

## Übung 07: JUnit

Abgabetermin: 11.5.2017, 8:15

Name: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

Informatik:  G1 (Marr)       G2 (Prähofer)       G3 (Prähofer)       G4 (Löberbauer)

WIN:       G1 (Khalil)       G2 (Hummel)       G3 (Khalil)

Aufgabe	Punkte	abzugeben schriftlich	abzugeben elektronisch	korr.	Punkte
Übung 7	24	Java-Programm, JUnit-Testfälle Ausgabe eines Testlaufs	Java-Programm, JUnit- Testfälle	<input type="checkbox"/>	

### Übung 7: Ataxx

Ataxx ist ein Brettspiel für zwei Personen, das auf einem quadratischen Brett mit 7x7 Feldern gespielt wird. Zu Beginn werden für die beiden Spielenden jeweils zwei Steine in die diagonalen Ecken gesetzt (siehe Abbildung 1). Dann ziehen die Spielenden abwechselnd. Es gibt zwei Arten von Zügen (siehe Abbildung 2):

- *Setzen eines neuen Steins:* Hat der Spielende einen Stein auf einem bestimmten Feld, kann er oder sie einen Stein auf ein freies Nachbarfeld setzen (mit „+“ gekennzeichnete Felder in Abbildung 2).
- *Springen mit einem Stein:* Der Spielende kann mit einem Stein ein Feld überspringen und den Stein auf das nächste leere Feld setzen (mit „-“ gekennzeichnete Felder in Abbildung 2).

Nach einem Zug werden gegnerische Steine, die sich auf Nachbarfeldern des neu belegten Feldes befinden, zu eigenen Steinen (siehe Abbildung 3).

2						1
1						2

Abbildung 1)  
Startkonfiguration

-	-	-	-	-		
-	+	+	+	-		
-	+	X	+	-		
-	+	+	+	-		
-	-	-	-	-		

Abbildung 2) Mögliche Zugpositionen: Spieler X kann einen Stein auf die mit „+“ markierten Felder dazulegen oder auf die mit „-“ markierten Felder springen. Das Zielfeld muss in jedem Fall leer sein.

2						1
	-	-	-	-	-	
	-	+	+	+	-	
	-	+	1	+	2	
	1	1	1	1	2	
1	-	-	-	-	2	2

2						1
			1		1	
	1	1	1	1	1	
1					1	2

Abbildung 3) Übernehmen gegnerischer Felder durch Setzen eines eigenen Steines neben diese Felder.

**Aufgabe 7.a) Datenmodell**

**(12 Punkte)**

Schreiben Sie eine Klasse *AtaxxField*, die das Datenmodell mit der Spiellogik umsetzt. Folgende Funktionen muss Ihr Datenmodell erfüllen:

- a) Konstruktor, mit dem man ein neues Spielfeld anlegen kann (Stellung wie in Abbildung 1)
- b) Abfragen eines Feldes am Spielfeld (ist das Feld leer, Spieler 1 oder 2 am Feld?)
- c) Durchführen eines Spielzugs, entweder Setzen eines neuen Steins oder Springen mit einem Stein
- d) Abfragen des aktuellen Spielers
- e) Abfragen der Punkte (Anzahl der vom Spielenden besetzten Felder)

Prüfen Sie bei den Methoden, ob die übergebenen Parameter sowie der Zustand des Felds gültig sind, wie folgt:

Eine *IllegalArgumentException* muss in folgenden Fällen geworfen werden:

- a. ungültiges Startfeld
- b. ungültiges Zielfeld
- c. Start- und Zielfeld zu weit entfernt
- d. Start- und Zielfeld sind identisch

Eine *IllegalStateExceptions* muss in folgenden Fällen geworfen werden:

- e. Startfeld ~~besetzt~~ leer
- f. Zielfeld besetzt
- g. Derselbe Spieler zieht wiederholt (d.h. die Spielenden wechseln nicht ab)

**Aufgabe 7.b) JUnit-Test**

**(8 Punkte)**

Schreiben Sie JUnit-Tests, um die Implementierung der Klasse *AtaxxField* zu testen. Testen Sie, ob die:

- Steine richtig gesetzt werden
- gegnerischen Steine richtig zu eigenen Steinen werden
- Exceptions richtig geworfen werden

**Aufgabe 7.c) Konsolenspiel**

**(4 Punkte)**

Schreiben Sie ein einfaches Konsolen-Spiel, das abwechselnd die Spielenden nach ihren Zügen fragt, diese an das Spielfeld weitergibt und die Fehlermeldungen sowie das Spielfeld auf der Konsole ausgibt. Sie können für die Ein- und Ausgaben die *In-* bzw. *Out-*Klasse verwenden.

Ein Spiel könnte beispielsweise wie folgt beginnen (Eingaben der Spielenden in *grün und kursiv*):

```

|0|1|2|3|4|5|6|
0|0| | | | |X|
1| | | | | | |
2| | | | | | |
3| | | | | | |
4| | | | | | |
5| | | | | | |
6|X| | | | |0|

```

Let's start this game, you can choose any player!

Please enter your source and target positions, row and column for each (anything but a number to quit): *6 0 6 1*

```

|0|1|2|3|4|5|6|
0|0| | | | |X|
1| | | | | | |
2| | | | | | |
3| | | | | | |
4| | | | | | |
5| | | | | | |
6|X|X| | | |0|

```

Score player 0: 2

Score player X, 3

It's the turn of player: 0

Please enter your source and target positions, row and column for each (anything but a number to quit): *0 0 2 2*

```

|0|1|2|3|4|5|6|
0| | | | | |X|
1| | | | | | |
2| | |0| | | |
3| | | | | | |
4| | | | | | |
5| | | | | | |
6|X|X| | | |0|

```

Score player 0: 2

Score player X, 3

It's the turn of player: X

Please enter your source and target positions, row and column for each (anything but a number to quit): *6 1 2 2*

The maximum distance of a move is two rows and columns.

**Anmerkung:**

Halten Sie Ihren Entwurf einfach und verständlich. Sie müssen die nächste Übung darauf aufbauen.