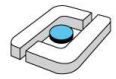




Programmierung 1

Aufgabenblatt 6 (Arrays)

Prof. Dr.-Ing. Heiko Tapken / Programmier-Team
Wintersemester 2020/21
Bearbeitung: KW 48, Testat KW 49
Erreichbar: 14 Punkte, Bestehensgrenze: 10 Punkte



Hochschule Osnabrück
University of Applied Sciences

Aufgabe 1 (2-Dimensionales Array) [7 Punkte]

Vielleicht kennen Sie aus Ihrer Kindheit das so genannte **Slider-Spiel**. Es ist ein Spiel für eine Person. Das Spielgerät ist eine Tafel mit $4 * 4$ Feldern. In dieser Tafel sind auf 15 Feldern Plättchen mit den Ziffern 1 bis 15 platziert. Ein Feld ist leer. Die Plättchen sind verschiebbar. Gegeben eine bestimmte Ausgangsstellung der Plättchen ist es das Ziel, die Plättchen in der Reihenfolge 1 bis 15 anzuordnen.

Schreiben Sie ein Programm, mit dem ein Benutzer das Slider-Spiel spielen kann. Beachten und behandeln Sie falsche Benutzereingaben. Achten Sie auf einen sauberen Programmentwurf (prozedurale Zerlegung)! Wählen Sie aussagekräftige Bezeichner! Die Ausgangsstellung können Sie beliebig vorgeben.

Beispielablauf:

```
+---+---+---+---+
| 11|12|13|14|
+---+---+---+---+
| 1| 2| 3| 4|
+---+---+---+---+
| 8| 7| 6| 5|
+---+---+---+---+
|  | 9|10|15|
+---+---+---+---+
```

Zeile: 4
Spalte: 2

```
+---+---+---+---+
| 11|12|13|14|
+---+---+---+---+
| 1| 2| 3| 4|
+---+---+---+---+
| 8| 7| 6| 5|
+---+---+---+---+
| 9|  |10|15|
+---+---+---+---+
```

...

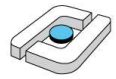
```
+---+---+---+---+
|  | 1| 2| 3|
+---+---+---+---+
| 4| 5| 6| 7|
+---+---+---+---+
| 8| 9|10|11|
+---+---+---+---+
```



Programmierung 1

Aufgabenblatt 6 (Arrays)

Prof. Dr.-Ing. Heiko Tapken / Programmier-Team
Wintersemester 2020/21
Bearbeitung: KW 48, Testat KW 49
Erreichbar: 14 Punkte, Bestehensgrenze: 10 Punkte



Hochschule Osnabrück
University of Applied Sciences

| 12 | 13 | 14 | 15 |
+--+--+--+--+

Achtung: Das Problem ist nicht mit jeder Startbelegung lösbar!

Für Fortgeschrittene eine Version mit GUI: Implementieren Sie im Projekt `SliderMitGUI_StudiVersion` (Ilias) die Klasse `Spiefeld.java`, sodass Sie das Spiel grafisch spielen können.

Aufgabe 2 [7 Punkte]:

Sie alle kennen das Spiel **Schiffe versenken**. Ihre Aufgabe besteht darin, eine sehr einfache Version des Spiels zu implementieren. Zwei Spieler treten dabei gegeneinander an.

Jeder Spieler hat ein Spielfeld beliebiger, aber fester Größe, auf dem auf 10% der Felder (abgerundet) je ein Schiff mit 1x1 Felder Ausdehnung platziert ist. Der prozentuale Anteil der zu belegenden Felder ist in einer Konstante zu speichern. Die Spielfelder der Kandidaten sind natürlich gleich groß. Sie sind rechteckig. Es wird zu Beginn festgelegt, wie viele Punkte ein Spieler erreichen muss, um das Spiel zu gewinnen. Um zu überprüfen, ob diese Zahl gültig ist, wird die maximal erreichbare Punktzahl auf den Spielfeldern gezählt.

Die Spieler erraten abwechselnd ein Feld des Gegners. Wird ein Schiff getroffen, erhält der ratende Spieler Punkte in Höhe der Quersumme der Indizes, das Feld wird markiert und der Spieler rät noch einmal. Vor dem Raten wird dem Spieler das Spielfeld des Gegners (natürlich ohne die Schiffe) angezeigt. Verfehlt er die Schiffe des Gegners ist der Gegner am Zug. Wird ein bereits erratenes Spielfeld erneut erraten wird der Zug ebenfalls beendet. Das Spiel ist beendet, wenn ein Spieler die Gewinnpunktzahl erreicht hat. Bitte beachten Sie: Errät Spieler ein Schiff auf dem Feld [0][0] so erhält er keine Punkte dafür.

Nach Spielende werden das Spielergebnis und die Spielfelder der Spielfelder samt Platzierung der Schiffe ausgegeben.

Implementieren Sie dieses Spiel OHNE die Verwendung von globalen Variablen. Nur Konstanten dürfen global definiert werden.

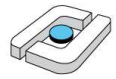
Fortgeschrittene Programmierer dürfen gern das „normale“ Schiffe versenken programmieren, wobei die Punkteverteilung analog zum beschriebenen Verfahren bestehen bleiben soll. Das Spiel ist dann aber erst dann beendet, wenn sich kein Schiff mehr auf dem Spielfeld befindet.



Programmierung 1

Aufgabenblatt 6 (Arrays)

Prof. Dr.-Ing. Heiko Tapken / Programmier-Team
Wintersemester 2020/21
Bearbeitung: KW 48, Testat KW 49
Erreichbar: 14 Punkte, Bestehensgrenze: 10 Punkte



Hochschule Osnabrück
University of Applied Sciences

Hinweise zur Aufgabe „Schiffe versenken“

In der Zip-Datei schiffe.zip befindet sich eine mögliche Implementierung der Aufgabe.
Entpacken Sie die Datei und starten Sie von der Kommandozeile mit dem Befehl:

```
java -jar schiffe.jar  
(Doppelklick reicht nicht!!!)
```

Hinweis:

Zur Vereinfachung wird in dieser Version die Lage der Schiffe mit ausgegeben.

Ein beispielhafter Ablauf mit ergänzenden Testausgaben.

Breite des Feldes: 8

Wie viele Punkte bis zum Gewinn? 999 Anm: bewusst falsche Eingabe

0 ;7 Anm: Platzierungen der Schiffe zur Verdeutlichung der Punkteberechnung

1 ;3

5 ;1

5 ;3

5 ;6

6 ;5

Es können 47 Punkte erreicht werden Anm: Erreichbare Punktzahl für Spieler 1, 999>47, also nochmal

Wie viele Punkte bis zum Gewinn? 10 Anm: korrekte/erreichbare Eingabe

0 ;7

1 ;3

5 ;1

5 ;3

5 ;6

6 ;5

Es können 47 Punkte erreicht werden Anm: Erreichbare Punktzahl für Spieler 1, 10<47, also ok

0 ;0

2 ;2

3 ;2

3 ;3

4 ;7

7 ;2

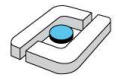
Es können 35 Punkte erreicht werden Anm: Erreichbare Punktzahl für Spieler 2, 10<35, also ok, das Spiel beginnt



Programmierung 1

Aufgabenblatt 6 (Arrays)

Prof. Dr.-Ing. Heiko Tapken / Programmier-Team
Wintersemester 2020/21
Bearbeitung: KW 48, Testat KW 49
Erreichbar: 14 Punkte, Bestehensgrenze: 10 Punkte



Hochschule Osnabrück
University of Applied Sciences

Spieler 1 ist dran:

	0	1	2	3	4	5	6	7
0								1
1				1				
2								
3								
4								
5		1		1			1	
6						1		
7								

y0

x0

Spieler 1 hat 0 von 10 Punkte.

Spieler 2 ist dran:

	0	1	2	3	4	5	6	7
0	1							
1								
2			1					
3			1	1				
4								1
5								
6								
7			1					

y0

x0

TREFFER

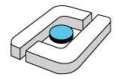
Spieler 2 hat 0 von 10 Punkte.



Programmierung 1

Aufgabenblatt 6 (Arrays)

Prof. Dr.-Ing. Heiko Tapken / Programmier-Team
Wintersemester 2020/21
Bearbeitung: KW 48, Testat KW 49
Erreichbar: 14 Punkte, Bestehensgrenze: 10 Punkte



Hochschule Osnabrück
University of Applied Sciences

Spieler 1 ist dran:

	0	1	2	3	4	5	6	7
0	3							1
1				1				
2								
3								
4								
5		1		1			1	
6						1		
7								

y6

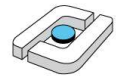
x5

TREFFER

Spieler 1 hat 11 von 10 Punkte.

