

**Object Georiënteerd Ontwerp**

**Kassa App**

Boussabat Reda

Geysegoms Andreas

Vanheel Thomas

2019, Toegepaste Informatica en 7-8 NS-BS

23/11/2019

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

https://github.com/13r44m1335/2\_Boussabat\_Geysegoms\_Vanheel\_KassaApp\_2019

# 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 | 1. Communicatie tussen winkel en product overview pane.  5. Verwijderen van een artikel  6. Hold zetten van verkopen.  9. Loggen van verkopen. |  |
| Strategy | 1. Rekening houdend op uitrbreiding van invoer  2. Idem (1)  7. Uitbreidbaarheid nieuwe korting  10. Uitbreidbaarheid nieuwe headers of footers |  |
| Simple Factory | 2. Aanmaken / inlezen van artikels.  7. Aanmaken van korting.  10. Aanaken van rekening |  |
| Façade |  |  |
| Singleton | 2. Aanmaken / inlezen van artikels.  7. Aanmaken van korting  10. Aanmaken vna rekening |  |
| State | 9. Bij te houden in welke staat de verkoop is |  |
| MVC | Controllers als middel om view en model te scheiden. |  |
| Decorator | 10. Aanmaken van rekeningen en tijdens runtime wijzigen. |  |
| Template method | 1. inladen en wegschrijven als 1 functie in subklassen. |  |

# 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 | 1. type observer  2. type input file  7. type korting |  |
| Properties | 2. properties input file eenvoudig opslaan.  10. Rekening opties opslaan in properties. |  |
| Reflection | 2. Aanmaken / inlezen van artikels.  7. Aanmaken van korting |  |
| Andere… |  |  |

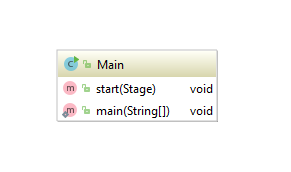
# Werkverdeling

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

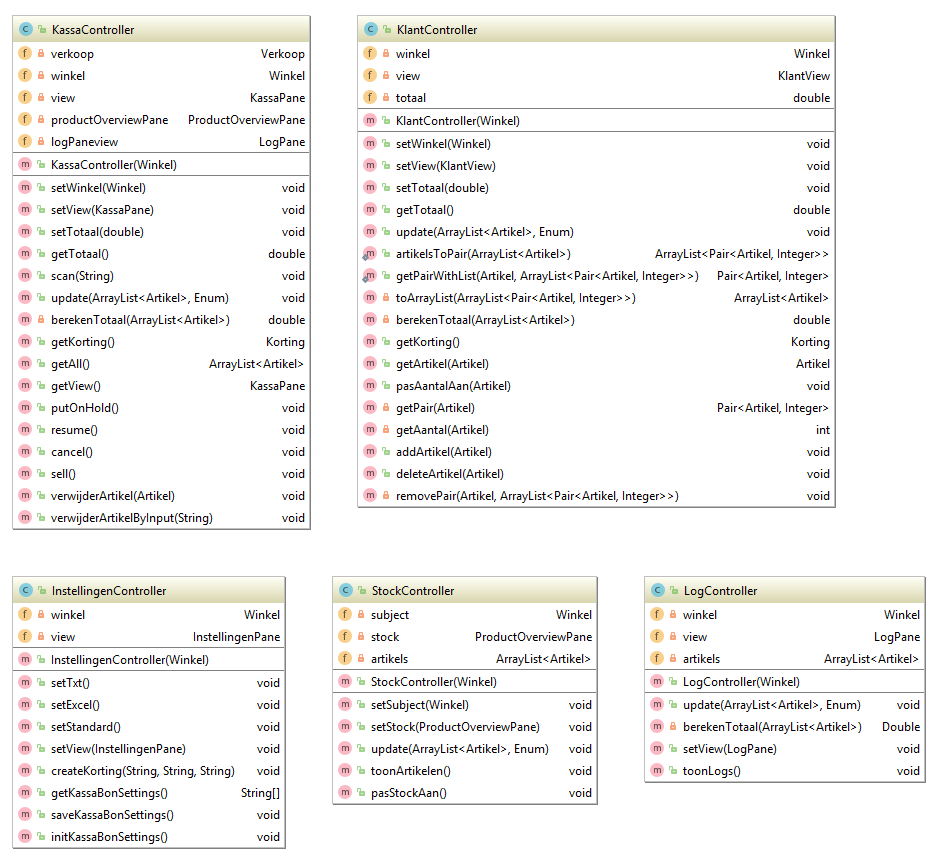
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Boussabat | Geysegoms | Vanheel | Totaal |
| Ontwerp | 10% | 70% | 20% | 100% |
| Klassendiagrammen | 28,5% | 57% | 14,5% | 100% |
| Implementatie | 21,4% | 43,6% | 35% | 100% |
| Verslag | 10% | 80% | 10% | 100% |

