

Fachbereich Informatik

Einführung in die Programmierung - Projektwoche WS 2022/23

Aufgabenblatt B – „Konsolen-Spiele“

Aufgabe 1 – Solitär

Erstellen Sie eine Java Konsolen-Anwendung für das Spiel „Solitär“ nach den unten angegebenen 10 Punkten.

Spielfeld: 5*5-Quadrat-Solitär

```
    0 1 2 3 4
    -----
0|# # # # #
1|# # # # #
2|# # # # #
3|# # _ # #
4|# # # # #
```

- (1) Implementieren Sie ein quadratisches 5*5 Solitär Spiel.
- (2) Halten Sie die Spielfeldgröße trotz der festen Angabe von 5*5 Feldern variabel.
- (3) Erstellen Sie eine Klasse `Solitaer` mit der Hauptmethode `main`.
- (4) Erstellen Sie eine Klasse `Spielfeld`.
- (5) Wird ein Pin(#) von einem anderen übersprungen, wird er vom Spielfeld genommen.
- (6) Ein Pin darf nicht aus dem Spielfeld gezogen werden.
- (7) Ein Pin darf nicht auf einen anderen gezogen werden.
- (8) Implementieren Sie eine Methode `ausgabeSpielfeld`
- (9) Implementieren Sie eine Methode `starteSpiel`, die ein Spielfeld nach dem oben aufgeführten Beispiel erstellt.
- (10) Erstellen Sie eine Methode `zug`, und implementieren Sie in dieser Ihre Spiellogik. In dieser Methode soll auch die Eingabe der Koordinaten vorgenommen werden (es wird der Pin gewählt, **der in das leere Feld** gesteckt werden soll).

Aufgabe 2 – Vier Gewinnt

Erstellen Sie eine Java Konsolenanwendung für das Spiel „Vier Gewinnt“, welches als Zweispieler-Version spielbar ist.

- (1) Die Klasse `VierGewinnt` verwaltet die zwei Spieler und steuert den Spielablauf.
- (2) Die Klasse `Spieler` beinhaltet einen Konstruktor, dem ein Namen und die Spieler-Nr. (1 oder 2) übergeben wird.
- (3) Es gibt jeweils eine `get`-Methode für die beiden Attribute.
- (4) Die Klasse `Spielfeld` besitzt einen Konstruktor, dem die Größe des Spielfeldes (Standard: 6 Reihen mit 7 Spalten) übergeben wird und der anschließend das Spielfeld als zweidimensionales Feld (Array) implementiert.
- (5) Die Methode `initialisiereSpielfeld` initialisiert das Spielfeld anfänglich mit 0 (Null).
- (6) Es gibt eine Methode `druckeSpielfeld`, die das aktuelle Spielfeld, samt gesetzter Spielsteine, ausgibt.
- (7) Ein Spielstein hat die entsprechende Spieler-Nr. als Wert.
- (8) Die Methode `setzeSpielstein` setzt, je nach aktuellem Spieler und der jeweiligen Einwurfsstelle, den Spielstein in die entsprechende Spalte.
- (9) Die Methode `testeSieg` testet, je nach Spieler, das Spielfeld auf einen möglichen Sieg. Man hat gewonnen, wenn 4 Spielsteine horizontal, vertikal oder diagonal miteinander verbunden sind.
Sind alle 42 Spielsteine gespielt, ohne dass ein Spieler gewonnen hat, endet das Spiel unentschieden.
- (10) Erstellen Sie für die Klasse `Spielfeld`, falls benötigt, `get`- und `set`-Methoden.

Spielfeld mit einem möglichen Spielverlauf:

VIER-GEWINNT

1		0		0		0		0		0		0		0	
2		0		0		0		0		0		0		0	
3		0		0		0		0		0		0		0	
4		0		0		0		0		0		0		0	
5		0		0		0		0		0		0		0	
6		0		0		0		0		0		0		0	

1	2	3	4	5	6	7
---	---	---	---	---	---	---

Spieler 1: Donald

Spalte: 4

1		0		0		0		0		0		0		0	
2		0		0		0		0		0		0		0	
3		0		0		0		0		0		0		0	
4		0		0		0		0		0		0		0	
5		0		0		0		0		0		0		0	
6		0		0		0		1		0		0		0	

1	2	3	4	5	6	7
---	---	---	---	---	---	---

Spieler 2: Mickey

Spalte: 1

1		0		0		0		0		0		0		0	
2		0		0		0		0		0		0		0	
3		0		0		0		0		0		0		0	
4		0		0		0		0		0		0		0	
5		0		0		0		0		0		0		0	
6		2		0		0		1		0		0		0	

1	2	3	4	5	6	7
---	---	---	---	---	---	---

Spieler 1: Donald

Spalte: 4

1		0		0		0		0		0		0		0	
2		0		0		0		0		0		0		0	
3		0		0		0		0		0		0		0	
4		0		0		0		0		0		0		0	
5		0		0		0		1		0		0		0	
6		2		0		0		1		0		0		0	

1	2	3	4	5	6	7
---	---	---	---	---	---	---

Aufgabe 3 – Schiffe versenken

Bei dem Spiel „Schiffe versenken“ geht es darum, dass 2 Spieler verdeckt auf ihrem eigenen Spielfeld Schiffe platzieren. Es sollen folgende Schiffe gesetzt werden:

- 3 U-Boote (Länge 2)
- 2 Zerstörer (Länge 3)
- 1 Kreuzer (Länge 4)
- 1 Schlachtschiff (Länge 5)

Ziel des Spiels ist es, alle Schiffe des Gegners zu versenken, bevor dieser die eigenen Schiffe versenkt hat. Es wird abwechselnd immer ein Schuss auf das gegnerische Feld abgegeben. Der Schütze erhält Rückmeldung, ob er ein Schiff oder das Wasser getroffen hat.

Am Ende der Aufgabenstellung sehen Sie ein Beispiel für ein laufendes Spiel.

Hinweis: Sie können jederzeit Hilfsmethoden implementieren, die Ihnen z.B. beim Überprüfen helfen, ob ein angegebenes Feld gültig ist oder nicht.

- (1) Erstellen Sie ein Spielfeld der Größe 10*10. Schreiben Sie eine Methode `printField`, mit der Sie das Spielfeld mit Beschriftung darstellen können. Für die Reihen verwenden Sie die Buchstaben: A – J und für die Spalten die Zahlen: 1 – 10.

```
      1  2  3  4  5  6  7  8  9  10
A
B
C
D
E
F
G
H
I
J
Schiff der Laenge 2 setzen:
Startpunkt Reihe angeben:
```

Bedenken Sie, dass Sie beim eigenen Spielfeld die Schiffe sehen sollen, beim Gegner aber nicht.

Für das Spielfeld gibt es folgende Beschriftungen:

- S – Schiff
- O – Schuss ins Wasser
- * – Treffer eines Schiffes

Felder, auf denen Wasser ist oder die der Spieler beim Gegner nicht einsehen kann, werden leer dargestellt.

- (2) Schreiben Sie eine Methode `printFields`, die das Spielfeld des Gegners und des Spielers ausgibt:

```
Gegner:
  1  2  3  4  5  6  7  8  9 10
A
B
C
D
E
F
G
H
I
J

-----

Spieler:
  1  2  3  4  5  6  7  8  9 10
A
B
C
D
E
F
G
H
I
J

Reihe angeben:
```

- (3) Schreiben Sie eine Methode `setShip` zum Setzen eines Schiffes auf dem Spielfeld. Übergeben werden soll der Start und Endpunkt. Bedenken Sie, dass Schiffe nur in gerader Linie gesetzt werden dürfen. Sie müssen innerhalb des Spielfeldes anfangen sowie aufhören, und sie dürfen sich nicht überschneiden. Außerdem müssen die Positionen mit der Schiffslänge übereinstimmen. Das bedeutet, dass ein U-Boot der Länge 2, welches auf A1 beginnt, nur auf A2 oder B1 enden kann. Geben Sie Rückmeldung über die Konsole, wenn diese Vorgabe nicht eingehalten wird.

```
    1  2  3  4  5  6  7  8  9 10
A
B
C
D
E
F
G
H
I
J
Schiff der Laenge 2 setzen:
Startpunkt Reihe angeben: B
Startpunkt Spalte angeben: 2
Endpunkt Reihe angeben: B
Endpunkt Spalte angeben: 5
Schiffslaenge nicht korrekt.
Startpunkt Reihe angeben:
```

- (4) Erstellen Sie eine Methode `shoot` für den Schuss auf ein Feld. Lassen Sie die Auswirkung des Schusses sichtbar werden:

○ – Schuss ins Wasser
* – Treffer eines Schiffes

Natürlich muss hier wieder ein gültiges Feld gewählt werden. Befindet sich das anvisierte Feld außerhalb des Spielfeldes, geben Sie bitte, wie schon unter (3), eine Rückmeldung.

Geben Sie ebenfalls eine Rückmeldung, wenn das Feld bereits beschossen wurde, und lassen Sie den Spieler (oder die „Künstliche Intelligenz“, die unter (7) programmiert werden soll) erneut schießen, bis ein Feld gewählt wird, das frei ist.

- (5) Erstellen Sie eine Methode `start`, in der ein Spieler seine Schiffe setzen kann. Dies soll mit Hilfe eines Scanners:

```
Scanner sc = new Scanner(System.in);
```

geschehen. Die Koordinaten werden der Methode unter (3) übergeben. Nach jedem gesetzten Schiff soll der Spieler sein Spielfeld sehen können.

- (6) Erstellen Sie eine Methode `update`, in der eine Schleife solange läuft, bis einer das Spiel gewonnen hat. Innerhalb dieser Schleife kann der Spieler ein Feld beschießen, indem er die Position angibt. Diese Position kann wieder mit dem Scanner eingelesen und an die Methode unter (4) übergeben werden.

- (7) Ergänzen Sie die Methoden unter (4) und (5) um eine einfache Gegner-KI. Beim Setzen soll sie die Startposition zufällig bestimmen. Das kann über:

```
Random r = new Random();
r.nextInt();
```

geschehen. Das Beschießen von Feldern kann ebenfalls zufällig geschehen. Die KI muss sich nicht danach richten, ob sie bereits ein Schiff getroffen hat oder nicht.

Nachfolgend ist eine beispielhafte Spielsituation aufgeführt:

```
Reihe angeben: G
Spalte angeben: 6
Gegner:
  1  2  3  4  5  6  7  8  9 10
A *  *  0      *  0
B      *
C      0      *  0
D      0
E      0      0  0
F      0      0  0
G      0      0
H
I
J
```

```
-----
Spieler:
  1  2  3  4  5  6  7  8  9 10
A      S  S      0      *
B      0      0      S      0  S
C      0      0      S      S
D      0      S
E      0      S      0
F      0  S      S  *  S  S
G  S  S  0  S      0
H      S
I      0  *
J      0  0  0  0      S  S  S
```

Im Feld des Gegners werden nur Treffer (*) bzw. Nicht-Treffer (0) der Schiffe angezeigt. Im Feld des Spielers sind zusätzlich noch die Positionen der Schiffe sichtbar.