

ЛАБОРАТОРНАЯ РАБОТА 6. ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ СРЕДСТВ ДЛЯ ОТОБРАЖЕНИЯ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ

Цель лабораторной работы: изучить возможности построения графиков с помощью компонента отображения графической информации **Chart**. Написать и отладить программу построения на экране графика заданной функции.

6.1. Как строится график с помощью компонента Chart

Обычно результаты расчетов представляются в виде графиков и диаграмм. Библиотека .NET Framework имеет мощный элемент управления Chart для отображения на экране графической информации (рис. 6.1).

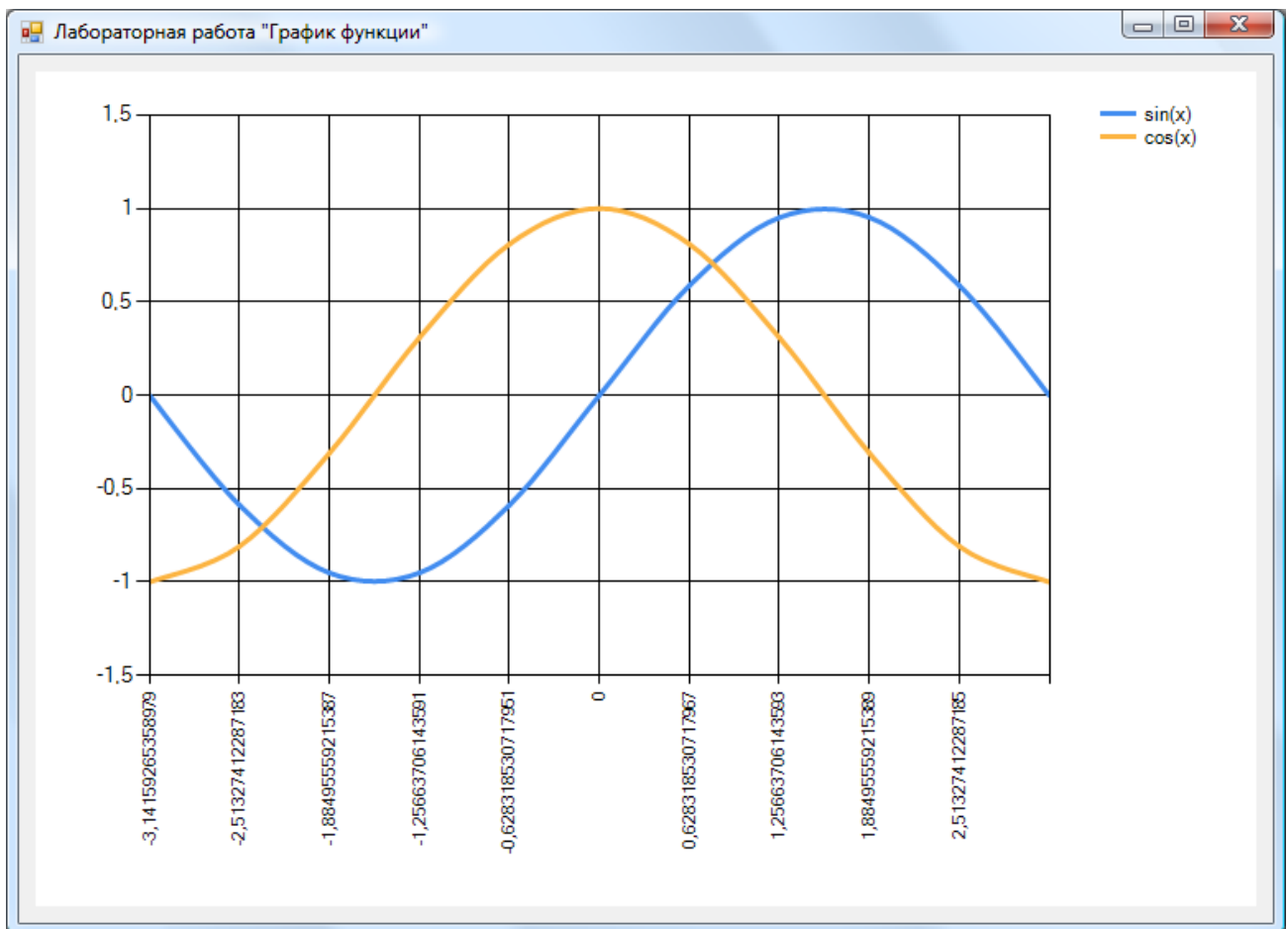


Рис 6.1. Окно программы с элементом управления.

Построение графика (диаграммы) производится после вычисления таблицы значений функции $y=f(x)$ на интервале $[X_{min}, X_{max}]$ с заданным шагом. Полученная таблица передается в специальный массив **Points** объекта **Series** компонента **Chart** с помощью метода **DataBindXY**. Компонент **Chart** осуществляет всю работу по отображению графиков: строит и размечает оси, рисует координатную сетку, подписывает название осей и самого графика, отображает переданную таблицу в виде всевозможных графиков или диаграмм. При необходимости компоненту **Chart** передаются данные о толщине, стиле и цвете линий, параметрах шрифта подписей, шагах разметки координатной сетки и другие настройки. В

процессе работы программы изменение параметров возможно через обращение к соответствующим свойствам компонента Chart. Так, например, свойство AxisX содержит значение максимального предела нижней оси графика и при его изменении во время работы программы автоматически изменяется изображение графика.

6.2. Пример написания программы

Задание: составить программу, отображающую графики функций $\sin(x)$ и $\cos(x)$ на интервале $[X_{\min}, X_{\max}]$. Предусмотреть возможность изменения разметки координатных осей, а также шага построения таблицы.

Прежде всего, следует определить в коде класса все необходимые переменные и константы. Конечно, можно обойтись и без этого, вставляя значения в виде чисел прямо в формулы, но это, во-первых, снизит читабельность кода программы, а во вторых, значительно усложнит изменение каких-либо параметров программы, например, интервала построения графика.

```
/// <summary>
/// Левая граница графика
/// </summary>
private double XMin = -Math.PI;

/// <summary>
/// Правая граница графика
/// </summary>
private double XMax = Math.PI;

/// <summary>
/// Шаг графика
/// </summary>
private double Step = (Math.PI * 2) / 10;

// Массив значений X - общий для обоих графиков
private double[] x;

// Два массива Y - по одному для каждого графика
private double[] y1;
private double[] y2;
```

Также в коде класса следует описать глобальную переменную типа Chart, к которой мы будем обращаться из разных методов:

```
Chart chart;
```

Поскольку данный класс не входит в пространства имен, подключаемые по умолчанию, следует выполнить дополнительные действия. Во-первых, в Обзоре

вателе решений нужно щёлкнуть правой кнопкой по секции Ссылки и добавить ссылку на библиотеку визуализации (рис. 6.2):

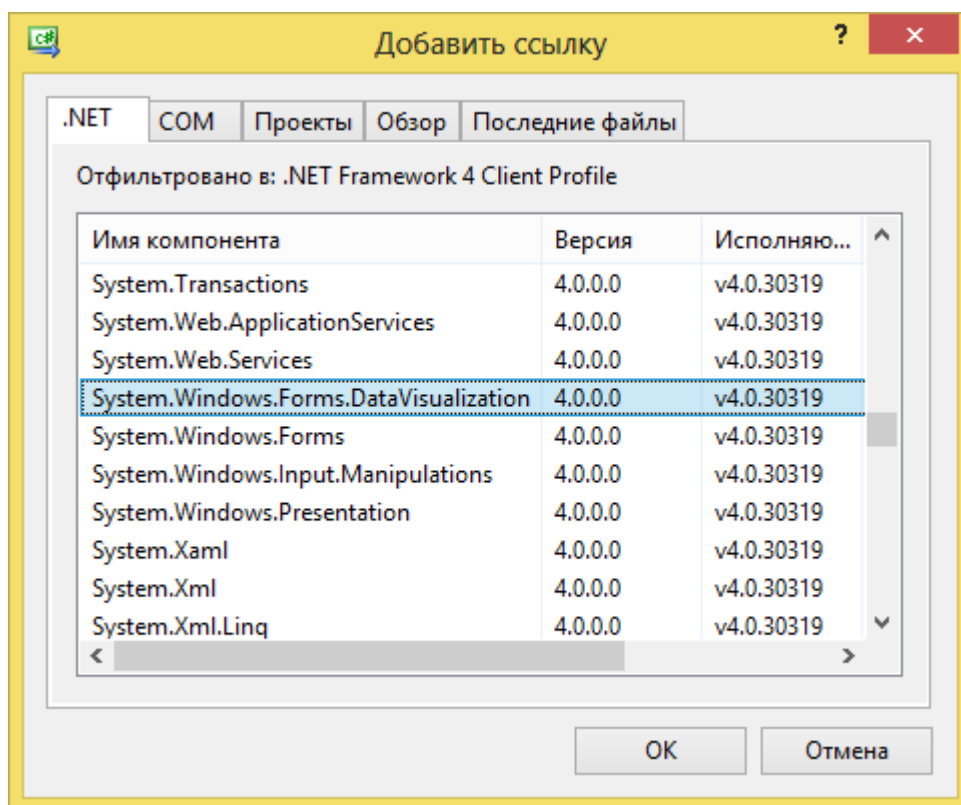


Рис. 6.2. Добавление ссылки на библиотеку визуализации.

Кроме того, следует подключить соответствующее пространство имен:

```
using System.Windows.Forms.DataVisualization.Charting;
```

Далее следует определить метод, который будет рассчитывать количество шагов и вычислять значения функций в каждой точке, внося вычисленные значения в массивы x , $y1$ и $y2$:

```
/// <summary>  
/// Расчёт значений графика  
/// </summary>  
private void CalcFunction()  
{  
    // Количество точек графика  
    int count = (int)Math.Ceiling((XMax - XMin) / Step)  
    + 1;  
    // Создаём массивы нужных размеров  
    x = new double[count];  
    y1 = new double[count];  
    y2 = new double[count];  
    // Рассчитываем точки для графиков функции
```

```

for (int i = 0; i < count; i++)
{
// Вычисляем значение X
x[i] = XMin + Step * i;
// Вычисляем значение функций в точке X
y1[i] = Math.Sin(x[i]);
y2[i] = Math.Cos(x[i]);
}
}

```

После расчёта значений нужно отобразить графики на форме с помощью элемента Chart. Элемент управления Chart нельзя выбрать с помощью панели элементов – его нужно создавать прямо в коде программы. Вторым шагом следует создать область отображения графика и настроить внешний вид осей:

```

/// <summary>
/// Создаём элемент управления Chart и настраиваем его
/// </summary>
private void CreateChart()
{
// Создаём новый элемент управления Chart
chart = new Chart();
// Помещаем его на форму
chart.Parent = this;
// Задаём размеры элемента
chart.SetBounds(10, 10, ClientSize.Width - 20,
ClientSize.Height - 20);

// Создаём новую область для построения графика
ChartArea area = new ChartArea();
// Даём ей имя (чтобы потом добавлять графики)
area.Name = "myGraph";
// Задаём левую и правую границы оси X
area.AxisX.Minimum = XMin;
area.AxisX.Maximum = XMax;
// Определяем шаг сетки
area.AxisX.MajorGrid.Interval = Step;
// Добавляем область в диаграмму
chart.ChartAreas.Add(area);

// Создаём объект для первого графика
Series series1 = new Series();
// Ссылаемся на область для построения графика
series1.ChartArea = "myGraph";
// Задаём тип графика - сплайны

```

```

series1.ChartType = SeriesChartType.Spline;
// Указываем ширину линии графика
series1.BorderWidth = 3;
// Название графика для отображения в легенде
series1.LegendText = "sin(x)";
// Добавляем в список графиков диаграммы
chart.Series.Add(series1);
// Аналогичные действия для второго графика
Series series2 = new Series();
series2.ChartArea = "myGraph";
series2.ChartType = SeriesChartType.Spline;
series2.BorderWidth = 3;
series2.LegendText = "cos(x)";
chart.Series.Add(series2);

// Создаём легенду, которая будет показывать названия
Legend legend = new Legend();
chart.Legends.Add(legend);
}

```

Наконец, все эти методы следует откуда-то вызвать. Чтобы графики появлялись сразу после запуска программы, надо вызывать их в обработчике события Load формы:

```

private void Form1_Load(object sender, EventArgs e)
{
    // Создаём элемент управления
    CreateChart();

    // Расчитываем значения точек графиков функций
    CalcFunction();

    // Добавляем вычисленные значения в графики
    chart.Series[0].Points.DataBindXY(x, y1);
    chart.Series[1].Points.DataBindXY(x, y2);
}

```

6.3. Выполнение индивидуального задания

Построить графики функций, самостоятельно выбрать удобные параметры настройки.

$$1. \quad t = \frac{2 \cos\left(x - \frac{\pi}{6}\right)}{0.5 + \sin^2 y} \left(1 + \frac{z^2}{3 - z^2/5}\right).$$

2. $u = \frac{\sqrt[3]{8 + |x - y|^2 + 1}}{x^2 + y^2 + 2} - e^{|x-y|} (tg^2 z + 1)^x.$
3. $v = \frac{1 + \sin^2(x + y)}{\left| x - \frac{2y}{1 + x^2 y^2} \right|} x^{|y|} + \cos^2 \left(\operatorname{arctg} \frac{1}{z} \right).$
4. $w = |\cos x - \cos y|^{(1+2\sin^2 y)} \left(1 + z + \frac{z^2}{2} + \frac{z^3}{3} + \frac{z^4}{4} \right).$
5. $\alpha = \ln \left(y^{-\sqrt{|x|}} \right) \left(x - \frac{y}{2} \right) + \sin^2 \operatorname{arctg}(z).$
6. $\beta = \sqrt{10(\sqrt[3]{x} + x^{y+2})} (\arcsin^2 z - |x - y|).$
7. $\gamma = 5 \operatorname{arctg}(x) - \frac{1}{4} \arccos(x) \frac{x + 3|x - y| + x^2}{|x - y|z + x^2}.$
8. $\varphi = \frac{e^{|x-y|} |x - y|^{x+y}}{\operatorname{arctg}(x) + \operatorname{arctg}(z)} + \sqrt[3]{x^6 + \ln^2 y}.$
9. $\psi = \left| x^{\frac{y}{x}} - \sqrt[3]{\frac{y}{x}} \right| + (y - x) \frac{\cos y - \frac{z}{(y - x)}}{1 + (y - x)^2}.$
10. $a = 2^{-x} \sqrt{x + 4\sqrt{|y|}} \sqrt[3]{e^{x-1/\sin z}}.$
11. $b = y^{\sqrt[3]{|x|}} + \cos^3(y) \frac{|x - y| \left(1 + \frac{\sin^2 z}{\sqrt{x + y}} \right)}{e^{|x-y|} + \frac{x}{2}}.$
12. $c = 2^{(y^x)} + (3^x)^y - \frac{y \left(\operatorname{arctg} z - \frac{\pi}{6} \right)}{|x| + \frac{1}{y^2 + 1}}.$
13. $f = \frac{\sqrt[4]{y + \sqrt[3]{x-1}}}{|x - y| (\sin^2 z + tgz)}.$
14. $g = \frac{y^{x+1}}{\sqrt[3]{|y-2|} + 3} + \frac{x + \frac{y}{2}}{2|x + y|} (x + 1)^{-1/\sin z}.$
15. $h = \frac{x^{y+1} + e^{y-1}}{1 + x|y - tgz|} (1 + |y - x|) + \frac{|y - x|^2}{2} - \frac{|y - x|^3}{3}.$

16. $q = b - 2f$
17. $p = ta$
18. $k = g + c/2$
19. $l = f + 1/f$
20. $m = v - |u|$
21. $n = \ln(a + \arcsin(a))$
22. $o = 1 - ah$
23. $d = \varphi - 1/t$
24. $s = \frac{\sin(|h|)}{|h|}$
25. $r = \sqrt{|\gamma - 1|}$