

Grafikrechner Entwicklerdokumentation

Lars Strölin, Michael Geigges, Ilja Kononenko

10. Juli 2018

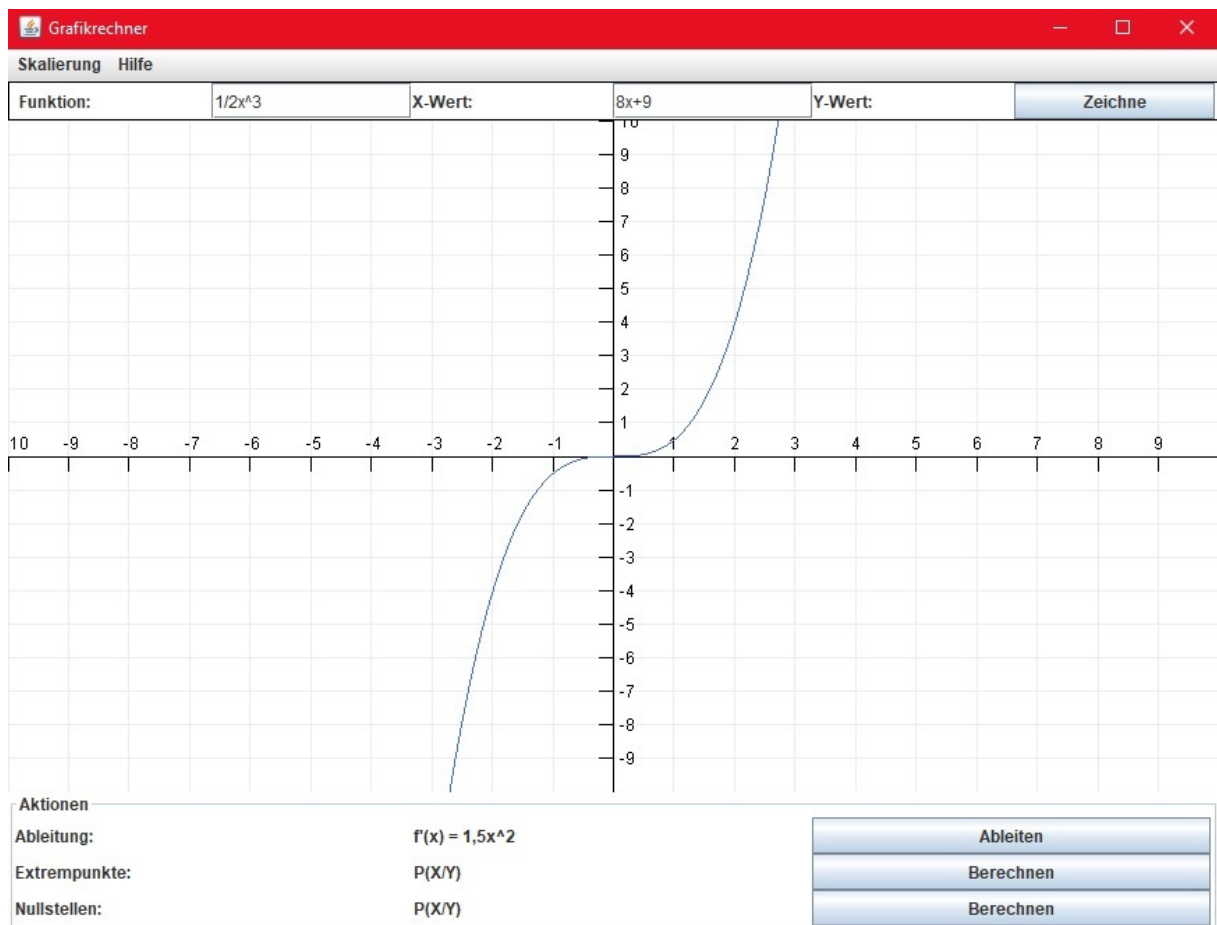


Abbildung 1: Grafikrechner

Inhaltsverzeichnis

1	Vorwort	2
2	Aufbau	3
2.1	Klassendiagramm	3
3	Methoden	4
3.1	Funktionsgleichung zerlegen	4
3.2	Berechne Y-Wert	5
3.3	Bestimme Exponent	6
3.4	Bestimme Konstante des Produkts	7
3.5	Überprüfe Funktion	8
3.6	Bestimme Ableitung	9
4	Fazit	10

1 Vorwort

Diese Entwicklerdokumentation dient allein nur für die reibungslose Weiterentwicklung bzw. zeigt den aktuellen Aufbau des Grafikrechner-Programms.

