

Работа 14. Графика. Функции. Фигуры

1. Графика. Рисуем функции

Предмет исследований

- Графические средства C#.
- Компонент Chart

Контрольные вопросы

1. Класс Graphics (графический объект).
2. Класс Font - шрифт.
3. Класс Pen - перо.
4. Класс Brush - кисть.
5. Вывод текста. Метод DrawString.
6. Вывод линии. Метод DrawLine.
7. Компонент Chart – средство отображения диаграмм.
8. ChartAreas – области диаграммы.
9. Series – ряды диаграммы.
10. Сколько рядов надо для рисования N графиков в одной области.
11. Legendes – легенды диаграммы..
12. Задание функции для отображения в компоненте Chart.
13. Стили отображения диаграмм.
14. Использование легенды в компоненте Chart.

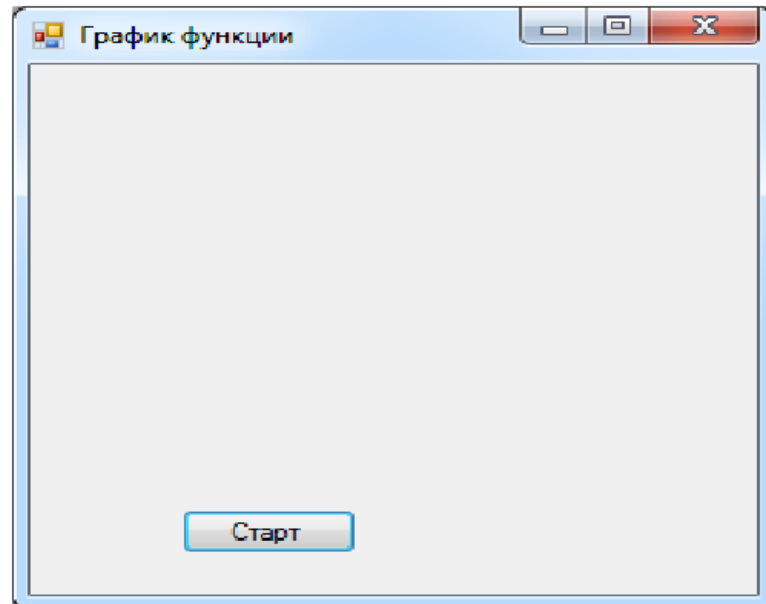
1.1. Текст и график функции в форме

Создать программу вывода текста и рисования в форме графиков функций.
Проект – WindowsForm приложение.

Варианты заданий

№	График функции $y(x)$		
	Функция	<u>х начальное</u>	<u>х конечное</u>
1.	$\sin(x)$	0	6π
2.	$\cos(x)$	0	6π
3.	$\sin(x)+\sin(2x)$	0	6π
4.	$\sin(x)-\sin(2x)$	0	6π
5.	$\sin(x)+\cos(2x)$	0	6π
6.	$\sin(x)-\cos(2x)$	0	6π
7.	$\sin(x)*\exp(x)$	0	6π
8.	$\cos(x)*\exp(x)$	0	6π
9.	$\sin(x)*\exp(-x)$	0	6π
10.	$\cos(x)*\exp(-x)$	0	6π

Пример. Создать программу рисования в форме графика функции с поясняющим текстом.
Проект – WindowsForms приложение. Функция - синус. Программа предусматривает рисование графика в форме Form1 линиями с помощью метода DrawLine. Над графиком с помощью метода DrawString выводится поясняющий текст. Кнопка **Старт** вызывает построение графиков в окне.



Листинг программы

```
using System;
using System.Drawing;
using System.Windows.Forms;

namespace GraphicFunction
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            int imax = 100;           //число точек в периоде
            int t=2;                  //число периодов
            int amp=70;               //амплитуда
            int h = 40;               //отступ для текста
            int x0=20;                //начала координат
            int y0 = h+amp;
            double[] f = new double [imax*t+10];
            // Функция
            for (int i = 0; i < imax * t; i++)
            {
                f[i] = Math.Round(amp * Math.Sin(2 * Math.PI / imax * i));
            }
            // Инструменты рисования
            Graphics g = Graphics.FromHwnd(this.Handle);           // Где рисуем
            Pen pen = Pens.Black;                                   // Чем рисуем
            // Текст заголовка
            g.DrawString("График синусоиды", new Font("Arial", 14),
                Brushes.Red, 0, 0); //Вывод текста
            //textBox1.Text = "График синусоиды";
        }
    }
}
```

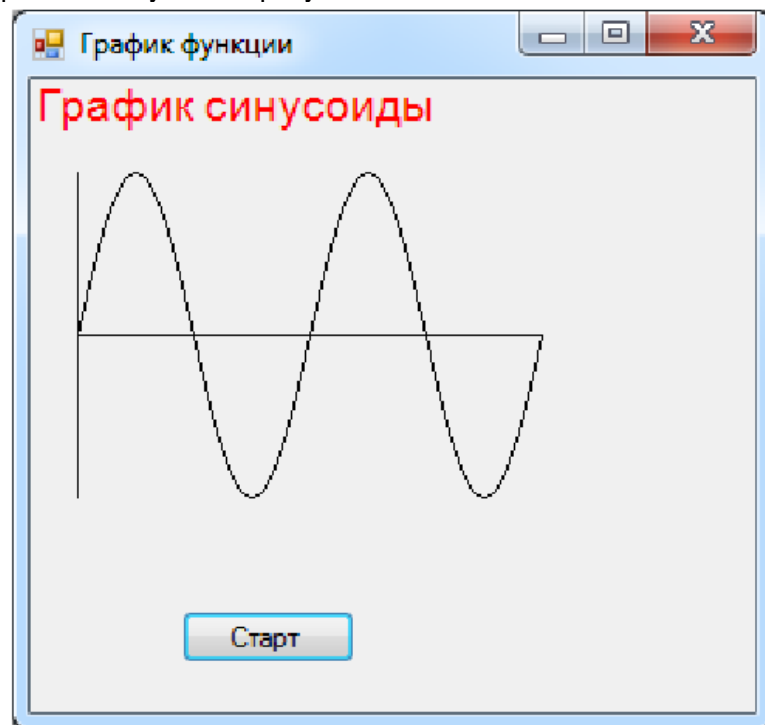
```

//Рисуем график
g.DrawLine(pen , x0, y0, x0+imax*t, y0);    //Рисуем ось X
g.DrawLine(pen, x0, y0-amp, x0, y0+amp);    //Рисуем ось Y
for (int i = 0; i < imax * t; i++)          //Рисуем график
{
    int f1 = y0 - (int)ff[i];                //Координата Y[i]
    int f2 = y0 - (int)ff[i + 1];            //Координата Y[i+1]
    g.DrawLine(pen, x0+i, f1, x0+i+1, f2);
}
}

private void Form1_Load(object sender, EventArgs e)
{
}
}
}

```

При прогоне программы получается результат:



1.2. Компонент Chart

Создать программу отображения диаграммы двух функций с использованием компонента Chart. Проект – WindowsForm приложение.

Варианты заданий

№	График функции $y(x)$		
	Функция 1	Функция 2	Стиль линий
1.	$\sin(x)$	$\sin(2x)$	Spline
2.	$\sin(x)$	$-\sin(2x)$	Point
3.	$\cos(x)$	$\sin(2x)$	StepLine
4.	$\cos(x)$	$-\sin(2x)$	Line
5.	$\sin(x)$	$\sin(2x)$	Colomn
6.	$\sin(x)$	$-\sin(2x)$	Spline
7.	$\cos(x)$	$\sin(2x)$	Point
8.	$\cos(x)$	$-\sin(2x)$	StepLine
9.	$\sin(x)$	$\sin(3x)$	Line
10.	$\sin(x)$	$-\sin(3x)$	Colomn

Пример. Создать программу отображения диаграммы двух функций $\sin(x)$ и $\cos(x)$ с использованием компонента Chart. Проект – WindowsForm приложение. Стиль линий Spline.

Создаем – WindowsForm приложение. В форму заносим компоненты chart1 для отображения диаграммы и button1 для создания стартового обработчика событий.

В окне свойств Button1 задаем его свойству Text значение Старт.

В окне компонента Chart1 в разделе Seies определяем две серии:

- для функции $\sin(x)$ с именем Синус.
- для функции $\cos(x)$ с именем Косинус.

Двойным щелчком по кнопке создаем в окне кода шаблон обработчика события нажатия кнопки. Функциональная часть обработчика включает задание в цикле наборов данных для серий.

Листинг программы

```
using System;
using System.Windows.Forms;

namespace Chart
{
    public partial class Form1 : Form
    {
        public Form1()
    }
}
```

```

{
    InitializeComponent();
}

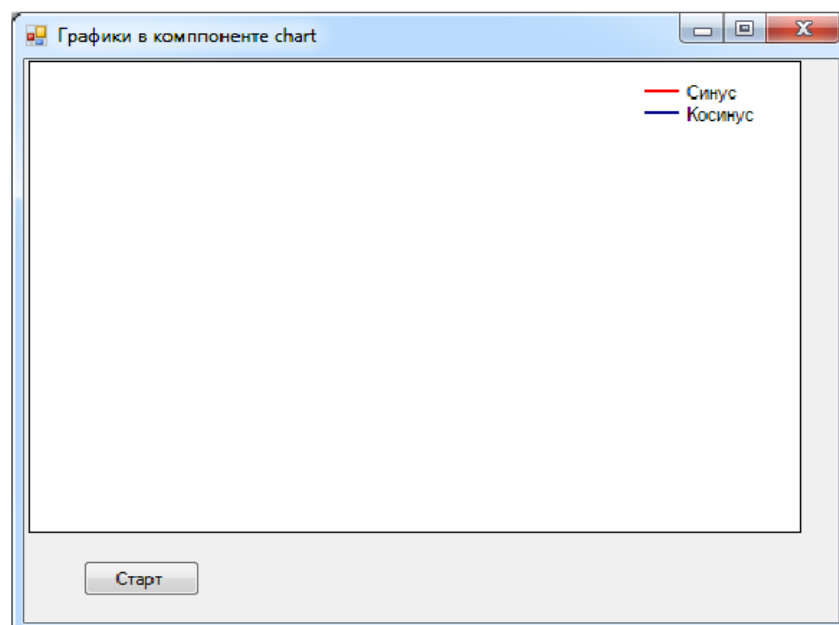
private void chart1_Click(object sender, EventArgs e)
{
}

private void button1_Click(object sender, EventArgs e)
{
    double y = 0;
    for (int x = 0; (x <= 19); x++)
    {
        y = Math.Sin(Math.PI / 5 * x);
        chart1.Series["Синус"].Points.AddXY(x, y);
        y = Math.Cos(Math.PI / 5 * x);
        chart1.Series["Косинус"].Points.AddXY(x, y);
    }
}

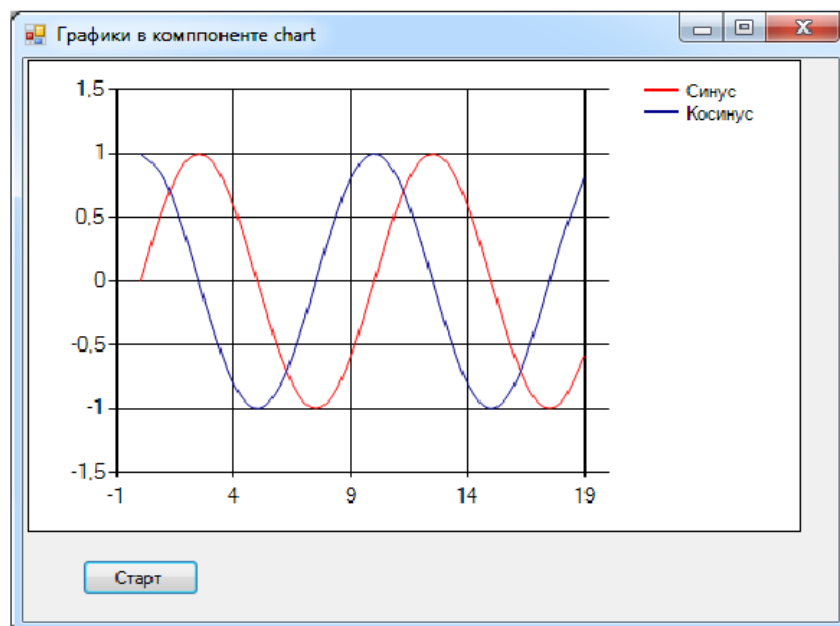
private void Form1_Load(object sender, EventArgs e)
{
}
}

```

При запуске программы отображается форма, в которой прорисовываются диаграмма с двумя поименованными сериями и кнопка старта. Самых графиков пока нет, так как данные для них формирует обработчик.



Кнопка button1 (Старт). Она вызывает построение графиков в окне.



2. Графика. Рисуем фигуры

Предмет исследований

- Методы работы с графическими фигурами.
- Методы работы с залитыми графическими фигурами.

Контрольные вопросы

1. Назначение графических примитивов.
2. Кривая Безье и метод DrawBezier.
3. Кривая и метод DrawCurve.
4. Замкнутая кривая и метод DrawClosedCurve.
5. Прямоугольник и метод DrawRectangle.
6. Полигон и метод DrawPolygon.
7. Эллипс и метод DrawEllipse.
8. Дуга эллипса и метод DrawArc.
9. Торт и метод DrawPie.
10. Метод заполнения замкнутых фигур (заполнить область),
11. Стили линий.
12. Стили заливки.

