

Lehrziele:

- Methoden
- UnitTest
- mehrdimensionalen Blockarrays

Aufgabenstellung

GameOfLife

Das Spiel des Lebens (Game of Life) wurde vom Mathematiker Conway erfunden und basiert auf folgender Idee:

Die Spielwelt besteht aus einer Matrix von Zellen, die entweder leben können oder tot sind. Jede Zelle hat 8 Nachbarn. Randzellen haben die Zellen des gegenüberliegenden Randes als Nachbarn. Aus der momentanen Zellenpopulation kann man die Population in der nächsten Generation durch folgende Regeln berechnen:

- Hat eine tote Zelle genau 3 lebende Nachbarn, erwacht sie zum Leben.
- Hat eine lebende Zelle 2 oder 3 lebende Nachbarn, bleibt sie am Leben.
- Alle anderen lebenden Zellen sterben.

Schreibe ein Programm, das eine Zahl n sowie eine $n \times n$ – Matrix field aus Nullen und Einsen liest. Die Matrix verkörpert die Initialpopulation: Eine Eins bedeutet eine lebende Zelle, eine Null bedeutet eine tote Zelle. Implementiere eine Methode

```
public static bool[,] CalculateNextGeneration(bool[,] world)
```

die aus einer gegebenen Population `world` die nächste Generation berechnet. Das Programm soll nacheinander beliebig viele Generationen erzeugen und am Bildschirm ausgeben können. Siehe Kapitel „Bildschirmausgabe“.

Vereinfachung

Ein Array mit unendlicher Kantenlänge ist relativ schwer zu implementieren – beschränken Sie sich bei Ihrer Arbeit auf einen Array, der sich gerade noch darstellen lässt.

Hintergrundinfo

Die Regeln zum Game of Life wurden 1970 vom Mathematiker J. H. Conway entwickelt. Er wies nach, dass das Verhalten „chaotisch“ und „unvorhersagbar“ (im mathematischen Sinne) ist. Im WWW finden Sie dbzgl. mit Sicherheit mehr.

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

In dem bereitgestellten Programm-Template sind `UnitTests` vorhanden. Implementieren Sie die Methoden so, dass **alle** Unittests erfolgreich ausgeführt werden können. Änderungen an den Unittests sind nicht erlaubt.

Zusätzlich zur oben beschriebenen Methode muss eine „Hilfsmethode“ implementiert werden – diese wird von den Unittest ebenfalls überprüft:

```
public static int CountNeighbours(bool[,] world, int row, int col)
```

BildschirmAusgabe

Game of Life

=====

Bei einer negativen Größe wird eine zufällige Welt mit der positiven Größe erstellt.

Größe des Spielfelds: [-100..100]: 10

```
1. Row: 0000001010
2. Row: 0000001111
3. Row: 1111111111
4. Row: 0001010101
5. Row: 1111001111
6. Row: 1100000000
7. Row: 1110000001
8. Row: 1111000011
9. Row: 0000000011
10. Row: 1010101010
```

Ausgangswelt:

```
      X X
     XXXX
XXXXXXXXXX
   X X X X
  XXXX  XXXX
 XX
 XXX      X
XXXX     XX
      XX
X X X X X
```

Start mit Eingabetaste ...

Nach dem Drücken der Eingabetaste

Welt nach Runde: 1

```
      X
     XXXX
     XXX
      XX XX
      X  X
      X  X
      X
      X  X
```

Eingabetaste für nächste Runde oder x für Ende:

Nach dem Drücken der Eingabetaste

Welt nach Runde: 2

```
     XXXX
    X  X
    X  X
     X
    XX XX
    XX XXX
    XXX
```

Eingabetaste für nächste Runde oder x für Ende: