassen Vlucht en Luchthaven

De klassen Vlucht en Luchthaven worden hieronder weergegeven in C#-code.

```
public class Vlucht
{
    public String Vluchtnummer { get; set; }
    public DateTime Vertrektijd { get; set; }
    public DateTime Aankomsttijd { get; set; }

    public Luchthaven Vertrekpunt { get; set; }
    public Luchthaven Bestemming { get; set; }
}

public class Luchthaven
{
    public string Naam { get; set; }
}
```

Geef de klassen Vlucht en Luchthaven weer in een klassendiagram, ervan uitgaande dat het gebruikt wordt als een domeinmodel.

3. Klasse Persoon

Een persoon heeft een achternaam en één of meerdere voornamen. Hoe zou je dat in een klassendiagram weergeven ervan uitgaande dat je dit modelleert als één klasse en twee attributen? Met name de meerdere voornamen vind je misschien lastig. Probeer uit te vinden hoe je dit doet.

4. Klassen Persoon en Bedrijf

- a. Een persoon werkt bij een bedrijf. Modelleer dit in een klassendiagram.
- b. **Klassen Persoon en Bedrijf: associatierollen**
En stel nu dat je bovenstaande niet met een associatienaam, maar door middel van associatierollen wilt weergeven. Hoe zou dit er dan uit zien?
- c. **Klassen Persoon en Bedrijf: associatierollen en associatienamen**
En stel nu dat je zowel associatienaam als associatierol wilt weergeven. Hoe ziet het klassendiagram er dan uit.

5. Voetbalteam: domeinmodel

Een voetbalteam bestaat uit een aantal spelers. Een speler kan bij maximaal één voetbalteam ingedeeld worden.

- a. Modeleer bovenstaande in een domeinmodel
- b. Hoe zou je kunnen modeleren dat een voetbalteam als compleet wordt gezien als deze uit minimaal 11 spelers bestaat? Deze opdracht is wellicht een uitdaging voor je. Voor het geval je er niet uitkomt: op het eind van dit document staat een tip.

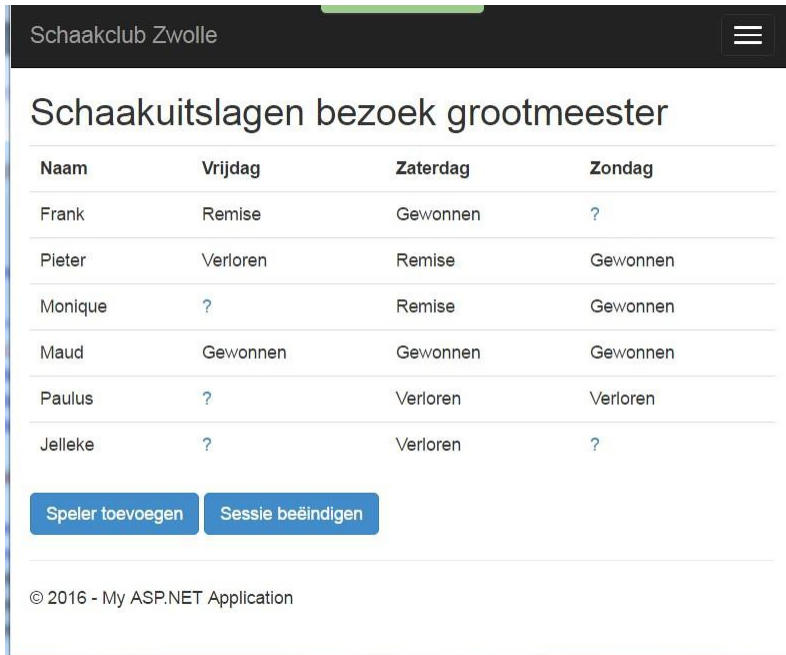
6. Schaaktoernooi: domeinmodel

Voor het vak Server is een oefententamen beschikbaar, waarbij de opdracht is om een eenvoudige applicatie te bouwen voor een schaaktoernooi. Voor deze opdracht is een specificatie beschikbaar waarvan hieronder een deel wordt weergegeven.

