Objects & Classes  
Jeroen Boogaard

## Decomposition

Bij Decomposition wordt de complexiteit verdeeld over (minder complexe) Components die elk een eigen duidelijk afgebakende veranwoordelijkheid hebben. De verschillende Components dragen samen bij aan een bepaalde taak. De totale complexiteit is daarmee niet afgenomen (soms zelfs toegenomen) maar verplaatst. Decomposition maakt het gemakkelijker om in te zoomen op de details van een bepaalde Component (Encapsulation) en de interne werking van andere Components te negeren (Information Hiding). Hierdoor wordt het geheel gemakkelijker te onderhouden (Modifiability) en kunnen onderdelen hergebruikt (Reusability) en soms ook onafhankelijk geschaald worden (Scalability).

Decomposition vindt plaats op verschillende abstractieniveaus:

* Een Application combineert verschillende (Micro)Services bijv. OV App
* Een Service bestaat uit Modules die ieder bijdragen aan een gezamenlijke taak bijv. het beschikbaar maken van aankomst- en vertrektijden.
* Binnen een Module kunnen variabelen en functies zijn ondergebracht in verschillende Classes.

Een Class is een abstractie van een entiteit binnen de context van een applicatie.  
In een Class worden variabelen en functies gegroepeerd zodanig dat

* het een herkenbare entiteit in het toepassingsdomein vertegenwoordigd bijv. Car
* de onderdelen een sterke onderlinge samenhang (strong Cohesion) heeft bijv. Steering wheel, Engine, Tires, etc.
* er weinig afhankelijkheid is (Low Coupling) van andere Classes zoals Petrol Station

Met een Class worden Objects gegenereerd met vooraf gedefinieerde attributen (variabelen) en methoden (functies). Een Object wordt geïnstantieerd door het toekennen van waarden (attribute values) bijv. color = “blue”. Elk object heeft zijn eigen Attribute Values (state) en is daarmee een unieke Instance van een Class.