

Практична робота №19

Тема: Програмування подій, пов'язаних із обробкою таймера.

Мета: Навчитися використовувати графічні процедури та функції, процедури обробки таймера.

Хід роботи

1. Створити у своїй робочій папці каталог PRACT19.
2. Запустити середовище Delphi та створити новий додаток.
3. Виконати File – Save All, вказавши пунктом призначення створений Вами каталог PRACT19, під іменами, які пропонує середовище Delphi.

Умова задачі. Написати програму, яка в діалоговому вікні виводить зображення працюючого годинника із годинною, хвилинною та секундною стрілками.

Деякі зауваження щодо проектування форми.

На формі розміщено лише компонент таймер (Timer1).

Деякі зауваження щодо створення програмного коду.

У заголовку форми доцільно відобразити, окрім назви практичної роботи, також прізвища та імена виконавців.

До переліку стандартних потрібно додати бібліотеку ExtCtrl

uses

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, ExtCtrls;
```

У програмі використовуються три процедури обробки подій, а також дві процедури користувача:

type

```
TForm1 = class(TForm)  
    Timer1: TTimer;  
    procedure Timer1Timer(Sender: TObject);  
    procedure FormPaint(Sender: TObject);  
    procedure FormCreate(Sender: TObject);  
    //ці оголошення вставлені користувачем  
    procedure Vector (x0, y0, a, l: integer);  
    procedure DrawClock;  
private  
    { Private declarations }  
public  
    { Public declarations }  
end;
```

Перед початком роботи програми доцільно обумовити деякі початкові значення:

var

```
Form1: TForm1;
```

implementation

```
{ $R *.dfm }
```

uses

```
DateUtils; //для доступу до SecondOf, MinuteOf, HourOf
```

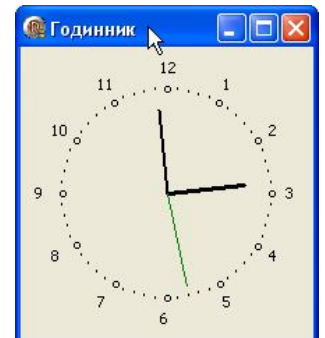
const

```
r=75; //радіус циферблату годинника
```

var

```
x0, y0: integer; //центр циферблату
```

```
ahr, amin, asec: integer; //положення стрілок (кут)
```



Процедура для ініціалізації форми має такий програмний код:

```
//ініціалізація форми  
procedure TForm1.FormCreate(Sender: TObject);  
var  
    t: TDateTime;  
begin  
    //задамо розмір форми у відповідності до розміру циферблата  
    clientheight:=(r+30)*2;  
    clientwidth:=(r+30)*2;  
    x0:=r+30;  
    y0:=r+30;  
    t:=now();  
    //положення стрілок  
    ahr:=90-HourOf(t)*30-(MinuteOf(Today) div 12) *6;  
    amin:=90-MinuteOf(t)*6;  
    asec:=90-SecondOf(Today)*6;  
    timer1.Interval:=1000; //період сигналів від таймера 1 с  
    timer1.enabled:=true; //пуск таймера  
end;
```

Процедура, за допомогою якої зображується вектор певної довжини, має наступний програмний код:

```
//намалювати вектор заданої довжини із точки (x0, y0)  
procedure TForm1.Vector(x0,y0: Integer; a,l: Integer);  
//x0, y0 - початок вектора  
//a - кут між віссю x та вектором  
//l - довжина вектора  
const  
    grad=0.0174532; //коефіцієнт перерахунку кута із градусів у радіани  
var  
    x,y: integer; //координати кінця вектора  
begin  
    canvas.MoveTo(x0,y0);  
    x:=round(x0+l*cos(a*grad));  
    y:=round(y0-l*sin(a*grad));  
    canvas.LineTo(x,y);  
end;
```

Процедура, за допомогою якої будуються поточні положення стрілок годинника, має такий програмний код:

```
//намалювати поточні положення стрілок годинника  
procedure TForm1.Timer1Timer(Sender: TObject);  
begin  
    DrawClock;  
end;
```

Процедура користувача, яка має призначення – малювати стрілки годинника, а також процедура, яка малює циферблат та початкові стрілки, мають такий програмний код:

Намалювати стрілки годинника	Намалювати циферблат та початкові стрілки
<pre> //намалювати стрілки procedure TForm1.DrawClock; var t: TDateTime; begin //крок секундної та хвилинної стрілок - 6 градусів //крок годинникової стрілки - 30 градусів //вигляд зображення стрілок canvas.Pen.color:=clBtnFace; canvas.Pen.Width:=3; //годинникову vector (x0,y0,ahr,r-20); //хвилинну vector (x0,y0,amin,r-15); //секундну vector (x0,y0,asec,r-7); t:=now(); //нове положення стрілок ahr:=90-HourOf(t)*30-(MinuteOf(t) div 12) *6; amin:=90-MinuteOf(t)*6; asec:=90-SecondOf(t)*6; //намалювати стрілки //годинникова стрілка canvas.Pen.Width:=3; canvas.Pen.color:=clBlack; vector (x0,y0,ahr,r-20); //хвилинна стрілка canvas.Pen.Width:=2; canvas.Pen.color:=clBlack; vector (x0,y0,amin,r-15); //секундна стрілка canvas.Pen.Width:=1; canvas.Pen.color:=clGreen; vector (x0,y0,asec,r-7); end; </pre>	<pre> //намалювати циферблат і початкові стрілки procedure TForm1.FormPaint(Sender: TObject); var x,y: integer; //координати маркера на циферблаті a: integer; //кут між Ох та прямою (x0,y0)-(x,y) h: integer; //мітка годинникової риски bs: TBrushStyle; //стиль кисті pc: TColor; //колір олівця pw: integer; //ширина олівця begin bs:=canvas.Brush.Style; pc:=canvas.Pen.Color; pw:=canvas.Pen.Width; canvas.Brush.Style:=bsClear; canvas.Pen.Width:=1; canvas.Pen.color:=clBlack; a:=0; //мітки ставимо від 3 годин, напроти годинникової стрілки h:=3; //кут 0 градусів - це 3 години //циферблат while a<360 do begin x:=x0+round(r*cos(a*2*pi/360)); y:=y0-round(r*sin(a*2*pi/360)); form1.canvas.MoveTo (x,y); if (a mod 30)=0 then begin canvas.Ellipse (x-2,y-2,x+3,y+3); //цифри по великому радіусу x:=x0+round((r+15)*cos(a*2*pi/360)); y:=y0-round((r+15)*sin(a*2*pi/360)); canvas.TextOut (x-5,y-7,IntToStr(h)); dec(h); if h=0 then h:=12; end else canvas.Ellipse(x-1,y-1,x+1,y+1); a:=a+6; //1 хвилина - 6 градусів end; //відновити олівець - кисть canvas.Brush.Style:=bs; canvas.pen.Width:=pw; canvas.Pen.color:=pc; DrawClock; end; </pre>

4. Зберегти отриманий додаток на диску та запустимо його на виконання.
5. Завершити роботу із середовищем Delphi.
6. Із операційної оболонки запустити файл із розширенням exe, який утворився у створеній Вами папці PRACT19.
7. Створити звіт із результатами виконання практичної роботи.
8. Навести порядок на робочому місці.