Projekt Teilnehmerverwaltung

Ziel:

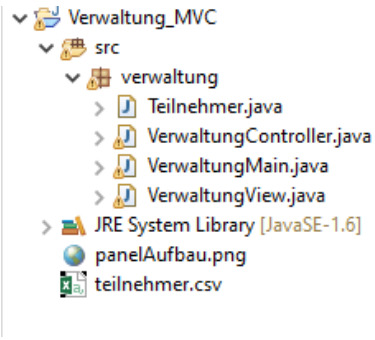
Es soll ein Java-Programm geschrieben werde, dass eine bestehende CSV-Datei mit Teilnehmerdaten einliest und diese zur weiteren Bearbeitung (löschen, ändern) in einer grafischen Oberfläche anzeigt.

Zusätzlich soll es möglich sein neue Teilnehmer anzulegen.

Die veränderten/ neu generierten Daten sollen in der ursprünglichen Datei gespeichert werden können.

Das Programm soll nach dem Model-View-Controller-Prinzip aufgebaut sein.

Ordnerstruktur:



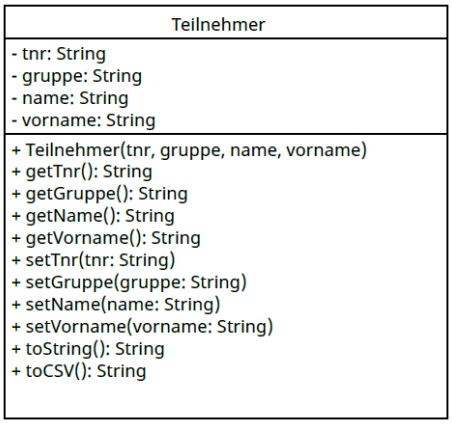
Dateistruktur:

	A	B	C	D
1	Nr.	Gruppe	Name	Vorname
2	1	Comic	Maus	Minni
3	2	Comic	Duck	Donald
4	3	Zauberer	Gans	Gustav
5	4	Zauberer	Dumbledore	Albus
6	5	Zauberer	Potter	Harry
7	6	Comic	Maus	Micky
8	8	Personen	Christel	Kind

```
Nr.;Gruppe;Name;Vorname;
1;Comic;Maus;Minni
2;Comic;Duck;Donald
3;Zauberer;Gans;Gustav
4;Zauberer;Dumbledore;Albus
5;Zauberer;Potter;Harry
6;Comic;Maus;Micky
8;Personen;Christel;Kind
```

Model:

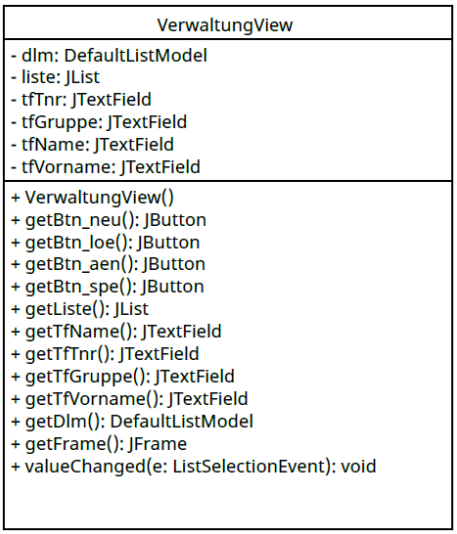
Die **Teilnehmer**-Klasse repräsentiert Teilnehmerdaten und dient als **Model**. Sie beinhaltet die Attribute tnr, gruppe, name, vorname. Es werden **Getter** und **Setter** angelegt, um den Zugriff auf die Attribute zu ermöglichen.



