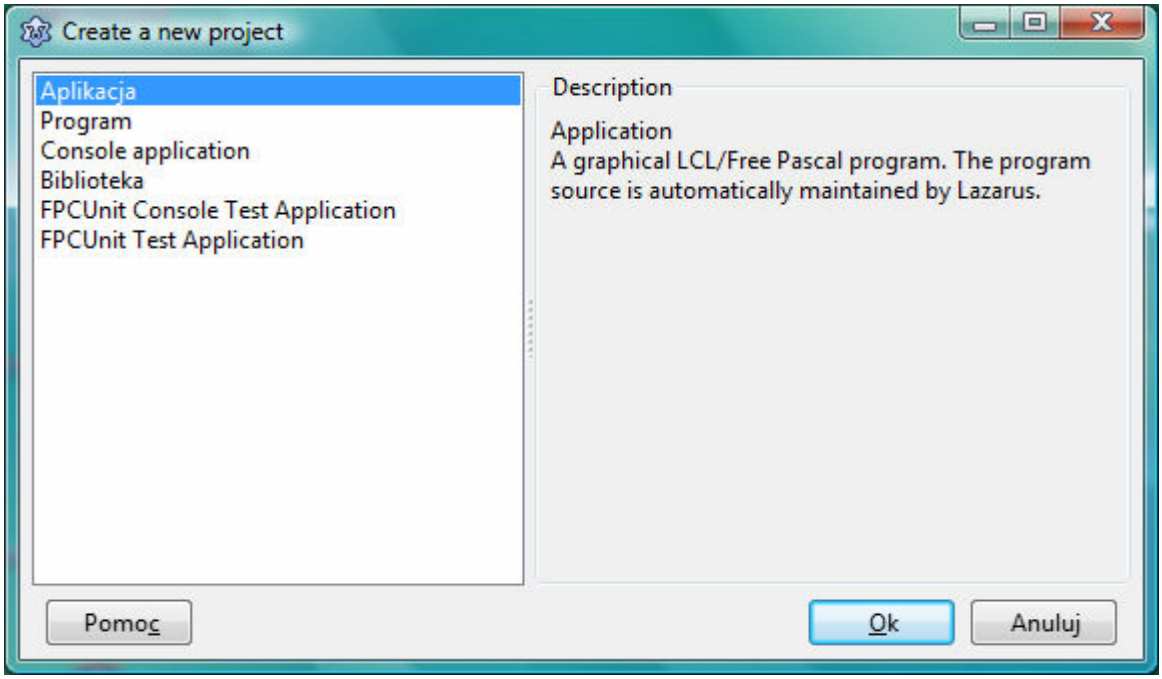
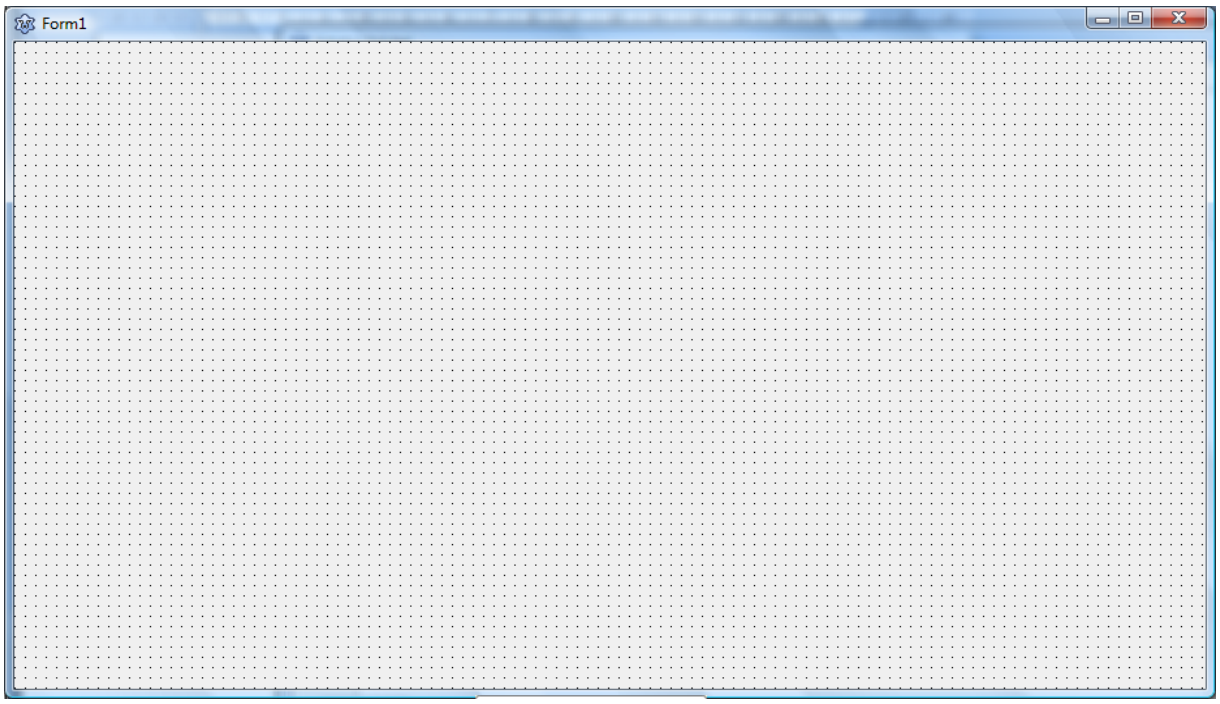
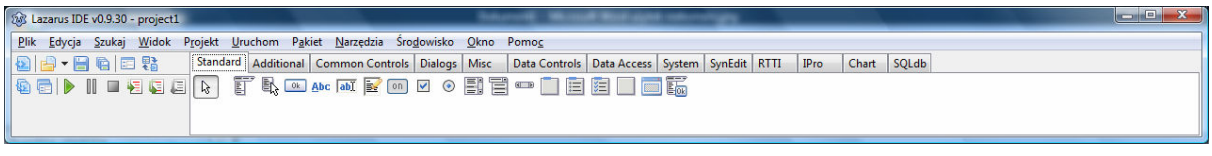
Po uruchomieniu Lazarusa należy wybrać z paska górnego opcję „Projekt” i następnie ”Nowy Projekt”. Pokaże się okno:



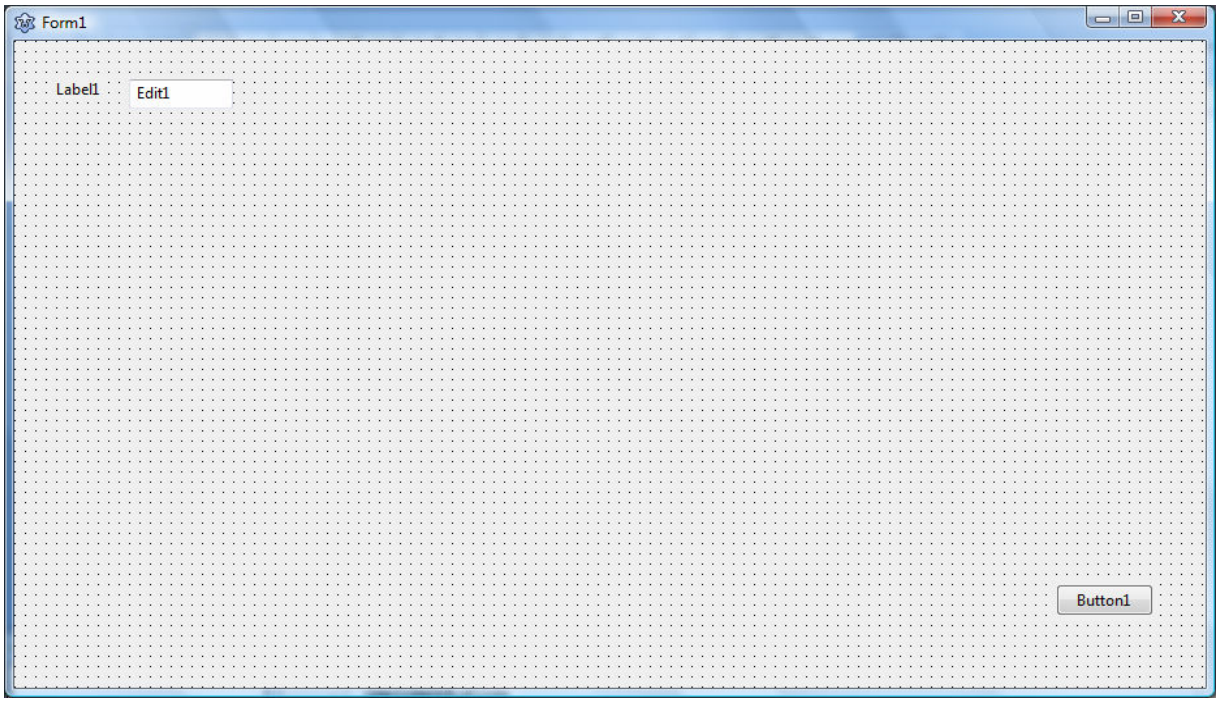
Należy wybrać „Aplikacja”, pojawi się puste okno Form1:



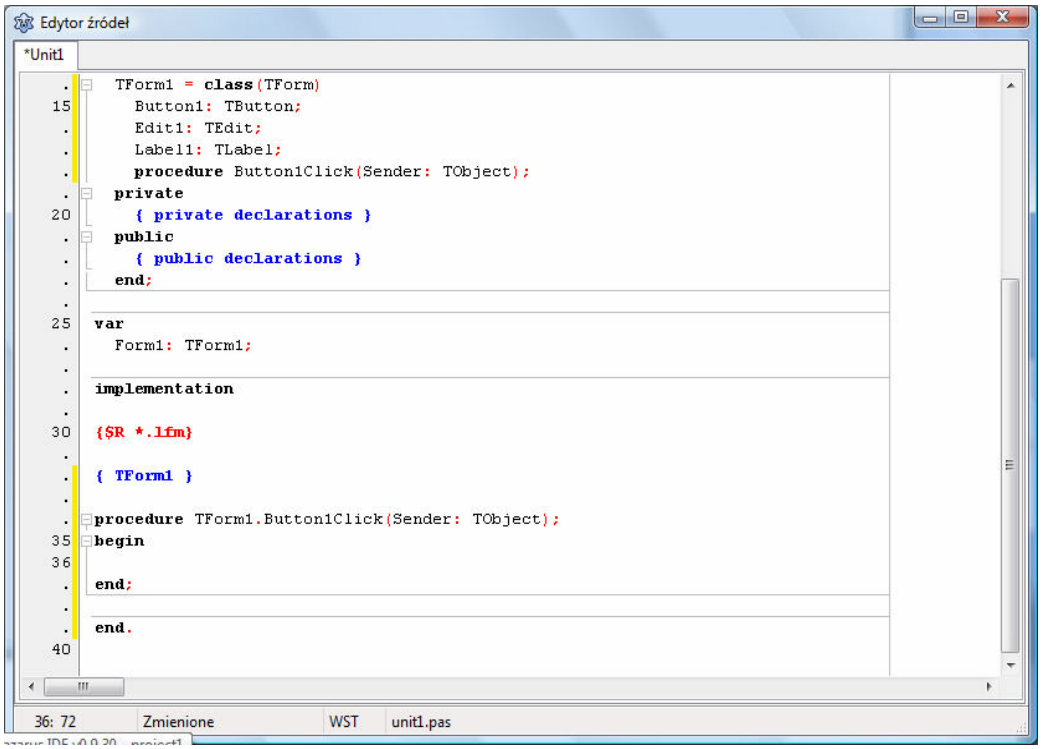
Z zakładki standard należy wprowadzić obiekty: label i Edit poprzez kliknięcie na ikonę a następnie kliknięcie na formularzu „Form1” w miejsce, gdzie chcemy wprowadzić element. Podobnie robimy wprowadzając obiekt „Button”



Operując myszką należy ustawić względem siebie oba obiekty jak na rysunku poniżej:



Podwójne kliknięcie w klawisz Button1 powoduje przejście do okienka edytora źródeł w miejsce procedury



procedure TForm1.Button1Click(Sender: TObject);
begin

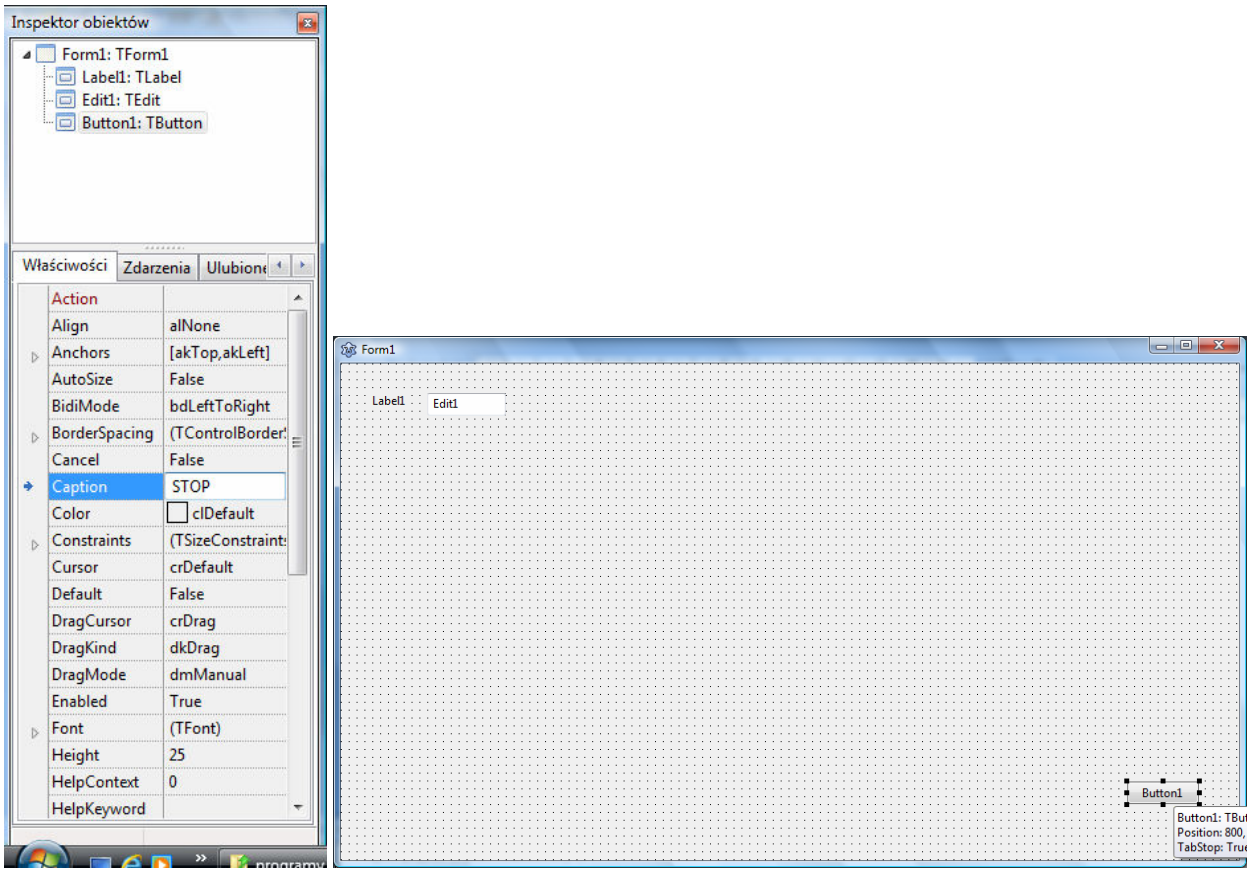
end;

Wpisując pomiędzy **begin** i **end**; polecenie

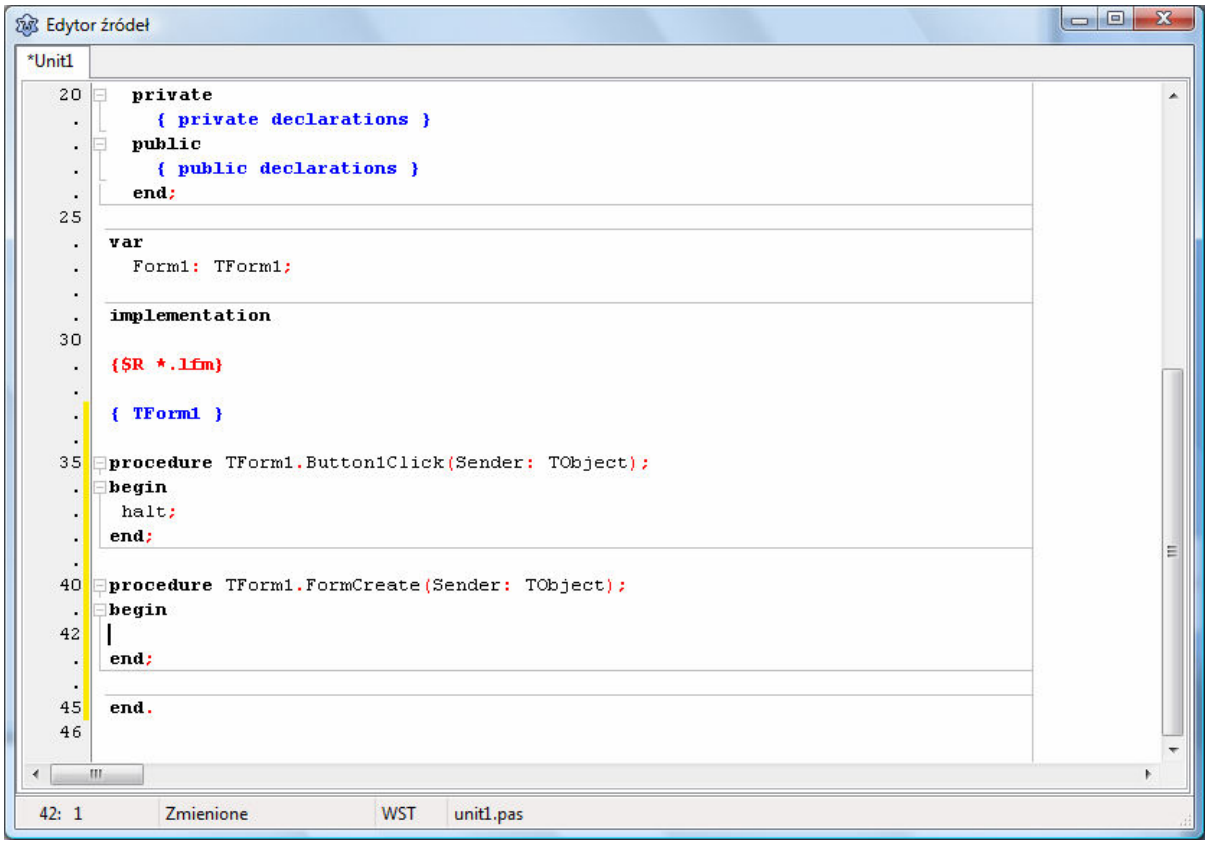
Halt;

Uzyskujemy procedurę, która po wciśnięciu klawisza Button1 zatrzyma wykonywanie programu. Możemy sprawdzić działanie klawisza wciskając klawisz F9 lub przycisk Play na panelu górnym.

W trybie edycji formularza, po zaznaczeniu przycisku Button1 możemy zmienić napis wyświetlany na przycisku korzystając z panelu Inspektora obiektów, znajdującego się po lewej stronie ekranu, zmieniając w polu caption napis z Button1 na np. STOP:



Podwójne kliknięcie w obszar formularza Form1 powoduje przejście do okienka edytora źródeł w miejsce procedury:



procedure TForm1.FormCreate(Sender: TObject);
begin

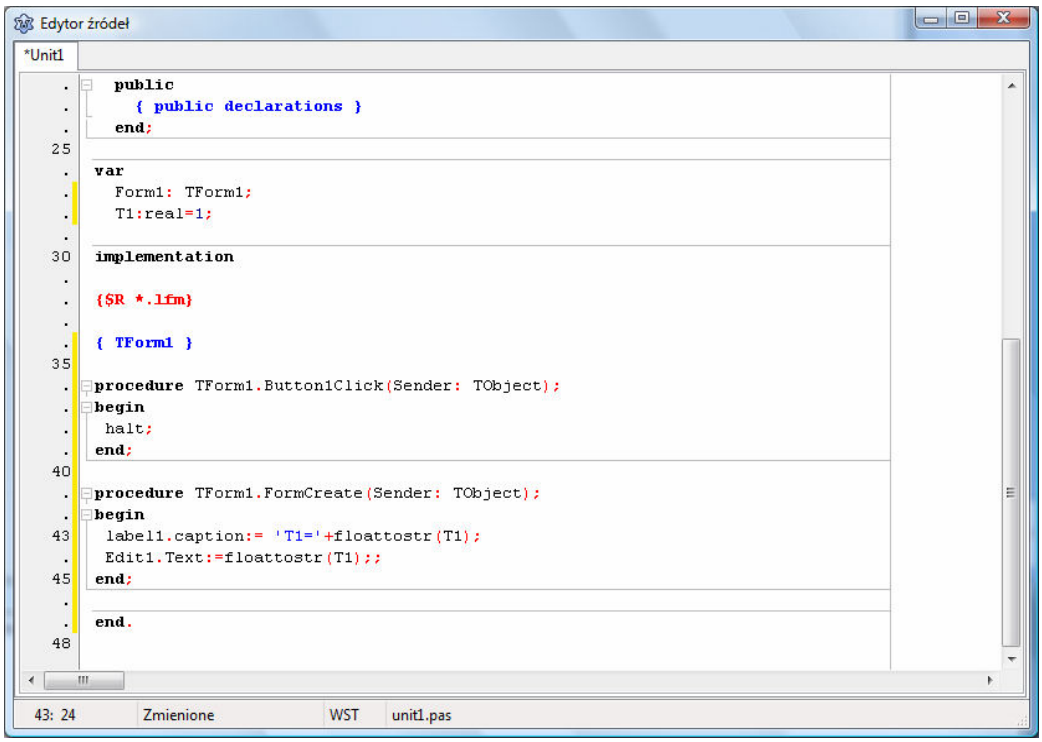
end;

która to procedura odpowiada za to, co pojawia się na formularzu po uruchomieniu programu.

