

TERMINATOR

INHALTSVERZEICHNIS

Teilauftrag 1 - Projektbeschreibung	2
Graphische Elemente	2
Teilauftrag 2 – Mockups	3
Startseite	3
Neuer Termin	4
Zufügen eines Faches	5
Alle Termine	6
Teilauftrag 3 – Zustandsdiagramm	7
Teilauftrag 4 – Klassendiagramm	8
Teilauftrag 5 – Zustandsdiagramm der Schnittstellen	9
Begründung der Wahl	9
Diagramm	10
Endprodukt - Bericht	11

TEILAUFGTRAG 1 - PROJEKTBESCHREIBUNG

Seit nun fast zwei Wochen herrscht der Ausnahmezustand in der Schweiz, weshalb vieles geschlossen wurde. Auch die Schulen, weshalb wir Schüler nun Aufträge über «Distance-Learning» erledigen müssen. Da diese Situation für alle neu ist, kommt es vor allem bei den Schülern schnell mal zur Unübersichtlichkeit. Dieses Problem soll nun mein kleines Programm lösen und den Schülern ein wenig Last von den Schultern nehmen.

Mein Programm dient zur Erfassung aller erhaltenen Aufträge und Tests. Es ist im Grunde genommen eine elektronische Agenda. Der Benutzer (hier der Schüler) gibt die Bezeichnung der Aufgabe ein und kann mittels einer Checkbox festsetzen, ob es sich um einen Test oder eine Abgabe handelt. Des Weiteren muss er der Aufgabe ein Fach zuweisen, welches er durch ein Dropdown-Menü auswählen kann. Ist das Fach nicht vorhanden, so kann er ein neues Fach dem Dropdown hinzufügen, ohne dabei die bereits eingegebenen Daten zu verlieren. Hat der Benutzer die Aufgabe einem Fach zugeordnet, muss er nun das Datum eingeben, auf wann die Aufgabe zu erledigen ist. Zusätzlich kann er mittels einer Text-Area Notizen. Der Benutzer kann jederzeit die Eingabe abbrechen, wodurch der Eintrag gelöscht wird. Nachdem er den Eintrag gespeichert (oder halt eben abgebrochen) hat gelangt er zur Startmaske, wo ihm alle Aufgaben und Termine, die auf heute sind, in Form einer Liste angezeigt werden. Alle Tests werden in roter Farbe angezeigt.

Ausserdem hat der Benutzer die Möglichkeit, **alle** Aufgaben angezeigt zu bekommen. Diese werden, sortiert nach Datum, ebenso in einer Liste angezeigt.

Falls mir noch Zeit bleibt, wären noch folgende Funktionen möglich: Bei der Ansicht aller Aufgaben kann der Benutzer diese nach Fach oder Tests filtern. Ebenso kann er erledigte Aufträge mithilfe einer Checkbox abhaken. Somit weiss er, was er bereits erledigt hat. Nebst der Filter-Funktion kann der Benutzer durch eine Checkbox bestimmen, ob abgehakte Aufträge in der Liste angezeigt werden sollen oder nicht.

GRAPHISCHE ELEMENTE

- | | |
|-----------------|--------------------------|
| • Schaltflächen | JButton |
| • Checkboxen | JCheckBox |
| • Tabellen | JTable |
| • Combobox | JCombobox |
| • Texteingaben | JTextField |
| • Scrollbar | JScrollPane / JScrollBar |

TEILAUFGTRAG 2 – MOCKUPS

Das Programm enthält eine gewisse Anzahl an Daten. Daten werden als Textdaten erfasst und stellt eine Ansicht, ebenso in Textform, dar. Dieses Verhalten lässt sich am ehesten mit der **Formular-Metapher** beschreiben.

Das ganze Programm setzt auf den Grundsatz **Selbstbeschreibungsfähigkeit**, denn alle Elemente beschreiben sich selbst in einer kurzen und doch für den Benutzer verständlichen Bezeichnung.

STARTSEITE

Fach	Aufgabe	Erledigt
Englisch	Buch S34	<input checked="" type="checkbox"/>
English	Voci	<input type="checkbox"/>
Franz	Pronomen üben	<input type="checkbox"/>
Deutsch	Grammatik Test	<input checked="" type="checkbox"/>

☒ Erledigte anzeigen

Beim Starten des Programmes gelangt der Benutzer zur oben abgebildeten Startseite. Hier werden ihm alle Aufgaben und Tests in Form einer Liste angezeigt. Hat er diese erledigt, kann er diese abhaken. Zusätzlich kann er mithilfe der Checkbox abgehakte Aufgaben anzeigen oder verborgen lassen. Dies entspricht dem Grundsatz der **Individualisierbarkeit** am ehesten. Tests werden rot angezeigt, damit der Benutzer diese gleich erkennen kann, was dem Grundsatz **Aufgabenangemessenheit** entspricht. Ausserdem kann der Benutzer wählen ob er alle Aufgaben/Termine angezeigt haben oder eine neue Aufgabe erstellen möchte.

NEUER TERMIN

Will er neue Aufgaben respektive Termine einfügen, so drückt der Benutzer auf die Schaltfläche «Neuer Termin» und gelangt zur folgenden Maske:

The screenshot shows a window titled 'Terminator' with a standard Windows-style title bar (minimize, maximize, close buttons). Inside the window, the title 'Neuer Termin' is centered at the top. Below the title, there are several input fields and controls:

- A text input field labeled 'Bezeichnung'.
- A checkbox labeled 'Test'.
- A dropdown menu labeled 'Fach' with a list of options: 'Englisch', 'Franz', and 'Neues Fach'.
- A date input field labeled 'dd/mm/yyyy' next to a calendar icon.
- At the bottom, there are two buttons: 'Abbruch' (highlighted with a black border) and 'Speichern' (highlighted with a green background).

Hier trägt der Benutzer die entsprechenden Daten ein, wie zum Beispiel die Bezeichnung der Aufgabe und das Datum, auf wann die Aufgabe zu erledigen respektive wann der Termin. Des Weiteren kann er mittels angewählter Checkbox festlegen, ob die Aufgabe ein Test ist.

ZUFÜGEN EINES FACHES

Ausserdem muss der Benutzer die Aufgabe einem Fach zuordnen. Neue Fächer kann er mittels der Funktion im Dropdown hinzufügen. Dies würde sich nach der Aufgabe N2-1 richten.

The screenshot shows the 'Terminator' application window with the title 'Terminator'. Inside, there is a form titled 'Neuer Termin'. The form contains the following elements:

- A text input field labeled 'Bezeichnung'.
- A checkbox labeled 'Test'.
- A date input field labeled 'dd/mm/yyyy' with a calendar icon.
- A dropdown menu labeled 'Fach' with the following options: '...', 'Englisch', 'Franz', and 'Neues Fach'.
- Two buttons at the bottom: 'Abbruch' and 'Speichern'.

A red arrow points from the 'Fach' dropdown menu to a secondary window titled 'N2-1'. This window shows a list of sports: 'Fussball', 'Rennen', 'Schwimmen', and 'Volleyball'. The 'Volleyball' option is currently selected. At the bottom of the 'N2-1' window is an 'add Item' button.

Der Benutzer kann die Funktion jederzeit abbrechen und zurück zur Startmaske kommen, indem er die Schaltfläche «Abbruch» betätigt. Dadurch gehen die Daten jedoch verloren. Hat er alle zwingenden Daten angegeben, kann er die Aufgabe speichern und gelangt anschliessend zurück zur Startmaske. Hier liegt der Grundsatz der **Fehlerrobustheit** beziehungsweise der **Erwartungskonformität** vor.

ALLE TERMINE

Will er alle Aufgaben respektive Termine angezeigt bekommen, so drückt der Benutzer auf die Schaltfläche «alle Termine» und gelangt zur folgenden Maske:

Fach	Aufgabe	Erledigt	Datum
Englisch	Voci	<input type="checkbox"/>	03/04/2020
English	Buch S34	<input checked="" type="checkbox"/>	03/04/2020
Franz	Pronomen üben	<input type="checkbox"/>	03/04/2020
Deutsch	Grammatik Test	<input checked="" type="checkbox"/>	03/04/2020
Deutsch	Buch S44	<input checked="" type="checkbox"/>	05/04/2020
Franz	Test Pronomen	<input type="checkbox"/>	11/04/2020

