

Diese Aufgabe dient lediglich als weiteres Übungsmaterial zur selbständigen Bearbeitung. Es erfolgt keine Abgabe und keine Korrektur.

## Aufgabe 9.5.1 (P) Battleship

Es ist die folgende Variante des Spiels *Schiffe-Versenken* zu implementieren. Gespielt wird auf einem quadratischen Spielfeld der Seitenlänge 10, d.h. es gibt 100 einzelne Felder. Die Felder eines Spielfeldes werden über ihre Koordinaten nach rechts (0 bis 9) sowie nach oben (0 bis 9) identifiziert. Das Feld (0,0) liegt also links unten in der Ecke. Zu Beginn markiert der Computer auf dem Spielfeld *zufällig* Bereiche, die entweder aus horizontal benachbarten Feldern bestehen oder aus vertikal benachbarten Feldern. Diese bezeichnen wir als *Schiffe*. Zwei Schiffe dürfen nicht aneinander angrenzen, auch nicht diagonal. D.h. zwischen zwei Feldern, welche zu unterschiedlichen Schiffen gehören, liegt immer mindestens ein Feld, welches zu keinem Schiff gehört. Es soll von den Längen 2, 3 und 4 jeweils ein Schiff platziert werden. Zum Platzieren der Schiffe können Sie die Methode `myRandom` aus der vorgegebenen Klasse `Field` verwenden, die Zufallszahlen zwischen `low` (inklusive) und `high` (exklusive) erzeugt. Beachten Sie, dass bei einer zufälligen Platzierung keine zulässige (d.h. regelkonforme) Möglichkeit von vornherein ausgeschlossen werden darf.

Im Verlauf des Spiels rät der Spieler wiederholt *ein Feld*, auf dem er ein Schiff vermutet. Sind alle Felder eines Schiffs geraten („getroffen“), gilt das Schiff als *gesunken*. Das Spiel endet, wenn alle Schiffe gesunken sind. Das Programm gibt nach jedem Raten das Spielfeld aus und markiert darauf, welche Felder einen Treffer ergaben, welche Felder bereits Teil eines gesunkenen Schiffs sind und welche Felder bisher ohne Treffer (Schuss ins Wasser) geraten wurden.

Zum Testen des Programms soll es während des Spielverlaufs alternativ zum Raten eine Möglichkeit geben, sich die Platzierung der Schiffe anzeigen zu lassen. Zum Beispiel kann diese Ausgabe bei Eingabe von ungültigen Koordinaten (außerhalb vom Spielfeld) anstelle eines Fehlerhinweises erfolgen.

Implementieren Sie das Spiel objektorientiert! Verwalten Sie das Spielfeld, die Spielfeldkoordinaten und die Schiffe jeweils in einer eigenen Klasse.

*Hinweis:* Nennen Sie Ihr Programm `Battleship.java`. Die Praktikums-Webseite bietet eine Vorlage, die Sie nutzen können und bereits eine Spielfeldausgabe enthält. Denken Sie daran, Ihr Programm ausreichend zu kommentieren, insbesondere wenn Sie die Vorlage nicht nutzen.

*Tipps:* Verwenden Sie zum zufälligen Platzieren eines Schiffs eine Schleife, welche immer wieder zufällig eine Position erzeugt und verlassen wird, sobald die Schiffsposition erlaubt ist.

*Erweiterung:* Studierende mit fortgeschrittenen Kenntnissen können ihr Spiel auf zwei Spieler erweitern, die abwechselnd Schiffe des anderen Spielers raten.