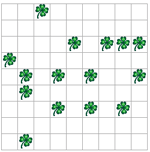
Aufgaben zu JavaKara: Arrays

# Kleeblattregen

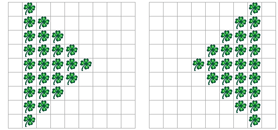
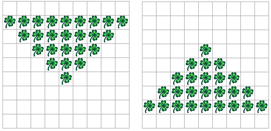


Schreiben Sie ein Programm, das fünfzehn Kleeblätter an zufälligen Koordinaten in der Welt platziert. Es sollen fünfzehn Kleeblätter platziert werden, auch wenn zufälligerweise die gleiche Koordinate zwei Mal vorkommen würde.

Sie können mit dem Befehl tools.random(n) eine Zufallszahl im Bereich 0..n (bis und mit n) erzeugen. Die Grösse der Welt können Sie mit world.getSizeX() bzw. world.getSizeY() abfragen.

Testen Sie Ihr Programm an Welten verschiedener Grössen, und variieren Sie auch die Anzahl zu regnender Kleeblätter.

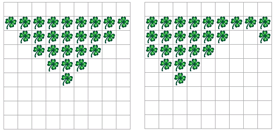
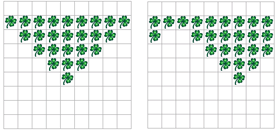
# Welt spiegeln

Schreiben Sie ein Programm, das die Welt horizontal spiegelt, und ein Programm, dass die Welt vertikal spiegelt.

Testen Sie Ihr Programm an Welten verschiedener Grössen (gerade und ungerade Anzahl Spalten bzw. Zeilen).

# Welt verschieben

Schreiben Sie ein Programm, das die Welt um eine vom Benutzer eingegebene Anzahl Spalten nach links verschiebt, und ein Programm, das die Welt um eine vom Benutzer eingegebene Anzahl Spalten nach rechts verschiebt.

Die Welt soll dabei "rundherum" verschoben werden: Spalten, die links bzw. rechts "rausfallen" würden, sollen auf der gegenüberliegenden Seite wieder erscheinen, wie in der Abbildung oben dargestellt.

Ein paar Hinweise für die Lösung:

* Sie können mit der Klasse tools einfach Benutzereingaben erfragen. Beispiel:   
  int verschiebenUm = tools.intInput("Um wieviel verschieben?");
* Verschieben Sie die Welt Zeile für Zeile. Gehen Sie dabei wie folgt vor:  
  1. Erstellen Sie einen Array mit der berechneten verschobenen Zeile. Den Array können Sie wie folgt initialisieren: boolean[] neueZeile = new boolean[world.getSizeX()]; Um das zu üben, kopieren Sie zunächst einfach eine Zeile in neueZeile rein, und ersetzen dann eine andere Zeile durch die Werte in neueZeile (also noch ohne Verschieben). So sind Sie sicher, dass das Schreiben und Lesen der Welt und des neu erstellten Arrays funktioniert.
  2. Soll die Welt zum Beispiel um 4 Felder nach links verschoben werden, so würde das neue Feld mit x=0 mit dem Inhalt von Feld x=4 ersetzt, das neue Feld x=1 mit dem Inhalt von Feld x=5, ... das neue Feld x=6 mit dem Inhalt von Feld x=10 – aber es gibt keine Spalte x=10 in der Standardwelt von Kara, die nur neun Spalten breit ist. Da "rundherum" kopiert werden soll, soll das neue Feld x=6 mit dem Inhalt von Feld x=0 ersetzt werden, das neue Feld x=1 mit dem Inhalt von Feld mit x=1, ...

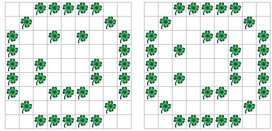
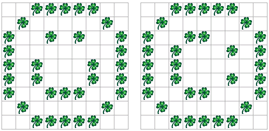
Eine elegante Methode, um nach diesem Schema die Spalten X zu berechnen, von denen her kopiert werden soll, ist den Rest der Ganzzahldivision durch die Anzahl Spalten: int kopiereVonX = (x + verschiebenUm) % world.getSizeX(); Solange x+verschiebeUm < world.getSizeX(), beträgt der Rest immer (x + verschiebenUm) (weil die Ganzzahldivision als Resultat 0 liefert). Sobald x+verschiebeUm >= world.getSizeX(), fängt der Rest wieder bei 0 an zu zählen – damit schliesst sich sozusagen der Kreis.