Spieler 1 hat gewonnen

	0	1	2	3	4	5	6	7
0	3							1
1				1				
2								
3								
4								
5		1		1			1	
6						2		
7								



Programmierung 1

Aufgabenblatt 6 (Arrays)

Prof. Dr.-Ing. Heiko Tapken / Programmier-Team
Wintersemester 2020/21
Bearbeitung: KW 48, Testat KW 49
Erreichbar: 14 Punkte, Bestehensgrenze: 10 Punkte

Hochschule Osnabrück
University of Applied Sciences

Tipps zum Vorgehen:

Wenn die Aufgabe zu komplex ist beginnen Sie mit stark vereinfachten Versionen des Programms und orientieren Sie sich an einem realen Spiel: erst aufbauen, dann spielen. Nehmen Sie nach und nach Funktionalität hinzu.

- Vereinfachung 1: Implementieren Sie das Spiel zunächst nur für einen Spieler.
- Vereinfachung 2: Spielen Sie zunächst ggf. auf einem eindimensionalen Feld.
- Vereinfachung 3: Platzieren sie Schiffe (z.B. durch 1 repräsentiert) händisch auf dem Spielfeld.
- Vereinfachung 4: Vereinfachung der Punktezahlung: Für jedes geratene Schiff gibt es einen Punkt (statt Quersummen über Indizes). Es müssen genauso viele Schiffe geraten werden, wie auf dem Spielfeld platziert sind. Weitere Vereinfachung: Es soll zunächst nur ein Schiff erraten werden, das Programm endet dann.

Überlegen Sie sich vorab(!!!!), wie Sie an die Aufgabe herangehen. Bei der Komplexität des Programms können Sie das Programm nicht mittels Trial&Error erstellen! Skizzieren Sie Ihr Vorgehen auf Papier! Besprechen Sie Ihre Ideen / Ihr Konzept mit mir oder Ihren Praktikumsbetreuern und programmieren Sie nicht drauflos. Wir geben dann Anregungen zum weiteren Vorgehen *ausgehend von Ihrem Konzept*. Ohne Konzept keine Unterstützung!

Überlegen Sie dazu:

- Wie wollen Sie vorgehen? Welche Funktionalität möchten Sie nach und nach hinzunehmen? Welche Speicherungsstruktur benötigen sie jeweils, um Ihr Ziel zu erreichen? Welche Methoden benötigen Sie dazu? Was benötigen die Methoden, um ihre Funktionalität zu erreichen?

Bsp:

Datenstruktur: `int[][] spielFeldSpieler1 = int[groesse][groesse]`

Kodierung:

- 0 = frei,
- 1 = Boot,
- 2 = getroffen,
- 3 = Feld geraten, aber kein Boot.

Einige Methoden (sie benötigen mehr!):

Platzieren eines Boots durch eine Methode:

```
{  
public static boolean bootPlatzieren(int[][] spielFeld, int y, int x)  
  
    bekommt eine Referenz auf ein Spielfeld und setzt ein Boot an die Stelle spielFeld[y][x], wenn  
    der Platz noch nicht mit einem Boot belegt ist. Liefert true, wenn das Boot platziert werden  
    konnte, false sonst.
```

```
public static int anzahlBoote(int breite) {  
    liefert die Anzahl der zu platzierenden Boote.
```

Der prozentuale Anteil der zu belegenden Felder (10%) ist in der globalen Konstante

```
public static final int PROZENTUALERANTEIL = 10;
```

definiert.

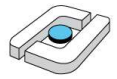


Programmierung 1

Aufgabenblatt 6 (Arrays)

Prof. Dr.-Ing. Heiko Tapken / Programmier-Team
Wintersemester 2020/21

Bearbeitung: KW 48, Testat KW 49
Erreichbar: 14 Punkte, Bestehensgrenze: 10 Punkte



Hochschule Osnabrück
University of Applied Sciences

Zeichnen sie sich bei Bedarf den Ablauf des (zunächst ggf. noch vereinfachten) Spiels auf Papier auf und machen Sie sich über die Abläufe ein Bild.