

Pflichtenheft

Inhalt

1. Zielbestimmung.....	2
2. Produkteinsatz.....	2
2.1.Regula Falsi.....	2
2.2.Sekantenverfahren	3
2.3.Newton-Verfahren	3
2.4.Bisektion	4
2.5.Euler-Tschebyschow-Verfahren	4
2.6.Steffensen-Verfahren	4
2.7.Beschreibung des Problembereichs.....	4
3. Produktfunktionen	5
3.1. Use Case Diagramme	5
3.2. Möglichkeiten des Benutzers	5
3.2.1. Funktion eingeben	5
3.2.2. Anzahl Kommastellen eingeben	5
3.2.3. Start- und Endwert festlegen	5
3.2.4. Anzahl der Einträge festlegen.....	6
3.2.5. 3 Verfahren auswählen.....	6
3.3. Aufgaben des Systems.....	6
3.3.1. Berechnung mit gewählten Verfahren	6
3.3.2. Visualisierung der Funktion	6
4. Reengineering (Ist-Zustand)	7
5. Grobentwurf (Soll-Zustand)	8
6. Qualitätsmerkmale.....	9
6.1. Typ USE: Benutzbarkeitsanforderung	9
6.2. Typ EFFIZIENZ: Effizienzanforderung	9
6.3. Typ PFLEGE: Wartbarkeits- und Portierbarkeitsanforderung	9
6.4. Typ SICHER: Sicherheitsanforderung	9
6.5. Typ LEGAL: Gesetzliche Anforderung.....	9
7. Umsetzungsdetails	9
8. Projektplan	9

Projekt: Näherungsverfahren

Auftraggeber: Gaby Campidell

Oberschulzentrum J. Ph. Fallmerayer
+39 349 602 9960

Auftragnehmer: CDT

Oberschulzentrum J. Ph. Fallmerayer
Dantestraße, 39E, 39042 Brixen

1. Zielbestimmung

Unser System behandelt das mathematische Thema *Näherungsverfahren*.

Das System beinhaltet die Berechnung, sowie die Visualisierung.

Die Näherungsverfahren werden dazu verwendet, sich schrittweise den Nullstellen von Funktionen zu nähern.

Die Zielgruppe die das System erreichen will, sind vor Allem Schüler, die dieses Programm für ihre Matheaufgaben benutzen können, aber auch Lehrer, wie unter Anderen auch unser Kunde *Frau Campidell*, die dieses Programm für ihre Lehrstunden gebrauchen können.

2. Produkteinsatz

Das Produkt soll in der Mathematik Einsatz finden. Es soll das Verstehen und Anwenden von verschiedenen Näherungsverfahren erleichtern.

2.1. Regula Falsi

Regula Falsi ist im Prinzip eine Mischung aus dem Bisektions- und Sekantenverfahren. Es verbindet sozusagen die Vorteile der beiden zuvor genannten Verfahren.

Das Verfahren startet mit zwei Stellen (nahe der Nullstelle), a_0 und b_0 , deren Funktionswerte **unterschiedliche** Vorzeichen haben. Das bedeutet also, dass sich in diesem Intervall $[a,b]$ eine Nullstelle befindet. Nun verkleinert man immer weiter das Intervall und nähert sich somit immer weiter der Nullstelle.

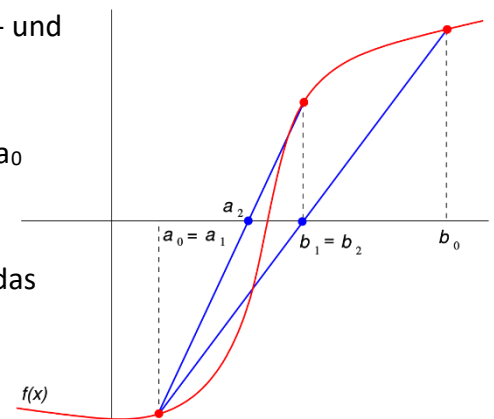


Abbildung 1: Regula-Falsi

2.2. Sekantenverfahren

Beim Sekantenverfahren handelt es sich im Prinzip um eine Vereinfachung des Newton-Verfahrens, weil man hier nicht die Ableitung der Funktion benötigt.

