

Übung 08: JUnit

Abgabetermin: 21. 5. 2015, 8:15

Name: _____

Matrikelnummer: _____

 Informatik: ☐ G1 (Prähofer) ☐ G2 (Prähofer) ☐ G3 (Grimmer) ☐ G4 (Grimmer)

 WIN: ☐ G1 (Khalil) ☐ G2 (Kusel) ☐ G3 (Kusel)

Aufgabe	Punkte	abzugeben schriftlich	abzugeben elektronisch	korr.	Punkte
Übung 8	24	Java-Programm, JUnit-Testfälle, Coverage-Report + Erklärung Ausgabe eines Spielablaufs	Java-Programm, JUnit-Testfälle	<input type="checkbox"/>	

Othello

Othello (oder die sehr ähnliche Variante *Reversi*) ist ein Strategiespiel für zwei Personen. In dieser Übung sollen Sie ein Programm entwerfen und implementieren, mit dem man Othello spielen kann.

Spielregeln (aus <http://de.wikipedia.org/wiki/Reversi>):

Das Spiel wird auf einem Schachbrett mit 8 x 8 Steinen und von zwei Spielern mit Steinen gespielt, die auf einer Seite schwarz und auf der anderen Seite weiß sind. Die Anfangsstellung ist wie in Abbildung 1 gezeigt: weiße Steine befinden sich auf den Feldern D4 und E5 und schwarze Steine auf den Feldern D5 und E4. Schwarz beginnt. Ein Spieler kann einen Stein auf ein Feld setzen, wenn er damit in einer beliebigen Richtung (senkrecht, waagrecht oder diagonal; Richtungen werden mit den Himmelsrichtungen N, NW, W, SW, S, SE, E, NE angegeben) gegnerische Steine einschließt. Das heißt, in einer Reihe muss direkt folgend ein oder mehrere gegnerische Steine und dann ein eigener Stein folgen. Setzt ein Spieler den Stein auf ein solches Feld, werden alle damit eingeschlossenen gegnerischen Steine umgedreht (siehe Abbildung 2). Gibt es für den Spieler kein freies Feld, mit dem er gegnerische Steine einschließen könnte, kann er nicht ziehen und muss passen. Das Spiel endet unentschieden, wenn das Spielbrett komplett gefüllt ist oder beide Spieler nicht mehr ziehen können. Es hat der Spieler gewonnen, der am Ende mehr Steine am Spielfeld hat.

Siehe <http://de.wikipedia.org/wiki/Reversi> für eine detaillierte Beschreibung der Spielregeln.

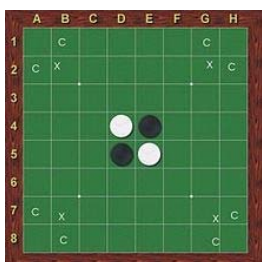


Abbildung 1

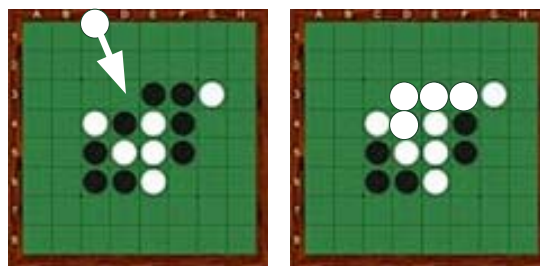


Abbildung 2

Im Download finden Sie eine Implementierung des Spiels, wobei das Spielbrett (Board) nicht implementiert ist. Die Implementierung besteht aus folgenden Klassen, Interfaces und Enumerationen:

- Package `othello`:
 - `Pos`: Enumeration zur Repräsentation der Positionen auf dem Spielfeld (Positionen wie auf dem Schachbrett).
 - `Direction`: Enumeration für die unterschiedlichen Richtungen auf dem Spielfeld.
 - `Stone`: Enumeration zur Darstellung der Belegung der Felder, d.h., `White`, `Black`, `Free`.
 - `GameState`: Enumeration zur Darstellung der unterschiedlichen Spielzustände.
 - `Board`: Interface mit Zugriffsmethoden auf das Spielfeld.
 - `BoardImpl`: Implementierung von Interface `Board` (in Ihrer Vorgabe ein `TODO`)
 - `Player`: Interface für die Implementierung von Spieler.
 - `UserInteface`: Interface zur Ausgabe des Spielstandes.
 - `Game`: Klasse, die ein Spiel aufbaut und abwickelt.

- Package `othello.ui`:
 - `ConsolePlayer`: Implementierung von Interface `Player`, der die Züge von der Konsole einliest
 - `ConsoleUI`: Implementierung von Interface `UserInterface`, der die Züge von der Konsole einliest
- Package `othello.ai`:
 - `ArtificialPlayer`: Implementierung von Interface `Player`, der die Züge berechnet
- Package `othello.app`:
 - `OthelloMain`: Klasse mit `main`-methode der ein Spiel mit einem `ConsolePlayer` und einem `ArtificialPlayer` aufbaut und startet.

Folgend ist ein Spielverlauf exemplarisch dargestellt.

	A	B	C	D	E	F	G	H
1	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-
4	-	-	-	0	X	-	-	-
5	-	-	-	X	0	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-

Move of AI with stone X
Stone X set on C_4

	A	B	C	D	E	F	G	H
1	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-
4	-	-	X	X	X	-	-	-
5	-	-	-	X	0	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-

Move of White with stone 0
Selection one of the following positions:

- 1: C_3
- 2: E_3
- 3: C_5

Select: 2
Stone 0 set on E_3

	A	B	C	D	E	F	G	H
1	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-
3	-	-	-	-	0	-	-	-
4	-	-	X	X	0	-	-	-
5	-	-	-	X	0	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-

Move of AI with stone X
Stone X set on F_2

	A	B	C	D	E	F	G	H
1	-	-	-	-	-	-	-	-
2	-	-	-	-	-	X	-	-
3	-	-	-	-	X	-	-	-
4	-	-	X	X	0	-	-	-
5	-	-	-	X	0	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-

Move of White with stone 0
Selection one of the following positions:

- 1: E_2
- 2: C_3
- 3: B_4
- 4: C_5
- 5: C_6

Select: 4
Stone 0 set on C_5

	A	B	C	D	E	F	G	H
1	-	-	-	-	-	-	-	-
2	-	-	-	-	-	X	-	-
3	-	-	-	-	X	-	-	-
4	-	-	X	X	0	-	-	-
5	-	-	0	0	0	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-

Move of AI with stone X
Stone X set on E_6

	A	B	C	D	E	F	G	H
1	-	-	-	-	-	-	-	-
2	-	-	-	-	-	X	-	-
3	-	-	-	-	X	-	-	-
4	-	-	X	X	X	-	-	-
5	-	-	0	0	X	-	-	-
6	-	-	-	-	X	-	-	-
7	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-

Move of White with stone 0
Selection one of the following positions:

- 1: G_1
- 2: B_3
- 3: C_3
- 4: D_3
- 5: F_3
- 6: F_5

Select:

Ihre Aufgaben bei diese Übung sind nun folgende:

a) Implementierung der Klasse `BoardImpl`

(10 Punkte)

Implementieren Sie die Klasse `BoardImpl` entsprechend den Vorgaben aus dem Interface `Board`.

b) Test von `BoardImpl` in einem umfangreichen JUnit-Test

(10 Punkte)

Schreiben Sie einen umfangreichen JUnit-Test für die Klasse `BoardImpl`. Der JUnit-Test soll Züge durchführen und den Zustands des Boards auf Korrektheit entsprechend den Regeln des Spiels testen.

c) Code Coverage und Analyse

(4 Punkte)

Ermitteln Sie die Codecoverage Ihrer Testfälle mit dem Eclipse Plugin `EclEmma` [1] (oder vergleichbaren Tools). Versuchen Sie mit ihren Unittests eine möglichst hohe Coverage zu erreichen. Erklären Sie warum dennoch manche Ihrer Statements nicht getestet wurden / nicht getestet werden können.

Abzugeben sind: Coverage Report (z.B. Screenshots inkl. einer kurzen schriftlichen Analyse).

Hinweis:

- In Übung Übung 9 werden wir interaktive und graphische Varianten des Spiels programmieren. Wenn Sie in dieser Übung eine gute Basis schaffen, können Sie einiges davon in Übung 9 wiederverwenden.

[1] <http://www.eclemma.org/>