Hier werden die jeweiligen Themen bzw. Arbeitsfunktionen anhand von Klassendiagrammen, Sequenzdiagrammen und falls notwendig Struktogrammen erklärt und wichtige Funktionen dargestellt, sowie Ansatzpunkte zur Weiterentwicklung weiter gegeben. Damit der nachfolgende Entwickler die Arbeit von uns schnell nachvollziehen und in diese leicht einsteigen kann, wird die Erklärung so kurz wie möglich gehalten und nur das Wichtigste erklärt. Daher sind alle leicht nachvollziehbaren Sachen vom Quellcode bzw. vom Javadoc zu entnehmen. Am Ende dieser Entwicklerdokumentation werden die größten Schwierigkeiten und Probleme aufsummiert und geschildert, damit der nachfolgende Entwickler davon eine grobe Einschätzung bekommen kann, was für Problematiken auf ihn in diesem Projekt zu kommen werden.

2 Aufbau

Um im Programm eine klare und geordnete Struktur zu haben, war es notwendig viele Klassen zu erstellen, um so jede Funktion des Programms geordnet und einzeln bearbeiten zu können und um das ganze Programm ohne Probleme in richtiger Art und Weise auszuführen. Deren Funktion dient größtenteils dazu um spätere Fehler schneller und einfacher zu lösen, sowie ein Überblick vom Code zu bekommen.

Die wichtigste Klasse ist selbstverständlich die Steuerungsklasse. In ihr befinden sich alle wichtigen Algorithmen für die mathematischen Formeln, sowie die Methoden welche in die anderen Klassen implementiert werden.

Die Implementation der Methoden werden im Folgenden durch Sequenzdiagramme veranschaulicht und die schwierigen Algorithmen mit Struktogrammen dargestellt und zusätzlich erklärt.

