

**Object Georiënteerd Ontwerp**

Zeeslag

[Naam en Voornaam]

[Jaar, Opleiding en Reeks]

[Datum]

University Colleges Leuven-Limburg  
Campus Proximus  
Geldenaaksebaan 335  
3001 Heverlee  
Tel. +32 (0) 16 375 700

Inhoud

[Algemene opmerking: 3](#_Toc403305641)

[Vereisten 3](#_Toc403305642)

[Patterns 5](#_Toc403305643)

[Speciale topics 6](#_Toc403305644)

[Werkverdeling 7](#_Toc403305645)

[Klassediagramma 8](#_Toc403305646)

# Algemene opmerking:

Dit verslag gaat over de opdracht ‘Zeeslag’, dat zal meetellen voor 5 punten van je totaalscore van dit vak. Je dient je code in via GIT voor elk nieuw labo. Dit verslag vormt de toelichting bij jouw code. Binnen dit document gebruiken wij een Chronometer project als voorbeeld bij de verschillende topics. Jij mag natuurlijk enkel voorbeelden uit je eigen project aanhalen.

Je wordt geacht om je voor het verslag volledig aan deze template te houden. Buiten deze ‘algemene opmerking’ zijn alle hoofdingen verplichte onderdelen van het verslag. Je dient dit verslag in via Toledo voor de allerlaatste les van dit vak.

**EXTRA: Je bent verplicht om je in code gebruik te maken van het @author commentaarveld!**

Zeeslag

|  |  |  |
| --- | --- | --- |
| Vereiste | OK? | Issues? (indien niet ok) |
| Bij het starten wordt *App zeeslag getoond.* | ja | / |
| De zeeslag App heeft 2 borden 1 voor de spelers en 1 voor de computer (AI). | ja | / |
| De Zeeslag heeft een aparte shippicker panel, waarmee je via de afrollijst kan kiezen welke boat je in het board wil plaatsen. | ja | / |
| Men kan kiezen met checkbox, hoe zij hun boat wil plaatsen horizontaal of vertical | Ja | / |
| Er is ook 2 knoppen voorzien ‘Start’ and ‘Reset’ | ja | / |
| Zo lang men geen 5 boaten plaatst op het boat kan, hij de spel niet opstarten, niet klikbaar | ja | / |
| Wanneer men op de start knop klikt, wordt de start knop gedisabled | ja | / |
| Wanneer men op de reset knop klikt wordt de board volledig leeggemaakt (technisch implementatie: nieuwe boarden gemaakt createcell(x,y)) | ja | / |
| Wanneer men op maximize klik word de venster groter maar de size van panel niet gewijzigt | Ja | / |
| Als men op X klikken, gaat de applicatie sluiten | Ja | / |

7

# Vereisten

Geef voor het gehele project aan welke vereisten (evt. zelf verder uitgewerkt/opgesplitst aan de hand van de opgave) je succesvol hebben geïmplementeerd, en welke topics niet gelukt zijn. Zie hieronder voor een voorbeeld van een hele andere toepassing, met name Chronometer. Indien je een deel van de voorziene vereisten niet afgewerkt hebt, geef dan aan waarom niet… De reden kan “tijdsgebrek” zijn, het kan een issue zijn “wist niet hoe, het crashte”, of het kan zijn dat je een zeer goede reden had om het niet te implementeren…

Voeg het definitieve gegenereerde klassendiagramma van je code toe, als afzonderlijke afbeelding, als bijlage bij dit rapport.

Als er zaken zijn uit de opdracht die je niet hebt kunnen uitwerken of die je beter zou willen uitwerken, dan som je deze hier op.

Een voorbeeldje van mogelijke vereisten voor je project kan je hieronder terugvinden. Dit is **zeker niet** volledig! Vul het verder aan naargelang de vordering van het project. Zorg dat je niets vergeet!

|  |  |  |
| --- | --- | --- |
| Vereiste | OK? | Issues? (indien niet ok) |
| Ik kan een product toevoegen aan een verkoop |  |  |
| Ik kan als klant het totaalbedrag zien dat ik moet betalen |  |  |
| Het totaalbedrag wordt automatisch geüpdatet wanneer er een nieuw product wordt toegevoegd |  |  |
| De kassier kan zien welke producten werden toegevoegd |  |  |
| … |  |  |

# Patterns

Geef voor elk gezien patroon aan waar je het gebruikt hebt (mogelijkerwijs meer dan eens). Genereer een klassendiagramma voor elke situatie waarin je het patroon gebruikt hebt. Ons gegeven voorbeeldje komt uit het Chronometer project. Jij gebruikt uiteraard alleen voorbeelden uit je eigen project.

Geef extra informatie (voordelen / waarom / …). Als je een patroon niet toegepast hebt, leg je uit waarom niet.

|  |  |  |
| --- | --- | --- |
| Observer  (voorbeeld) | Ja  In Chronometer:  Pas *infopanel* automatisch aan wanneer de tijd verhoogt |  |

|  |  |  |
| --- | --- | --- |
| Observer |  |  |
| Strategy |  |  |
| Simple Factory |  |  |
| Façade |  |  |
| Adapter |  |  |
| Singleton |  |  |
| State |  |  |
| Template |  |  |
| MVC |  |  |
| …enz. |  |  |

Je mag zoveel rijen toevoegen als je nodig hebt!

# Speciale topics

Geef voor elke “speciale topic” aan of je het gebruikt hebt of niet, en zo ja, waar. Toon aan met een klassendiagramma.

|  |  |  |
| --- | --- | --- |
| Enum (voorbeeld) | Ja, in chronometer, om mogelijke statussen aan te geven (started, resumed, …) |  |

|  |  |  |
| --- | --- | --- |
| Enum |  |  |
| Properties |  |  |
| Class |  |  |
| JDBC |  |  |
| Mockito |  |  |
| Streams |  |  |
| Lambda-expressies |  |  |
| Enz… |  |  |

# Werkverdeling

Geef aan in percentages hoeveel je bij benadering gespendeerd hebt aan deze opdracht, **buiten de contacturen**.

|  |  |  |
| --- | --- | --- |
|  | [Naam] | [Naam] |
| Ontwerp |  |  |
| Klassendiagrammen |  |  |
| Implementatie |  |  |
| Verslag |  |  |

# Klassediagramma

Voeg het definitieve gegenereerde klassendiagram van je code toe, als afzonderlijk afbeelding-bestand.