Grundlagen der Programmkonstruktion
Übungsblatt 6

1 Dateien, Validierung von Dateien (3)

In dieser Aufgabe müssen Sie eine Datei behandeln, die Daten von Studierenden enthält. Das Format der einzulesenden Datei sieht folgendermaßen aus:

```
Max Mustermann 0325532 15
Julia Musterfrau 1106214 43
Gustav Gans 9944625 88
```

Eine Zeile ist folgendermaßen aufgebaut: Der erste String ist der Vorname, der zweite String ist der Nachname, der dritte String ist die Matrikelnummer und der vierte String ist die Punktezahl (von 0 bis 100 inklusive). Sonderfälle wie mehrteilige Namen oder Doppelnamen müssen Sie nicht berücksichtigen.

1.1 Eingabe (1)

Schreiben Sie eine Methode, die die Datei einliest. Am Ende der Methode geben Sie aus, wieviele Zeilen eingelesen wurden.

1.2 Validieren (1)

Testen Sie, ob das Eingabeformat korrekt ist, d.h. ob die Felder das enthalten, was sie enthalten sollen. Die folgenden Zeilen enthalten typische Fehler, die Sie erkennen sollten:

```
MaxMustermann 05532 63
Gustav Gans 994462512
0023122 Donald Duck 11
Dagobert Duck 6625542 434
TestEintrag TestEintrag 0 -1
```

Bei einem falschen Eintrag soll das Programm ausgeben, in welcher Zeile der Fehler aufgetreten ist. Am Ende soll ausgeben werden, wieviele Zeilen eingelesen wurden und wieviele davon fehlerhaft waren. Fehlerhafte Zeilen sollen im Folgenden ignoriert werden.

1.3 Ausgabe (1)

Schreiben Sie in eine Datei die Daten aller Personen, die weniger als 50 Punkte erreicht haben. Dabei sollen Matrikelnummer, Nachname, Vorname in dieser Reihenfolge durch Beistriche getrennt ausgegeben werden.

Die neue Datei soll in etwa folgendermaßen aussehen:

```
0325532,Mustermann,Max
1106214,Musterfrau,Julia
```

2 Bruchzahlen (2)

Implementieren Sie eine einfache Klasse `Fraction`. Diese Klasse soll Bruchzahlen darstellen können (also Zahlen wie $\frac{3}{4}$). Eine Bruchzahl besteht aus Zähler und Nenner ($\frac{Zähler}{Nenner}$).

Zähler und Nenner sollen so weit wie möglich gekürzt sein. Bei negativen Zahlen soll *nur* der Zähler negativ sein.

2.1 Konstruktor (0.5)

Schreiben Sie einen Konstruktor `Fraction(int zaehler, int nenner)` für die Klasse. Beachten Sie Sonderfälle wie Null-Werte und überlegen Sie sich geeignete Lösungen für diese.

2.2 void add(Fraction f) (0.5)

Schreiben Sie eine Methode `add(Fraction f)`, die den Wert von `f` zu `this` addiert. Beachten Sie, dass die Bruchzahl auch nach Aufruf von `add` gekürzt sein müssen und nur der Zähler negativ sein darf.

2.3 equals und toString (0.5)

Schreiben Sie die Methoden `equals` und `toString`.

2.4 Interface Comparable (0.5)

Implementieren Sie das Interface `java.lang.Comparable<Fraction>`. Die Dokumentation dazu finden Sie unter `http://docs.oracle.com/javase/6/docs/api/java/lang/Comparable.html`.