



MobA 1 Android Apps – Christian Sutter 27374 – WiSe 18-19
Prof. Dr.-Ing. Thorsten Weiss

Dokumentation

Java Klassen:

- Affirmation
- MainActivity
- NewCardActivity
- EditCardActivity
- TimePickerFragment
- RVAdapter
- AlarmReceiver
- ServiceClass
- TinyDB

Affirmation.java



Affirmation-Objekt mit Parcelable-Interface. Kern der Anwendung. Wird mit Daten aus NewCardActivity.java befüllt, als Intent-Extra weitergereicht, und über MainActivity.java mit TinyDB-Aufruf in den SharedPreferences gespeichert.

Enthält die Variablen:

String:	affirmation	-> Affirmations-Text
Boolean:	remindMeOnceADay	-> Notification-Häufigkeit
Boolean:	remindMeTwiceADay	
Boolean:	remindMeThriceADay	
String:	firstReminderTime	-> Eingestellte Zeit
Boolean:	isEnabled	-> Zustand der Affirmation
String:	affirmationKeyString	-> Key für TinyDB

MainActivity.java

Zuständig für das Befüllen & Speichern der Affirmationen mit Daten aus NewCardActivity.java und EditCardActivity.java. Handlet die **onClick()**-Events der Affirmations-Karten. Initialisiert & updatet das Drawer-Menu.

onCreate() -> Erzeugt Instanz von TinyDB, **myTinyDB**. Prüft ob eine Affirmation von NewCardActivity.java bzw. EditCardActivity.java übergeben wurde. Schafft **onClick()**-Events für die EnableDisable-Buttons und die Edit-Buttons der Affirmations-Karten. Schafft Listener für FAB und Drawer.

FAB-**onClick()** -> Startet NewCardActivity.java.

onEnabledDisabledButtonClick() -> Ruft **enableDisableItem()** der jeweiligen Karte auf, benachrichtigt den Adapter, und ruft dann **initializeMenu()** auf um das Drawer-Menu zu updaten.

onEditButtonClick() -> Startet EditCardActivity.java mit den Daten der jeweiligen Karte aus den SharedPreferences (per TinyDB-Aufruf) als Intent-Extra.

enableDisableItem() -> Togglet die **isEnabled** Variable der jeweiligen Karte in der RecyclerView. Speichert die Änderung per TinyDB-Aufruf. Ruft **extractCodeFromKey()** mit **affirmationKeyString** der Affirmation als Parameter auf, um einen Integer Key für den PendingIntent des AlarmManagers zu erzeugen. Nachdem die Zeit mit **extractTime()** Aus dem **affirmationKeyString** extrahiert wurde, Cancelt bzw. erzeugt es den Alarm (je nachdem, ob **isEnabled** true oder false gesetzt wurde.)



initializeData() -> Durchläuft die Liste "myKeys" aus myTinyDB, in der die Keys aller Affirmationen gespeichert sind, und fügt die dazugehörige Affirmation der Liste affirmations hinzu. (Aus der Liste affirmations erstellt der RVAdapter.java später dann Affirmation-CardViews für die RecyclerView.)

Dann, wird der Zustand von isEnabled jeder Affirmation geprüft. Bei true wird der Service gestartet, der den Alarm für die jeweilige Affirmation setzt, bei false wird der Service gestoppt.

extractTime() -> Erzeugt aus dem firstReminderTime String einer Affirmation die Integer hour und minute, damit sie für das setzen der Alarm-Zeit verwendet werden können. (Siehe Aufruf in enableDisableItem())

Bsp.: firstReminderTime = "8:30 AM" --> hour = "8", minute = "30"

extractRecCodeFromKey() -> Erzeugt aus dem affirmationKeyString einer Affirmation, eine Integer Variable keyRecCode, damit dieser als Key für den PendingIntent eines AlarmManagers benutzt werden kann. Damit kann aus jeder Affirmation der dazugehörige Alarm abgeleitet werden. (Siehe Aufruf in enableDisableItem())

Bsp.: affirmationKeyString = "affirmation0" --> keyRecCode = "0"

initializeMenu() -> Erzeugt für die NavigationView des NavigationDrawers die beiden SubMenus subMenuEnabled und subMenuDisabled. Durchläuft dann die Liste "myKeys" aus myTinyDB und überprüft für jede Affirmation die Variable isEnabled. Jede Affirmation mit isEnabled == true wird als SubMenu dem SubMenu subMenuEnabled hinzugefügt. Jede Affirmation mit isEnabled == false wird als SubMenu dem SubMenu subMenuDisabled hinzugefügt.

makeKey() -> Erzeugt String Key für myTinyDB aus "affirmation" und dem Parameter i (welcher die Länge der Liste "myKeys" aus myTinyDB ist) (siehe Aufruf in onCreate()) Bsp.: "affirmation" + "0" = "affirmation0"

onBackPressed() -> Schließt Navigation-Drawer Menu wenn daneben geklickt wird.

getTinyDB() -> Macht die Instanz myTinyDB für andere Klassen erreichbar.

getAffirmations() -> Macht die Liste affirmations für andere Klassen erreichbar.

onNavigationItemSelected() -> **OnClick()**-Event für die SubMenus von subMenuEnabled und subMenuDisabled. Durchläuft die Liste "myKeys" aus myTinyDB und gleicht den Titel des SubMenus mit der affirmation Variable der jeweiligen Affirmation ab, bis es die entsprechende Affirmation findet, scrollt zur Position der Affirmations-Karte in der RecyclerView und schließt dann das Navigation-Drawer Menu.

NewCardActivity.java



Zuständig für den Nutzer-Input über `activity_new_card.xml`. Speichert Daten aus der `EditText`-View, den `RadioButtons`, und dem `TimePicker`-Dialog in einer `Affirmation`.

`onCreate()` -> Holt sich die Views aus `activity_new_card.xml`. Setzt `onClickListener` für `cancel`-, und `saveButton`, außerdem für den "Back"-Pfeil der `ActionBar`. `Cancel` und "Back" führen zur `MainActivity.java` zurück, `Save` speichert die Daten als `Affirmation` und schickt sie als `Intent-Extra` an `MainActivity.java`.

`hideKeyboard()` -> Versteckt das `Soft-Keyboard`, wenn außerhalb des `Keyboards` getippt wird.

`showTimePickerDialog()` -> Legt neues `TimePicker`-Fragment an, und zeigt es.

`onRadioButtonClicked()` -> einen `RadioButton` zu checken, unchecked alle anderen.

EditCardActivity.java

Zuständig für den Nutzer-Input über `activity_edit_card.xml`. Erhält eine `Affirmation` als `Intent-Extra` von der `MainActivity.java`, und füllt die Views von `activity_edit_card.xml` mit deren Daten. Setzt `onClickListener` für `delete`- und `saveButton`, außerdem für den "Back"-Pfeil der `ActionBar`. `Save` speichert die eingegebenen Daten als `editedAffirmation`, übernimmt aber den `TinyDB-Key` der `editAffirmation` (so wird die `Affirmation`, nachdem sie per `Intent-Extra` an `MainActivity.java` zurückgegeben wurde überspeichert.) "Back" führt zur `MainActivity.java` zurück. `Delete` ruft `showDeleteAffirmationDialog()` auf.

`onCreate()` -> Holt sich die Views aus `activity_edit_card.xml`. Speichert `Intent-Extra` als `editAffirmation` und füllt die Views der `activity_edit_card.xml` mit den `editAffirmation`-Daten.

`onTouch()` -> Listener, der das `Soft-Keyboard` versteckt, wenn außerhalb des `Keyboards` geklickt wird.

`showDeleteAffirmationDialog()` -> Nutzt `AlertDialog.Builder` um einen `Delete-/Cancel`-Dialog zu öffnen. `Delete` löscht die `Affirmation` (und den dazugehörigen `Key` in "`myKeys`") und führt zur `MainActivity.java` zurück. `Cancel` bricht ab.

`hideKeyboard()` / `onRadioButtonClicked()` / `showTimePickerDialog()` bzw. `showTimePickerDialog()`-> Siehe `NewCardActivity.java` bzw. `MainActivity.java`.

TimePickerFragment.java



TimePickerFragment, das von NewCardActivity.java, bzw. EditCardActivity.java erzeugt wird.

