

Hausaufgaben 10

04.12./05.12.2019

Abgabe der Lösung am 10.12.2019

Schachfigur

Ziel dieser Aufgabe ist es, für eine Schachfigur auszugeben, welche Felder sie prinzipiell (d.h. ohne Berücksichtigung anderer Figuren) erreichen kann. Programmieren Sie dazu die Klassen `Chessman` für eine Schachfigur und `Position` für eine Position:

Die abstrakte Klasse `Chessman` (Schachfigur) besitzt die Position der Figur als Attribut und umfasst folgende Methoden:

- **public** `Chessman(Position pos)`
Setzt die Figur auf den angegebenen Punkt.
- **public** `Position getPosition()`
Gibt die aktuelle Position zurück.
- **public** `void moveTo(Position pos)`
Setzt die Figur auf den angegebenen Punkt. Löst eine (selbst zu schreibende) `WrongMoveException` aus, falls das Feld nicht in einem Zug erreicht werden kann. Andere Figuren auf dem Brett werden nicht berücksichtigt.
- **public abstract** `ArrayList<Position> getMoveList()`
Gibt die Liste der Positionen zurück, die mit der Figur in einem Zug erreicht werden können. Muss in den Unterklassen überschrieben werden. Andere Figuren auf dem Brett werden nicht berücksichtigt.
- **public boolean** `canMoveTo(Position pos)`
Gibt zurück, ob die Figur die angegebene Position in einem Zug erreichen kann.

Die Klasse `Position` hat als Inhalt die x-y-Position einer Schachfigur. Beachten Sie, dass die Koordinaten bei 1 beginnen und Werte bis inkl. 8 annehmen können. Ein Positions-Objekt ist unveränderlich. Die Klasse `Position` hat folgende Methoden:

- **public** `Position(int x, int y)`
Setzt die Position auf den Wert (x,y). Wirft eine (selbst zu schreibende) `WrongPositionException`, falls die Position außerhalb des Bretts liegt.
- **public int** `getX()`
Gibt die x-Koordinate der Position zurück
- **public int** `getY()`
Gibt die y-Koordinate der Position zurück.
- **public boolean** `equals(Position p)`
Gibt zurück, ob die Positionen `this` und `p` gleich sind.
- **public static boolean** `isValid(int x, int y)`
Gibt zurück, ob die angegebenen Koordinaten eine gültige Position auf dem Schachbrett definieren.
- **public** `String toString()`

Ferner sollen die Klassen `Knight` (Springer) und `Rook` (Turm) implementiert werden. Diese Klassen erben von der abstrakten Klasse `Chessman`. Implementieren Sie die abstrakte Methode. Überschreiben Sie auch die entsprechende `toString()`-Methode. Andere Arten von Figuren müssen nicht hinzugefügt werden.

Testfall:

```
public static void main(String[] args) {
    // Test eines Turms
    Rook rook = new Rook(new Position(4,5));
    System.out.println(rook);
    rook.moveTo(new Position(4,1));
    System.out.println(rook);
    System.out.println(rook.getMoveList());

    // Test eines Springers
    Knight knight = new Knight(new Position(5,4));
    System.out.println(knight);
    knight.moveTo(new Position(6,6)); // mitten auf dem Feld
    System.out.println(knight);
    System.out.println(knight.getMoveList());
    knight.moveTo(new Position(8,5)); // am Rand des Feldes
    System.out.println(knight);
    System.out.println(knight.getMoveList());
}
```

Ausgabe:

```
Turm: (4/5)
Turm: (4/1)
[(1/1), (4/2), (2/1), (4/3), (3/1), (4/4), (4/5), (5/1), (4/6), (6/1), (4/7),
(7/1), (4/8), (8/1)]
Springer: (5/4)
Springer: (6/6)
[(4/5), (4/7), (5/8), (7/8), (5/4), (7/4), (8/5), (8/7)]
Springer: (8/5)
[(6/4), (6/6), (7/7), (7/3)]
```

(Die Reihenfolge der erreichbaren Felder darf variieren).