

## Algorithmen und Datenstrukturen (Studiengang Wirtschaftsinformatik 3. Semester)

### Aufgabenstellung für das Praktikum am 14.12.2022

#### Aufgabe 30:

Erstellen Sie ein Java-Programm, das mittels des in der Vorlesung vorgestellten Lösungsschemas *Generate & Test* eine Lösung für das folgende *kryptoarithmetische Rätsel* ermittelt:

$$\begin{array}{r} \text{TWO} \\ + \text{TWO} \\ \hline \text{FOUR} \end{array}$$

Eine Lösung dieses Rätsels besteht darin, jeden Buchstaben durch eine andere Ziffer von 1 bis 9 (also ohne Nullen) zu ersetzen, so dass die Gleichung  $\text{TWO} + \text{TWO} = \text{FOUR}$  aufgeht. Wenn Sie etwa das „T“ durch eine 1, das „W“ durch eine 2, das „O“ durch eine 3, das „F“ durch eine 4, das „U“ durch eine 5 und das „R“ durch eine 6 ersetzen würden, dann wäre dies keine Lösung, da die Gleichung  $123 + 123 = 4356$  nicht aufgeht, also nicht korrekt ist.

Hinweis: Es gibt tatsächlich mehrere Lösungen für dieses Problem; hier konkret sechs verschiedene Kombinationen. Es reicht aus, wenn Sie eine Lösung berechnen und ausgeben!

#### Aufgabe 31:

Erstellen Sie ein Java-Programm, das mittels des in der Vorlesung vorgestellten Lösungsschemas *Backtracking* eine Lösung für das folgende *Sudoku-Rätsel*<sup>1</sup> ermittelt:

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Bei der Entwicklung des Algorithmus können Sie sich von der Implementierung für das Springer-Problem inspirieren lassen, die mittels Backtracking eine vollständige Springer-Tour über ein Schachbrett berechnet (s. aktueller Leseauftrag, S. 23ff).

<sup>1</sup> **Sudoku** (japanisch, kurz für *Sūji wa dokushin ni kagiru*, wörtlich so viel wie „Ziffern dürfen nur einmal vorkommen“) ist ein Typ von Logikrätseln, die aus den lateinischen Quadraten entstand. In der üblichen Version ist es das Ziel, ein 9×9-Gitter mit den Ziffern 1 bis 9 so zu füllen, dass jede Ziffer in jeder *Einheit* (*Spalte*, *Zeile*, *Block* = 3×3-Unterquadrat) genau einmal vorkommt – und in jedem der 81 Felder exakt eine Ziffer vorkommt. Ausgangspunkt ist ein Gitter, in dem bereits mehrere Ziffern vorgegeben sind.

Quelle: wikipedia.de / 6.1.2021



Das o.a. 9x9-Gitter kann in Java als zweidimensionales Feld repräsentiert werden, bei dem die noch offenen Felder mit einer Null, die anderen mit ihrem schon vorgegebenen Wert gefüllt sind:

```
int[][] beispiel_sudoku = { { 5 , 3 , 0 , 0 , 7 , 0 , 0 , 0 , 0 },  
                             { 6 , 0 , 0 , 1 , 9 , 5 , 0 , 0 , 0 },  
                             { 0 , 9 , 8 , 0 , 0 , 0 , 0 , 6 , 0 },  
                             { 8 , 0 , 0 , 0 , 6 , 0 , 0 , 0 , 3 },  
                             { 4 , 0 , 0 , 8 , 0 , 3 , 0 , 0 , 1 },  
                             { 7 , 0 , 0 , 0 , 2 , 0 , 0 , 0 , 6 },  
                             { 0 , 6 , 0 , 0 , 0 , 0 , 2 , 8 , 0 },  
                             { 0 , 0 , 0 , 4 , 1 , 9 , 0 , 0 , 5 },  
                             { 0 , 0 , 0 , 0 , 8 , 0 , 0 , 7 , 9 } };
```

Ihr Programm können Sie anschließend auch mit folgendem Sudoku testen:

```
int[][] noch_ein_sudoku = { { 0 , 0 , 0 , 8 , 0 , 2 , 0 , 0 , 0 },  
                             { 0 , 0 , 0 , 6 , 0 , 5 , 4 , 0 , 8 },  
                             { 0 , 4 , 6 , 0 , 0 , 0 , 2 , 0 , 7 },  
                             { 0 , 0 , 0 , 3 , 1 , 0 , 0 , 0 , 0 },  
                             { 0 , 5 , 8 , 9 , 0 , 0 , 1 , 0 , 0 },  
                             { 0 , 9 , 0 , 5 , 8 , 4 , 0 , 0 , 0 },  
                             { 2 , 1 , 0 , 0 , 5 , 9 , 3 , 0 , 0 },  
                             { 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },  
                             { 0 , 0 , 0 , 7 , 3 , 0 , 0 , 0 , 0 } };
```

Für dieses Rätsel gibt es mehrere Lösungen. Können Sie Ihr Programm so modifizieren, dass alle möglichen Lösungen ausgegeben werden? Wie viele verschiedene Lösungen gibt es?

**Viel Spaß und Erfolg bei den Aufgaben!**