

1. Welke diagrammen beschrijven het dynamisch gedrag van een applicatie?

- Use case-diagram
- Use case-beschrijving
- Activity diagram
- Sequentie diagram

2. Welke diagrammen beschrijven de structuur van de applicatie?

Klassendiagram in drie fasen:

- domeinmodel
- applicatiemodel
- implementatiemodel

3. Wat onderscheid een domeinmodel van een applicatiemodel?

Domeinmodel beschrijft alleen elementen die een rol spelen in de businessflow (vandaar dat het alleen klassen uit de business-layer toont (businessclass en factories). De concepten uit de business worden genoemd zonder daarbij al te denken in termen van een te bouwen applicatie.

In een applicatiemodel wordt al wel gedacht in termen van een te maken applicatie en worden in het klassenmodel de klassen toegevoegd die niet al in the business layer zitten zoals: forms, data-klassen en tasks.

4. UML is een internationale standaardmethode voor applicatieontwikkeling W/N

Niet Waar. UML is weliswaar een internationale standaard, maar het is een taal en geen methode. De taal kan gebruikt worden binnen verschillende te kiezen methoden.

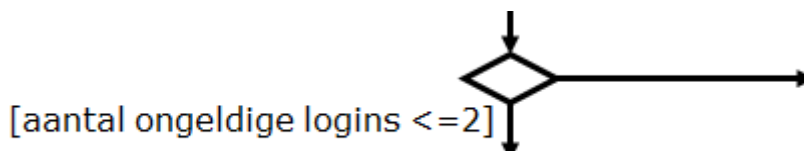
5. De semantiek van een taal wordt ook wel grammatica genoemd W/N

Niet waar

Syntaxis = grammatica

Semantiek = betekenis van (gecombineerde) symbolen

6. Teken een decision-node waarin getest wordt of het aantal foute logins kleiner is dan 3 (happy flow)



7. Het implementatiemodel beschrijft het hele gedrag van de applicatie W/N

Niet waar. In termen van "structuur" en "gedrag" is het implementatiemodel een diagram dat de structuur beschrijft en niet het gedrag.

8. Een use-case specificeert, hoe de binnenkant van de als “black box” voorgestelde applicatie er uit ziet W/N

Niet waar. Wanneer een use case wordt beschreven wordt functionaliteit beschreven vanuit het oogpunt van een gebruiker (van buitenaf).

9. In een use-case diagram worden de niet-functionele eisen zichtbaar W/N

Niet waar. Een use-case diagram beschrijft juist de functionele eisen.

10. Geef in de juiste notatie weer hoe de use case in het use case diagram wordt afgebeeld voor het registreren van een klanten.



11. Teken een actor symbool uit het use-case diagram.



12. Een <<extend>> relatie geeft aan dat een secundaire use-case een essentieel onderdeel uitmaakt bij het doorlopen van een de workflow van een primaire use-case. W/N.

Niet waar. Bij een <<include>> relatie is het doorlopen van de secundaire use-case essentieel. Bij een <<extend>> relatie is het doorlopen er van optioneel.

13. Welke benamingen zouden kunnen worden toegepast voor een actor: (1)“Klanten”, (2)“Ouder”, (3)“Directeur”. Waarom wel of niet.

(1) ‘niet’ vanwege het meervoud.

(2) ‘wel’ zie bijvoorbeeld casus ‘Pret voor 10’

(3) hangt af van de context. Directeur lijkt een rol die geen betrekking heeft op de applicatie zelf. Wanneer ‘Directeur’ een duidelijk te onderscheiden rol speelt binnen de context van de applicatie kan het een actor zijn.

14. Uit welk decennium stammen de eerste object-georiënteerde talen?

Uit de zestig jaren. De eerste OO-taal is Simula-67 (1967).

15. Wat is de Object Management Group?

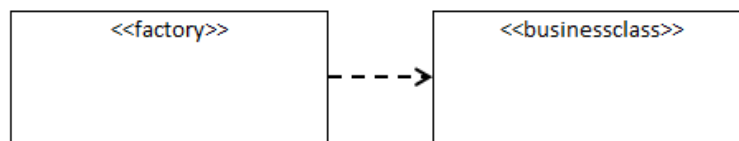
OMG is een internationale, non-profit projectgroep die de UML-taal standaard bewaard en onderhoudt.

16. In welk decennium werd de eerste UML versie (UML 1.0) gepresenteerd?

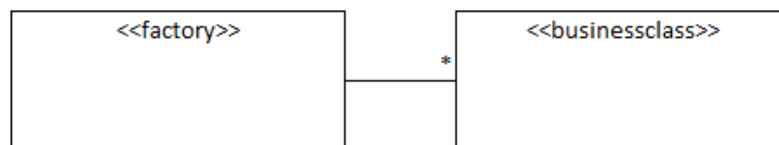
In de negentiger jaren. In 1997 werd een door OMG leden goedgekeurde eerste toepasbare standaard UML 1.0 geïntroduceerd.

