

Programmieren I

03. Übungsblatt

Aufgabe 16: 2007 wurde die *Internationale Standard-Buchnummer* (ISBN) 13-stellig eingeführt.

Beispielsweise ist 978-3-8274-1631-5 eine gültige 13-stellige ISBN. Die erste Zifferngruppe ist 978 oder 979. Die zweite Gruppe gibt die Sprachgruppe an. Der Ziffer 3 können wir entnehmen, dass das Buch im deutschsprachigen Raum erschienen ist. Die dritte Zifferngruppe 8274 gibt den Verlag an. Die vierte Zifferngruppe, hier 1631, identifiziert das betreffende Buch. Die letzte Ziffer ist eine sogenannte Prüfziffer. Mithilfe der Prüfziffer können z. B. einzelne fehlerhafte Ziffern oder zwei vertauschte Ziffern erkannt werden. Die Prüfziffer wird wie folgt berechnet: Man multipliziert die erste Ziffer mit 1, die zweite mit 3, die dritte mit 1 und abwechselnd weiter mit 3, 1, 3, 1, 3, 1, 3, 1 und 3. Die so gewonnenen Produkte werden addiert. Die Differenz zur nächsten durch 10 teilbaren Zahl ist die Prüfziffer. Für das obige Beispiel ergibt sich die Prüfziffer also wie folgt:

$$105 = 9 + 3 * 7 + 8 + 3 * 3 + 8 + 3 * 2 + 7 + 3 * 4 + 1 + 3 * 6 + 3 + 3 * 1$$
$$5 = 110 - 105$$

Ziel dieser Aufgabe ist es ein Prüf-Programm für ISBN zu schreiben. Im ersten Schritt soll das Tool eine ISBN als Kommandozeilen Parameter erhalten. Im zweiten Schritt soll geprüft werden, ob dies eine gültige 13-stellige ISBN darstellt. Wenn die Prüfziffer fehlerhaft ist, soll die ISBN mit korrekter Prüfziffer ausgegeben werden.

Beispiele:

```
javac Isbn.java
```

```
java Isbn 978-3-8274-1631-5
978-3-8274-1631-5 ist eine gültige ISBN.
```

```
java Isbn 978-3-446-44073-2
978-3-446-44073-2 ist eine fehlerhafte ISBN.
Gültig wäre 978-3-446-44073-9
```

Hinweise: Die Methode `char charAt(int index)` der Klasse `String` liefert das Zeichen, das an der Stelle `index` des Strings ist. Ein Beispiel ist `("Test".charAt(0)=='T')==true`. Durch den Typcast `(int)` erhält man das Zeichen als `int`-Wert. Die Zeichen `'0'`, ..., `'9'` haben die `int`-Werte 48, ..., 57. Wegen der Codierung (s. Vorlesung) besitzen die Zeichen `'0'`, ..., `'9'` nicht die `int`-Werte 0, ..., 9. Das Zeichen `'-'` besitzt den `int`-Wert 45.

Aufgabe 17: Die Wahrscheinlichkeit $p(n)$, dass mindestens zwei Personen aus einer Gruppe von n Menschen am gleichen Tage Geburtstag haben, beträgt

$$p(n) = 1 - \frac{365}{365} \cdot \frac{364}{365} \cdot \frac{363}{365} \cdot \dots \cdot \frac{365 - n + 1}{365}.$$

Schreiben Sie ein Java-Programm, um die Wahrscheinlichkeiten $p(2), \dots, p(60)$ zu berechnen und auszugeben. Wie viele Personen müssen der Gruppe mindestens angehören, damit die gesuchte Wahrscheinlichkeit größer als 50 % ist?

Aufgabe 18: Die Dezimalziffern einer natürlichen Zahl n , $n \geq 1$, werden einzeln quadriert und addiert. Anschließend wird mit der entstandenen Summe genauso verfahren. Die Ausgangszahl n ist *fröhlich*, wenn man bei diesem Vorgehen schließlich auf die Zahl 1 stößt, ansonsten ist sie *traurig*. Beispielsweise ist 7 eine *fröhliche* Zahl:

$$\begin{array}{lll} 7: & 7^2 & = 49 \\ 49: & 4^2 + 9^2 & = 97 \\ 97: & 9^2 + 7^2 & = 130 \\ 130: & 1^2 + 3^2 + 0^2 & = 10 \\ 10: & 1^2 + 0^2 & = 1 \end{array}$$

Schreiben Sie ein Java-Programm, das alle fröhlichen Zahlen zwischen einer Unter- und einer Obergrenze berechnet und die zugehörigen Folgen ausgibt. Der Dialog soll folgendermaßen ablaufen:

```

1  Bitte geben Sie die untere Grenze ein: 5
2  Bitte geben Sie die obere Grenze ein: 30
3  7 -> 49 -> 97 -> 130 -> 10 -> 1
4  10 -> 1
5  13 -> 10 -> 1
6  19 -> 82 -> 68 -> 100 -> 1
7  23 -> 13 -> 10 -> 1
8  28 -> 68 -> 100 -> 1

```

Hinweis: Für traurige Zahlen führt die Folge schließlich auf die Zahl 4.

Pflichtaufgabe 19: In dieser Aufgabe soll das bekannte Spiel Tic-Tac-Toe⁷ implementiert werden. Es müssen aber zwei kleinere Anpassungen durchgeführt werden:

- Das Spielfeld ist nicht 3x3 sondern 5x5
- Das Spiel wird gewonnen, wenn ein Spieler eine Reihe von 5 Feldern belegt hat und nicht 3

Ziel ist es, dass zwei Spieler gegeneinander spielen können. Eine KI, um gegen den Computer spielen zu können, ist *nicht* gefordert!

Benötigte Features:

- Einlesen von Koordinaten über die Kommandozeile.
- Programmieren ein Programm, dass das Tic-Tac-Toe Spiel umsetzt. Dieses soll:
 - Ein Spielfeld verwalten.
 - Es erlauben das aktuelle Spielfeld auszugeben. Die Ausgabe soll ähnlich formatiert werden wie in Abbildung 4.
 - Den Spielverlauf verwalten. Konkret ermittelt das Programm, welcher Spieler aktuell an der Reihe ist, oder ob das Spiel abgeschlossen wurde (durch Unentschieden oder Sieg eines Spielers).
 - Den Spieler mit entsprechenden Anweisungen und Ausgaben durch den Spielverlauf führen. Ein Beispiel hierfür ist in Abbildung 5 gegeben.

Hinweis: Zur Eingabe der Koordinaten bietet sich die Verwendung der **Scanner** Klasse, welche auf dem vorherigen Aufgabenblatt vorgestellt wurde, an.

Für die Umsetzung von jedem der obigen Teilaspekte erhalten Sie jeweils einen Punkt. Haben Sie mindestens einen der Teilaspekte korrekt bearbeitet, können Sie einen weiteren Punkt für Einhaltung der Programmierrichtlinien erhalten. Diese sind:

- Formatierung des Programms gemäß der Formatierungsrichtlinien.
- Javadoc Kommentare für jede Klasse, jedes Interface und jede public Methode/-Funktion.
- Keine überflüssigen Dateien (hier primär `.class` und `.jar` Dateien, aber auch Sonstiges, das nicht unter Quelltext oder Dokumentation fällt) im Git.

Für das Lösen dieser Pflichtaufgabe können Sie maximal 6 Punkte erhalten.

Berücksichtigt werden ausschließlich Ergebnisse, die bis zum 19.12.2019 um 23:59 in das Git Repository Ihrer Gruppe gepusht wurden.

⁷Für eine genauere Erklärung des Spiels siehe die Einleitung von <https://en.wikipedia.org/wiki/Tic-tac-toe>.

```
| | | | |
-----

Bitte die Koordinate (Reihe und Spalte, jeweils 0-4) eingeben:
4
4
Jetzt am Zug: Spieler 2

-----
|X| | | |
-----
| | | | |
-----
| | |O| |
-----
| | | | |
-----
| | | |X|
-----

Bitte die Koordinate (Reihe und Spalte, jeweils 0-4) eingeben:
|
```

Abbildung 4: Ausgabe des Tic-Tac-Toe Spielfelds mit 3 Feldern.

Um Ihre Lösung bewertet zu bekommen, müssen Sie diese in der Woche vom 06.01.2020 - 10.01.2020 dem Tutor/der Tutorin in Ihrer Übungsgruppe vorstellen. Desweiteren muss die abgegebene Lösung compilieren und ein (teil-)funktionstüchtiges Programm darstellen um Teilpunkte zu bekommen. Ihre Lösung muss in den bereits für dieses Übungsblatt angelegten Ordner (Blatt03) im Git Repository gespeichert werden, damit diese gewertet wird.

```

1 > java -jar tictactoe.jar
2 Jetzt am Zug: Spieler 1
3 -----
4 | | | | |
5 -----
6 | | | | |
7 -----
8 | | | | |
9 -----
10 | | | | |
11 -----
12 | | | | |
13 -----
14
15 Bitte die Koordinate (Reihe und Spalte, jeweils 0-4) eingeben:
16 1
17 1
18 Jetzt am Zug: Spieler 2
19 -----
20 | | | | |
21 -----
22 | |X| | |
23 -----
24 | | | | |
25 -----
26 | | | | |
27 -----
28 | | | | |
29 -----
30
31 Bitte die Koordinate (Reihe und Spalte, jeweils 0-4) eingeben:
32 2
33 2
34 Jetzt am Zug: Spieler 1
35 -----
36 | | | | |
37 -----
38 | |X| | |
39 -----
40 | | |O| |
41 -----
42 | | | | |
43 -----
44 | | | | |
45 -----
46
47 Bitte die Koordinate (Reihe und Spalte, jeweils 0-4) eingeben:

```

Abbildung 5: Beispiel für den Spielverlauf.