**Пространство имен System.Drawing**(Рисование) обеспечивает доступ к функциональным возможностям графического интерфейса

**Пространства имен (дополнительно):**

 System.Drawing.Drawing2D,

System.Drawing.Imaging,

System.Drawing.Printing,

System.Drawing.Text

**Способы задания «холста»**

1. Графический объект — «холст» для рисования на форме Form1 (указатель this) можно задать, например, одним оператором:

Graphics g = this.CreateGraphics();

**Примечание. Заметим, что стандартным образом**

**Graphics g = new Graphics();**

**создать объект-холст не удается.**

**На этом операторе генерируется ошибка:**

**Для типа «System.Drawing.Graphics» не определен конструктор.**

**Объекты других классов из библиотеки System.Drawing**

Такие классы, как Rectangle (прямоугольник) и Point (точка). Отметим, что Point вообще то является структурой (struct) с полями x,y. Это уточнение несущественно, так как в C# структуры похожи на классы, a инициализация объекта-структуры point может выглядеть так же, как инициализация объекта-класса:

Point point= new Point();

Класс Pen (перо) используется для рисования линий и кривых, а классы, производные от класса Brush (кисть) используются для закраски замкнутых контуров.

Класс GraphicsPath позволяет задавать последовательность соединенных линий и кривых, класс Region описывает внутреннюю часть графической формы, состоящей из многоугольников и контуров.

**Класс Graphics**

Этот класс не наследуется. Методов в этом классе огромное количество, поэтому сначала представим их в таблице, а затем рассмотрим некоторые из них с примерами и пояснениями.

В третьем столбце таблицы указывается число перегрузок метода, различающихся набором параметров (используйте интеллектуальную подсказку IntelliSense для выбора нужного Вам варианта метода).

**Методы класса Graphics**

Clear(Color) Очищает всю поверхность рисования и выполняет заливку поверхности указанным цветом фона.

CopyFromScreen(Point, Point, Size) Выполняет передачу данных о цвете, соответствующих прямоугольной области пикселей, блоками битов с экрана на поверхность рисования объекта Graphics.

Dispose() Освобождает все ресурсы, используемые данным объектом Graphics.

DrawArc(Pen, Rectangle, Single, Single) Рисует дугу, которая является частью эллипса, заданного структурой Rectangle.

DrawBezier(Pen, Point, Point, Point, Point) Рисует кривую Безье, определяемую четырьмя структурами Point.

DrawBeziers(Pen, Point[]) Рисует несколько (N) кривых Безье, определяемых массивом из (3N+1) структур Point.

DrawCloseCurve(Pen, Point[ ]) Рисует замкнутый фундаментальный сплайн

DrawEllipse(Pen, Rectangle) Рисует эллипс

DrawIcon(Icon, Rectangle) Рисует значок

DrawImage(Image image, int x, int y) Рисует заданное изображение image, используя его фактический размер в месте с координатами (x,y)

DrawLine(Pen, Point, Point) Проводит линию, соединяющую две структуры Point.

DrawLines(Pen, Point[ ]) Рисует набор сегментов линий, которые соединяют массив структур Point.

DrawPath(Pen, gp) Рисует пером Pen объект GraphicsPath gp. 1

DrawPie(Pen, Rectangle, Single, Single) Рисует сектор, который определяется эллипсом, заданным структурой Rectangle и двумя радиалtьными линиями.

DrawPolygon(Pen, Point[]) Рисует многоугольник, определяемый массивом структур Point.

DrawRectangle(Pen, Rectangle) Рисует прямоугольник, определяемый структурой Rectangle.

DrawRectangles(Pen, Rectangle[]) Рисует набор прямоугольников, определяемых структурами Rectangle.

DrawString(String, Font, Brush, PointF) Создает указываемую текстовую строку в заданном месте с помощью определяемых объектов Brush и Font.

Equals(Object) Определяет, равен ли заданный объект текущему объекту. (Унаследовано от Object.)

ExcludeClip(Rectangle) Обновляет вырезанную область данного объекта Graphics, чтобы исключить из нее часть, определяемую структурой Rectangle.

