

Балтийский государственный технический университет «Военмех» им. Д. Ф. Устинова

Кафедра И5
«Информационные системы и программная инженерия»

ЛАБОРАТОРНАЯ РАБОТА № 4
По дисциплине «Визуальное программирование»
На тему
**«РАБОТА С ГРАФИКОЙ. КОМПОНЕНТ CHART И КЛАСС
TCANVAS»**

Вариант № 4

Выполнил:
Студент Васильев Н. А.
Группа И967

Преподаватель:
Ракова И. К.

Санкт-Петербург
2018

Цель работы:

Овладеть навыками анимации, освоить работу с компонентом Chart, а также изучить структуру и функциональность класса TCanvas.

Задание:

В соответствии с вариантом, выданным преподавателем, необходимо вывести зависимость $y = f(t)$ (вид зависимости см. в варианте задания). При выводе обязательно должно быть осуществлено масштабирование и “движение” графика функции. График должен быть построен в двух вариантах:

- 1) с использованием компонента Chart;
- 2) с помощью отрисовки ручным способом, используя класс TCanvas.

Менеджер памяти. Связный список – самый подходящий участок. Построить зависимость изменения объема занятой памяти во времени.

Текст программы:

Форма Canvas:

```
unit Unit3;
{$mode objfpc}{$H+}
interface
uses
  Classes, SysUtils, FileUtil,
  Forms, Controls, Graphics, Dialogs,
  ExtCtrls, Windows;
type
  { TForm3 }
  TForm3 = class(TForm)
    Img: TImage;
    procedure imgPaint(Sender:
TObject);
    procedure setStartSet;
    procedure addPoint(y:integer);
    procedure CanvasSetTextAngle(c:
TCanvas; d: single);
  private
    x0,y0,x,y: Integer;
    valueY: double;
    vertex : TStringList;
    procedure printAxis;
    procedure paintGraphic;
    procedure printGrig;
  public
  end;
```

```
var
  Form3: TForm3;
implementation
procedure TForm3.setStartSet;
begin
  vertex:=TStringList.Create;
  imgPaint(TObject(img));
  img.Canvas.MoveTo(x0,y0);
end;

procedure TForm3.imgPaint(Sender:
TObject);
begin
  img.Canvas.Brush.Color:=ClWhite;

  img.Canvas.FillRect(img.Canvas.ClipR
ect);

  printAxis;
  paintGraphic;
  printGrig;
end;

procedure
TForm3.addPoint(y:integer);
begin
  vertex.Add(inttostr(y));
  imgPaint(TObject(img));
```

```

end;

procedure
TForm3.CanvasSetTextAngle(c:
TCanvas; d: single);
var   LogRec: TLOGFONT;
begin

GetObject(c.Font.Handle,SizeOf(LogRe
c) ,Addr(LogRec) );
    LogRec.lfEscapement :=
round(d*10);
    c.Font.Handle :=
CreateFontIndirect(LogRec);
end;

procedure TForm3.printAxis;
begin
    // Рисуем оси
    x0:=65;
    y0:= img.Height - 40;
    x:=x0;
    With img.Canvas do
    begin
        Pen.Color:=clBlack;
        Pen.Width:=1;
        MoveTo(x0,5);
        LineTo(x0,y0);
        LineTo(ClientWidth-5,y0);
        Brush.Style := bsClear;
        Font.Size := 10;
        Textout(ClientWidth div
2,ClientHeight -
img.Canvas.TextHeight('Время'),
'Время');

CanvasSetTextAngle(img.Canvas,90);
        Textout(0,ClientHeight div 2,
'Память');

CanvasSetTextAngle(img.Canvas,0);
    end;