Filtern nach: Fach* Test

Fach: Englisch, Franz, Deutsch

☒ Erledigte anzeigen

Falls Zeit

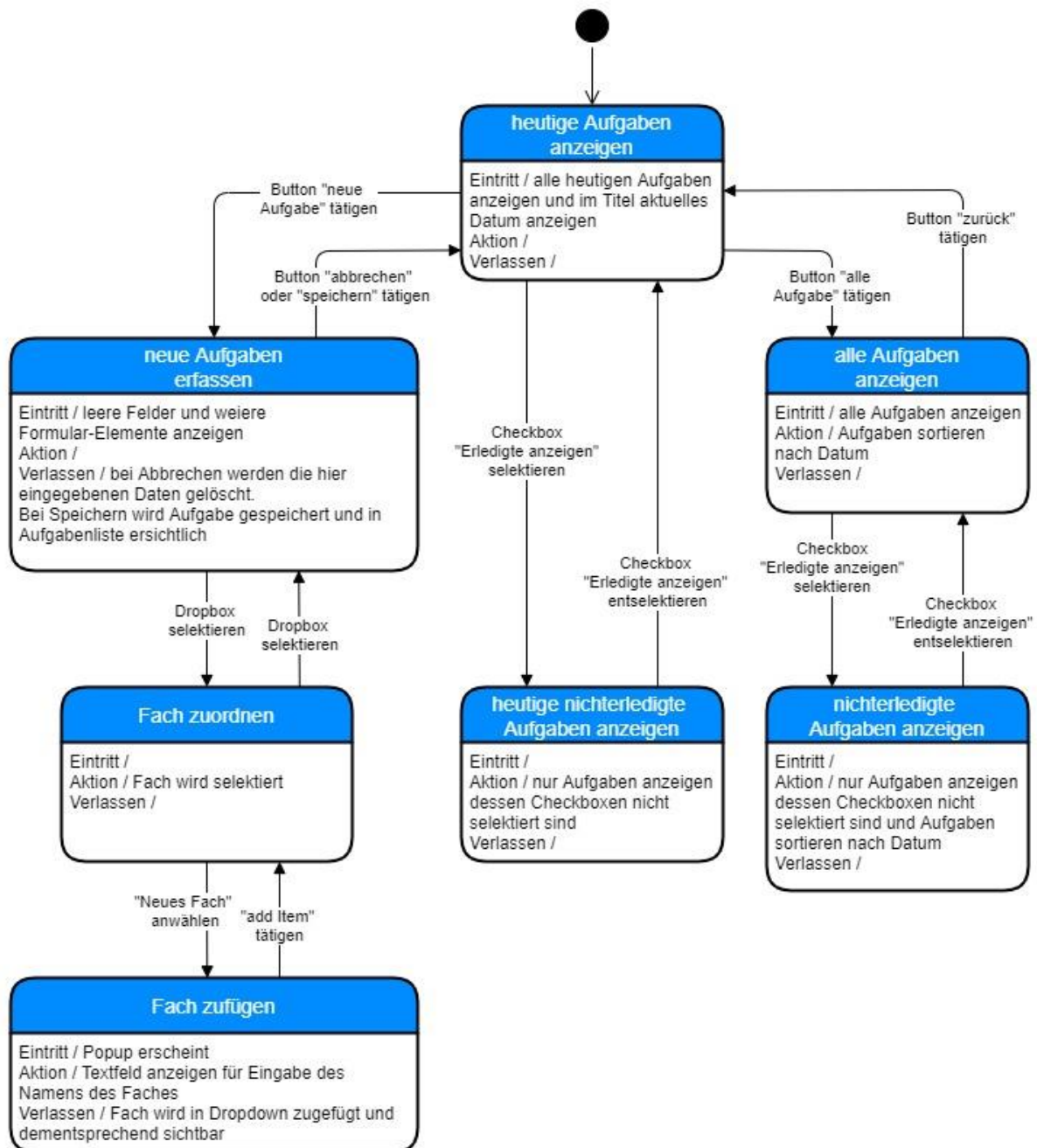
Falls Zeit

Hier ist dasselbe Prinzip wie bei der Startmaske. Der Unterschied hier ist, dass dem Benutzer nicht nur die Aufgaben und Termine des aktuellen Tages angezeigt werden, sondern, wie der Titel es beschreibt, alle Termine. Auch hier kann er mittels der Checkbox entscheiden, ob ihm abgehakte Aufgaben angezeigt werden oder nicht. Mittels der Schaltfläche neben der Beschreibung «Alle Termine» gelangt der Benutzer wieder zurück zur Startseite.

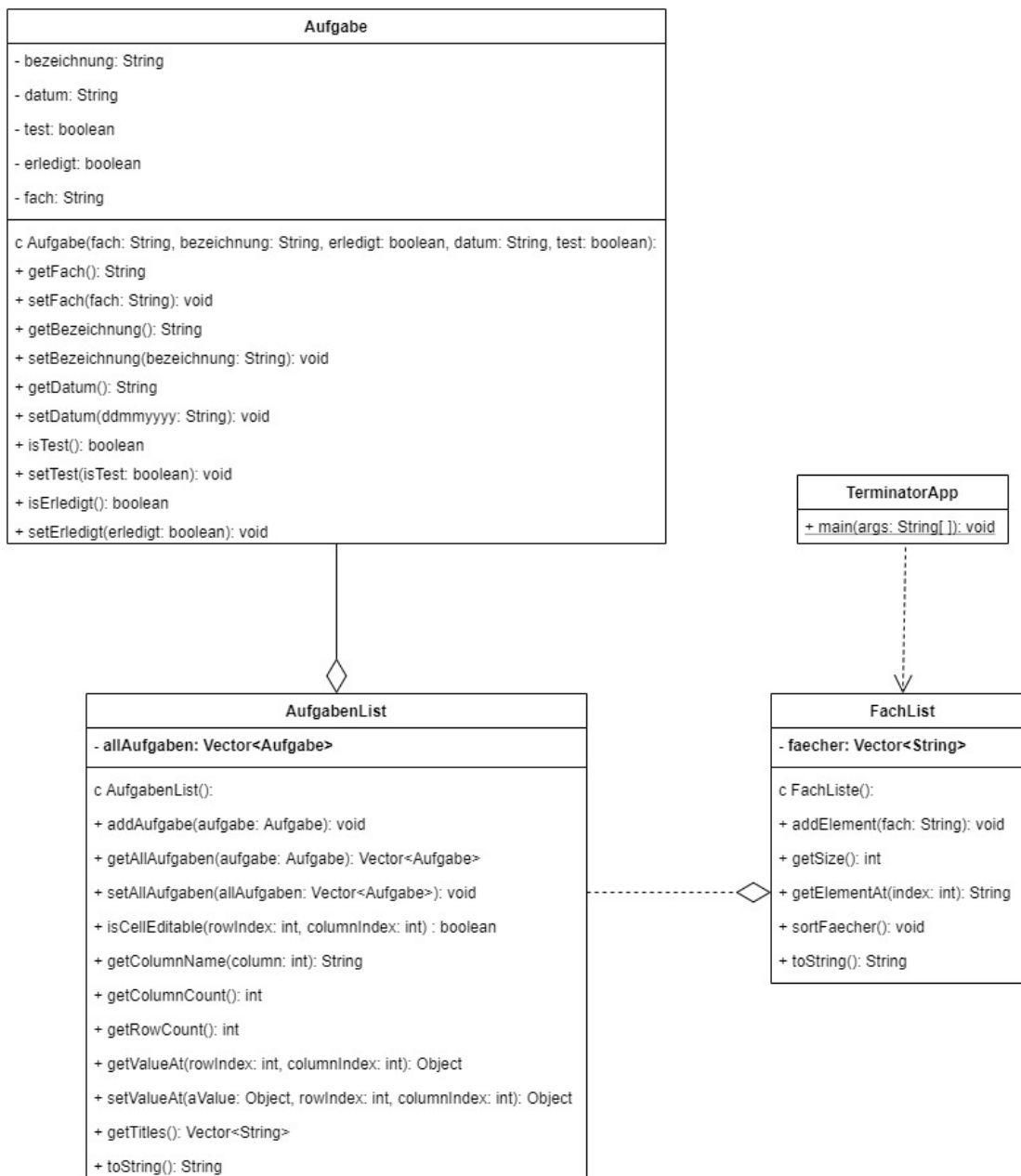
Wie in der Beschreibung bei Teilaufgabe 1 erwähnt, stehen dem Benutzer folgende Funktionen noch zur Verfügung, falls die Zeit dafür ausreicht:

Der Benutzer kann mittels des Dropdown-Menüs nach Tests, wobei ihm nur Tests angezeigt werden, oder nach Fach filtern. Sollte der Benutzer nach Fach filtern wollen, wird ihm das nächste Dropdown-Menü aktiviert, wo er dann die Fächer auswählen kann. Ansonsten bleibt dieses Dropdown-Menü deaktiviert. Hier kann auf den Grundsatz der **Fehlerrobustheit** verwiesen werden.

