

# Praktikum 12./13. April 2011

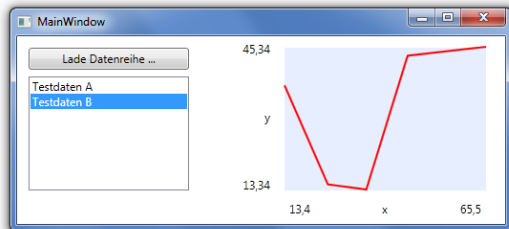
Jörn Loviscach

Versionsstand: 12. April 2011, 20:03



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.

In diesem Praktikum soll ein Plotter für  $xy$ -Datenreihen programmiert werden, die aus Textdateien gelesen werden.



Die grafische Oberfläche enthält eine `ListBox`, in der die Namen der Datenreihen aufgelistet sind. Man kann mit einem `Button` eine neue Datenreihe laden und zu der Liste hinzufügen. Der Inhalt der jeweils in der Liste angewählten Datenreihe erscheint als Diagramm in einer `Canvas`, die mit mehreren `Labels` beschriftet ist. An den beiden Enden jeder Achse stehen die Maximal- bzw. Minimalwerte der aktuellen Datenreihe.

Die Datenreihen sollen als Textdateien nach folgendem Muster gelesen werden:

```
Testreihe C
12, 3; 45, 6
78, 90; 12, 34
155, 6; -7, 89
```

Die  $x$ -Werte sind dabei aufsteigend. Erstellen Sie eine Handvoll entsprechender Testdateien zum Beispiel mit Notepad.

Erzeugen Sie ein neues Projekt (C#, WPF) und fügen Sie per Mausklick rechts auf das Projekt (nicht die Projektmappe!) im „Projektmappen-Explorer“ eine neue Klasse namens `DataRow` hinzu.<sup>c1</sup> Jede Instanz davon soll eine Datenreihe darstellen. Geben Sie der Klasse drei `private` Attribute: ein Zeichenkette für den Titel, und zwei `double`-Arrays für die  $x$ - bzw.  $y$ -Werte.

<sup>c1</sup>jl: Der Name `DataRow` wäre besser.

Geben Sie dieser Klasse einen öffentlichen Konstruktor, der einen Dateipfad entgegennimmt: `public DataRow(string path)`. Der Konstruktor soll mit

`System.IO.File.ReadAllLines` die Datei einlesen, den Titel aus der ersten Zeile der Datei lesen und die  $xy$ -Daten aus den übrigen Zeilen der Datei lesen. Verwenden Sie dazu die Methode `Split` der `string`-Klasse und die Methode `double.Parse`.<sup>c2</sup> Ignorieren Sie (noch) die Fehlerbehandlung.

<sup>c2</sup>jl: double statt int

Geben Sie dieser Klasse eine öffentliche Funktion `ToString()`, welche den Titel zurückliefert. Diese Funktion muss mit dem Zusatz `override` versehen werden, weil sie eine geerbte Funktion überschreibt (Details demnächst in der Vorlesung).

Fügen Sie zur Hauptfenster-Klasse einen Datei-Laden-Dialog vom Typ `Microsoft.Win32.OpenFileDialog` als `private` Attribut hinzu. Wenn der Laden-Button geklickt wird, soll dieser Datei-Laden-Dialog mit seiner `ShowDialog`-Methode auf den Schirm gebracht werden. Danach können Sie mit seiner Property `FileName` den ausgewählten Pfad erhalten. Rufen Sie den Konstruktor von `DataRow` damit auf und fügen Sie die neue `DataRow` mit `Items.Add` der `ListBox` hinzu.

Ergänzen Sie die Klasse `DataRow` um vier öffentliche, nur lesbare Properties, mit denen man das Minimum bzw. Maximum der  $x$ -Werte bzw.  $y$ -Werte abfragen kann.

Nun müssen noch die Daten der aktuell angewählten `DataRow` angezeigt werden. Das übernimmt eine `private` Methode der Hauptfenster-Klasse, die sie nun schreiben. Rufen Sie diese Methode auf,<sup>c1</sup> wenn sich die Auswahl der `ListBox` geändert hat (Ereignis `SelectionChanged`). Wenn eine neue Datenreihe geladen worden ist, soll die Auswahl der Liste auf diese Datenreihe gestellt werden.<sup>c2</sup>

<sup>c1</sup>jl: an zwei Stellen auf: wenn neue Datenreihe geladen worden und

<sup>c2</sup> text added by jl

Diese Methode holt mit `(DataRow) ... .SelectedItem` einen Verweis auf die aktuell in der `ListBox` angewählte `DataRow`. Dann setzt sie die Labels auf die Minimum- und Maximum-Werte dieser `DataRow`.

Es bleibt noch die Kurve zu zeichnen. Um Tipparbeit zu sparen, ergänzen Sie am Anfang der Datei der Klasse `DataRow` drei `using`-Anweisungen für `System.Windows.Shapes`, `System.Windows.Media`, `System.Windows`. Geben Sie der Klasse eine öffentliche Methode `Polyline BuildPolyline(double width, double height)`, welche die Kurve in einer vorgegebenen Größe liefert:  $x$  von 0 bis  $width-1.0$  und  $y$  entsprechend. Erzeugen Sie in dieser Methode eine neue `Polyline`, stellen Sie deren `Stroke` und `StrokeThickness` ein und fügen Sie deren `Points` mit `Add(new Point(..., ...))` Datenpunkte mit hinzu – natürlich mit passend auf `width` und `height` umgerechneten Koordinaten.

Die Methode des Hauptfensters, welche die Daten einträgt, kann nun diese `Polyline` in der Größe der `Canvas` holen und mit der Methode `Children.Add` der `Canvas` hinzufügen. Weil dort schon eine alte Kurve stehen könnte, sollte erst alles Alte mit der Methode `Children.Clear` der `Canvas` beseitigt werden.