

Labor Datenströme - Streams

In einer Textdatei stehen zeilenweise textuell geschriebene Integerwerte, Doublewerte oder die Strings `true` bzw. `false` oder beliebige andere Texte.

Beispiel: für eine Textdatei:

```
17
13.8
true
123
stuff
false
12345
```

Schreiben Sie ein Programm, das die Datei zeilenweise liest und dass eine Binärdatei nach folgenden Regeln erzeugt:

(Verwenden Sie nur LowLevel- bzw. Filter-Klassen des IO-Stream-API's)

1. Zunächst wird versucht, die Zeile in einen `Integer` zu parsen. Bei Erfolg, wird in die Binärdatei zunächst ein Zeichen `'i'`, dann der Integerwert binär geschrieben.
2. Kann der Text nicht in einen `Integer` geparkt werden, so wird versucht, aus der Zeile einen `Double` zu erzeugen. Bei Erfolg werden das Zeichen `'d'` und der binäre Doublewert in die Datei geschrieben.
3. Kann die Zeile auch nicht in einen `Double` geparkt werden, so wird überprüft, ob die Zeile `true` bzw. `false` lautet. In diesem Fall wird das Zeichen `'t'` bzw. `'f'` in die Datei geschrieben.
4. Alle anderen Zeilen werden übersprungen.

Obige Textdatei sollte also strukturell die folgende Binärdatei erzeugen:
(wobei die Zahlenwerte binär geschrieben sind)

```
i17d13.8ti123fi12345
```

Schreiben Sie auch eine zweite Methode, welche die Binärdatei liest und die eine `Map` zurückliefert, in welcher als Schlüssel die Texte `Integer`, `Double` und `Boolean` stehen, sowie als zugehörige Werte eine Liste aller entsprechenden Daten dieses Typs in gelesener Reihenfolge. Außerdem sollte diese Methode in eine übergebene typsichere Liste die Typen der gelesenen Daten mittels der Characters `i`, `d`, `t`, `f` in ihrer gelesenen Reihenfolge mitprotokollieren.

Mit der erstellten `Map` und der Liste mit der Typ-Lesereihenfolge ist nun eine neue Textdatei zu erstellen, welche gleiche Form wie die Ausgangsdatei besitzt, jedoch ohne jene Zeilen in welchen nur "Sinnlos"-Text stand.