Natürlich können Sie diese Berechnung auch durch entsprechende Fallunterscheidungen (if) ersetzen!

* 1. Ersetzen Sie für die aktuelle Zeile alle Felder mit den Werten aus dem Array neueZeile.

Testen Sie Ihr Programm an Welten verschiedener Grössen (gerade und ungerade Anzahl Spalten bzw. Zeilen).

# Welt drehen

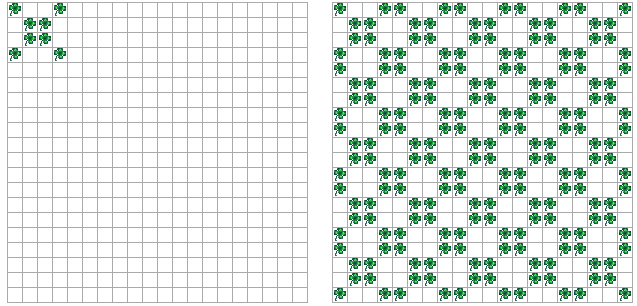
Schreiben Sie ein Programm, das die Welt um 90° nach links bzw. um 90% nach rechts dreht.

Ein paar Hinweise für die Lösung:

* Betrachten Sie zunächst nur eine einzelne Zeile. Überlegen Sie sich, wie die Koordinaten dieser Zeile transformiert werden müssen.
* Erstellen Sie einen zwei-dimensionalen Array mit der berechneten gedrehten Welt. Den Array können Sie wie folgt initialisieren: boolean[][] neueWelt = new boolean[world.getSizeX()][world.getSizeY()]; Um das zu üben, kopieren Sie zunächst einfach die bestehende Welt in neueWelt rein, und ersetzen dann die Welt durch die Werte in neueWelt (also noch ohne Drehen). So sind Sie sicher, dass das Schreiben und Lesen der Welt und des neu erstellten Arrays funktioniert.

Testen Sie Ihr Programm an Welten verschiedener Grössen (gerade und ungerade Anzahl Spalten bzw. Zeilen).

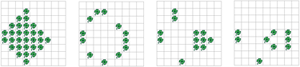
# Muster kopieren



Schreiben Sie ein Programm, das ein 4x4 Felder grosses Muster in eine 20x20 Felder grosse Welt kopiert, so dass am Ende das Muster 25x zu sehen ist!

Es gibt viele verschiedene Lösungsansätze. Eine kompakte Lösung verwendet vierfach geschachtelte Schleifen. Können Sie sich vorstellen, welche Schleife welche Funktion hat? Hinweis: Die Lösung wird dadurch vereinfacht, dass auch die Felder mit dem Orginalmuster beim Kopieren überschrieben werden.

# Game of Life



Kara möchte dem "Game of Life" zuschauen. Ausgedacht hat sich dieses "Spiel" der amerikanische Mathematiker Conway. Die Regeln sind einfach: Ein Feld in Kara's Welt ist entweder besetzt (Kleeblatt drauf) oder unbesetzt (kein Kleeblatt drauf). Die ganze Welt kann man sich als Population von Lebewesen vorstellen, aus der sich die nächste Generation nach folgenden Regeln entwickelt:

**1.** Ein leeres Feld wird in der nächsten Generation besetzt, wenn es genau drei besetzte Nachbarfelder hat. Beispiel: Das mittlere, leere Feld hat drei besetzte Nachbarfelder und wird daher "geboren":

C:\workspace\javakara\src\arrays\aufgaben\_05_gameoflife\gameoflife-dead.gif

**2.** Ein besetztes Feld bleibt auch in der nächsten Generation besetzt, wenn es zwei oder drei besetzte Nachbarfelder hat. Beispiel: Das mittlere Feld mit Kleeblatt (hell) hat drei besetzte Nachbarfelder und bleibt daher am Leben:

C:\workspace\javakara\src\arrays\aufgaben\_05_gameoflife\gameoflife-living.gif

**3.** Alle Felder, bei denen die Voraussetzungen der Regeln 1 und 2 nicht zutreffen, sind in der nächsten Generation unbesetzt. Beispiel: Das mittlere Feld mit Kleeblatt (hell) hat zu viele besetzte Nachbarfeldern und "stirbt" daher:

C:\workspace\javakara\src\arrays\aufgaben\_05_gameoflife\gameoflife-overcrowded.gif

Auch bei relativ einfachen Startwelten (mit vier oder fünf besetzten Feldern) ist es schwierig, die Entwicklung der nächsten Generationen vorauszusehen. Man muß sie wirklich durchspielen!