



Veiling

Design Patterns

Startdocument



Made by
students

Versie: 1.1

Emmen, 22-03-2022

Docent – Gerjan van Oenen

Studenten – Marvin Hoffenkamp, Kariem de Vries, Robin Goeree

Contact

Naam	Student nr.	E-mail
Marvin Hoffenkamp	4674626	marvin.hoffenkamp@student.nhlstenden.com
Kariem de Vries	4716876	kariem.de.vries@student.nhlstenden.com
Robin Goeree	4697081	robin.goeree@student.nhlstenden.com

Opdrachtgegevens

Ontwikkelomgeving: Visual Studio 2019;

Programmeertaal: C#;

Versiebeheer: Github, <https://github.com/MarvinHoffenkamp/Design-Patterns> (Staat momenteel op privé).

Opdracht

Voor de eindopdracht design patterns is het plan om een veiling te simuleren. Daarvoor is het plan om vier verschillende patterns te implementeren.

Om een veiling te hosten willen wij gebruik maken van een builder pattern. Dit pattern kan gebruikt worden om verschillende soorten veilingen aan te maken d.m.v. verschillende locaties, veilingmeesters en verkoopobjecten. De verschillende locaties kunnen online zijn of in verschillende warenhuizen. Elke veilingmeester zal uniek zijn door te verschillen op het startbod en het volgende bod, het startbod en het volgende bod zal in percentages bepaald worden door de veilingmeester en op basis van de aangegeven einddoelprijs berekend worden. Het einddoelprijs wordt bepaald door de gebruiker van de veiling simulatie applicatie, tijdens het bouwen van de veiling. De verkoopobjecten zullen afhankelijk zijn van de locatie waar de veiling wordt gehouden, wordt de veiling bijvoorbeeld in een klein warenhuis gehouden kunnen hier alleen kleine objecten verkocht worden.

Wij willen voor de veilingmeester een state pattern gebruiken. Dit pattern zou gebruikt kunnen worden om aan te geven of de veilingmeester een veiling laat beginnen, bezig is met een veiling of een veiling aan het afsluiten is. Afhankelijk wat de state is van de veilingmeester kunnen er verschillende acties worden ondernomen door de veiling of kopers.

Wij willen voor de verschillende verkoop objecten een factory method gebruiken. Zo zouden er basis objecten gemaakt kunnen worden die onderling verschillen, maar nog wel met dezelfde functionaliteiten functioneren.

Als laatste willen wij voor de kopers het observer pattern gebruiken. Door dit pattern te gebruiken zouden de kopers kunnen observeren wat de veilingmeester doet. Als de veilingmeester bijvoorbeeld een nieuwe veiling begint kunnen kopers hierop de keuze maken de veiling bij te wonen. Verder zou de veilingmeester tijdens veilingen nieuwe biedingen aangeven aan de kopers, waardoor de koper hier weer bovenop kan opbieden.

Klassen Diagram

