

30.5.2017

Übungsblatt 8

Aufgabe 1 (Streams und Java 8 – 3 Punkte)

Gegeben sei eine Liste von Spielern eines von Ihnen zu bestimmenden Spiels, die in der Datei `playersOLAT.txt` tabulatorgetrennt wie folgt gespeichert sind:

```
id Vorname Nachname Spielstärke
```

- (1 Punkt) Erstellen Sie eine Klasse `PlayerManager` und definieren Sie darin eine Methode `List<Player> readPlayers()`, die das File einliest und eine Liste aller `Player` zurückgibt. Verwenden Sie dazu einen `java.util.Scanner` und dessen Methode `next()`, um auf die einzelnen Attribute zuzugreifen. Behandeln Sie etwaige Fehler auf sinnvolle Weise.
- (1 Punkt) Erstellen Sie zwei Varianten einer Methode, die die Spielstärke aller Spieler um 1 erhöht:
 - Erstellen Sie die Methode `increaseStrength7()`, in der Sie keine Java 8 Konstrukte verwenden.
 - Erstellen Sie die Methode `increaseStrength8()`, in der Sie `foreach()` aus Java 8 verwenden.
 - Überprüfen Sie die Richtigkeit Ihrer Methoden mit 2 Tests (die Spielstärke ist ohne Erhöhung immer zwischen 1 und 3, d.h. es reicht zu überprüfen, ob diese nach der Erhöhung zwischen 2 und 4 liegt).
- (1 Punkt) Erstellen Sie zwei Varianten einer Methode, die die Anzahl der Spieler je Spielstärke berechnet und eine `Map<Integer, Long>` zurückgibt (Spielstärke → Anzahl der Spieler):
 - Erstellen Sie die Methode `groupByStrength7()`, in der Sie keine Java 8 Konstrukte verwenden.
 - Erstellen Sie die Methode `groupByStrength8()`, in der Sie u.a. `Collectors.groupingBy()` aus Java 8 verwenden.
 - Überprüfen Sie in einem Test, ob die zurückgegebenen Maps identisch sind.

Aufgabe 2 (The Joke Generator – 7 Punkte)

In dieser Aufgabe soll ein Programm entwickelt werden, das zufällige Witze nach gewissen Kriterien aus einer gegebenen Internetquelle lädt und am Bildschirm anzeigt. Als Quelle wird dabei die *Chuck Norris Database* (<http://www.icndb.com/api/>) verwendet. Gehen Sie dabei wie folgt vor:

- (1 Punkt) Erstellen Sie eine Klasse `URLJoke` und erzeugen Sie darin eine Hilfsmethode `String readURL(String urlString)`, die eine Anfrage an die gegebene String-Repräsentation der URL sendet und das Ergebnis als `String` zurückliefert. Verwenden Sie dafür die Methode `openStream()` der Klasse `java.net.URL` in Kombination mit entsprechenden Readern. Reichen Sie potentiell auftretende Exceptions weiter.

- b) (2 Punkte) Testen Sie Ihre Methode temporär (z.B. in einer main-Methode) mit der Adresse `http://api.icndb.com/jokes/random` und sehen Sie sich den Output an. Erstellen Sie nun eine Methode `String loadJokeFromURL(String urlString)`, die den Inhalt einer gegebenen Joke-URL liest und den gelieferten Witz parst und zurückgibt.

Das Ergebnis einer Anfrage wird im JSON-Format zurückgeliefert - ein sehr häufig eingesetztes Mittel, um Daten im Web auszutauschen. Entsprechend gibt es viele Parser, die aus JSON-Text Java-Objekte erzeugen. Für unsere Zwecke reicht es vorerst, wenn Sie den Witz mittels einer Regular Expression lokalisieren und extrahieren. Verwenden Sie dafür folgendes Muster (`java.util.regex.Pattern`):

```
"\"joke\": \"([^\"]*)\""
```

Kompilieren Sie dieses Pattern (`compile()`), erzeugen Sie einen `java.util.regex.Matcher` (`matcher()`) und verwenden Sie dessen Methoden `find()` und `group()` zum Durchsuchen. Fangen Sie sämtliche Fehler in dieser Methode und werfen Sie eine eigene `NoJokeFoundException` mit einer entsprechenden Fehlermeldung weiter, sollte ein Fehler passieren.

- c) (0,5 Punkte) Erstellen Sie die Methode `String randomJoke()`, die einen zufälligen Witz von der URL `http://api.icndb.com/jokes/random` zurückgibt. Reichen Sie Exceptions weiter.
- d) (1 Punkt) Erstellen Sie die Methode `String jokeWithNumber(int number)`, die den Witz mit der gegebenen Nummer von der URL `http://api.icndb.com/jokes/{number}` zurückgibt¹. Laden Sie vorher an einer geeigneten Stelle einmalig die maximale Zahl der verfügbaren Witze über die URL `http://api.icndb.com/jokes/count` und parsen Sie die Zahl wieder mit einer entsprechenden Regex. Reichen Sie Exceptions weiter bzw. werfen Sie eine neue `NoJokeFoundException` mit einer entsprechenden Meldung, wenn die übergebene Nummer größer als die maximale Anzahl an Witzen ist.
- e) (1 Punkt) Die API bietet die Möglichkeit, zwischen (aktuell) 2 Kategorien von Witzen zu unterscheiden: `explicit` und `nerdy`. Erstellen Sie dafür ein Enum, das sowohl den Namen des Wertes als auch die ID speichert (Sie benötigen dafür eine Membervariable sowie einen eigenen Konstruktor innerhalb des Enums). Beispielsweise würde der Wert für die `nerdy`-Kategorie wie folgt erstellt werden:

```
enum JokeCategory {
    JokeCategoryNerdy("nerdy"), ...
    ...
}
```

Erstellen Sie nun die Methode `String jokeWithCategory(JokeCategory category)`, die einen zufälligen Witz der übergebenen Kategorie von der URL `http://api.icndb.com/jokes/random?limitTo={category}` zurückgibt². Reichen Sie Exceptions weiter.

- f) (1 Punkt) Schlussendlich bietet die API die Möglichkeit, den Namen *Chuck Norris* durch einen selbst definierten Namen zu ersetzen. Dafür wird für jede der vorigen Abfragen folgendes an die URL angehängt: `?firstName=Vorname&lastName=Nachname`, z.B.: `http://api.icndb.com/jokes/478?firstName=Daisy&lastName=Duck`. Im Falle der Kategorie wird das `'?'` durch ein `'&'` ersetzt: `http://api.icndb.com/jokes/random?limitTo=[nerdy]&firstName=Daisy&lastName=Duck`.

Erstellen Sie nun die entsprechenden Methoden:

- `String randomJoke(String firstname, String lastname)`
- `String jokeWithNumber(int jokeNumber, String firstname, String lastname)`

¹z.B. `http://api.icndb.com/jokes/478`

²z.B. `http://api.icndb.com/jokes/random?limitTo=[nerdy]`

- `String jokeWithCategory(JokeCategory category, String firstname, String lastname)`

Verwenden Sie so weit wie möglich die bestehenden Programmteile.

- g) (0,5 Punkte) Erstellen Sie eine `main`-Methode und geben Sie einen zufälligen Witz, einen Witz nach Nummer und einen Witz der Kategorie `nerdy` jeweils mit und ohne spezielle Namen aus.

Wichtig: Laden Sie bitte Ihre Lösung (.txt, .java oder .pdf) in OLAT hoch und geben Sie mittels der Ankreuzliste auch unbedingt an, welche Aufgaben Sie gelöst haben. Die Deadline dafür läuft am Vortag des Proseminars um 23:59 (Mitternacht) ab.