2.1 Klassendiagramm

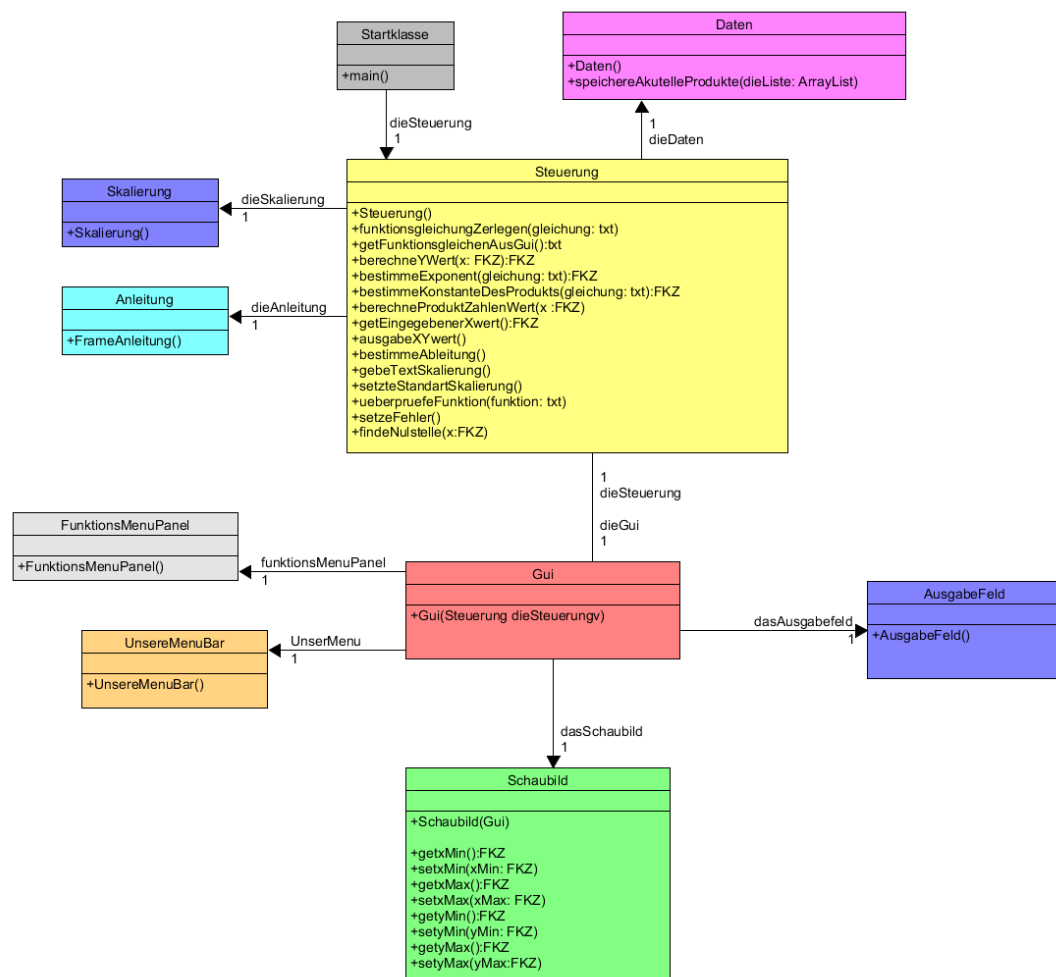


Abbildung 2: Programm in der Übersicht

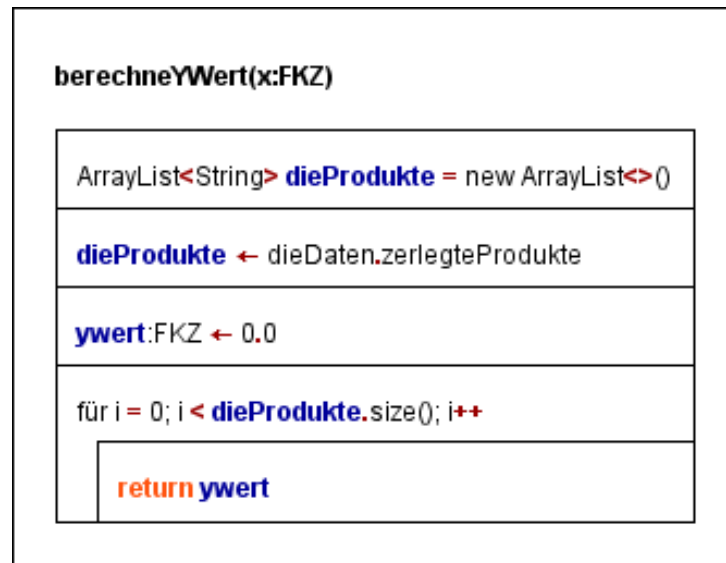
3 Methoden

3.1 Funktionsgleichung zerlegen



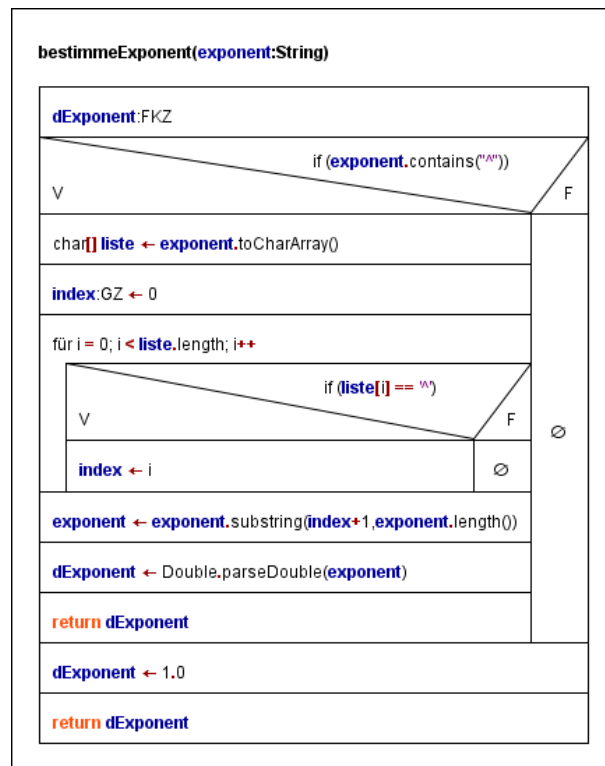
In der Methode funktionsgleichungZerlegen wird die eingegebene Funktion nach den Rechenzeichen der Addition und Subtraktion zerlegt. Beispielsweise $2x^2 - 3x$ wird zerlegt in $2x^2$ und $-3x$. Diese zerlegten Strings werden dann in eine ArrayList gespeichert, die in der Datenklasse übergeben wird.