# Klassen diagramma’s

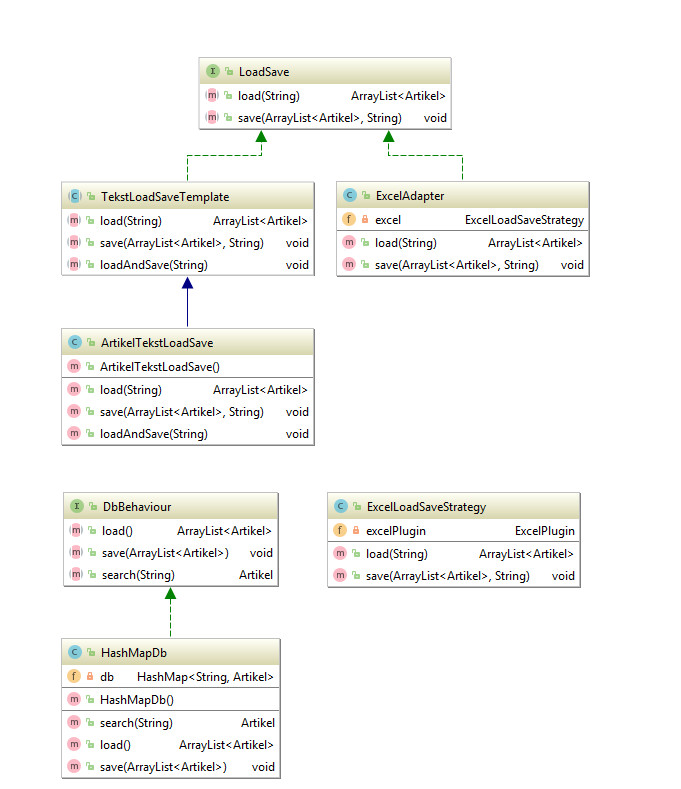
Application:



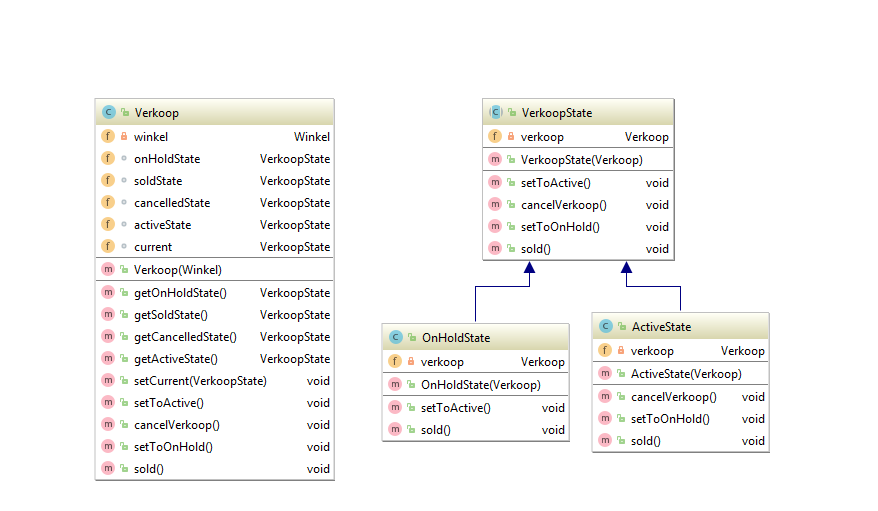
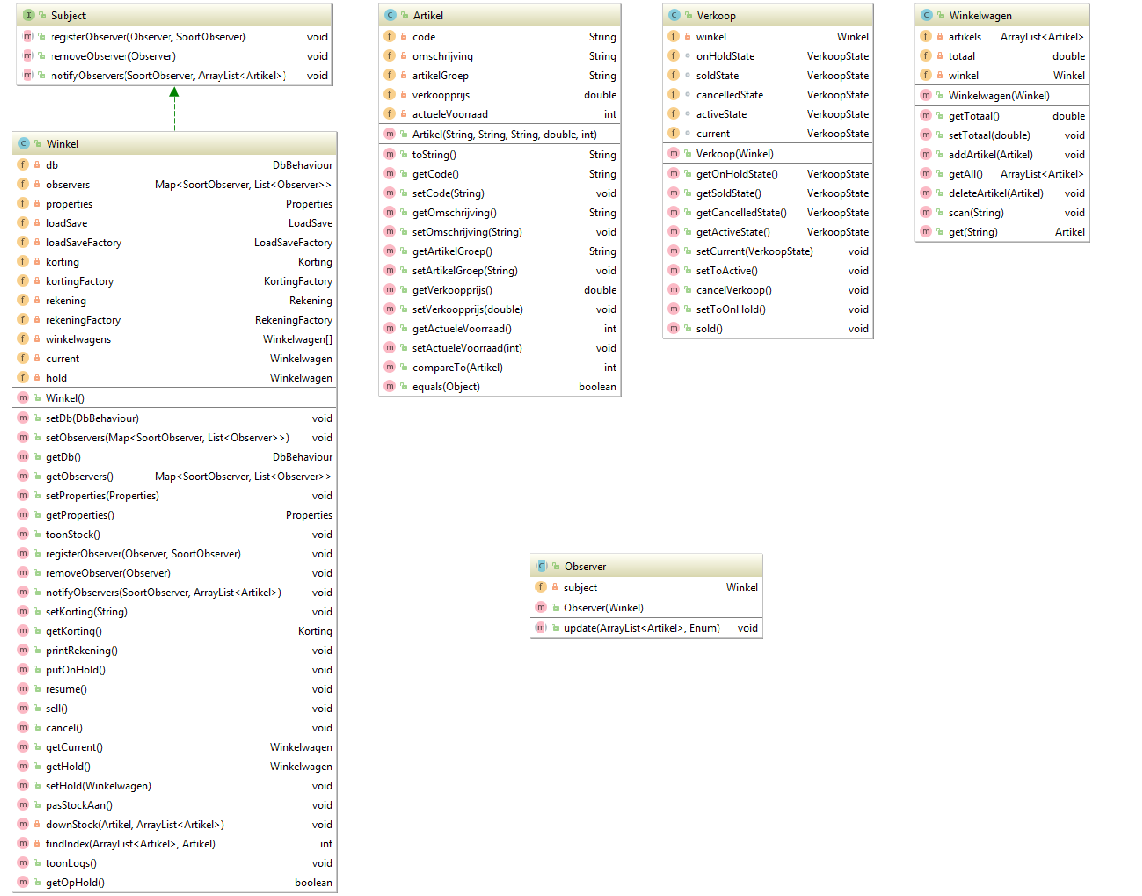
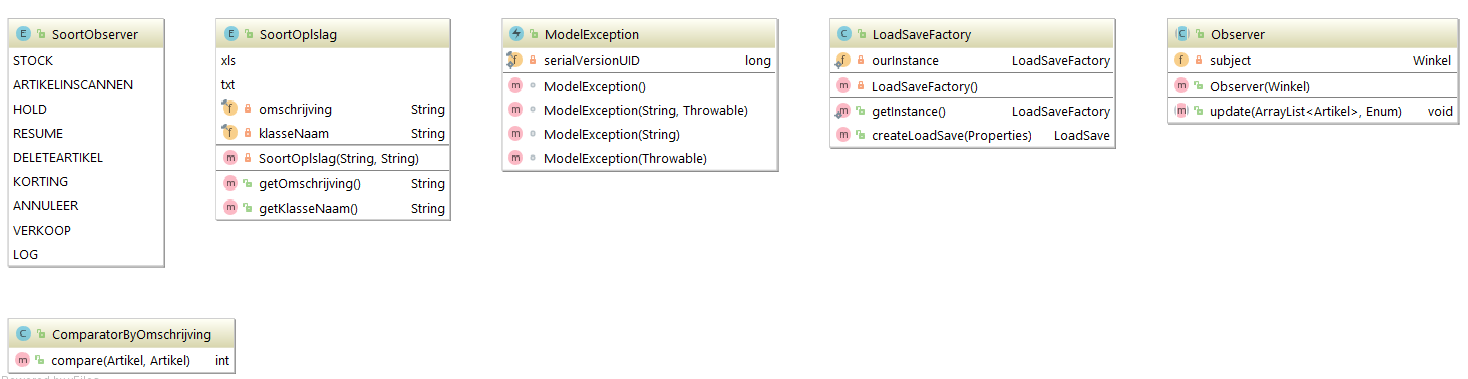
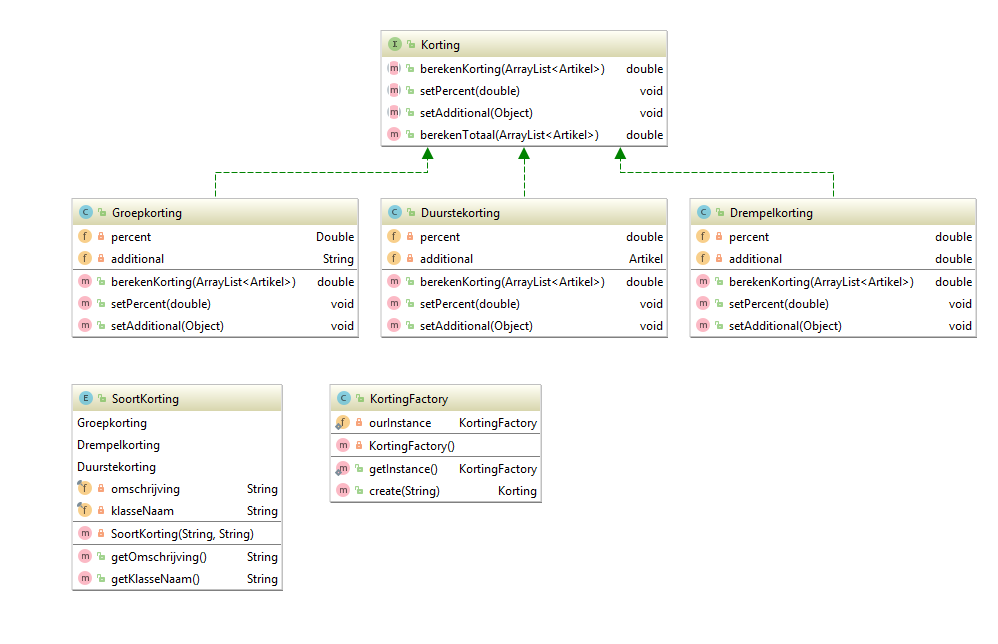
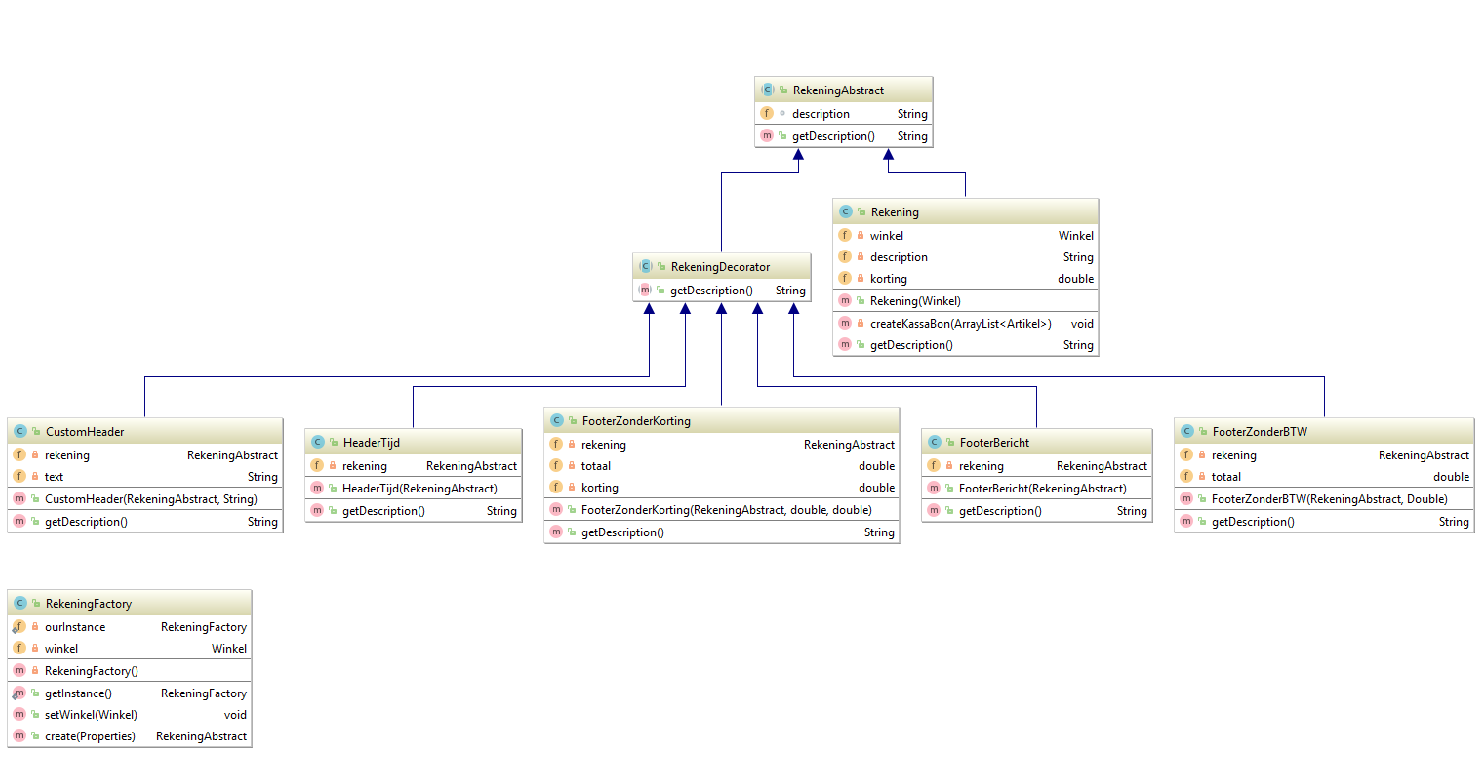
Controller:



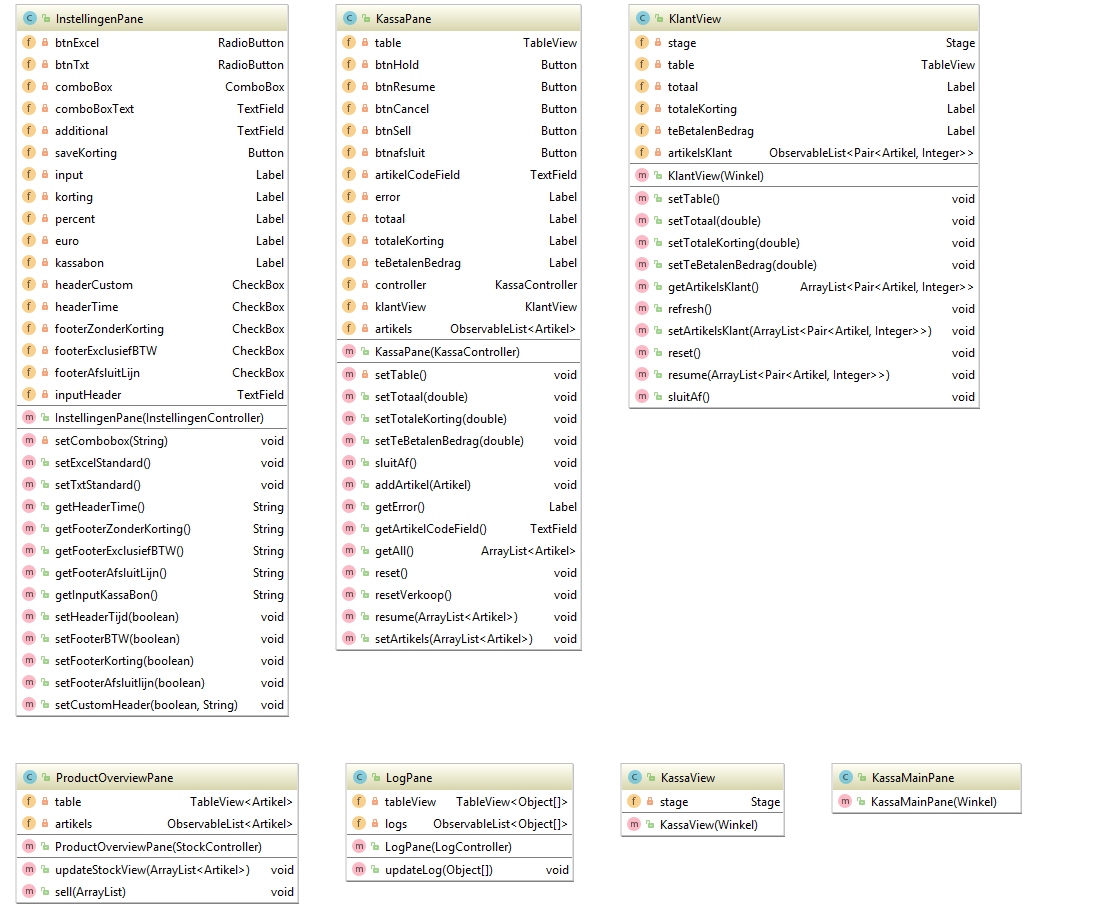
Database:



Model:



View:



Algemeen:

