

Использование компонента для построения диаграмм

В проект добавить две ссылки:

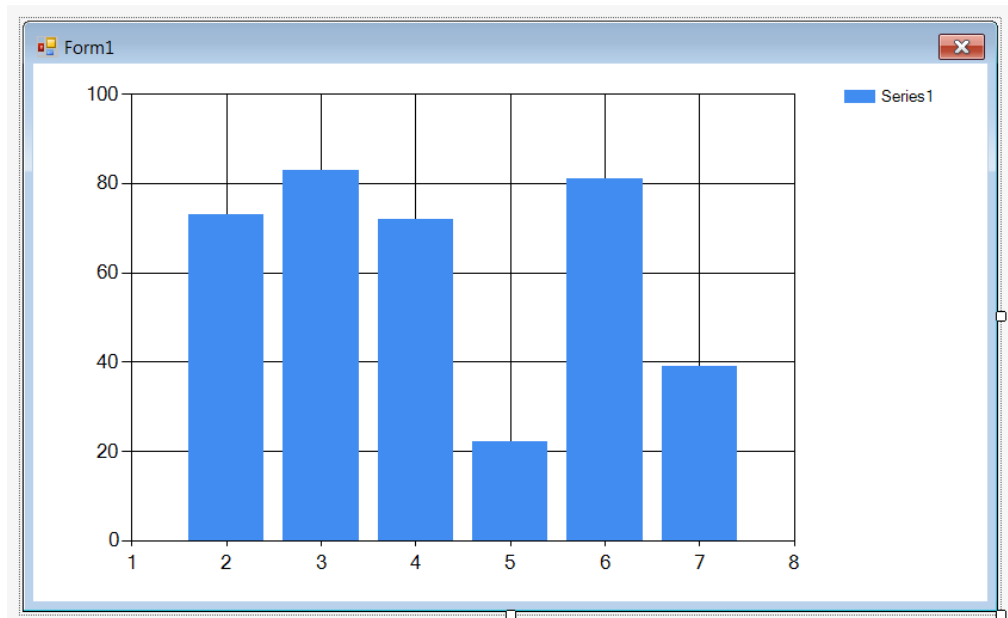
System.Windows.Forms.DataVisualization

System.Windows.Forms.DataVisualization.Design

На форму поместить компонент **Chart** из группы «Данные» Панели элементов.

Настроить для него свойство **Dock** равным **Fill**.

Все прочие настройки будут выполнены в коде проекта и в обработчиках событий.



```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Windows.Forms.DataVisualization.Charting;
using System.IO;

namespace Charts
{
    public partial class Form1 : Form
    {
        string chartName = "";
        string curDir = "";

        static MarkerStyle[] markerStyle = { MarkerStyle.Square, MarkerStyle.Circle,
            MarkerStyle.Triangle, MarkerStyle.Diamond, MarkerStyle.Cross };
        static int[] size = { 5, 6, 7, 6, 6 };
        static Color[] color = { Color.Blue, Color.Red,
            Color.Green, Color.Magenta, Color.Gray };

        static void CreateChart(Chart ch, string title, int dataCount)
        {
            ch.BorderlineColor = Color.Black;
            ch.BorderlineDashStyle = ChartDashStyle.Solid;
            ch.BorderlineWidth = 2;

            ChartArea area = ch.ChartAreas[0];
            area.AxisY.MajorGrid.Enabled = false;
            area.AxisY.MinorGrid.Enabled = false;
        }
    }
}
```

```

        area.AxisX.MajorGrid.Enabled = false;
        area.AxisX.MinorGrid.Enabled = false;
        if (dataCount > 200)
            area.AxisX.Interval = 100;
        else if (dataCount > 20)
            area.AxisX.Interval = 10;
        else
            area.AxisX.Interval = 1;

        Legend legend = ch.Legends[0];
        legend.Docking = Docking.Top;
        legend.Alignment = StringAlignment.Center;
        legend.Font = new Font("Arial", 12);

        ch.Series.Clear();
        ch.Titles.Clear();
        ch.Titles.Add(title);
        ch.Titles[0].Font = new Font("Arial", 14);
        ch.Tag = double.MaxValue;
    }

    static void AddSeries(Chart ch, int id, string name, double[] data)
    {
        Series series = ch.Series.Add(name);
        series.ChartType = SeriesChartType.Line;
        series.ShadowColor = Color.Black;
        series.ShadowOffset = 1;
        series.MarkerStyle = markerStyle[id];
        series.MarkerSize = size[id];
        series.Points.DataBindY(data);
        series.Color = color[id];
        double min = data.Min();
        if ((double)ch.Tag > min)
            ch.Tag = min;
    }

    void Error(string s)
    {
        chart1.Visible = false;
        FormClosed -= Form1_FormClosed;
        MessageBox.Show(s, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        Application.Exit();
    }

    void ProcessChartFile(string fileName)
    {
        chartName = Path.ChangeExtension(fileName, null);
        Text = "Charts - " + Path.GetFullPath(fileName);
        bool title = false;
        bool chartCreated = false;
        string sTitle = "";
        bool ser = false;
        bool legend = false;
        int id = 0;
        double[] data = null;
        foreach (var s0 in File.ReadLines(fileName, Encoding.Default))
        {
            string s = s0.Trim();
            if (s == "" || s.StartsWith("%"))
                continue;
            if (s.StartsWith("="))
            {
                if (!title)
                {
                    Error("В файле не указан заголовок диаграммы.");
                    return;
                }
                ser = true;
                legend = false;
                try

