Entwurfsdokumentation

Visuelle Programmiersprache für den Physikunterricht zur Datenerfassung auf einem Raspberry Pi

Version 0.0.0

David Gawron Stefan Geretschläger Leon Huck
Jan Küblbeck Linus Ruhnke

30. Juni 2019

Inhaltsverzeichnis

1 Ziel der Entwurfsdokumentation

Die Entwurfsdokumentation soll, aufbauend auf das Pflichtenheft, Entwurfsentscheidungen festhalten. Der Rahmen des Entwurfes wird durch einen *Model-View-Controller* (MVC) gebildet. Die Daten werden durch das Backend zu der Verfügung gestellt. Jedes dieser Pakete kommuniziert über eine Fassade. Dadurch werden die Pakete von einander abgekoppelt. Durch diesen grundlegenden Aufbau wird die Software in vier unabhängige Komponenten aufgeteilt, die unabhängig voneinander implementiert und später erweitert werden können.

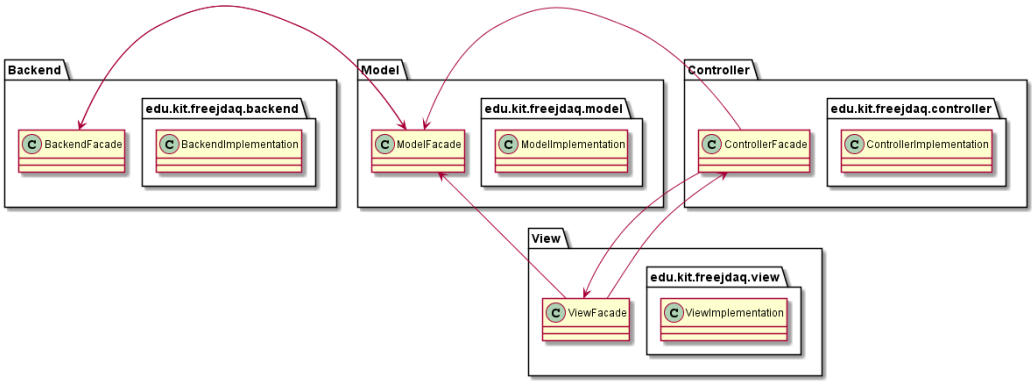

Abbildung 1: Die grobe Struktur des Entwurfs

2 Klassenbeschreibung

Im folgenden sollen alle Klassen mit ihren Funktion beschrieben werden. Der Aufbau orientiert sich dabei an der in **??** aufgeführten Struktur.

2.1 Backend

2.2 Model

2.3 Controller

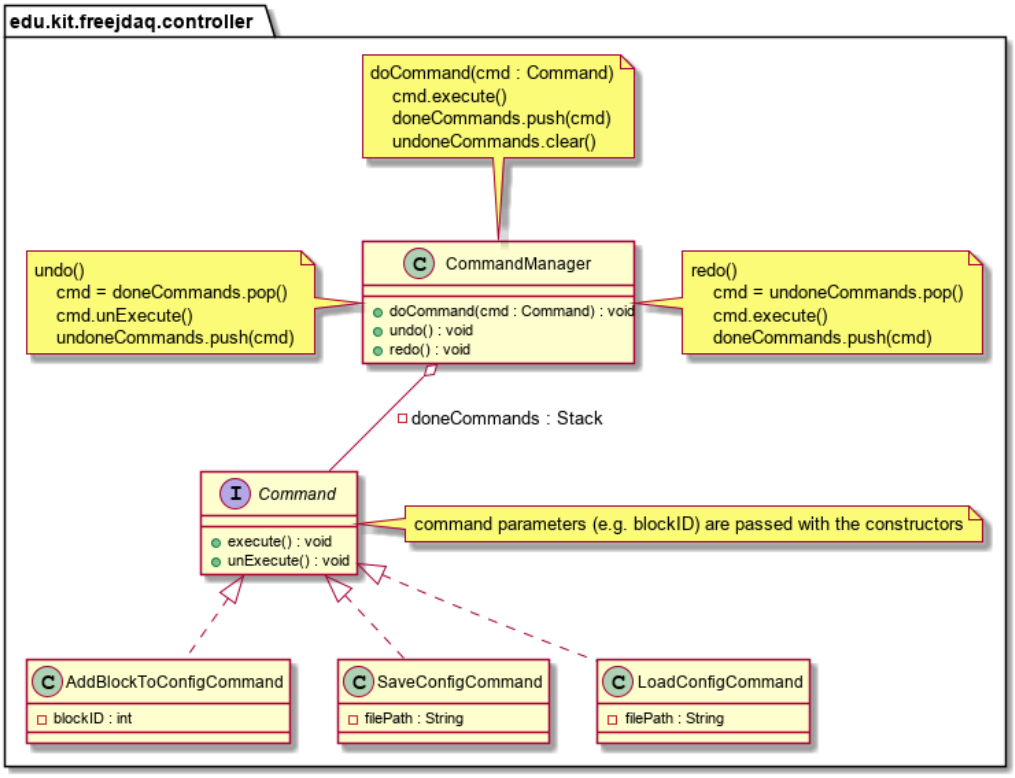

Abbildung 2: Die Struktur des Controllers

2.4 View

Das Paket View, stellt gemäß des MVC- Entwurfmusters die Darstellungen von Daten des Modells dar und realisiert Benutzerinteraktionen.

2.4.1 MainWindow

Die Klasse MainWindow stellt den Rahmen der Benutzeroberfläche dar. Da MainWindow, dass Entwurfsmuster Singleton verwendet, kann die Anwendung nur ein MainWindow besitzen soll.

2.4.2 ConfigurationField

Die Klasse KonfigurationField stellt das Konfigurationfeld dar, in welchem der Benutzer eine Messkonfiguration aufbauen kann. Konfigurationsbausteine, welche der Benutzer in das Konfigurationsfeld plaziert werden in einer Liste gespeichert. Konfigurationsbausteine, welche der Benutzer aus dem Konfigurationsfeld entfernt, werden aus der Liste gelöscht Beim Plazieren der Konfigurationsbausteine in das Konfigurationsfeld wird dem Konfigurationsbaustein eine eindeutige Postition zugeteilt, welche in Form von x- und y-Koordinaten darsgestellt wird.

2.4.3 BuildingBlockView

Die Klasse BuildingBlockView ist die Überklasse der Darstellungen der Konfigurationsbausteine. Konfigurationsbausteine besitzen eine eindeutige ID, einen Namen, falls sie im Konfigurationsfeld platziert werden ihre Position anhand der Koordinaten x und y. Form und Farbe sind ebenfalls festegelegt.

2.4.4 SensorBlockView

Die Klasse SensorBlockView stellt einen Sensorbaustein dar. Sensorbausteine, welche in dem Konfigurationsfeld platziert werden, können mit anderen Bausteinen verbunden werden, was im Messlauf einen Datenfluss über die verbundenen Bausteine erlaubt. Sensorbausteine besitzen, im Gegensatz zu anderen Konfigurationsbausteinen nur Datenausgänge, über welche sie verbunden werden können.

2.4.5 TransformationBlockView

Die Klasse TransformationBlockView stellt einen Transformationsbaustein dar. Transformationsbausteine besitzen eine vordefinierte Funktion, welche die Messdaten verändern.

2.4.6 RepresentationBlockView

Die Klasse RepresentationBlockView stellt einen Darstellungbaustein dar, dieser bestimmt, wie die Messdaten visualisiert werden. - Er bestitzt nur Eingänge, aber keine Ausgänge

2.4.7 PrototypeField

Die Klasse PrototypeField ist die Über-Klasse zu SensorBlockField, TransformationBlockField und RepresentationBlockField. Sie stellt eine Fläche dar, in welcher vordefinierte Konfigurationsbausteine dargestellt werden und der Benutzer sie mit dem Mauszeiger in das Konfigurationsfeld ziehen und damit positionieren kann.

3 Sequenzdiagramme

4 Änderungen am Pflichtenheft

5 Formale Spezifikationen von Kernkomponenten

6 Weitere UML Diagramme

7 Anhang

7.1 Vollständiges Klassendiagramm

8 Glossar