

Template

Zuletzt bearbeitet von Alexander Bazo

Übungsblatt 12

Wichtige Informationen zur Bearbeitung der Aufgabe

- Informationen zur Entwicklungsumgebung *IntelliJ IDEA*
- Informationen zum Im- und Export von Projekten

Starterpaket

Ein vorbereitetes Starterpaket zur selbständigen Implementierung der Aufgabe finden Sie hier.

Flight Planner

Ihre Aufgabe ist es, eine KonsolenApplikation zu schreiben, welche aus einer Datei Flugziele von verschiedenen Städten ausliest und dem Nutzer erlaubt, einen Round-Trip mit dem Flugzeug zu planen.

Die Konversation soll nach folgendem Schema aufgebaut sein:

```
Let's plan a round trip:
Here is a list of all the destinations in our database:
San Jose
New York
Anchorage
Honolulu
Denver
San Francisco
Which city do you want to start in:
Honolulu
Where do you want to go from Honolulu?
New York
San Francisco
San Francisco
Where do you want to go from San Francisco?
New York
Denver
Honolulu
Honolulu
```



```
You completed your trip!  
Your trips spanned the following destinations:  
Honolulu -> San Francisco -> Honolulu
```

Hinweis: Eine besondere Schwierigkeit in dieser Aufgabe ist, angemessene Datenstrukturen zu modellieren, um sich die Informationen, die für den weiteren Ablauf benötigt werden, abzuspeichern. Man muss sich sowohl die möglichen Flüge aus der Datei `flights.txt` merken, als auch die Route, die der Nutzer schon zurückgelegt hat. Überlegen Sie hier, wann eine `ArrayList` und wann eine `HashMap` sinnvoll zu benutzen sind.

Die Flugdaten stehen in einer Datei `data/flights.txt` im Starterpaket zur Verfügung. Diese Datei ist wie folgt formatiert:

- Jede Zeile besteht aus einem Schlüssel-Wert-Paar getrennt durch `->`
- Die zugehörigen Werte sind durch `,` getrennt
- Die Datei soll der Übersicht halber Leerzeilen enthalten können (die beim Auslesen zu ignorieren sind)

Die vorgegebene Datei enthält den unten stehenden Inhalt. Ihr Programm sollte aber auch mit anderen Städte-Kombinationen funktionieren, die in der selben Weise formatiert sind.

```
San Jose -> San Francisco, Anchorage  
New York -> Anchorage, San Jose, San Francisco, Honolulu  
Anchorage -> New York, San Jose  
Honolulu -> New York, San Francisco  
Denver -> San Jose  
San Francisco -> New York, Denver, Honolulu
```

Um die eingelesenen Zeilen in *Teilstrings* zu unterteilen können sie die Methode `String.split()` nutzen. Übergeben sie der Methode das/die Zeichen bei dem sie den String trennen wollen. Das Ergebnis ist ein `Array` welches alle 'Teilstrings' beinhaltet.

Achten sie darauf in ihrem Programm überflüssige Leerzeichen zu entfernen. Hierzu können sie die Methode `String.trim()` nutzen.

Der Code zum einlesen der Datei und der einzelnen Zeilen ist bereits gegeben. Dazu wird beim Programmstart die statische Methode `readLinesFromFile` aufgerufen. Deren Rückgabe stellt ein `Array` dar, in dem jede Zeile der Datei als einzelner String vorliegt. Zerlegen sie den String der in jeder Zeile ausgelesen wird sinnvoll in `Start` und `Ziele`. Überlegen sie in welchen Datenstrukturen `Start` und `Ziele` abgebildet werden können. Finden sie außerdem eine sinnvolle Möglichkeit `Start` und `Ziele` miteinander zu verknüpfen.

Um eine Nutzereingabe zu lesen kann die Klasse `java.util.Scanner` verwendet werden. Eine darauf basierende Methode `readLine()` welche eine Zeile ausliest und den Inhalt als String zurückgibt ist bereits vorgegeben. Denken sie daran falsche Eingaben (z.B. Tippfehler in Städtenamen) abzufangen.

Ihr Programm soll folgende Anforderungen erfüllen:

- Die ausgelesenen Zeilen der Datei `flights.txt` sinnvoll aufbereiten und in einer geeigneten Datenstruktur abspeichern.
- Die komplette Städteliste anzeigen.
- Den Benutzer eine Stadt auswählen lassen, von der aus die Rundreise beginnt.



- In einer Schleife alle Städte anzeigen, die von der aktuellen Stadt aus direkt erreicht werden können und den Nutzer nach der nächsten Stadt fragen.
- Sobald der Nutzer wieder zur Anfangsstadt kommt, bricht die Schleife ab und das Ergebnis der Flugroute wird ausgegeben.