



## Übungsblatt 7: Arrays, Rekursion

Ausgabe: 18.11.19

Abgabe: 27.11.19 10:00 Uhr elektronisch mittels Git

### Aufgabe 1

Sudoku wird in einem 9x9-Kästchenfeld gespielt. Die Ziffern 1 bis 9 dürfen in jedem Quadranten sowie in jeder senkrechten und in jeder waagrechten Reihe nur einmal vorkommen. Beispiel-Aufgabenstellung:

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

Entwickeln Sie eine Klasse `Sudoku` wie folgt:

- Erstellen Sie eine Methode `static void print(int[] [] field)`, die ein zweidimensionales `int`-Array als Parameter bekommt und dessen Inhalt als 9x9-Spielfeld übersichtlich auf die Konsole schreibt.
- Erstellen Sie eine Methode `static boolean isCorrect(int[] [] field)`, die für das übergebene Array (9x9) bestimmt, ob alle Einträge nach den Sudoku-Regeln korrekt sind. Falls ja, liefert die Methode `true` als Ergebnis, sonst `false`.
- Erstellen Sie eine Methode `static void solve(int[] [] field, int row, int col)`, die versucht, rekursiv eine Lösung für das Sudoku zu finden. Existiert eine Lösung, wird sie mittels `print` ausgegeben. Danach geht die Suche nach weiteren Lösungen weiter.
- Erstellen Sie eine `main`-Methode, die ein 9x9-Sudoku-Spielfeld anlegt, wobei alle fehlenden Einträge eine 0 enthalten sollen. Mit diesem Feld wird dann `solve` aufgerufen.

Hilfestellungen:

- Das folgende Codefragment (an der passenden Stelle eingefügt) gibt das aktuelle Spielfeld auf der Konsole aus und wartet dann eine Sekunde. So können Sie den Ablauf Ihres Programms verfolgen.

```
print(field);
try {
    Thread.sleep(1000);
} catch (InterruptedException e) {}
```

- Die Methode `solve` übernimmt die eigentlich Arbeit. Vergessen Sie nicht, bereits überprüfte Werte wieder zu "löschen".
- Überlegen Sie, ob Ihnen weitere Hilfsmethoden das Lösen der Aufgabe erleichtern!
- Alle Ihre Methoden können sich darauf verlassen, dass nur mit 9x9-Feldern gearbeitet wird.

## Allgemeines

- Legen Sie für die Bearbeitung dieses Übungsblattes ein Paket namens `uebung07` an, in dem Sie Ihre Klassen anlegen.
- Sie können sich darauf verlassen, dass nur ganze Zahlen eingegeben werden (keine Buchstaben oder Sonderzeichen).
- In Moodle finden Sie Testklassen `...1stTest`, mit der Sie prüfen können, ob Ihre Methoden korrekt definiert sind und zumindest die einfachsten Eingaben korrekt behandelt werden.
- Erlaubt sind `MakeItSimple`-Funktionen (keine nicht besprochene Funktionalität aus der Java-Standard-Bibliothek) und das bisher erworbene Wissen aus den PR1-Vorlesungen. Sie müssen alle(!) von Ihnen verwendeten Konstrukte der Sprache sowie alle verwendeten Methoden, die nicht aus der Hilfsbibliothek `MakeItSimple` stammen, gut erklären und nötigenfalls im Testat selbst programmieren können!
- Sie geben ab, indem Sie vor Ende der Abgabefrist Ihr Projekt mit den lauffähigen Programmen des Übungsblattes in das Repository auf *wilma* pushen. Achten Sie darauf, ob Sie die Kontroll-E-Mail bekommen! Die letzte hochgeladene Version Ihres Projekts wird gewertet. Andere Abgaben, ob elektronisch oder auf Papier, zählen als **nicht abgegeben!**
- Die Aufgaben sind in Eclipse zu bearbeiten und beim Testat vorzuführen.
- Nutzen Sie die Übungsstunde, um eventuelle Fragen zur Vorlesung, zur Übung oder allgemein zum Studium zu stellen! Die Betreuer sind da, um Ihnen solche Fragen zu beantworten und Ihnen bei Bedarf zu helfen!