

Windowsprogrammierung – WiSe17

Übung 8

Aufgabe 8.1 Grafik in XAML

Schreiben Sie ein WPF-Programm, das die Grafik in Abbildung 1 möglichst originalgetreu anzeigt. Verwenden zur Erstellung der Grafik ausschließlich XAML-Code. Folgende Maße werden vorgegeben:

- Bei der hellgrauen Hintergrundfläche handelt es sich um ein *Canvas* mit den festen Maßen (500,500). Verwenden Sie dieses Panel als Elternelement für alle weiteren graphischen Elemente.
- Das blaue Rechteck besitzt eine Breite von 300 und eine Höhe von 200.
- Das grüne Objekt besitzt eine Höhe von 100.
- Die Breite und Höhe des Kreuzes sowie die Diagonalen des Quadrates haben alle den Wert 50.
- Die Strichstärke aller Linien beträgt 3.

Tipp: Die Struktur in Lila kann über eine XOR-Verknüpfung zweier Ellipsen umgesetzt werden.

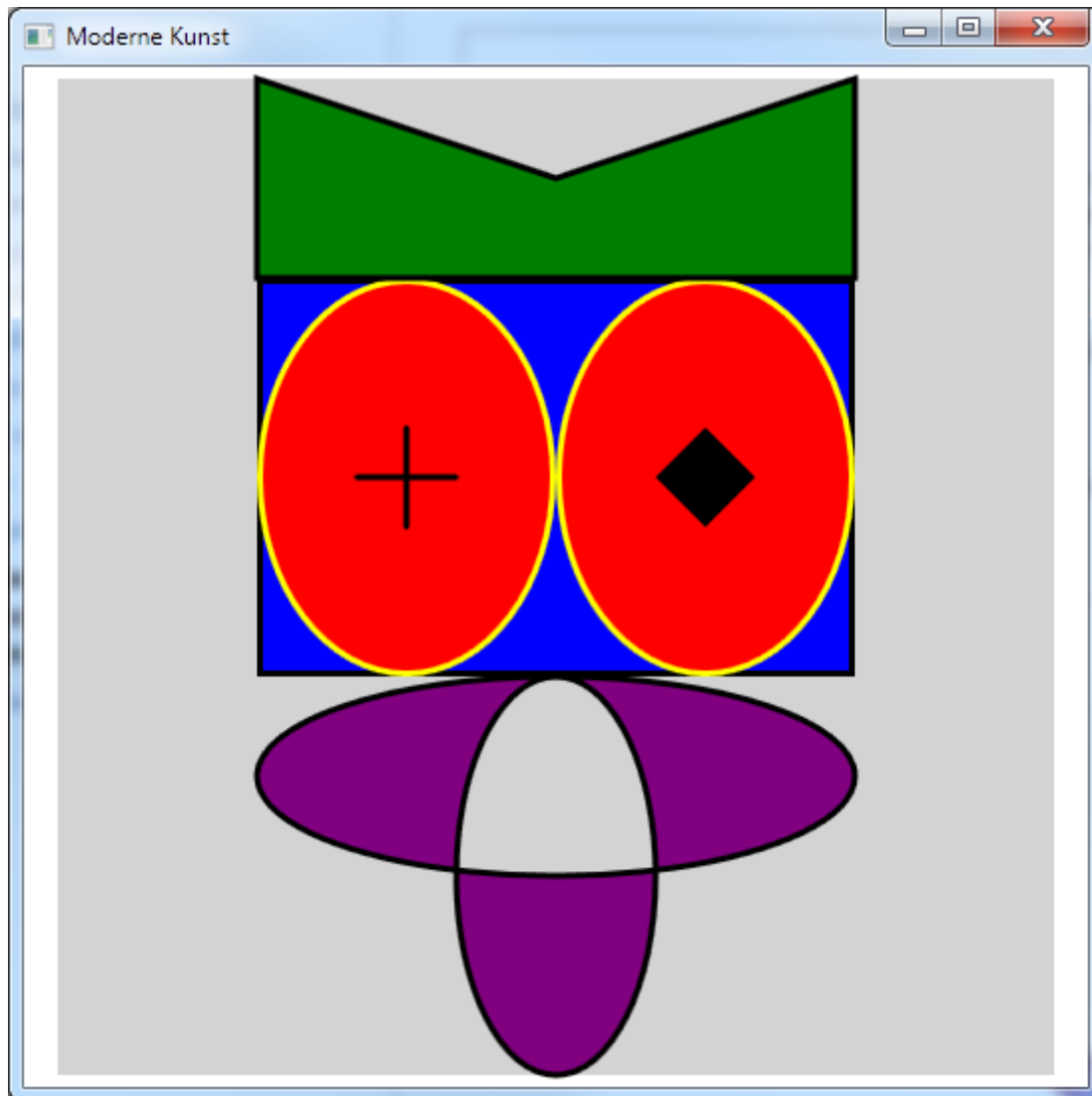


Abbildung 1

Aufgabe 8.2 Grafik in C#

Schreiben Sie eine Klasse **Plot** zur Darstellung eines Funktionsgraphen mitsamt zugehörigem Koordinatensystem. Lassen Sie diese Klasse von *Canvas* erben, um auf die Basisfunktionalität eines GUI-Elementes zugreifen zu können. Die Klasse soll mindestens über die folgenden Felder und zugehörigen öffentlichen Properties mit Schreib- und Lesezugriff verfügen:

- **XMin, XMax, YMin, YMax** vom Typ *double* (zur Einstellung des Wertebereichs des Koordinatensystems)
- **DeltaX** vom Typ *double* (Erläuterung siehe unten)
- **Funktion** vom Typ *Func<double, double>* (bei *Func<double, double>* handelt es sich um einen vordefinierten Delegat-Typ, der mit allen Methoden kompatibel ist, die einen *double*-Wert als Argument annehmen und einen *double*-Wert zurückgeben. Eine

Instanz vom Typ *Func<double, double>* kann also wie eine reelle Funktion verwendet werden)

Der Funktionsgraph selbst soll in Form eines linearen Splines dargestellt werden, dessen Stützstellen einen Abstand von **DeltaX** zueinander haben. Die niedrigste Stützstelle soll **XMin** betragen, die höchste Stützstelle soll kleiner oder gleich **XMax** sein.

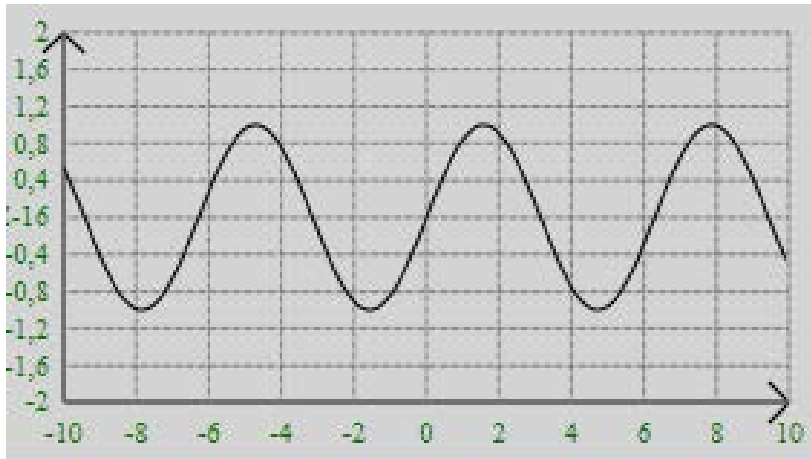


Abbildung 2

Die resultierende Grafik soll ungefähr so aussehen, wie in Abbildung 2 dargestellt. Die Auswahl von Farben, Strichstärke und -typ, Schriftart usw. bleibt natürlich Ihren eigenen ästhetischen Ansprüchen überlassen. Im besten Falle bieten Sie eine Reihe von Properties an, damit der Anwender Ihrer Klasse diese Parameter bei Bedarf selbst festlegen kann. Beachten Sie folgendes:

- Der eigentliche Funktionsgraph erstreckt sich nicht über die gesamte Zeichenfläche. Sie sollten also stets differenzieren zwischen den Grenzen der Zeichenfläche und den Grenzen des Bereichs für den Funktionsgraphen.
- Pro Achse soll es 11 Stellen für die Beschriftung und die Rasterlinien geben. Zwei dieser Stellen sind jeweils bei **XMin** und **XMax** bzw. bei **YMin** und **YMax**, die restlichen 9 liegen äquidistant dazwischen.
- Vergessen Sie nicht, die Pfeilspitzen der Achsen zu zeichnen!
- Der Plot soll beliebig skalierbar sein.
- Zum Zeichnen überschreiben Sie die Methode *OnRender* der Klasse *Panel*, von der auch die Klasse *Canvas* erbt und nutzen deren Übergabeobjekt vom Typ *DrawingContext*, um die gewünschten Formen direkt zu zeichnen.

Sie können die folgende Methode benutzen, um ein Array aus Punkten als linearen Spline zu visualisieren:

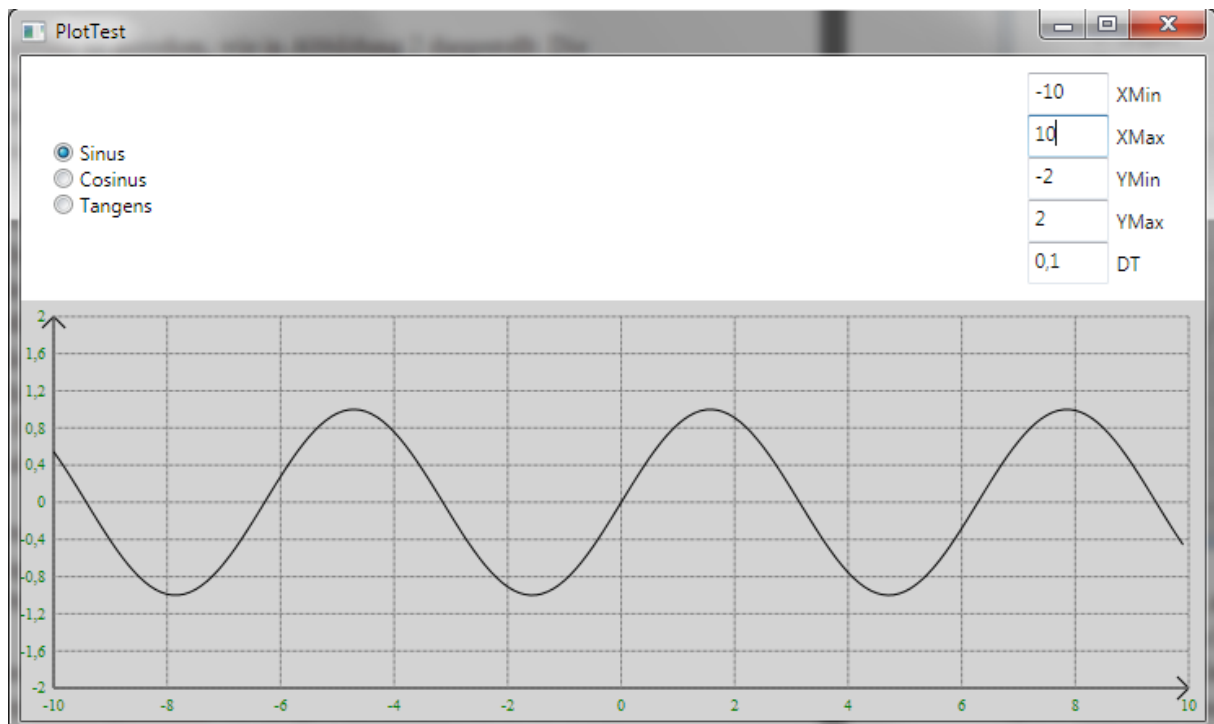
```
void DrawPolyline(DrawingContext dc, Pen pen, Point[] points)
{
    PathGeometry geometry = new PathGeometry();
    PathFigure figure = new PathFigure();
```

```

figure.StartPoint = points[0];
figure.Segments.Add(new PolyLineSegment(points, true));
geometry.Figures.Add(figure);
dc.DrawGeometry(null, pen, geometry);
}

```

Schreiben ein WPF-Testprogramm für Ihre **Plot**-Klasse, das über das folgende Aussehen verfügt:



Verwenden Sie zur Anordnung der einzelnen Elemente ein Grid, das aus 2 Zeilen und 2 Spalten besteht. Die Höhe der ersten Zeile soll sich dabei an ihrem Inhalt orientieren, die Höhe der zweiten Zeile soll den restlichen Platz im Fenster einnehmen. Die Breite der 1. Spalte und die Breite der 2. Spalte sollen sich die verfügbare Fensterbreite gleichmäßig aufteilen. Der Funktionsplot soll sich über die beiden unteren Zellen hinausspannen. In der Zelle links oben befindet sich ein Stackpanel, das wiederum einige Radio-Buttons zur Auswahl verschiedener Funktionen beinhaltet. In der Zelle oben rechts befindet sich ebenfalls ein Stackpanel, das wiederum aus horizontal ausgerichteten Stackpanels besteht, die ihrerseits jeweils eine Textbox und ein Label beinhalten. Bei einem Klick auf einen Radiobutton oder nach der Eingabe des Textes in eine Textbox (Event: LostFocus) soll der dargestellte Inhalt im Funktionsplot entsprechend angepasst werden.

Hiweise zur Darstellung des Splines

- Benutzen sie die Math Klasse um die Stützstellenwerte der trigonometrischen Funktionen zu berechnen.
- Die Anzahl der Stützstellen ist über $(XMax - XMin) / \Delta x$ zu berechnen.

- Benutzen Sie eine *Rect*-Struktur, in der Sie die Koordinaten sowie die Breite und Höhe des Zeichenbereichs speichern und rechnen Sie die Stützstellen des Splines als Punkte der *Rect*-Struktur um.
- Die Funktion *InvalidateVisual()* veranlasst das Neuzeichnen und den Aufruf der *OnRender*-Methode