3.2 Berechne Y-Wert



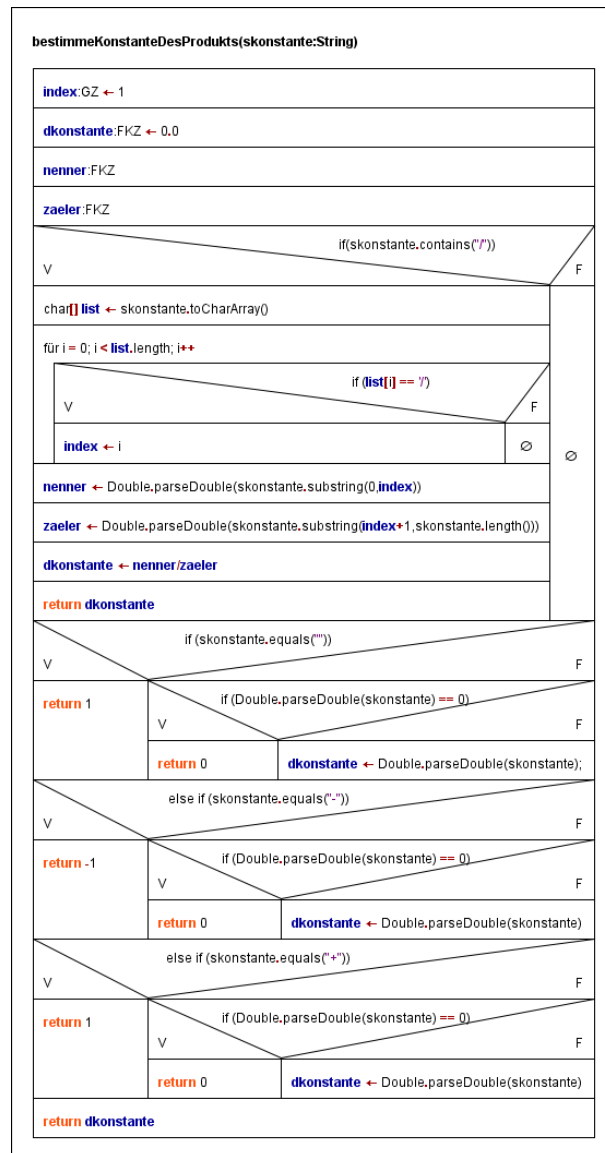
In der `berechneYWert` Methode wird der Y-Wert für einen bestimmten Zahlenwert berechnet. Dabei ruft die Methode die Daten der zerlegten Funktionsgleichung ab und berechnet für jede Funktion einen Y-Wert. Dieser Y-Wert wird dann mit dem des vorherigen addiert. Die Methode ist dann komplett durchgelaufen sobald der Y-Wert für alle Funktionen berechnet wurde.

3.3 Bestimme Exponent



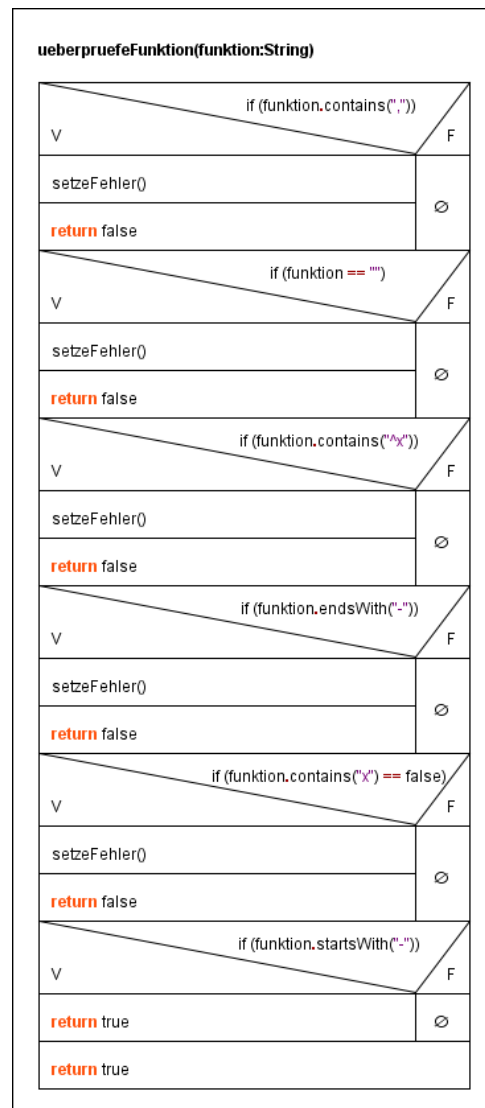
In der Methode `bestimmeExponent` wird bestimmt welcher Funktionsgrad vorliegt. Dabei wird die jeweilige Funktion, nach einem Hochzeichen untersucht und danach anhand der gefundenen Stelle, die Hochzahl bestimmt und als Kommazahl umformatiert.

3.4 Bestimme Konstante des Produkts



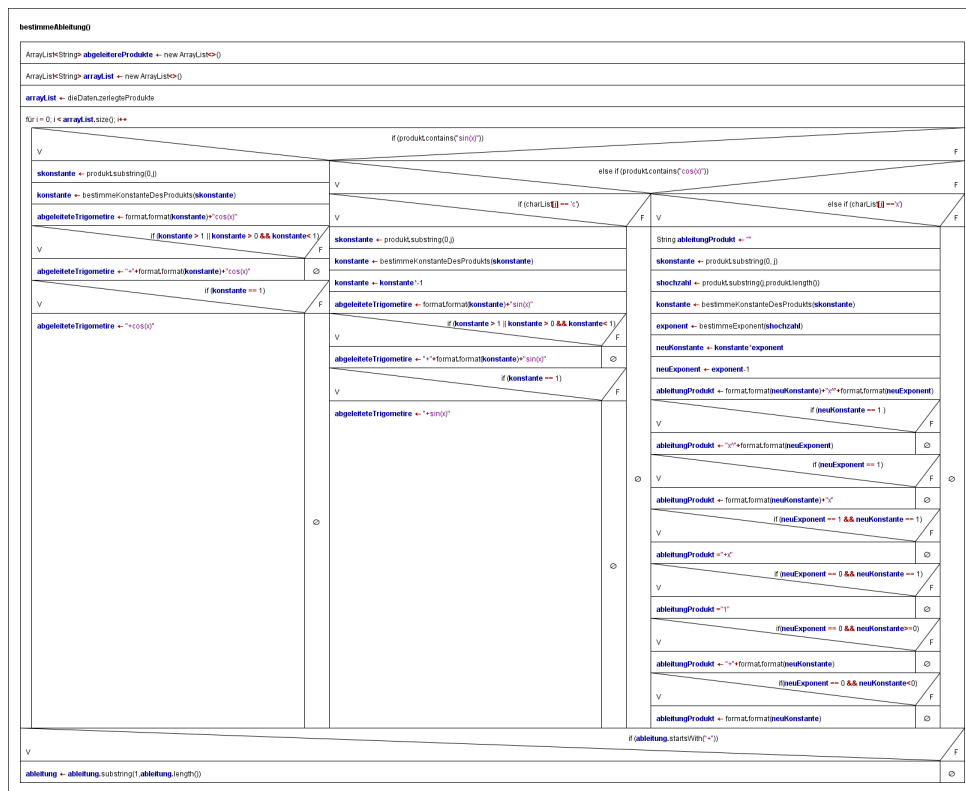
Die Methode `bestimmeKonstanteDesProduktes` bestimmt die jeweilige Konstante eines einzelnen Produktes. Beispielsweise bei $3x^2$ ist die Konstante 3.

3.5 Überprüfe Funktion



Die Methode ueberpuefeFunktion überprüft die Funktionsgleichung auf Fehler wie beispielsweise ein Komma statt Punkt oder das verwendet einer falschen Variable. Sie ist zurzeit jedoch nicht aktiv da sie Probleme bei $\sin(x)$ und $\cos(x)$ verursacht.

3.6 Bestimme Ableitung



Die Methode `ableitung` ermöglicht es Funktionen abzuleiten. Hierbei wird die zerlegte Funktion aus der Datenklasse verwendet. Von der zerlegten Funktion werden jeweils die Konstante und der Exponent berechnet und je nach Funktionstyp abgeleitet. Bisher werden nur GRF und Trigonometrische Funktionen in der Normalform unterstützt.

4 Fazit

Im Großen und Ganzen ist es uns gelungen ein Programm zu entwickeln, welches durch Eingabe einer Funktion diese in einem Schaubild korrekt darstellt und dazu auch besondere Werte, wie z.B. Nullstellen und Extrempunkte ausrechnet. Es war aber dennoch für uns sehr schwierig diese Aufgabe zu realisieren, da wir uns erst Gedanken machen mussten, wie wir die Erstellung unseres Programms angehen und vor allem wie wir die mathematischen Formeln programmieren.

Die größten Schwierigkeiten und Problematiken waren deshalb die Codes für die Ausrechnung der besonderen Werte für unser Programm zu entwickeln. Doch genau aus diesem Grund sind wir zufrieden es geschafft zu haben und können jetzt im Programm Verbesserungen durchführen. Die Oberfläche muss auf jeden Fall noch benutzerfreundlicher gestaltet und zum Teil automatisiert werden.

Außerdem hatten wir uns eigentlich vorgenommen eine Funktion zu entwickeln, in der man bestimmte Funktionen abspeichern und wieder aufrufen kann, da uns aber die Zeit dazu gefehlt hat und diese Funktionen optional waren, fielen sie für die Abgabe des jetzigen Programms raus.