

1. Praktikum

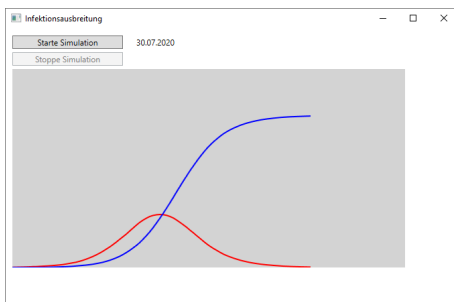
Jörn Loviscach

Versionsstand: 21. Mai 2020, 14:37



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.

Die Ausbreitung einer Infektionskrankheit soll simuliert werden: Wie entwickeln sich die Zahlen von Infizierten und Genesenen über die Zeit?



Die Infektion soll nicht tödlich sein und soll zu einer dauerhaften Immunität führen. Infizierte sollen sofort ab der Infektion und bis zur Genesung ansteckend sein. Die Zeit zwischen Infektion und Genesung soll zufällig 7 bis 13 Tage betragen (gleichverteilt).

Es soll insgesamt 100.000 Personen geben, davon 100 am ersten Tag Infizierte. Jede Person hat pro Tag zehn Treffen mit anderen Personen (ggf. mit einigen mehrfach). Bei jedem Treffen

wird eine Infektion mit einer Wahrscheinlichkeit von zwei Prozent übertragen.

Legen Sie zur Modellierung eine Klasse `Person` an, die speichern kann, ob die Person infiziert worden ist, und, wenn ja, wann das geschehen ist und zu welchem Zeitpunkt die Genesung eintreten wird. Geben Sie dieser Klasse außerdem diese drei Methoden und füllen Sie diese sinnvoll mit Code (also einer Implementierung):

```
public void ErhalteErreger(DateTime wann)
public bool IstInfiziert(DateTime wann)
public bool KannInfizieren(DateTime wann)
```

Außerdem benötigt die Klasse einen Pseudo-Zufallszahlen-Generator, um die Dauer bis zur Genesung würfeln zu können:

```
static Random würfel = new Random();
```

Legen Sie in der XAML-Datei für das Fenster zwei `Buttons` an. Der eine der beiden `Buttons` soll die Simulation starten, der andere soll sie stoppen (Events = Ereignisse!). Legen Sie außerdem für die Ausgabe des aktuellen Simulationsdatums ein `Label` an und als Zeichenfläche eine `Canvas`, in letzterer für die Kurven zwei `Polylines`. Stellen Sie für diese `Stroke` auf eine passende Farbe

und `StrokeThickness` zum Beispiel auf 2. Geben Sie beiden Kurven Namen; dann können Sie aus dem Programm nach diesem Muster Punkte anhängen:

```
kurvenname.Points.Add(new Point(23.0, 42.0));
```

Damit die Simulation Schritt für Schritt statt auf einen Schlag ablaufen kann, legen Sie in der Klasse für das Fenster einen `DispatcherTimer` als Attribut an:

```
DispatcherTimer timer = new DispatcherTimer();
```

Beachten Sie das Glühlampen-Popup für das nötige `using`.

Der Timer ruft für jeden Schritt immer wieder dieselbe Methode auf. Legen Sie diese an:

```
void Animiere(object sender, EventArgs e)
```

Stellen Sie im Konstruktor der Fensterklasse den Timer ein:

```
timer.Interval = TimeSpan.FromMilliseconds(100);  
timer.Tick += Animiere; // Methodename, kein Aufruf!
```

Füllen Sie nun die Click-Methoden für die beiden Buttons und die Methode `Animiere` passend mit Code.

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

Mögliche Erweiterungen:

- Beschriften Sie das Diagramm – insbesondere mit Achsen und Einheiten.
- Erlauben Sie eine Eingabe der Simulationsparameter, zum Beispiel der Infektionswahrscheinlichkeit.
- Sehen Sie eine Zeitverzögerung zwischen der eigenen Infektion und der Infektiösität für andere vor.

Dieses Modell ist – wie viele andere Modelle in der aktuellen Diskussion – extrem simpel. Insbesondere berücksichtigt es nicht, dass man einige Menschen sehr häufig trifft (Familie, Schule, Arbeit usw.), andere aber extrem selten. Es ist für keine belastbaren Vorhersagen geeignet!
