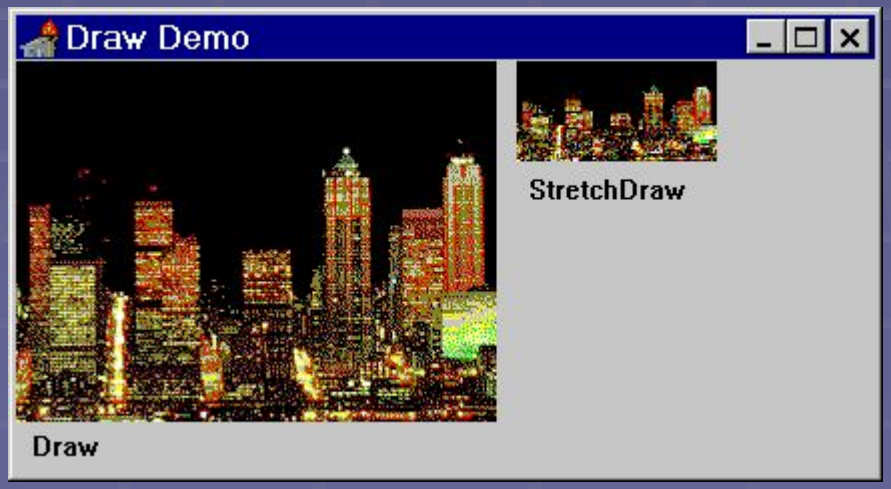
Отображение графической информации в Delphi



Среда визуального программирования Delphi, как и Windows, поддерживает графический интерфейс пользователя (GDI – Graphic Delphi Interface).

В Delphi существует два способа вывода графической информации:

- вывод заранее подготовленных изображений;
- рисование из программы.

Первый способ основан на использовании компонентов **Image** и **Shape**. Можно воспользоваться готовой картинкой(пиктограммой) или создать их самостоятельно, используя Редактор Изображений **Image Editor.**

Второй способ – это формирование изображений программным способом, используя **объект Canvas.**

Delphi имеет в своём распоряжении специальный объект, который оформлен в виде свойства *Canvas*. Оно доступно *только во время работы приложения*, так что управлять им можно только из программы, написав нужный код на языке Object Pascal.

Если у объекта есть свойство *Canvas*, на его поверхности можно рисовать. Наиболее подходящими кандидатами на эту роль являются - сама форма и специальный компонент **PaintBox.**

Объект Canvas

Свойства:

Pen (Перо) –свойство для рисования линий и границ геометрических фигур. Перо следует командам графического курсора и, в свою очередь, имеет свои вложенные свойства:

- *Color* – определяет цвет линии (по умолчанию чёрный);
- *Mode* – стиль рисования (имеет множество значений, которые здесь не приводятся);
- *Style* – стиль линии, который может принимать значения:
  psSolid – сплошная (по умолчанию);
  psDosh – штриховая;
  psDot – пунктирная;
  psDoshDot – штрих пунктирная (и др.свойства);

- *Widh* – толщина линии (по умолчанию 1 пиксель);

Brush (Кисть) – свойство для заполнения фигур, имеющие следующие вложенные свойства :

- *Color* – цвет кисти (по умолчанию – белый);
- *Style* – орнамент кисти, который может принимать значения:

bsClear – сплошная раскраска;

bsHorizontal – горизонтальные линии;

bsVertical – вертикальные линии;

bsFDiagonal – левые диагональные линии;

bsBDiagonal – правые диагональные линии;

bsCross – клетка;

bsDiagCross – косая клетка;

Font (Шрифт) – свойство для вывода текста, имеющее следующие вложенные свойства :

- *Color* – цвет символов;
- *Height* – высота шрифта в пикселях;
- *Name* – имя шрифта;
- *Size* – размер шрифта;
- *Style* –стиль шрифта, который может принимать следующие значения:

fsBold – полужирный;

fsItalic – курсив;

fsUnderline – подчёркнутый;

fsStrikeOut – перечёркнутый;

***PenPos* (*Позиция пера*)** – свойство для хранения текущей позиции рисования (определяет положение пера в области рисования в данный момент времени);

Pixels [x,y: integer] –свойство-массив для записи и считывания координат отдельных точек области рисования («холста»).

Методы объекта Canvas

- **MoveTo(x,y: integer)** –перемещает перо с текущей позиции в точку с заданными координатами **x, y** без рисования линии;
- **LineTo(x.y: integer)** -перемещает перо с текущей позиции в точку с заданными координатами **x, y** с прочерчиванием линии;
- **Arc(x1, y1, x2, y2, x3, y3, x4, y4: integer)** – рисует дугу эллипса, вписанного в прямоугольник с координатами **(x1,y1)** и **(x2,y2).** Дуга определяется радиусами эллипса, проходящими через точки **(x3,y3) и (x4,y4**);

- **Chord(x1, y1, x2, y2, x3, y3, x4, y4: integer)** – рисует хорду эллипса по описанию, приведённому для метода Arc;
- **Ellipse(x1, y1, x2, y2: integer)** – рисует эллипс, вписанный в прямоугольник с левым верхним углом в точке **(x1, y1)** и нижним правым углом в точке **(x2, y2)**;
- **FillRect(Rect (x1, y1, x2, y2: integer))** – заполняет прямоугольник цветом, заданным в текущей кисти (Brush). Использует функцию *Rect*, которая представляет прямоугольник с заданными координатами;

- **FloodFill(x,y : integer; Color: TColor; FillStyle: TFillStyle)** – заполнение текущим цветом, заданным в свойстве Brush:

  при *FillStyle=fsBorder* – замкнутой области от точки с координатами **x, y** до границы, определённой цветом Color;

  при *FillStyle=fsSurface* – тот участок поверхности, который имеет цвет Color;

- **Pie(x1, y1, x2, y2, x3, y3, x4, y4: integer)** – рисует сектор эллипса, вписанного в прямоугольник с координатами **(x1, y1)** и **(x2, y2).** Сектор определяется двумя радиусами эллипса, проходящими через точки **(x3,y3)** и **(x4, y4)**;

- **Polyline (Points: array of TPoint)** – рисует ломаную линию, последовательно соединяя точки массива **Points**;

- **Polygon (Points: array of TPoint)** – вычерчивает многоугольники, последовательно соединяя точки массива **Points**. Отличается от метода Polyline тем, что автоматически соединяет конец ломаной с её начало;

- **Rectangle (x1, y1, x2, y2: integer)** – рисует прямоугольник с левым верхним углом в точке **(x1, y1)** и нижним правым углом в точке **(x2,y2**);

- **Retresh** –метод вызывается при необходимости перерисовки изображения;

- **RoundRect (x1, y1, x2, y2, x3, y3: integer)** – рисует прямоугольник с закруглёнными углами. Углы рисуются как четверти эллипса с шириной **x3** и высотой **y3**;

- **TextOut (x, y :integer, Text :String**) –вывод текста, указанного в параметре *Text*. Текст вписывается в прямоугольник, верхний левый угол которого имеет координаты **x, y.**