

**Object Georiënteerd Ontwerp**

**Kassa App**

Sibren Jacobs

Yarne Vandenplas

Alexandre Vryghem

2019, Toegepaste Informatica , ME-1TI/9

[Datum]

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

# Algemene opmerking

Dit verslag gaat over de opdracht ‘Kassa app’, dat zal meetellen voor 5 punten van je totaalscore van dit vak. Dit verslag vormt de toelichting bij jouw code.

**Je wordt geacht om je voor het verslag volledig aan deze template te houden. Buiten deze ‘algemene opmerking’ zijn alle paragrafen verplichte onderdelen van het verslag. Je print dit verslag af (dubbelzijdig) dit in tijdens het herexamen 000**

**Je maakt een zip file van al je source code (.java bestanden, geen .class bestanden) en van alle bestanden die van belang zijn voor deze OOO opdracht. Je voegt tevens de laatste versie van je verslag (Word document) toe aan deze zip file.**

**. Je laadt deze zip file op via Toledo uiterlijk op maandag 23 december 2019 - 23.59 uur. Naam van de zip file: volgnr\_familienaam1\_familienaam2\_familienaam3\_Kassa\_OOO2019. Het volgnr krijg je van de lector.**

**Het spreekt voor zich dat de code op Toledo overeenstemt met de code in je repository op GitHub.**

# URL GITHUB repository

**Copy/paste hier de URL van je Github repository met je zelfevaluatie app project**

# URL

<https://github.com/r0747249/26_Jacobs_Vandenplas_Vryghem_Kassa_OOO2019>

# 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. 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 tevens op (in laatste rij van tabel).

|  |  |  |
| --- | --- | --- |
| User story | OK? | indien niet ok – wat lukt niet (zie acceptatie-criteria) en waarom? |
| 1. Overzicht artikelen tonen | OK |  |
| 1. Excel bestand inlezen | OK |  |
| 1. Registreren kassaverkoop | OK |  |
| 1. Tonen kassaverkoop aan klant | OK |  |
| 1. Verwijderen artikel uit kassaverkoop | OK |  |
| 1. Kassaverkoop on hold | OK |  |
| 1. Kortingen toepassen | OK |  |
| 1. Kassaverkoop afsluiten | OK |  |
| 1. Kassaverkoop betalen | OK |  |
| 1. Kassabon printen (op console) | OK |  |

# 

# Design patterns

Geef voor elk gezien patroon aan waar je het gebruikt hebt (mogelijkerwijs meer dan eens). Genereer een klassendiagram voor elke situatie waarin je het patroon gebruikt hebt. Geef extra informatie (voordelen / waarom / …). Als je een patroon niet toegepast hebt, leg uit waarom niet.

|  |  |  |
| --- | --- | --- |
|  | Toegepast (ja/nee)  In welke stories(nr)  Waarom toegepast(voordeel) | Bijhorend klassendiagram (gegenereerd vanuit je java code) |
| Observer | Ja  Story 4: wanneer er een element aan de winkelkar toegevoegd wordt moet de lijst geüpdatet worden.  Story 6: na de lijst te kopiëren om onhold te zetten moet de kassa view en de klanten view geüpdatet worden zodat we een lege lijst zien  Story 8 & 9: om de klantview te veranderen naar het afbetaalscherm en weer naar een nieuw leeg scherm  Het voordeel is dat het programma niet constant zal moeten kijken als er een element toegevoegd werd aan de lijst of als er op een knop gedrukt werd. |  |
| Strategy | Ja  Story 1: voor het type database  Story 2: voor het inladen van de DataBase vanuit een bestand  Story 7: gebruikt om verschillende soorten kortingen te implementeren  Het voordeel is dat je dus makkelijk een nieuwe type korting hier bv kan toevoegen of een nieuwe manier om een ander soort bestand in te lezen kan toevoegen. |  |
| Simple Factory | Ja  Story 2: voor het maken van de database  Story 7: voor het berekenen van de korting  Story 10: voor het maken van het kassaticket | Geen UML want met niets verbonden |
| Façade | Nee |  |
| Singleton | Nee |  |
| State | Ja, story 9  Een verkoop kan in verschillende states zijn oftewel actief, geannuleerd of onholdState.  Het voordeel aan states te gebruiken is dat we sommige methodes alleen beschikbaar kunnen maken voor een bepaalde state bent bv als je in een active state bent dan kan je een winkelkar onHold zetten. |  |
| MVC | Ja, overal  Voor elke view is er een controller die de view met het model verbindt met de code van model  Voordeel | De klassendiagrammen kan u op de 3 laatste pagina’s van dit verslag terugvinden. |
| Decorator | Ja, story 10  Hiermee kan je op een makkelijke manier de lay-out van het kassaticket veranderen bv de boodschap veranderen of de footer wel of niet tonen enz. |  |
| Template method | Ja  Story 1:  Ksnap ni echt het nut van templates |  |

# Speciale topics

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

|  |  |  |
| --- | --- | --- |
|  | Toegepast (ja/nee)  In welke stories(nr)  Waarom toegepast(voordeel) | Bijhorend klassendiagram of extra uitleg |
| Enum | Ja, in story 7 & ?.  Deze enum bevat de beschrijving/naam van de verschillende productgroepen  Story 10: Deze enum bevat alle onderdelen van de decorator met hun beschrijving |  |
| Properties | Ja, in story 7  Hiermee worden de instellingen bijgehouden zelf na het afsluiten van de kassa app | Geen uml ni? |
| Reflection |  |  |

# Werkverdeling

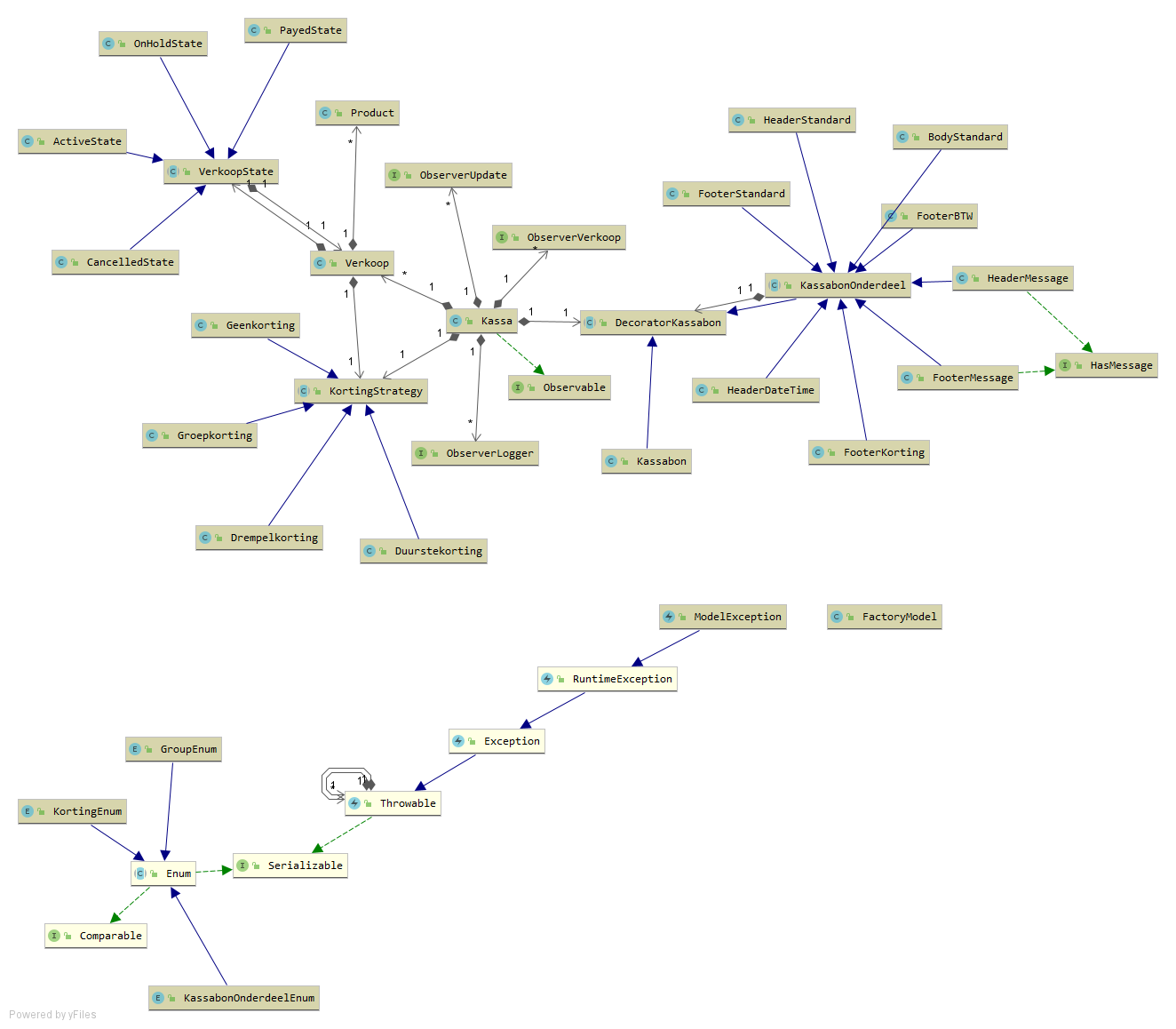
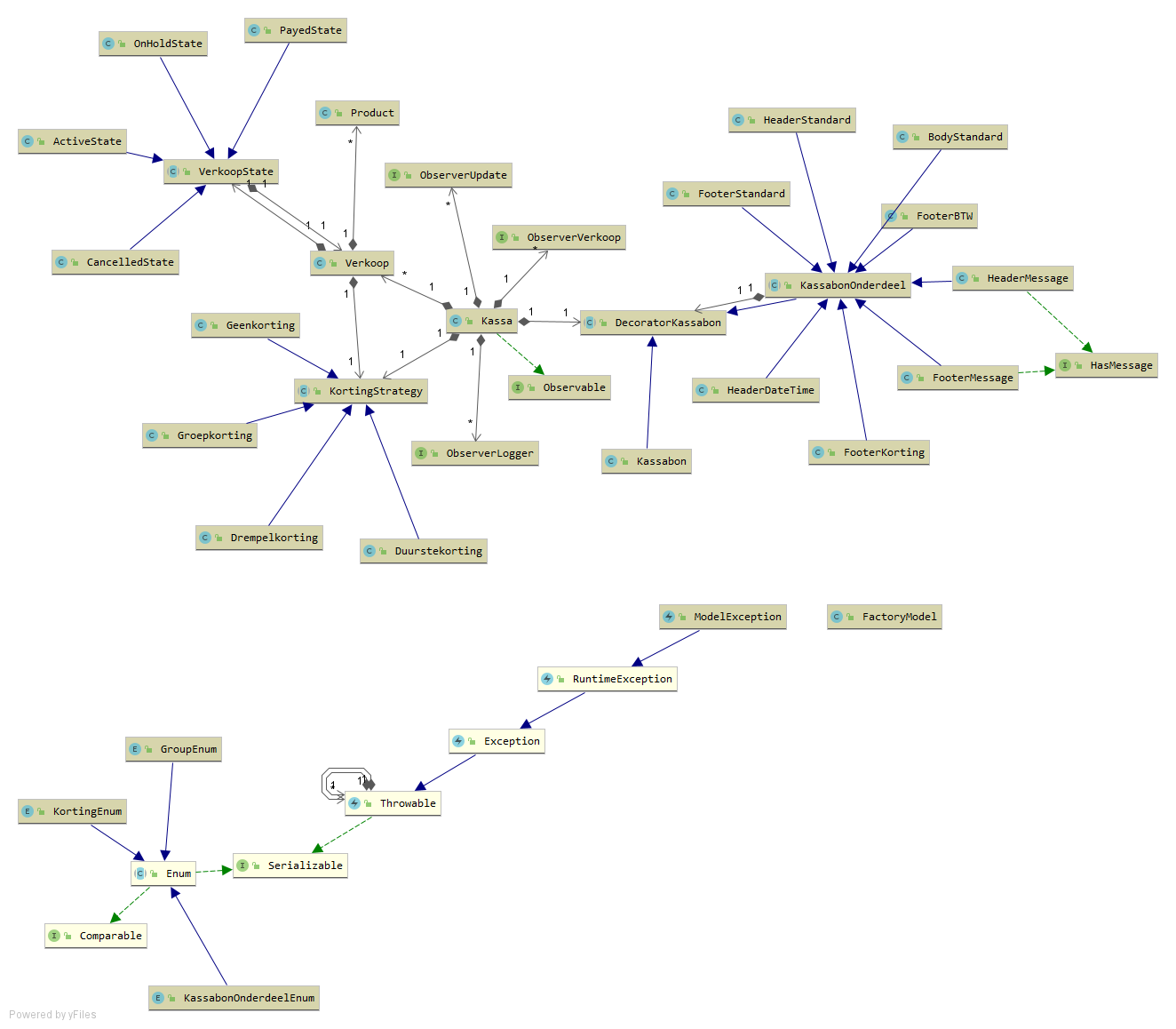
