

```
using System;
using System.Collections.Generic;
using System.Windows.Shapes;

namespace Plotmaker
{
    public partial class MainWindow : Window
    {
        /*
         * Maturitní projekt
         *
         * Informatika a výpočetní technika
         * (dokumentace)
         *
         *
         * Václav Pelíšek
         * Oktáva B
         * 2013/2014
         *
         *
         * Gymnázium Mikulášské náměstí 23
        */
    }
}
```

Obsah

1. Popis aplikace
2. Popis zdrojového kódu
 - 2.1 Parser
 - 2.2 Vytvoření kolekce bodů
 - 2.3 Vykreslení grafu
3. Ukázka

Popis aplikace:

Aplikace slouží k vykreslení grafu jakékoliv funkce. Používá k tomu vlastní evaulátor výrazů a C# WPF prostředí k vykreslování.

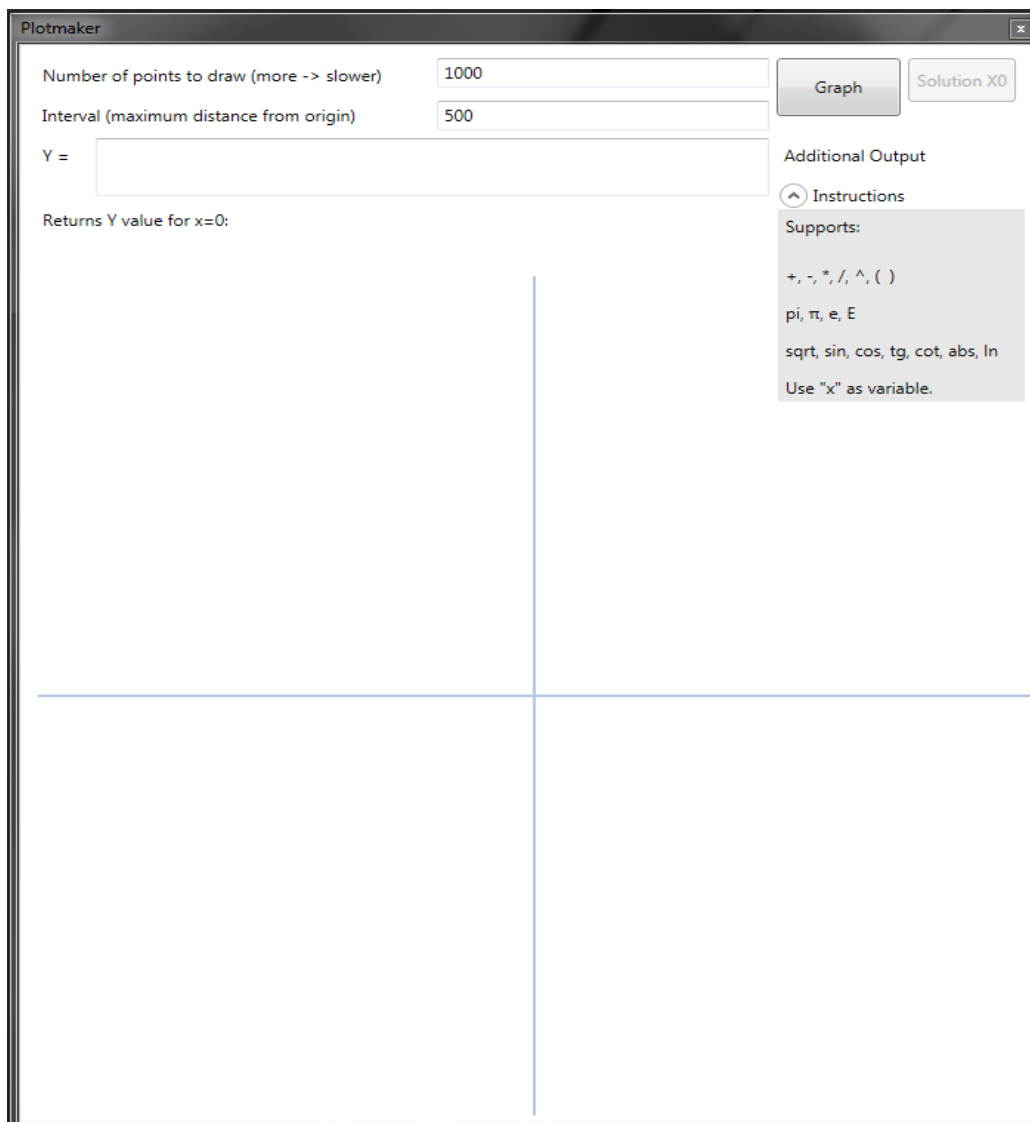
Ikona aplikace:



Vše je zasazeno do minimalistického prostředí (obrázek níže).
Uživatel si sám zadá předpis funkce, interval i počet bodů které chce vykreslit.

Okno obsahuje:

- 3 vstupní textboxy (počet bodů, interval a předpis funkce)
- 2 výstupní labely (hodnota Y pro $x=0$ a „errorlog“)
- 2 tlačítka (vykreslí graf, vyhodnotí Y pro $x=0$)
- expander s instrukcemi
- plátno pro graf



Popis zdrojového kódu:

Parser:

Hlavní součástí programu je tzv. „Parser“ který vypočítá hodnotu výrazu. Ten samotný se skládá ze 3 částí a 2 tzv. „Asistentů“.

```
while (text.Contains("ln") == true || text.Contains("abs") == true || text.Contains("sqrt") == true ||
```

První část zkontroluje výraz zda obsahuje některou z podporovaných funkcí. Pokud ano vytvoří „podvýraz“ vnitřku funkce a odešle ho k vyhodnocení. Pak teprve vypočítá funkční hodnotu.

```
while (text.Contains("(") == true)
{
    if (text.Contains(")") == true)
    {
```

Druhá část se zbavuje závorek. Najde první index uzavřené závorky a k ní najde odpovídající otevřenou závorku pomocí cyklu „for“. Pošle k vyhodnocení a nahradí závorku výsledkem.

```
^solver 100%
```

```
*solver 100%
```

```
/solver 100%
```

```
+ - solver 100%
```

Třetí část vyhodnocuje výraz. Postupuje podle priorit numerických operací. Nejdříve umocní, pak násobí a nakonec scítá. Pak odešle výsledek.

```
// 3.2) Get-your-numbers™ engine
```

```
Get number after operator:
```

```
Get number before operator:
```

Využívá k tomu 2 asistenty kteří vyhodnotí okolí daného operátoru a vrátí čísla mezi kterými má operace proběhnout.

Asistenti jsou velmi rozvětvení a počítají s desetinnými čárkami, zápornými čísly, E notacemi i +-nekonečny.

Vytvoření kolekce:

```
step = 2 * interval / pocetbodu;
if (step > 0)
{
    for (decimal i = -interval; i < interval; i += step)
    {
        string text = defaulttext.Replace("x", "(" + i.ToString() + ")");
        string reseni = vyres(text);
    }
}
```

Program si podle uživatelem zadaného intervalu a počtu bodů vypočítá o kolik má měnit hodnotu X. Pak za X jednotlivě dosazuje a pomocí Parseru dostane výsledky Y. Ty pak přidá do kolekce.

Vykreslení grafu:

```
tmpL = new Line();
tmpL.Stroke = Brushes.Black;
tmpL.X1 = Convert.ToDouble((bodyX[i - 1] + interval) * xScale);
tmpL.X2 = Convert.ToDouble((bodyX[i] + interval) * xScale);
tmpL.Y1 = graphCanvas.ActualHeight / 2 - bodyY[i - 1] * yScale;
tmpL.Y2 = graphCanvas.ActualHeight / 2 - bodyY[i] * yScale;
tmpL.StrokeThickness = lineWidth;
graphCanvas.Children.Add(tmpL);
lines.Add(tmpL);
```

WPF obsahuje jednoduché funkce pro grafiku. Program si založí „Line“, nastaví její barvu, šířku a dosadí souřadnice.

Jelikož kolekce X je v hodnotách decimal je třeba je konvertovat.

Hodnoty xScale a yScale jsou vypočteny podle rozměrů plátna a počtu bodů a určují jejich vzdálenosti od sebe.

Ukázky:

Graf fce $\sqrt{x} \cdot \ln(x)$ v intervalu $(-5,5)$:

Number of points to draw (more -> slower)

1000

Interval (maximum distance from origin)

5

Y = $\sqrt{x} \cdot \ln(x)$

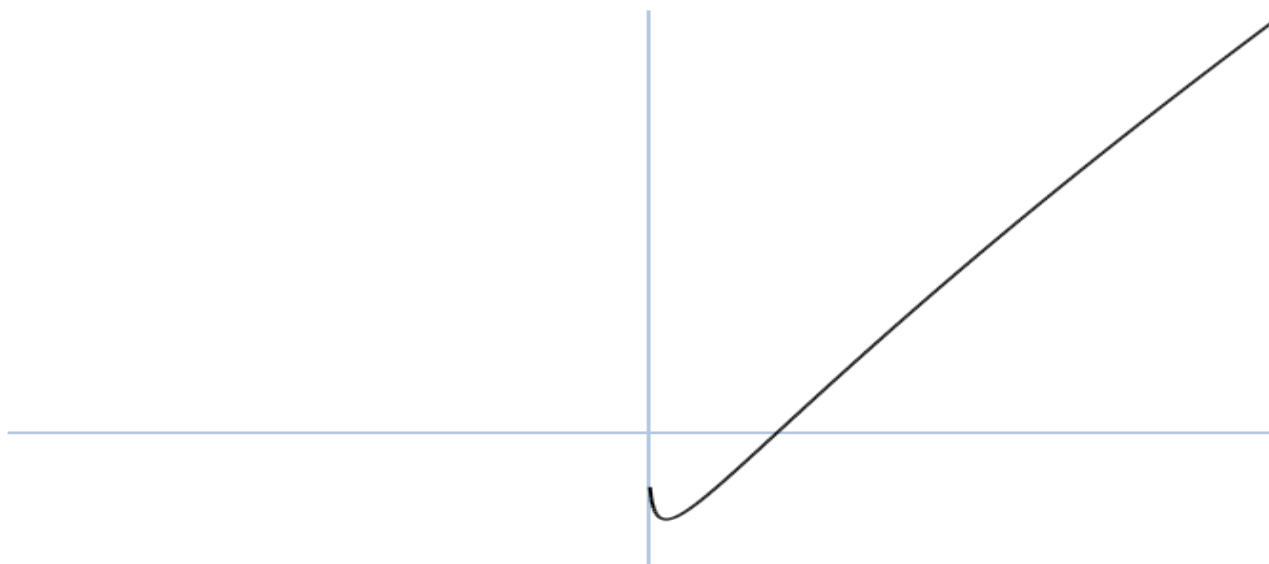
Graph

Solution X0

Additional Output

Instructions

Returns Y value for x=0: Není číslo



Graf fce $x^2/10 - \sin(x)$ v intervalu $(-5,5)$:

Number of points to draw (more -> slower)

1000

Interval (maximum distance from origin)

5

Y = $x^2/10 - \sin(x)$

Graph

Solution X0

Additional Output

Instructions

Returns Y value for x=0: 0

