# Week 12-15: Een systeembeschrijving maken en implementeren

De opgaven 1 t/m 4 zijn oefenopgaven. Met deze opgaven kun je oefenen. Je kunt de docent feedback vragen op je uitwerking van deze opgaven.

**Opgave 5 is een toetsopgave. Deze moet je inleveren.** De opgave wordt beoordeeld. Je moet de uitwerking mondeling verdedigen. Tijdens de verdediging krijg je feedback op de uitwerking. Er wordt verwacht dat je de feedback gebruikt voor het verbeteren van volgende opgaven.

## Leermiddelen

Heb je hulp nodig bij een onderwerp van deze week, dan bieden onderstaande links hulp.  
(klik op de tekst met ctrl+click om de links te openen)

* De sheets
* Boek: Praktisch UML, hoofdstuk 4.6: Werkwijze om tot een klassediagram te komen.
* Boek: Praktisch UML, hoofdstuk 8: Het use case diagram

## Opgave 1

Maak een use case diagram voor de volgende casus:

Een software bedrijf krijgt de opdracht van de bank MoneyAreUs om de software voor een geldautomaat te maken. De geldautomaat biedt klanten de mogelijkheid met een pasje en een pincode het saldo op te vragen of geld op te nemen. Het spaartegoed moet ook opgevraagd kunnen worde. En geld moet kunnen worden overgemaakt van de spaarrekening naar de betaalrekening.

Als het pasje niet van de MoneyAreUs bank is, wordt toegang geweigerd. Als het pasje wel van de MoneyAreUs bank is, worden de gegevens gelezen en wordt de pincode gevraagd. Indien de juiste pincode niet na drie pogingen niet is ingegeven, wordt het pasje door de machine ingenomen. Het invoeren van de pincode kan geannuleerd worden met de annuleerknop.  
  
Nadat de klant (met pas en pincode) toegang heeft verkregen, verschijnt een welkomstmenu. De klant kan kiezen voor geld opnemen of saldo opvragen.

Indien voor ‘geld opnemen’ wordt gekozen, geeft de gebruiker het gewenste bedrag in en bevestigt zijn keuze. Het maximum op te nemen bedrag is €500 per dag. Als er niet genoeg contant geld meer in de automaat is, wordt het nog aanwezige bedrag getoond. Als er geen geld meer in de automaat zit, wordt dit in het welkoms menu getoond en kan geld opnemen niet gekozen worden.

Er kan naar keuze een bonnetje van de transactie worden afgedrukt. Als het papier op is, wordt dit gemeld en kan niet worden afgedrukt.

De automaat heeft een sensor waardoor gezien kan worden of het geld is uitgenomen. Er is geen sensor om te zien of het pasje na teruggave is uitgenomen.

Na uitnemen van het geld, wordt de klant automatisch afgemeld, waarna het pasje wordt teruggegeven.

Kiest de klant voor het opvragen van het saldo, dan wordt het saldo getoond en kan de gebruiker eventueel een bonnetje laten afdrukken met de saldogegevens. De klant kan ervoor kiezen om geld op te nemen of zich direct afmelden. Na af te melden wordt het pasje teruggegeven.

## Opgave 2

Maak een use case diagram voor een reisbureau dat vliegreizen verkoopt aan klanten. Of maak een use case diagram voor BOL.COM, voor het online winkelen en kopen van producten door klanten.

## Opgave 3

Kies één van de twee casussen en maak de bijbehorende opdracht.

**casus 1:**In week 1, opgave 3 t/m 6, heb je een programma gemaakt voor het administreren en bevragen van resultaten van studenten. Maak voor dit programma een klassediagram.

Zie je, na het maken van het klassediagram voor dit programma, ruimte voor verbetering van de programmastructuur. Dan mag je die zeker doorvoeren in het klassediagram.

**casus 2:**

Maak een klassediagram voor de volgende casus:

*Een webshop verkoopt spellen. Een klant van de webshop kan spellen bestellen door een bestelling te maken en daar spellen aan toe te voegen die hij wil ontvangen. De klant kan zich ook bedenken en spellen uit de bestelling halen. Er zijn spellen in de categorie RPG, Platform en Shooter en van elk spel is bekend hoeveel er op voorraad zijn. Een spel heeft naast een naam, uitgever en release datum ook een leeftijdscategorie. De klant kan alleen spellen bestellen als zijn leeftijd binnen de leeftijdscategorie van het spel valt. Bij elke bestelling hoort een factuur waarin o.a. wordt vastgelegd of de betaling voldaan is en wat het factuuradres is.*

## Opgave 4

Voor een catalogussysteem van een museum moet een applicatie ontworpen worden.   
Het museum beheert kunstwerken. Informatie over de kunstwerken en de maker ervan wil men vastgelegd hebben in een catalogus. Elk kunstwerk is gemaakt door een artiest. Naast de artiest is van elk kunstwerk ook de naam van het kunstwerk en de kunststroming waartoe het kunstwerk behoort bekend (impressionisme, barok, art deco, realisme, etc). En men legt de datering (datum waarop het kunstwerk gemaakt is) vast. Van een artiest is de naam bekend en wil men altijd weten wat de geboorte en sterfdag is. En wat het land van herkomst van de artist is. Van een artiest moeten alle kunstwerken opgevraagd kunnen worden. Van schilderijen moet de verfsoort en de afmeting vastgelegd worden en van beelden de materiaalsoort en het gewicht.

Het museum wil op basis van de catalogusinhoud ook kunnen bepalen welke kunstwerken er aan de bezoekers getoond worden en wanneer. Met dit doel stelt het museum collecties van kunstwerken samen. Elke collectie heeft een naam en is een bepaalde periode te bezichtigen in het museum. Een collectie kan ook uitgeleend zijn aan een ander museum. Deze kan dan niet getoond worden en is dus niet beschikbaar.

Maak een klassendiagram voor de gegeven casus.

## Opgave 5 Deze opgave moet je inleveren en wordt beoordeeld.

Verzin zelf een leuke applicatie waar je alle PRC2 leerdoelen in kunt passen. Beschrijf deze applicatie met een korte tekst (conceptbeschrijving), met een use-casediagram en met een klassediagram. Implementeer vervolgens dit systeem. De gemaakte code moet getest zijn.

Laat de conceptbeschrijving, het use-casediagram en het klassediagram z.s.m. door de docent keuren. Hiermee voorkom je dat je een applicatie maakt die niet voldoende bewijslast bevat voor deze toets. Maar… **wacht niet** met het implementeren tot er gekeurd is.

De criteria voor de beoordeling zijn:

* De gebruikte notatie voor het use-casediagram is correct.
* Het use-casediagram beschrijft de functionaliteit van het hele systeem.
* De gebruikte notatie voor het klassediagram is correct.
* De attributen en operaties van klassen komt overeen met de verantwoordelijkheid van de klassen.
* Het klassediagram is juist vertaald naar code.
* De functionele eisen, die uit de use cases voorkomen, zijn geïmplementeerd. De implementatie is getest.
* Je kan het ‘hoe en waarom’ van het gemaakte use case diagram, het klassediagram en de code verdedigen.
* Het programma vertoont geen bugs (blijft hangen, crashed, …)
* De gemaakte code is netjes: goede encapsulatie (private/public), duidelijke variabele namen, netjes inspringen, geen dubbele code, zelf toegevoegde methoden hebben een duidelijk doel en een duidelijke naam.
* Juist gebruik van loops (for/while/foreach)
* Juist gebruik van keuze structuren (if/else/else if/switch).
* Alle PRC2 leerdoelen worden aangetoond.
* Je heb de feedback op je vorige toetsopdrachten verwerkt.