Casus beschrijving:

- De schaakclub Zwolle krijgt ieder jaar bezoek van een schaakgrootmeester. Er wordt dan een toernooi georganiseerd waarbij clubleden iedere toernooiday een partij tegen de grootmeester spelen.
- We gaan ervan uit dat een toernooi 3 dagen duurt, maar je gaat het wel zodanig programmeren dat dit vrij gemakkelijk kan worden aangepast.
- Om de uitslagen vast te leggen wordt een webapplicatie ontwikkeld. Je gaat dat in dit tentamen uitwerken. Hieronder wordt een indruk gegeven van de te ontwikkelen applicatie.
- Een speler heeft een naam. De naam is verplicht en heeft een minimale lengte van 2 karakters.

Hieronder wordt een indruk gegeven van de te ontwikkelen applicatie.



Naam	Vrijdag	Zaterdag	Zondag
Frank	Remise	Gewonnen	?
Pieter	Verloren	Remise	Gewonnen
Monique	?	Remise	Gewonnen
Maud	Gewonnen	Gewonnen	Gewonnen
Paulus	?	Verloren	Verloren
Jelleke	?	Verloren	?

Speler toevoegen Sessie beëindigen

© 2016 - My ASP.NET Application

De docent kan een werkende applicatie demonstreren of je kunt het filmpje bekijken dat in de folder 'aanvullend materiaal opdrachten' vinden is. Wij gaan deze opdracht gebruiken om te oefenen met het maken van een domeinmodel.

Opdracht: Maak het bijbehorende domeinmodel. Voor deze opdracht willen we dat jullie voor alle attributen het type en eventuele constraints opnemen. Voor alle relaties willen we de multipliciteit zien. Uiteraard ontbreken de enumeraties niet in het domeinmodel!

N.b.: UML verplicht je niet om alle attributen, types, multipliciteiten e.d. te tonen in het domeinmodel. Jij als ontwerper bepaalt gegeven de context het detailleringsniveau. Maar voor deze opdracht willen we een zeer gedetailleerd model.

7. Speelstukken Stratego

Een uitgever van computerspelletjes wil het spel 'Stratego' op de markt brengen.

Spel in het kort: beide spelers hebben de beschikking over 40 speelstukken. 33 daarvan zijn soldaten die verschillende rangen hebben. Daarnaast zijn er nog 6 bommen en een vlag.

De spelers plaatsen hun eigen stukken op het bord, zodanig dat de tegenstander de rang niet kan zien. De spelers mogen om beurten een stuk één veld verplaatsen (naar voren, achter, links, of rechts), naar een vrij veld of naar een veld waar al een stuk van de tegenpartij staat. In het laatste geval is er sprake van een aanval, de soldaat met de hoogste rang wint (bij gelijke rang worden beide speelstukken van het bord verwijderd). Een verkenner mag echter onbeperkt over lege velden springen. Een bom wint van alle soldaten, behalve van een mineur. Bommen en vlaggen kunnen niet bewegen. Het doel van het spel is om de vlag van de tegenpartij te slaan.

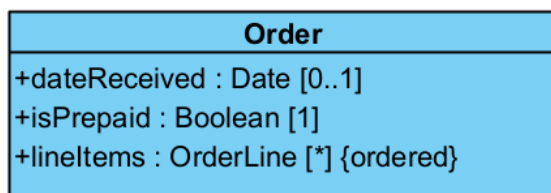
De rangen van hoog naar laag zijn: maarschalk, generaal, kolonel, majoor, kapitein, luitenant, sergeant, mineur, verkenner en spion. Als de spion de maarschalk aanvalt wint de spion. Valt de maarschalk echter de spion aan, dan wint de maarschalk.

Maak een klassendiagram voor de speelstukken van Stratego.

