ПРАКТИЧЕСКАЯ РАБОТА №21 (ПРОДОЛЖЕНИЕ).

**ГРАФИКИ ФУНКЦИЙ**

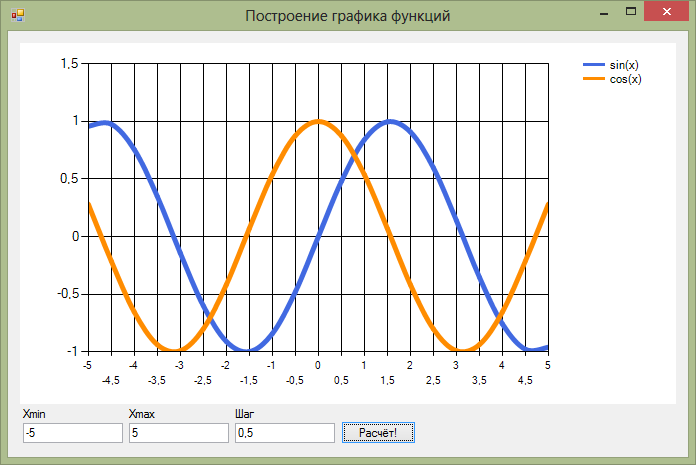
**Цель лабораторной работы:**изучить возможности построения графиков с помощью элемента управления Сhart. Написать и отладить программу построения на экране графика заданной функции.

**Краткие теоретические сведения:**

Как строится график с помощью элемента управления Chart

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

Построение графика (диаграммы) производится после вычисления таблицы значений функции *y=f(x)* на интервале [Xmin, Xmax] с заданным шагом. Полученная таблица передается в специальный массив Points объекта Series элемента управления Сhart с помощью метода DataBindXY. Элемент управления 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. В элементе управления Сhart можно настроить толщину, стиль и цвет линий, параметры шрифта подписей, шаги разметки координатной сетки и многое другое. В процессе работы программы изменение параметров возможно через обращение к соответствующим свойствам элемента управления Chart. Так, например, свойство AxisX содержит значение максимального предела нижней оси графика и при его изменении во время работы программы автоматически изменяется изображение графика.



Пример написания программы

Задание: составить программу, отображающую графики функций *sin(x)* и *cos(x)* на интервале [Xmin, Xmax]. Предусмотреть возможность изменения разметки координатных осей, а также шага построения таблицы.

Прежде всего, следует поместить на форму сам элемент управления Chart. Он располагается в панели элементов в разделе *Данные*.

Список графиков хранится в свойстве Series, который можно изменить, выбрав соответствующий пункт в окне свойств. Поскольку на одном поле требуется вывести два отдельных графика функций, нужно добавить ещё один элемент. Оба элемента, и существующий и добавленный, нужно соответствующим образом настроить: изменить тип диаграммы ChartType на Spline. Здесь же можно изменить подписи к графикам с абстрактных *Series1* и *Series2* на *sin(x)* и *cos(x)* – за это отвечает свойство Legend. Наконец, с помощью свойства BorderWidth можно сделать линию графика потолще, а затем поменять цвет линии с помощью свойства Color.

Ниже приведён текст обработчика нажатия кнопки «Расчёт!», который выполняет все требуемые настройки и расчёты и отображает графики функций:

private void buttonCalc\_Click(object sender, EventArgs e)

{

// Считываем с формы требуемые значения double Xmin = double.Parse(textBoxXmin.Text); double Xmax = double.Parse(textBoxXmax.Text); double Step = double.Parse(textBoxStep.Text);

// Количество точек графика

int count = (int)Math.Ceiling((Xmax - Xmin) / Step)

+ 1;

// Массив значений X – общий для обоих графиков double[] x = new double[count];

// Два массива Y – по одному для каждого графика double[] y1 = new double[count];

double[] y2 = new double[count];

// Расчитываем 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]);

}

// Настраиваем оси графика chart1.ChartAreas[0].AxisX.Minimum = Xmin; chart1.ChartAreas[0].AxisX.Maximum = Xmax;

// Определяем шаг сетки chart1.ChartAreas[0].AxisX.MajorGrid.Interval = Step;

// Добавляем вычисленные значения в графики chart1.Series[0].Points.DataBindXY(x, y1); chart1.Series[1].Points.DataBindXY(x, y2);

}

Выполнение индивидуального задания

