

## W04P01 - Schiffe Versenken

Präsenzaufgabe

Submission due: 5 months ago

Points: 32

Optional

Assessment: automatic ?

Practice

Clone Repository

Tasks:

?

?

?

?

## Schiffe versenken

In dieser Aufgabe sollst du ein Programm schreiben, mit dem man "Schiffe versenken" gegen den Computer spielen kann. Der Spielablauf ist dabei wie folgt: Das Spielfeld besteht aus 8x8 Feldern. Auf diesem Spielfeld befinden sich feindliche Schiffe, die für den Spieler nicht sichtbar sind. Der Spieler kann nun eines der Felder aussuchen, um darauf zu schießen. Befindet sich dort ein feindliches Schiff, wird das Feld als Treffer markiert, andernfalls als Fehlwurf. Sobald alle feindlichen Schiffe getroffen wurden, gewinnt der Spieler.

Im Programm wird das Spielfeld als zweidimensionales `int`-Array repräsentiert, wobei der erste Index die Zeile angibt und der Zweite die Spalte, gezählt von links oben. Das Feld `[0][0]` ist die linke obere Ecke, das Feld `[1][0]` befindet sich darunter und `[0][1]` rechts daneben. An jeder Position im Array steht eine Zahl, die angibt, was sich am jeweiligen Punkt befindet: 0 steht für Wasser, 1 für ein noch nicht entdecktes Schiff, 2 für einen Fehlschuss und 3 für einen Treffer. Dem Spieler gegenüber werden die Zeilen mit den Ziffern 1-8 und die Spalten mit den Buchstaben a-h nummeriert. Nutzen Sie die `PinguLib`-Methode `generateBoard()`, die Ihnen ein `int[][]` Array mit verdeckten Schiffen zurückgibt.

Der Programmablauf soll mit Methoden strukturiert werden. Schreiben Sie dazu folgende Methoden. Gehen Sie davon aus, dass das `board` quadratisch und nicht `null` ist und mindestens Größe 1 hat.

## 1. ? Eingabe No results

Schreibe eine Methode `static int readNumberInInterval(int min, int max, String question)`, die mit dem Text `question` eine Zahl zwischen `min` und `max` (einschließlich) einliest und zurückgibt; Die Eingabeaufforderung soll wiederholt werden, bis die Eingabe korrekt ist. Die Eingabe entspricht dann der gewünschten Zeile auf unserem Spielbrett. Dabei befindet sich in dem Array für das Spielfeld Zeile 1 an Index 0, Zeile 2 an Index 1, usw. . Schreibe zudem eine Methode `static char readCharacterInInterval(char min, char max, String question)`, die mit dem Text `question` einen Kleinbuchstaben vom Nutzer zwischen `min` und `max` einliest. Die Eingabeaufforderung soll wiederholt werden, bis die Eingabe korrekt ist. Die Rückgabe soll die Position des Buchstabens im Alphabet sein, also 0 für a, 1 für b usw... Nutze zum Einlesen des Zeichens die `PinguLib`-Methode `readChar()`.

## 2. ? Ausgabe No results

Schreibe eine Methode `static void printBoard(int[][] board)`, die das übergebene Spielfeld wie folgt ausgibt: Zuerst kommt ein " ", danach folgt die Spaltennummerierung in Buchstaben (also z.B. "abc" bei Länge 3). Danach folgen die Zeilen des Feldes, wobei ein Leerzeichen für Wasser oder ein verdecktes Schiff, ein " " für einen Fehlschuss und ein " " für einen Treffer ausgegeben wird, außerdem wird jeder Zeile die Zeilennummer vorangestellt. Betrachten Sie bei Unklarheiten die Beispielausgabe.

## 3. ? Test auf Ende No results

Schreibe eine Methode `static boolean shipsLeft(int[][] board)`, die prüft, ob das übergebene Spielfeld noch verdeckte Schiffe enthält. Enthält das übergebene Feld keine verdeckten Schiffe, soll `false` zurückgegeben werden, andernfalls `true`.

