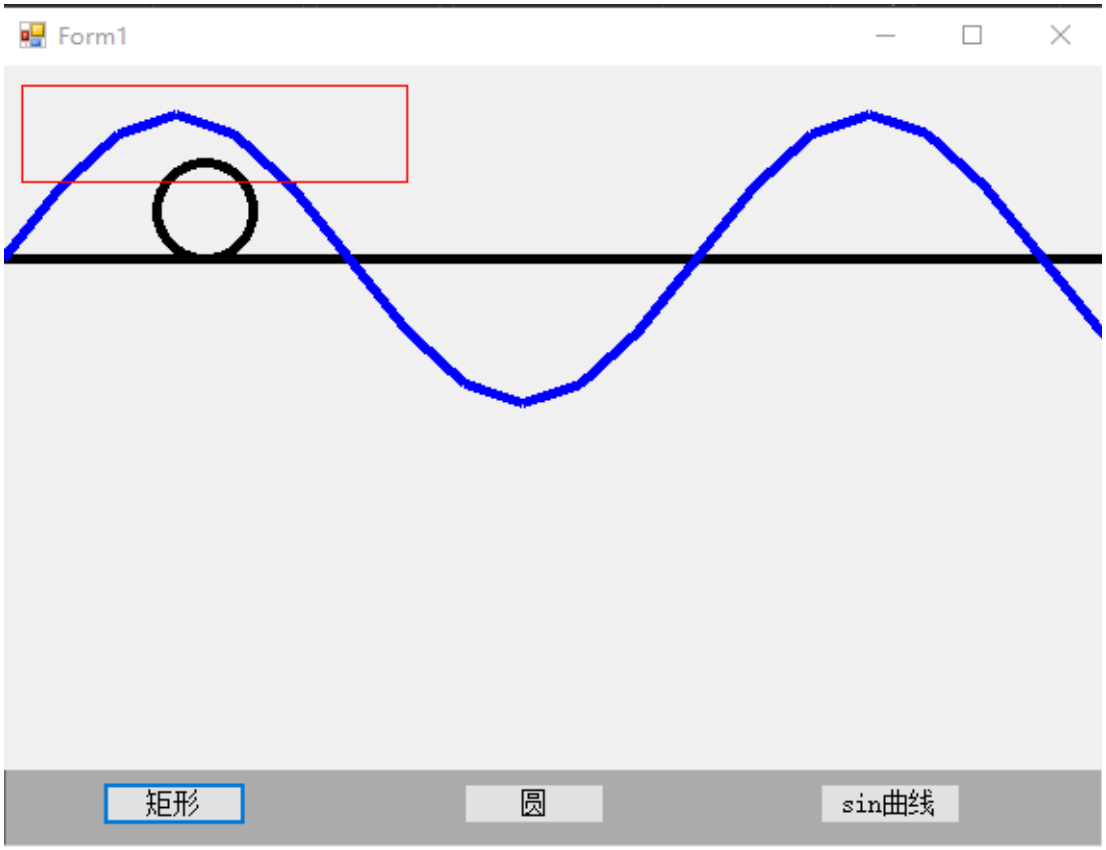
1、在一个 PictureBox 里画出矩形、圆、sin 曲线。




```
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;

namespace WindowsFormsApp1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

        }


        private void Form1_Load(object sender, EventArgs e)
```

```
        {

        }

        private void pictureBox1_Click(object sender, EventArgs e)
        {
            Point p0 = pictureBox1.PointToClient(MousePosition);
            Graphics g = pictureBox1.CreateGraphics();
            Pen pen = new Pen(Color.Red, 1);
            g.DrawRectangle(pen, new Rectangle(p0.X, p0.Y, 10, 30));
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Graphics g = this.CreateGraphics();
            Pen pen = new Pen(Color.Red, 1);

            g = pictureBox1.CreateGraphics();
            g.DrawRectangle(pen, new Rectangle(10, 10, 200, 50));
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Graphics g = this.CreateGraphics();
            Pen pen = new Pen(Color.Black, 5);

            g = pictureBox1.CreateGraphics();
            g.DrawEllipse(pen, new Rectangle(80, 50, 50, 50));
        }

        private void button3_Click(object sender, EventArgs e)
        {
            Graphics g = this.CreateGraphics();
            Pen MyPenBlue = new Pen(Color.Blue, 5);
            Pen MyPenBlack = new Pen(Color.Black, 5);
            //Pen MyPenControl = new Pen(Color.FromName("Control"),
3);
            //Pen MyPenRed = new Pen(Color.FromArgb(128, 255, 0, 0),
3);
            g = pictureBox1.CreateGraphics();
            g.DrawLine(MyPenBlack, 0, 100, 800, 100); //画直线
            int i;
            double angle1;
            float x1, y1, x2, y2;
```

```
            for (i = 0; i < 720; i += 30) //画正弦函数
            {
                angle1 = (float)i * 2 * 3.14 / 360;
                x1 = i;
                y1 = ((float)(300 - 150 * Math.Sin(angle1))-100)/2;


                x2 = i + 30;
                float angle2 = (float)((i + 30) * 2 * 3.14 / 360); //
折线


                y2 = ((float)(300 - 150 * Math.Sin(angle2))-100)/2;


                g.DrawLine(MyPenBlue, x1, y1, x2, y2);

            }
        }
    }
}
```

