

## Übungsaufgabe 3

# Laufapp

## 1 Aufgabe

Schreiben Sie eine Anwendung, mit der durch Eingabe der zurückgelegten Strecke, der dafür benötigten Zeit und eventuell eingelegten Pausen die Durchschnittsgeschwindigkeit (*Pace* in min/km) und der Kalorienverbrauch berechnet werden.

Benutzen Sie eine **Activity**, **EditText**-Views um die Eingabe zu ermöglichen sowie weitere Views und Buttons für die beschriebenen Funktionen.

Dem Nutzer soll es möglich sein, die Strecke (in km), die Zeit (in min) und die Pausen (in min) anzugeben und sich anschließend durch Drücken eines Buttons das Ergebnis anzeigen zu lassen.

## 2 Hinweise

- Die Abfrage des eingegebenen Inhalts eines **EditTexts** erfolgt über getter-Methoden. Um mögliche Fehleingaben vorzubeugen, kann dem Nutzer gleich bei Auswahl eine Zahlentastatur angezeigt werden. Hierzu kann das Attribut **inputType** entsprechend gesetzt werden. (**inputType** für Eingabefelder finden Sie in der API unter <http://developer.android.com/reference/android/R.attr.html#inputType>).
- Sie können in der **Activity** zusätzliche Methoden definieren. Nutzen Sie dies, um bestimmte Abläufe auszulagern. Die Berechnungen (*Pace*, Kalorien) übernimmt die bereitgestellte **Calculator**-Klasse.
- Methoden und Attribute der in der Android-Programmierung verwendeten Klassen können in der Android-API-Dokumentation eingesehen werden: <http://developer.android.com/reference/packages.html>
- Methoden und Attribute der in der Java-Programmierung verwendeten Klassen können in der Java-API-Dokumentation eingesehen werden: <http://docs.oracle.com/javase/7/docs/api/>
- Nutzen Sie die automatische Codevervollständigung durch die Entwicklungsumgebung. Hinweise zur Beseitigung von Fehlern erhalten Sie mittels **Alt + Enter**. Mehr hierzu unter <https://developer.android.com/sdk/installing/studio-tips.html>

### 3 Vorgehen

#### 3.1 1. Teil:

- a) Laden Sie das Starter-Paket zur Übung herunter. Dabei handelt es sich um ein leeres Standardprojekt, das nur um die Klasse `Calculator` erweitert worden ist.
- b) Diese Klasse stellt die drei öffentlichen Methoden `setValues(int distance, int time, int pause)`, `calculatePace()` und `calculateKcal()` zur Verfügung.
- c) Verändern Sie das vorgegebene Layout. Stellen Sie sicher, dass Sie ein `LinearLayout` verwenden. Es werden drei Elemente für die Eingabe (`EditText`) und drei Labels (`TextView`) benötigt. Außerdem sind ein `Button`, der die Berechnung startet, sowie ein Textfeld, welches das Ergebnis anzeigt, erforderlich.
- d) Ergänzen Sie `strings.xml` um Elemente für Beschriftung der `TextViews` und des `Buttons`.
- e) Erstellen Sie notwendige Klassenvariablen (nur) für die zu manipulierenden Views (Werte auslesen oder Listener).
- f) Referenzieren Sie die Views in der `onCreate()`-Methode. Weisen Sie dem Button einen `OnClickListener` zu:
  - i. Nutzen Sie das bekannte Vorgehen, um die `onClick()`-Methode zu überschreiben.
  - ii. Lesen Sie in der `onClick()`-Methode die eingegebenen Werte aus und berechnen Sie aus diesen Werten die Durchschnittsgeschwindigkeit und den Kalorienverbrauch. Nutzen Sie dazu die bereitgestellte Klasse `Calculator`.
  - iii. Geben Sie anschließend das Ergebnis in den entsprechenden `TextViews` aus.
  - iv. Sorgen Sie dafür, dass mögliche Falscheingaben, z.B. ein Wort anstatt einer Zahl behandelt werden.

#### 3.2 2. Teil: Intents und Extras

- a) Bauen Sie die App um. Das Ergebnis soll jetzt in einem zweiten Bildschirm angezeigt werden. Auf diesem befinden sich nur die Views für die Anzeige der Ergebnisse. Verwenden Sie die Möglichkeit, an `Intents` `Extras` anzuhängen um die Werte aus den Textfeldern in die andere `Activity` zu schicken. Dort nutzen Sie dann wie gehabt die `Calculator`-Klasse zur Berechnung des Ergebnisses.
- b) Verwenden Sie als Schlüssel (*Key-Value*-Mechanismus) für die `Extras` keine hart kodierten `Strings`, sondern lagern Sie diese in eine Klasse `Constants.java` aus.
- c) Sorgen Sie nun abschließend dafür, dass die Ergebnisse in einem sinnvollen Format dargestellt werden: Kalorien auf zwei Nachkommastellen gerundet (verwenden Sie die

Klasse `DecimalFormat` der Java-API) und die Pace umgerechnet in Minuten und Sekunden (Das Ergebnis von `calculatePace()` ist dezimal, d.h. ein Ergebniswert von 4,5 bedeutet 4 Minuten 30 Sekunden).

- d) **Bonus:** Die Screenshots zu Teil 2 zeigen auch eine `ActionBar`, die in Android zur Navigation zwischen verschiedenen Bildschirmen genutzt wird. In der Ergebnis-`Activity` hat der Nutzer mithilfe des „Zurück“-Pfeils links oben die Möglichkeit, zur Eingabemaske zurückzukehren. Implementieren Sie diese Funktionalität. (<https://developer.android.com/training/basics/actionbar/index.html> und <https://developer.android.com/design/patterns/navigation.html>)

## 4 Anhang: Screenshots

### 4.1 Teil 1

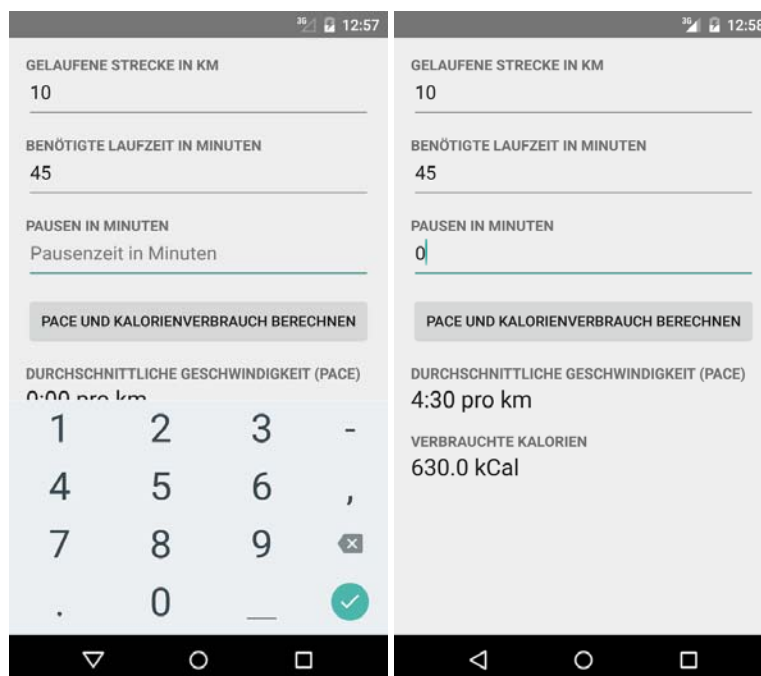
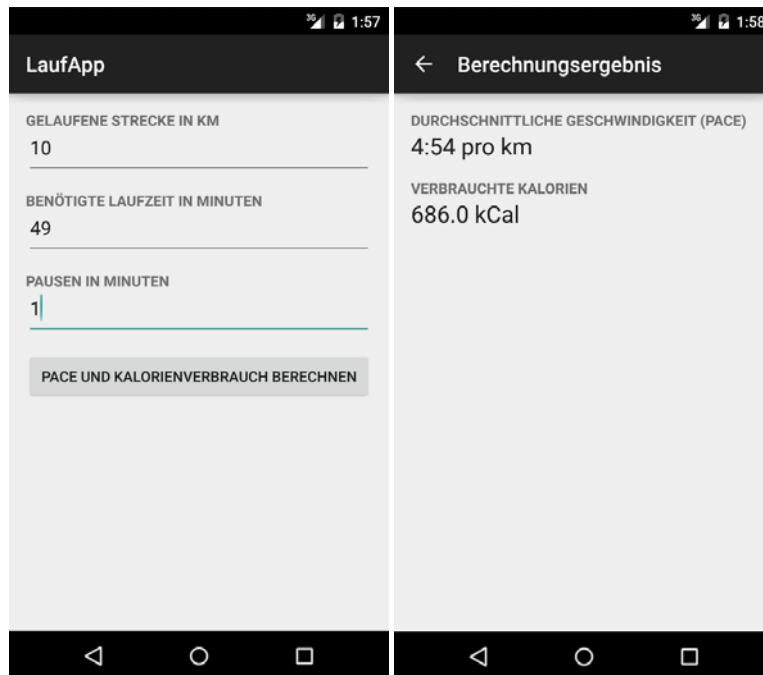


Abbildung 1: So sollte Ihre Anwendung nach Teil 1 aussehen

## 4.2 Teil 2



LaufApp	← Berechnungsergebnis
GELAUFENE STRECKE IN KM 10	DURCHSCHNITTliche GESCHWINDIGKEIT (PACE) 4:54 pro km
BENÖTIGTE LAUFZEIT IN MINUTEN 49	VERBRAUCHTE KALORIEN 686.0 kCal
PAUSEN IN MINUTEN 1	
PACE UND KALORIENVERBRAUCH BERECHNEN	

Abbildung 2: So sollte Ihre Anwendung nach Teil 2 aussehen