

# 3. Praktikum

Jörn Loviscach

Versionsstand: 12. Juni 2021, 23:41



This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Germany License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/de/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

Es soll ein minimales Tabellenkalkulationsprogramm geschrieben werden, in das man ganze Zahlen sowie Summen-Formeln nach dem Muster  $=B1+C2+A3$  eingeben kann. In der aktuell angewählten Zelle ist die Formel sichtbar und editierbar. Von den Formeln in allen anderen Zellen sieht man nur die Ergebnisse. Ändert man einen Wert oder eine Formel, ändern sich die angezeigten Werte aller Zellen, die direkt oder indirekt darauf basieren.

	A	B	C
1	0	7	0
2	0	0	23
3	123	=B1+C2+A3	0
4	0	0	0

Damit die Oberfläche des Programms eine Tabelle wird, kann man dem Grid im C#-Code oder in der XAML-Datei `RowDefinitions` und `ColumnDefinitions` geben. Die Elemente der Tabelle (Labels für die Beschriftung und `TextBox`en für die Zellen) erhalten dann noch eine Angabe, in welcher `Row` und welcher `Column` sie stehen sollen.

Ob eine `TextBox` für die Eingabe angewählt ist, erfährt man mit Hilfe von deren Ereignissen `GotFocus` und `LostFocus`. Bei `GotFocus` muss auf die editierbare Formel umgeschaltet werden, bei `LostFocus` muss der Wert angezeigt werden.

Um die Zellen upzudaten, auf die sich eine Änderung auswirkt, soll jede Zelle wissen, welche Zellen direkt von ihr abhängen. Dann kann sie eine Kaskade von Änderungen anstoßen: Sie benachrichtigt die Zellen, die direkt von ihr abhängen; diese benachrichtigen wiederum die Zellen, die direkt von ihnen abhängen usw.

*Diese Anleitung ist bewusst nicht vollständig – als Übung dafür, die automatische Code-Vervollständigung zu nutzen und die richtigen Fragen an die Suchmaschinen zu stellen.*