

Лабораторная работа номер 6

```
LabWork6 LabWork6.Fx _points
1 using System;
2 using System.Collections.Generic;
3 using System.Drawing;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7
8 namespace LabWork6
9 {
10     public class Fx
11     {
12         /// <summary>
13         /// Коллекция точек [X;Y] графика
14         /// </summary>
15         private List<PointF> _points = new List<PointF>();
16         private Color _lineColor = Color.DarkGray;
17         private float _lineWidth = 1;
18         private Pen _pencil;
19
20         /// <summary>
21         /// Свойство, которое позволяе получить или установить массив точек графика
22         /// </summary>
23         public List<PointF> Points { get => _points; set => _points = value; }
24
25         /// <summary>
26         /// Свойство, которое позволяет проверить пустой ли массив точек в графике
27         /// </summary>
28         public bool IsEmpty { get => _points.Count == 0; }
29
30         /// <summary>
31         /// Свойство, которое позволяет получить или установить значение ширины линии графика
32         /// </summary>
33         public float LineWidth { get => _lineWidth; set => _lineWidth = value; }
34
35         /// <summary>
36         /// Свойство, которое позволяет получить или установить значение цвета линии графика
37         /// </summary>
38         public Color LineColor { get => _lineColor; set => _lineColor = value; }
39
40         /// <summary>
41         /// Свойство, которое позволяет получить или установить значение пера графика
42         /// </summary>
43         public Pen Pencil { get => _pencil; set => _pencil = value; }
44     }
45 }
```

класс Fx представляет собой класс функции, которая имеет такие атрибуты как:

- точки графика [X; Y]
- цвет
- Линии



```

{
    // Отрисовка (функции + таймер отрисовки)
    private List<Fx> _functions = new List<Fx>();
    private Timer _timer = new Timer();

    // Средства отрисовки
    private Pen _defaultPencil = new Pen(Color.DarkGray, 2);

```

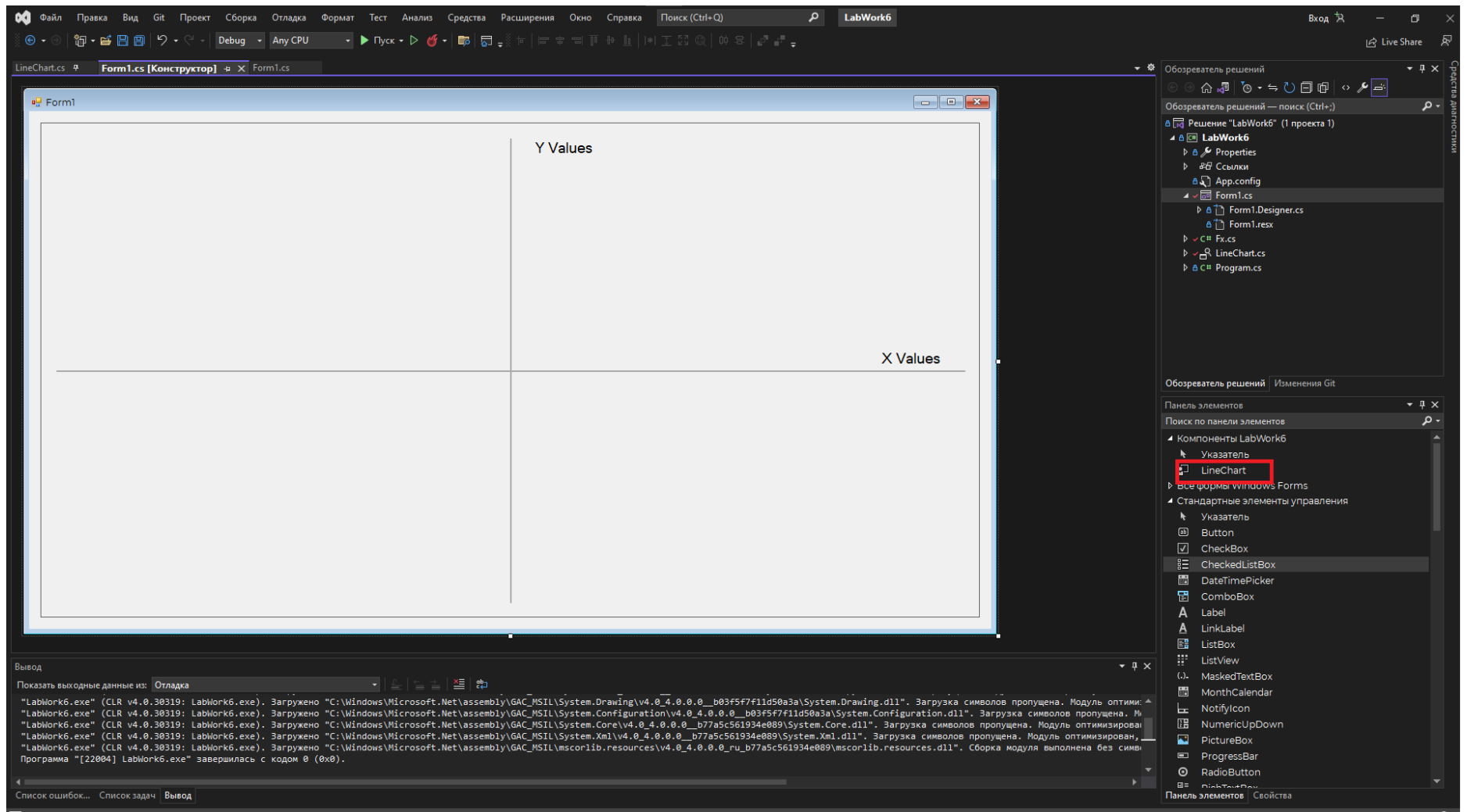
таймер для отрисовки всего по времени  
 набор функций, которые будут отрисовываться  
 Pen - карандаш для отрисовки осей

```


// Подписи осей
private string _axisXName = "X";
private string _axisYName = "Y";
private Label _labelXName = new Label();
private Label _labelYName = new Label();

```

подписи осей и их тексты на форме



Добавляем компонент через панель

AutoSizeMode	GrowOnly
AutoValidate	EnablePreventFocusChange
AxisXName	<b>X Values</b>
AxisYName	<b>Y Values</b>
BackColor	 Control

в свойствах можно изменить подписи осей

CausesValidation	True
ChartMargin	20
ContextMenuStrip	(нет)

отступы по бокам

далее переходим в форму

```
using System.Windows.Forms;

namespace LabWork6
{
    Ссылка: 3 | LancelotOzerny, 4 ч назад | Автор: 1, изменений: 4
    public partial class Form1 : Form
    {
        Ссылка: 1 | LancelotOzerny, 4 ч назад | Автор: 1, изменений: 4
        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

```
public Form1()
{
    InitializeComponent();

    Fx sin = new Fx();
    Fx cos = new Fx();
    Fx giperbola = new Fx();
}
```

Создаем новые функции. К примеру функции синуса, косинуса и гипербола

```
// Добавляем графики на отрисовку в компонент
grapher.AddFunction(sin);
grapher.AddFunction(cos);
grapher.AddFunction(giperbola);
```

и отсылаем их в отрисовку. Теперь они будут отображаться у нас.

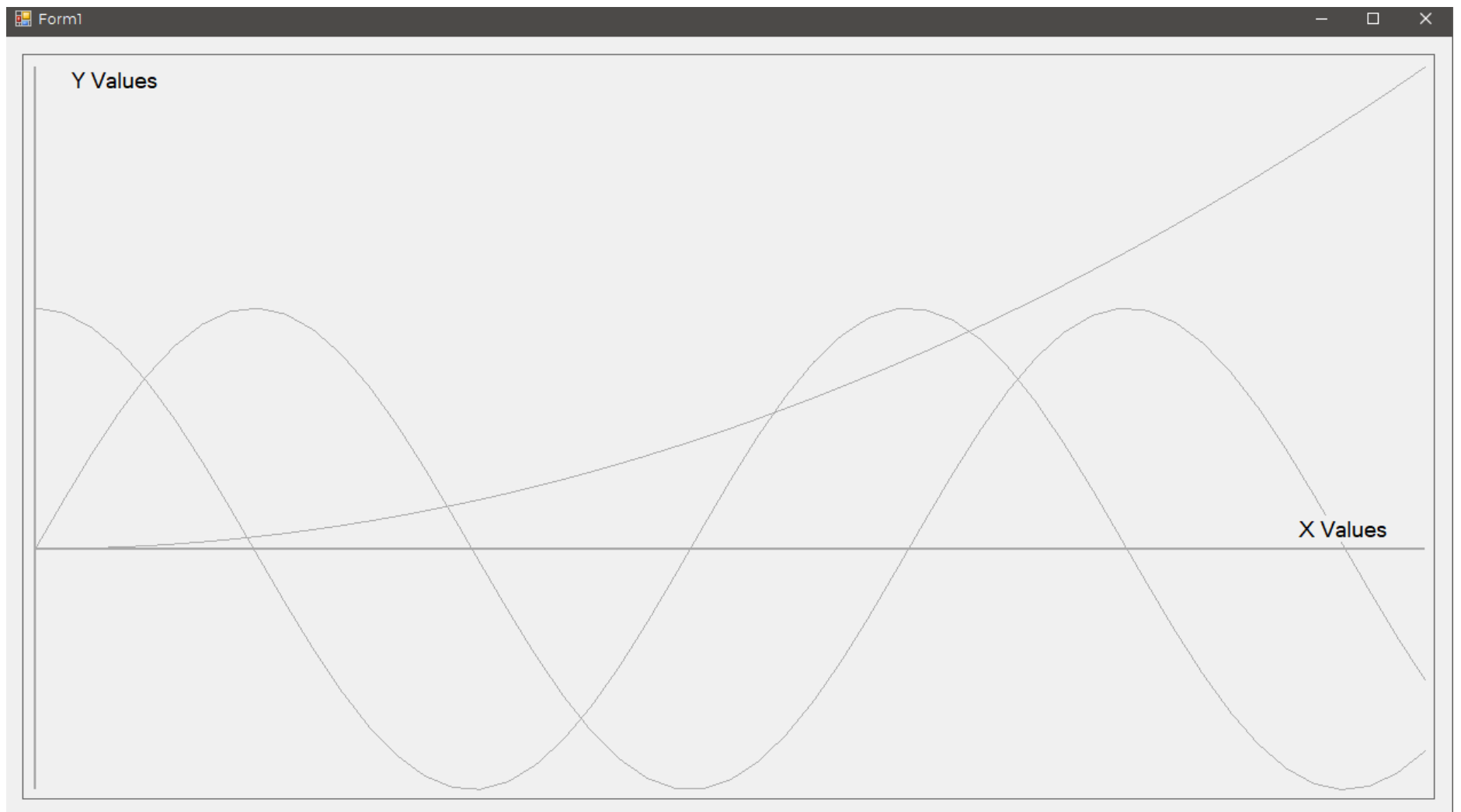


Но только после добавления значений

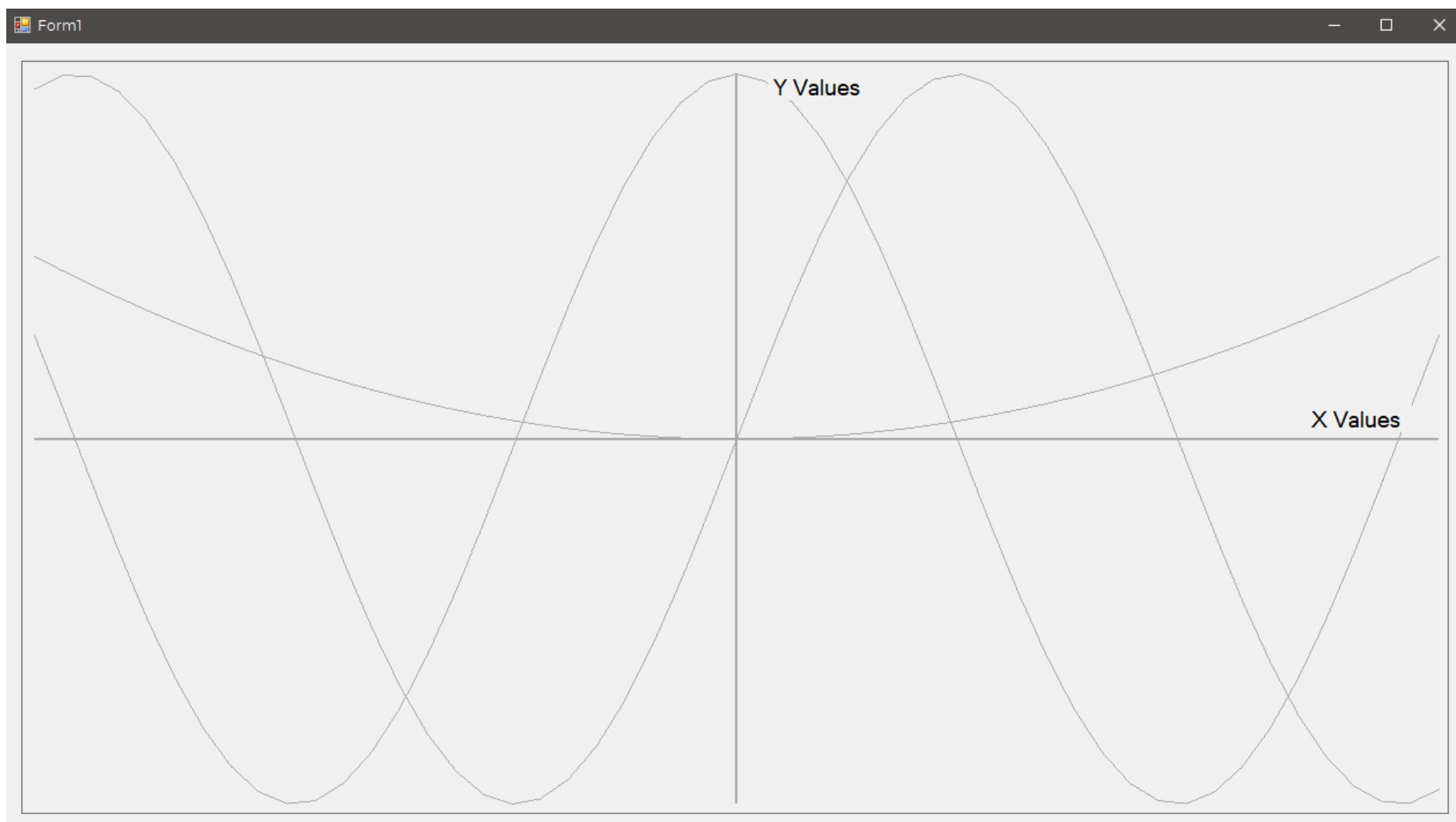


