

Theoretische Aspecten van Programmatuur

HC 4 (week 5)

Bob Diertens

Theory of Computer Science (TCS)

date: Maart, 8

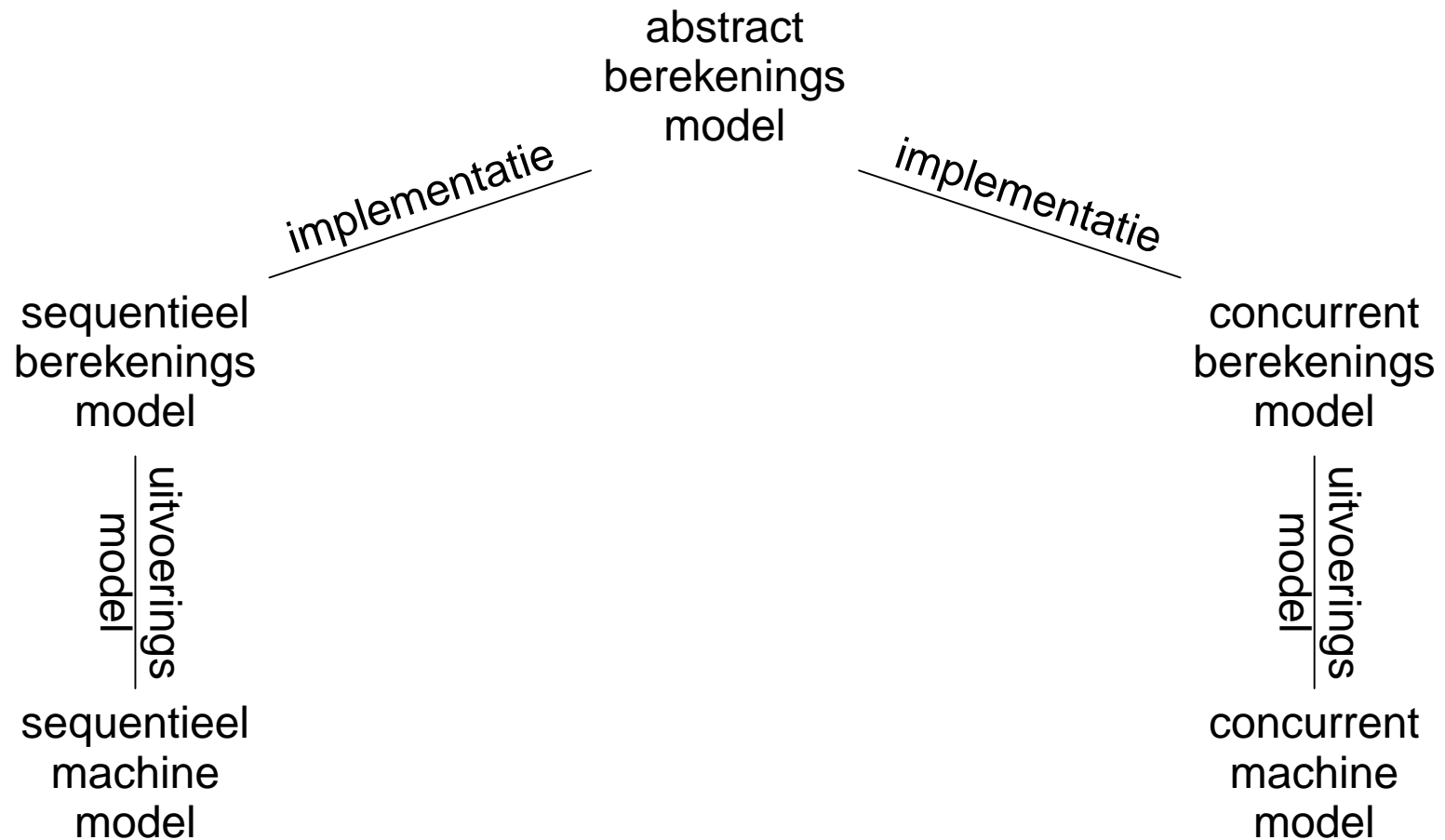
version: March 3, 2017

Wat

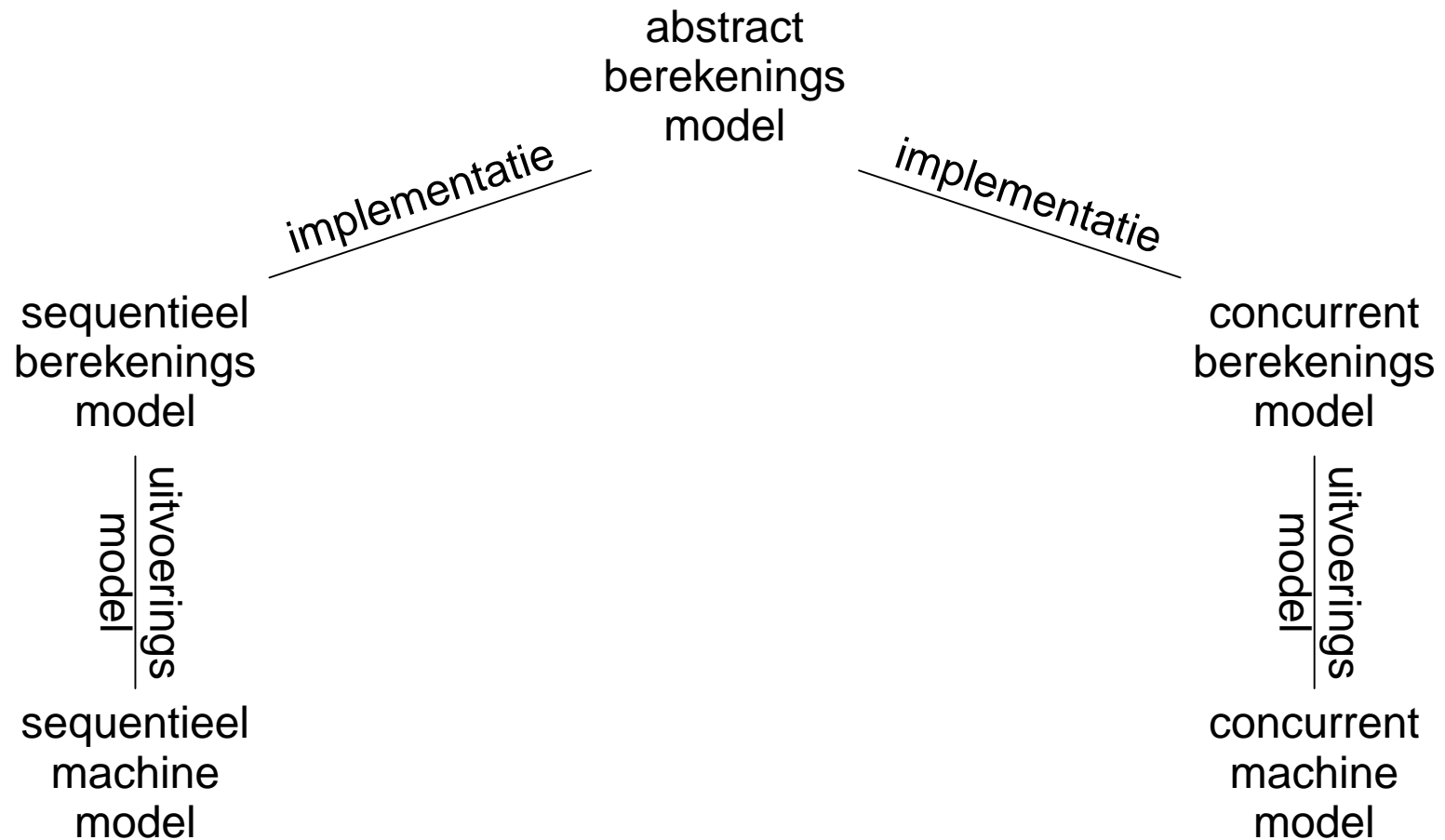
- Bestudering programmeer-constructies.
- Uitbreiding PGA (vnl. basis-instructies)
- Functies
- Berekeningsmodellen voor functies
- Concurrency support
- Communicatie
- Berekeningsmodel voor objecten
- Object-orientatie
- Gereedschap: Generalisatie & Abstractie

