Prof. Dr. Stefan Brass
Dipl.-Inform. Annett Thüring

Martin-Luther-Universität Halle-Wittenberg
Institut für Informatik
Wintersemester 2018/19

Objektorientierte Programmierung: Hausaufgabenblatt 4

Abgabe: 19.11.2018, 11:00

Hausaufgabe 1: (6 Punkte)

Bitte beachten Sie: die Einsendung der Aufgabe erfolgt nur im YAPEX.
Open exercises via code → Freigabecode: `5coo815ohp53-c65b` (Statistische Kennzahlen)

Schreiben Sie ein Programm, eine Reihe reeller Zahlen $x_1, \ldots, x_n \in \mathbb{R}$ einliest, und dafür einige statistische Kennzahlen berechnet und ausgibt. Im Detail sollen Sie die folgenden Kennzahlen berechnen:

- Anzahl der eingelesenen Zahlen: $n \in \mathbb{N}$
- Summe: $s = \sum_{i=1}^{n} x_i$
- Mittelwert (Durchschnitt): $d = \frac{s}{n}$
- Maximum: $x_{max} = \max(\{x_1, \ldots, x_n\})$
- Minimum: $x_{min} = \min(\{x_1, \ldots, x_n\})$
- zweitkleinste Zahl (Sie können davon ausgehen, dass keine Zahl doppelt vorkommt): $\min(\{x_1, \ldots, x_n\} - \{x_{min}\})$

Ihr Programm soll eine beliebige Anzahl an reellen Zahlen einlesen können. Verwenden Sie dafür die bekannte `Scanner` Klasse. Um zu überprüfen, ob Sie noch eine weitere Zahl einlesen müssen, können Sie die Funktion `hasNextDouble()` ihres Scanner-Objekts verwenden:

```
Scanner input = new Scanner(System.in);
input.hasNextDouble() // wahr, wenn eine weitere Zahl vorhanden ist
                      // falsch, wenn keine weitere Zahl eingelesen werden kann
```

Beachten Sie, dass `hasNextDouble()` ggf. auf eine Eingabe wartet, wenn Sie Ihr Programm auf der Kommandozeile ausprobieren. Sie können irgendein Zeichen eingeben, das keine Gleitkommazahl beginnen kann, z.B. `x` (unter Linux ist `Ctrl+D` das Dateiende).

Eingabe: `1,0 2,0 3,0 4,0 5,0`
Ausgabe:

```
Anzahl: 5
Summe: 15.0
Mittelwert: 3.0
Maximum: 5.0
Minimum: 1.0
Zweitkleinste Zahl: 2.0
```


Hausaufgabe 2: (4 Punkte)

Bitte beachten Sie: die Einsendung der Aufgabe erfolgt nur im YAPEX.
Open exercises via code → Freigabecode: `5coo815p01hp-4332` (Multiplikationstabelle)

Schreiben Sie ein Programm, welches eine Multiplikationstabelle für ein gegebenes `n` zwischen 1 und 9 ausgibt. Die Tabelle für `n` hat immer die Größe `n+1`. In der ersten Zeile sowie der ersten Spalte stehen die Zahlen von 1 bis n. In den übrigen Einträgen der Tabelle steht das Ergebnis der Multiplikation der Zahl dieser Spalte/Zeile. Zusätzlich soll im Feld oben links das Multiplikationszeichen `*` ausgegeben werden.

Beispiel n=5

```
*  1  2  3  4  5
1  1  2  3  4  5
2  2  4  6  8 10
3  3  6  9 12 15
4  4  8 12 16 20
5  5 10 15 20 25
```

Eingabe

Ihr Programm soll die Zahl `n` einlesen. Verwenden Sie dafür die bekannte Klasse 'Scanner'. überprüfen Sie außerdem die Eingabe auf den korrekten Wertebereich von 1 bis 9. Sie dürfen davon ausgehen, dass immer eine Ganzzahl eingegeben wird. Im Falle einer ungültigen Eingabe soll Ihr Programm den folgenden Hinweis ausgeben:

```
Ungueltige Eingabe! Nur Zahlen von 1 bis 9 akzeptiert.
```

Ausgabe

Die einzelnen Spalten der Tabelle sind durch ein Leerzeichen voneinander getrennt. Damit sich die Spalten der Tabelle nicht verschieben, muss vor einstelligen Zahlen jedoch ein zusätzliches Leerzeichen mehr ausgegeben werden.
Tipp: Denken Sie daran, dass Sie einen Zeilenumbruch mit dem Sonderzeichen `\n` auch ohne eine `println`-Anweisung ausgeben können.

Bedingung

Ihr Programm soll die Multiplikationstabelle selbstständig berechnen. Es ist ausdrücklich nicht erlaubt, die Tabelle per Hand für alle 9 Fälle zu schreiben und im Quellcode als String zu speichern.