2、完成课堂上所讲的图像处理程序，至少实现其中的两种处理功能。


Form1
祝福
祝福
扩散效果
去色
底片效果


Form1
祝福
祝福
扩散效果
去色
底片效果


```
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;

namespace WindowsFormsApp1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

        }


        private void Form1_Load(object sender, EventArgs e)
```

```
        {

        }

        private void pictureBox1_Click(object sender, EventArgs e)
        {
            Point p0 = pictureBox1.PointToClient(MousePosition);
            Graphics g = pictureBox1.CreateGraphics();
            Pen pen = new Pen(Color.Red, 1);
            g.DrawRectangle(pen, new Rectangle(p0.X, p0.Y, 10, 30));
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Bitmap bitmapOriginal = new Bitmap(pictureBox1.Image);
            Bitmap bitmapResult = new Bitmap(pictureBox1.Image);
            int x = 0, y = 0, bitNumber, distance = 1;
            Random rand = new Random();
            for (int r = 2; r < bitmapOriginal.Height - 2; r++)
            {
                for (int c = 2; c < bitmapOriginal.Width - 2; c++)
                {
                    bitNumber = rand.Next(1, 4);
                    switch (bitNumber)
                    {
                        case 1: x = c - distance; y = r; break;
                        case 2: x = c; y = r - distance; break;
                        case 3: x = c + distance; y = r; break;
                        case 4: x = c; y = r + distance; break;
                    }
                    bitmapResult.SetPixel(c, r,
((Bitmap)pictureBox1.Image).GetPixel(x, y));
                }
            }
            pictureBox2.Image = bitmapResult;

        }

        private void button2_Click(object sender, EventArgs e)
        {
            Bitmap bitmapOriginal = new Bitmap(pictureBox1.Image);
            Bitmap bitmapResult = new Bitmap(pictureBox1.Image);
            int gray;
            for (int c = 0; c < bitmapResult.Width; c++)
```

```
            {
                for (int r = 0; r < bitmapResult.Height; r++)
                {
                    gray = (int)(bitmapOriginal.GetPixel(c, r).R *
0.3f +
                                                bitmapOriginal.GetPixel(c,
r).G * 0.6f +
                                                bitmapOriginal.GetPixel(c,
r).B * 0.1f);
                    bitmapResult.SetPixel(c, r, Color.FromArgb(gray,
gray, gray));
                }
            }
            pictureBox2.Image = bitmapResult;

        }

        private void button3_Click(object sender, EventArgs e)
        {
            //以底片效果显示图像
            try
            {
                int Height = this.pictureBox1.Image.Height;
                int Width = this.pictureBox1.Image.Width;
                Bitmap newbitmap = new Bitmap(Width, Height);
                Bitmap oldbitmap = (Bitmap)this.pictureBox1.Image;
                Color pixel;
                for (int x = 1; x < Width; x++)
                {
                    for (int y = 1; y < Height; y++)
                    {
                        int r, g, b;
                        pixel = oldbitmap.GetPixel(x, y);
                        r = 255 - pixel.R;
                        g = 255 - pixel.G;
                        b = 255 - pixel.B;
                        newbitmap.SetPixel(x, y, Color.FromArgb(r, g,
b));
                    }
                }
                this.pictureBox2.Image = newbitmap;
            }
            catch (Exception ex)
            {
```

```
                MessageBox.Show(ex.Message, "信息提示",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            }

        }
        }
    }
```