

## Шпаргалка по графике в Windows Forms

### 1. Основные классы и пространства имен

- System.Drawing: Основное пространство имен для работы с графикой.

Классы:

- Graphics: Для рисования на формах и элементах управления.
- Pen: Для рисования контуров.
- Brush: Для заливки фигур.
  - SolidBrush: Заливка сплошным цветом.
  - HatchBrush: Заливка с узором.
  - LinearGradientBrush: Линейный градиент.
- Image: Абстрактный класс для работы с изображениями.
- Bitmap: Для работы с растровыми изображениями.
- Color: Для работы с цветами.

### 2. Основные операции с графикой

Получение объекта Graphics:

Событие Paint:

```
private void MainForm_Paint(object sender, PaintEventArgs e) {  
    Graphics g = e.Graphics;  
    // Рисование здесь  
}
```

### 3. Рисование объектов

```
// Линия  
g.DrawLine(Pens.Blue, 0, 0, 100, 100);  
  
// Прямоугольник  
g.DrawRectangle(Pens.Red, 10, 10, 100, 50); // Контур  
g.FillRectangle(Brushes.Red, 10, 10, 100, 50); // Заполненный  
  
// Эллипс  
g.DrawEllipse(Pens.Green, 10, 10, 100, 50); // Контур  
g.FillEllipse(Brushes.Green, 10, 10, 100, 50); // Заполненный  
  
// Текст  
g.DrawString("Привет!", new Font("Arial", 14), Brushes.Black, new Point(10, 10));
```

```
// Пример рисования треугольника
private void DrawTriangle(PaintEventArgs e)
{
    Graphics g = e.Graphics;

    /*
     * Определяем точки треугольника:
     * - Верхняя вершина (100, 50)
     * - Левая нижняя вершина (50, 150)
     * - Правая нижняя вершина (150, 150)
     */
    Point[] triangle = {
        new Point(100, 50),
        new Point(50, 150),
        new Point(150, 150)
    };

    // Создаем перо для контура треугольника
    using (Pen pen = new Pen(Color.Blue, 2)) // Синий цвет, толщина 2 пикселя
    using (SolidBrush brush = new SolidBrush(Color.LightBlue)) // Кисть для заливки
    {
        /*
         * Заливаем треугольник светло-голубым цветом
         * Метод FillPolygon заполняет внутреннюю область фигуры
         */
        g.FillPolygon(brush, triangle);

        /*
         * Рисуем контур треугольника синим цветом
         * Метод DrawPolygon рисует только границы фигуры
         */
        g.DrawPolygon(pen, triangle);
    }
}
```

Работа с Pen (перо)

```
// Создание пера
using(Pen pen = new Pen(Color.Blue, 2)) // Цвет и толщина
{
    pen.DashStyle = DashStyle.Dash; // Пунктирная линия
    g.DrawLine(pen, 0, 0, 100, 100);
}

// Предопределенные перья
Pens.Black
Pens.Red
Pens.Blue
```

Работа с Brush (кисть)

```
// Сплошная заливка
using(SolidBrush brush = new SolidBrush(Color.Red))
{
    g.FillRectangle(brush, 10, 10, 100, 50);
}

// Градиентная заливка
using(LinearGradientBrush gradientBrush = new LinearGradientBrush(
    new Point(0, 0),
    new Point(100, 100),
    Color.Red,
    Color.Blue))
{
    g.FillRectangle(gradientBrush, 0, 0, 100, 100);
}

// Предопределенные кисти
Brushes.Black
Brushes.Red
Brushes.Blue
```

## Работа со шрифтом (простой текст)

```
private void Form1_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;

    // Создаем шрифт Arial размером 20 пунктов
    Font font = new Font("Arial", 20, FontStyle.Regular);

    // Текст для отображения
    string text = "Пример текста";

    // Рисуем текст черным цветом в позиции (50, 50)
    g.DrawString(text, font, Brushes.Black, new PointF(50, 50));

    /*
    * - Font: Определяет шрифт, размер и стиль текста.
    * - Brushes.Black: Используется для заливки текста черным цветом.
    * - PointF(50, 50): Координаты начальной точки текста.
    */
}
```

## Работа со шрифтом (текст с тенью)

```
private void Form1_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;

    // Создаем шрифт Times New Roman размером 24 пункта жирным начертанием
    Font font = new Font("Times New Roman", 24, FontStyle.Bold);

    // Текст для отображения
    string text = "Текст с тенью";

    // Позиция текста
    Point position = new Point(50, 50);

    // Рисуем тень
    using (SolidBrush shadowBrush = new SolidBrush(Color.FromArgb(128, 0, 0, 0)))
    {
        /*
        * Тень:
        * - Color.FromArgb(128, 0, 0, 0): Полупрозрачный черный цвет (альфа-канал 128).
        * - Смещение на 5 пикселей вправо и вниз для создания эффекта тени.
        */
        g.DrawString(text, font, shadowBrush,
            new Point(position.X + 5, position.Y + 5));
    }

    // Рисуем основной текст
    using (SolidBrush textBrush = new SolidBrush(Color.Yellow))
    {
        /*
        * Основной текст:
        * - Желтый цвет текста.
        * - Рисуется поверх тени для создания эффекта объема.
        */
        g.DrawString(text, font, textBrush, position);
    }
}
```

## 4. Работа с компонентом PictureBox

Основные свойства:

- Image: для задания изображения.
- SizeMode: определяет, как изображение будет отображаться.
  - Normal: натуральный размер.
  - StretchImage: растягивает изображение.
  - Zoom: масштабирует пропорционально.

Методы:

- Load(string path): загружает изображение.
- LoadAsync(string path): асинхронная загрузка изображения.

```
// Создание в коде
PictureBox pictureBox = new PictureBox();
pictureBox.Size = new Size(200, 200);
pictureBox.SizeMode = PictureBoxSizeMode.StretchImage;

// Загрузка изображения
pictureBox.Image = Image.FromFile("picture.jpg");
// или
pictureBox.Load("picture.jpg");

// Асинхронная загрузка
pictureBox.LoadAsync("picture.jpg");

// Режимы отображения
pictureBox.SizeMode = PictureBoxSizeMode.Normal; // Обычный размер
pictureBox.SizeMode = PictureBoxSizeMode.StretchImage; // Растянуть
pictureBox.SizeMode = PictureBoxSizeMode.AutoSize; // Автоматический размер
pictureBox.SizeMode = PictureBoxSizeMode.CenterImage; // По центру
pictureBox.SizeMode = PictureBoxSizeMode.Zoom;
```

## 5. Цвета

```
//Создание цвета с помощью ARGB:
Color customColor = Color.FromArgb(255, 0, 128, 255); // Непрозрачный синий

//Использование predefined цветов:
Brush brush = Brushes.Red; // Использование красной кисти
```

### Важные замечания:

- Всегда используйте using для объектов, реализующих IDisposable (Pen, Brush, Font)
- Для постоянного рисования используйте событие Paint
- Метод Invalidate() вызывает перерисовку форм