Geschreven door Lars van Vliet

Versie 1.2

Calculator

Technisch ontwerp

# Revisietabel

|  |  |  |  |
| --- | --- | --- | --- |
| Versienummer | Naam | Aanpassingen | Datum |
| Versie 1.0  Versie 1.1  Versie 1.2 | Lars van Vliet  Lars van Vliet  Lars van Vliet | Basis opmaak  Klassendiagram bijgewerkt  Tekst bij structuur toegevoegd | **27-5-2019**  **29-5-2019**  **29-5-2019** |

Inhoudsopgave

[Toelichting gekozen oplossingen 1](#_Toc9845827)

[Eindproduct 2](#_Toc9845828)

[Objecten 3](#_Toc9845829)

[Kwaliteitseisen 4](#_Toc9845830)

[Afhankelijkheden 5](#_Toc9845831)

[Structuren 6](#_Toc9845832)

# Toelichting gekozen oplossingen

Het probleem is: het maken van een calculator. Deze calculator moet kunnen optellen, vermenigvuldigen, aftrekken en delen. Er wordt verwacht dat het een simpele interface krijgt. Een interface die je zonder handleiding kan gebruiken.

Daarom is er besloten om een schets te maken van het programma, deze is te vinden in het functioneel ontwerp.

De calculator wordt met het MVC patroon gemaakt, zodat de code opgedeeld kan worden in duidelijke delen. Dit maakt het makkelijker om de code te testen.

# Eindproduct

Het eindproduct is een calculator. Deze calculator kan simpele berekeningen uitvoeren en deze tonen op het scherm. Het is belangrijk dat de gebruikers zonder handleiding dit programma kunnen gebruiken.

# Onderdelen

Het programma bestaat uit de volgende klassen:

* Model
* View
* Controller
* Calculator

In het model wordt een calculator object aangemaakt, deze is verantwoordelijk voor het uitvoeren van berekeningen en deze ook te kunnen returnen.

In de view wordt het scherm geregeld, deze toont de ingevoerde toetsen en daarna laat hij het antwoord zien.

De controller regelt de input. Als er op een knop wordt gedrukt, regelt de controller welke methoden die knop oproept.

De calculator kan een berekening uitvoeren, opslaan en tonen. Het is handig om hier een aparte klasse voor te hebben, zodat het in het model wat overzichtelijker is.

# Kwaliteitseisen

In de code wordt er gebruik gemaakt van camelCase in zowel de methoden en variabelen.   
Voor de klassen wordt de eerste letter van elk woord een hoofdletter. En met tabs wordt de indentatie gedaan.

# Afhankelijkheden

De GUI wordt gemaakt met JavaFX, verder zijn er geen afhankelijkheden.

# Structuren C:\Users\larsv\Downloads\versie1.2.png

Zoals je kunt zien ‘kennen’ de model en view elkaar niet. Dit wordt allemaal geregeld via de controller. Het model regelt de berekeningen, de controller de input en de view toont gegevens op het scherm. Deze gegevens krijgt de view van de controller. Er is ook nog een calculator klasse gemaakt, deze doet de berekeningen.