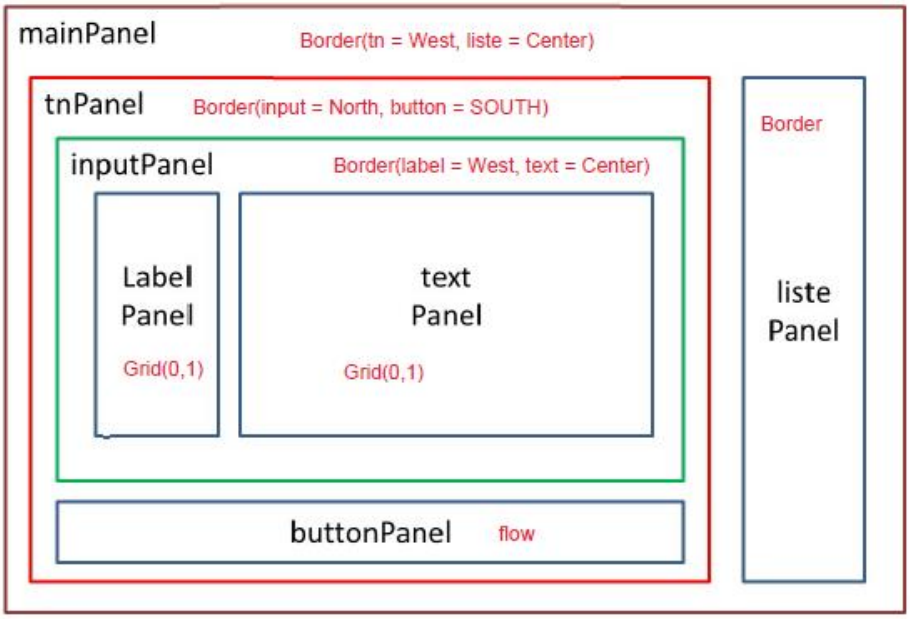
View (GUI):

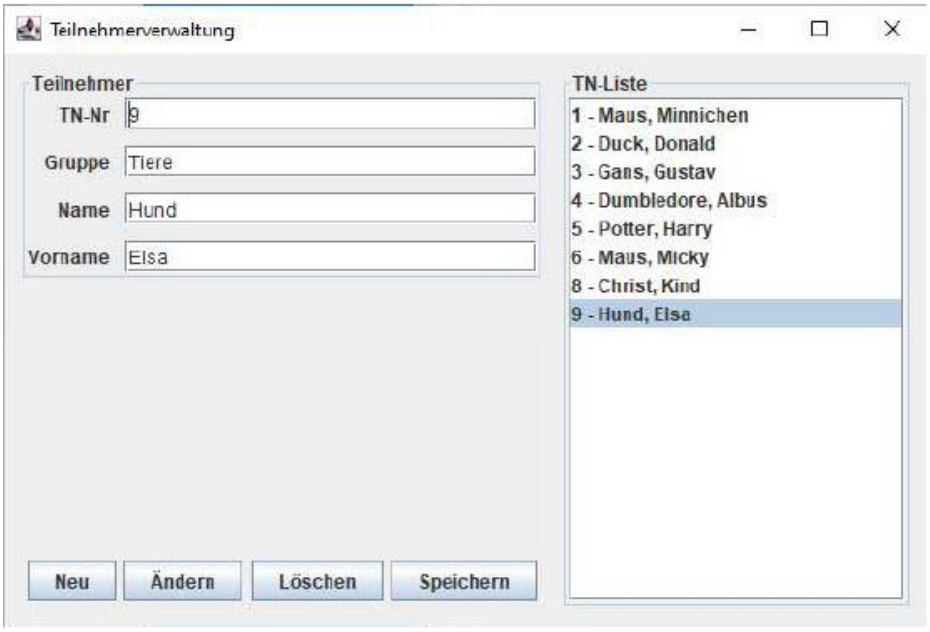
Die **VerwaltungView**-Klasse erstellt die Benutzeroberfläche. Sie enthält eine Liste zur Anzeige von Teilnehmern, Textfelder für Eingaben und Buttons zur Interaktion.

Dies ist der Teil deiner Anwendung, der die **Benutzeroberfläche** darstellt und die visuellen Elemente enthält. Die Benutzeroberfläche wird mit Swing-Elementen erstellt und die **Listener** (ActionListener und ListSelectionListener) hinzugefügt, um auf Benutzerinteraktionen zu reagieren. Die View ist dafür verantwortlich, die Benutzeroberfläche anzuzeigen und die Ereignisse an den Controller weiterzugeben.

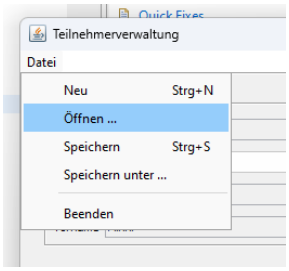


GUI-Entwurf:





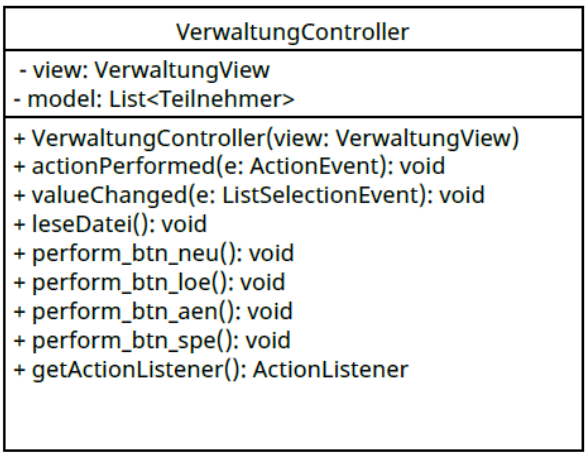
Menustruktur:



Aufbau Controller:

Die **VerwaltungController**-Klasse enthält die Geschäftslogik der Anwendung und reagiert auf Benutzeraktionen. Sie verwaltet eine Liste von Teilnehmern und ermöglicht das Hinzufügen, Ändern und Löschen von Teilnehmern.

Dies ist der Teil der Anwendung, der die Geschäftslogik enthält und die Verarbeitung von Benutzeraktionen steuert. Es werden separate Methoden im Controller erstellt, um auf die Aktionen der Buttons und die Auswahl in der Liste zu reagieren. Der Controller ist dafür verantwortlich, die Geschäftslogik zu implementieren, Daten zu speichern oder zu ändern und die View zu aktualisieren.



Die Kommunikation zwischen View und Controller erfolgt über **Listener**, die auf Benutzerinteraktionen reagieren. Die View hat Verweise auf die Buttons und die Liste, und sie registriert die Listener, um Benutzeraktionen abzufangen. Wenn eine Aktion ausgelöst wird, ruft die View die entsprechende Methode im Controller auf, um die Verarbeitung durchzuführen. Der Controller führt die erforderlichen Aktionen aus und aktualisiert gegebenenfalls die View, um Änderungen anzuzeigen.

Die **VerwaltungMain**-Klasse dient als Einstiegspunkt für deine Anwendung. In der main-Methode wird zunächst eine Instanz der **VerwaltungView** und des **VerwaltungController** erstellt, um deine Anwendung zu initialisieren. Dies ist ein typisches Muster für die MVC-Architektur. Hier wird die Benutzeroberfläche (View) und die zugehörige Logik (Controller) erstellt und miteinander verknüpft.

- Erstellen der **VerwaltungView**: Hier wird ein Objekt der VerwaltungView-Klasse erstellt, die die Benutzeroberfläche deiner Anwendung darstellt.
- Erstellen des **VerwaltungController**: Dann wird ein Objekt der VerwaltungController-Klasse erstellt, die die Logik deiner Anwendung enthält. Dieser Controller wird mit der VerwaltungView verknüpft, um die Interaktion zwischen der Benutzeroberfläche und der Logik zu ermöglichen.

Indem diese beiden Instanzen in der **main-Methode** erstellt und der Controller mit der View verbunden wird, wird die **Anwendung initialisiert**. Wenn das Programm ausgeführt wird, wird die Benutzeroberfläche angezeigt, und der Controller ist bereit, auf Benutzeraktionen zu reagieren.