Berekenings Model voor Objecten

Raamwerk voor Uitvoering Functies



Raamwerk voor Uitvoering Functies



Objecten?

Uitvoering van Objecten

Sequentieel

Uitvoering van Objecten

Sequentieel

- sequentieel uitvoerings model voor objecten is gebaseerd op sequentieel uitvoerings model voor functies

Uitvoering van Objecten

Sequentieel

- sequentieel uitvoerings model voor objecten is gebaseerd op sequentieel uitvoerings model voor functies
- objecten worden niet uitgevoerd, maar de methoden van de objecten

Uitvoering van Objecten

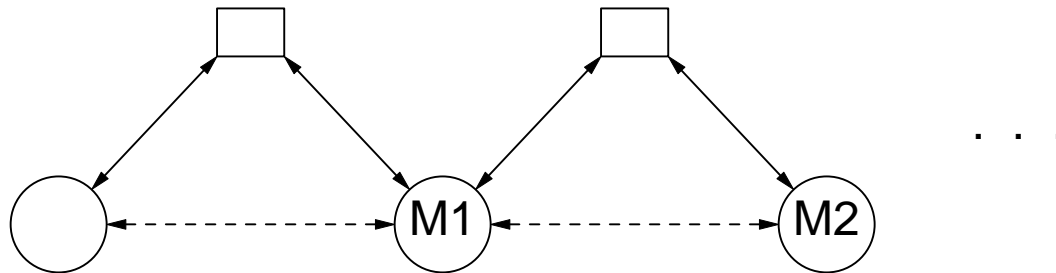
Sequentieel

- sequentieel uitvoerings model voor objecten is gebaseerd op sequentieel uitvoerings model voor functies
- objecten worden niet uitgevoerd, maar de methoden van de objecten
- \Rightarrow zelfde raamwerk als voor functies

Uitvoering van Objecten

Sequentieel

- sequentieel uitvoerings model voor objecten is gebaseerd op sequentieel uitvoerings model voor functies
- objecten worden niet uitgevoerd, maar de methoden van de objecten
- \Rightarrow zelfde raamwerk als voor functies



Uitvoering van Objecten (cont'd)

Concurrent

Uitvoering van Objecten (cont'd)

Concurrent

- uitbreiden raamwerk voor concurrente uitvoering van objecten

Uitvoering van Objecten (cont'd)

Concurrent

- uitbreiden raamwerk voor concurrente uitvoering van objecten
- bij het uitvoeren van een method wordt er naar het object geschakeld (door scheduler)!!!

Uitvoering van Objecten (cont'd)

Concurrent

- uitbreiden raamwerk voor concurrente uitvoering van objecten
- bij het uitvoeren van een method wordt er naar het object geschakeld (door scheduler)!!!
- schakeling wordt inline gedaan samen met de scheduling van de methode

Uitvoering van Objecten (cont'd)

Concurrent

- uitbreiden raamwerk voor concurrente uitvoering van objecten
- bij het uitvoeren van een method wordt er naar het object geschakeld (door scheduler)!!!
- schakeling wordt inline gedaan samen met de scheduling van de methode
- schakeling is de scheduling van het object

Uitvoering van Objecten (cont'd)

Concurrent

- uitbreiden raamwerk voor concurrente uitvoering van objecten
- bij het uitvoeren van een method wordt er naar het object geschakeld (door scheduler)!!!
- schakeling wordt inline gedaan samen met de scheduling van de methode
- schakeling is de scheduling van het object
- scheiden van scheduling van object en scheduling van methode

Uitvoering van Objecten (cont'd)

