

**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 |  |

# 

|  |  |  |
| --- | --- | --- |
|  | Toegepast (ja/nee)  In welke stories(nr)  Waarom toegepast(voordeel) | Bijhorend klassendiagram (gegenereerd vanuit je java code) |
| Observer | Ja  Story 4: wanneer er een product aan een verkoop toegevoegd wordt moet de lijst geüpdatet worden.  Story 6: nadat verkoop on hold wordt gezet 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 afsluit menu of naar het input menu  Het voordeel is dat het programma niet constant zal moeten kijken of er een product toegevoegd werd aan de lijst of als er op een knop gedrukt werd. | Afbeelding met schermafbeelding  Automatisch gegenereerde beschrijvingAfbeelding met kaart, tekst  Automatisch gegenereerde beschrijvingAfbeelding met schermafbeelding  Automatisch gegenereerde beschrijving |
| Strategy | Ja  Story 1: voor het type database  Story 2: voor het inladen van de producten vanuit een bestand  Story 7: gebruikt om verschillende soorten kortingen te implementeren  Het voordeel is dat je gemakkelijk kan kiezen welke methodes je wilt gebruiken om iets uit te voeren | Afbeelding met schermafbeelding  Automatisch gegenereerde beschrijvingAfbeelding met schermafbeelding  Automatisch gegenereerde beschrijving |
| Simple Factory | Ja  Story 2: voor het maken van ProductDBstrategy en voor LoadSaveStrategy  Story 7: voor het maken van KortingStrategy  Afbeelding met schermafbeelding  Automatisch gegenereerde beschrijvingStory 10: voor het maken van het kassaticket | Afbeelding met schermafbeelding  Automatisch gegenereerde beschrijving |
| Façade | Ja, alle stories. De kassa is een facade, de controllers weten alleen maar van kassa af.  Het voordeel hiervan is dat de UI maar van één klasse in het model moeten afweten |  |
| Singleton | Ja, story 2, 7 en 10 voor elke factory.  Zodat je maar één instantie hebt voor je factory | Afbeelding met schermafbeelding  Automatisch gegenereerde beschrijvingAfbeelding met schermafbeelding  Automatisch gegenereerde beschrijving |
| State | Ja, story 9  Een verkoop kan in verschillende states zijn actief, geannuleerd, betaald of on hold .  Het voordeel aan states te gebruiken is dat verkoop ander gedrag heeft als hij een andere state heeft. |  |
| MVC | Ja, overal  Voor elke view is er een controller die de view met het model verbindt.  Het voordeel hiervan is dat je afzonderlijk aan het model kunt werken. Je moet je geen zorgen maken over de view en controller | Afbeelding met schermafbeelding  Automatisch gegenereerde beschrijving  Afbeelding met schermafbeelding  Automatisch gegenereerde beschrijving  Afbeelding met schermafbeelding  Automatisch gegenereerde beschrijving  Afbeelding met schermafbeelding  Automatisch gegenereerde beschrijving |
| 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. | Afbeelding met schermafbeelding  Automatisch gegenereerde beschrijving |
| Template method | Ja  Story 1:  Zodat je later gemakkelijk bijvoorbeeld een klasse PersoneelTekstLoadSave  Kunt maken. | Afbeelding met schermafbeelding  Automatisch gegenereerde beschrijving |

# 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.

# 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 2, 7 en 10. Hier zijn Enums gebruikt zodat we reflection konden gebruiken in de factory | Afbeelding met schermafbeelding  Automatisch gegenereerde beschrijving |
| Properties | Ja, in story 2, 7 en 10. Hiermee worden de instellingen bijgehouden zelf na het afsluiten van de kassa app | Afbeelding met schermafbeelding  Automatisch gegenereerde beschrijving |
| Reflection | Ja, in story 2, 7 en 10. We hebben reflection gebruikt in de factory | Afbeelding met schermafbeelding  Automatisch gegenereerde beschrijving |

# Werkverdeling

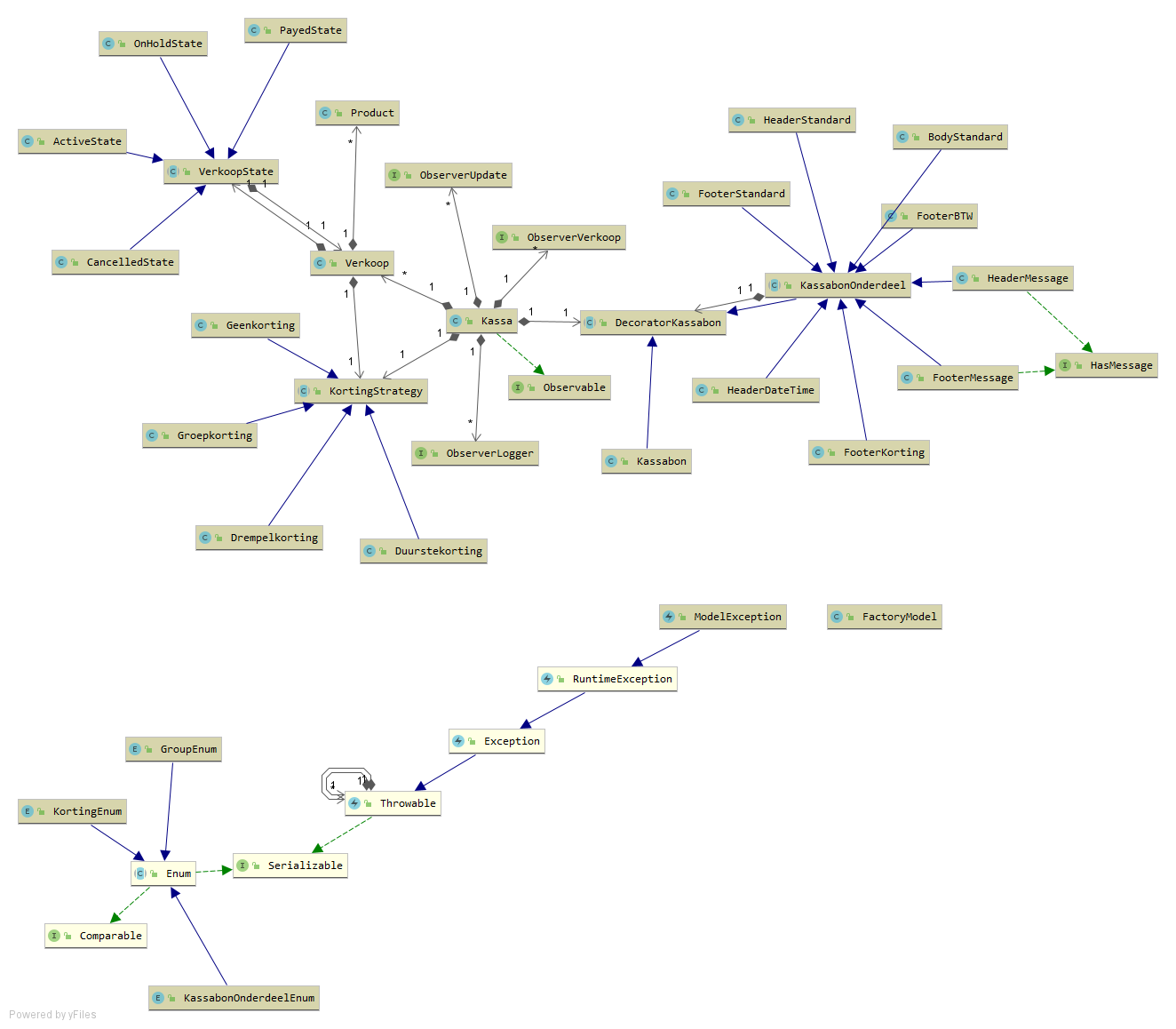
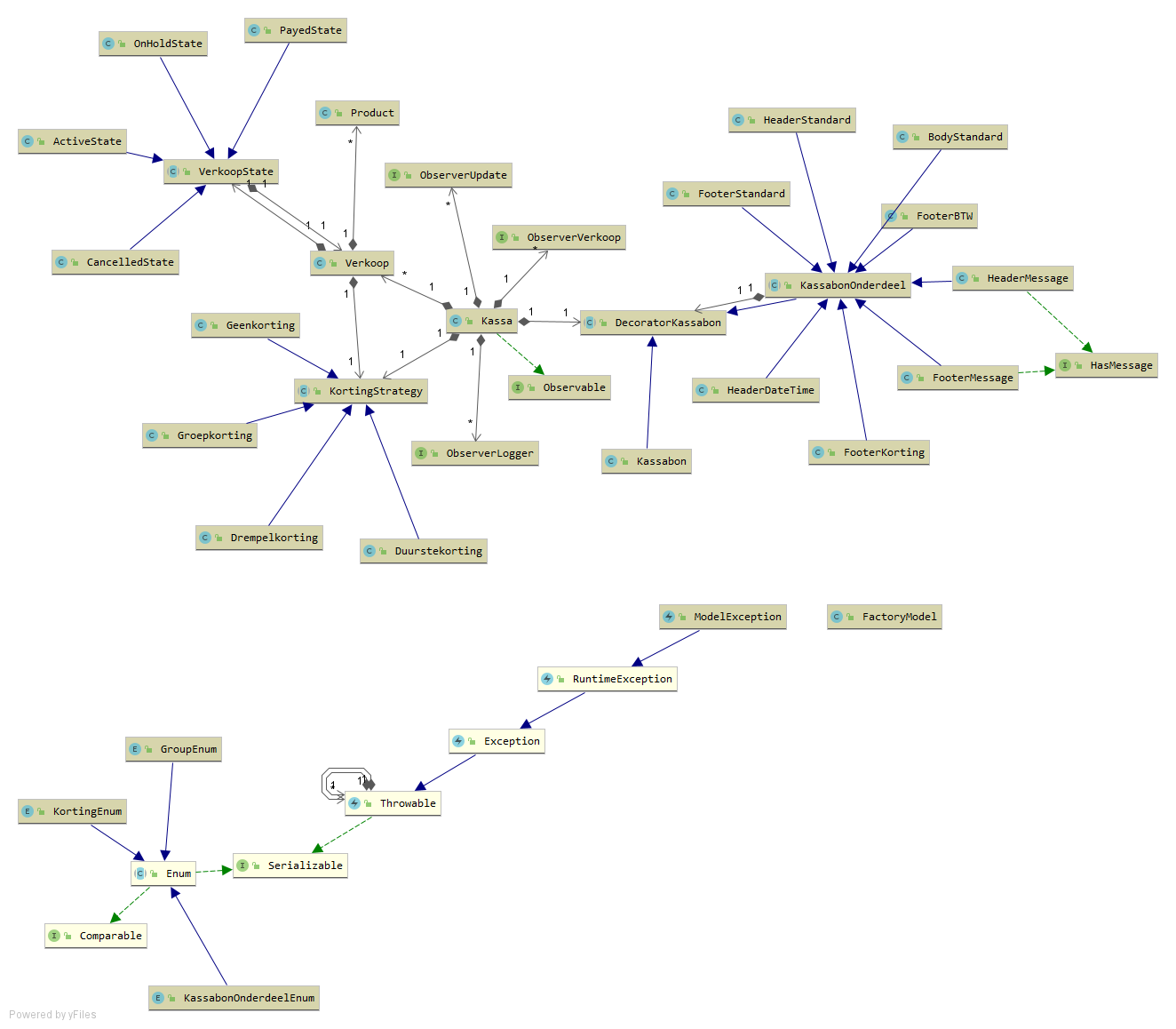
Geef aan in percentages hoeveel je bij benadering gespendeerd hebt aan deze opdracht.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 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% |

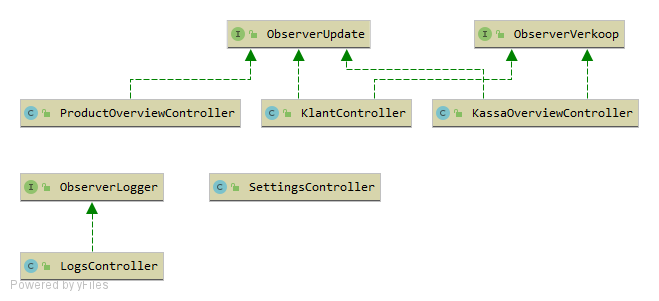
# Klassen diagramma’s

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: