



OOSE-OOAD

Object Oriented Analysis and Design



There are two ways of constructing a software design: One way is to make it so simple that there are obviously no deficiencies, and the other way is to make it so complicated that there are no obvious deficiencies. The first method is far more difficult.

--C.A.R. Hoare



Algemene gegevens

Titel	Object Oriented Analysis and Design (OOAD)
Studiejaar	2018-2019
Opleiding	HBO-ICT, profielen Software Development, Data Solutions Development, Web Development
Auteurs	Marco Engelbart

Deze studiehandleiding is een samenvatting van de OWE-beschrijving OOAD, onderdeel van het Opleidingsstatuut ICA 2018-2019, deel 3B van de opleiding HBO-ICT.

Aan deze handleiding kunnen geen rechten worden ontleend. Het Opleidingsstatuut, inclusief bijlagen, is te vinden op HAN Insite.



INHOUDSOPGAVE

1	UIT DE PRAKTIJK	1
2	PRAKTISCHE INFORMATIE.....	2
2.1	Studiepunten, studiebelasting, docententeam	2
2.2	Inhoud en onderdelen	2
2.3	Competenties	2
2.4	Beoordelingscriteria.....	3
2.5	Deeltentamens	5
2.6	Inleveren & Archivering	6
2.7	Middelen	6
3	Overige documenten	7



Object Oriented Analysis and Design

1 UIT DE PRAKTIJK

"Programming is fun, but developing quality software is hard. In between the nice ideas, the requirements or the 'vision', and a working software product, there is much more than programming.

Analysis and design -- defining how to solve the problem, what to program, capturing this design in ways that are easy to communicate, to review, to implement, and to evolve ..."

Philippe Kruchten

Uit de bovenstaande uitspraak van Phillippe Kruchten, een vooraanstaand software engineer, valt op te maken dat het ontwikkelen van een software product meer is dan programmeren. Een belangrijk deel van het werk van een softwareontwikkelaar bestaat dan ook uit het (in teamverband) analyseren van het probleem en het in kaart brengen van mogelijke oplossingen.

Ontwikkelteams in kamers met whiteboards met daarop schetsen van use case diagrams, domain models, sequence diagrams, class diagrams of andere UML-diagrammen. Samen brainstormen en discussiëren, uitvegen en verbeteren. Dat is de praktijk van softwareontwerp!

Hieronder zie je foto's van deze manier van werken. Deze foto's zijn afkomstig van de website www.craiglarman.com/wiki/index.php?title=Agile_Modeling_Pictures.





2 PRAKTISCHE INFORMATIE

2.1 Studiepunten, studiebelasting, docententeam

Deze OWE heeft een omvang van 7,5 studiepunten (sp). Dit komt neer op een studiebelasting van 20 uur per week.

Bij de uitvoering zijn de volgende docenten betrokken: Marco Engelbart, Yakup Küçük, Ben Nieuwhof en Herman Telman.

2.2 Inhoud en onderdelen

Deze course richt zich op het analyseren en het ontwerpen van een gedistribueerde OO applicatie met behulp van notatietechnieken uit UML (Unified Modeling Language) en het verantwoorden van ontwerpkeuzes.

De belangrijkste onderwerpen zijn:

- Specificeren en classificeren van functionele en niet-functionele requirements;
- Opstellen van een use case model en use case scenarios;
- Opstellen van een domeinmodel op basis van een domeinbeschrijving;
- Maken van sequence diagrams van geïdentificeerde systeemoperaties;
- Maken van een design class diagram op grond van domeinmodel en sequence diagrams;
- Maken van activity diagrams en state machine diagrams voor systeem- en objectgedrag;
- Toepassen van softwareontwerpprincipes (abstractie, koppeling en cohesie, decompositie en modulariteit, encapsulatie en information hiding) en design patterns;
- Vertalen van een OO ontwerp naar een plan voor de implementatie in een OO-taal;
- Toepassen van objectgeoriënteerde analyse en ontwerp bij een waterval- en een iteratieve aanpak.

2.3 Competenties

OOAD-1. De student analyseert een concreet probleem in de reële wereld op basis van relevante documenten, en levert als resultaat van deze analyse een requirementsspecificatie op voor een systeem dat een oplossing biedt voor het probleem.

OOAD-2. De student kan een concreet probleem in de reële wereld analyseren op basis van relevante documenten, en als resultaat van deze analyse concepten in de reële wereld beschrijven middels een domeinmodel waarin die concepten, verbanden tussen die concepten, en eigenschappen van die concepten vastgelegd zijn.

OOAD-3. De student modelleert het gewenste interactiegedrag van objecten m.b.v. sequence diagrams, die consistent zijn met de use case scenarios en het domeinmodel.

OOAD-4. De student modelleert het systeem- en objectgedrag m.b.v. activity diagrams en state machine diagrams, die consistent zijn met de use case scenario's en het domeinmodel.



Object Oriented Analysis and Design

OOAD-5. De student ontwerpt de programmeertaalafhankelijke softwarestructuur m.b.v. class diagrams, die consistent zijn met de opgestelde sequence diagrams.)

OOAD-6. De student past softwareontwerpprincipes (abstractie, koppeling en cohesie, decompositie en modulariteit, encapsulatie en information hiding) en design patterns toe bij het opstellen van een OO ontwerp.

OOAD-7. De student vertaalt een OO ontwerp naar (initiële) code voor een OO programmeertaal.

OOAD-8. De student beschrijft de OO analyse en ontwerp geïntegreerd in een document dat qua taal en vorm correct is en waarin de gemaakte keuzes worden verantwoord.

2.4 Beoordelingscriteria

Competentie	Beoordelingscriteria	Deeltentamens
OOAD-1	<ul style="list-style-type: none">De functionele eisen van het te ontwikkelen systeem zijn beschreven in termen van op user goals gebaseerde use cases, m.b.v. een use case diagram en bijbehorende brief descriptions van de use cases.De aanvullende eisen zijn ingedeeld volgens FURPS+ en eenduidig, begrijpelijk en testbaar geformuleerd.De fully dressed use cases zijn opgesteld in een twee-kolomsformaat, en geven de actor aan, pre- en postcondities, hebben een korte beschrijving, een main success scenario, en bevatten alle alternative flows die onderscheiden kunnen worden, behalve eenvoudige foutmeldingen. Elke use case is UI-onafhankelijk en black-box geformuleerd.	Case study OOAD
OOAD-2	<ul style="list-style-type: none">Gegeven een concreet probleem worden de concepten beschreven in een volledig (implementatie-)platformonafhankelijk UML class diagram, met correcte classes, associaties en multipliciteiten. Generalisatie/specialisatie is toegepast waar mogelijk en zinvol.Het domeinmodel bevat geen ontwerpbeslissingen.De betekenis van de domeinconcepten is verduidelijkt in een glossary.Conceptuele classes, associaties en multipliciteiten zijn op inzichtelijke wijze bepaald.De correctheid en compleetheid van het domeinmodel wordt gemotiveerd.	Toets UML Case study OOAD
OOAD-3	<ul style="list-style-type: none">De sequence diagrams behorende bij een use case realiseren, tezamen uitgaande van de precondition van die use case de postconditie van die use case, waarbij ten minste wordt vastgelegd wat er op instantieniveau in het domeinmodel wordt gewijzigd.	Toets UML Toets OO Design Case study OOAD



Object Oriented Analysis and Design

OOAD-4	<ul style="list-style-type: none">• De activity diagrams beschrijven alle relevante aspecten van de flow of control en de object flow van het te modelleren systeem.• De state diagrams beschrijven alle relevante systeem- en objecttoestanden en bijbehorende toestandsovergangen van het te modelleren systeem.• De activity diagrams en state diagrams zijn consistent met de use cases.	Toets UML Case study OOAD
OOAD-5	<ul style="list-style-type: none">• In het class diagram zijn alle operaties en attributen met hun visibility aangegeven, en er wordt getoond welke rollen van een associatie navigeerbaar zijn.• Het class diagram is consistent met domeinmodel, sequence diagrams, activity diagrams en state diagrams.	Toets UML Toets OO Design Case study OOAD
OOAD-6	<ul style="list-style-type: none">• In de sequence diagrams en het class diagram zijn op een bewuste manier keuzes gemaakt die bijdragen aan hoge cohesie en lage koppeling.• Een ontwerp waarin te hoge koppeling en/of te lage cohesie optreedt is verbeterd door het toepassen van een Design Pattern.• Alternatieve ontwerpen zijn opgesteld, uiteindelijke keuzes zijn goed onderbouwd.	Toets OO design Case study OOAD
OOAD-7	<ul style="list-style-type: none">• De initiële code voor een specifieke OO programmeertaal is consistent met sequence diagrams en design class diagram.• Een prototype van de OO applicatie is gebruikt om de kwaliteit van het ontwerp te testen.	Case study OOAD
OOAD-8	<ul style="list-style-type: none">• Het ontwerpdocument heeft een duidelijke en logische structuur.• Past grammaticaal correct taalgebruik toe en hanteert de juiste spelling.• De UML-diagrammen zijn conform de standaard.	Case study OOAD



Object Oriented Analysis and Design

2.5 Deeltentamens

Toets UML	
Code tentamen	S_UML
Naam deeltentamen	Toets UML
Deeltentamenvorm	Schriftelijk tentamen
Deeltentamenbeschrijving	Toets UML, met open vragen
Aantal examinatoren	1
Beoordeling	Individueel cijfer
Minimaal resultaat	5.5
Weging (%)	30
Deeltentamenmoment	Zie toetsrooster in i-SAS

Toets OO Design	
Code tentamen	S_OODesign
Naam deeltentamen	Toets OO Design
Deeltentamenvorm	Schriftelijk tentamen
Deeltentamenbeschrijving	Toets OO Design, met open vragen
Aantal examinatoren	1
Beoordeling	Individueel cijfer
Minimaal resultaat	5.5
Weging (%)	30
Deeltentamenmoment	Zie toetsrooster in i-SAS

Case Study OOAD	
Code tentamen	O_CaseStudy
Naam deeltentamen	Case Study
Deeltentamenvorm	Opdracht
Deeltentamenbeschrijving	Case Study OOAD, Analyse- en ontwerpdocument
Aantal examinatoren	1
Beoordeling	Individueel cijfer
Minimaal resultaat	5.5
Weging (%)	40
Deeltentamenmoment	Zie toetsrooster in i-SAS



2.6 Inleveren & Archivering

Het beroepsproduct van het deeltentamen **Case Study OOAD** dient via iSAS te worden ingeleverd. Deze case study is groepswork. Elk groepslid dient de volledige uitwerking van de groep te uploaden via iSAS.

2.7 Middelen

Literatuur

- Verplicht: Craig Larman, Applying UML and Patterns : An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition), Prentice Hall, ISBN: 0131489062

Overig lesmateriaal

- Studiehandleiding course OOSE-OOAD (digitaal)
- Werkboek course OOSE-OOAD (digitaal)
- Powerpoint-presentaties course OOSE-OOAD (digitaal)
- PluralSight

Software

UML-tool: Astah Professional



3 Overige documenten

Voor alle andere documenten kan je terecht op OnderwijsOnline, de onderwijseenheid I-OOSE OOAD 18/19.