

# Aufgabe

Die Aufgabe besteht darin einen Computerspieler für das Spiel „Schiffe versenken“ zu implementieren. Dazu wurde ein Interface definiert, dessen Implementierung später von einer Kampfarena ausgeführt wird um gegen andere Computerspieler anzutreten. Dieses Interface gilt es zu implementieren.

**Abgabekriterien:**

1. Es wird eine .jar-File mit den kompilierten Klassen an das Organisationteam auf einem USB-Stick ausgeliefert. Die jar folgt der Nomenklatur:

com.nextlevel.nlb.battleship.<Teamname>.jar

Es müssen alle Abhängigkeiten enthalten sein um die Computerstrategie auszuführen zu können.

1. Die Strategie, die in der Arena ausgeführt werden muss folgenden Namen tragen und public sein:

com.nextlevel.nlb.battleship.<Teamname>.ArenaStrategy<Teamname>

1. Alle sichtbaren Klassen eines Teams müssen im Namepace

com.nextlevel.nlb.battleship.<Teamname>.\* liegen

Verletzung der Kriterien führt zu Strafpunkten im Battle oder zur Disqualifizierung.

Hinweis: Es dürfen auch mehrere Strategien implementiert werden, von denen dann nur die beste ins Rennen geschickt wird.

**Mindestanforderungen:**

1. Die BattleShipStrategy muss nach den angegeben Konventionen implementiert sein. Siehe dazu Javadoc.
2. Der Computerspieler muss mindestens ein 2-Dimensionales Feld mit beliebigen Größen innerhalb von positiven Integer handhaben können. Für die Meisterschaft genügt es die Length und Width-Koordinaten (10x10) zu berücksichtigen.

# API

Die API wird in einem jar-File an alle Teilnehmer per USB-Stick ausgeliefert. Diese heißt

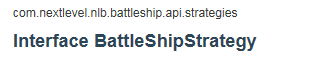
com.nextlevel.nlb.battleship.api.jar

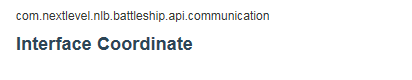
und muss in den Classpath eingebunden werden.

Enthalten sind:

* 1. Zwei Referenz-Implementierungen des Computerspielers (The Flying Doofmanns) zum Test gegen seine eigene Implementierung + eine Commandlinestrategie, die Schüsse von der Konsole entgegennimmt.
  2. Eine ausführbare Kampfarena um a) zu testen.
  3. JavDoc der API  + API selbst.

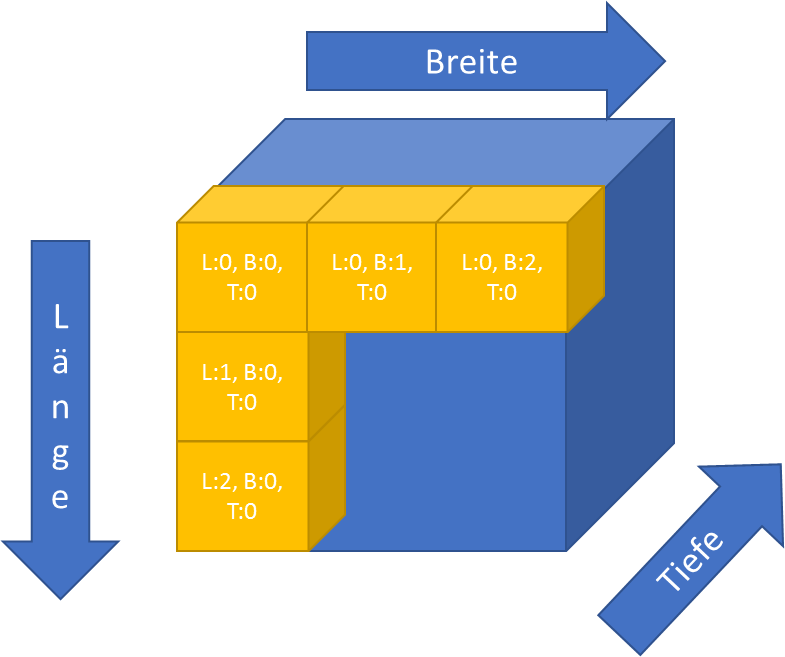
Das Interface das Implementiert werden muss lautet:

**und**



Die Beschreibung der zum implementierenden Klassen findet man im JavaDoc.

**Hinweis zum Interface Coordinate**: Die Lage der Koordinaten auf dem Spielbrett ist wie folgt:



# Testumgebung

Ausgeliefert wird auch ein Testszenario mit dem der eigene Spieler getestet werden kann, bestehend aus den Teststrategien und der TestArena.