Задание. Создать программу рисования в форме графических фигур без заливки и с заливкой.

Варианты заданий

№	Фигура	Стиль линий	Метод заливки	Стиль заливки
1.	Кривая Безье	Dash	FillClosedCurve	Cross
2.	Кривая	DashDot	FillRectangle	DiagonalCross
3.	Замкнутая кривая	DashDotDot	FillPolygon	ForwardDiagonal
4.	Прямоугольник	Dot	FillEllipse	BackwardDiagonal
5.	Полигон	Dash	FillPie	Cross
6.	Эллипс	DashDot	FillClosedCurve	DiagonalCross
7.	Торт	DashDotDot	FillRectangle	ForwardDiagonal
8.	Дуга эллипса	Dot	FillPolygon	BackwardDiagonal
9.	Кривая	Dash	FillEllipse	Cross
10.	Замкнутая кривая	DashDot	FillPie	DiagonalCross

Пример. Создать программу рисования в форме графических фигур без заливки и с заливкой.

В левом верхнем углу формы будут рисоваться два графика: верхний – фигура без заливки, нижний – фигура с заливкой.

В форме размещаем 4 выпадающих списка ComboBox и метки Label для заголовков списков:

- Фигура - Combobox1 и Label1.
- Стиль линии – Combobox2 и Label2.
- Фигура с заливкой – Combobox3 и Label3.
- Стиль заливки – Combobox4 и Label4.

Кнопка button1 с надписью старт для запуска обработчика события.

В окне свойств Combobox1 определим фигуры рисования:

- DrawBezier – кривая Безье.
- DrawCurve – кривая.
- DrawClosedCurve – замкнутая кривая.
- DrawRectangle – прямоугольник.
- DrawPolygon – многоугольник.
- DrawEllipse – эллипс.
- DrawArc – дуга эллипса.
- DrawPie – торт.

В окне свойств Combobox2 определим стили линий фигур:

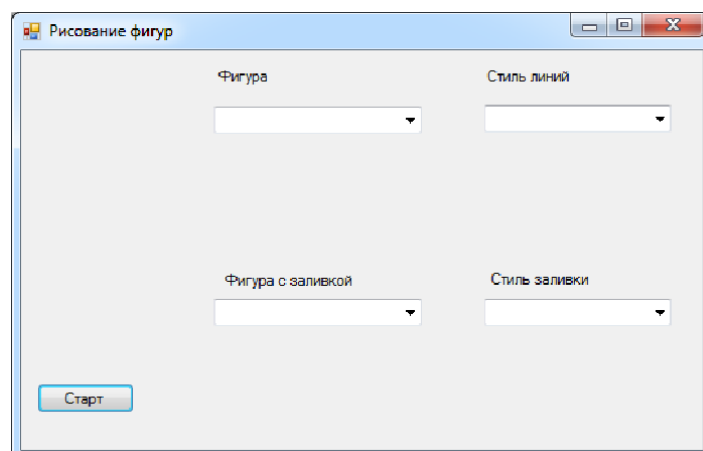
- Dash – тире.
- DashDot – тире + точки.
- DashDotDot - тире + 2 точки.
- Dot - точки.

В окне свойств Combobox3 определим фигуры с заливкой:

- FillClosedCurve - замкнутая кривая.
- FillRectangle - прямоугольник.
- FillPolygon - многоугольник.
- FillEllipse - эллипс.
- FillPie - торт.

В окне свойств Combobox4 определим стили заливки фигур:

- Cross – сетка.
- DiagonalCross – диагональная сетка.
- ForwardDiagonal – диагональ прямая.
- BackwardDiagonal – диагональ обратная.



В листинге программы дополнительно задаются размеры рисуемых фигур.

Листинг программы

```
using System;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Windows.Forms;

namespace Figures
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Graphics g = Graphics.FromHwnd(this.Handle);    // Где рисуем
            Pen pen = new Pen(Color.Black);                // Чем рисуем
            Brush brush = Brushes.White;
            Brush brush1 = Brushes.Blue;
            //stem.Drawing.Drawing2D.FillMode;
            HatchBrush brush2 = new HatchBrush(HatchStyle.Cross,
                ForeColor, BackColor);
            HatchBrush brush3 = new HatchBrush(HatchStyle.DiagonalCross,
                ForeColor, BackColor);
            HatchBrush brush4 = new HatchBrush(HatchStyle.ForwardDiagonal,
                ForeColor, BackColor);
            HatchBrush brush5 = new HatchBrush(HatchStyle.BackwardDiagonal,
                ForeColor, BackColor);
            Point[] p =
            {
                new Point(10, 0),
                new Point(80,70),
                new Point(90,50),
                new Point(50,90),
            };
            Point[] p1 =
            {
                new Point(0,110),
                new Point(80,180),
                new Point(90,160),
                new Point(50,200),
            };
            Rectangle rect = new Rectangle(0,0, 110, 220);
            Rectangle rect1 = new Rectangle(10,0, 100, 100);
            Rectangle rect2= new Rectangle(10,110, 100, 100);
        }
    }
}
```

```

g.FillRectangle(brush, rect);

if (comboBox3.SelectedItem == "Dash")
{
    pen.DashStyle = DashStyle.Dash;
}
if (comboBox3.SelectedItem == "DashDot")
{
    pen.DashStyle = DashStyle.DashDot;
}
if (comboBox3.SelectedItem == "DashDotDot")
{
    pen.DashStyle = DashStyle.DashDotDot;
}

if (comboBox3.SelectedItem == "Dot")
{
    pen.DashStyle = DashStyle.Dot;
}
if (comboBox4.SelectedItem == "Cross")
{
    brush1 = brush2;
}
if (comboBox4.SelectedItem == "DiagonalCross")
{
    brush1 = brush3;
}
if (comboBox4.SelectedItem == "ForwardDiagonal")
{
    brush1 = brush4;
}
if (comboBox4.SelectedItem == "BackwardDiagonal")
{
    brush1 = brush5;
}
if (comboBox1.SelectedItem == "DrawRectangle")
{
    g.DrawRectangle(pen, rect1);
}
if (comboBox2.SelectedItem == "FillRectangle")
{
    g.FillRectangle(brush1, 10, 110, 100, 100);
}
if (comboBox1.SelectedItem == "DrawPolygon")
{
    g.DrawPolygon(pen, p);
}
if (comboBox2.SelectedItem == "FillPolygon")
{
    g.FillPolygon(brush1, p1);
}
if (comboBox1.SelectedItem == "DrawEllipse")
{
    g.DrawEllipse(pen, rect1);
}
if (comboBox2.SelectedItem == "FillEllipse")

```

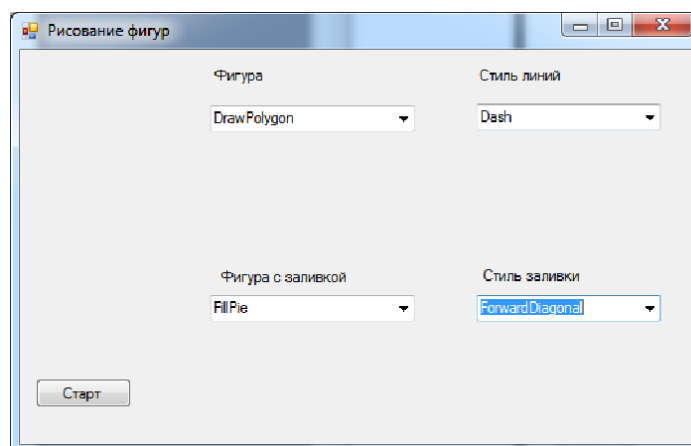
```

    {
        g.FillEllipse(brush1, 10, 110, 100, 100);
    }
    if (comboBox1.SelectedItem == "DrawPie")
    {
        g.DrawPie(pen, rect1, 50, 250);
    }
    if (comboBox2.SelectedItem == "FillPie")
    {
        g.FillPie(brush1, rect2, 50, 250);
    }
    if (comboBox1.SelectedItem == "DrawCurve")
    {
        g.DrawCurve(pen, p);
    }
    if (comboBox1.SelectedItem == "DrawClosedCurve")
    {
        g.DrawClosedCurve(pen, p);
    }
    if (comboBox2.SelectedItem == "FillClosedCurve")
    {
        g.FillClosedCurve(brush1, p1);
    }
    if (comboBox1.SelectedItem == "DrawArc")
    {
        g.DrawArc(pen, rect1, 50, 250);
    }
    if (comboBox1.SelectedItem == "DrawBezier")
    {
        g.DrawBezier(pen, p[0], p[1], p[2], p[3]);
    }
}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
}
}
}

```

Пример запуска.



А вот результат:

