

## U05 | ToDo-Liste-Mit-Datum



Figure 1: Cover für die vierte Übungsaufgabe

### Aufgabe

Diese Aufgabe erweitert die ToDo-Liste aus U04. Beachten sie aber, dass nicht nur Code hinzugefügt werden, sondern auch ersetzt bzw. entfernt werden muss. Der Nutzer muss nun, zusätzlich zum Task-Titel, auch ein Fälligkeitsdatum für jeden Task angeben. Dies geschieht über ein zusätzliches Eingabefeld (*siehe Screenshots*). Modellieren Sie den komplexen Datentyp (neue Klasse) `TaskItem` der den Titel und das Fälligkeitsdatum enthält. Anschließend müssen Sie ein Layout für die Listeneinträge anlegen. Um das View mit den Daten der `TaskItems` zu befüllen, schreiben Sie noch einen `ToDoListAdapter`, welcher von `ArrayAdapter` erbt und diesen erweitert.

### Hinweise

1. Der `ArrayAdapter` wird benötigt, um Daten mit einem `View` zu verbinden. Für komplexere `Views`, wie in dieser Übungsaufgabe, muss von `ArrayAdapter` geerbt und die Methode `getView()` überschrieben werden.
2. Der `DatePickerDialog` fordert den Benutzer auf, ein Datum aus einem Kalender auszuwählen. Der `DatePickerDialog` ist bereits im Starter-Projekt integriert.
3. Ein Datum wird in Java gewöhnlich als Objekt der Klasse `Date` gespeichert. Um dieses in einen formatierten `String` umzuwandeln (und umgekehrt)

benutzt man die Klasse `DateFormat`. Für Umwandlungen zu einzelnen **Integern**, etwa für Tag, Monat und Jahr, benutzt man die Klasse `GregorianCalendar`. Methoden dafür sind im Starterpaket schon enthalten.

4. Tragen sie alle *Texte* die dem Nutzer angezeigt werden in die `strings.xml`-Datei ein.
5. Damit die Höhe der `TaskItems` den Android Design Guidelines entspricht, kann man diese im Layout folgendermaßen definieren:  
`android:layout_height="?android:attr/listPreferredItemHeight"`

## Vorgehen

### Layout

1. Laden Sie das Starter-Paket zur Übung herunter. Dabei handelt es sich um die Lösung zur Aufgabe von letzter Woche, die um einige Methoden und die Klasse `TaskItem` ergänzt wurde. Des Weiteren wurde die `MainActivity` um Methoden zur Abfrage des Datums ergänzt.
2. Passen Sie zunächst die `activity_main.xml` Layout-Datei an, wie in den Screenshots im Anhang zu sehen. Mit Hilfe des Attributs `android:focusable` können Sie verhindern, dass der Text im Datums-Textfeld manuell geändert werden kann (die Eingabe soll über den `DatePickerDialog` geschehen).
3. Erstellen Sie nun eine neue `layout.xml`-Datei im `layout`-Ordner, welche das Aussehen eines einzelnen `TaskItems` definiert. Es soll der Titel und das Fälligkeitsdatum nebeneinander gezeigt werden.

### TaskItem-Klasse

1. Ergänzen Sie nun die Klasse `TaskItem`, welche den Titel eines Tasks und sein Fälligkeitsdatum speichert.
2. Erstellen Sie die benötigten Instanzvariablen und einen Konstruktor mit entsprechenden Parametern und weisen sie die Variablen entsprechend zu.
3. Die Methode `getTask()` soll den Titel des Tasks zurückgeben.
4. Mit der Methode `getDateFromString()` können sie das übergebene Datum in ein `GregorianCalendar` Objekt umwandeln welches sich zum speichern von Daten eignet.

### ToDoListAdapter implementieren

Es fehlt nun noch ein Adapter, welcher den komplexen Datentyp `TaskItem` mit dem Layout verknüpft. Ergänzen Sie hierfür die Klasse `ToDoListAdapter`. Diese soll von `ArrayAdapter<TaskItem>` erben.

1. Überschreiben Sie die Methode `getView(int position, View convertView, ViewGroup parent)`. Sie soll mit Hilfe des `LayoutInflater`s einen `View` mit dem Layout des `ToDoItems` erstellen und die `TextView`s des Layouts mit den Daten des entsprechenden `TaskItems` befüllen.
2. Passen Sie im Konstruktor die Referenz auf die Referenz des oben erstellten Layouts an.
3. Nutzen sie die `getFormattedDate()`-Methode um das Datum als String aus einem `TaskItem` Objekt zu holen.

### **ToDoActivity implementieren**

1. Tauschen Sie nun den bisher verwendeten `ArrayAdapter` durch den von Ihnen vervollständigten `ToDoListAdapter` aus.
2. Um ein neues `TaskItem` zu erstellen und der Liste hinzuzufügen, brauchen sie zuerst den Titel und das Datum, welche sie aus den entsprechenden Views auslesen können.
3. Passen Sie ggf. die vorhandenen Methoden an, oder erstellen Sie neue.
4. Um den Dialog zur Erfassung des Datums aufzurufen, verwenden sie die bereits angelegte Methode `createDatePickerDialog()`. Rufen sie auf dem zurückgegeben Objekt die Methode `show()` auf um den Dialog anzuzeigen.
5. Passen sie die `onDateSet()`-Methode so an, dass beim Aufruf dieser der Inhalt des Textfeldes ihres Date-EditText auf das gewählte Datum geändert wird.
6. Nun können Sie ein Objekt der Klasse `TaskItem` erstellen und der `ArrayList<TaskItem>` hinzufügen.

### **Quellen**

Große Teile dieser Anleitung wurden aus der Übungsaufgabe 5 des Sommersemesters 2013 von Thomas Wagner übernommen, welche von Thomas Wilhelm, Jonas Roberts und Tim Eichinger überarbeitet wurden.

### **Screenshots**