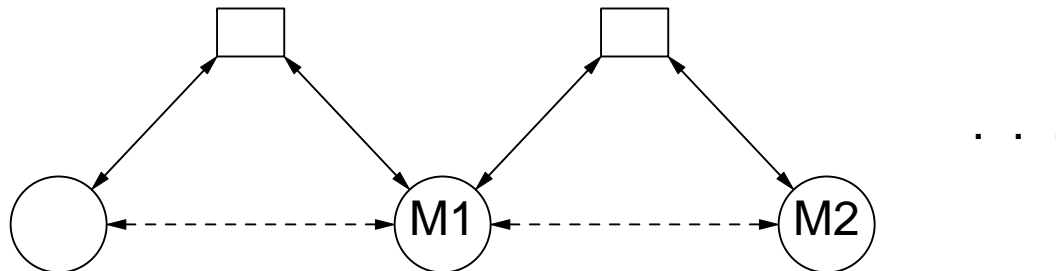
Concurrent

- uitbreiden raamwerk voor concurrente uitvoering van objecten
- bij het uitvoeren van een method wordt er naar het object geschakeld (door scheduler)!!!
- schakeling wordt inline gedaan samen met de scheduling van de methode
- schakeling is de scheduling van het object
- scheiden van scheduling van object en scheduling van methode
- \Rightarrow scheduler voor objecten en ieder object regelt scheduling van de methoden van het object

Uitvoering van Objecten (cont'd)

Concurrent

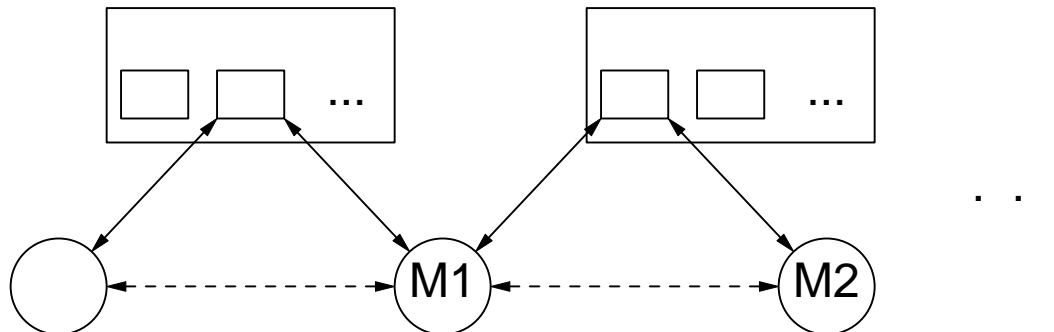
- uitbreiden raamwerk voor concurrente uitvoering van objecten
- bij het uitvoeren van een method wordt er naar het object geschakeld (door scheduler)!!!
- schakeling wordt inline gedaan samen met de scheduling van de methode
- schakeling is de scheduling van het object
- scheiden van scheduling van object en scheduling van methode
- \Rightarrow scheduler voor objecten en ieder object regelt scheduling van de methoden van het object



Uitvoering van Objecten (cont'd)

Concurrent

- uitbreiden raamwerk voor concurrente uitvoering van objecten
- bij het uitvoeren van een method wordt er naar het object geschakeld (door scheduler)!!!
- schakeling wordt inline gedaan samen met de scheduling van de methode
- schakeling is de scheduling van het object
- scheiden van scheduling van object en scheduling van methode
- \Rightarrow scheduler voor objecten en ieder object regelt scheduling van de methoden van het object



Uitvoering van Objecten (cont'd)

Abstractie

Uitvoering van Objecten (cont'd)

Abstractie

- abstraheren van hoe objecten worden gescheduled

Uitvoering van Objecten (cont'd)

Abstractie

- abstraheren van hoe objecten worden gescheduled
- \Rightarrow berekenings model voor concurrente uitvoering van objecten

Uitvoering van Objecten (cont'd)

Abstractie

- abstraheren van hoe objecten worden gescheduled
- \Rightarrow berekenings model voor concurrente uitvoering van objecten
- elk object bepaalt zelf hoe de methoden gescheduled worden

Uitvoering van Objecten (cont'd)

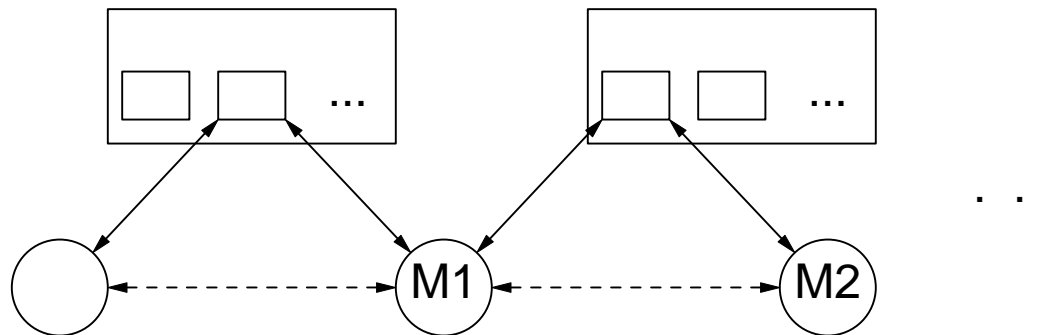
Abstractie

- abstraheren van hoe objecten worden gescheduled
- \Rightarrow berekenings model voor concurrente uitvoering van objecten
- elk object bepaalt zelf hoe de methoden gescheduled worden
- scheduling van methoden kan inline, dus alleen concurrentie voor objecten

Uitvoering van Objecten (cont'd)

Abstractie

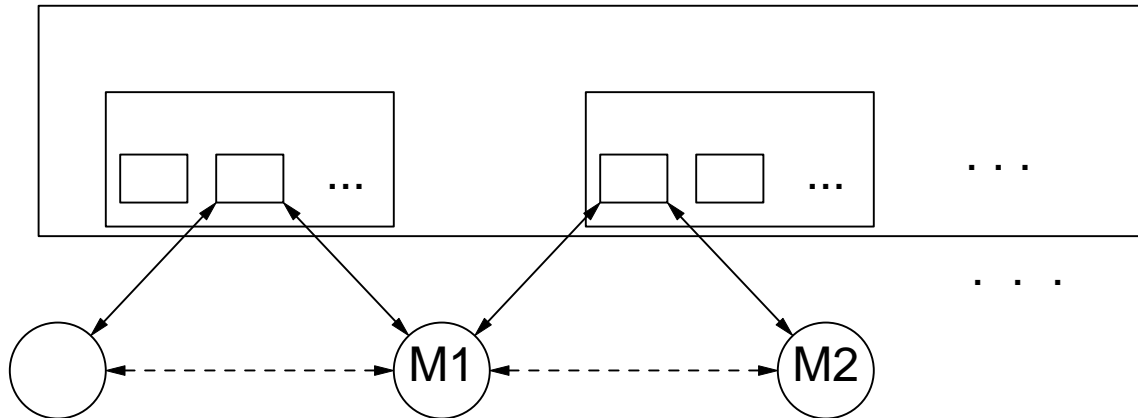
- abstraheren van hoe objecten worden gescheduled
- \Rightarrow berekenings model voor concurrente uitvoering van objecten
- elk object bepaalt zelf hoe de methoden gescheduled worden
- scheduling van methoden kan inline, dus alleen concurrentie voor objecten



Uitvoering van Objecten (cont'd)

Abstractie

- abstraheren van hoe objecten worden gescheduled
- \Rightarrow berekenings model voor concurrente uitvoering van objecten
- elk object bepaalt zelf hoe de methoden gescheduled worden
- scheduling van methoden kan inline, dus alleen concurrentie voor objecten



Uitvoering van Objecten (cont'd)

Implementatie

Uitvoering van Objecten (cont'd)

Implementatie

- vanuit dit model implementatie van model voor concurrente uitvoering van functies of methoden

Uitvoering van Objecten (cont'd)

Implementatie

- vanuit dit model implementatie van model voor concurrente uitvoering van functies of methoden
- maar ook, implementatie van model voor concurrente uitvoering van objecten

Uitvoering van Objecten (cont'd)

Implementatie

- vanuit dit model implementatie van model voor concurrente uitvoering van functies of methoden
- maar ook, implementatie van model voor concurrente uitvoering van objecten

Resultaat

Uitvoering van Objecten (cont'd)

Implementatie

- vanuit dit model implementatie van model voor concurrente uitvoering van functies of methoden
- maar ook, implementatie van model voor concurrente uitvoering van objecten

Resultaat

- voorheen het aanroepen van een methode van een object

Uitvoering van Objecten (cont'd)

Implementatie

- vanuit dit model implementatie van model voor concurrente uitvoering van functies of methoden
- maar ook, implementatie van model voor concurrente uitvoering van objecten

Resultaat

- voorheen het aanroepen van een methode van een object
- nu een verzoek aan het object

Uitvoering van Objecten (cont'd)

Implementatie

- vanuit dit model implementatie van model voor concurrente uitvoering van functies of methoden
- maar ook, implementatie van model voor concurrente uitvoering van objecten

Resultaat

- voorheen het aanroepen van een methode van een object
- nu een verzoek aan het object
- met abstractie van hoe de methode wordt uitgevoerd

Object Orientatie

Introductie

What is Object-Orientatie?

Introductie

What is Object-Orientatie?

Verleden

Introductie

What is Object-Orientatie?

Verleden

- OO is het resultaat van evolutie.

Introductie

What is Object-Orientatie?

Verleden

- OO is het resultaat van evolutie.
- Er was al sprake van "Objecten" voor OO.

Introductie

What is Object-Orientatie?

Verleden

- OO is het resultaat van evolutie.
- Er was al sprake van "Objecten" voor OO.
 - System Simulation - objecten in Simula
 - Operating Systems - monitors by Hoare

Introductie

What is Object-Orientatie?

Verleden

- OO is het resultaat van evolutie.
- Er was al sprake van "Objecten" voor OO.
 - System Simulation - objecten in Simula
 - Operating Systems - monitors by Hoare
 - Artificial Intelligence - frames als eenheden van kennis

Introductie

What is Object-Orientatie?

Verleden

- OO is het resultaat van evolutie.
- Er was al sprake van "Objecten" voor OO.
 - System Simulation - objecten in Simula
 - Operating Systems - monitors by Hoare
 - Artificial Intelligence - frames als eenheden van kennis
 - data abstracties

Introductie

What is Object-Orientatie?

Verleden

- OO is het resultaat van evolutie.
- Er was al sprake van "Objecten" voor OO.
 - System Simulation - objecten in Simula
 - Operating Systems - monitors by Hoare
 - Artificial Intelligence - frames als eenheden van kennis
 - data abstracties
 - 'net' programmeerwerk

Introductie

What is Object-Orientatie?

Verleden

- OO is het resultaat van evolutie.
- Er was al sprake van "Objecten" voor OO.
 - System Simulation - objecten in Simula
 - Operating Systems - monitors by Hoare
 - Artificial Intelligence - frames als eenheden van kennis
 - data abstracties
 - 'net' programmeerwerk
- Reden: om kunnen gaan met complexiteit van software.

Introductie

What is Object-Orientatie?

Verleden

- OO is het resultaat van evolutie.
- Er was al sprake van "Objecten" voor OO.
 - System Simulation - objecten in Simula
 - Operating Systems - monitors by Hoare
 - Artificial Intelligence - frames als eenheden van kennis
 - data abstracties
 - 'net' programmeerwerk
- Reden: om kunnen gaan met complexiteit van software.
- Karakteristiek: een object is een logisch of fysieke onafhankelijke entiteit.

Introductie

What is Object-Orientatie?

Verleden

- OO is het resultaat van evolutie.
- Er was al sprake van "Objecten" voor OO.
 - System Simulation - objecten in Simula
 - Operating Systems - monitors by Hoare
 - Artificial Intelligence - frames als eenheden van kennis
 - data abstracties
 - 'net' programmeerwerk
- Reden: om kunnen gaan met complexiteit van software.
- Karakteristiek: een object is een logisch of fysieke onafhankelijke entiteit.
- Belangrijk: scheiding van Wat en Hoe (abstractie).

Introductie

What is Object-Orientatie?

Verleden

- OO is het resultaat van evolutie.
- Er was al sprake van "Objecten" voor OO.
 - System Simulation - objecten in Simula
 - Operating Systems - monitors by Hoare
 - Artificial Intelligence - frames als eenheden van kennis
 - data abstracties
 - 'net' programmeerwerk
- Reden: om kunnen gaan met complexiteit van software.
- Karakteristiek: een object is een logisch of fysieke onafhankelijke entiteit.
- Belangrijk: scheiding van Wat en Hoe (abstractie).

(zie: [L.F. Capretz, *A Brief History of the Object-Oriented Approach*, 2003])

Introductie (cont'd)

Verwarring en Onduidelijkheid

Introductie (cont'd)

Verwarring en Onduidelijkheid

Om een beschrijving van OO te geven moet eerst een aantal dingen duidelijk gemaakt worden.

Introductie (cont'd)

Verwarring en Onduidelijkheid

Om een beschrijving van OO te geven moet eerst een aantal dingen duidelijk gemaakt worden.

Kwesties:

- Elementen
- Abstractie en Generalisatie
- Encapsulatie en Informatie Verberging
- Beschrijving

Kwestie: Elementen

Kwestie: Elementen

Object-orientatie wordt vaak uitgelegd met wat een OO programmeertaal te bieden heeft.

Kwestie: Elementen

Object-orientatie wordt vaak uitgelegd met wat een OO programmeertaal te bieden heeft.

Wanneer is een PT OO?

Kwestie: Elementen

Object-orientatie wordt vaak uitgelegd met wat een OO programmeertaal te bieden heeft.

Wanneer is een PT OO?

- klassificatie
- overerving
- polymorphisme
- encapsulatie
- abstractie

Kwestie: Elementen

Object-orientatie wordt vaak uitgelegd met wat een OO programmeertaal te bieden heeft.

Wanneer is een PT OO?

- klassificatie
- overerving
- polymorfisme
- encapsulatie
- abstractie

Zijn het eigenschappen van OO of slechts elementen van PT (mogelijk ter ondersteuning van OO)?

Kwestie: Elementen

Object-orientatie wordt vaak uitgelegd met wat een OO programmeertaal te bieden heeft.

Wanneer is een PT OO?

- klassificatie
- overerving
- polymorfisme
- encapsulatie
- abstractie

Zijn het eigenschappen van OO of slechts elementen van PT (mogelijk ter ondersteuning van OO)?

Voor OO zijn deze elementen niet noodzakelijk!

Kwestie: Elementen

Object-orientatie wordt vaak uitgelegd met wat een OO programmeertaal te bieden heeft.

Wanneer is een PT OO?

- klassificatie
- overerving
- polymorfisme
- encapsulatie
- abstractie

Zijn het eigenschappen van OO of slechts elementen van PT (mogelijk ter ondersteuning van OO)?

Voor OO zijn deze elementen niet noodzakelijk!

Behalve abstractie, maar die term is hier verkeerd gebruikt.

Kwestie: Elementen

Object-orientatie wordt vaak uitgelegd met wat een OO programmeertaal te bieden heeft.

Wanneer is een PT OO?

- klassificatie
- overerving
- polymorfisme
- encapsulatie
- abstractie

Zijn het eigenschappen van OO of slechts elementen van PT (mogelijk ter ondersteuning van OO)?

Voor OO zijn deze elementen niet noodzakelijk!

Behalve abstractie, maar die term is hier verkeerd gebruikt.

Beschrijven van OO in termen van geboden elementen van een PT die OO ondersteunen

⇒ een PT is OO als het die elementen ondersteund.

Kwestie: Elementen

Object-orientatie wordt vaak uitgelegd met wat een OO programmeertaal te bieden heeft.

Wanneer is een PT OO?

- klassificatie
- overerving
- polymorfisme
- encapsulatie
- abstractie

Zijn het eigenschappen van OO of slechts elementen van PT (mogelijk ter ondersteuning van OO)?

Voor OO zijn deze elementen niet noodzakelijk!

Behalve abstractie, maar die term is hier verkeerd gebruikt.

Beschrijven van OO in termen van geboden elementen van een PT die OO ondersteunen

⇒ een PT is OO als het die elementen ondersteund.

Circulaire redenering.

Kwestie: Elementen (cont'd)

Nadruk op elementen van een OO PT, zoals klassificatie, overerving en polymorfisme:

Kwestie: Elementen (cont'd)

Nadruk op elementen van een OO PT, zoals klassificatie, overerving en polymorfisme:

- leid tot specificatie van data objecten

Kwestie: Elementen (cont'd)

Nadruk op elementen van een OO PT, zoals klassificatie, overerving en polymorfisme:

- leid tot specificatie van data objecten
- leid af van de OO concepten van gedrags-abstractie

Kwestie: Elementen (cont'd)

Nadruk op elementen van een OO PT, zoals klassificatie, overerving en polymorfisme:

- leid tot specificatie van data objecten
- leid af van de OO concepten van gedrags-abstractie

Op hoger niveau van abstractie in ontwerp is er geen noodzaak voor die elementen,
met als gevolg is dat ontwerp op het niveau van programmeren plaats vind.

Kwestie: Elementen (cont'd)

Nadruk op elementen van een OO PT, zoals klassificatie, overerving en polymorfisme:

- leid tot specificatie van data objecten
- leid af van de OO concepten van gedrags-abstractie

Op hoger niveau van abstractie in ontwerp is er geen noodzaak voor die elementen,

met als gevolg is dat ontwerp op het niveau van programmeren plaats vind.

Het lijkt erop dat het gebruik van die elementen van een OO PT het doel is geworden.

Kwestie: Abstractie en Generalisatie

Kwestie: Abstractie en Generalisatie

- veel verwarring over wat het is
- worden verwisseld
- zijn beslist niet hetzelfde

Kwestie: Abstractie en Generalisatie

- veel verwarring over wat het is
- worden verwisseld
- zijn beslist niet hetzelfde

Abstractie

Een detail wordt weggelaten als het niet belangrijk is in een beschrijving op een hoger niveau van abstractie.

Kwestie: Abstractie en Generalisatie

- veel verwarring over wat het is
- worden verwisseld
- zijn beslist niet hetzelfde

Abstractie

Een detail wordt weggelaten als het niet belangrijk is in een beschrijving op een hoger niveau van abstractie.

Generalisatie

Het detail wordt niet weggelaten, maar op een algemene manier beschreven op hetzelfde niveau van abstractie.

Kwestie: Abstractie en Generalisatie

- veel verwarring over wat het is
- worden verwisseld
- zijn beslist niet hetzelfde

Abstractie

Een detail wordt weggelaten als het niet belangrijk is in een beschrijving op een hoger niveau van abstractie.

