

Практическая работа №9, Генератор случайных чисел

Постановка задачи

Создать проект, который будет выводить на экран окружности, генерируя случайным образом координаты (X, Y) и радиус R.

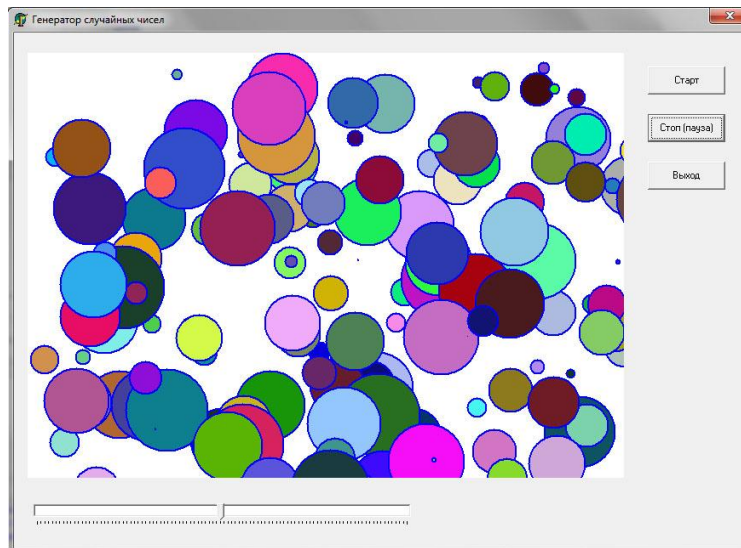


Рис.1

Новыми в этой программе являются:

- построение графических изображений (окружностей), использование свойства **Canvas**,
- компонента **TrackBar** – элемент управления в виде ползунка, который пользователь может перемещать курсором мыши или клавишами во время выполнения.

Немного теории

Построение графических изображений осуществляется на поверхность объекта (формы или др.компонент). У ряда объектов из библиотеки визуальных компонент есть свойство **Canvas**. Для того чтобы вывести на поверхность объекта графический элемент (прямую линию, окружность, прямоугольник и т. д.), необходимо применить к свойству **Canvas** этого объекта соответствующий метод.

Canvas является в свою очередь объектом, объединяющим в себе поле для рисования, карандаш (**Pen**), кисть (**Brush**) и шрифт (**Font**). **Canvas** обладает также рядом графических методов: **LineTo**, **Ellipse**, **Rectangle**, **TextOut**, **Arc**, и др. Эти методы обеспечивают вывод графических примитивов (линий, окружностей, прямоугольников, текстов и т. д.), а свойства позволяют задать характеристики выводимых графических примитивов: цвет, толщину и стиль линий, цвет и вид заполнения областей, характеристики шрифта при выводе текстовой информации.

Методы вывода графических примитивов рассматривают свойство **Canvas** как некоторый абстрактный холст, на котором они могут рисовать (*canvas* переводится как «поверхность», «холст для рисования»).

Холст состоит из отдельных точек – пикселей. Положение пикселя характеризуется его горизонтальной (X) и вертикальной (Y) координатами. Левый верхний пиксель имеет координаты (0, 0). Координаты возрастают сверху вниз и слева направо.

Значения координат правой нижней точки холста зависят от размера холста.

Объект.Canvas.Ellipse(x1,y1, x2,y2)

где: объект – имя объекта (компонента), на поверхности которого выполняется вычерчивание,

x1, y1, x2, y2 – координаты прямоугольника, внутри которого вычерчивается эллипс или, если прямоугольник является квадратом, окружность.

Свойство **Canvas** доступно только во время работы приложения.

Canvas.Brush – свойство фона.

Свойство **Canvas.Brush.Color** определяет необходимый цвет

Инструкция **Canvas.FillRect(ClientRect)** очищает всю рабочую область компонента, заливая ее установленным ранее цветом.

Свойство **Canvas.Brush.Bitmap** устанавливает рисунок в качестве фона – присвоить переменную (имя файла) с растровым рисунком.

Canvas.MoveTo(x,y) – устанавливает перо в заданную точку, где x и y - координаты точки, относительно компонента. После этой команды перо установлено, но точка не нарисована.

Canvas.LineTo(x,y) – провести линию от текущего положения пера до заданной точки (x,y).

Canvas.Pixels[x,y]:=ЦВЕТ_ТОЧКИ – поставить на холсте точку с координатами (x,y) определенного цвета.

Canvas.Pen – свойства линии.

Canvas.Pen.Width:=ТОЛЩИНА_В_ТОЧКАХ.

Canvas.Pen.Color:=ЦВЕТ.

План разработки программы

1. Откройте новый проект.
2. Разместите в форме экземпляры компонентов в соответствии с рис. 2.

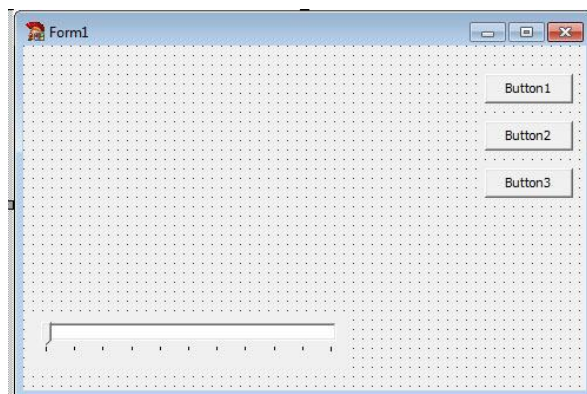


Рис.2

3. Сохраните код программы и проект под именами, например, **Unit15.pas** и **Pr15.dpr**.