## Teststrategien

In der API sind zwei nicht sehr spielstarke Implementierungen (Referenzen) vorhanden, die zum Test verwendet werden können. Genannt die Doofmänner:

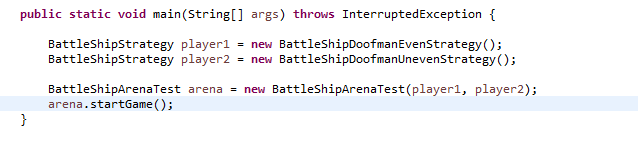
[com](eclipse-javadoc:%E2%98%82=com.nextlevel.battle.ship/src%3Ccom).[nextlevel](eclipse-javadoc:%E2%98%82=com.nextlevel.battle.ship/src%3Ccom.nextlevel).[nlb](eclipse-javadoc:%E2%98%82=com.nextlevel.battle.ship/src%3Ccom.nextlevel.nlb).[battleship](eclipse-javadoc:%E2%98%82=com.nextlevel.battle.ship/src%3Ccom.nextlevel.nlb.battleship).BattleShipDoofmanEvenStrategy

[com](eclipse-javadoc:%E2%98%82=com.nextlevel.battle.ship/src%3Ccom).[nextlevel](eclipse-javadoc:%E2%98%82=com.nextlevel.battle.ship/src%3Ccom.nextlevel).[nlb](eclipse-javadoc:%E2%98%82=com.nextlevel.battle.ship/src%3Ccom.nextlevel.nlb).[battleship](eclipse-javadoc:%E2%98%82=com.nextlevel.battle.ship/src%3Ccom.nextlevel.nlb.battleship).BattleShipDoofmanUnevenStrategy

[com](eclipse-javadoc:%E2%98%82=com.nextlevel.battle.ship/src%3Ccom).[nextlevel](eclipse-javadoc:%E2%98%82=com.nextlevel.battle.ship/src%3Ccom.nextlevel).[nlb](eclipse-javadoc:%E2%98%82=com.nextlevel.battle.ship/src%3Ccom.nextlevel.nlb).[battleship](eclipse-javadoc:%E2%98%82=com.nextlevel.battle.ship/src%3Ccom.nextlevel.nlb.battleship).BattleShipCommandLineStrategy

## TestArena

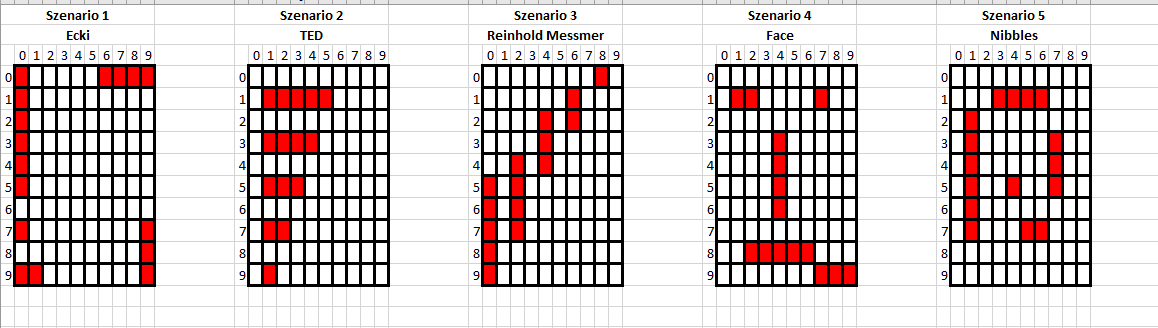
In der API ist eine TestArena enthalten, in der der Computerspieler gegen sich selbst oder die Doofmann-Strategien getestet werden kann. Aufruf erfolgt beispeilhaft so:



Es wird jeweils folgendes Testszenario gespielt werden.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Szenario** | | | | | | | | | | |
| **TEST** | | | | | | | | | | |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0 |  |  |  |  |  |  |  |  |  |  |
| 1 |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |  |  |  |  |

Ausgabe erfolgt auf der Konsole. Der Wettbewerb selber wird auf 5 verschiedenen Szenarien ausgetragen.



# Turnier

Die Bewertung erfolgt rein auf der Leistung des Computerspielers. Es wird kein Codereview o.ä. durchgeführt. Da wir in den Krieg ziehen, gilt: **Es ist alles erlaubt!** Von Java 8 bis Code-Duplizieren, von googlen bis externen Libs.

Gespielt wird die Meisterschaft (wer die meisten Punkte hat gewinnt).

Im Folgenden wird das Team, das den ersten Schuss abgeben darf „Heimteam (HT)“ genannt. Das andere Team heißt „Auswärtsteam (AT)“.

## Regeln

1. Das Spielfeld ist 2-Dimensional (nur Länge und Breite wird genutzt, Tiefe wird ignoriert)
2. Beide Teams haben die gleiche Schiffkonstellation. Die Konstellation kann aus 5 vorgegebenen Szenarien gewählt werden, die zu Beginn der Meisterschaft vorgestellt werden.
3. Spielfeldgröße ist 10 x 10 (5 Schiffe, Länge 5,4,3,2,1, Schiffe dürfen sich nicht horizontal oder vertikal berühren).

Das AT darf das Szenario wählen.

1. Es wird immer abwechselnd geschossen. Auch nach einem Treffer ist der Gegner wieder dran.

Gewinnkriterien:

* + - * + Alles Schiffe des Gegners sind versenkt.
        + Jedes Teams darf maximal so viele Schüsse abgeben, wie das Spielfeld groß ist. (Also 100 pro Team). Ist das Spiel dann noch nicht entschieden gewinnt der, der die meisten Treffer hat. Bei Gleichstand dann unentschieden.

Gewinn: 3 Punkte

Remis: jeweils 1 Punkt

Verlustkriterien:

* + - * + Die Berechnung eines Schusses darf maximal eine Sekunde dauern. Dauert diese länge verliert das Team.
        + Verursacht der Computerspieler eines Teams einen Programabbruch verliert er

1. Jeder spielt gegen jeden. Jeweils als Heim- und Auswärtsmannschaft. Auswärtstreffer zählen nicht doppelt.