

## 2-dimensionale Arrays: Beispiel Schiffeversenken

Ein nettes Beispiel für ein zweidimensionales Array ist das Spiel Schiffeversenken.

Sicherlich könnte man die Felder auch von 1 bis 100 durchnummerieren, aber viel praktischer ist es, wenn man, ähnlich wie beim Koordinatensystem in Mathe, x- und y-Koordinaten vergibt.

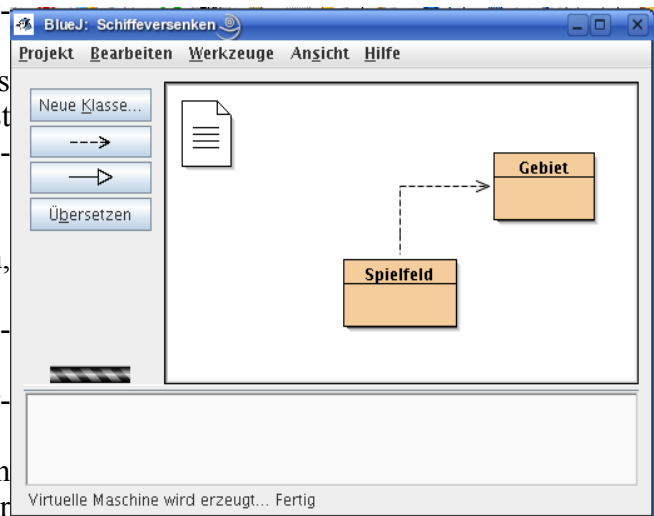
Java unterstützt dies.

Schauen wir uns aber zunächst die Klassen an, die benötigt werden:

Gebiet soll ein einzelnes Feld auf dem 10x10 Gebiete großen Spielfeldes sein.

Ein Gebiet kann ein Teil eines Schiffs beherbergen.

Ein Gebiet kann beschossen worden sein. Dann ist es markiert (denn dort kann kein Schiff mehr stehen). Aus dieser Idee resultieren folgende Klassen.



Das Spielfeld hat im Wesentlichen die Aufgabe, die Gebiete zu verwalten. Daher bekommt es nur eine Eigenschaft: *feld* ist ein zweidimensionales Array vom Typ Gebiet. Das „zweidimensional“ wird durch zwei leere eckige Klammerpaare angedeutet: Gebiet [] [].

```
public class Spielfeld
{
    // Eigenschaften
    private Gebiet[][] feld;

    // Konstruktor
    public Spielfeld()
    {
        feld = new Gebiet[10][10];
        // Anfangswerte setzen für alle 100 Felder
        for (int i=0;i<10;i++){ // senkrecht
            for (int j=0;j<10;j++){ // waagerecht
                feld[j][i]=new Gebiet();
            }
        }
        feld[1][2].setzeSchiff();
        feld[2][2].setzeSchiff();
        feld[3][2].setzeSchiff();
    }
}
```

(...)

### Erläuterungen zur Klasse Spielfeld

Das Anlegen eines zweidimensionalen Arrays geschieht ähnlich wie das eines eindimensionalen.

Hier legt man die Größe des Arrays fest: 10x10-Felder beim Schiffeversenken. Diese Konstruktion sorgt dafür, dass die Felder nicht den Anfangswert *null* haben. Dazu müssen alle Felder abgegangen werden (mit Hilfe einer verschachtelten Schleife: Erst Feld 0,0 dann 0,1 dann 0,2 .... 0,9 dann 1,0 dann 1,1 dann 1,2... dann 1,9 dann 2,0 uws. bis 9,9)

Hier wird beispielhaft ein 3er Schiff gesetzt. Dazu müssen 3 Felder einzeln angesprochen werden.

### Übung:

- 1) Analysieren Sie auf Grundlage der beiden UML-Klassendiagramme und auf Grundlage der Kenntnisse über das Schiffeversenkenspiel die Klassen Gebiet und Spielfeld. Schätzen

<b>Übung:</b>	
	Sie ein, was die einzelnen Methoden bewirken, ohne dass Sie den folgenden Quelltext anschauen. Vergleichen Sie anschließend mit dem Quelltext.
2)	Nennen Sie weitere Beispiele aus dem Alltag, bei denen man Zwei- oder Mehrdimensionale Arrays benutzen könnte.
3)	Analysieren Sie die Methode <i>schieße(int x, int y)</i> und <i>gibSpielfeldAufKonsoleAus()</i> Wiederholung: Warum kann Spielfeld nicht direkt die Eigenschaften von Gebiet verwenden?
4)	Entwerfen Sie die Methoden <i>setzeSenkrechtesSchiff()</i> und <i>setzeWaagerechtesSchiff()</i> , die in Schiff auf das Spielfeld eintragen. Überlegen Sie sich zunächst, welche Übergabeparameter benötigt werden. Entwerfen Sie einen Plan, wie man vermeiden kann, dass der Spielfeldrand überschritten wird.
5)	Schreiben Sie eine Methode <i>prüfeObGewonnen()</i> , die dann true zurückgibt, wenn auf dem gesamten Spielfeld kein Schiff mehr steht.

### Quelltexte

<pre>public class Gebiet { // Eigenschaften private boolean feldWurdeBeschossen; private boolean schiffIstAufFeld;  // Konstruktor public Gebiet() { feldWurdeBeschossen=false; schiffIstAufFeld=false; }  // Methoden public boolean beschiesseFeld() { feldWurdeBeschossen=true; if (schiffIstAufFeld) { versenkeSchiff(); return true; } else return false; }  public void setzeSchiff() { schiffIstAufFeld=true; }  public void versenkeSchiff() { schiffIstAufFeld=false; }  public boolean isFeldWurdeBeschossen() { return feldWurdeBeschossen; }  public boolean isSchiff() { return schiffIstAufFeld; } } // Ende der Klasse</pre>	<pre>public class Spielfeld { // Eigenschaften private Gebiet[][] feld;  // Konstruktor public Spielfeld() { feld = new Gebiet[10][10]; // Anfangswerte setzen f r alle 100 Felder for (int i=0;i&lt;10;i++){ // senkrecht for (int j=0;j&lt;10;j++){ //waagerecht feld[j][i]=new Gebiet(); } } feld[1][2].setzeSchiff(); feld[2][2].setzeSchiff(); feld[3][2].setzeSchiff(); }  // Methoden public String schieesse(int x, int y){ if (feld[x][y].beschiesseFeld()) return "Treffer"; else return "Daneben"; }  public void gibSpielfeldAufKonsoleAus() { System.out.println("\n 0123456789"); // Leerzeile + Beschriftung for (int i=0;i&lt;10;i++){ System.out.print(i+" "); for (int j=0;j&lt;10;j++){ if (feld[j][i].isFeldWurdeBeschossen()) System.out.print("*"); else { if (feld[j][i].isSchiff()) System.out.print("X"); else System.out.print("-"); } } System.out.println(); // Zeilenwechsel } } } // Ende der Klasse</pre>
---	--