17. Teken een businessclass en een bijbehorende factory en de tussenliggende relatie.

(a) De relatie tussen een factory en een businessclass kan worden weergegeven als een dependency relatie (pijl met stippellijn, zie blz. 163 van Hoogendoorn).



(b) In de dia's wordt de relatie ook vaak weergegeven als een gewone one-to-many relatie.



18. Wat komt eerst, een domeinmodel, of een use-case diagram?

Ze worden allebei gemaakt in de fase van het functioneel ontwerp. Kip-of-het-ei kwestie oftewel "ongeveer tegelijkertijd"

19. Wat is de andere benaming voor een volledig gedetailleerd klassendiagram?

Implementatiemodel.

20. Welke elementen maken standaard deel uit van een use-case beschrijving?

1. naam Use Case
2. samenvatting/korte beschrijving
3. actoren
4. precondities
5. stappenplan (happy flow)
6. uitzonderingen (alternative flows)
7. postcondities

21. Wat wordt verstaan onder de happy flow?

Normale voortgang van activiteiten binnen het stappenplan van een use-case beschrijving.

22. Hoe wordt de relatie genoemd tussen een actor en een use-case?

Associatie.

23. Van mogelijke uitzonderingen beschrijft men de-flow.

...alternative...

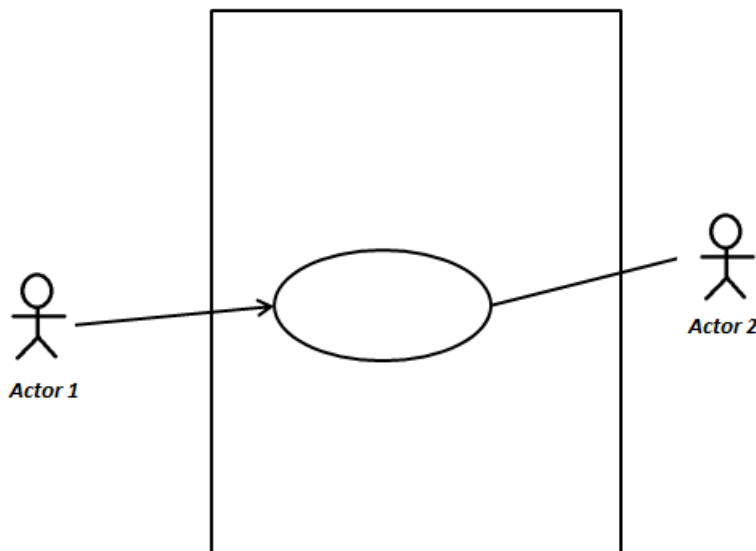
24. Wat verstaat men onder de waterval methode? Hoe noemen we de tegenhanger ervan?

Bij de ontwikkeling van een applicatie wordt gesproken over 'waterval' wanneer men wil modelleren tot in het laatste detail, om vervolgens 'in één keer' volgens de volledig uitgewerkte ontwerpen en schemas te coderen tot een werkende applicatie.

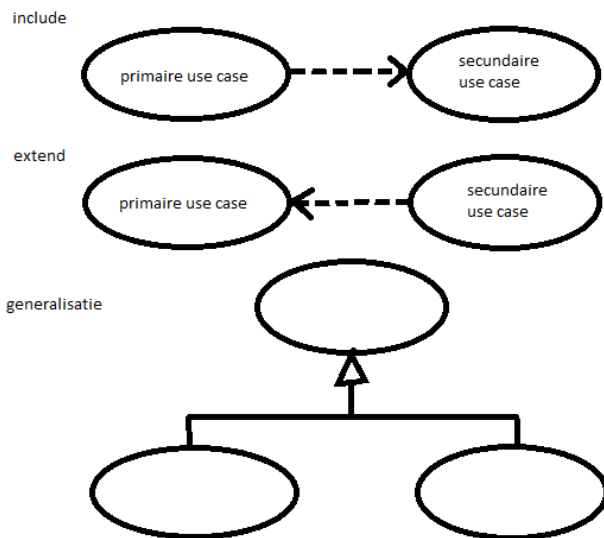
De tegenhanger van 'waterval' is de iteratieve methode. Hierbij wordt afwisselend gemodelleerd en gecodeerd om zodoende steeds dichterbij een volledig werkende applicatie toe te werken.

25. Hoe kan men onderscheid maken tussen de primaire actor en de secundaire actor wanneer beide een associatie hebben met één en dezelfde use-case?

De primaire actor is de actor die de use-case initieert. Dit is weer te geven met een pijl.



26. Noem drie mogelijke associaties tussen twee use-cases en teken van elk een voorbeeld.



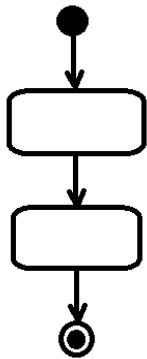
27. Waarom wordt de use-case “Tonen Webpage” gedurende het functioneel ontwerp liever niet gebruikt?

Webpage is al teveel gericht op de te kiezen technologie.

28. Wanneer in het stappenplan “Valideren Invoergegevens” stapnummer 5 is, welk nummer krijgt de uitzondering “Verplicht veld oningevuld”

Valideren zal zowieso plaats vinden. Na valideren van gegevens zal in de “happy flow” stap 6 worden genomen. De uitzonderingsstap waar hier naar wordt gevraagd zal daarom nummer 6a krijgen.

29. Teken een initial node, twee activiteiten ("Actor doet A", Systeem doet B") en een final node. Twee mogelijke pijlpunten



30. Hoe herken je het verschil tussen een decision node en een join/merge node?

Decision node: één pijl gaat de node in, twee pijlen verlaten de node

Join/merge node: twee pijlen gaan de node in, één pijl verlaat de node

31. Wat hebben pre-condities en postcondities met elkaar te maken?

De post-condities van de ene use-case moeten valide pre-condities zijn voor de use-case die er na wordt uitgevoerd.

32. Wat is een scenario?

Wanneer er beslismomenten voorkomen in een stappenplan, zijn er verschillende manieren om het stappenplan te doorlopen. Dit zijn de scenario's van een use case.

33. Hoe kan een activity-diagram gebruikt worden voor het systematisch uitvoeren van een test van een use-case?

Aan de hand van het activity diagram kunnen alle mogelijke scenario's worden gevonden. Het systematisch doorlopen van al deze scenarios kan het uitgangspunt vormen voor een 'volledig' testplan.

34. Hoe maak ik uit een activity-diagram op hoeveel post-condities en beschreven worden?

Check alle final nodes en kijk of er daar van elkaar verschillende postcondities optreden.

35. Bij elke activiteit in een activity-diagram wordt aangegeven of deze door een actor of het systeem wordt uitgevoerd W/N.

Waar. Dit wordt vaak vergeten. De activiteiten uit het stappenplan worden in het activiteitendiagram vooraf gegaan door "actor" of "systeem".

36. Hoe wordt het klassendiagram genoemd waarin minimaal alle te gebruiken klassen worden afgebeeld?

Het applicatiemodelmodel

37. Een object is een van een klasse.

...instantie...

38. Wanneer wordt gebruik gemaakt van factories?

Wanneer er bij het runnen van een applicatie steeds nieuwe objecten worden aangemaakt van een bepaalde klasse en er van deze nieuwe objecten iets van een gestructureerde lijst moet bijgehouden worden zodat één of meerdere objecten naderhand gezocht/geselecteerd kunnen worden.

39. Noem de stereotypes van 5 verschillende type klassen.

<<business>> <<factory>> <<data>> <<form>> <<task>>

40. Wat kunnen we terug vinden in één volledig ingevulde klasse uit het implementatiemodel?

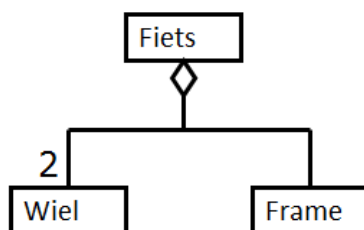
- Stereotype die het type van de klasse aangeeft
- Naam
- Methods
- Accessidentifiers ("+", "-", "#")
- Datatypes

41. Hoe wordt aangegeven dat een object A geassocieerd kan zijn met minimaal 3 objecten B?

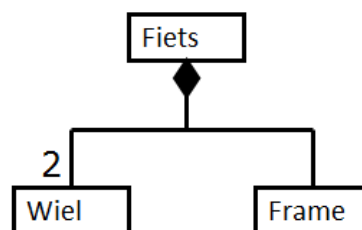
Door "3..*" bij de associatielijijn te plaatsen (aan het uiteinde dat grenst aan B)

42. Teken een compositie-relatie en een aggregatie-relatie tussen wielen, frame en fiets.

Aggregatie



Compositie



43. Wat is het verschil tussen een aggregatie- en een compositierelatie?

Bij een compositie kunnen de onderdelen niet los van elkaar bestaan (lifetime dependency), bij een aggregatie wel.

44. Hoe schrijf je een correcte klassenaam?

BewerkenProfiel (PascalCase en alleen letters)

45. Noem in de gebruikelijke volgorde de vier primaire lagen van de referentiestructuur.

- Presentation
- Process
- Business
- Data

46. Geef bij elk van de lagen van de referentiestructuur de belangrijkste bijbehorende klasstypen.

- Presentation: <<form>>
- Process: <<task>>
- Business: <<businessclass>>, <<factory>>
- Data: <<data>>

47. Waarom wordt de presentation-layer vaak bovenaan getekend?

Het is precies de laag die de gebruiker waarneemt.

48. Binnen welk klassentype bevindt zich de proceslogica?

In de task.

49. Binnen welk klassentype wordt het stappenplan van een use case uitgevoerd?

In de task.

50. Wat gebeurt er binnen de task voordat het stappenplan mag worden uitgevoerd?

De precondities worden gevalideerd.