Game of life

Schau Dir das folgende Video an:

https://www.youtube.com/watch?v=CgOcEZinQ2I

Es beschreibt das so genannte Game of Life, das wir in Greenfoot programmieren wollen. Die Regeln hierfür sind wie folgt:

- Eine tote Zelle mit genau drei lebenden Nachbarn wird in der Folgegeneration neu geboren.

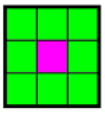
rot: Tote Zelle, die in der nächsten Generation geboren wird

grün: Lebende Nachbarn der Zelle

- Lebende Zellen mit weniger als zwei lebenden Nachbarn sterben in der Folgegeneration an Einsamkeit.

- Eine lebende Zelle mit zwei oder drei lebenden Nachbarn bleibt in der Folgegeneration am Leben.

- Lebende Zellen mit mehr als drei lebenden Nachbarn sterben in der Folgegeneration an Überbevölkerung.



magenta: Lebende Zelle, die betrachtet wird

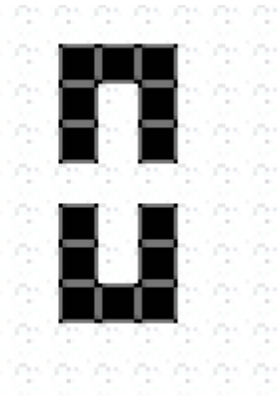
grün: Lebende Nachbarn der Zelle

Du findest das vorbereitete „leere“ Szenario vor, in dem Du die Klasse **MyCell** so programmieren sollst, dass die Regeln korrekt umgesetzt werden! Du kannst dafür folgende Methoden nutzen:

```
int gibAnzahlLebenderNachbarn()
boolean gibIstLebendig()
void setzeLebendig()
void setzeTot()
```

1. Setze die Regeln in der act()-Methode entsprechend um.

Um zu testen, ob das bereits funktioniert, erstelle folgende Konfiguration, am besten in der Mitte:



Nach 54 Muster müsste es eine leere Welt erzeugen, wenn Du alles richtig gemacht hast!

2. Überlege, warum Dein GoL evtl. nicht funktioniert. Bedenke: In Greenfoot werden alle Zellen nacheinander bearbeitet und für jedes MyCell-Objekt wird nacheinander die act()-Methode aufgerufen. Um das GoL also richtig zu programmieren, muss sich jede Zelle in einem Schritt merken, wie der Folgezustand aussieht und erst im nächsten act() müssen dann alle Zellen ihren Zustand ändern. Dann folgt wieder die Auswertung, dann die Änderung, dann die Auswertung, dann die Änderung usw. usw.

3. Ändere Dein Szenario so ab, dass dieses funktioniert.