Generalisatie

Het detail wordt niet weggelaten, maar op een algemene manier beschreven op hetzelfde niveau van abstractie.

VOORBEELD rood object, groen object, en blauw object

Kwestie: Abstractie en Generalisatie

- veel verwarring over wat het is
- worden verwisseld
- zijn beslist niet hetzelfde

Abstractie

Een detail wordt weggelaten als het niet belangrijk is in een beschrijving op een hoger niveau van abstractie.

Generalisatie

Het detail wordt niet weggelaten, maar op een algemene manier beschreven op hetzelfde niveau van abstractie.

VOORBEELD rood object, groen object, en blauw object

generalisatie: objecten, ieder met een bepaalde kleur (klassificering: object(kleur) - instanties: object(rood), ...)

Kwestie: Abstractie en Generalisatie

- veel verwarring over wat het is
- worden verwisseld
- zijn beslist niet hetzelfde

Abstractie

Een detail wordt weggelaten als het niet belangrijk is in een beschrijving op een hoger niveau van abstractie.

Generalisatie

Het detail wordt niet weggelaten, maar op een algemene manier beschreven op hetzelfde niveau van abstractie.

VOORBEELD rood object, groen object, en blauw object

generalisatie: objecten, ieder met een bepaalde kleur (klassificering: object(kleur) - instanties: object(rood), ...)

abstractie: objecten (als de kleur er niet toe doet) (concretisering: rood object, ...)

Kwestie: Encapsulatie en Informatie Verberging

Kwestie: Encapsulatie en Informatie Verberging

- worden verwisseld
- gebruikt met zelfde betekenis

Kwestie: Encapsulatie en Informatie Verberging

- worden verwisseld
- gebruikt met zelfde betekenis

Encapsulatie

Voorkoming van interactie met object anders dan gedefinieerd.

Verbergt niets van het object.

Kwestie: Encapsulatie en Informatie Verberging

- worden verwisseld
- gebruikt met zelfde betekenis

Encapsulatie

Voorkoming van interactie met object anders dan gedefinieerd.

Verbergt niets van het object.

Informatie Verberging

Maakt het onmogelijk bepaalde informatie te zien van het object.

Voorkomt op zich zelf geen interactie met het object.

Kwesties: Beschrijving

Kwesties: Beschrijving

Objecten worden vaak beschreven in termen van data en een set van functies

Kwesties: Beschrijving

Objecten worden vaak beschreven in termen van data en een set van functies

Denk aan de Unified Modelling Language (UML).

Kwesties: Beschrijving

Objecten worden vaak beschreven in termen van data en een set van functies

Denk aan de Unified Modelling Language (UML).

Maar niveau van abstractie van UML stijgt niet uit boven dat van OO PT.

Kwesties: Beschrijving

Objecten worden vaak beschreven in termen van data en een set van functies

Denk aan de Unified Modelling Language (UML).

Maar niveau van abstractie van UML stijgt niet uit boven dat van OO PT.

Details van data en functies wordt wel weggelaten, maar er is geen abstractie van de implementatie van objecten.

Kwesties: Beschrijving

Objecten worden vaak beschreven in termen van data en een set van functies

Denk aan de Unified Modelling Language (UML).

Maar niveau van abstractie van UML stijgt niet uit boven dat van OO PT.

Details van data en functies wordt wel weggelaten, maar er is geen abstractie van de implementatie van objecten.

Dus nutteloos op hogere niveaus van abstractie.

Kwesties: Beschrijving

Objecten worden vaak beschreven in termen van data en een set van functies

Denk aan de Unified Modelling Language (UML).

Maar niveau van abstractie van UML stijgt niet uit boven dat van OO PT.

Details van data en functies wordt wel weggelaten, maar er is geen abstractie van de implementatie van objecten.

Dus nutteloos op hogere niveaus van abstractie.

Objecten moeten beschreven worden door hun gedrag, ipv in termen van data en functies.

Object Orientatie

Object Orientatie

OO is een manier voor het organiseren van een systeem in termen van objecten en hun relaties.

Object Orientatie

OO is een manier voor het organiseren van een systeem in termen van objecten en hun relaties.

Het dient dichterbij te staan bij de 'echte wereld' dan voorgaande technieken.

Object Orientatie

OO is een manier voor het organiseren van een systeem in termen van objecten en hun relaties.

Het dient dichterbij te staan bij de 'echte wereld' dan voorgaande technieken.

Het kenmerk is het onderscheid tussen observeerbaar gedrag van objecten en de implementatie van het gedrag.

Object Orientatie (cont'd)

Fundamenten

Object Orientatie (cont'd)

Fundamenten

Objecten

Een object heeft de volgende karakteristieken.

toestand

voor het vastleggen van het verleden waarop toekomstig gedrag gebaseerd kan worden

gedrag

observeerbare effecten gebaseerd op de toestand en relaties met andere objecten

identiteit

zoals het object bekend is bij andere objecten, dmv naam of referentie

Object Orientatie (cont'd)

Fundamenten

Objecten

Een object heeft de volgende karakteristieken.

toestand

voor het vastleggen van het verleden waarop toekomstig gedrag gebaseerd kan worden

gedrag

observeerbare effecten gebaseerd op de toestand en relaties met andere objecten

identiteit

zoals het object bekend is bij andere objecten, dmv naam of referentie

Relaties

Relaties tussen objecten worden uitgedrukt door interacties in de vorm van bericht uitwisseling.

Object Orientatie (cont'd)

Fundamenten

Objecten

Een object heeft de volgende karakteristieken.

toestand

voor het vastleggen van het verleden waarop toekomstig gedrag gebaseerd kan worden

gedrag

observeerbare effecten gebaseerd op de toestand en relaties met andere objecten

identiteit

zoals het object bekend is bij andere objecten, dmv naam of referentie

Relaties

Relaties tussen objecten worden uitgedrukt door interacties in de vorm van bericht uitwisseling.

Abstractie

Manipulatie van een object kan alleen plaatsvinden dmv de relaties met andere objecten. Waarbij implementatie van het gedrag en de vastlegging van de toestand niet van belang is.

Object Orientatie (cont'd)

Ondersteuning

Object Orientatie (cont'd)

Ondersteuning

Typen

In het type van een object wordt de toestand en het gedrag gedefinieerd.

Object Orientatie (cont'd)

Ondersteuning

Typen

In het type van een object wordt de toestand en het gedrag gedefinieerd.

Bericht Uitwisseling

De manier waarop berichten tussen objecten worden uitgewisseld kan in verschillende vormen ondersteund worden.

Object Orientatie (cont'd)

Ondersteuning

Typen

In het type van een object wordt de toestand en het gedrag gedefinieerd.

Bericht Uitwisseling

De manier waarop berichten tussen objecten worden uitgewisseld kan in verschillende vormen ondersteund worden.

Encapsulatie

Encapsulatie voorkomt dat objecten relaties onderhouden met andere objecten dan de manieren van bericht uitwisseling die ondersteund worden.

Object Orientatie (cont'd)

Ondersteuning

Typen

In het type van een object wordt de toestand en het gedrag gedefinieerd.

Bericht Uitwisseling

De manier waarop berichten tussen objecten worden uitgewisseld kan in verschillende vormen ondersteund worden.

Encapsulatie

Encapsulatie voorkomt dat objecten relaties onderhouden met andere objecten dan de manieren van bericht uitwisseling die ondersteund worden.

Informatie Verberging

Verbergen van informatie omtrent een object kan gedaan worden door deze informatie expres niet observeerbaar te maken.

Object Orientatie (cont'd)

Structuren

Object Orientatie (cont'd)

Structuren

Type Compositie

Een type object kan samengesteld worden uit andere typen object.

referentie

een type object kan refereren naar een object van een bepaald type object

inclusie

een type object kan een ander type object invoegen

Object Orientatie (cont'd)

Structuren

Type Compositie

Een type object kan samengesteld worden uit andere typen object.

referentie

een type object kan refereren naar een object van een bepaald type object

inclusie

een type object kan een ander type object invoegen

Om aan de OO fundamenteën te voldoen moet de modelleertaal de compositie verbergen.

Object Orientatie (cont'd)

Structuren

Type Compositie

Een type object kan samengesteld worden uit andere typen object.

referentie

een type object kan refereren naar een object van een bepaald type object

inclusie

een type object kan een ander type object invoegen

Om aan de OO fundamenteën te voldoen moet de modelleertaal de compositie verbergen.

Dit kan door de elementen van de compositie alleen beschikbaar te maken van binnen uit, of van buiten af maar dan als elementen van het type object zelf.

Object Orientatie (cont'd)

Structuren

Type Compositie

Een type object kan samengesteld worden uit andere typen object.

referentie

een type object kan refereren naar een object van een bepaald type object

inclusie

een type object kan een ander type object invoegen

Om aan de OO fundamenteën te voldoen moet de modelleertaal de compositie verbergen.

Dit kan door de elementen van de compositie alleen beschikbaar te maken van binnens uit, of van buiten af maar dan als elementen van het type object zelf.

verticale compositie

Object Orientatie (cont'd)

Structuren

Type Compositie

Een type object kan samengesteld worden uit andere typen object.

referentie

een type object kan refereren naar een object van een bepaald type object

inclusie

een type object kan een ander type object invoegen

Om aan de OO fundamenteën te voldoen moet de modelleertaal de compositie verbergen.

Dit kan door de elementen van de compositie alleen beschikbaar te maken van binnen uit, of van buiten af maar dan als elementen van het type object zelf.

verticale compositie

Object Compositie

Een aantal onderling gerelateerde objecten vormen een cluster. Door te abstraheren van de onderlinge relaties kan dit cluster als een enkel object functioneren.

Object Orientatie (cont'd)

Structuren

Type Compositie

Een type object kan samengesteld worden uit andere typen object.

referentie

een type object kan refereren naar een object van een bepaald type object

inclusie

een type object kan een ander type object invoegen

Om aan de OO fundamenteën te voldoen moet de modelleertaal de compositie verbergen.

Dit kan door de elementen van de compositie alleen beschikbaar te maken van binnen uit, of van buiten af maar dan als elementen van het type object zelf.

verticale compositie

Object Compositie

Een aantal onderling gerelateerde objecten vormen een cluster. Door te abstraheren van de onderlinge relaties kan dit cluster als een enkel object functioneren.

horizontale compositie

Abstracte Typen Object

Een abstract type object bevat elementen die objecten representeren die op een lager niveau van abstractie ingevuld moeten worden.

Abstracte Typen Object

Een abstract type object bevat elementen die objecten representeren die op een lager niveau van abstractie ingevuld moeten worden.

Een abstract type object kan op een lager niveau van abstractie omgevormd worden tot een generiek type object, met parameters voor de betreffende elementen.

Van Functies naar Object Orientatie

Gebruikers

Gebruikers

- Gebruik functies en objecten uitgelegd aan de hand van programmeertaal.

Gebruikers

- Gebruik functies en objecten uitgelegd aan de hand van programmeertaal.
- Geen uitleg berekenings modellen.

Gebruikers

- Gebruik functies en objecten uitgelegd aan de hand van programmeertaal.
- Geen uitleg berekenings modellen.
- Modellen zijn impliciet aanwezig bij vertalers en de machines.

Gebruikers

- Gebruik functies en objecten uitgelegd aan de hand van programmeertaal.
- Geen uitleg berekenings modellen.
- Modellen zijn impliciet aanwezig bij vertalers en de machines.
- Voor begrijpen semantiek functies, objecten, en hun relaties is kennis berekenings modellen essentieel.

FO naar OO

FO naar OO

- Sinds introductie OO pogingen omzetten van FO programma's naar OO programma's.
- Resultaat ?

FO naar OO

- Sinds introductie OO pogingen omzetten van FO programma's naar OO programma's.
- Resultaat ?
- Gebrek aan expliciete beschrijvingen en berekenings modellen.

FO

- programma
- functie
- variabele

Relaties met OO?

Functie

Functie

- Functies en objecten:
 - instantiëring van uit definitie
 - uitvoering
 - vernietiging

Functie

- Functies en objecten:
 - instantiëring van uit definitie
 - uitvoering
 - vernietiging
- Bij functie zijn instantiatie en vernietiging impliciet.
- Bij object zijn instantiatie en vernietiging (meestal) expliciet.
- Toestand functie in (gedeelde) variabelen.
- Een functie is een object.

Programma

Programma opgebouwd uit functies:

Programma

Programma opgebouwd uit functies:

- Uitvoering programma is uitvoering object.

Programma

Programma opgebouwd uit functies:

- Uitvoering programma is uitvoering object.
- De functies zijn de methoden van het object.

Programma

Programma opgebouwd uit functies:

- Uitvoering programma is uitvoering object.
- De functies zijn de methoden van het object.
- Variabelen vormen de toestand van het object.

Programma

Programma opgebouwd uit functies:

- Uitvoering programma is uitvoering object.
- De functies zijn de methoden van het object.
- Variabelen vormen de toestand van het object.
- Een programma is een object.

Variable

Variable

- instantiëring
- manipulatie
- vernietiging

Variable

- instantiëring
- manipulatie
- vernietiging
- Een variabele is een object.

Alles is een object

Alles is een object

- Een functie is een object.
- Een programma is een object.
- Een variable is een object.

Alles is een object

- Een functie is een object.
- Een programma is een object.
- Een variable is een object.

Verschil:

- berekenings modellen

Dus Object-Orientatie?

Alles is een object

- Een functie is een object.
- Een programma is een object.
- Een variable is een object.

Verschil:

- berekenings modellen

Dus Object-Orientatie?

- Overeenstemming met OO paradigma nodig.

Alles is een object

- Een functie is een object.
- Een programma is een object.
- Een variable is een object.

Verschil:

- berekenings modellen

Dus Object-Orientatie?

- Overeenstemming met OO paradigma nodig.
- Moet voldoen aan fundamenteen OO.

Overeenstemming met Object-Orientatie

Overeenstemming met Object-Orientatie

Objecten

Een object heeft de volgende karakteristieken.

toestand

voor het vastleggen van het verleden waarop toekomstig gedrag gebaseerd kan worden

gedrag

observeerbare effecten gebaseerd op de toestand en relaties met andere objecten

identiteit

zoals het object bekend is bij andere objecten, dmv naam of referentie

Relaties

Relaties tussen objecten worden uitgedrukt door interacties in de vorm van bericht uitwisseling.

Abstractie

Manipulatie van een object kan alleen plaatsvinden dmv de relaties met andere objecten. Waarbij implementatie van het gedrag en de vastlegging van de toestand niet van belang is.

Objecten

Objecten

raamwerk berekenings modellen:

abstractie van manier van scheduling -> functies kunnen gezien worden als objecten.

Objecten

raamwerk berekenings modellen:

abstractie van manier van scheduling -> functies kunnen gezien worden als objecten.

- identiteit

Objecten

raamwerk berekenings modellen:

abstractie van manier van scheduling -> functies kunnen gezien worden als objecten.

- identiteit
 - verkregen bij creatie instantie
 - kan verspreid worden naar andere instanties

Objecten

raamwerk berekenings modellen:

abstractie van manier van scheduling -> functies kunnen gezien worden als objecten.

- identiteit
 - verkregen bij creatie instantie
 - kan verspreid worden naar andere instanties
- toestand
 - kan leeg zijn
 - lokale variabelen
 - via andere functie instantie

Objecten

raamwerk berekenings modellen:

abstractie van manier van scheduling -> functies kunnen gezien worden als objecten.

- identiteit
 - verkregen bij creatie instantie
 - kan verspreid worden naar andere instanties
- toestand
 - kan leeg zijn
 - lokale variabelen
 - via andere functie instantie
- gedrag
 - observeerbaar dmv effecten op omgeving
 - gebaseerd op toestand en relaties met andere instanties

Relaties

Relaties

- Communicatie mechanismen zijn zelf functies, en dus objecten

Relaties

- Communicatie mechanismen zijn zelf functies, en dus objecten
- Beperking van communicatie tot alleen deze mechanismen
⇒ alleen interactie dmv bericht uitwisseling

Abstractie

Abstractie

Cluster van objecten en communicatie met buiten alleen via een object

Abstractie

Cluster van objecten en communicatie met buiten alleen via een object

- observeerbaar gedrag kan gezien worden als gedrag van dat ene object

Abstractie

Cluster van objecten en communicatie met buiten alleen via een object

- observeerbaar gedrag kan gezien worden als gedrag van dat ene object
- intern gedrag is niet observeerbaar

Abstractie

Cluster van objecten en communicatie met buiten alleen via een object

- observeerbaar gedrag kan gezien worden als gedrag van dat ene object
- intern gedrag is niet observeerbaar
- abstractie van het interne gedrag is mogelijk

Abstractie

Cluster van objecten en communicatie met buiten alleen via een object

- observeerbaar gedrag kan gezien worden als gedrag van dat ene object
- intern gedrag is niet observeerbaar
- abstractie van het interne gedrag is mogelijk

⇒ beperking communicatie tot gedefinieerde relaties maakt abstractie van implementatie gedrag en toestand mogelijk

Conclusie

Dmv abstractie FO en OO gerelateerd.

- alles is een object (hoog niveau van abstractie)
- mogelijk verschil in berekenings model (op lager niveau van abstractie)
- blijven objecten, maar we noemen het anders (op lager niveau van abstractie)

CONTENTS

Theoretische Aspecten van Programmatuur	1
Wat	2
Raamwerk voor Uitvoering Functies	4
Uitvoering van Objecten	5
Uitvoering van Objecten (cont'd)	6
Uitvoering van Objecten (cont'd)	7
Uitvoering van Objecten (cont'd)	8
Introductie	10
Introductie (cont'd)	11
Kwestie: Elementen	12
Kwestie: Elementen (cont'd)	13
Kwestie: Abstractie en Generalisatie	14
Kwestie: Encapsulatie en Informatie Verberging	15
Kwesties: Beschrijving	16
Object Orientatie	17
Object Orientatie (cont'd)	18
Object Orientatie (cont'd)	19
Object Orientatie (cont'd)	20
Gebruikers	23
FO naar OO	24
FO	25
Functie	26
Programma	27
Variable	28
Alles is een object	29
Overeenstemming met Object-Orientatie	30
Objecten	31
Relaties	32
Abstractie	33
Conclusie	34