8. Klasse Order: attributen weergeven als associaties

Een attribuut is soms ook goed als een associatie weer te geven in een klassendiagram.

a) Zet onderstaande klasse om in vier klassen met associaties waarbij er geen informatie verloren mag gaan.

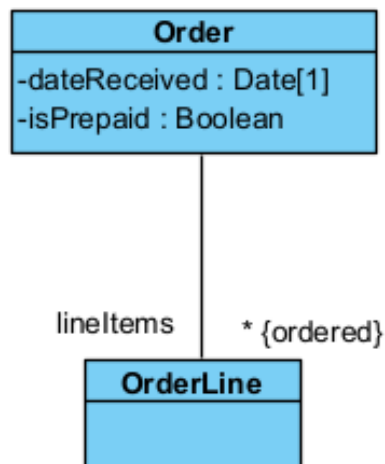


Opmerking. Overgenomen uit *UML Distilled* (p. 37) door Martin Fowler. Pearson Education (third edition). Copyright 2004, Pearson Education, Inc.

b) In de omzetting naar vier Klassen zijn we gezien de context iets te ver doorgeslagen in het weergeven van attributen in klassen. Wat zou een passende uitwerking zijn?

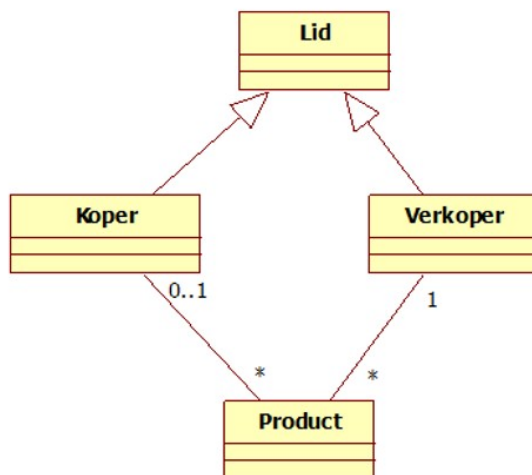
9. Associatie Order - OrderLine: aggregatie of compositie?

In onderstaand klassendiagram is een associatie tussen Order en OrderLine getekend. Zou je hier ook een aggregatie of compositie van kunnen maken?



10. Lid in rol van Koper en Verkoper

Een opdrachtgever wil een webapplicatie laten maken waar leden producten kunnen aanbieden en/of kopen. De opdrachtgever wil hiervoor scrum inzetten. In een eerste gesprek is onderstaande initiële klassendiagram gemaakt.



De product owner vindt dat dit te veel suggereert dat een lid óf verkoper óf koper is. Het is echter de bedoeling dat een lid zowel producten kan verkopen als kopen. Probeer dit klassendiagram eens zonder overerving te realiseren, terwijl je toch aangeeft dat een lid producten kan kopen en verkopen.

11. Domeinmodel vliegtuigvluchten

Onderstaande opdracht is gebaseerd op een klassendiagram uit het boek Praktisch UML van Jos Warmer en Anneke Kleppe (Jos Warmer en Anneke Kleppe, 2011, p. 80)

Maak op basis van de volgende beschrijving een bijpassend klassendiagram.

Het betreft een systeem voor het reserveren en bemannen van vliegtuigvluchten.

Een vlucht heeft een vluchtnummer (bestaande uit letters en cijfers), vertrektijd en een aankomsttijd. De duur van de vlucht is af te leiden van de vertrek- en aankomsttijd. Van een vlucht is het vertrekpunt (naam van de Luchthaven) en de bestemming (naam van de luchthaven) bekend.

Voor iedere vlucht worden de passagiers geregistreerd. De volgorde van aanmelding wordt vastgelegd. Van de passagiers worden de naam en geboortedatum vastgelegd.

Een vliegtuig heeft een naam en een capaciteit (aantal beschikbare plaatsen). Van het vliegtuig is ook de naam van de luchtvaartmaatschappij bekend.