Vervollständigen Sie zuletzt die `battleship()`-Methode, die den Gesamtablauf unter Verwendung der Methoden umsetzt:

- Zuerst wird mittels `generateBoard()` das Spielfeld generiert.
- Anschließend wird "Willkommen bei Schiffe-Versenken!" sowie darunter das zu Beginn "leere" Feld ausgegeben.
- Danach wird der Nutzer nach der Zahl- sowie Buchstabenkoordinate gefragt, auf die geschossen werden soll, mit den Fragen "Zahl koordinaten eingeben" respektive "Buchstaben-Koordinate eingeben".
- Das gewählte Feld wird entsprechend markiert. Wurde es allerdings bereits beschossen, soll stattdessen "Feld bereits beschossen!" ausgegeben werden.
- Das neue Feld wird ausgegeben und die nächste Eingabe verlangt.
- Sobald alle Schiffe versenkt wurden, soll "Glückwunsch! Sie haben gewonnen!" ausgegeben werden.

## ? Beispielablauf No results

```
Willkommen bei Schiffe-Versenken!
+abcdefgh
1
2
3
4
5
```

```
6
7
8
Zahl-Koordinate eingeben:
> 0
Zahl-Koordinate eingeben:
> 1
Buchstaben-Koordinate eingeben:
> i
Buchstaben-Koordinate eingeben:
> c
+abcdefgh
1 x
2
3
4
5
6
7
8
Zahl-Koordinate eingeben:
> 1
Buchstaben-Koordinate eingeben:
> c
Feld bereits beschossen!
Zahl-Koordinate eingeben:
> 2
Buchstaben-Koordinate eingeben:
> c
+abcdefgh
1 x
2 x
3
4
5
6
7
8
Zahl-Koordinate eingeben:
> 3
Buchstaben-Koordinate eingeben:
> c
+abcdefgh
1 x
2 x
3 o
4
5
6
7
8
Zahl-Koordinate eingeben:
> 4
Buchstaben-Koordinate eingeben:
> a
+abcdefgh
1 x
2 x
3 o
4x
5
6
7
8
Zahl-Koordinate eingeben:
> 4
Buchstaben-Koordinate eingeben:
> b
+abcdefgh
1 x
2 x
3 o
4xx
5
6
7
8
Zahl-Koordinate eingeben:
> 4
Buchstaben-Koordinate eingeben:
> c
+abcdefgh
1 x
2 x
3 o
4xxx
5
6
7
8
Zahl-Koordinate eingeben:
> 4
Buchstaben-Koordinate eingeben:
> d
+abcdefgh
1 x
2 x
```

```

3 o
4xxxo
5
6
7
8
Zahl-Koordinate eingeben:
> 4
Buchstaben-Koordinate eingeben:
> e
+abcdefgh
1 x
2 x
3 o
4xxxox
5
6
7
8
Zahl-Koordinate eingeben:
> 4
Buchstaben-Koordinate eingeben:
> f
+abcdefgh
1 x
2 x
3 o
4xxxoxo
5
6
7
8
Zahl-Koordinate eingeben:
> 8
Buchstaben-Koordinate eingeben:
> h
+abcdefgh
1 x
2 x
3 o
4xxxoxo
5
6
7
8 o
Zahl-Koordinate eingeben:
> 5
Buchstaben-Koordinate eingeben:
> e
+abcdefgh
1 x
2 x
3 o
4xxxoxo
5 x
6
7
8 o
Zahl-Koordinate eingeben:
> 6
Buchstaben-Koordinate eingeben:
> e
+abcdefgh
1 x
2 x
3 o
4xxxoxo
5 x
6 x
7
8 o
Zahl-Koordinate eingeben:
> 7
Buchstaben-Koordinate eingeben:
> e
+abcdefgh
1 x
2 x
3 o
4xxxoxo
5 x
6 x
7 x
8 o
Glückwunsch! Sie haben gewonnen!

```

Die Zeilen, die mit "> " beginnen, markieren dabei die Nutzereingaben und werden nicht von dem Programm ausgegeben. Sie dienen nur der besseren Verständlichkeit.

[Lösungsvorschlag](#)

[Test](#)

W04P01 - Schiffe Versenken **Präsenzaufgabe**Submission due: **5 months ago**Points: 32 **Optional** Assessment: automatic ?

Practice

Clone Repository

Tasks:

?

?

?

?

## Schiffe versenken

In dieser Aufgabe sollst du ein Programm schreiben, mit dem man "Schiffe versenken" gegen den Computer spielen kann. Der Spielablauf ist dabei wie folgt: Das Spielfeld besteht aus 8x8 Feldern. Auf diesem Spielfeld befinden sich feindliche Schiffe, die für den Spieler nicht sichtbar sind. Der Spieler kann nun eines der Felder aussuchen, um darauf zu schießen. Befindet sich dort ein feindliches Schiff, wird das Feld als Treffer markiert, andernfalls als Fehlwurf. Sobald alle feindlichen Schiffe getroffen wurden, gewinnt der Spieler.

Im Programm wird das Spielfeld als zweidimensionales `int`-Array repräsentiert, wobei der erste Index die Zeile angibt und der Zweite die Spalte, gezählt von links oben. Das Feld `[0][0]` ist die linke obere Ecke, das Feld `[1][0]` befindet sich darunter und `[0][1]` rechts daneben. An jeder Position im Array steht eine Zahl, die angibt, was sich am jeweiligen Punkt befindet: 0 steht für Wasser, 1 für ein noch nicht entdecktes Schiff, 2 für einen Fehlschuss und 3 für einen Treffer. Dem Spieler gegenüber werden die Zeilen mit den Ziffern 1-8 und die Spalten mit den Buchstaben a-h nummeriert. Nutzen Sie die `Pingulib`-Methode `generateBoard()`, die Ihnen ein `int[][]` Array mit verdeckten Schiffen zurückgibt.

Der Programmablauf soll mit Methoden strukturiert werden. Schreiben Sie dazu folgende Methoden. Gehen Sie davon aus, dass das `board` quadratisch und nicht `null` ist und mindestens Größe 1 hat.

### 1. ? Eingabe No results

Schreibe eine Methode `static int readNumberInInterval(int min, int max, String question)`, die mit dem Text `question` eine Zahl zwischen `min` und `max` (einschließlich) einliest und zurückgibt; Die Eingabeaufforderung soll wiederholt werden, bis die Eingabe korrekt ist. Die Eingabe entspricht dann der gewünschten Zeile auf unserem Spielbrett. Dabei befindet sich in dem Array für das Spielfeld Zeile 1 an Index 0, Zeile 2 an Index 1, usw. . Schreibe zudem eine Methode `static char readCharacterInInterval(char min, char max, String question)`, die mit dem Text `question` einen Kleinbuchstaben vom Nutzer zwischen `min` und `max` einliest. Die Eingabeaufforderung soll wiederholt werden, bis die Eingabe korrekt ist. Die Rückgabe soll die Position des Buchstabens im Alphabet sein, also 0 für a, 1 für b usw... Nutze zum Einlesen des Zeichens die `Pingulib`-Methode `readChar()`.

### 2. ? Ausgabe No results

Schreibe eine Methode `static void printBoard(int[][] board)`, die das übergebene Spielfeld wie folgt ausgibt: Zuerst kommt ein " ", danach folgt die Spaltennummerierung in Buchstaben (also z.B. "abc" bei Länge 3). Danach folgen die Zeilen des Feldes, wobei ein Leerzeichen für Wasser oder ein verdecktes Schiff, ein " " für einen Fehlschuss und ein " " für einen Treffer ausgegeben wird, außerdem wird jeder Zeile die Zeilennummer vorangestellt. Betrachten Sie bei Unklarheiten die Beispielausgabe.

### 3. ? Test auf Ende No results

Schreibe eine Methode `static boolean shipsLeft(int[][] board)`, die prüft, ob das übergebene Spielfeld noch verdeckte Schiffe enthält. Enthält das übergebene Feld keine verdeckten Schiffe, soll `false` zurückgegeben werden, andernfalls `true`.

