

Programmierung II – Übung 7

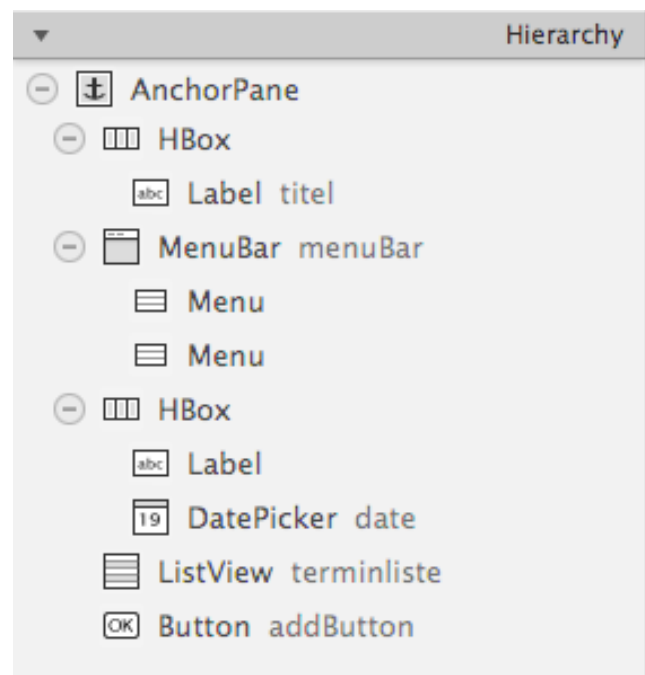
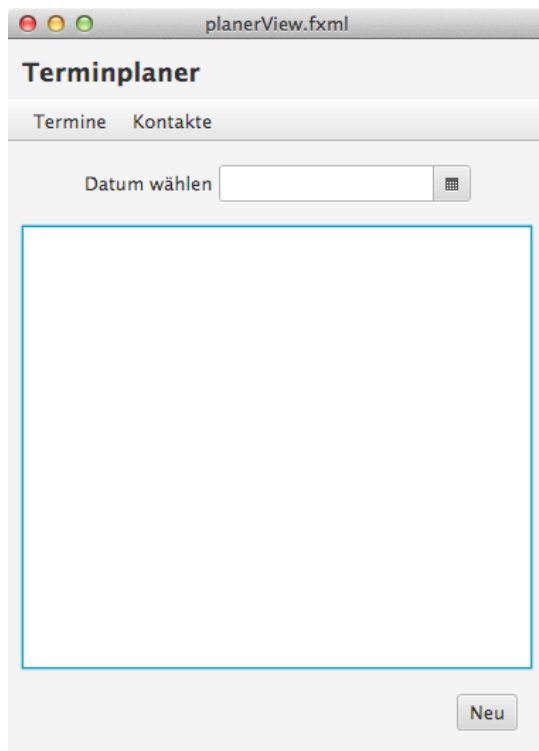
In dieser Übung soll eine GUI für den Terminkalender gebaut werden.

Öffnen Sie Ihr Adressbuch-Projekt aus der letzten Übung und legen Sie darin ein neues Paket *terminplaner* an. Kopieren Sie die Java-Dateien aus dem Ordner *Sourcen* in das Paket. Sie finden hier Klassen für die Terminverwaltung (diese werden in der nächsten Vorlesung genauer vorgestellt) sowie die Klasse *Terminplaner*, *StartPlaner*, *PlanerViewController* und die FXML-Datei für den Aufbau der Planerview.

Sehen Sie sich zunächst die Klasse *Terminplaner* an. Diese ist von *TerminVerwaltung* abgeleitet und erweitert diese um ein Attribut für das Speichern eines Terminplaner-Besitzers. Im Konstruktor der Klasse werden ein paar Testtermine für den 24.10.2014 erstellt und dem Terminplaner hinzugefügt. Die Methoden der Klasse *Terminplaner* werden von der GUI des Terminkalenders aufgerufen, um die Termine zu verwalten.

Aufbau der Anzeige für die Termine eines ausgewählten Tages

Öffnen Sie die Datei *planerView.fxml* im *SceneBuilder*. Diese enthält lediglich eine *AnchorPane* in der Größe der zu bauenden View der Anwendung. Bauen Sie die unten angezeigte GUI nach. Versehen Sie Ihre GUI-Elemente mit den *fx:ids*, die unten in der Hierarchie-Ansicht zu sehen sind. Die Menüs benötigen keine *ids*, weil diese im Controller über einen Index angesprochen werden können.