```

```

        {
            data = s.TrimStart('=')
                .Split(new char[] { ' ' },
                    StringSplitOptions.RemoveEmptyEntries)
                .Select(e => double.Parse(e)).ToArray();
            if (!chartCreated)
            {
                CreateChart(chart1, sTitle, data.Length);
                chartCreated = true;
            }
        }
        catch
        {
            if (s.Length > 100)
                s = s.Substring(0, 100) + " ...";
            Error($"Неверная последовательность данных: {s}.");
            return;
        }
        continue;
    }
    if (!ser && !title)
    {
        title = true;
        sTitle = s;
        continue;
    }
    if (ser && !legend)
    {
        legend = true;
        AddSeries(chart1, id, s, data);
        data = null;
        ++id;
    }
}
if (id == 0)
{
    Error("В файле отсутствуют наборы данных или не указаны их легенды.");
    return;
}
double d = (double)chart1.Tag;
chart1.ChartAreas[0].AxisY.Minimum = Math.Floor(d) - 1;
}

public Form1()
{
    InitializeComponent();
}

private void Form1_Load(object sender, EventArgs e)
{
    curDir = Environment.CurrentDirectory;
    if (File.Exists(curDir + "\\Charts.ini"))
        try
        {
            {
                string[] s = File.ReadAllText("Charts.ini").Split();
                Width = int.Parse(s[0]);
                Height = int.Parse(s[1]);
            }
        }
        catch { }
    Left = (Screen.PrimaryScreen.WorkingArea.Width - Width) / 2;
    Top = (Screen.PrimaryScreen.WorkingArea.Height - Height) / 2;
    Text = "Charts";
    MaximizeBox = false;
    MinimizeBox = false;
}

private void Form1_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Escape)
        Close();
}

```

```

}

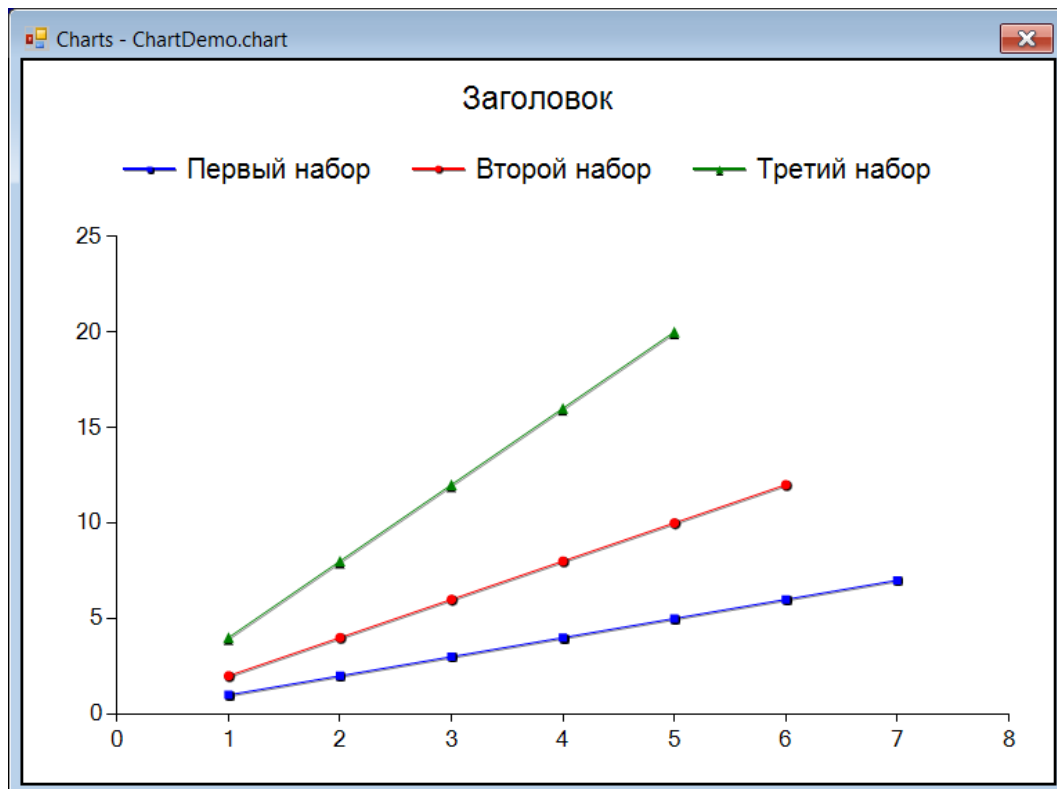
private void Form1_FormClosed(object sender, FormClosedEventArgs e)
{
    File.WriteAllText(curDir + "\\Charts.ini", Width + " " + Height);
    chart1.SaveImage(chartName + ".png", ChartImageFormat.Png);
}

private void Form1_Shown(object sender, EventArgs e)
{
    if (Environment.GetCommandLineArgs().Length < 2)
    {
        Error("Не указан файл с данными для построения диаграммы.");
        return;
    }
    string f = Environment.GetCommandLineArgs()[1];
    if (!File.Exists(f))
    {
        Error($"Файл {f} не найден.");
        return;
    }
    ProcessChartFile(f);
}
}
}
}

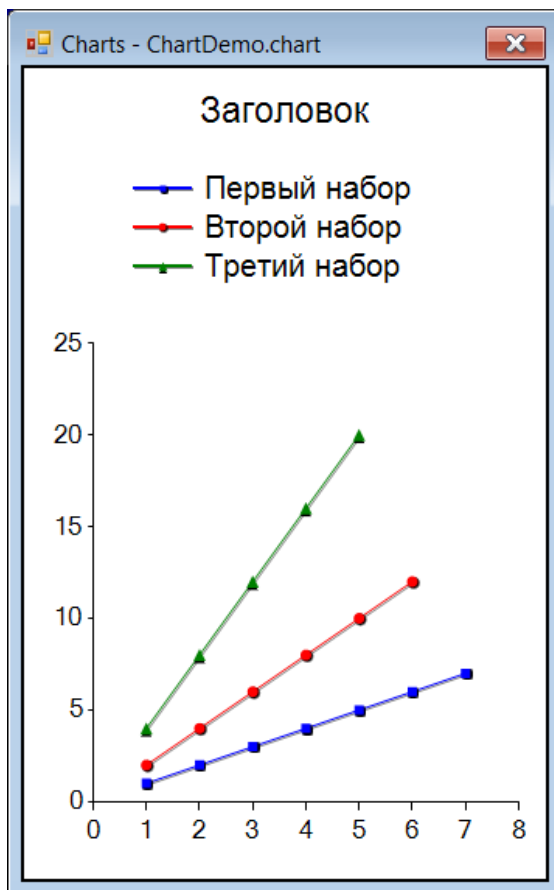
```

Образец chart-файла (ChartDemo.chart):

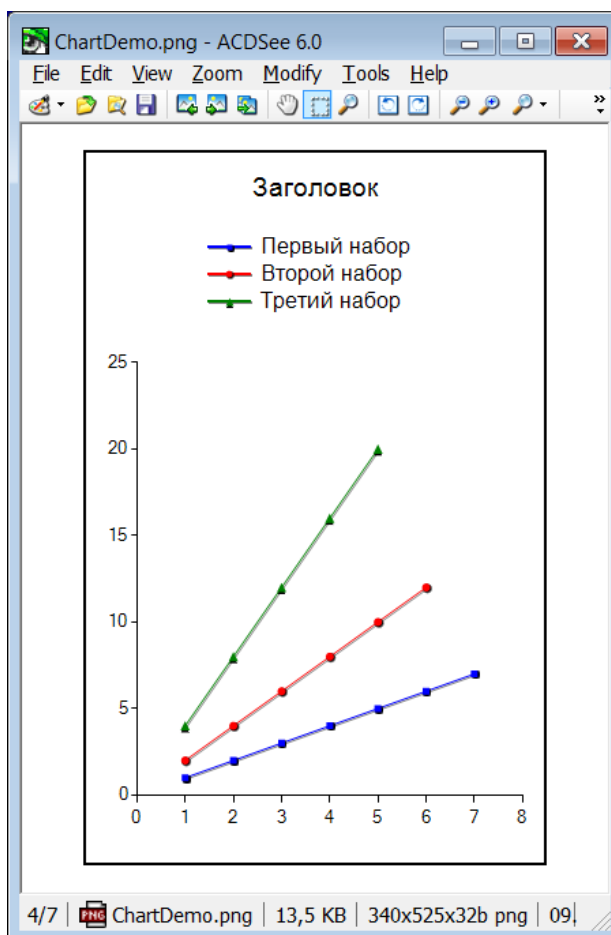
Заголовок
 =1 2 3 4 5 6 7
 Первый набор
 =2 4 6 8 10 12
 Второй набор
 =4 8 12 16 20
 Третий набор



Размеры окна можно изменять.



При нажатии Esc программа завершается, а диаграмма сохраняется в файле ChartDemo.png.



При очередном запуске программы восстанавливаются предыдущие размеры окна. Если связать расширение chart с данной программой, то запускать программу можно при щелчке на любом chart-файле. Для связывания надо выделить какой-либо chart-файл, выполнить для него команду контекстного меню «Открыть с помощью...» и настроить требуемую связь.