Jeżeli chcemy zdefiniować zmienną T1 o wartości początkowej np. 1, która po uruchomieniu programu będzie się wyświetlać na formularzu (label1) i w okienku edit1. Należy na początek wprowadzić definicję zmiennej z wartością początkową w obszarze var programu, znajdującym się w górnej części programu w edytorze źródeł.

Zapis : **T1:real=1;**

Oznacza, że zdefiniowaliśmy zmienną typu rzeczywistego T1 o wartości początkowej 1.



Po powrocie do procedury

TForm1.FormCreate (Sender: TObject);

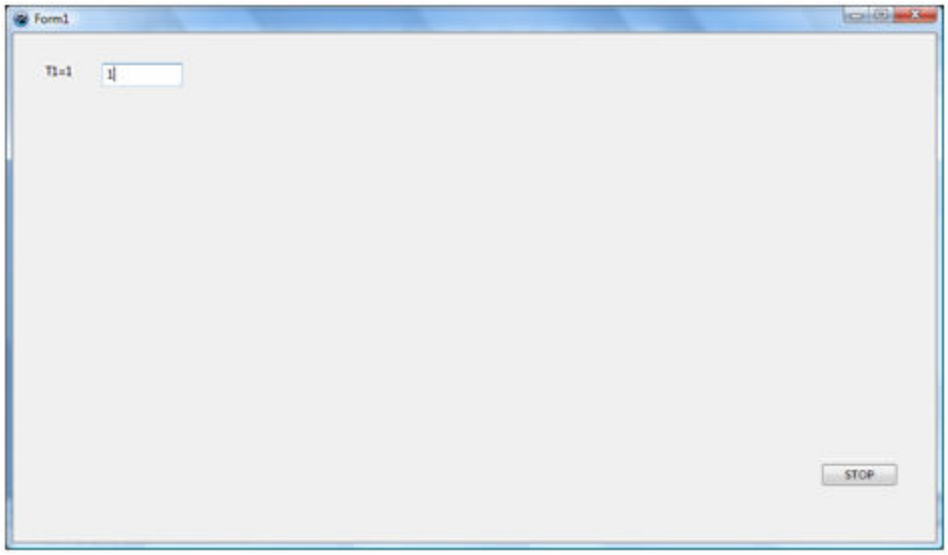
Pomiędzy Begin i end; wpisujemy:

label1.caption:= 'T1='+floattostr(T1);

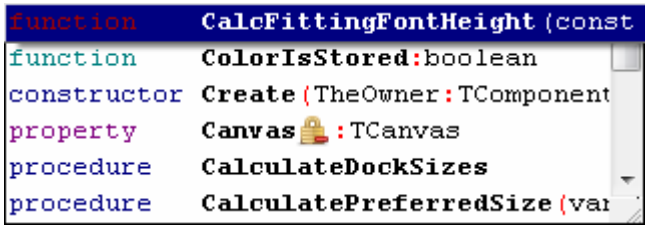
Edit1.Text:=floattostr(T1);

label1.caption:= oznacza, że napis na etykiecie będzie taki jak po znaku równa się, przy czym 'T1' oznacza wprowadzenie tekstu takiego jak między ' ', a **floattostr(T1)** oznacza zamianę zmiennej rzeczywistej na tekst.

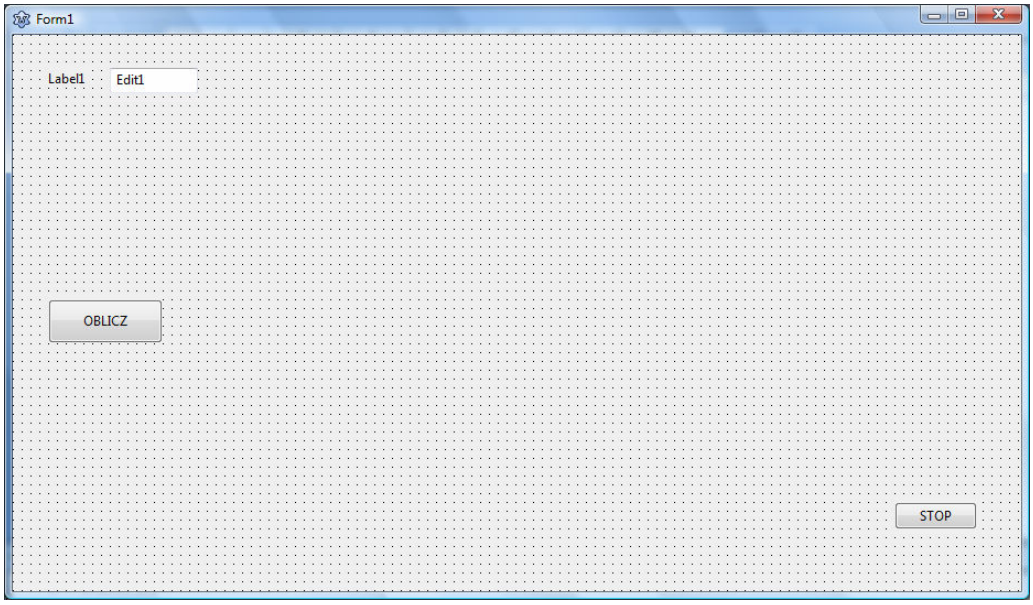
Edit1.Text:= oznacza, że w okienku Edit1 pojawi się wartość zmiennej T1.



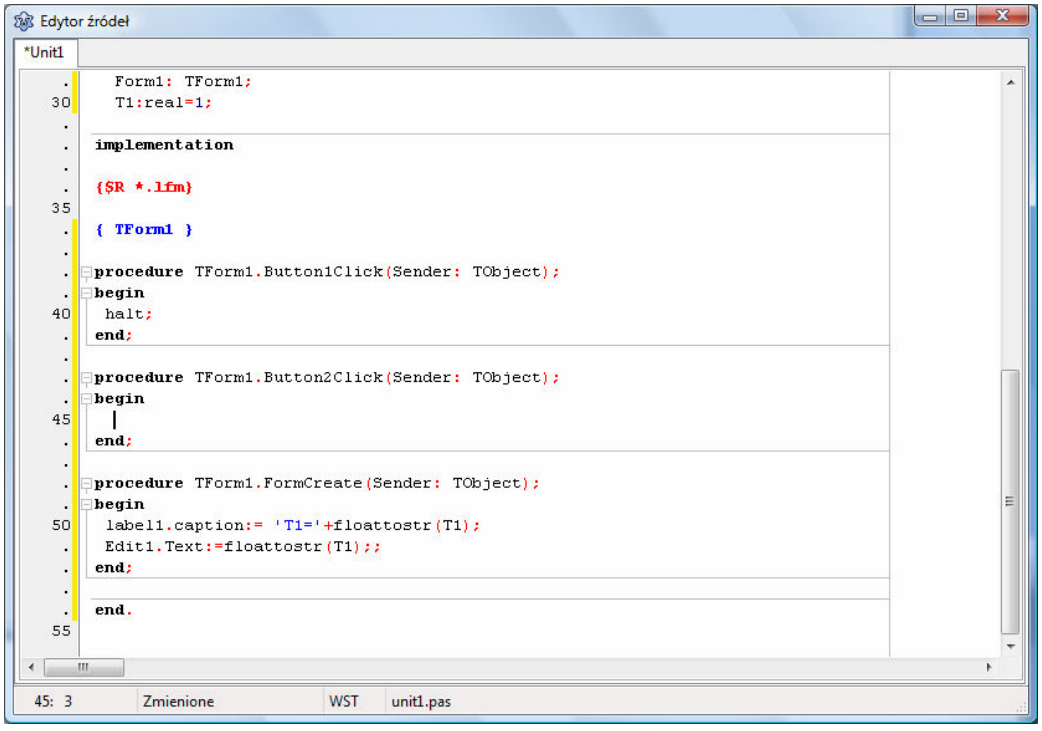
Podczas wpisywania można korzystać z okienka asystenta, pojawiającego się automatycznie w czasie pisania:



Wprowadzimy teraz kolejny przycisk, który tym razem będzie uruchamiał procedurę obliczeniową. Postępujemy tak samo jak w przypadku pierwszego przycisku. Okno Form1 będzie wyglądało jak poniżej:



A w edytorze powstanie procedura



procedure TForm1.Button2Click(Sender: TObject);
begin

end;

Wpisanie w tej procedurze :

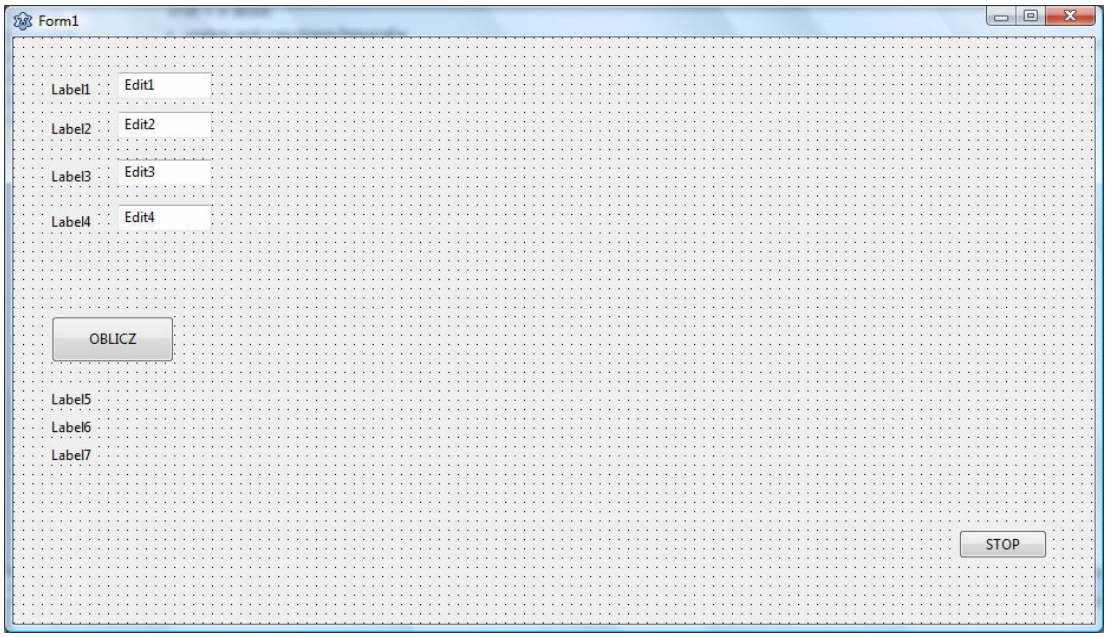
T1:=strtofloat(edit1.text);

label1.caption:= 'T1='+floattostr(T1);

daje możliwość zmiany parametru T1 i zatwierdzenia tej zmiany przy wciśnięciu przycisku „OBLICZ”.

Ponieważ stworzono kilka obiektów w programie, dobrze byłoby to zapisać. Dobrze jest pamiętać, aby zapisywać każdy projekt w oddzielnym katalogu, wykorzystując polecenie „zapisz wszystko”.

Następnie wprowadzamy kolejne obiekty label i edit jak na rysunku poniżej



Należy zdefiniować trzy zmienne z wartością początkową: dt=0,001, Umax=1, i tp=1. I połączyć je z obiektami Label i Edit od nr 2 do 4. Obszar var będzie wyglądał następująco:

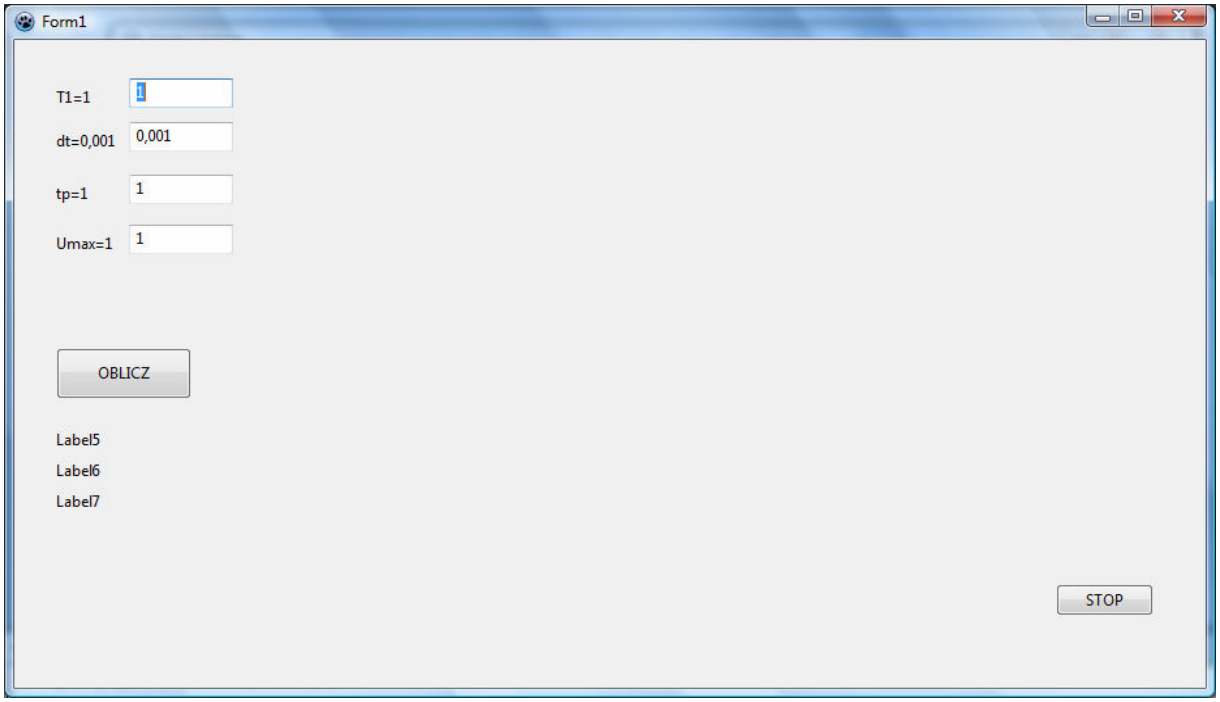
```
var
  Form1: TForm1;
  T1:real=1; dt:real=0.001;  tp:real=1; Umax:real=1;
```

Procedury **TForm1.Button2Click** oraz **TForm1.FormCreate** przyjmą następujące postacie:

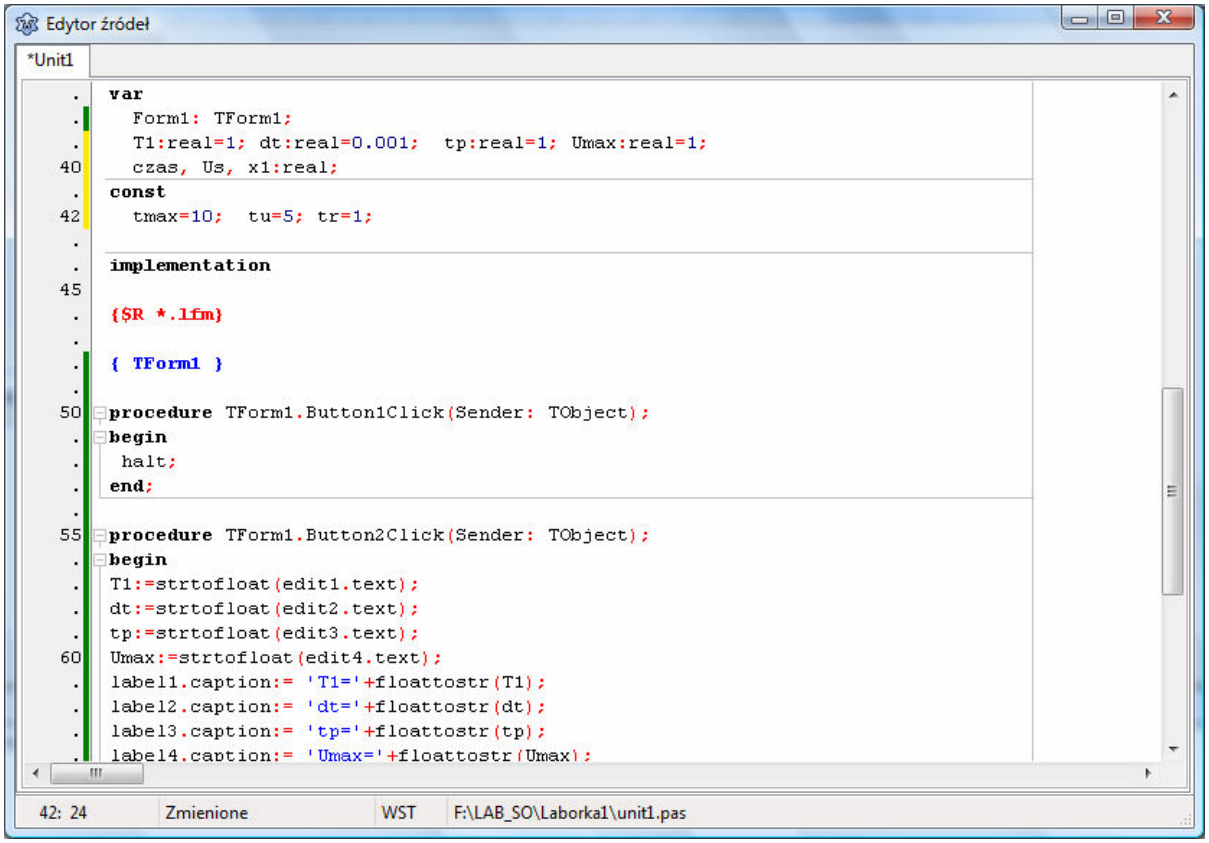
```
procedure TForm1.Button2Click(Sender: TObject);
begin
T1:=strtofloat(edit1.text);
dt:=strtofloat(edit2.text);
tp:=strtofloat(edit3.text);
Umax:=strtofloat(edit4.text);
label1.caption:= 'T1='+floattostr(T1);
label2.caption:= 'dt='+floattostr(dt);
label3.caption:= 'tp='+floattostr(tp);
label4.caption:= 'Umax='+floattostr(Umax);
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
 label1.caption:= 'T1='+floattostr(T1);
 Edit1.Text:=floattostr(T1);
 label2.caption:= 'dt='+floattostr(dt);
 Edit2.Text:=floattostr(dt);
 label3.caption:= 'tp='+floattostr(tp);
 Edit3.Text:=floattostr(tp);
 label4.caption:= 'Umax='+floattostr(Umax);
 Edit4.Text:=floattostr(Umax);
end;
```

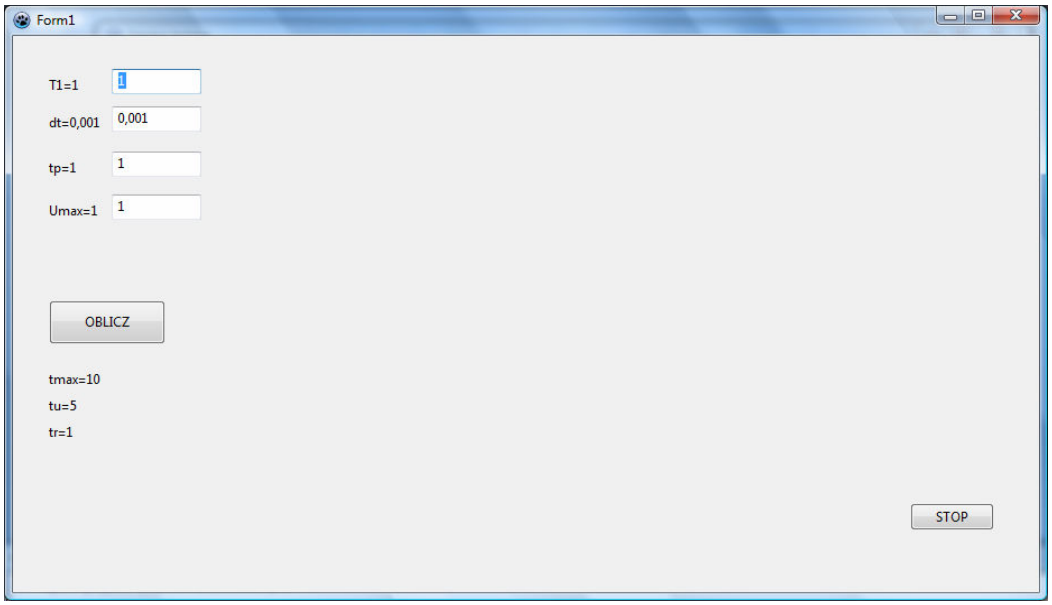
Po tych zabiegach uruchomienie projektu spowoduje pojawienie się następującego okienka:



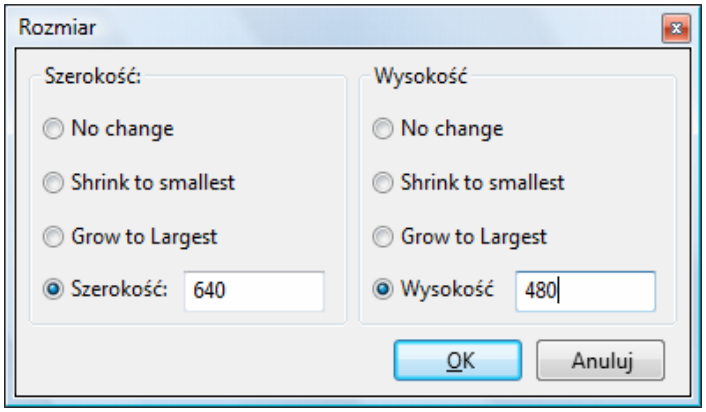
Następnie należy zdefiniować zmienne typu rzeczywistego: czas, Us, x1 oraz stałe (poprzedzone poleceniem **const**) tmax=10; tu=5; tr=1;



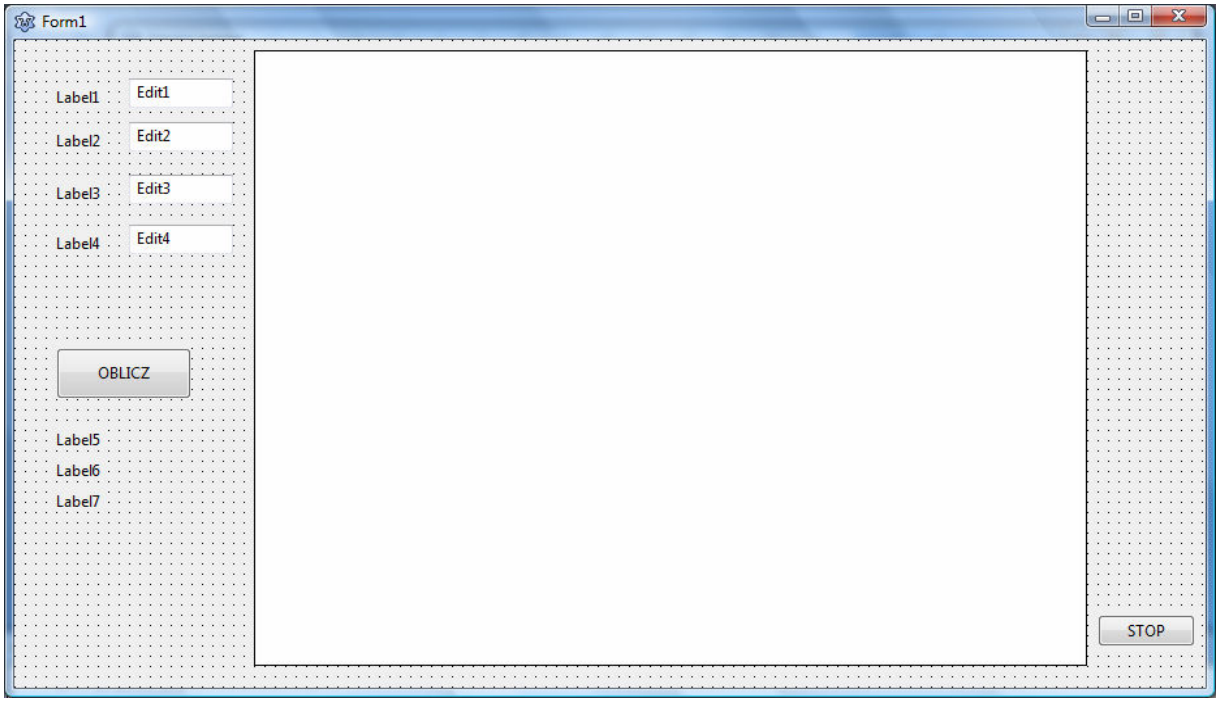
Powiązanie etykiet label od 5 do 7 ze stałymi spowoduje wyświetlenie ich wartości w odpowiednich miejscach jak na rys. poniżej. Tych wartości w czasie pracy programu nie będziemy zmieniać.



Kolejnym etapem będzie wprowadzenie obiektu Tshape, w którym będziemy rysować przebiegi funkcji. Obiekt wprowadzamy z zakładki Additional i po wprowadzeniu, klikamy na nim prawym klawiszem myszy i wybieramy polecenie rozmiar. Następnie zaznaczamy i wpisujemy jak na rysunku poniżej i potwierdzamy:



Przy użyciu myszy dopasowujemy położenie prostokąta i przycisku **STOP**, ew. zmieniamy wielkość formularza.



Wprowadzenie na początku procedury **TForm1.Button2Click** poleceń:

Shape1.canvas.moveto(0,240);
Shape1.canvas.lineto(640,240);
Spowoduje narysowanie poziomej linii na środku prostokąta shape1, ale dopiero po wciśnięciu przycisku **OBLICZ**..
W tej samej procedurze wprowadzimy polecenie pętli repeat:
Procedura przyjmie wówczas następującą postać:

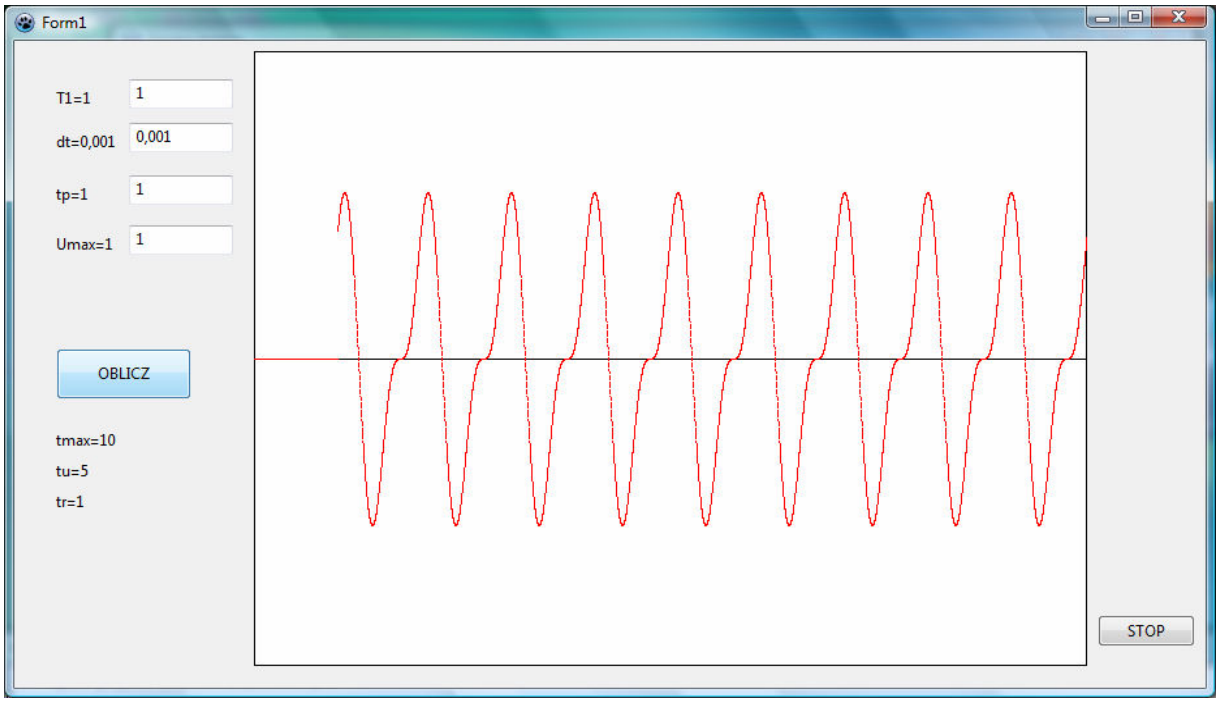
```
procedure TForm1.Button2Click(Sender: TObject);
begin
shape1.canvas.moveto(0,240);
Shape1.canvas.lineto(640,240);
T1:=strtofloat(edit1.text);
dt:=strtofloat(edit2.text);
tp:=strtofloat(edit3.text);
Umax:=strtofloat(edit4.text);
label1.caption:= 'T1='+floattostr(T1);
label2.caption:= 'dt='+floattostr(dt);
label3.caption:= 'tp='+floattostr(tp);
label4.caption:= 'Umax='+floattostr(Umax);

czas:=0;
repeat
czas:=czas+dt;
until czas>tmax;
end;
```

