

## Простейший вывод отображения графического файла в форму

Поставим задачу вывода в форму какого-нибудь изображения растрового графического файла формата BMP, JPEG, PNG или других форматов.

Работать с графикой в форме можно по-разному. В первом варианте покажем работу с графикой через переопределение метода OnPaint.

Метод OnPaint является членом класса Form. Этот метод можно увидеть в раскрывающемся списке *методов и свойств* объекта Form1, набрав "this. " внутри какой-нибудь процедуры. Во всплывающей подсказке приводится синтаксис *переопределения* метода OnPaint. Переопределить метод означает, что можно добавить к уже существующим функциям собственные. Для этого в окне программного кода вместо private void

Form1\_Load...{ } напишем:

```
protected override void OnPaint(PaintEventArgs e){ }
```

В этом случае транслятор в окне **ERROR LIST** выдаст сообщение об ошибке, что наша программа: **does not contain a definition for Form1\_Load**. Щелкнув двойным щелчком на этом сообщении, мы попадем на вкладку программного кода дизайнера формы **Form1.Designer.cs**. Здесь красной волнистой линией будет подчеркнута подписка на событие Form1\_Load — нам следует просто удалить эту строку.

Теперь вернемся на вкладку программного кода **Form1.cs** и дополним имеющийся программный код командами для вывода в форму изображения из растрового файла poryv.png

### Вывод растрового изображения в форму (вариант 1)

// Программа выводит в форму растровое изображение из графического файла

```
using System;
using System.Drawing;
using System.Windows.Forms;
// Другие директивы using удалены, поскольку они не используются в данной
// программе
namespace SimpleImage1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        protected override void OnPaint(PaintEventArgs e)
        {
            this.Text = "Рисунок";
            // Размеры формы
            this.Width = 240; this.Height = 240;
            // Создаем объект для работы с изображением
            Image Рисунок = (Image) new Bitmap(@"D:\poryv.png");
            // Вывод изображения в форму
            e.Graphics.DrawImage(Рисунок, 5, 5);
            // x=5, y=5 — это координаты левого верхнего угла рисунка в
            // системе координат формы: ось x — вниз, ось y — вправо
        }
    }
}
```

```
}  
}
```

Как видно из текста программы, вначале указываем размеры формы с помощью свойств Width и Height, хотя размеры формы *удобно регулировать* на вкладке конструктора формы визуально. Далее создаем объект Рисунок для работы с изображением с указанием пути к файлу рисунка. Bitmap (дословный перевод с английского — карта битов) — метод представления изображения в форме прямоугольной матрицы, каждое значение которой соответствует цвету одной из точек изображения.

Теперь обращаемся непосредственно к методу рисования изображения в форме DrawImage, извлекая графический объект Graphics из аргумента e процедуры OnPaint.

Заметим, что это не единственный способ работы с графикой. Другой способ — это вызов того же метода OnPaint косвенно, через событие формы OnPaint. Такой способ работы с графикой представлен в следующей нашей программе — назовем ее SimpleImage2. Создадим процедуру обработки данного события обычным образом, т. е. на вкладке конструктора формы в панели свойств Properties щелкнем на значке молнии и в появившемся списке всех событий для объекта Form1 выберем событие Paint. Объект Graphics получаем из аргумента e события Paint.

### **Вывод растрового изображения в форму (вариант 2)**

// Простейший вывод изображения в форму

```
using System;  
using System.Drawing;  
using System.Windows.Forms;  
// Другие директивы using удалены, поскольку они не используются в данной  
// программе  
namespace SimpleImage2  
{  
    public partial class Form1 : Form  
    {  
        public Form1()  
        {  
            InitializeComponent();  
        }  
        private void Form1_Paint(object sender, PaintEventArgs e)  
        {  
            // В свойствах формы щелкнем на значке молнии и в появившемся  
            // списке всех событий для объекта Form1 выберем событие Paint.  
            // Событие Paint — это событие рисования формы:  
            this.Text = "Рисунок";  
            // Создаем объект для работы с изображением:  
            var Рисунок = Image.FromFile(@"D:\poryv.png");  
            // или var Рисунок = new Bitmap(@"D:\poryv.png");  
            // Вывод изображения в форму:  
            e.Graphics.DrawImage;  
        }  
    }  
}
```

Покажем еще один способ вывода графики в форму — соответствующую программу назовем SimpleImage3. В этом способе при щелчке на командной кнопке происходит непосредственное создание объекта класса Graphics.

### Вывод растрового изображения в форму (вариант 3)

// Простейший вывод изображения в форму

```
using System;
using System.Drawing;
using System.Windows.Forms;
// Другие директивы using удалены, поскольку они не используются в данной программе
namespace SimpleImage3
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void Form1_Load(object sender, EventArgs e)
        {
            // Событие загрузки формы:
            this.Text = "Рисунок";
            button1.Text = "Показать рисунок";
        }
        private void button1_Click(object sender, EventArgs e)
        {
            // Событие "щелчок на кнопке"
            var Рисунок = new Bitmap(@"D:\poryv.png");
            // Создание графического объекта:
            var Графика = this.CreateGraphics();
            // или var Графика = CreateGraphics();
            Графика.DrawImage;
        }
    }
}
```

Убедиться в работоспособности приведенных программ можно, открыв соответствующие решения в папках SimpleImage1, SimpleImage2 и SimpleImage3 сопровождающего книгу электронного архива.

В заключение замечу, что с рассмотренными в этом примере методами можно работать *не только* для вывода изображений графических файлов в форму, но и решать многие *другие задачи, связанные с графикой*.

### Использование элемента *PictureBox* для отображения растрового файла с возможностью прокрутки

Обычно для отображения точечных рисунков, рисунков из метафайлов, значков, рисунков из файлов в формате BMP, JPEG, GIF, PNG и проч. используется объект класса **PictureBox** (графическое поле). Часто рисунок оказывается слишком большим и не помещается целиком в пределах элемента управления PictureBox. Можно воспользоваться свойством элемента SizeMode, указав ему значение StretchImage. В этом случае изображение будет

вытягиваться или сужаться, чтобы в точности соответствовать размеру PictureBox. Чаще всего такой подход не устраивает разработчика, поскольку изображение значительно деформируется. Напрашивается решение в организации возможности прокрутки изображения (AutoScroll), но такого свойства у PictureBox нет. Зато такое свойство есть у элемента управления Panel. То есть, разместив PictureBox на элементе Panel с установленным свойством AutoScroll = true и указав при этом для PictureBox SizeMode = AutoSize, имеем шанс решить задачу прокрутки изображения.

Запустим Visual Studio и в окне New Project выберем шаблон Windows Forms Application Visual C#. Новую программу назовем СкроллингБольшогоРисунка. Из панели Toolbox перетащим на форму элемент управления Panel, а на него поместим элемент PictureBox. Далее перейдем на вкладку программного кода и введем текст, представленный в листинге

### **Вывод изображения на *PictureBox* с возможностью прокрутки**

```
// Программа выводит изображение из растрового файла в PictureBox,  
// размещенный на элементе управления Panel, с возможностью прокрутки  
// изображения
```

```
using System;  
using System.Drawing;  
using System.Windows.Forms;  
// Другие директивы using удалены, поскольку они не используются в данной  
// программе  
namespace СкроллингБольшогоРисунка  
{  
    public partial class Form1 : Form  
    {  
        public Form1()  
        {  
            InitializeComponent();  
        }  
        private void Form1_Load(object sender, EventArgs e)  
        {  
            this.Text = "Скроллинг";  
            // Назначаем размеры панели:  
            // panel1.Size = new Size(200, 151);  
            // Назначаем имя файла рисунка:  
            pictureBox1.Image = Image.FromFile(@"D:\Ris.JPG");  
            // или pictureBox1.Image = new Bitmap(@"D:\Ris.JPG");  
            // Размеры PictureBox в точности соответствуют изображению:  
            pictureBox1.SizeMode = PictureBoxSizeMode.AutoSize;  
            // Разрешаем прокрутку изображения:  
            panel1.AutoScroll = true;  
        }  
    }  
}
```