Geef aan in percentages hoeveel je bij benadering gespendeerd hebt aan deze opdracht.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Jacobs | Vandenplas | Vryghem | Totaal |
| Ontwerp | 33,33% | 33,33% | 33,33% | 100% |
| Klassendiagrammen | 32,5% | 32,5% | 35% | 100% |
| Implementatie | 40% | 30% | 30% | 100% |
| Verslag | 30% | 25% | 35% | 100% |

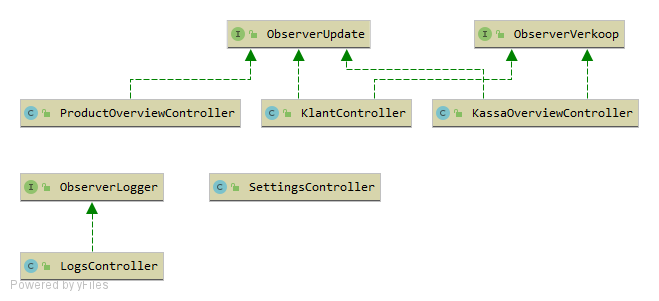
# Klassen diagramma’s

Voeg het definitieve gegenereerde klassendiagram van je code toe), als afzonderlijk afbeelding-bestand. **Dit klassendiagram moet goed leesbaar zijn.** Spreid het over meerdere pagina’s (bv. 1 blad per package (MVC) en 1 overzicht klassendiagram (zonder attributen en methoden). GEEN KLASSENDIAGRAM MET ZWARTE ACHTERGRONKLEUR AFPRINTEN!!!!!!!!!!

Model:



Controllers:



View: