

Динамическое создание и удаление компонентов – Delphi vs C#

Таблица №1. Создание одного компонента.

```

Глобальные переменные
  tx, ty: Integer;
  lbl: TLabel;

procedure TForm1.FormCreate(Sender: TObject);
begin
  lbl:=TLabel.Create(self);
  lbl.Name := 'Label_01';
  lbl.Caption := 'Label_01';
  lbl.Top := 15; lbl.Left := 15;

  lbl.OnMouseDown := label_MouseDown;
  lbl.OnMouseMove := label_MouseMove;

  lbl.Parent:=self;
end;

procedure TForm1.label_MouseDown(Sender: TObject; Button:
TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
  tx := x; ty := y;
end;

procedure TForm1.label_MouseMove(Sender: TObject; Shift:
TShiftState; X, Y: Integer);
begin
  if Shift=[ssLeft] then
  begin
    lbl.Left := lbl.Left+X-tx;
    lbl.Top := lbl.Top+Y-ty;
  end;
end;

```

```

Глобальные переменные
  int x, y;
  Label lbl;

public Form1()
{
  InitializeComponent();
  FormCreate();
}

private void FormCreate()
{
  lbl = new Label();
  lbl.Name = "label_01";
  lbl.Text = "label_01";
  lbl.Top = 15; lbl.Left = 15;

  lbl.MouseDown += new MouseEventHandler(label_MouseDown);
  lbl.MouseMove += new MouseEventHandler(label_MouseMove);

  this.Controls.Add(lbl);
}

private void label_MouseDown(object sender, MouseEventArgs e)
{
  x = e.X; y = e.Y;
}

private void label_MouseMove(object sender, MouseEventArgs e)
{
  if (e.Button == MouseButtons.Left)
  {
    lbl.Left = lbl.Left + e.X - x;
    lbl.Top = lbl.Top + e.Y - y;
  }
}

```

Таблица №2. Создание статического массива компонентов.

Глобальные значения

```

const count = 5;
    tx, ty: Integer;
    albl: array[0..4] of TLabel;

procedure TForm1.FormCreate(Sender: TObject);
var i: Integer;
begin
    for i:=0 to 4 do
        begin
            albl[i]:=TLabel.Create(self);
            albl[i].Name:='Label_0'+IntToStr(i);
            albl[i].Caption:='Label_0'+IntToStr(i);
            albl[i].Top:=15+25*i; albl[i].Left:=15;

            albl[i].MouseDown:=label_MouseDown;
            albl[i].MouseMove:=label_MouseMove;
            albl[i].Parent:=self;
        end;
    end;

procedure TForm1.label_MouseDown(Sender: TObject; Button:
TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
    tx := x; ty := y;
end;

procedure TForm1.label_MouseMove(Sender: TObject; Shift:
TShiftState; X, Y: Integer);
begin
    if Shift=[ssLeft] then
        begin
            (Sender as TLabel).Left:=(Sender as TLabel).Left+X-tx;
            (Sender as TLabel).Top:=(Sender as TLabel).Top+Y-ty;
        end;
    end;
end;

```

Глобальные значения

```

const int count = 5;
    int x, y;
    Label[] albl = new Label[count];

public Form1()
{
    InitializeComponent();
    FormCreate();
}

private void FormCreate()
{
    for (int i = 0; i < count; i++)
    {
        albl[i] = new Label();
        albl[i].Name = "label_0"+Convert.ToString(i);
        albl[i].Text = "label_0" + Convert.ToString(i);
        albl[i].Top = 15+25*i; albl[i].Left = 15;

        albl[i].MouseDown += new MouseEventHandler(label_MouseDown);
        albl[i].MouseMove += new MouseEventHandler(label_MouseMove);
        this.Controls.Add(albl[i]);
    }
}

private void label_MouseDown(object sender, MouseEventArgs e)
{
    x = e.X; y = e.Y;
}

private void label_MouseMove(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButtons.Left)
    {
        (sender as Label).Left = (sender as Label).Left + e.X - x;
        (sender as Label).Top = (sender as Label).Top + e.Y - y;
    }
}

```

Таблица №3. Создание динамического массива компонентов.

Глобальные переменные

```
tx, ty, count: Integer;
albl: array of TLabel;
```

```
procedure TForm1.FormCreate(Sender: TObject);
begin
count:=0; SetLength(albl,count);
end;
```

```
procedure TForm1.FormMouseDown(Sender: TObject; Button:
TMouseButton; Shift: TShiftState; X, Y: Integer);
var i: integer;
begin
if Shift=[ssMiddle] then
begin
Inc(count); i:=count-1;
SetLength(albl,count);
albl[i]:=TLabel.Create(self);
albl[i].Name:='Label_0'+IntToStr(i);
albl[i].Caption:='Label_0'+IntToStr(i);
albl[i].Top:=15+25*i; albl[i].Left:=15;

albl[i].OnMouseDown:=label_MouseDown;
albl[i].OnMouseMove:=label_MouseMove;
albl[i].Parent:=self;
end;
end;
```

```
procedure TForm1.label_MouseDown(Sender: TObject; Button:
TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
tx := x; ty := y;
end;
```

```
procedure TForm1.label_MouseMove(Sender: TObject; Shift:
TShiftState; X, Y: Integer);
begin
if Shift=[ssLeft] then
begin
(Sender as TLabel).Left:=(Sender as TLabel).Left+X-tx;
(Sender as TLabel).Top:=(Sender as TLabel).Top+Y-ty;
end;
end;
```

Глобальные переменные

```
int x, y;
List<Label> albl = new List<Label>();
```

```
public Form1()
{
InitializeComponent();
}
```

```
private void Form1_MouseDown(object sender, MouseEventArgs e)
{
if (e.Button == MouseButtons.Middle)
{
albl.Add(new Label());
int i = albl.Count - 1;

albl[i].Name = "label_0" + Convert.ToString(i);
albl[i].Text = "label_0" + Convert.ToString(i);
albl[i].Top = 15 + 25 * i; albl[i].Left = 15;

albl[i].MouseDown += new MouseEventHandler(label_MouseDown);
albl[i].MouseMove += new MouseEventHandler(label_MouseMove);
this.Controls.Add(albl[i]);
}
}
```

```
private void label_MouseDown(object sender, MouseEventArgs e)
{
x = e.X; y = e.Y;
}
```

```
private void label_MouseMove(object sender, MouseEventArgs e)
{
if (e.Button == MouseButtons.Left)
{
(sender as Label).Left = (sender as Label).Left + e.X - x;
(sender as Label).Top = (sender as Label).Top + e.Y - y;
}
}
```

Таблица №4. Удаление элементов из динамического массива компонентов.

```

Глобальные переменные
  tx, ty, count: Integer;
  albl: array of TLabel;

procedure TForm1.FormCreate(Sender: TObject);
begin
  count:=0;  SetLength(albl,count);
end;

procedure TForm1.FormMouseDown(Sender: TObject; Button:
TMouseButton; Shift: TShiftState; X, Y: Integer);
var i: integer;
begin
  if Shift=[ssMiddle] then
  begin
    Inc(count); i:=count-1;
    SetLength(albl,count);
    albl[i]:=TLabel.Create(self);
    albl[i].Name:='Label_0'+IntToStr(i);
    albl[i].Caption:='Label_0'+IntToStr(i);
    albl[i].Top:=15+25*i; albl[i].Left:=15;

    albl[i].OnMouseDown:=label_MouseDown;
    albl[i].OnMouseMove:=label_MouseMove;
    albl[i].Parent:=self;
  end;
  if Shift=[ssRight] then
  begin
    Dec(count); albl[count].Free;
    SetLength(albl,count);
  end;
end;

procedure TForm1.label_MouseDown(Sender: TObject; Button:
TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
  tx := x; ty := y;
end;

procedure TForm1.label_MouseMove(Sender: TObject; Shift:
TShiftState; X, Y: Integer);
begin
  if Shift=[ssLeft] then
  begin
    (Sender as TLabel).Left:=(Sender as TLabel).Left+X-tx;
    (Sender as TLabel).Top:=(Sender as TLabel).Top+Y-ty;
  end;
end;

```

```

Глобальные переменные
  int x, y;
  List<Label> albl = new List<Label>();

public Form1()
{
  InitializeComponent();
}

private void Form1_MouseDown(object sender, MouseEventArgs e)
{
  if (e.Button == MouseButtons.Middle)
  {
    albl.Add(new Label());
    int i = albl.Count - 1;

    albl[i].Name = "label_0" + Convert.ToString(i);
    albl[i].Text = "label_0" + Convert.ToString(i);
    albl[i].Top = 15 + 25 * i; albl[i].Left = 15;

    albl[i].MouseDown += new MouseEventHandler(label_MouseDown);
    albl[i].MouseMove += new MouseEventHandler(label_MouseMove);
    this.Controls.Add(albl[i]);
  }
  if (e.Button == MouseButtons.Right && albl.Count > 0)
  {
    albl[albl.Count - 1].Dispose();
    albl.RemoveAt(albl.Count - 1);
  }
}

private void label_MouseDown(object sender, MouseEventArgs e)
{
  x = e.X; y = e.Y;
}

private void label_MouseMove(object sender, MouseEventArgs e)
{
  if (e.Button == MouseButtons.Left)
  {
    (sender as Label).Left = (sender as Label).Left + e.X - x;
    (sender as Label).Top = (sender as Label).Top + e.Y - y;
  }
}

```