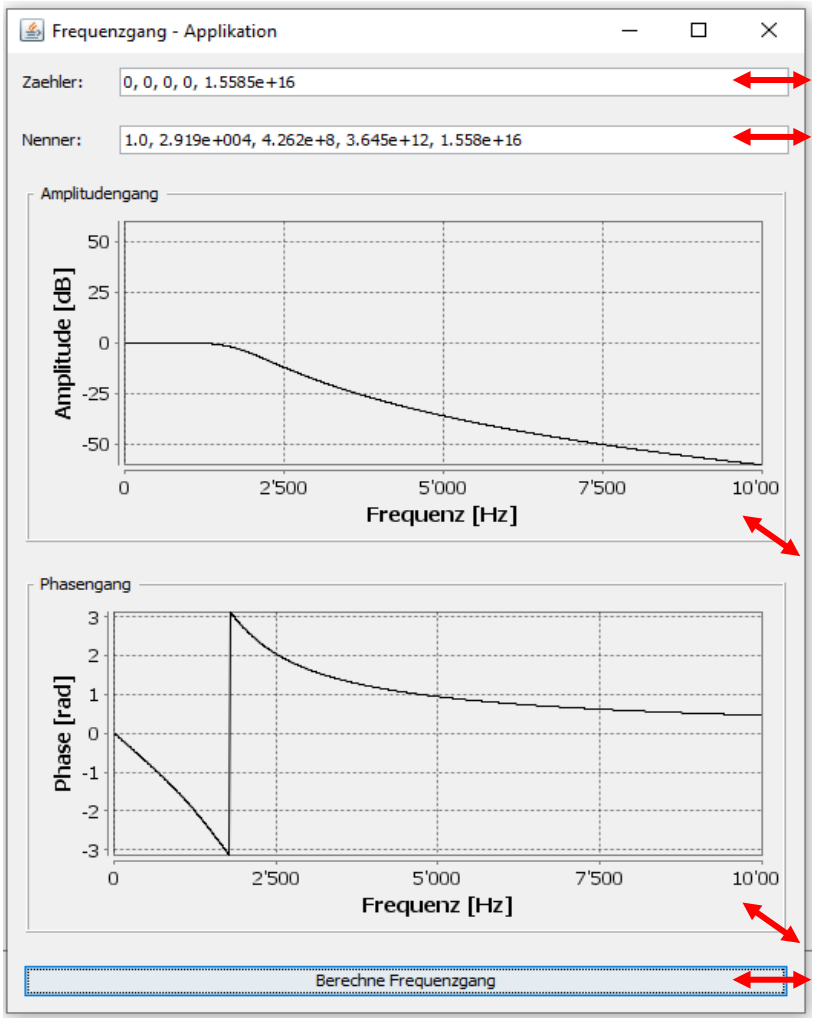
Übung: Frequenzgang (Umgang mit komplexen Zahlen und Polynomen)

In dieser Übung wollen wir anhand der Übertragungsfunktion den Frequenzgang eines Systems berechnen und darstellen. Hierzu müssen komplexwertige Polynome berechnet werden. Der Frequenzgang ist angelehnt an Matlab gegeben als

$$H(\omega) = \frac{b_0 \cdot (j \cdot \omega)^N + b_1 \cdot (j \cdot \omega)^{N-1} + b_2 \cdot (j \cdot \omega)^{N-2} + \ldots + b_N}{a_0 \cdot (j \cdot \omega)^M + a_1 \cdot (j \cdot \omega)^{M-1} + a_2 \cdot (j \cdot \omega)^{M-2} + \ldots + a_M}.$$

Die Applikation soll folgende Ein- und Ausgabe Optionen haben:



Die Koeffizienten des Zähler- und Nennerpolynoms werden mittels der beiden Textfelder eingegeben. Als Trennzeichen soll ein Komma oder Leerzeichen verwendet werden. Die Klassen zur Darstellung von Amplituden- und Phasengang sind bereits gegeben. Hierzu wird das Package **jfreechart** verwendet (http://www.jfree.org/jfreechart/). Die Applikation basiert auf dem Model-View-Controller Pattern.

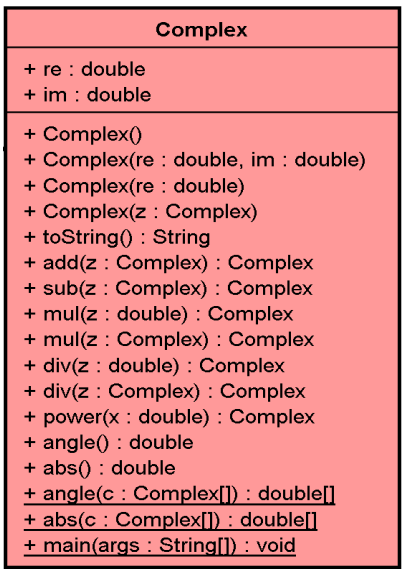
Aufgabe 1:

Machen Sie sich mit anhand des Klassendiagramms mit der Struktur des Programms vertraut. Beachten Sie insbesondere das Design gemäss MVC-Entwurfspattern.

Aufgabe 2: Klasse Complex

Wie die meisten Programmiersprachen kann Java nicht per se mit komplexen Zahlen rechnen. Der Umgang mit komplexen Zahlen wird durch die Klasse Complex realisiert, wo die Methoden gemäss Klassendiagramm zu implementieren sind.

a) Schreiben Sie die Klasse Complex, welche die benötigten komplexen Rechenoperation und die zu einer komplexen Zahl gehörenden Real- und Imaginärteil definiert (kapselt).
b) Überlegen Sie, wie die Klasse vorab getestet werden könnte



Aufgabe 3: Erstellen der Klasse View

Das User-Interface ist als **GridBagLayout** organisiert. Das Verhalten bei Vergrösserung ist im Bild definiert. Die Klassen **AmplitudenPlot** und **PhasenPlot** sind bereits vorgegeben und müssen nur noch platziert werden.

a) Erstellen Sie das User-Interface entsprechend der Angaben und der Abbildung.
b) Implementieren Sie die restlichen Methoden der Klasse **View** gemäss MVC-Konzept.

Aufgabe 4: Erstellen der Klasse Controller

Der Controller agiert als Schaltzentrale bezüglich User-Interaktion. Mit dazu gehört die Aufgabe, aus den Zeichenketten in den Textfeldern double - Arrays zu erzeugen

a) Implementieren Sie die Methoden der Klasse **Controller.** Der Controller soll bei Knopfdruck die Übertragungsfunktion des Models setzen.
b) Überlegen sie Sich, wie die Methode **stringToCoeff()** getestet werden kann.

Aufgabe 5: Erstellen der Klasse PicoMatlab

Die Klasse PicoMatlab beinhaltet die statische Methode **freqs(double[] b, double[] a, double[] f)**, welche aus den Zähler- und Nennerpolynomen, den komplexwertigen Frequenzgang berechnet. Hierzu wird der entsprechende Polynomwert mit der Methode **polyval(double[] p, Complex jw)** berechnet. Die Polynome sind als Koeffizienten in den double arrays hinterlegt.

a) Implementieren Sie die Methoden der Klasse PicoMatlab
b) Testen Sie die Methoden der Klasse

Aufgabe 6: Erstellen der Klasse Model:

Die Klasse Model beheimatet die Daten des Frequenzganges, nämlich die Frequenzachse und den komplexwertigen Frequenzgang. Mit den Methoden **getFaxis**(), **getAmplitude**() und **getPhase**() kann die Information ausgelesen werden. Mit **setUTF**() wird eine neue Übertragungsfunktion festgelegt.

Implementieren Sie die Methoden der Klasse Model.

Klassendiagramm: