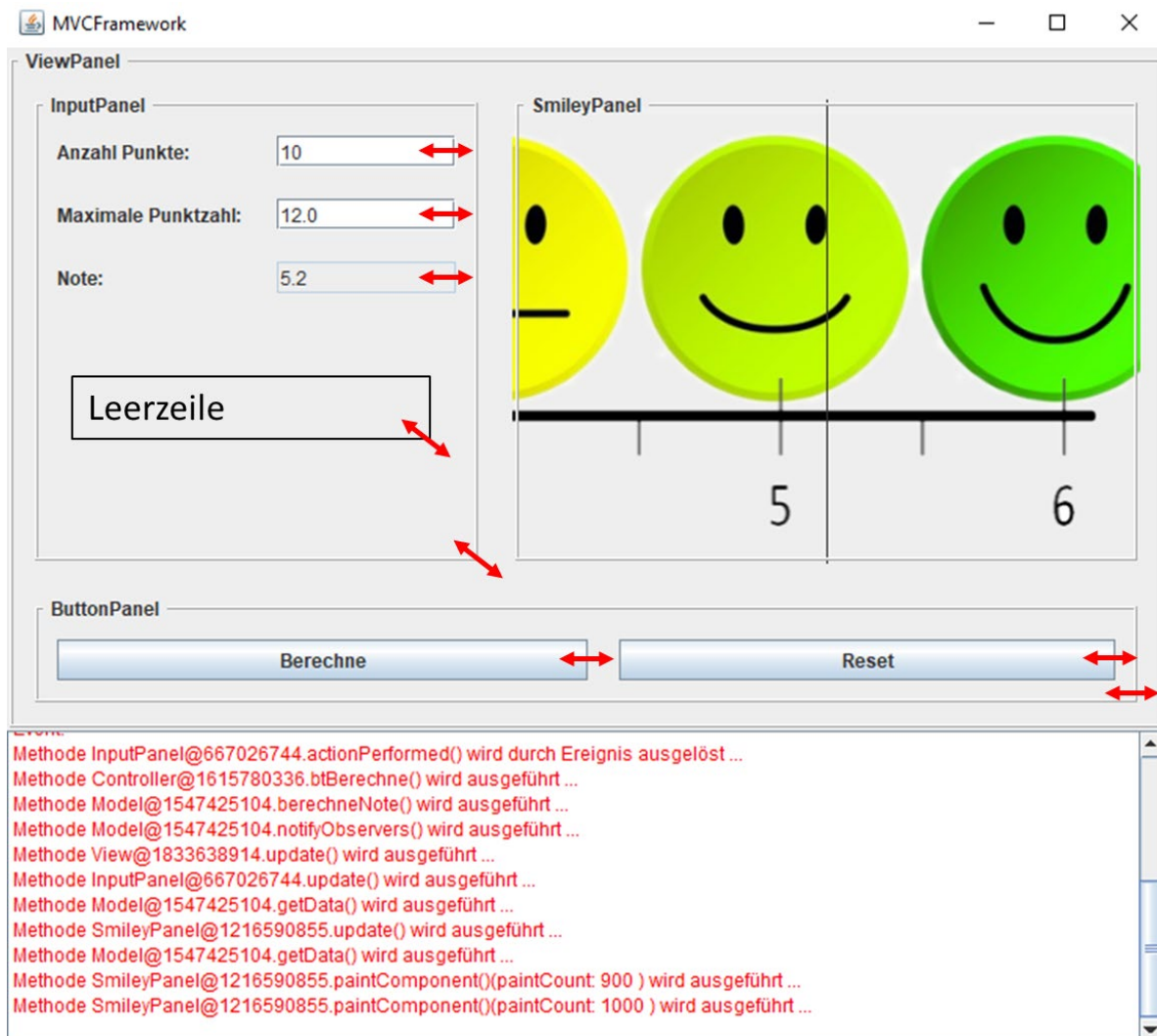


## Übung: Notenrechner Applikation

Ziel dieser Übung ist es, eine GUI Applikation zu programmieren, die aufgrund der Maximalen Punktzahl und der erreichten Punktzahl die Note errechnet



Der Rechner ist im **Model-View-Controller** Entwurfsmuster gehalten. Der View kommt die Möglichkeit zur Eingabe der Daten und die Repräsentation des Resultates zu. Sie ist zu diesem Zwecke in drei Panels gegliedert: Das **InputPanel** beheimatet die benötigten Eingabetextfelder, das Textfeld zur Ausgabe der Note sowie drei Label. Das **SmileyPanel** zeigt das zur Note passende Gesicht mittels animierter Smileys und ist im Null-Layout gehalten. Das **ButtonPanel** verfügt über die Buttons **Berechne** und **Reset** zum Berechnen der Note und Zurücksetzen auf den. Dem **Controller** obliegt die Steuerung der Applikation. Er liest auf das entsprechende Ereignis hin die Werte aus den Textfeldern aus und delegiert die Berechnung an das Model. Das Model löst nach der Berechnung mittels Observer-Observable Entwurfsmuster ein Update der repräsentierten Daten aus.

Das **ViewPanel** (CENTER) ist bereits mit einer JScrollPane (SOUTH) für Konsolenausgaben im BorderLayout platziert (siehe Skizze).

### Aufgabe1 :

Überlegen Sie sich die Aufgaben der einzelnen Klassen und Methoden anhand des Klassen-  
diagramms und verifizieren Sie das MVC Entwurfsmuster.

### Aufgabe2:

Erstellen Sie anhand der Vorlage und der obigen Skizze das GUIs mittels GridBagLayout.

- Beginnen Sie mit der Klasse View und der Platzierung der drei Panels im Konstruktor der Klasse.
- Erzeugen Sie nun die Elemente auf dem InputPanel und ButtonPanel in den Konstruktoren der Klassen.

### Aufgabe 3:

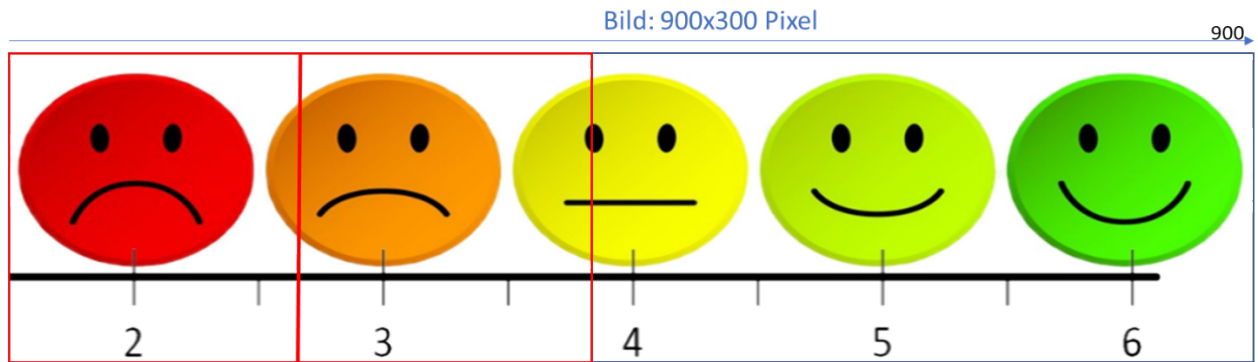
Implementieren Sie die Methoden der einzelnen Klassen.

- Beginnen Sie mit der Implementierung des **Controllers** und sorgen Sie für die Weiterleitung der Daten an das Model. Die Events aus den Panels müssen an geeigneter Stelle an den Controller übermittelt werden.
- Danach folgt die Berechnung der Note und Darstellung im **InputPanel**. Implementieren Sie die Methoden des Models und sorgen Sie für ein Update der aktualisierten Daten. Die Note soll nach folgender Kurve berechnet werden. Sorgen Sie in der Darstellung des Ergebnisses dafür, dass die Note auf eine Nachkommastellen gerundet wird.



- Anhand der Note soll beim Update das Smiley Bild so platziert werden, dass die erzielte Note mit dem mittleren Bildstrich übereinstimmt. Hierzu wird das Bild nach einer zu definierenden Rechenvorschrift nach links (negativer x-Wert) oder rechts verschoben. Für  $X = 0$  ergibt sich das untere Bild und die Bildmitte des Panels erscheint bei der Note 2.6. Ermitteln Sie davon ausgehend die Verschiebevorschrift. und implementieren Sie die

Methoden der Klasse SmileyPanel. Der **SimpleTimer** im SmileyPanel soll nun noch dafür sorgen, dass bei einer neuen Berechnung das Bild animiert den neuen Wert anfährt. Das lässt sich mit einer Gewichtung von alten und neuen x-Werten realisieren (Filterung).



-

# Anhang

Klassendiagramm:

