

## События визуальных компонент

### Цели:

- Закрепить понятие о событийно-управляемом интерфейсе;
- Закрепить навыки создания обработчиков событий;
- Получить представление о событиях отображения компонентов VCL.

## События визуальных компонент

- Отображение визуальных компонент.

Визуальные компоненты **VCL** – это компоненты, способные отображать себя на экране. Такие компоненты имеют холст, на котором они нарисованы (свойство **Canvas** типа **TCanvas**).

Все визуальные компоненты являются наследниками **TWinControl**.

Класс **TForm** имеет событие, отвечающее за отображение этого компонента на экране (**OnPaint**).

## События визуальных компонент

- TCanvas.

**TCanvas** (холст) – это класс, содержащий область, на которой изображен компонент в виде массива. Каждый элемент этого массива имеет тип **TColor**, содержащий цвет соответствующей точки (пикселя) холста.

Данный массив находится в свойстве **Pixels** класса **TCanvas**.

**TColor** задает цвет пикселя в виде **RGB(r,g,b)**. Здесь **r,g** и **b** являются числами от 0 до 255, задающими интенсивность оттенков **красного**, **зеленого** и **синего** соответственно.

RGB(0,0,0) соответствует черному цвету.

RGB(255,255,255) соответствует белому цвету.

- Рисование на холсте.

Пиксельная графика позволяет рисовать непосредственно на компоненте, с помощью задания значений массива **Pixels** свойства **Canvas** компонента.

**Создадим в Delphi новый проект**

Добавим на форму компонент **TButton** и зададим обработчик **OnClick**.

```
procedure TForm1.Button1Click(Sender: TObject);
var
  x,y : integer;
begin
  for x:=1 to 100 do
    for y:=1 to 100 do
      Form1.Canvas.Pixels[x,y]:=RGBToColor(123,0,123);
end;
```

В результате на форме отобразиться сиреневый квадрат. Однако, если поверх формы вывести другое окно, а затем опять переключиться на форму – квадрат пропадет. Это происходит потому, что окна **перерисовывают** свой холст. В процессе перерисовки возникает событие **OnPaint**, соответственно, осуществив рисование в обработчике этого события, рисунок будет появляться после каждой перерисовки объекта.

```
procedure TForm1.FormPaint(Sender: TObject);  
begin  
  for x:=1 to 100 do  
    for y:=1 to 100 do  
      Form1.Canvas.Pixels[x,y]:=RGBToColor(123,0,123);  
end;
```

- Битовые карты (bitmap).

**TBitmap** — это компонент, служащий для хранения графических изображений в виде битовых карт. Данный компонент не является визуальным, соответственно он только хранит битовую карту, а не отображает ее.

Данный компонент **отсутствует на палитре компонент**, соответственно для использования, необходимо создавать его **динамически**.

Битовые карты можно хранить в виде файлов (**bmp, dib**). Компонент **TBitmap** имеет методы **LoadFromFile** и **SaveToFile**, позволяющие осуществить это. В качестве параметра данным методам передается **имя файла**.

## Desktop для формы

Запустите Delphi и создайте новое приложение. В раздел **var** модуля добавьте описание переменной для создаваемой картинки:

```
bmp : TBitmap;
```

Для инициализации картинки создайте событие **OnCreate** формы:

```
procedure TForm1.FormCreate(Sender: TObject);  
begin  
    bmp:=TBitmap.Create;  
    bmp.LoadFromFile('C:\WINDOWS\Coffee Bean.bmp');  
end;
```

Зададим обработчик для вывода изображения:

```
procedure TForm1.FormPaint(Sender: TObject);  
begin  
    Form1.Canvas.StretchDraw(Rect(0,0,Width,Height),bmp);  
end;
```

Метод `StretchDraw` объекта `Canvas` позволяет выводить на холст графическое изображение целиком, масштабируя его по размерам заданной с помощью `Rect()` области.

Так как переменная `bmp` создавалась динамически, необходимо уничтожить ее после завершения работы программы.

```
procedure TForm1.FormClose(Sender: TObject; var Action:  
TCloseAction);  
begin  
    bmp.Free;  
end;
```



# Готовое приложение

Скомпилируем и запустим приложение. На форме появится изображение, которое будет перерисовываться каждый раз, когда форма перерисовывается.

Однако, когда изменяется размер формы – картинка деформируется. При изменении размера компонента генерируется событие `OnResize`. Если в обработчике этого события вызвать перерисовку компонента (метод `Paint`), изменение размеров формы будет осуществляться корректно.

## Дополнительные задания

- 1. Задайте событие `OnResize`, чтобы картинка корректно растягивалась на всю форму при изменении ее размеров.*
- 2. Попробуйте осуществить вывод 2х картинок, каждую на одну половину формы.*