Beim Verfahren wird durch zwei Punkte der Funktion eine Sekante gelegt. Der Schnittpunkt mit der x-Achse (, der sich nun näher an der Nullstelle befindet), wird nun als x-Koordinate des neuen Punktes auf der Funktion verwendet und es wird eine Sekante durch die beiden Punkte gelegt. Somit nähert man sich immer weiter der Nullstelle.

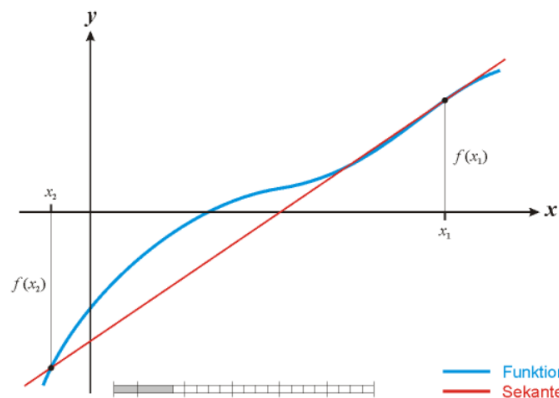


Abbildung 2: Sekantenverfahren step 1

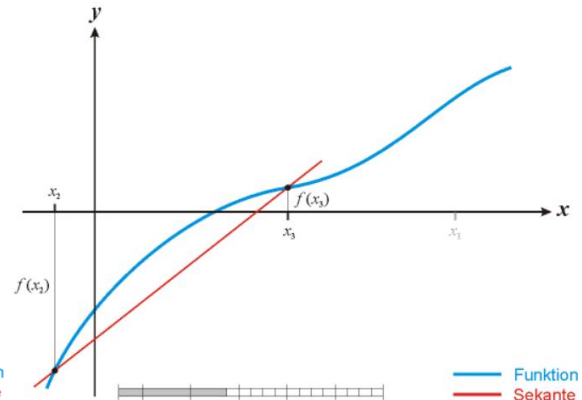


Abbildung 3: Sekantenverfahren step 2

2.3. Newton-Verfahren

Beim Newton-Verfahren (auch Tangenten- bzw. Tangentennäherungsverfahren genannt) wird die Sekante von Regula Falsi durch die Tangente am Punkt x_0 ersetzt. Die Nullstelle dieser Tangente wird dann als verbesserte Näherung der Nullstelle der Funktion verwendet. Dann wird wieder eine Tangente durch den Punkt gelegt und immer so weiter. Somit nähert man sich immer weiter der gesuchten Nullstelle.

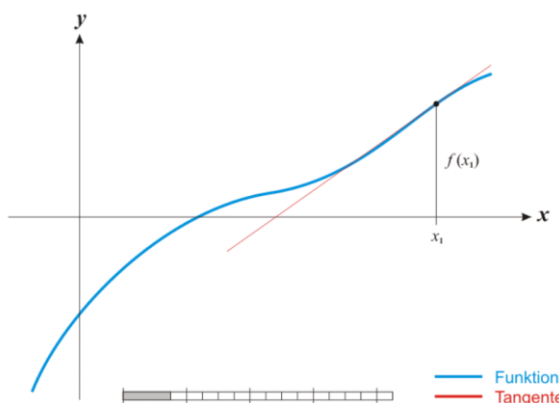


Abbildung 4: Newton-Verfahren step 1

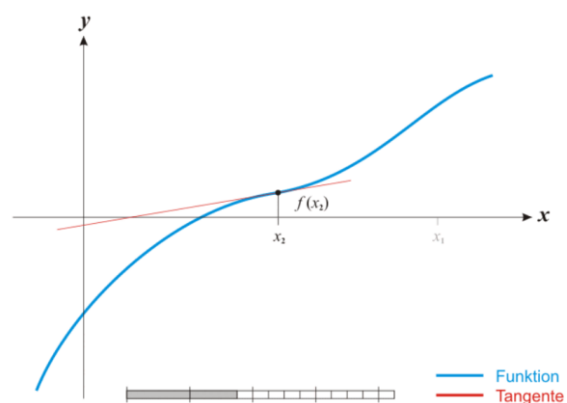


Abbildung 5: Newton-Verfahren step 2

2.4. Bisektion

Die Bisektion wird auch fortgesetzte Bisektion oder Intervallhalbierungsverfahren genannt.

Hierbei wird ein Intervall gewählt, in dem sich die Nullstelle befinden soll. Dieses Intervall wird dann halbiert und je nachdem ob sich die Nullstelle vor oder nach diesem erhaltenen Punkt befindet wird das neue Intervall gewählt und erneut halbiert. So nähert man sich schrittweise der Nullstelle.

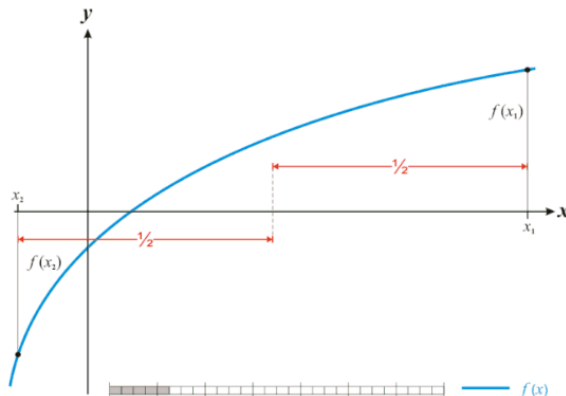


Abbildung 6: Bisektion step 1

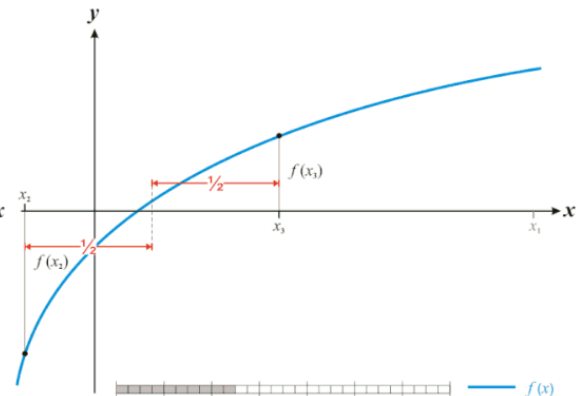


Abbildung 7: Bisektion step 2

2.5. Euler-Tschebyschow-Verfahren

Das Euler-Tschebyschow-Verfahren ist vergleichbar mit dem Newton-Verfahren. Im Gegensatz zum Newton-Verfahren benötigt man aber die 2. Ableitung der Funktion. Das Verfahren wird durch den mehrdimensionalen Fall im Halley-Verfahren hergeleitet.

2.6. Steffensen-Verfahren

Auch das Steffensen-Verfahren ist dem Newton-Verfahren ähnlich.

Will man die Nullstelle in einem gewissen Bereich finden so, wird in diesem Bereich nach der Nullstelle gesucht, indem eine x-Koordinate gesucht wird, bei der die Steigung zwischen -1 und 0 liegt. Dies muss aber nicht unbedingt der Fall sein. Es kann auch ohne dieser Bedingung zum gewünschten Ergebnis führen.

Das Verfahren gilt als nicht besonders effizient.

2.7. Beschreibung des Problembereichs

Das Programm soll dem Benutzer durch Eingabe der Mindestanforderungen Näherungswerte der Nullstellen der Funktion liefern, sowie die Funktion für ein besseres Nachvollziehen und Verständnis visualisieren.

3. Produktfunktionen

3.1. Use Case Diagramme

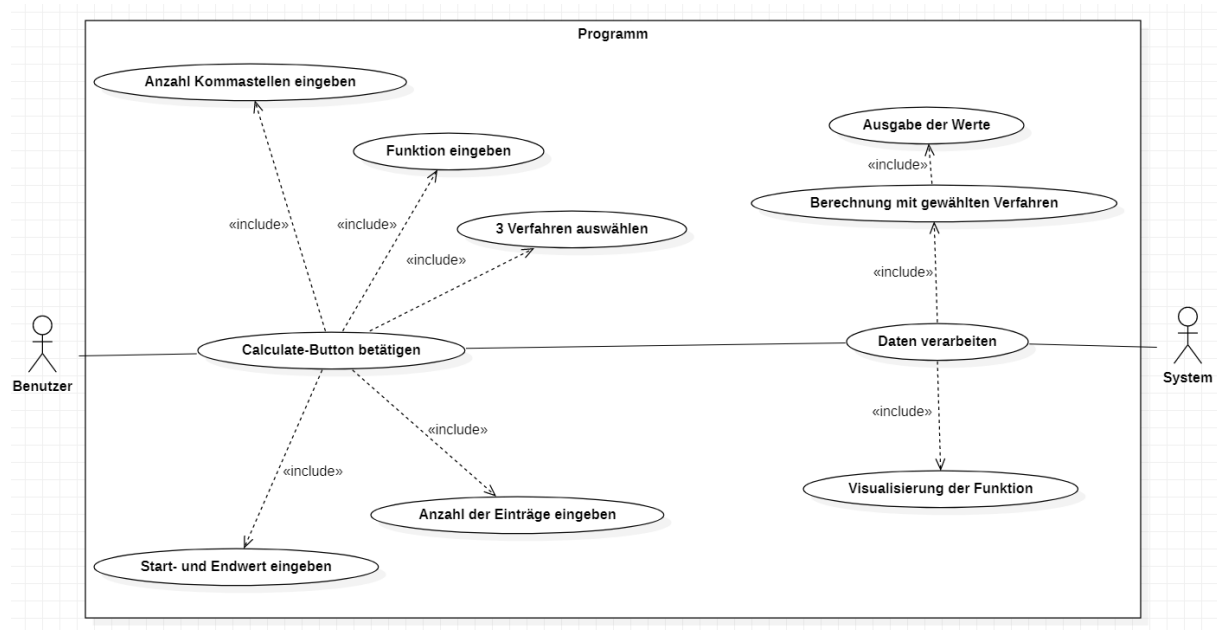


Abbildung 8: Use-Case-Diagramm

3.2. Möglichkeiten des Benutzers

3.2.1. Funktion eingeben

Der Benutzer soll in einem Textfeld die gewünschte Funktion, die dargestellt und mit der gerechnet werden soll, eingeben können (z.B. $\sin(3x)$).

3.2.2. Anzahl Kommastellen eingeben

Der Benutzer soll entscheiden können mit welcher Genauigkeit, d.h. mit wie vielen Nachkommastellen, die Werte vom System berechnet werden sollen.

Beispiel: Der Benutzer gibt 5 ein, also haben alle berechneten Werte 5 Nachkommastellen.

3.2.3. Start- und Endwert festlegen

Der Benutzer soll den Start- und Endwert des gewünschten Intervalls selbst wählen können. In dem vom Benutzer gewählten Intervall werden später Nullstellen gesucht.

Beispiel: Gibt der Benutzer 4 und 5 für den Start- und Endwert ein, so werden die Nullstellen in dem Intervall $[4, 5]$ gesucht.

3.2.4. Anzahl der Einträge festlegen

Der Benutzer kann auswählen, wie oft das Verfahren wiederholt werden soll, d.h. wie oft der Vorgang des Verfahrens ausgeführt werden soll. Je höher diese Zahl gewählt wird, desto genauer ist das Ergebnis, das man für die gesuchte Nullstelle erhält, da sich die Verfahren ja immer weiter an das gewünschte Ergebnis, also die tatsächliche Nullstelle, annähert.

3.2.5. 3 Verfahren auswählen

Der Benutzer kann aus den ausprogrammierten Verfahren 3 auswählen. Mit diesen Verfahren werden dann die Nullstellen berechnet.

3.3. Aufgaben des Systems

Das System muss zunächst alle vom Benutzer getätigten Einstellungen analysieren und anwenden. Die Werte, die der Benutzer eingegeben hat, können nun zur Berechnung verwendet werden.

3.3.1. Berechnung mit gewählten Verfahren

Mit den vom Benutzer gewählten Verfahren werden nun jeweils die Nullstellen berechnet.

Die Ergebnisse der Berechnung werden auf der GUI dargestellt.

Außerdem werden die Zwischenergebnisse zu jedem Verfahren auf der GUI dargestellt.

Bsp.: Das Newton-Verfahren braucht bei einer bestimmten Funktion 14 Zwischenschritte, bis das Ergebnis exakt ist. Zuvor hat der Benutzer eine *Anzahl der Einträge* von 20 gewählt. Somit werden alle Werte dieser 14 benötigten Zwischenschritte in einem Feld auf der GUI dargestellt.

Überschreitet beispielsweise ein Verfahren die maximale *Anzahl der Einträge*, so werden natürlich, wie in diesem Beispiel hier, nur die ersten 20 Zwischenergebnisse ausgegeben.

3.3.2. Visualisierung der Funktion

Die vom Benutzer eingegebene Funktion soll in einem eigenen Bereich der GUI visualisiert werden, damit der Benutzer erkennen kann, dass die errechneten Nullstellen mit den auf der Grafik dargestellten Nullstellen übereinstimmen. Die Achsen des Koordinatensystems sollen variabel bewegbar sein, damit der Benutzer die dargestellte Funktion besser betrachten kann.

4. Reengineering (Ist-Zustand)

Das Verhalten zwischen Benutzer und System soll wie folgt geklärt werden:

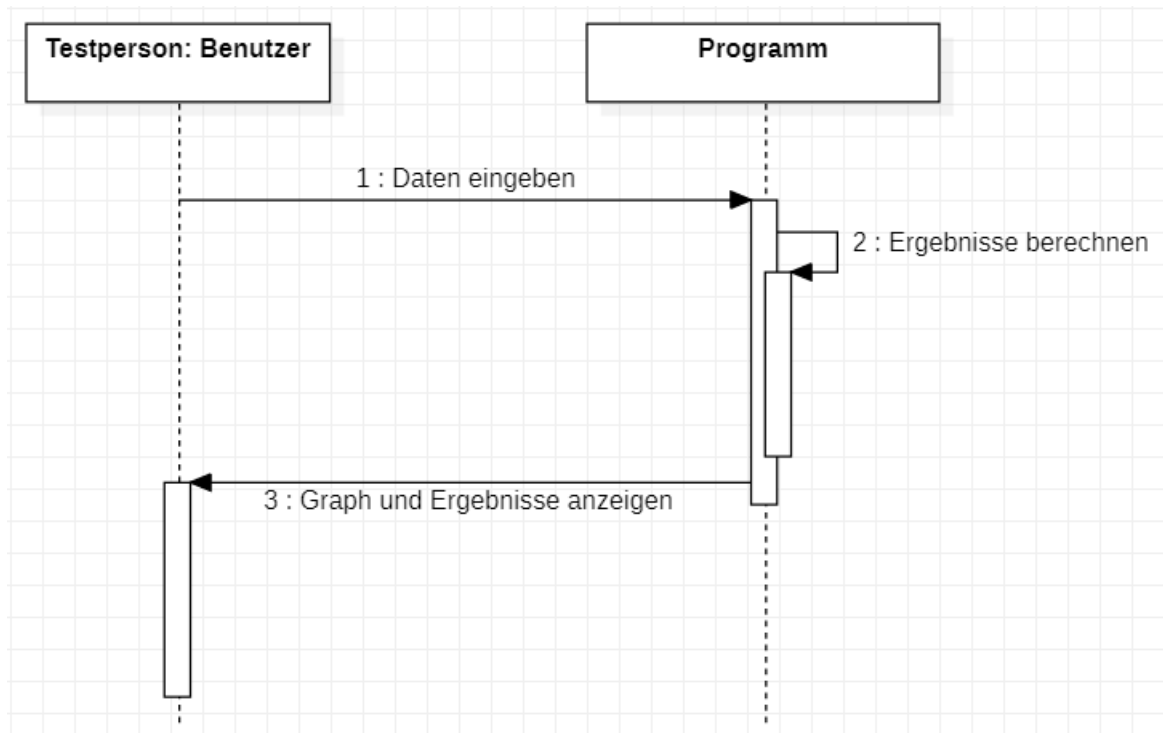


Abbildung 9: Sequenzdiagramm

Detailliert kann man hier die Aufgaben bzw. Möglichkeiten des Benutzers und des Systems sehen:

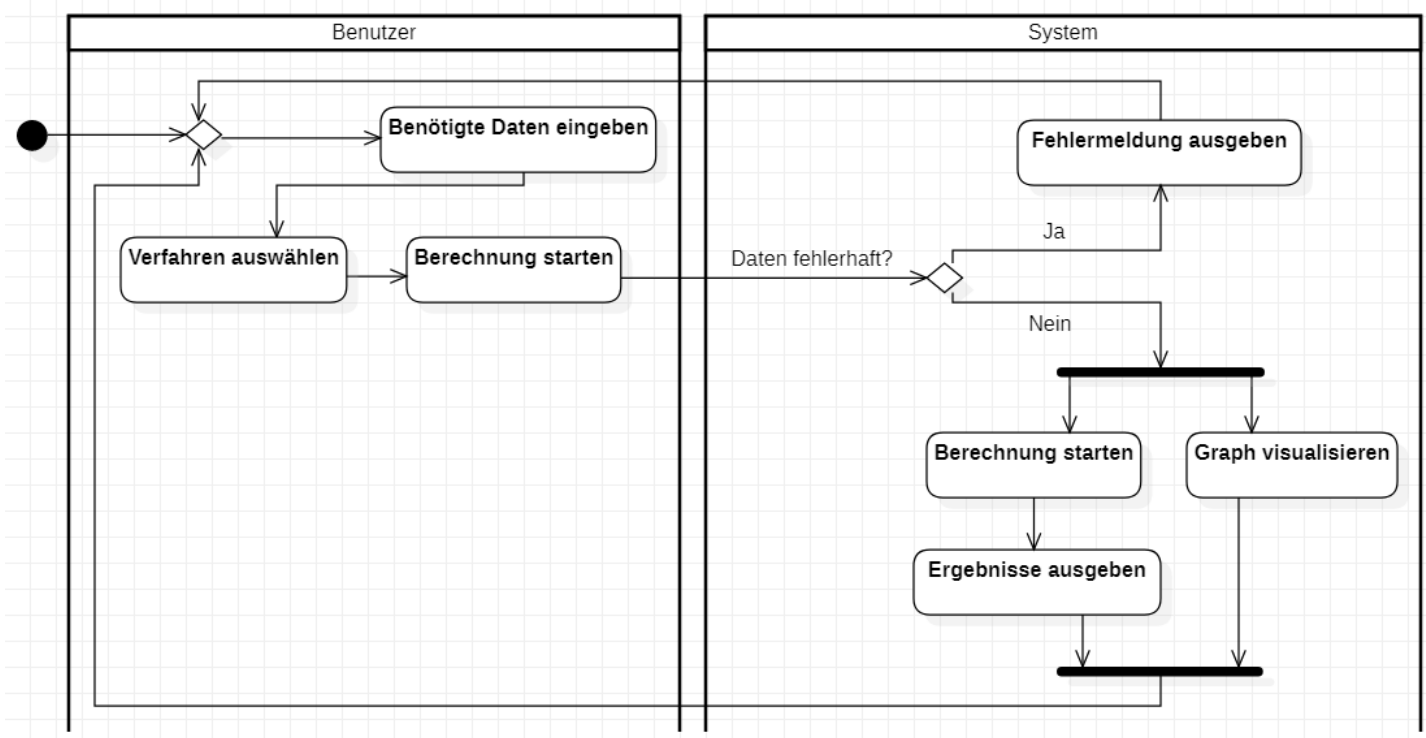


Abbildung 10: Aktivitätsdiagramm

5. Grobentwurf (Soll-Zustand)

Der Austausch zwischen den einzelnen Klassen des Systems wird wie folgt geregelt:

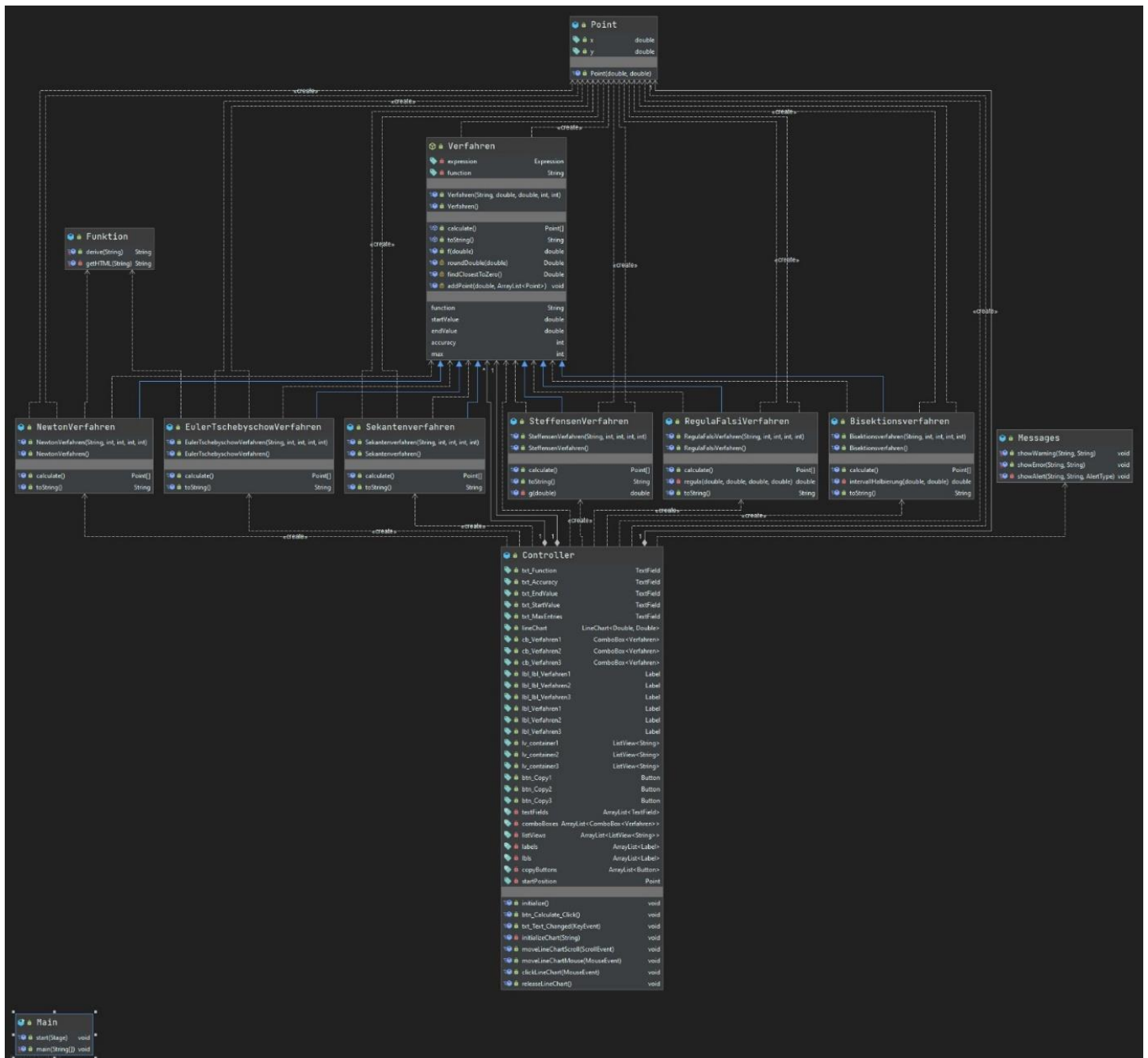


Abbildung 11: Klassendiagramm

6. Qualitätsmerkmale

6.1. Typ USE: Benutzbarkeitsanforderung

Für die Zielgruppe des Systems soll die GUI des Systems möglichst benutzerfreundlich, also einfach, übersichtlich und ansprechend gestaltet sein.

6.2. Typ EFFIZIENZ: Effizienzanforderung

Die Berechnung soll für den Benutzer sehr schnell ablaufen. Wartezeiten sollen vermieden werden, damit der Fluss des Programms nicht unterbrochen wird.

6.3. Typ PFLEGE: Wartbarkeits- und Portierbarkeitsanforderung

Damit zukünftig weitere Verfahren zum System hinzugefügt werden können, werden verschiedene Maßnahmen getroffen.

Zunächst wird eine abstrakte Klasse Verfahren verwendet, da alle Verfahren im Prinzip die gleichen Hauptfunktionen und Hauptvariablen benötigen. So wird das Hinzufügen von weiteren Verfahren in der Zukunft erleichtert.

6.4. Typ SICHER: Sicherheitsanforderung

Da der Benutzer sich bei diesem System nicht anmelden muss, werden keine persönlichen Daten des Benutzers an das System weitergegeben.

6.5. Typ LEGAL: Gesetzliche Anforderung

Es müssen keinerlei spezifische gesetzliche Anforderungen beachtet werden.

7. Umsetzungsdetails

Für das Programmieren des Systems wird Java (1.8) verwendet. Zum Erstellen des GUI wird JavaFX verwendet.

Es werden 2 zusätzliche Bibliotheken verwendet: exp4j (Für die leichtere Interpretation von den Funktionen, die vom Benutzer eingegeben werden) und JUnit (Für Tests).

8. Projektplan

Das Projekt soll voraussichtlich bis 27.04.2020 abgeschlossen werden.

Es sind keinerlei Kosten eingeplant.