

## Übungsblatt 8: Objektorientierte Programmierung

Ausgabe: 01.12.2017

Abgabe: 10.12.2017

### Aufgabe 1: Sudoku (18 Punkte)

Ihre Aufgabe ist es ein Sudokuspiel zu programmieren. Sie müssen dazu eine Klasse `TestSudoku` und eine Klasse `Sudoku` erstellen. Die Klasse `Sudoku` verwaltet das Spielfeld der Größe  $9 \times 9$  in Form eines zweidimensionalen Feldes vom Typ `byte` und besitzt folgende Funktionen:

- einen Konstruktor, welcher als Parameter den Namen einer einzulesenden Datei erhält und anhand dieser das Spielfeld erstellt. Dies ist Ihnen bereits vorgegeben (siehe weiter unten stehenden Quellcode bzw. bei der Übungsplattform hochgeladene Dateien), so dass Sie lediglich Ergänzungen vornehmen müssen (falls z. B. weitere Attribute initialisiert werden müssen).
- eine `print`-Methode, welche keine Parameter und keinen Rückgabetyt hat, aber das gesamte Spielfeld ausgibt. Für jede Stelle im Spielfeld wird auf der Konsole entweder die Ziffer oder, falls diese nicht bestimmt wurde, ein `*` ausgegeben. Es sollten des Weiteren die einzelnen Quadranten erkennbar sein (durch Nutzung der Zeichen `|` und `-`). Die Zeilen- und Spaltennummerierung ist optional. Eine mögliche Ausgabe ist auf der übernächsten Seite angegeben.
- eine `setze`-Methode, welche als Parameter die Zeile, die Spalte und den neuen Wert erhält. Der Rückgabetyt ist `char`. Hiermit werden mögliche Fehler codiert:

'p' falls die **P**osition des einzufügenden Wertes nicht korrekt war (es gibt nur die Zeilen bzw. Spalten 1 bis 9).

'w' falls der eingegeben **W**ert nicht zwischen 1 und 9 lag.

'z' falls in der **Z**eile, wo der Wert eingefügt werden soll, bereits solch ein Wert vorkommt.

's' falls in der **S**palte, wo der Wert eingefügt werden soll, bereits solch ein Wert vorkommt.

'q' falls in dem **Q**uadranten (dem  $3 \times 3$  Teilfeld) dieser Wert bereits vorkommt.

'f' falls der Wert eingefügt wurde und das gesamte Rätsel gelöst wurde und somit **f**ertig ist.

'l' falls der Wert eingefügt wurde (der Zug **l**egitim war), aber das Rätsel noch nicht gelöst wurde.

Diese Methode muss somit insbesondere diese Fälle überprüfen, ggf. das Spielfeld entsprechend anpassen und den passenden Wert zurückgeben. Es ist explizit erlaubt Hilfsmethoden zu erstellen, die einzelne Überprüfungen vornehmen.

- eine `widerrufen`-Methode, welche keine Parameter und keinen Rückgabetyt hat. Diese Methode soll immer den letzten Zug zurücknehmen. Es kann immer nur der letzte Zug zurückgenommen werden, d. h. falls ein Wert im Spielfeld geändert wurde, kann diese Änderung rückgängig gemacht werden, aber nicht die Änderung davor.

Die Klasse `TestSudoku` enthält die `main`-Methode und verwendet die Klasse `Sudoku` und erzeugt ein neues Objekt (der Konstruktor erhält als Parameter den Wert `"beispiel.txt"`, diese Datei, welche Sie ebenfalls von der Übungsplattform herunterladen können, muss somit insbesondere im gleichen Ordner sein, wo sich auch die Klasse befindet).

Sie soll dem Benutzer ermöglichen Werte im Spielfeld zu setzen. Dazu soll Sie, insofern eine Änderung auftrat, das Spielfeld ausgeben und fragen, ob der letzte Zug zurückgenommen werden soll. Anschließend soll sie den Nutzer fragen, an welcher Steller welcher Wert gesetzt werden soll und die entsprechenden Methoden der Klasse Sudoku aufrufen. Anhand des Fehlercodes, welchen die setze-Methode der Klasse Sudoku liefert, soll sie auch entsprechende Fehlermeldungen erzeugen.

---

Quellcode 1: Klassenvorgabe für Sudoku

---

```
1 import java.io.*;
2 public class Sudoku {
3     private byte[][] spielfeld;
4     // weitere Attribute !
5     public Sudoku(String file) throws IOException {
6         // Initialisierung des Spielfeldes
7         spielfeld = new byte[9][9];
8         // Laden der Werte aus der Datei file und ...
9         BufferedReader in = new BufferedReader(new FileReader(file));
10        String temp;
11        for(int i=0; i<9; i++) {
12            temp = in.readLine();
13            // ... speichern dieser Werte in das Spielfeld
14            for(int j=0; j<9; j++) {
15                if(temp.charAt(j) == '*') spielfeld[i][j] = 0;
16                else {
17                    spielfeld[i][j] = (byte)(temp.charAt(j) - '0');
18                }
19            }
20        }
21        // weitere Initialisierungen !
22    }
23    // weitere Methoden !
24 }
```

---

---

Quellcode 2: Klassenvorgabe für TestSudoku

---

```
1 import java.io.*;
2 public class TestSudoku {
3     public static void main(String[] args) throws IOException {
4         Sudoku test = new Sudoku("beispiel.txt");
5         // ...
6     }
7 }
```

---

```

$ java TestSudoku
123 456 789

1|***|9**|728|
2|278|**3|*1*|
3|*9*|***|64*|

4|*5*|*6*|2**|
5|**6|***|3**|
6|*1*|*5*|***|

7|1**|7*6|*34|
8|***|5*4|***|
9|7*9|1**|8*5|

Zeile: 12
Spalte: 7
Wert: 5
Als Zeilen und Spalten
sind nur Werte
zwischen 1 und 9
zulaessig!
Zeile: 2
Spalte: 7
Wert: 50
Der Wert muss zwischen
1 und 9 liegen!
Zeile: 2
Spalte: 7
Wert: 5
123 456 789

1|***|9**|728|
2|278|**3|51*|
3|*9*|***|64*|

4|*5*|*6*|2**|
5|**6|***|3**|
6|*1*|*5*|***|

7|1**|7*6|*34|
8|***|5*4|***|
9|7*9|1**|8*5|

Letzten Zug widerrufen?
(j fuer ja , n fuer
nein) n
Zeile: 3
Spalte: 9
Wert: 1
Der Quadrant enthaelt
bereits eine 1!
Zeile: 2
Spalte: 5
Wert: 1
In dieser Zeile kommt
bereits eine 1 vor!

```

```

Zeile: 1
Spalte: 1
Wert: 1
In dieser Spalte kommt
bereits eine 1 vor!
Zeile: 5
Spalte: 8
Wert: 5
123 456 789

1|***|9**|728|
2|278|**3|51*|
3|*9*|***|64*|

4|*5*|*6*|2**|
5|**6|***|35*|
6|*1*|*5*|***|

7|1**|7*6|*34|
8|***|5*4|***|
9|7*9|1**|8*5|

Letzten Zug widerrufen?
(j fuer ja , n fuer
nein) n
Zeile: 3
Spalte: 3
Wert: 1
123 456 789

1|***|9**|728|
2|278|**3|51*|
3|*91|***|64*|

4|*5*|*6*|2**|
5|**6|***|35*|
6|*1*|*5*|***|

7|1**|7*6|*34|
8|***|5*4|***|
9|7*9|1**|8*5|

Letzten Zug widerrufen?
(j fuer ja , n fuer
nein) j
123 456 789

1|***|9**|728|
2|278|**3|51*|
3|*9*|***|64*|

4|*5*|*6*|2**|
5|**6|***|35*|
6|*1*|*5*|***|

7|1**|7*6|*34|
8|***|5*4|***|
9|7*9|1**|8*5|

```

weitere Ein- und Ausgaben ...

```

123 456 789

1|634|915|728|
2|278|643|519|
3|591|278|643|

4|457|369|281|
5|986|421|357|
6|312|857|496|

7|125|786|934|
8|863|594|172|
9|749|132|8*5|

Letzten Zug widerrufen?
(j fuer ja , n fuer
nein) n
Zeile: 9
Spalte: 8
Wert: 6
123 456 789

1|634|915|728|
2|278|643|519|
3|591|278|643|

4|457|369|281|
5|986|421|357|
6|312|857|496|

7|125|786|934|
8|863|594|172|
9|749|132|865|

Gratuliere , Sie haben
das Raetsel geloest!

```