4. Выполните следующие действия:

Выделенный объект	Вкладка окна Object Inspector	Имя свойства/ Имя события	Значение/Действие
Form1	Properties	Position	poDesktopCenter при каждом запуске и при любом экранном разрешении всегда объект будет ровно посередине экрана
		BorderStyle	bsSingle установка неизменяемого размера окна
		BorderIcons	Во вложенных свойствах BorderIcons устанавливаем для biSystemMenu значение true, остальные – false. Значение свойства определяет, какие кнопки управления окном будут доступны пользователю во время работы программы. В нашем случае: окно нельзя свернуть в значок, развернуть во весь экран, окно имеет иконку в заголовке.
PaintBox (System)	Properties	Height	500
		Width	600
Button1	Properties	Caption	Старт
Button2	Properties	Caption	Стоп (пауза)
Button3	Properties	Caption	Выход
	Events	OnClick	Close;
TrackBar (Win32)	Properties	Min	1 Минимальное значение ползунка.
		Max	100 Максимальное значение ползунка.
		Position	50 Текущее значение(положение) ползунка.

Сразу после слова **implementation** в модуле окна объявляем глобальные переменные, которые будут доступны из любой процедуры в этом модуле, и процедура для формирования псевдослучайного числа.

```

Const    NX = 700 ; //максимальный размер формы по X
           NY = 500 ; //максимальный размер формы по Y
           NR = 100 ; //максимальный размер радиуса

Var      STOP : Boolean; // флаг рисования или прекращения рисования
           X, Y, R : Integer; // координаты окружности и текущий радиус

Procedure RAND (NA: Integer; Var AA: Integer) ;
begin
AA:= Random(NA) ;
end;
    
```

Выделенный объект	Вкладка окна Object Inspector	Имя свойства/ Имя события	Значение/Действие
Button2	Events	OnClick	STOP:=true; установить флаг остановки процесса рисования
Form1	Events	OnClose реакция перед закрытием окна	Stop:=true; остановить (если включен) цикл рисования

Реакция на событие нажатия на кнопку Button1 (Начало рисования)

```

procedure TForm1.Button1Click(Sender: TObject);
begin
    PaintBox1.Canvas.Brush.Color:= clWhite; // цвет фона белый
    PaintBox1.Canvas.FillRect(ClientRect); // заливка всей рабочей области
    STOP:=false; // флаг старта процесса рисования
    randomize;
    While not STOP do // бесконечный цикл, пока флаг остановки не будет равен TRUE
        begin
            PaintBox1.Canvas.Pen.Color:= clBlue; // цвет карандаша голубой
            PaintBox1.Canvas.Pen.Width:=2; // толщина карандаша
            RAND (NX,X); // обращение к подпрограмме для формирования
            RAND (NY,Y); // чисел случайным образом
            RAND (NR,R);
            PaintBox1.Canvas.Ellipse(X,Y,X+R,Y+R); // рисуем окружность

            Sleep(TrackBar1.Position); // Процедура «засыпает» на заданное время в
                // миллисекундах; Sleep(1000) – задержка на 1 сек
            Application.ProcessMessages; // Обработка всей очереди сообщений
        end;
    end;
    
```

Пояснения

Как только нажата кнопка **Старт**, компонент **PaintBox1** очищается (заливается белым цветом) и начинается бесконечный цикл **While**. Выйти из этого цикла можно только тогда, когда переменная **STOP** примет значение **True**. Это можно сделать, нажав кнопки **Стоп (пауза)** – приостановить работу или **Выход** – завершить работу. Соответствующая процедура обработается во время **Application.ProcessMessages**. С помощью бегунка **TrackBar1** можно менять скорость рисования кругов. Значение этого параметра является параметром процедуры **Sleep**.

Задание

1. Внесите изменения в программу, чтобы круги были цветные. Необходимо задать цвет кисти перед тем, как рисовать круг.
2. Внесите изменения в программу, чтобы цвет у кругов был различный. Цвет можно задавать как шестнадцатеричное число в диапазоне от \$000000 до \$FFFFFF. Для формирования случайного цвета круга можно воспользоваться созданной процедурой `RAND ($FFFFFF, X) ;`.
3. Внесите изменения в программу, чтобы на экран выводилось 10 окружностей, затем первая окружность стиралась и выводилась на экран следующая, которая становится 10-ой. Этот процесс повторяется до нажатия клавиши.

Примечание

TTrackBar - компонент

Элемент управления в виде ползунка, который пользователь может перемещать курсором мыши или клавишами во время выполнения.

Свойство	Описание
Position	При перемещении пользователем ползунка можно прочитать значение Position , характеризующее позицию, в которую пользователь переместил ползунок. Position - целое, значение которого может изменяться в пределах, задаваемых свойствами.
Min и Max	Минимальное и максимальное значение Position .
Orientation	Определяет ориентацию ползунка
TickMarks	Указывает размещение шкалы относительно компонента
TickStyle	Определяет способ изображения шкалы. При TickStyle = tsAuto (автоматическое проставление меток) частота меток определяется свойством Frequency . Это свойство задает, сколько возможных значений Position лежит между метками.
LineSize и PageSize	Определяют, насколько смещается ползунок, если пользователь управляет им с помощью соответственно клавиш со стрелками или клавишами PageUp и PageDown .
SelStart и SelEnd	Позволяют визуально выделить на шкале некоторый диапазон, который о чем-то говорит пользователю, например, рекомендуемый диапазон значений.