

Bimaru

Ein Bimaruspiel (Battleship oder Marinespiel) ist eine Art Schiffeversenken. Das Spielfeld ist eine quadratische Matrix. Die Schiffe darin haben alle die Breite Eins (1 Feld) und die Längen 1, 2, 3 bzw. 4 Felder. Wir bezeichnen die Schiffe daher auch Einer, Zweier, Dreier oder Vierer. Ein Schiff der Länge 3 hat also 3 Schiffsteile und nimmt 3 aneinanderliegende Quadrätchen in Anspruch. Schiffe berühren sich dabei nicht, auch nicht diagonal. Das heißt, die Schiffe sind komplett von Wasser umgeben. Beim Lösen eines Bimaru ist rechts (bzw. unten) pro Zeile (bzw. Spalte) angegeben, wie viele Schiffsteile total vorkommen. Einige Schiffsteile sind dabei manchmal schon vorgegeben. Dabei sieht man jeweils, ob es sich um ein Schiffsanfang oder -ende oder um eine Schiffsmitte handelt. In anderen Feldern ist vorgegeben, dass es sich um Wasser (also nicht um einen Schiffsteil) handelt. Weiter ist vorgegeben, wie viele Schiffe von welcher Länge (welchen Typs) total vorkommen. (Ob das hier abgebildete Bimaru lösbar ist, ist nicht Inhalt dieser Aufgabe).

Author: Philipp G. Freimann
(BBW
(Berufsbildungsschule
Winterthur)
<https://www bbw.ch>)

< img src="/var/www/html/web../uploads/eximage-1d88f6.JPG" alt="" width="242" height="344" />

Schreiben Sie eine Datenstruktur, die ein (noch ungelöstes) Bimaru wie oben speichern kann. Schreiben Sie zudem ein Programm, das die einzelnen Informationen beschreiben, bzw lesen kann. Dabei ist eine Position (feld) immer durch einen der folgenden Werte bestimmt:

String feld = ...

"unbestimmt" : Dieses Feld ist noch nicht gelöst.

"Wasser" : An dieser Position befindet sich sicher kein Schiff.

"Schiffsmitte" : hier ist ein Schiffsteil, jedoch kein Schiffsende
weder Bug noch Heck (Quadrat).

"Einzelschiff" : Schiff bestehend aus nur einer Zelle (Kreis).

"Ende_Oben" : Bug oder Heck oben
(wie im Bild 6. Zeile, 8. Spalte).

"Ende_Unten" : analog

"Ende_Links" : analog

"Ende_Rechts" : analog

"Schiffsteil" : An dieser Position befindet sich ein Teil eines Schiffes (also kein Wasser), es ist jedoch noch nicht bekannt, um welchen Typ der obigen Definitionen es sich handeln wird.

Definieren Sie die obigen nominalen Werte und schreiben Sie danach die folgenden Methoden (Dabei muss die Variable feld den entsprechenden nominalen Wert aufweisen).

setzeFeld(Bimaru b, int x, int y, String feld)

leseFeld (Bimaru b, int x, int y) : String

-> x, y = SpaltenNummer, ZeilenNummer

setzeSchiffeProZeile (Bimaru b, int y, int anzahl)

setzeSchiffeProSpalte(Bimaru b, int x, int anzahl)

leseSchiffeProZeile (Bimaru b, int y) : int

leseSchiffeProSpalte (Bimaru b, int x) : int

-> Das sind die am Rande aufgeführten Zahlen

setzeAnzahlSchiffe(Bimaru b, int typ, int anzahl)

leseAnzahlSchiffe (Bimaru b, int typ) : int

-> typ: Einer, Zweier, Dreier oder Vierer (= Länge der Schiffe)

anzahl = Wie viele Schiffe des Typs sind total vorhanden.

Die "lese-" bzw. "setze"-Methoden werden im Englischen getter- bzw. setter-Methoden genannt.

Option: Ersetzen Sie "String" durch einen Enumerationstypen.

//TODO: Evtl. als Kleinprojekt formulieren, denn nur so, macht es vielleicht gar keinen Spaß.