

Eindtoets

INTRODUCTIE

Deze eindtoets is bedoeld als voorbereiding op het tentamen. Het is belangrijk dat u de eindtoets pas probeert te maken op het moment dat u denkt klaar te zijn met de tentamenvoorbereiding. Heeft u over dat laatste nog twijfels, bekijk dan nog eens de leerdoelen van de diverse hoofdstukken om te ontdekken welke onderdelen u nog onvoldoende beheerst.

Terugkoppeling

Mogelijke uitwerkingen van de opgaven staan in de terugkoppeling. We willen benadrukken dat u het meest leert als u eerst de opgaven maakt en pas daarna de antwoorden controleert.

Beoordeling

Het aantal punten dat u per opgave kunt behalen, staat bij die opgave vermeld. U kunt in totaal 100 punten behalen. Voor een voldoende voor het tentamen moet u tenminste 55 punten behalen.

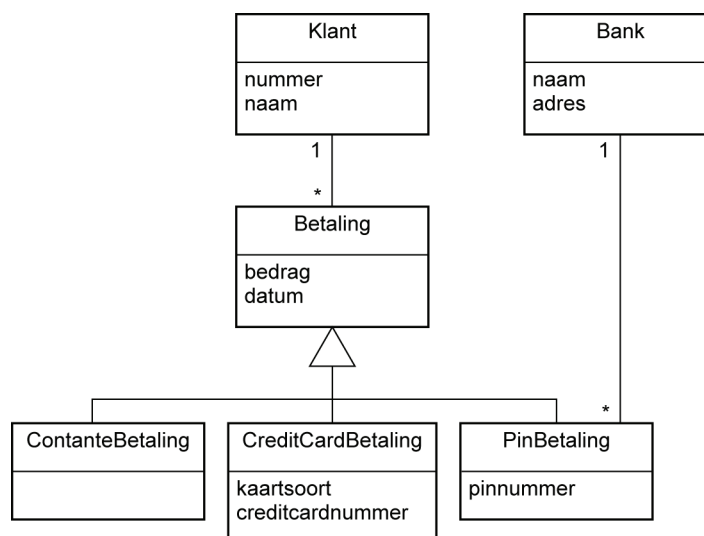
Studeeraanwijzingen

De studielast van deze eindtoets is ongeveer 4 uur, inclusief het nakijken van de opgaven aan de hand van de terugkoppeling.

Opgaven

10 punten

- Gegeven is het domeinmodel van figuur 1.



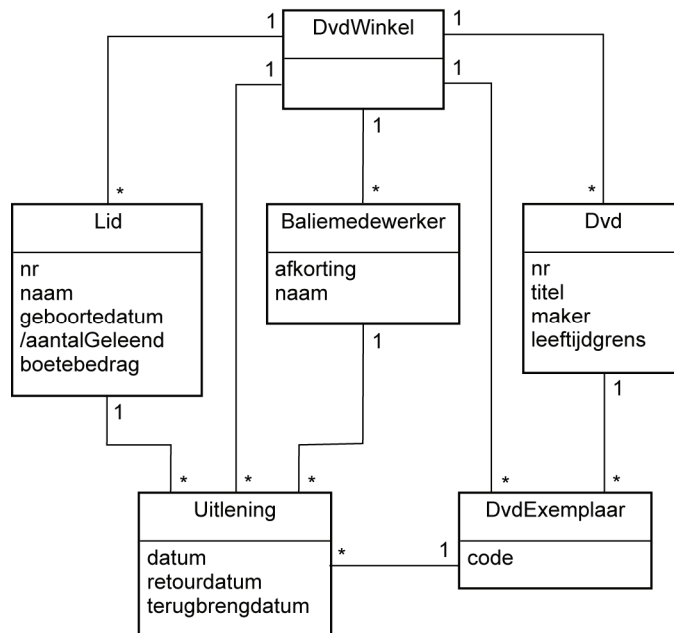
FIGUUR 1 Domeinmodel voor betalingen

We willen alle klassen persistent maken. Daartoe moet het domeinmodel omgezet worden naar een equivalente structuur in een relationele database. Geef de structuur weer in een strokendiagram. Daarin moet u ook duidelijk eventuele primaire sleutels en verwijssleutels aangeven.

