Übung 02: Spiel des Lebens

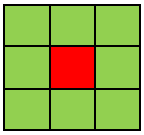
Bevor wir mit der objektorientierten Programmierung durchstarten, üben wir nochmals prozedurale Programmierung in Java. Ihre Aufgabe besteht darin, ***Conways Spiel des Lebens*** in Java zu implementieren und zu simulieren.

Die Übung setzt eine Matrix der Dimension 10 x 10 voraus, siehe rechte Abbildung. Zu Beginn (*1. Generation*) ist jede Zelle mit einer Wahrscheinlichkeit von 50% ***tot*** bzw. ***lebendig***. Tote Zellen werden durch „**0**" bezeichnet, lebendige Zellen durch ein „**X**". Aus der *n*. Generation berechnet man die Nachfahrengeneration *n+1* anhand von 4 festen **Regeln**:

1. Lebende Zellen mit weniger als 2 lebenden Nachbarn sterben in der Folgegeneration an Einsamkeit.
2. Lebende Zellen mit mehr als 3 lebenden Nachbarn sterben in der Folgegeneration an Überbevölkerung.
3. Lebende Zellen mit 2 oder 3 lebenden Nachbarn bleibt in der Folgegeneration am Leben.
4. Eine tote Zelle mit genau 3 lebenden Nachbarn wird in der nächsten Generation neugeboren.

O	X	O	X	X	X	O	O	X	O
X	X	O	O	O	O	X	O	X	X
O	O	X	X	O	X	O	O	X	X
X	O	X	O	O	X	X	O	O	X
O	X	O	X	X	X	O	O	X	O
X	O	X	O	O	O	X	X	O	X
O	X	O	X	O	X	O	O	X	X
O	O	X	X	O	X	O	X	O	O
X	X	O	O	O	O	X	O	X	X
0	X	X	X	X	X	O	O	X	X

Ausgehend von einer Anfangsgeneration kann man ***iterativ*** die Folgegenerationen berechnen. Jede Zelle ist dabei von genau 8 ***Nachbarn*** umgeben. Die rechte Abbildung zeigt die 8 (grünen) Nachbarn der roten Zelle. Eine ***Ausnahme*** bilden Zellen am linken, rechten, oberen oder unteren Rand.



Man kann sich mit dem Modell des „Spiel des Lebens" aus verschiedenen Sichtweisen beschäftigen, z.B. unter biologischen Aspekten (Mikrokosmos) oder unter ökonomischen Aspekten (Modell des Computerhandels bei Finanzmärkten). Eine genauere Beschreibung dieses „Spiels" sowie dessen Anwendungen finden Sie in der Wikipedia:

https://de.wikipedia.org/wiki/Conways_Spiel_des_Lebens

Verwenden Sie das in der Community vorgegebene Programmgerüst.

Schritt 1: Initialisierung

Initialisieren Sie zu Beginn der `main`-Methode das vorgegebene 2-dimensionale Array `cells`. Es soll gewürfelt werden, ob eine Zelle tot oder lebendig ist. Die Wahrscheinlichkeit für eine tote Zelle sei 0,5.

Hinweise: Ggfs. Internetrecherche wie man ein 2-dimensionales Array in Java einsetzt. Verwenden Sie den Aufruf `Math.random()`, der eine `double` Zufallszahl aus dem Bereich [0;1[zurückliefert: https://docs.oracle.com/javase/8/docs/api/java/lang/Math.html#random--

Schritt 2 `private static void printCells(boolean[][] c)`

Implementieren Sie eine Methode, die als Parameter ein 2-dimensionales Array c bekommt und auf der Konsole die aktuelle Belegung des Arrays in ähnlicher Form ausgibt wie in der ersten Abbildung (X: Lebende Zelle, O: tote Zelle).

Schritt 3 `private static int countLivingNeighbors(boolean[][] c, int x, int y)`

Implementieren Sie eine Methode, die zählt wie viele lebende Nachbarn eine bestimmte Zelle `c[x][y]` hat, die die *x+1*.te Zeile und die *y+1*. Spalte repräsentiert.

(umblättern)

Schritt 4: Berechnung der Nachfolgegeneration

```
private static boolean[][] computeNextGenCells(boolean[][] cells)
```

Implementieren Sie eine Methode, die unter Beachtung der 4 Regeln, die Nachfolgegeneration berechnet und als 2-dimensionales Array zurückliefert.

Lassen Sie Ihr Programm laufen, setzen Sie ggfs. Debugging ein!

Beobachten Sie ob sich bei verschiedenen Ausgangskonfigurationen ggfs. typische Objektmuster ergeben, z.B. Objekte die ganz verschwinden, die unverändert bleiben, die sich periodisch verändern oder ob vielleicht sogar die Population komplett ausstirbt. (siehe auch Wikipedia).