Voor iedere uit te voeren vlucht wordt de bemanning vastgelegd. Een bemanningslid heeft een naam en een functie. We onderscheiden de functies piloot en stewardess.

12. Theater reserveringssysteem: domeinmodel

Voor het vak Server is een oefententamen beschikbaar dat gaat over het reserveren van stoelen in een theater. De studenten krijgen de opdracht om op basis van een specificatie een applicatie te bouwen. Deze applicatie gaan we nu gebruiken om te oefenen met het domeinmodel. De docent kan een werkende applicatie demonstreren of je kunt het filmpje bekijken dat in de folder 'aanvullend materiaal opdrachten' vinden is.

De functionaliteit is nog zeer beperkt. Zo ligt het aantal stoelen en de aanduiding (A1 t/m E6) daarvan vast, en kan er slechts één voorstelling in het vooruit gereserveerd worden.

Een Klant heeft de attributen:

- Naam (string) [eis: minimaal 2 karakters]
- Adres (string) [eis: verplicht]
- Woonplaats (string) [eis: verplicht]
- Email (string) [eis: geldig emailadres]

Opdracht: Maak het bijbehorende domeinmodel. Voor alle relaties willen we de multipliciteit zien. Voor de attributen is het in deze opdracht niet nodig om het type aan te geven als dit evident is. Wel willen we dat je eventuele constraints opneemt.

13. Theater reserveringssysteem uitbreiding: domeinmodel

De applicatie uit voorgaande opdracht heeft nog zeer beperkte functionaliteit. Deze gaan we uitbreiden.

- a. Neem een voorstelling erin op. Een voorstelling heeft een datum en een naam.
- b. In plaats van één zaal, maken we het aantal zalen onbeperkt.
- c. Per zaal is het aantal stoelen variabel. Als aanduiding voor een rij gebruiken we nog steeds een letter (dus maximaal 26 rijen). Daarbinnen heeft een stoel een nummer.
- d. Een voorstelling wordt altijd gekoppeld aan een zaal.
- e. Een klant kan voor meerdere voorstellingen een reservering vastleggen.

14. Poor man's console MVC applicatie: software klassen

Op <https://dzone.com/articles/poor-mans-console-mvc> vind je de beschrijving van een 'Poor man's console MVC' applicatie. Ook wordt een fictieve beschrijving gegeven van de totstandkoming van deze applicatie. De code kun je downloaden.

Deze applicatie gaan we gebruiken voor oefenen met een use case diagram, klassendiagram en sequentiediagram. In deze les beginnen we met een klassendiagram. In de volgende lessen komen het use case diagram en sequentie diagram aan bod.

Voorbeelden van invoer om de applicatie uit te proberen:

- add TUESDAY 13:45 paul hond,kat
- next
- add FRIDAY 8:05 frits vlieg,mug
- previous

Maak voor de applicatie een klassendiagram en houd je daarbij aan het volgende:

- Maak het diagram m.b.v. Visual Paradigm. We gaan het klassendiagram in een van de volgende lessen gebruiken als basis voor het schrijven van sequentiediagrammen voor deze applicatie.
- Maak gebruik van packages.
- Evidente attributen en evidente operaties laat je achterwege (probeer zelf in te schatten wat hier evident is)
- Neem in principe attribuut- en parametertypes op. Blijf echter pragmatisch Het kan bijvoorbeeld voorkomen dat een operatie dan wel erg veel ruimte vraagt in het diagram en de typering misschien niet zoveel toevoegt. Overweeg bijvoorbeeld
addVisit(dayOfWeek, time, ownerName, petNames) i.p.v.
addVisit(dayOfWeek: DayOfWeek, time : LocalTime, ownerName : string,
petNames : string[]) : void
- Laat zien of een associatie private of public is.
- Toon zowel de compartimenten van attributen als operaties, ook als deze leeg zijn.

15. Tips

1. Tip voor opgave 5b: maak gebruik van een 'derived attribute'.