TEILAUFGTRAG 3 – ZUSTANDSDIAGRAMM



TEILAUFGTRAG 4 – KLASSENDIAGRAMM



Die Klasse **Aufgabe** repräsentiert eine Aufgabe mit zugehörigen Begriffen.

Die Klasse **AufgabenList** repräsentiert die Sammlung aller Aufgaben.

Die Klasse **FachList** repräsentiert eine Sammlung aller Fächer.

TEILAUFGTRAG 5 – ZUSTANDSDIAGRAMM DER SCHNITTSTELLEN

BEGRÜNDUNG DER WAHL

«heutige Aufgaben»

- Für die Auflistung der Aufgaben und Termine werde ich eine Tabelle (*JTable*) verwenden, da hier **mehre** Werte in einer geordneten und übersichtlichen Struktur angezeigt werden. Bei einer grossen Darstellung von Aufgaben, kann der Benutzer durch diese scrollen.
- Um eine neue Aufgabe zu erfassen oder zu einem anderen Fenster GUI-Fenster zu gelangen («alle Aufgaben»), werden hier Schaltflächen (*JButton*) verwendet.
- Um erledigte Aufgaben abhaken zu können, als auch erledigte Aufgaben anzeigen zu lassen, werden hier Checkboxen (*JCheckBox*) verwendet.

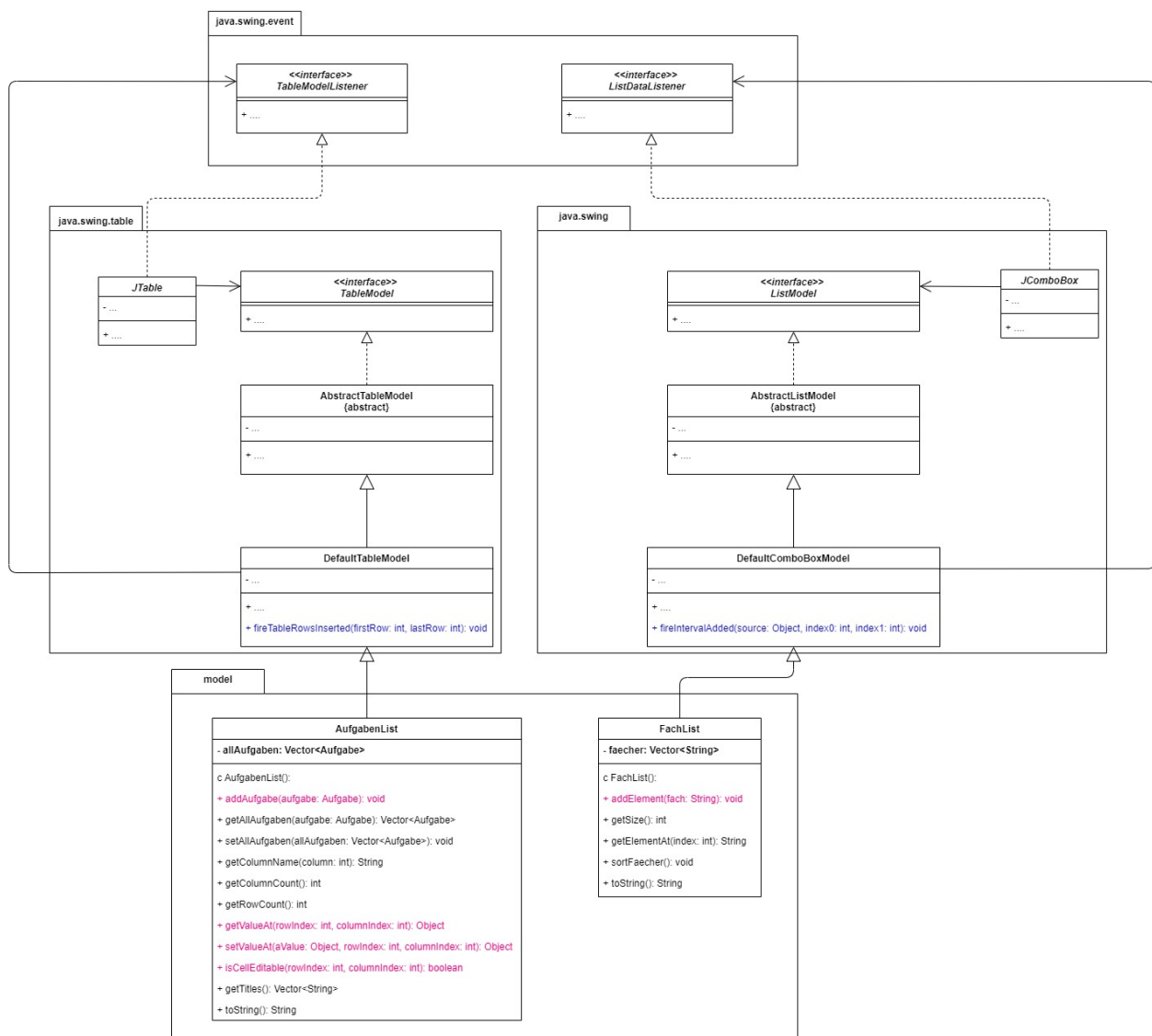
«alle Aufgaben»

