

Game Battle-Ship

Planung:

0) Methoden mHeader

- beim Ausführen der Datei wird immer das ASCII Bild im header angezeigt
- "mHeader()" im package pGame

1) Methode mPlate

- die Methode bekommt zwei integer Eingaben und soll damit eine Karte bzw. Spielfeld erstellen
- die Methode mPlate enthält Zwei Werte:
 - Columns ;
 - Rows ;
- Die Eingabe wird auf einen Integer von 20 begrenzt

2) Methode mShip

- es wird eine Integer-Eingabe aufgefordert, um eine Array-Länge (Schiffstyp) auszuwählen
 - Input = (int)n ;
 - aShip = n ;
- es wird mittels Character "v" oder "h" aufgefordert den Typ des Schiffes zu wählen
 - classe hShip wird mit "h" = Horizontal aufgerufen
 - classe vShip wird mit "v" = Vertikal aufgerufen
- es wird der Start-Punkt("aKoordinaten") des Arrays ("aShip") im Spielfeld ("mPlate") mit zwei integer angefordert
- wenn der character = "v" wird vom Start-Punkt ("aKoordinaten") vertikal nach unten die Felder(Cell of Columns) belegt
- wenn der character = "h" wird vom Start-Punkt ("aKoordinaten") horizontal nach rechts die Felder (Cell of Rows) belegt
- wenn die Länge n des mShip nicht in das Spielfeld ("mPlate") passt, geben wir eine IllegalArgumentException("Das Spielfeld ist belegt! Setze die Koordinaten neu") aus
- das Array("aShip") wird auf dem Spielfeld gespeichert
- nachdem das Array ("aShip") gespeichert wurde, rufen wir die Methode ("mStatus") auf
- wenn der Start-Punkt und die Länge des Schiffes ("mShip") mit einem anderen Schiff kollidiert, geben wir eine IllegalArgumentException("Das Spielfeld ist belegt! Setze die Koordinaten neu") aus

3) Methode mStatus

- in der Methode werden die Größe des Spielfeldes ("mPlate") und die Koordinaten der Schiffes ("aShip") angezeigt
- Beispiel für den Status:

```
. . # . . # # # . .
. . # . . . . . . .
. . # . . . . . # .
. . # . . . . . # .
. . # . . . . . # .
. . # . . . . . # .
```

4) Methode mHit

- die Methode mHit enthält Zwei Werte:
 - Cell of Column ;
 - Cell of Row ;
- die Methode mHit vergleicht die Eingabe als Treffer("vHit"), mit dem Array des Schiffes ("aShip")

Nice ToKnow

- Validiere die Eingabe mit ReGex für Parameter. Sofern ein Argument ungültig ist

class (hShip)

```
package pGame;

import java.util.Map;
import java.util.TreeMap;

public class hShip {
    public static Map<String , hShip> hShipById = new TreeMap<>(Map.of(
        "S001" , new hShip ( "S001" , "#" , "Schlauchboot" ) ,
        "S002" , new hShip ( "S002" , "##" , "Angelboot" ) ,
        "S003" , new hShip ( "S003" , "###" , "Speedboot" ) ,
        "S004" , new hShip ( "S004" , "####" , "Jacht" ) ,
        "S005" , new hShip ( "S005" , "#####" , "Schiff" ) ,
        "S006" , new hShip ( "S006" , "#####" , "Fregatte" ) ,
        "S007" , new hShip ( "S007" , "#####" , "Flugzeugträger" )
    ));

    public hShip(String Key, String Größe, String Typ) {

    }
}
```

class (vShip)

```
package pGame;

import java.util.Map;
import java.util.TreeMap;

public class vShip {
    public static Map<String , vShip> vShipById = new TreeMap<>(Map.of(
        "S001" , new vShip ( "S001" , "#" , "Schlauchboot" ) ,
        "S002" , new vShip ( "S002" , "##" , "Angelboot" ) ,
        "S003" , new vShip ( "S003" , "###" , "Speedboot" ) ,
        "S004" , new vShip ( "S004" , "####" , "Jacht" ) ,
        "S005" , new vShip ( "S005" , "#####" , "Schiff" ) ,
        "S006" , new vShip ( "S006" , "#####" , "Fregatte" ) ,
        "S007" , new vShip ( "S007" , "#####" , "Flugzeugträger" )
    ));

    public vShip(String Key, String Größe, String Typ) {

    }
}
```

record rHeader

```
package pGame;

public record rHeader() {
    public static void mHeader(){
        System.out.println("");
    }
}
```