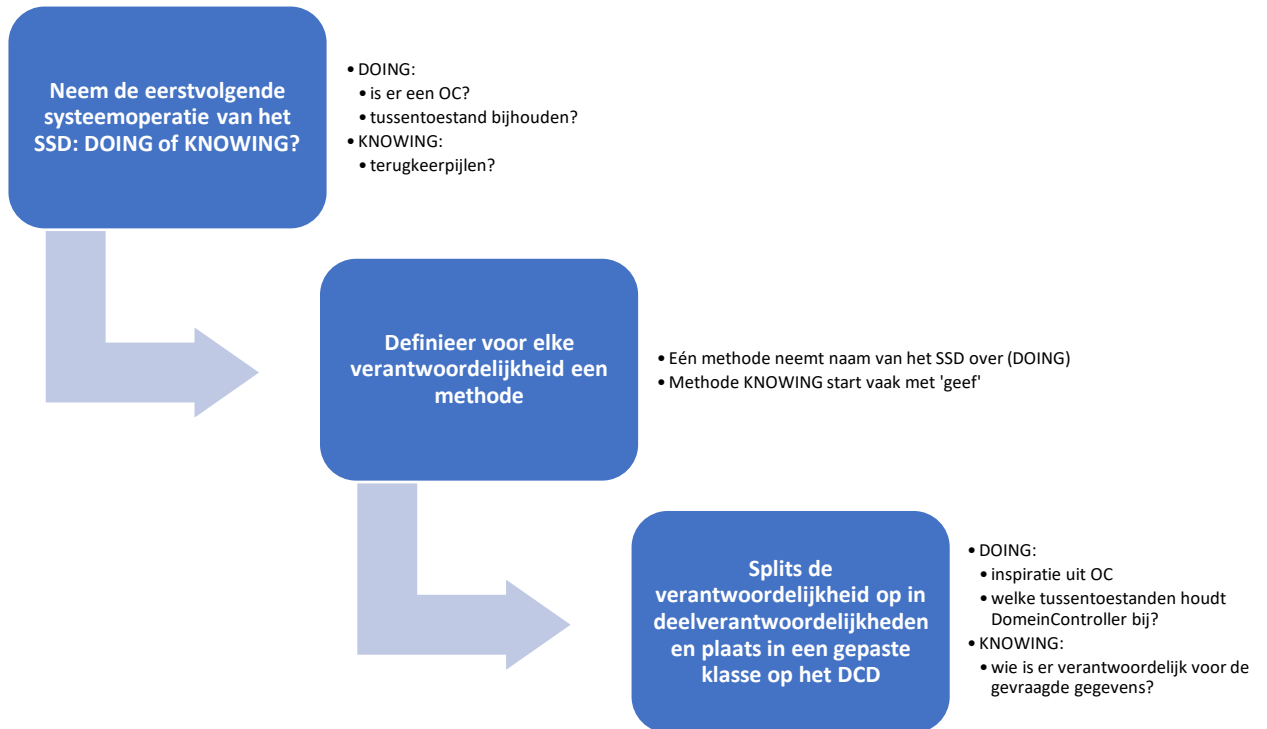


Stappenplan DCD ontwerpen

Kort samengevat



In detail

Je herhaalt volgende stappen voor elke systeemoperatie op het SSD.

Stap 1 – Neem de eerstvolgende systeemoperatie van het SSD

Neem de eerstvolgende systeemoperatie van het SSD die je nog niet uitgewerkt hebt.

Bepaal hiervoor of het DOING of KNOWING is. Indien je 'ja' antwoordt op één van onderstaande vragen, heb je die verantwoordelijkheid.

- DOING:
 - Is er een Operation Contract (OC) voor deze systeemoperatie?
 - Moeten we een tussentoestand bijhouden voor een latere systeemoperatie? (bv. aangemelde gebruiker, een geselecteerd object...)
- KNOWING:
 - Zijn er terugkeerpijlen?
 - elke terugkeerpijl = 1 KNOWING-verantwoordelijkheid
 - Een loop-voorwaarde op een SSD is altijd KNOWING

Stap 2 – Definieer voor elke verantwoordelijkheid een methode

In de volgende stap definieer je voor elke verantwoordelijkheid een methode (met signatuur zoals op DCD).

- DOING:
 - Neemt naam over vanop het SSD
 - Returntype altijd `void`
- KNOWING:
 - Start vaak met 'geef' voor o.a. terugkeerpijlen of berekende gegevens
 - Returntype: te bepalen afhankelijk van wat teruggegeven wordt.
 - Keuze uit primitieve datatypes en standaard beschikbare datatypes in Java (`String`, `LocalDate`...)

Stap 3 – Splits de verantwoordelijkheid op in deelverantwoordelijkheden en plaats in een gepaste klasse op het DCD

Je gaat elke verantwoordelijkheid (methoden uit stap 2) verder uitwerken en opsplitsen in deelverantwoordelijkheden. Daarna plaats je die methoden op het DCD in een gepaste klasse.

Laat je hierbij inspireren door deze bronnen:

- DOING:
 - Het OC
 - Denk na of de DomeinController tussentoestanden moet bijhouden voor volgende systeemoperaties
- KNOWING:
 - Wie is er verantwoordelijk voor de informatie die opgehaald/berekend moet worden? (= Expert)

Je baseert je daarbij op de 3 gekende GRASP patronen:

- **Controller**
 - Probleem: Welk is het eerste object, achter de User Interface, dat verantwoordelijk is voor het ontvangen en coördineren van een systeemoperatie?
 - Oplossing: Ken de verantwoordelijkheid toe aan een klasse die het volledige systeem representeert, een Controller klasse.

- **Creator**
 - Probleem: Wat is er verantwoordelijk voor het aanmaken van nieuwe instanties van een klasse?
 - Oplossing: Ken de verantwoordelijkheid om instanties van klasse A te creëren toe aan klasse B, wanneer 1 of meer van de volgende gelden:
 - B bevat instanties van A (associatie: bevat, heeft)
 - B is een aggregatie of een compositie van A
 - B gebruikt A intensief
 - B bevat de data om A te initialiseren

- **(Information) Expert:**
 - Probleem: Wat is het algemeen principe dat we kunnen hanteren voor het toekennen van verantwoordelijkheden aan objecten?
 - Oplossing: Ken de verantwoordelijkheid toe aan de Information Expert - dit is de klasse die alle informatie, nodig om de verantwoordelijkheid te realiseren, heeft.

Indien je een groep objecten wil bijhouden, dan maak je daar een **repository** voor. Dit is niet de verantwoordelijkheid van de DomeinController.