

Zu erstellen ist eine grafische Anwendung zur Verwaltung der Wahlversprechen einer recht unzuverlässigen Partei.

## 1. Teil) - Klassen für die Daten (~25%)

<i>Wahlversprechen</i>	<i>Parteiprogramm</i>
<ul style="list-style-type: none"> <li>- schlagwort: <i>String</i></li> <li>- beschreibung: <i>String</i></li> <li>- stimmenzuwachs: <i>float</i></li> <li>- wirdVerworfen: <i>String</i></li> <li>- gueltig: <i>boolean</i></li> <li>- ausrede: <i>String</i></li> </ul>	<ul style="list-style-type: none"> <li>- parteiname : <i>String</i></li> <li>- prozentStammwaehler : <i>float</i></li> <li>-wunschliste : <i>Liste (typisiert auf Wahlversprechen)</i></li> </ul>
<ul style="list-style-type: none"> <li>+ Wahlversprechen(...) <b>throws</b> ...</li> <li>+ get- &amp; set-Methoden</li> <li>+ toString() : <i>String</i></li> </ul> <p>// ev. noch zu erstellende Methoden: + ...</p>	<ul style="list-style-type: none"> <li>+ einfache get- &amp; set-Methoden</li> </ul> <p>// ev. noch zu erstellende Konstlr./Methoden:  + Parteiprogramm(...)  + hinzufuegen(...) <b>throws</b> ...  + loeschen(...) <b>throws</b> ...  + getWahlversprechen(int index):<i>Wahlversprechen</i>  + getAnzahlWahlversprechen() : <i>int</i>  + berechneSummeStimmenzuwachs() : <i>float</i>  + sortierteListeAlsText(...) : <i>String</i>  + speichern(dateiname : ...) <b>throws</b> ...  + laden(dateiname : ...) <b>throws</b> ...  + ...</p>

Das UML-Diagramm dient als Orientierung im Model. Die *Wahlversprechen*-Klasse ist bezüglich Überprüfungen & Exceptions komplett. Die *Parteiprogramm*-Klasse dient zur Verwaltung der Wahlversprechen und es sind nur die Attribute und einfache set- und get-Methoden vorgegeben. Die wesentlichen Methoden fehlen und sind zu implementieren.

## 2. Teil) - Grafische Anwendung (in Summe ~75%)

- **MVC** einhalten (~5%)

Der Aufbau der Anwendung soll gemäß dem **Model-View-Controller** Prinzip erfolgen!

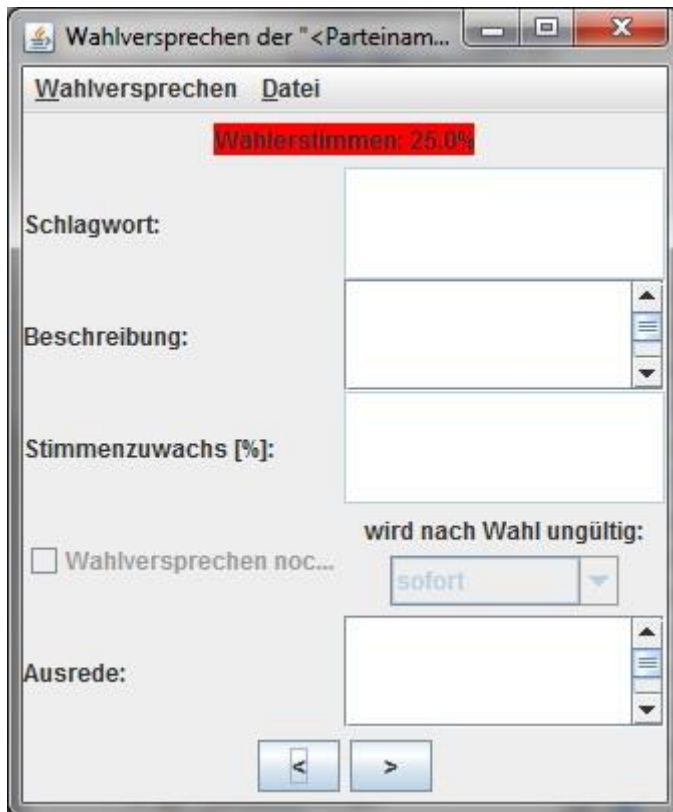
- und Weiterleitung von **Fehlermeldungen** (~5%)

Sämtliche Fehler werden mittels *Exceptions* oder Rückgabewerte aufgezeigt und **an die View- oder Controller Klasse weitergereicht**, welche sie z.B. in Form einer *JOptionPane* ausgibt.



- **Hauptfenster:** (~10%)

Das Hauptfenster hat eine **Menüleiste** für die wesentlichen Funktionen, eine Hauptpanel zur **Anzeige eines Wahlversprechens**, Tasten zum **Vor-** und **Zurückblättern** und eine Anzeige der **aktuellen Wählerstimmen in %** (= Stammwähler + %-Werte der Wahlversprechen).



Die Prozentanzeige ist unter 50% rot, darüber grün.

Das Hauptfenster ist nicht eingabebereit und bitte hier keinen großen Aufwand bezüglich Layout treiben, die Funktion ist wichtig!

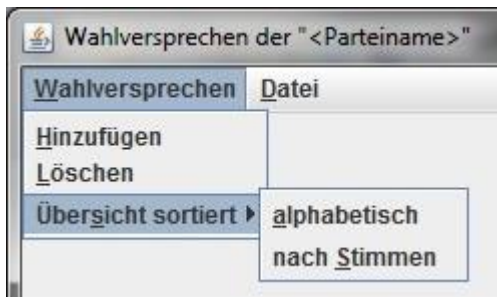
Der mittlere Bereich (*GridLayout*) mit den Eigenschaften eines Wahlversprechens basiert auf einer eigenen Panel-Klasse, die mehrfach verwendet wird.

Die Combobox, erlaubt auszuwählen, in welchem Zeitraum nach der Wahl, die Wahlversprechen zurückgenommen werden. Die vier möglichen Werte sind als Konstante in der Klasse *Wahlversprechen* definiert.

Mit den Vor- und Zurück-Tasten kann geblättert werden, sofern Daten vorhanden sind.

- **Menüleiste:** (~10%)

Das Hauptfenster erhält eine Menüleiste mit folgenden Möglichkeiten:



- Menüpunkte **Speichern / Laden:** (~10%)

Es öffnet sich ein entsprechender *JFileChooser*-Dialog, der Anwender wählt die betreffende Datei und die Liste der Wahlversprechen wird gespeichert bzw. geladen.

Tipp: Der Quelltext zur Verwendung des *JFileChooser*-Dialogs ist in der Java-API-Dokumentation zu finden. Es gibt unterschiedliche *show*-Methoden für die Dateiauswahl zum Laden oder Schreiben. Den *FileNameExtensionFilter* einfach weglassen!  
Zum Ausprobieren kann zunächst ein fix gewählter Dateiname verwendet werden.

- Menüpunkt **Hinzufügen**: (~15%)

Es öffnet sich ein Eingabefenster, in dem die Daten eines neuen Wahlversprechens eingegeben werden können. Nur wenn ein noch nicht verwendetes Schlagwort eingegeben wurde und alle übrigen Bedingungen erfüllt sind wird ein entsprechendes Objekt erzeugt und übernommen. Abschließend wird die Anzeige des Hauptfensters aktualisiert.

Tipp: hier wurde die Panel-Klasse des Hauptfensters wiederverwendet, dieses mal allerdings eingabebereit. Zusätzlich gibt es zwei Tasten zum Übernehmen oder Abbrechen der Eingabe.

- Menüpunkt **Löschen**: (~5%)

Das aktuell im Hauptfenster angezeigte Wahlversprechen wird gelöscht. Aktualisieren der Anzeige nicht vergessen!

- Menüpunkt **Übersicht sortiert...**: (~15%)

Es wird ein einfaches Textfenster geöffnet, indem eine sortierte Liste aller Wahlversprechen entsprechend dem gewählten Menüeintrag angezeigt wird.

### Vereinfachung:

Das Fenster **Hinzufügen** darf einen eigenen *ActionListener* verwenden bzw. implementieren und muss nicht die „zentrale“ Controller-Klasse zur Prüfung der Eingabe... nutzen.

**Gutes Gelingen!**