

## Übungsblatt 9

### Aufgabe 1 (Bauer, 3 Punkte)

Ändern Sie die Rätselaufgabe aus dem vorherigen Übungsblatt so ab, dass der Automat ein zwei-dimensionales Feld für die Zustandsübergangstabelle verwendet, statt vieler Fallunterscheidungen. Endzustände können entweder mit einem ein-dimensionalem Feld oder als negative Zustandsco-dierung in der Zustandsübergangstabelle gespeichert werden.

Fügen Sie eine Objektmethode beim Automaten hinzu, bei der alle Flussüberquerungen, in einem ein-dimensionales Feld als Parameter übergeben werden. Das Feld enthält in der Reihenfolge der Überquerungen jeweils eines der drei Dinge bzw. Nichts, das der Bauer bei dieser Überquerung mit nimmt. Die Methode soll die Zustandsübergänge mit jedem der Feldelemente ausführen. Wenn der Bauer am Ende mit allen Dingen am anderen Ufer ist, dann soll diese Methode true zurückgeben und false sonst.

Vergessen Sie nicht diese Methode mit JUnit zu testen.

### Aufgabe 2 (Sudokus überprüfen, 7 Punkte)

Implementieren Sie eine Klasse `Sudoku`, das mit einem ausgefüllte Sudoku gegeben als zwei-dimensionales 9x9 Feld in Konstruktor initialisiert wird und genau eine öffentliche Methode enthält, die true zurück gibt, wenn das Sudoku korrekt ist und false sonst.

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

Beim Sudoku muss in jeder Zeile, jeder Spalte und in jedem 3x3-Block die Zahlen von 1 bis 9 jeweils genau einmal enthalten sein.

Überlegen Sie sich erst, wie sie diese Regeln für eine Zeile überprüfen wollen. Testen Sie diese Idee vor der Implementierung mit den nachfolgenden korrekten und vor allem nicht korrekten Sudokus.

Implementieren Sie dann eine nicht öffentliche Methode, die für ein ein-dimensionales Feld diese Überprüfung durchführt. Versuchen Sie diese Methode wiederzuverwenden, um alle Regeln zu prüfen.

Hier das Sudoku als Java-Feld:

```
{{5, 3, 4, 6, 7, 8, 9, 1, 2},  
 {6, 7, 2, 1, 9, 5, 3, 4, 8},  
 {1, 9, 8, 3, 4, 2, 5, 6, 7},  
 {8, 5, 9, 7, 6, 1, 4, 2, 3},  
 {4, 2, 6, 8, 5, 3, 7, 9, 1},  
 {7, 1, 3, 9, 2, 4, 8, 5, 6},  
 {9, 6, 1, 5, 3, 7, 2, 8, 4},  
 {2, 8, 7, 4, 1, 9, 6, 3, 5},  
 {3, 4, 5, 2, 8, 6, 1, 7, 9}}  
}
```

Überlegen Sie sich auch Testfälle, bei denen falsche Sudokus als falsch erkannt werden. Probieren Sie die Daten von den nächsten Seiten dieses Übungsblattes. Die folgenden Beispiele sind *keine* gültigen Sudokus.

```
{{5, 3, 4, 6, 7, 8, 8, 1, 3},  
 {6, 7, 2, 1, 9, 5, 3, 4, 8},  
 {1, 9, 8, 3, 4, 2, 5, 6, 7},  
 {8, 5, 9, 7, 6, 1, 4, 2, 3},  
 {4, 2, 6, 8, 5, 3, 7, 9, 1},  
 {7, 1, 3, 9, 2, 4, 8, 5, 6},  
 {9, 6, 1, 5, 3, 7, 2, 8, 4},  
 {2, 8, 7, 4, 1, 9, 6, 3, 5},  
 {3, 4, 5, 2, 8, 6, 2, 7, 8}}  
}
```

```
{{5, 3, 7, 6, 4, 8, 9, 1, 2},  
 {6, 7, 2, 1, 9, 5, 3, 4, 8},  
 {1, 9, 4, 3, 8, 2, 5, 6, 7},  
 {8, 5, 9, 7, 6, 1, 4, 2, 3},  
 {4, 2, 6, 8, 5, 3, 7, 9, 1},  
 {7, 1, 3, 9, 2, 4, 8, 5, 6},  
 {9, 6, 8, 5, 3, 7, 2, 1, 4},  
 {2, 8, 1, 4, 7, 9, 6, 3, 5},  
 {3, 4, 5, 2, 8, 6, 1, 7, 9}}  
}
```

```
{5, 5, 5, 5, 5, 5, 5, 5, 5},
{5, 5, 5, 5, 5, 5, 5, 5, 5},
{5, 5, 5, 5, 5, 5, 5, 5, 5},
{5, 5, 5, 5, 5, 5, 5, 5, 5},
{5, 5, 5, 5, 5, 5, 5, 5, 5},
{5, 5, 5, 5, 5, 5, 5, 5, 5},
{5, 5, 5, 5, 5, 5, 5, 5, 5},
{5, 5, 5, 5, 5, 5, 5, 5, 5},
{5, 5, 5, 5, 5, 5, 5, 5, 5},
{5, 5, 5, 5, 5, 5, 5, 5, 5},
}
```

```
{5, 3, 4, 6, 7, 8, 9, 1, 2},
{6, 7, 2, 1, 9, 5, 3, 4, 8},
{8, 5, 9, 7, 6, 1, 4, 2, 3},
{4, 2, 6, 8, 5, 3, 7, 9, 1},
{7, 1, 3, 9, 2, 4, 8, 5, 6},
{9, 6, 1, 5, 3, 7, 2, 8, 4},
{1, 9, 8, 3, 4, 2, 5, 6, 7},
{2, 8, 7, 4, 1, 9, 6, 3, 5},
{3, 4, 5, 2, 8, 6, 1, 7, 9}
}
```

```
{5, 3, 4, 6, 7, 8, 9, 12, 2},
{6, 7, 2, 12, 9, 5, 3, 4, 8},
{12, 9, 8, 3, 4, 2, 5, 6, 7},
{8, 5, 9, 7, 6, 12, 4, 2, 3},
{4, 2, 6, 8, 5, 3, 7, 9, 12},
{7, 12, 3, 9, 2, 4, 8, 5, 6},
{9, 6, 12, 5, 3, 7, 2, 8, 4},
{2, 8, 7, 4, 12, 9, 6, 3, 5},
{3, 4, 5, 2, 8, 6, 12, 7, 9}
}
```