```

```

end;

procedure TForm3.paintGraphic;
var i,maxY:integer;
begin
    // Рисуем график
    img.Canvas.MoveTo(x0,y0);
    img.Canvas.Pen.Color:=clRed;
    y:= img.Height- y0;
    if vertex.Count > 0 then
maxY:=strtoint(vertex[0]);
        for i:=0 to vertex.Count-1 do
        begin
            if maxY<strtoint(vertex[i])
then maxY:=strtoint(vertex[i]);
            end;
            if vertex.Count > 0 then
valueY:=(ClientHeight-y-5)/maxY
            else valueY :=0;
            x:=x0;
            for i:=0 to vertex.Count-1 do
            begin
                x:=x+trunc((ClientWidth-x0-
15)/vertex.Count);

img.Canvas.LineTo(x,ClientHeight-y-
trunc(strtoint(vertex[i])*valueY));
            end;
        end;

procedure TForm3.printGrig;
var
i,maxY,stepY,StepX,row,col:integer;
begin
    maxY:=0; y:= img.Height- y0;
    for i:=0 to vertex.Count-1 do
    begin
        if maxY<strtoint(vertex[i])
then maxY:=strtoint(vertex[i]);
        end;
        With img.Canvas do
        begin

```

```

        Pen.Color:=clBlack;
        Font.Color:=clBlack;
        Font.Size := 8;
    end;
    row:=8;
    stepY:= round(maxY/row);
    for i:=1 to row do
    begin
        With img.Canvas do
        begin
            MoveTo(x0-
3,ClientHeight-y-
trunc(stepY*valueY));

LineTo(x0+3,ClientHeight-y-
trunc(stepY*valueY));

            Textout(x0-3-
TextWidth(inttostr(stepY)),ClientHei
ght-y-trunc(stepY*valueY)-
5,inttostr(stepY));

            Pen.Style:=psDot;

MoveTo(x0+3,ClientHeight-y-
trunc(stepY*valueY));

            LineTo(ClientWidth-
5,ClientHeight-y-
trunc(stepY*valueY));

            Pen.Style:=psSolid;
            stepY:=stepY +
round(maxY/row);
        end;
    end;
    x:=x0;
    col:=10;
    stepX:=round(vertex.Count/col);
    for i:=0 to col do
    begin
        With img.Canvas do
        begin
            x:=x+trunc((ClientWidth-
x0-15)/col);
            MoveTo(x,y0+3);

```

```

            LineTo(x,y0-3);

Textout(x,y0+5,inttostr(stepX));

            Pen.Style:=psDot;
            MoveTo(x,y0+3);
            LineTo(x,5);
            Pen.Style:=psSolid;

stepX:=stepX+round(vertex.Count/col)
;

        end;
    end;
end;

{$R *.lfm}
end.

Φopma Chart:
unit Unit2;
{$mode objfpc}{$H+}
interface
uses
    Classes, SysUtils, FileUtil,
    TAGraph, TASeries, Forms, Controls,
    Graphics,
    Dialogs;

type
    { TForm2 }
    TForm2 = class(TForm)
        Chart1: TChart;
        Chart1LineSeries1: TLineSeries;
        procedure addPoint(y:integer);
        procedure SetStartSet;
    private
        x:integer;
    public
        end;
    var
        Form2: TForm2;
    implementation
        procedure TForm2.SetStartSet;

```

```

begin
x:=0;
Chart1LineSeries1.SeriesColor:=clRed
;
end;

procedure
TForm2.addPoint(y:integer);
begin
inc(x);
Chart1LineSeries1.AddXY(x,y, '');
end;

{$R *.lfm}
end.

```

Главная форма:

```

TForm1 = class(TForm)
..
btnChart: TButton;
btnCanvas: TButton;
..
tChartGr: TForm2;
tCanvasGr: TForm3;
..
procedure
TForm1.btnCanvasClick(Sender:
TObject);
begin

```

```

tCanvasGr.Show;
end;
procedure
TForm1.btnChartClick(Sender:
TObject);
begin
tChartGr.Show;
end;
..
procedure
TForm1.StartButtonClick(Sender:
TObject);
..
tChartGr:=TForm2.Create(Self);
tCanvasGr:= TForm3.Create(Self);
tChartGr.SetStartSet;
tCanvasGr.setStartSet;
btnChart.Enabled := True;
btnCanvas.Enabled := True;
..
procedure TForm1.Timer1Timer(Sender:
TObject);
..
tChartGr.addPoint(usedMemory);
tCanvasGr.addPoint(usedMemory);
..

```

Остальной код формы и логики программы совпадает с кодом из Л.Р. №3.

Результаты работы программы:

