

получить MSDN справку набрав наименование метода в среде Visual Studio и нажав на нем клавишу F1. Также, при наборе метода в редакторе кода среда показывает краткую справку о передаваемых параметрах.

### ***9.5. Выполнение индивидуального задания***

Изучите с помощью справки MSDN методы и свойства классов **Graphics**, **Color**, **Pen** и **SolidBrush**. Создайте собственное приложение выводящий на форму рисунок, состоящий из различных объектов (линий, многоугольников, эллипсов, прямоугольников и пр.), не закрашенных и закрашенных полностью. Используйте разные цвета и стили линий (сплошные, штриховые, штрих-пунктирные).

## **ЛАБОРАТОРНАЯ РАБОТА 10. ПРОСТЕЙШАЯ АНИМАЦИЯ**

**Цель лабораторной работы:** изучить возможности Visual Studio по созданию простейшей анимации. Написать и отладить программу, выводящую на экран анимационное изображение.

### ***10.1. Работа с таймером***

Класс для работы с таймером (Timer) формирует в приложении повторяющиеся события. События повторяются с периодичностью, указанной в миллисекундах, в свойстве **Interval**. Установка свойства **Enabled** в значение **true** запускает таймер. Каждый тик таймера порождает событие **Tick**, обработчик которого обычно и создают в приложении. В этом обработчике могут изменяться какие либо величины, и вызваться принудительная перерисовка окна. Напоминаем, что вся отрисовка при создании анимации должна находиться в обработчике события **Paint**.

### ***10.2. Создание анимации***

Для создания простой анимации достаточно использовать таймер, при тике которого будут изменяться параметры изображения (например, координаты концов отрезка) и обработки события **Paint** для рисования по новым параметрам. При таком подходе не надо заботиться об удалении старого изображения (как в идеологии MS DOS), ведь оно создается в окне заново.

В качестве примера рассмотрим код анимации секундной стрелки часов:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        //описываем переменные доступные в любом обработчике событий класса Form1
        private int x1, y1, x2, y2, r;
```

```

private double a;
private Pen pen = new Pen(Color.DarkRed, 2);

public Form1()
{
    InitializeComponent();
}

private void Form1_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;
    g.DrawLine(pen, x1, y1, x2, y2); //рисует секундную стрелку
}

private void Form1_Load(object sender, EventArgs e)
{
    //определяем центр экрана
    x1 = ClientSize.Width / 2;
    y1 = ClientSize.Height / 2;
    r = 150; //задаем радиус
    a = 0; //задаем угол поворота
    //определяем конец часовой стрелки с учетом центра экрана
    x2 = x1 + (int) (r * Math.Cos(a));
    y2 = y1 - (int) (r * Math.Sin(a));
}

private void timer1_Tick(object sender, EventArgs e)
{
    a -= 0.1; //уменьшаем угол на 0,1 радиану
    //определяем конец часовой стрелки с учетом центра экрана
    x2 = x1 + (int) (r * Math.Cos(a));
    y2 = y1 - (int) (r * Math.Sin(a));
    Invalidate(); //вынудительный вызов перерисовки (Paint)
}
}
}

```

### 10.3. Выполнение индивидуального задания

Изучите с помощью справки MSDN методы и свойства классов **Graphics**, **Color**, **Pen** и **SolidBrush**. Создайте собственное приложение выводящий на форму рисунок, состоящий из различных объектов (линий, многоугольников, эллипсов, прямоугольников и пр.), не закрашенных и закрашенных полностью. Используйте разные цвета и стили линий (сплошные, штриховые, штрих-пунктирные).

- 1) Создайте программу, показывающую пульсирующее сердце.
- 2) Создайте приложение, отображающее вращающийся винт самолета.
- 3) Разработайте программу анимации движущегося человечка.
- 4) Создайте программу, показывающую движение окружности по синусоиде.
- 5) Создайте приложение, отображающее движение окружности по спирали.
- 6) Разработайте программу анимации падения снежинки.
- 7) Создайте программу, показывающую скачущий мячик.
- 8) Создайте приложение, отображающее движение окружности вдоль границы окна. Учтите возможность изменения размеров окна.
- 9) Разработайте программу анимации летающего бумеранга.

- 10) Создайте программу, показывающую падение нескольких звезд одновременно.
- 11) Создайте приложение, отображающее хаотичное движение звезды в окне.
- 12) Разработайте программу анимации взлета ракеты. Старт осуществляется по нажатию специальной «красной» кнопки.
- 13) Создайте программу, показывающую движение окружности вдоль многоугольника. Число вершин вводится пользователем до анимации.
- 14) Создайте приложение, отображающее броуновское движение молекулы в окне.
- 15) Разработайте программу анимации движения планет в солнечной системе.
- 16) Создайте программу, показывающую движение квадрата по траектории, состоящей из 100 точек, и хранящихся в специальном массиве.
- 17) Создайте приложение, имитирующие механические часы.
- 18) Разработайте программу анимации падения несколько листков с дерева. Движение не должно быть линейным.
- 19) Создайте программу, показывающую движение окружности по спирали с плавно изменяющейся скоростью.
- 20) Создайте приложение, отображающее движение автомобиля с вращающимися колесами.

## ЛАБОРАТОРНАЯ РАБОТА 11. ОБРАБОТКА ИЗОБРАЖЕНИЙ

**Цель лабораторной работы:** изучить возможности Visual Studio по открытию и сохранению файлов. Написать и отладить программу для обработки изображений.

### 11.1. Отображение графических файлов

Обычно для отображения точечных рисунков, рисунков из метафайлов, значков, рисунков из файлов в формате BMP, JPEG, GIF или PNG используется объект **PictureBox**, т.е. элемент управления **PictureBox** действует как контейнер для картинок. Можно выбрать изображение для вывода, присвоив значение свойству **Image**. Свойство **Image** может быть установлено в окне **Свойства** или в коде программы, указывая на рисунок, который следует отображать.

Элемент управления **PictureBox** содержит и другие полезные свойства, в том числе: **AutoSize** определяющее, будет ли изображение растянуто в элементе **PictureBox**, и **SizeMode**, которое может использоваться для растягивания, центрирования или увеличения изображения в элементе управления **PictureBox**.

Перед добавлением рисунка к элементу управления **PictureBox** в проект обычно добавляется файл рисунка в качестве *ресурса*. После добавления ресурса к проекту можно повторно использовать его. Например, может потребоваться отображение одного и того же изображения в нескольких местах.

Необходимо отметить, что поле **Image** само является классом для работы с изображениями, у которого есть свои методы. Например, метод **FromFile** используется для загрузки изображения из файла. Кроме класса **Image** существует