

Relexion

Prgrammierung eines Game of Life Spiels

Die Aufgabe, ein Game of Life Spiel zu programmieren wurde erfüllt. Die großen Distanzen zwischen den Projektmitgliedern stellte eine besondere Herausforderung dar. Gelöst wurde dieses Problem durch die Verwendung von GitHub als Versioningtool sowie durch Skypekonferenzen und andere Internetkommunikationsmittel. Wobei der Parser also das Einlesen der Anfangsbedingungen aus der einer Textdatei und die dazugehörige Fehlerbehandlung zu meinen Aufgaben zählten.

Werden die Spielregeln variiert und z.B. ein neuer Zelltyp eingeführt, so lässt sich das im Zusammenspiel mit den Rules, im Parser leicht durch ein weiteres switch-case realisieren. Soll das Spielfeld, also die Gesamtheit aller Zellen, auch andere Formen als das Rechteck zulassen, so können die Überprüfungen auf die Rechteckform weggelassen werden.

Zur Verbesserung des Parsers kann am besten in der Fehlerbehandlung angesetzt werden. Man könnte überprüfen welches Zeichen genau ungültig ist oder in welcher Zeile der Formatfehler für das Spielfeld auftritt. Den Parser für jede Eventualität abzusichern gestaltet sich sehr schwierig und zeitaufwändig. Die nichtkonventionale Schreibweise der Konstanten wurde auf Grund der Lesbarkeit gewählt. Diese Variante führt durch die seltene Verwendung des Namens nicht zu Verwirrungen.

Durch die Aufgabe des Parsers, greift interagiert er wenig mit anderen Klassen. So ist er kaum betroffen von Änderungen im weiteren Programmablauf und gibt letztendlich nur vor, dass die Generationen jeweils als zweidimensionale Arrays weitergegeben werden. Selbst bei einer Änderung dieses Vorgehens muss der Parser nicht umgeschrieben sondern lediglich erweitert werden. Wenn zum Beispiel die für die Generationen ein extra Datentyp angelegt wird. So müsste der Parser eine weitere Methode enthalten um den Array an Cells in ein Objekt der Klasse Generation umzuwandeln. Dieses Vorgehen würde sich auszahlen, da so das Handling mit den Readern einfacher von Hand geht.

Zusammenfassend ist ein flexibler Parser mit den Umständen entsprechend niedriger Fehlerbehandlung entstanden.