- 2 Een huwelijksbureau wil een (web)applicatie laten maken waarmee een aanstaand bruidspaar gasten kan laten kiezen uit een lijst met wensen. De gasten kunnen via de applicatie een of meerdere cadeaus uitzoeken en aangeven hoeveel exemplaren ze willen geven. Andere gasten kunnen deze cadeaus, als alle exemplaren al gegeven worden, dan niet meer kiezen. Op deze manier wordt bereikt dat gasten weten wat de wensen van het bruidspaar zijn en krijgt het bruidspaar geen dubbele cadeaus. Via een formulier meldt een aanvrager zich aan voor de service van het huwelijksbureau. Het formulier bevat de gegevens van de aanvrager (naam en adres), een lijst met cadeaus, een lijst met gasten en de datum van het huwelijk. Een aanvrager kan meerdere aanvragen doen (voor verschillende bruidsparen uiteraard). Per gast is de naam en het e-mailadres bekend. Per cadeau wordt een naam, een omschrijving, een gewenst aantal en een schatting van de prijs per stuk gegeven. Een medewerker voert de gegevens van de aanmelding in waarbij ook de datum van invoer wordt vastgelegd. Het systeem slaat de gegevens op en kent aan iedere gast een wachtwoord toe. Daarna worden de e-mailadressen en wachtwoorden doorgestuurd naar een extern e-mailsysteem dat iedere betrokkene een e-mail stuurt met een link naar de applicatie en het wachtwoord. (Het bruidspaar krijgt uiteraard geen toegang tot de applicatie). Een gast kan via de applicatie cadeaus selecteren of kan de door hem reeds geselecteerde cadeaus opvragen. (We zien voor het gemak af van de mogelijkheid om keuzes te wijzigen). Na inloggen krijgt de gast bij de eerste mogelijkheid een lijst met mogelijke cadeaus te zien waaruit gekozen kan worden met bijbehorend aantal. Bij een keuze voor een cadeau wordt ook het aantal vastgelegd (Zo kan een gast bijvoorbeeld kiezen om twee van de zes gevraagde stoelen te geven; in dit geval zal een volgende gast vier als aantal zien). De gast kan verschillende keuzes maken. Een gast kan bij de tweede mogelijkheid na inloggen de door hem gemaakt keuzes opvragen. Een dag na de bruiloft wordt door het systeem automatisch de lijst met cadeaus en gasten verwijderd en wordt aan een financieel systeem de benodigde gegevens verstrekt (duur service, aantal gasten en aantal cadeaus) opdat het bruidspaar een rekening krijgt voor het gebruik van de service.

- 5 punten a Welke actoren kunt u in de beschrijving onderkennen? Geef per actor aan of het een primaire of ondersteunende actor is.
- 5 punten b Teken een use-casediagram.
- 15 punten c Stel een domeinmodel op dat hoort bij de casusbeschrijvingen. Maak hierbij gebruik van een klasse Datum met attributen dag, maand en jaar.
- 10 punten d Geef de use-casebeschrijving waarin gasten cadeaus kiezen (eerste gebruiksmogelijkheid) in volledig formaat tot en met het hoofdsucces scenario. U mag het onderdeel belanghebbenden met hun belangen weglaten.

- 3 In deze opgave gebruiken we een vereenvoudigde versie van de dvd-winkel. Het domeinmodel ziet u in figuur 2. Hierbij is geen gebruik gemaakt van de klasse Uitleenregel zoals in het werkboek. De klasse Uitleening representeert hier de uitlening van één dvd, en heeft drie attributen: datum is de datum dat de dvd is geleend, retourdatum is de datum dat het geleende exemplaar teruggebracht moet worden, en terugbrengdatum is de datum dat het geleende exemplaar daadwerkelijk is teruggebracht.



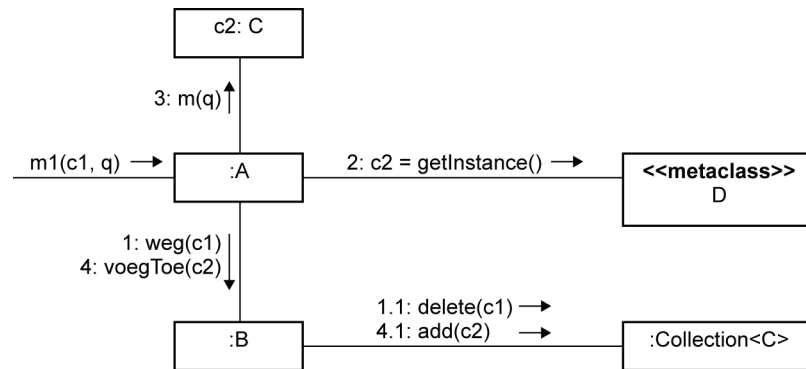
FIGUUR 2 Vereenvoudigd domeinmodel voor de dvd-casus

- 15 punten a Een van de mogelijke use cases betreft het verlengen van een uitlening. Het SSD dat bij het hoofdsuccesscenario van deze use case hoort, bevat een systeemaanroep `verleng(medewerkernaam, dvdcode, aantaldagen)`. Het resultaat moet zijn dat de retourdatum van de betreffende uitlening met aantaldagen opgehoogd moet zijn. Ook moet de uitlening nu gekoppeld zijn aan de medewerker die de verlenging afhandelt. Ontwerp hiervoor het communicatiediagram. Kort de systeemaanroep af met `verleng(naam, code, dagen)`.
- 3 punten b Volgens het domeinmodel van figuur 2 kunnen bij één dvd-exemplaar verschillende uitleningen horen. Dit betekent dat de historie van uitleningen bewaard blijft. Stel dat de historie van uitleningen niet bewaard wordt. Noem de wijziging(en) die daarvoor nodig zijn in het domeinmodel.
- 12 punten c We gaan uit van het in opdracht b gewijzigde domeinmodel, dus de historie van uitleningen blijft niet bewaard. In het hoofdsuccesscenario van de use case voor het terugbrengen van een dvd komt in een SSD de systeemaanroep `retour(code)` voor. Het resultaat moet zijn dat de bijbehorende uitlening uit het systeem verwijderd wordt en het aantal geleende exemplaren bij Lid met 1 wordt verlaagd. Ontwerp het bijbehorende communicatiediagram.

- 4 Gegeven is het communicatiediagram van figuur 3. In deze figuur is parameter q van het type Q.

15 punten

- a Teken het bijbehorende ontwerpklassendiagram. Per klasse dienen alleen de methoden (zonder parameters) aangegeven te worden. Neem ook de navigeerbaarheid en de minimale multipliciteit van de associaties op.



FIGUUR 3 Communicatiediagram bij opgave 4

5 punten

- b Geef de volgorde waarin de verschillende klassen geïmplementeerd en getest kunnen worden.

5 punten

- c Geef een implementatie voor klasse A.

TERUGKOPPELING

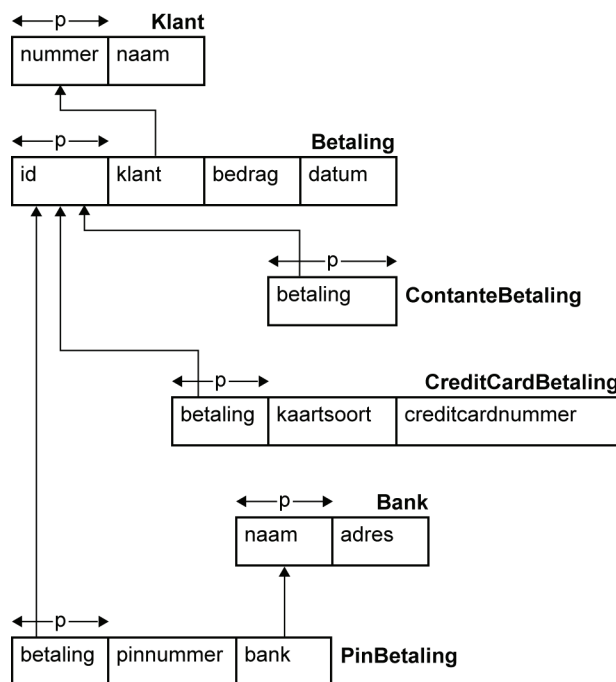
Uitwerking van de opgaven

- 1 Er zijn verschillende goede antwoorden mogelijk. Wanneer er voor gekozen wordt om alle betalingen in één tabel te stoppen, dan krijgen we het strokendiagram van figuur 4.



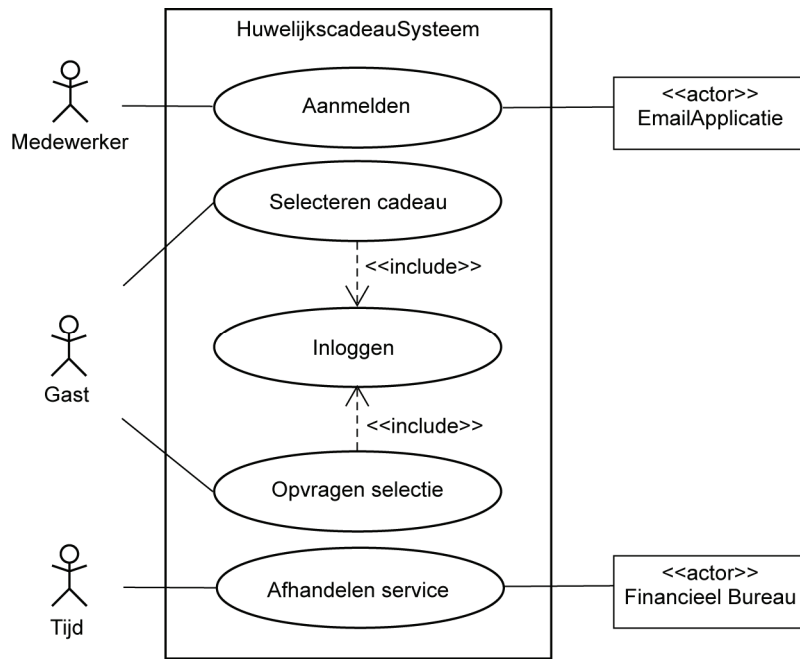
FIGUUR 4 Strokendiagram, oplossing 1

Figuur 5 toont een oplossing waarbij voor ieder soort betaling een eigen tabel wordt gebruikt.



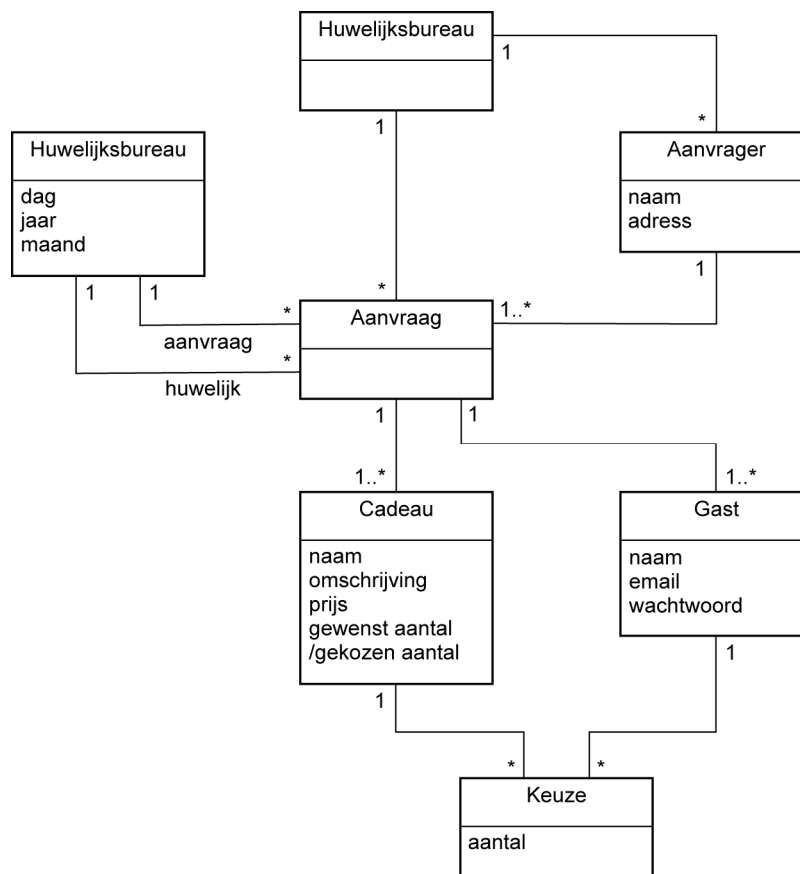
FIGUUR 5 Strokendiagram, oplossing 2

- 2
 - a Er zijn 3 primaire actoren (medewerker, gast, en tijd) en twee ondersteunende actoren (e-mailapplicatie en financieel bureau).
 - b Zie figuur 6.