Aufbau des Controllers

Erweitern sie die Klasse *PlanerViewController* um die annotierten Attribute für die GUI-Elemente mit einer *fx:id*.

Wie Sie sehen, arbeitet der Controller mit einem Objekt vom Typ *Terminplaner* (Attribut *planer*) und einem Objekt vom Typ *Adressbuch* (Attribut *adressen*) zusammen.

Methode ***initialize***:

Initialisieren Sie hier zunächst die beiden Attribute. Geben Sie bei der Erzeugung des *Terminplaner*-Objekts den Kontakt zum Namen „john“ aus dem Adressbuch als Besitzer an.

Setzen Sie bei dem *DatePicker* das aktuelle Datum als Startdatum.

Melden Sie als Event Handler des *DatePicker*-Objekts eine Methode *void showTermine()* an und als Event Handler des Buttons eine Methode *void addTermin()*. Die Rümpfe der Methoden können Sie zunächst noch leer lassen.

Lagern sie das Konfigurieren des Menüs aus in eine Methode *void configureMenu()* und rufen Sie diese in der Methode *initialize()* auf.

Lagern Sie das Konfigurieren der *ListView* in die Methode *void configureList()* aus und rufen Sie diese ebenfalls in *initialize()* auf.

Methode ***configureMenu***:

Bauen Sie hier in das erste Menü (Index 0 bzw. Bezeichner *Termine*) die Menüpunkte Laden und Speichern ein. Bauen Sie in das zweite Menü (Index 1 bzw. Bezeichner *Kontakte*) den Menüpunkt Kontakte Bearbeiten ein.

Melden Sie bei den Menüpunkten die folgenden Event Handler an:

Laden: *loadTermine()*

Speichern: *saveTermine()*

Kontakte Bearbeiten: *editKontakte()*

Methode ***configureList***:

Initialisieren Sie hier die *ListView* mit einer leeren *ObservableList* vom Typ *Termin*. In dieser Liste sollen die Termine eines ausgewählten Datums angezeigt werden.

Melden Sie als Event Handler für die Auswahl eines Termins der Liste eine Methode *void editTermin()* an. Der Methodenrumpf kann hier zunächst noch leer bleiben.

Methode ***getAdressbuch***:

Schreiben Sie eine getter-Methode für den Zugriff auf das Attribut *adressen*.

Methode ***getSelectedDate***:

Schreiben Sie eine öffentliche Methode *LocalDate getSelectedDate()*, die das aktuell im *DatePicker* eingestellte Datum zurückgibt.

Methode ***showTermine()***:

Hier sollen die Termine des aktuell im *DatePicker* eingestellten Datums aus dem Terminplaner geholt und in der *ListView* angezeigt werden. Sorgen Sie dafür, dass die *ListView* zunächst geleert wird. Rufen Sie diese Methode in der Methode *initialize()* auf, um die Termine des Startdatums anzuzeigen.

Probieren Sie zunächst aus, ob alles korrekt angezeigt wird. Bauen Sie das Projekt erst neu (mit *Run->Clean and Build Project*). Starten Sie das Projekt mit der Klasse *StartPlaner*. Stellen Sie im *DatePicker* den Termin 24.10.2014 ein und prüfen Sie ob alle Testtermine korrekt aufgelistet werden.

Koppeln mit der GUI des Adressbuchs

In der Methode *editKontakte()* soll die GUI des Adressbuchs geöffnet werden.

Da der *AdressbuchViewController* das *Adressbuch*-Objekt dieses Controllers benutzen soll, müssen Sie ihn um einen Konstruktor erweitern, dem man eine *Adressbuch*-Instanz übergeben kann. Schreiben Sie einen solchen Konstruktor.

Verwenden Sie den *ViewHelper*, um die zur FXML-Datei gehörige *URL* und eine *Controller*-Instanz in einem Extra-Fenster anzuzeigen. Wenn Sie nicht mehr wissen, wie das geht, sehen Sie in dem Skript zu Vorlesung 6 nach (zusätzliche Dialoge). Nutzen Sie hierfür die View *adressbuchView.fxml* und den *AdressbuchViewController* aus dem Paket *adressbuch*.

Damit das Ganze funktioniert, müssen sie bei der *adressbuchView.fxml* den Controller löschen, ansonsten kann diese nicht von außen mit einem Controller versehen werden. Bauen Sie das Projekt erst neu (mit *Run->Clean and Build Project*), da sonst Ihre Änderungen unter Umständen nicht wirksam werden.