`onCreateDialog()` -> Erzeugt TimePickerDialog mit aktueller Uhrzeit als Ausgang.

`onTimeSet()` -> Nutzt StringBuilder, um aus der angegebenen Zeit einen String zu erzeugen, und diesen in die `tvDisplayTime`-View der Activity (NewCardActivity.java, oder EditCardActivity.java) zu speichern)

Bsp.: `hour = "8"`, `minute = "30"` --> `String = "8:30 AM"`

RVAdapter.java

Adapter-Klasse. Verbindet das in `card_view_blueprint.xml` festgelegte Karten-Layout mit den in NewCardActivity.java, bzw. EditCardActivity.java festgelegten Daten, Außerdem ordnet es die `onClick()`-Events die in der RecyclerView passieren den jeweiligen Karten zu.

`onBindViewHolder()` -> Füllt die Views der `card_view_blueprint.xml`. Setzt den Text des `enableDisabledButtons` und die Farbe der Karte abhängig von der Variable `isEnabled` der Affirmation.

AlarmReceiver.java

Empfängt den Alarm zur festgelegten Uhrzeit, erstellt per NotificationCompat.Builder eine Notification mit dem Inhalt der Affirmation als Text.

`onReceive()` -> De-Serialisiert das Intent-Extra des PendingIntents der Service-Class.java (enthält den `affirmationKeyString` der Affirmation deren Alarm empfangen wurde.) **Dieses Extra geht VERLOREN, wenn die App geschlossen wird!** Falls für die Affirmation immer noch `isEnabled == true` gilt, wird per NotificationCompat.Builder eine Notification mit dem Inhalt der Affirmation als Text erstellt. Mit `notificationManager.notify()` wird die Notification an den Nutzer gesendet.

ServiceClass.java



Service-Klasse, die einen `Intent` mit `Affirmations-Key` als `Extra` erhält. Wird von `MainActivity.java` aufgerufen. Setzt `Alarm` je nachdem welcher der 3 `Booleans` (`remindMeOnceADay`, `remindMeTwiceADay`, `remindMeThriceADay`) `== true` ist.

`onStartCommand()` -> Erzeugt abhängig der `remindMe`-`Booleans` `Alar`me.
`remindMeOnceADay == true` erzeugt einen `Alarm` zur Zeit von `firstReminderTime`.

`remindMeTwiceADay == true` erzeugt einen ERSTEN `Alarm` zur Zeit von `firstReminderTime`, und einen ZWEITEN `Alarm` zur Zeit von `firstReminderTime + "3h"`.

`remindMeThriceADay` erzeugt einen ERSTEN `Alarm` zur Zeit von `firstReminderTime`, und einen ZWEITEN `Alarm` zur Zeit von `firstReminderTime + "2h"` und einen DRITTEN `Alarm` zur Zeit von `firstReminderTime + "4h"`.

`setAlarm()` -> Erzeugt `Alarm(e)` und verlängert diese um 24 Stunden, sollten sie verstrichen sein. Serialisiert den `affirmationsKeyString` der `Affirmation` und übergibt sie dem `Intent` als `Extra`, welcher wiederum dem `PendingIntent` mitgegeben wird. Dieser `PendingIntent` wird dann dem `AlarmManager` übergeben (und später von der `onReceive()`-Funktion der `AlarmReceiver.java`-Klasse empfangen). Dieses `Intent-Extra` geht VERLOREN, wenn die App geschlossen wird!

`onDestroy()` / `onTaskRemoved()` -> Versuch die `Alar`me am Leben zu erhalten, sollte die App aus dem Speicher geschoben werden. `Alarm` überlebt, `Intent-Extra` nicht.

`extractRecCodeFromKey()` / `extractTime()` -> Siehe `MainActivity.java`.

TinyDB.java

Vereinfacht Aufrufe an `SharedPreferences` in einzeilige Aufrufe. Kann auch Listen von `Strings` und `Integern` speichern.

`putObject()` / `getObject()` / `putListString()` / `getListString()` -> Wurden verwendet. Klasse wurde jeweils um `preferences.edit().commit();` erweitert, da `TinyDB` standardmäßig noch nicht commitet.



Sei wer du sein willst – schon heute!