FIGUUR 6 Use-casediagram voor huwelijksbureau

c Zie figuur 7.



FIGUUR 7 Domeinmodel voor huwelijksbureau

d De use case kan er als volgt uitzien:

Use case: Selecteren van cadeaus

Primaire actor: Gast

Precondities: Gast is aangemeld en geauthenticeerd bij het systeem: Include Login.

Postcondities: De selectie van cadeaus met gewenst aantallen is vastgelegd.

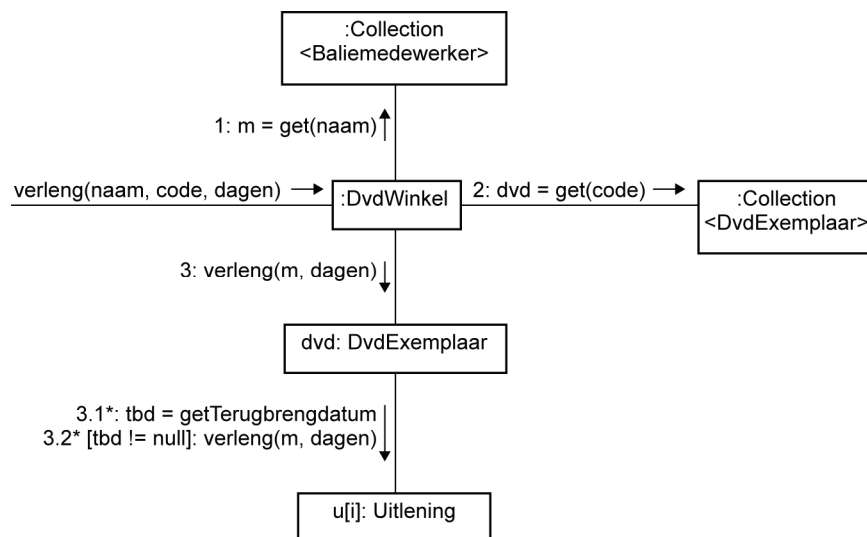
Hoofdsuccessscenario:

1. Gast start een nieuwe selectie.
2. Systeem toont de te selecteren cadeaus.
3. Gast selecteert een van de cadeaus en voert het gewenste aantal in.
4. Systeem valideert het gekozen cadeau en het gewenste aantal en slaat gegevens van deze keuze op.

Stappen 2 tot en met 4 worden herhaald totdat gast klaar is.

5. Gast stopt selectie.
-

- 3 a Zie figuur 8. Bericht 3.1 wordt aangeroepen op alle uitleningen van het exemplaar, en haalt de terugbrengdatum (tbd) op. Wanneer deze nog niet ingevuld is, dan hebben we te maken met de huidige uitlening van het dvd-exemplaar, en wordt de desbetreffende uitlening verlengd door bericht 3.2. Dit bericht verhoogt de retourdatum met parameter dagen en vervangt de met de uitlening geassocieerde baliemedewerker.



FIGUUR 8 Communicatiediagram voor verlengen uitlening van dvd

Dit is niet de enige oplossing. U kunt bijvoorbeeld ook bij het dvd-exemplaar een deelverzameling (met de select-opdracht) maken van de uitleningen met een lege terugbrengdatum en bericht verleng naar alle elementen van deze deelverzameling sturen.

b De kardinaliteit van de associatie van DvdExemplaar naar Uitlening moet aan de kant van Uitlening veranderd worden van * naar 0..1.

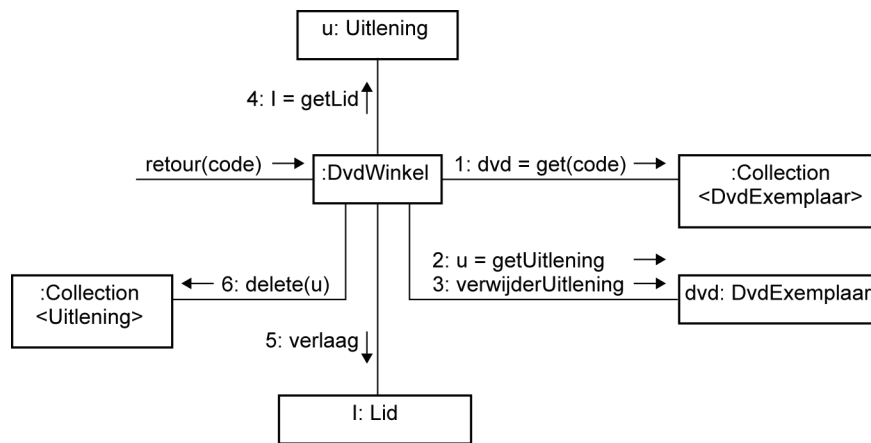
Wanneer een exemplaar niet is uitgeleend, dan is de waarde 0, is het exemplaar uitgeleend, dan is de waarde gelijk aan 1.

Bovendien kan de terugbrengdatum worden weggelaten: wanneer de dvd is teruggebracht wordt de uitlening toch verwijderd.

c Zie figuur 9. Bericht 2 vraagt de huidige uitlening van de dvd op.

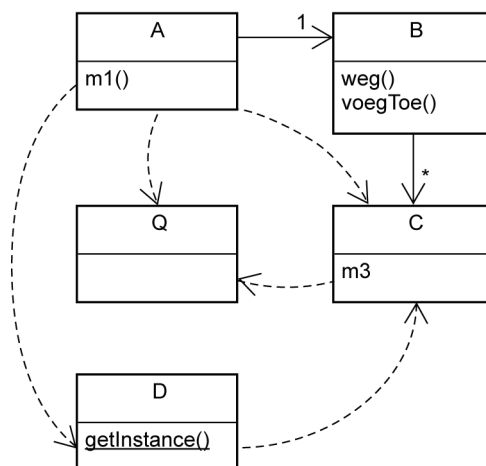
Bericht 3 verwijdert de link van het exemplaar naar de huidige uitlening (in Java zou het attribuut op null gezet worden). Dan wordt het lid bij de uitlening opgevraagd en het aantal geleende dvd's verlaagd (berichten 4 en 5). Als laatste wordt de uitlening uit de collectie met uitleningen van de dvd-winkel verwijderd.

Als u ook de uitlening verwijdert uit de collecties van Lid en Baliemedewerker is dat goed maar niet noodzakelijk omdat we voor deze use case ook lid en medewerker als attributen van een uitlening kunnen zien.



FIGUUR 9 Communicatiediagram voor terugbrengen van dvd

4 a Zie figuur 10.



FIGUUR 10 Klassendiagram

Toelichting:

- m1(c1,q): A heeft een methode m1 en parameter zichtbaarheid op C en Q.
- 1: weg(c1): B heeft methode weg en parameter zichtbaarheid op C.
- 2: getInstance wordt aangeroepen op klasse D, is dus een klassenmethode. Om deze te kunnen aanroepen is A afhankelijk van D. D levert een instantie van C, dus D is afhankelijk van C. A krijgt een instantie van C als terugkeerwaarde, dus A is afhankelijk van C maar die was er al
- 3:m3(q): C moet een methode m3 hebben en parameter zichtbaarheid op Q
- 1.1 en 4.1 leiden niet tot nieuwe methoden maar B moet een collectie van type C hebben, dus een permanente zichtbaarheid op C met multipliciteit *. Hierdoor verdwijnt de tijdelijke zichtbaarheid van B op C
- b De volgorde waarin de klassen geïmplementeerd en getest kunnen worden is:
 - Q
 - C
 - B en D (kunnen tegelijkertijd geïmplementeerd worden)
 - A
- c Een implementatie van klasse A wordt hieronder gegeven.

```
public class A {
    B b; // attribuut implementeert associatie van A naar B

    public void m1(C c1, Q q) {
        b.weg(c1);
        C c2 = D.getInstance();
        c2.m3(q);
        b.voegToe(c2);
    }
}
```