ExcludeClip(Region) Обновляет вырезанную область данного объекта Graphics, чтобы исключить из нее часть, определяемую структурой Region.

FillClosedCurve(Brush, Point[]) Заполняет внутреннюю часть замкнутой фундаментальной кривой, определяемой массивом структур Point.

FillEllipse(Brush, Rectangle) Заполняет внутреннюю часть эллипса, определяемого ограничивающим прямоугольником, который задан структурой Rectangle.

FillPath(Brush, GraphicsPath) Заполняет внутреннюю часть объекта GraphicsPath.

FillPie(Brush, Rectangle, Single, Single) Заполняет внутреннюю часть сектора, определяемого эллипсом, который задан структурой RectangleF, и двумя радиальными линиями.

FillPolygon(Brush, Point[]) Заполняет внутреннюю часть многоугольника, определяемого массивом точек, заданных структурами Point.

FillRectangle(Brush, Rectangle) Заполняет внутреннюю часть прямоугольника, определяемого структурой Rectangle.

FillRegion(Brush, Region) Заполняет внутреннюю часть объекта Region.

Flush() Вызывает принудительное выполнение всех отложенных графических операций и немедленно возвращается, не дожидаясь их окончания.

IntersectClip(Region) Обновляет вырезанную область данного объекта, включая в нее пересечение текущей вырезанной области и указанной структуры

ResetClip() Сбрасывает выделенную область g, делая ее бесконечной

**Класс Pen**

Класс Pen определяет объект, используемый для рисования прямых линий и кривых. Этот класс не наследуется. Конструкторы класса (тоже пример перегрузки методов):

1) Pen(Color) инициализирует новый экземпляр класса Pen с указанным цветом.

2) Pen(Color, Single) инициализирует новый экземпляр класса Pen с указанными свойствами Color и Width. (Width — устанавливает ширину пера Pen, в единицах объекта Graphics, используемого для рисования

Например:

Pen redPen = new Pen(Color.Red); // толщина пера по умолчанию 1 пиксель

или используя промежуточную переменную green

Color green = Color.Green;

Pen greenPen = new Pen(green, 4.5f);

Можно вводить новый объект без указания явного имени пера (пример создания динамического объекта — пера):

g.DrawRectangle(new Pen(Color.Red, 3), r);

например, для рисования прямоугольника r красным пером толщиной 3 пикселя, используя графический контекст («холст») g.

**Класс Brush**

Класс Brush определяет объекты, которые используются для заливки внутри графических фигур, таких как прямоугольники, эллипсы, круги, многоугольники и дорожки. Это абстрактный базовый класс, который не может быть реализован. Для создания объекта «кисть» используются классы, производные от Brush, такие как SolidBrush, TextureBrush и LinearGradientBrush, например:

SolidBrush br = new SolidBrush(Color.Aqua);

g.FillRectangle(br, r); // закраска прямоугольника r на «холсте» g

Или без явного объявления объекта «кисть» br:

g.FillRectangle(new SolidBrush(Color.Red), r);

**Вызов класса методов класса Graphics:**

**1 способ:**

**1.** SizeF messageSize = **e.Graphics**.MeasureString(HELLO\_WORLD, Font);

Graphics g = Graphics.FromHwnd(Handle); // получение статич. метода для

рисования

g.DrawRectangle(SystemPens.ActiveBorder, new Rectangle(10, 10, ClientSize.Width - 20, ClientSize.Height - 20));

**2 способ:**

Graphics g = this.CreateGraphics(); - вызов через this метода для рисования

**Получение координат мыши:**

**1. private** **void** Form1\_MouseMove(**object** sender, MouseEventArgs e)

        {

            x1 = x;

            y1 = y;

            x2 = e.X;

            y2 = e.Y;

                   }

**2.**

ToolTip trackTip;

private void TrackCoordinates()

{

trackTip = new ToolTip();

this.MouseMove += new MouseEventHandler(Form1\_MouseMove);

}

void Form1\_MouseMove(object sender, System.Windows.Forms.MouseEventArgs e)

{

String tipText = String.Format("({0}, {1})", e.X, e.Y);

trackTip.Show(tipText, this, e.Location);

Отображение координат мыши во всплывающем окне