```
// Добавляем значения в графики
for (float x = 0; x < 10; x += 0.2f)
{
    sin.AddPoint(x, (float)Math.Sin(x));
    cos.AddPoint(x, (float)Math.Cos(x));
    giperbola.AddPoint(x, (float)(x * x / 50));
}
```

(так как гипербола слишком высокая, то результат я погелил на 50)  
Добавляем точки через метод `AddPoint(float x; float y);`



и получаем результат



(результат при изменении X от -5 до 5)

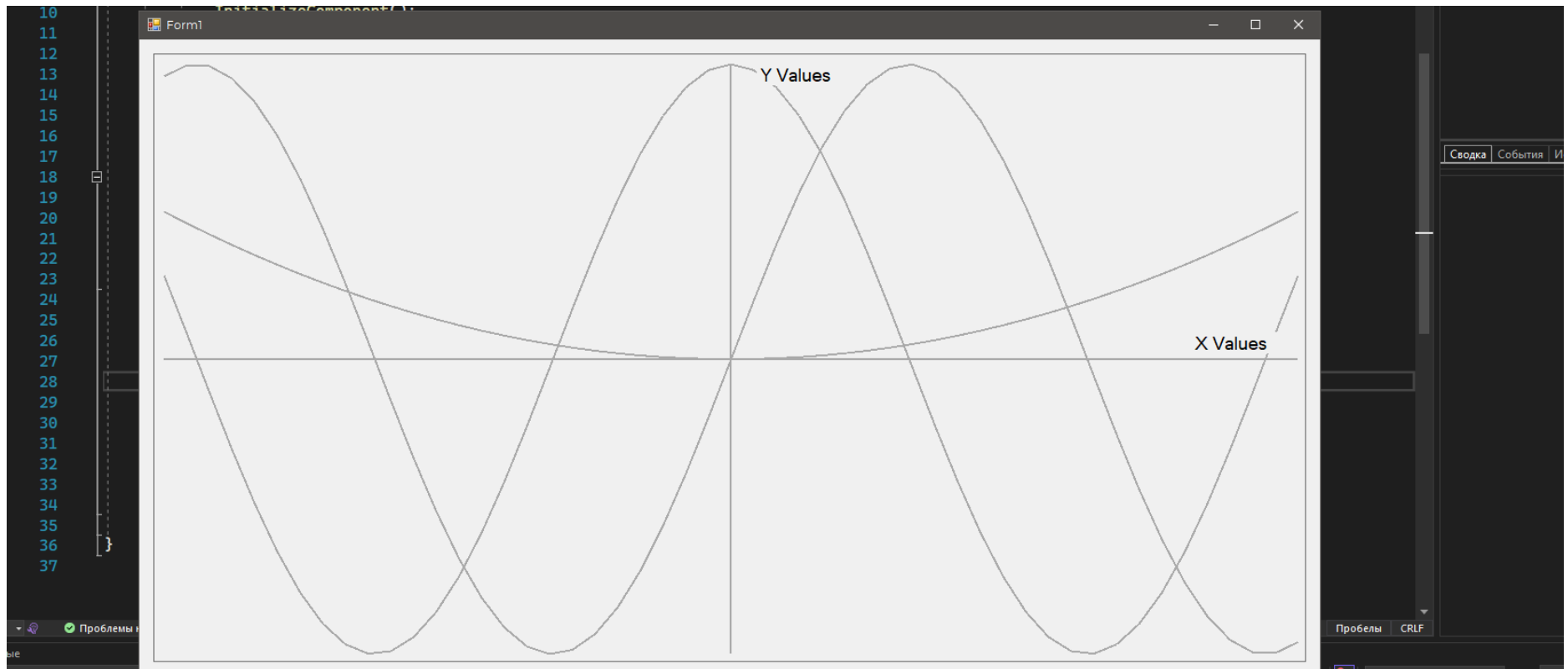
```

    giperbola.AddPoint(x, (float)(x *
}

// Меняем толщину линий
sin.LineWidth = 2;
cos.LineWidth = 2;
giperbola.LineWidth = 2;

```

Для более наглядного вида меняем ширину линий на 2



И поменяем цвет

```
// Меняем цвет линий  
sin.LineColor = System.Drawing.Color.Red;  
cos.LineColor = System.Drawing.Color.Green;  
giperbola.LineColor = System.Drawing.Color.Blue;
```

```
// Добавляем графики из структуры в компонент
```

В итоге получаем