Vervollständigen Sie zuletzt die `battleship()`-Methode, die den Gesamtablauf unter Verwendung der Methoden umsetzt:

- Zuerst wird mittels `generateBoard()` das Spielfeld generiert.
- Anschließend wird "Willkommen bei Schiffe-Versenken" sowie darunter das zu Beginn "leere" Feld ausgegeben.
- Danach wird der Nutzer nach der Zahl- sowie Buchstabenkoordinate gefragt, auf die geschossen werden soll, mit den Fragen "Zahl- koordinate eingeben" respektive "Buchstaben-Koordinate eingeben".
- Das gewählte Feld wird entsprechend markiert. Wurde es allerdings bereits beschossen, soll stattdessen "Feld bereits beschossen" ausgegeben werden.
- Das neue Feld wird ausgegeben und die nächste Eingabe verlangt.
- Sobald alle Schiffe versenkt wurden, soll "Glückwunsch! Sie haben gewonnen" ausgegeben werden.

Willkommen bei Schiffe-Versenken!

+abcdefgh

1

2

3

4

5

6

7

8

Zahl-Koordinate eingeben:

> 0

Zahl-Koordinate eingeben:

> 1

Buchstaben-Koordinate eingeben:

> i

Buchstaben-Koordinate eingeben:

> c

+abcdefgh

1 x

2

3

4

5

6

7

8

Zahl-Koordinate eingeben:

> 1

Buchstaben-Koordinate eingeben:

> c

Feld bereits beschossen!

Zahl-Koordinate eingeben:

> 2

Buchstaben-Koordinate eingeben:

> c

+abcdefgh

1 x

2 x

3

4

5

6

7

8

Zahl-Koordinate eingeben:

> 3

Buchstaben-Koordinate eingeben:

> c

+abcdefgh

1 x

2 x

3 o

4

5

6

7

8

Zahl-Koordinate eingeben:

> 4

Buchstaben-Koordinate eingeben:

> a

+abcdefgh

1 x

2 x

3 o

4x

5

6

7

8

Zahl-Koordinate eingeben:

> 4

Buchstaben-Koordinate eingeben:

> b

+abcdefgh

1 x

2 x

3 o

4xx

5

6

7

8

Zahl-Koordinate eingeben:

> 4

Buchstaben-Koordinate eingeben:

> c

+abcdefgh

1 x

2 x

3 o

4xxx

5

6

```
7
8
Zahl-Koordinate eingeben:
> 4
Buchstaben-Koordinate eingeben:
> d
+abcdefgh
1 x
2 x
3 o
4xxxo
5
6
7
8
Zahl-Koordinate eingeben:
> 4
Buchstaben-Koordinate eingeben:
> e
+abcdefgh
1 x
2 x
3 o
4xxxox
5
6
7
8
Zahl-Koordinate eingeben:
> 4
Buchstaben-Koordinate eingeben:
> f
+abcdefgh
1 x
2 x
3 o
4xxxoxo
5
6
7
8
Zahl-Koordinate eingeben:
> 8
Buchstaben-Koordinate eingeben:
> h
+abcdefgh
1 x
2 x
3 o
4xxxoxo
5
6
7
8 o
Zahl-Koordinate eingeben:
> 5
Buchstaben-Koordinate eingeben:
> e
+abcdefgh
1 x
2 x
3 o
4xxxoxo
5 x
6
7
8 o
Zahl-Koordinate eingeben:
> 6
Buchstaben-Koordinate eingeben:
> e
+abcdefgh
1 x
2 x
3 o
4xxxoxo
5 x
6 x
7
8 o
Zahl-Koordinate eingeben:
> 7
Buchstaben-Koordinate eingeben:
> e
+abcdefgh
1 x
2 x
3 o
4xxxoxo
5 x
6 x
7 x
8 o
Glückwunsch! Sie haben gewonnen!
```

Die Zeilen, die mit "> " beginnen, markieren dabei die Nutzereingaben und werden nicht von dem Programm ausgegeben. Sie dienen nur der besseren Verständlichkeit.

[Lösungsvorschlag](#)