**Softwareontwerp & -architectuur 3**

**Casusopdracht Bioscoop Deel 4 - State Pattern**

# Inleiding

In de eerdere opdrachten hebben we een casus uitgewerkt rondom een bioscoopsysteem. We hebben vastgesteld dat de uitwerkingen lastig testbaar waren en dat we onder andere middels het meter van de *cyclomatic complexity* de kwaliteit van de code kwantificeerbaar konden maken. We hebben ook verbeteringen aangebracht middels het Strategy Pattern. In dit deel richten we ons op het State Pattern. We gaan ons nu richten op state-gedrag en daarbij houden we rekening met testbaarheid, onderhoudbaarheid etc.

# Casus

Een order doorloopt een aantal stadia. Eerst wordt een order aangemaakt. Deze bevat dan oa het gewenste aantal kaartjes voor de opgegeven voorstelling, optioneel parkeerkaart etc. In dit stadium zijn de plaatsen in de zaal nog niet definitief vastgelegd. De klant kan daarna de order ‘submitten’ en daarmee zijn de plaatsen gereserveerd. De order is dan opgeslagen bij het profiel van de klant. Wijzigen is dan nog mogelijk.

De klant kan vervolgens betalen; wijzigen is dan niet meer mogelijk. Het systeem verzendt na betaling de tickets en markeert ze daarna als ‘afgehandeld’. Zolang de klant nog niet heeft betaald, kan hij de order annuleren.

Indien een order 24 uur voor aanvang van de voorstelling nog niet betaald is, wordt een herinnering verstuurd. De order wordt dan aangemerkt als provisional (voorlopig). Indien 12 uur voor aanvang de betaling nog niet is afgerond, wordt de order automatisch geannuleerd en daarmee de plaatsen vrijgegeven.

# Opdracht

* Bestudeer hoofdstuk 10 van Head First Design Patterns.
* Pas je verkregen kennis toe op je uitwerking van de bioscoop-casus. Omdat we hier niet direct relatie hebben met het Strategy-pattern van je vorige uitwerking, kun je met de basis-code misschien wel beter een nieuw project starten. Teken hiervoor eerst relevante UML- diagrammen: class diagram + state transition diagram. Voeg een korte beschrijving toe van de OO-ontwerpprincipes die je hebt toegepast.
* Maak testcases en programmeer die.
* Draai SonarQube opnieuw over jouw code. Hoe zijn je metrics nu?