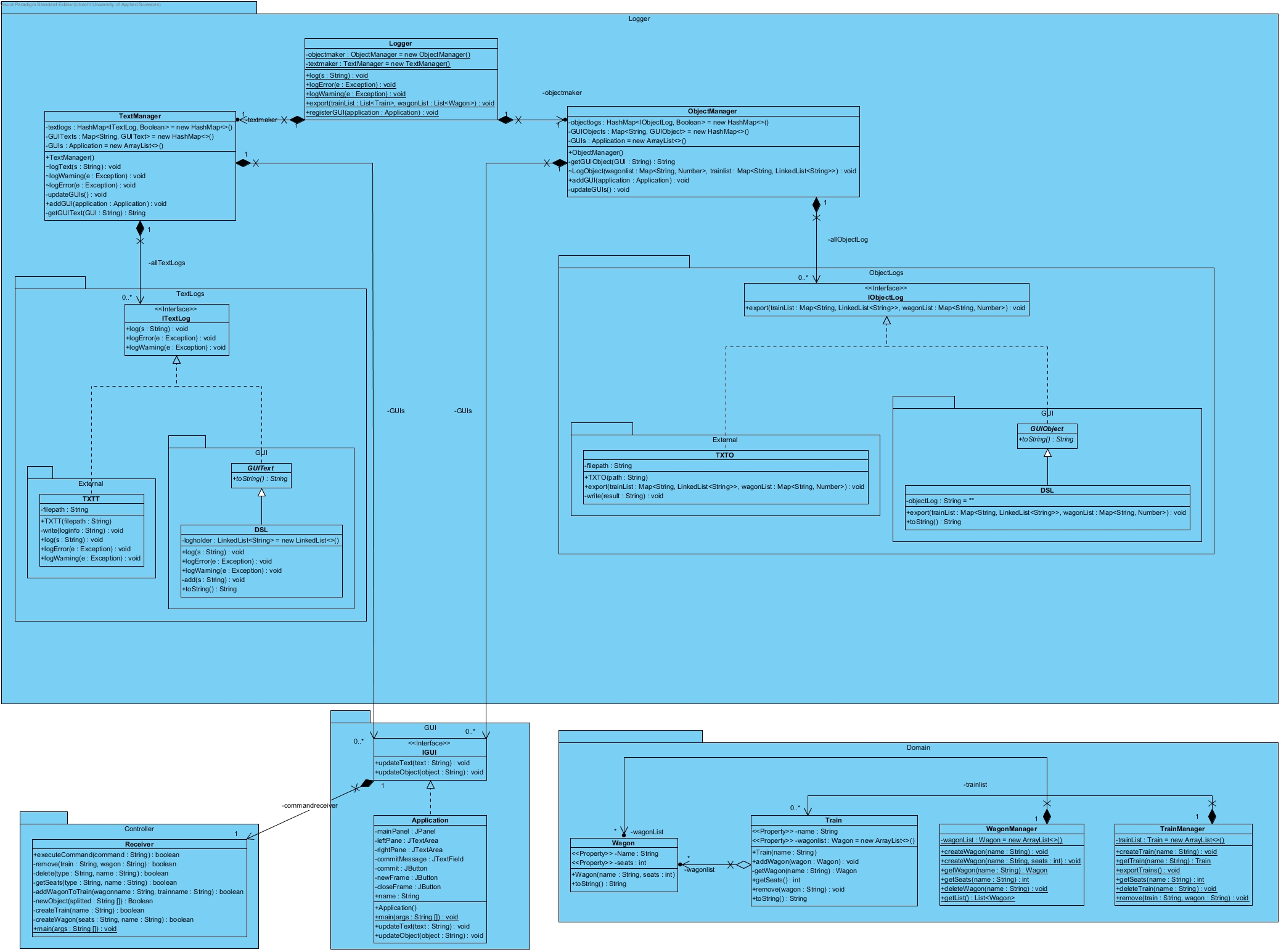
Motivatie Code



(klassendiagram: model.jpg)

**Voor een beter kwaliteit en inzoomen, zie model.jpg.**

# GUI

Interface IGUI zorgt voor makkelijk gebruik van verschillende GUI’s zonder dat er wijzigingen buiten deze layer nodig is. Door een observer pattern met Textmanager en ObjectManager worden de GUIoutputs up to date gehouden. Dit samen zorgt ervoor dat er voor het toevoegen van een andere soort GUI alleen een klasse hoeft aangemaakt worden aan IGUI en (als de binnenkomende Text en Object anders gegeven moet worden) een klasse aan GUIText en GUIObject. (Open-closed principle)

# Controller

Receiver krijgt commando’s binnen van GUI’s, decodeert ze en stuurt de acties door naar de juiste domein-klassen.

# Domain

Trainmanager en WagonManager gebruiken de factory pattern op train en wagon voor het beheren van objecten van deze klassen. Dit zorgt er ook voor dat alle klassen maar één taak hebben (Single-responsibility Principle)

Acties die effect hebben op trains of wagons sturen een commando door naar logger om de overzichten (object genoemd) up te daten.

# Logger

Logger stuurt de Textoutputs door naar de textmanager en de Objectoutputs naar de objectmanager. Ook GUI observer registraties worden hier doorgestuurt naar beide managers. Dit samen zorgt dat de logger de enige aanspreekpunt is voor klassen buiten deze layer, die de rest van deze layer aanstuurt.

Omdat Textoutputs en Objectoutputs voor GUI’s en externe files dezelfde kennis (mogen) hebben en binnenkrijgen, worden deze (als ITEXTLog en IObjectLog) bijgehouden door dezelfde manager (TextLogs en ObjectLogs natuurlijk wel gescheiden) (Dependency Inversion principle). Omdat Externe log-schrijvers direct wegschrijven en GUIlogs intern opslaan om door te sturen naar de GUI’s worden GUI’s gescheiden door ze een sub-interface GUIobject/log te geven (Liskov substitution principle & Interface segregation principle). Bij nieuwe outputs sturen de managers de informatie gelijk door naar de subscribed GUI’s via het observer-pattern.

IObjectLogs en ITextLogs zijn adapters voor de data om ze op de juiste manier weg te schrijven of door te geven aan de GUI. (Open-closed principle).

**De SOLID Principles zijn vaak meerdere keren op dezelfde plekken terug te vinden.**