**Softwareontwerp & -architectuur 3**

**Casusopdracht Bioscoop Deel 3 - Strategy Pattern**

# Inleiding

In het eerste deel van de opdracht hebben we een casus uitgewerkt rondom een bioscoopsysteem. Tijdens het tweede deel hebben we geconstateerd dat de uitwerkingen hiervan over het algemeen moeilijk testbaar en redelijk complex waren. In het college hebben we geconstateerd dat door het meten van de *cyclomatic complexity* die complexiteit gekwantificeerd kon worden. We gaan nu aan de hand van het Strategy Pattern verbetering aanbrengen in de onderhoudbaarheid van onze code.

In volgende delen van de casusopdracht zullen we andere patterns toepassen om te zien wat de effecten zijn. Merk hierbij op dat het toepassen van design patterns in een eenvoudige setting als onze casusopdracht nogal “overengineered” kan voelen. Dat is logisch: het is een eenvoudig voorbeeld om te oefenen. Design patterns stralen in een complexere setting – maar dat is weer lastiger te volgen en niet geschikt om te oefenen.

# Opdracht

* Bestudeer hoofdstuk 1 van Head First Design Patterns en het lesmateriaal van het college uit Les 3
* Pas je verkregen kennis over het strategy pattern toe op je uitwerking Deel 1 en Deel 2 van de casusopdracht, zowel voor het berekenen van de prijs, als op het exporteren van de order. Teken hiervoor eerst een nieuw class diagram. Voeg een korte beschrijving toe van de OO-ontwerpprincipes die je hebt toegepast.
* Pas je (test)code aan op je nieuwe uitwerking (of hoeft dat niet :-).
* Draai SonarQube opnieuw over jouw code. Hoe zijn je metrics nu?

Je werkt in duo’s - dezelfde duo’s als voor de eindopdracht.