- Für die Auflistung der Aufgaben und Termine werde ich eine Tabelle (*JTable*) verwenden, da hier **mehre** Werte in einer geordneten und übersichtlichen Struktur angezeigt werden. Bei einer grossen Darstellung von Aufgaben, kann der Benutzer durch diese scrollen.
- Um erledigte Aufgaben abhaken zu können, als auch erledigte Aufgaben anzeigen zu lassen, werden hier Checkboxen (*JCheckBox*) verwendet.
- Um wieder zurück zur Startseite (heutige Aufgaben) zu kommen, wird hier eine Schaltfläche (*JButton*) verwendet.

«neue Aufgabe»

- Für die Eingabe der Bezeichnung der Aufgabe wird ein Textfeld (*JTextField*) verwendet, da hier nur eine Kurzbeschreibung von maximal zwanzig Zeichen erfasst werden soll.
- Für die Eingabe des Datums wird ein Textfeld (*JTextField*) verwendet, da hier nur 10 Zeichen erfasst werden sollen.
- Für die Angabe, dass die Aufgabe ein Test ist, wird hier eine Checkbox (*JCheckBox*) verwendet.
- Für die Zuordnung eines Faches respektive Auswahl der Fächer wird eine Dropdownbox (*JComboBox*) verwendet. Es ist immer nur ein Begriff in der Auswahl und er soll dem Benutzer klar angezeigt werden.
- Zur Bestätigung der Eingabe wird je eine Schaltfläche (*JButton*) verwendet. Das entspricht der Erwartungskonformität der Benutzer.

DIAGRAMM



Die pink markierten Methoden müssen in der Model-Klasse überschrieben werden, um die Daten in der entsprechenden GUI-Komponente korrekt wiederzugeben.

Die blau markierte Methode muss in der Model-Klasse aufgerufen werden, sobald Daten geändert wurden (hier bei `addElement(...)`, `addAufgabe(...)` und `setValueAt(...)`).

ENDPRODUKT - BERICHT

Sehr vieles konnte ich so implementieren wie ich es mir vorgestellt hatte. Jedoch gab es während der Erstellung einige Probleme im Privatleben, welche mich bei der Arbeit sehr eingeschränkt und beeinflusst hatten. Dennoch blieb ich weiter am Ball und versuchte alles so gut wie möglich zu erledigen.

Einige Features konnte ich leider nicht so erledigen, wie ich es mir vorgenommen hatte, da dies zu kompliziert war wie zum Beispiel die Checkboxen in der JTable. Ebenso war es sehr aufwändig die Reihe der JTable, welche Tests waren, rot zu färben und wenn man diese ausblendet, dass die Reihe eingefärbt bleibt, obwohl dieser Termin eigentlich ausgeblendet wurde. Hierbei holte ich mir Hilfe von meinem Freund und so googelten wir nach der entsprechenden Lösung. Die Quellen sind dementsprechend angegeben im Code.

Leider kam ich auch nicht dazu die Filter-Features miteinzubauen, doch bin ich sehr stolz auf das Endprodukt, da es sehr viel Arbeit gekostet hatte, aber mir die Freude nie entgangen ist daran zu arbeiten.