gdzie czas:=czas+dt; oznacza przyrost argumentu (w każdym kroku o dt), a czas>tmax to warunek wyjścia z pętli. Obliczenie nowej wartości czasu musi być w ostatniej linijce pętli.
Wprowadzenie wzoru na x1 (obliczanie dowolnej funkcji) i polecenia Shape1.Canvas.Pixels spowoduje obliczanie w pętli wartości funkcji i rysowanie tych wartości na ekranie w określonej skali.

```
czas:=0; x1:=0;
repeat
  if czas>=tp then x1:=cos(2*pi/T1*(czas-tp))+sin(2*pi/T1*(czas-tp))*cos(2*pi/T1*(czas-tp));
  Shape1.Canvas.Pixels[round(640/tmax*czas),240-round(x1*100)]:=clred;
czas:=czas+dt;
until czas>tmax;
```

Spróbuj rozszyfrować składnię polecenia shape1.canvas.pixels.
Do czego służy polecenie round ?
Ostateczny efekt po uruchomieniu programu i wciśnięciu przycisku OBLICZ będzie jak poniżej:



Pełny listing programu poniżej:

```
unit Unit1;
{$mode objfpc}{$H+}
interface
uses
  Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs, StdCtrls,
  ExtCtrls;
type
  { TForm1 }
  TForm1 = class(TForm)
    Button1: TButton;
    Button2: TButton;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Edit4: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Shape1: TShape;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
```

```
  private
    { private declarations }
  public
    { public declarations }
  end;

var
  Form1: TForm1;
  T1:real=1; dt:real=0.001;  tp:real=1; Umax:real=1;
  czas, Us, x1:real;
const
  tmax=10;  tu=5; tr=1;

implementation

{$R *.lfm}

{ TForm1 }

procedure TForm1.Button1Click(Sender: TObject);
begin
 halt;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
shape1.canvas.moveto(0,240);
Shape1.canvas.lineto(640,240);
T1:=strtofloat(edit1.text);
dt:=strtofloat(edit2.text);
tp:=strtofloat(edit3.text);
Umax:=strtofloat(edit4.text);
label1.caption:= 'T1='+floattostr(T1);
label2.caption:= 'dt='+floattostr(dt);
label3.caption:= 'tp='+floattostr(tp);
label4.caption:= 'Umax='+floattostr(Umax);

czas:=0; x1:=0;
repeat
  if czas>=tp then x1:=cos(2*pi/T1*(czas-tp))+sin(2*pi/T1*(czas-tp))*cos(2*pi/T1*(czas-tp));
  Shape1.Canvas.Pixels[round(640/tmax*czas),240-round(x1*100)]:=clred;
czas:=czas+dt;
until czas>tmax;
end;
```

```
procedure TForm1.FormCreate(Sender: TObject);
begin
 label1.caption:= 'T1='+floattostr(T1);
 Edit1.Text:=floattostr(T1);
 label2.caption:= 'dt='+floattostr(dt);
 Edit2.Text:=floattostr(dt);
 label3.caption:= 'tp='+floattostr(tp);
 Edit3.Text:=floattostr(tp);
 label4.caption:= 'Umax='+floattostr(Umax);
 Edit4.Text:=floattostr(Umax);
 label5.caption:= 'tmax='+floattostr(tmax);
 label6.caption:= 'tu='+floattostr(tu);
 label7.caption:= 'tr='+floattostr(tr);
end;

end.
```

Zmiana parametru np. T1 i ponowne przeliczenie powoduje, że w oknie Shape1 pojawia się nowy wykres, ale nie kasuje się stary. Można temu zaradzić pisząc np. polecenie rysowania prostokąta w tym kształcie na początku procedury

TForm1.Button2Click

Polecenie posiada następującą składnię:

shape1.canvas.rectangle (0,0,640,480);

Kolejnym etapem będzie określenie funkcji wymuszającej trapezowej. Na początek określimy narastanie i wartość ustaloną.

Aby tego dokonać należy w pętli repeat wprowadzić następujący zapis:

if czas>0 then Us:=0;

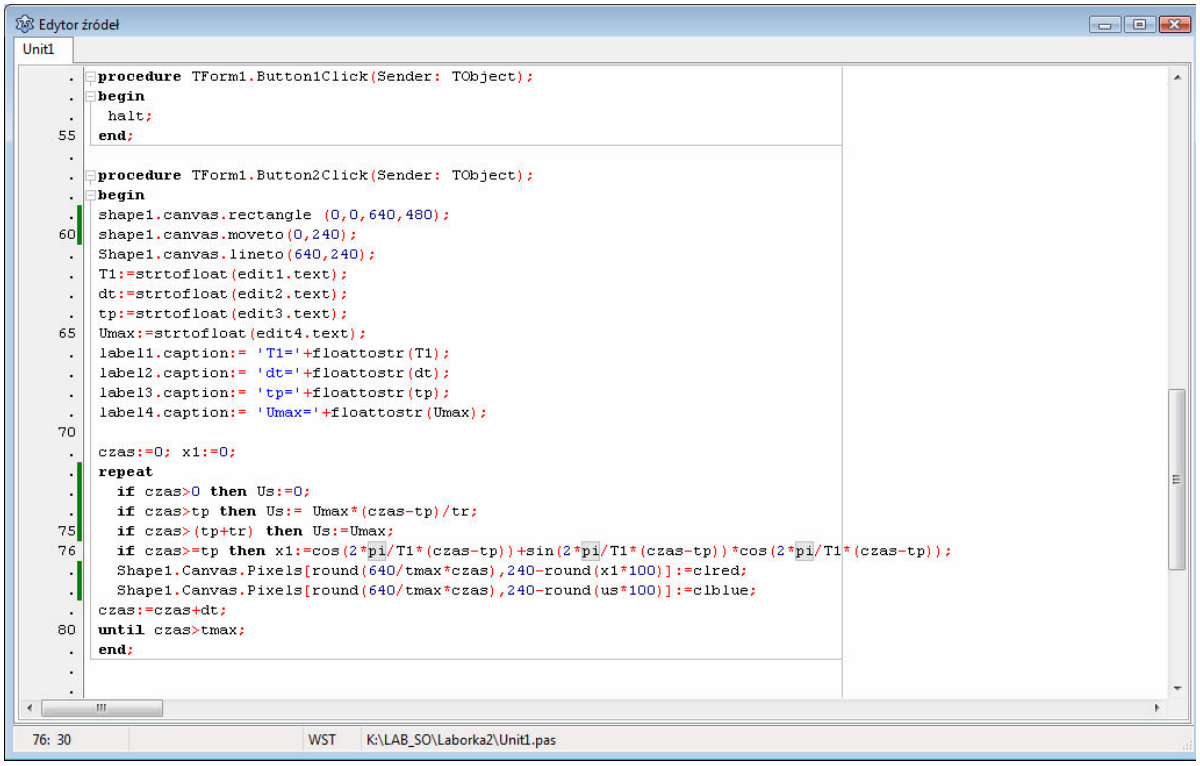
if czas>tp then Us:= Umax*(czas-tp)/tr;

if czas>(tp+tr) then Us:=Umax;

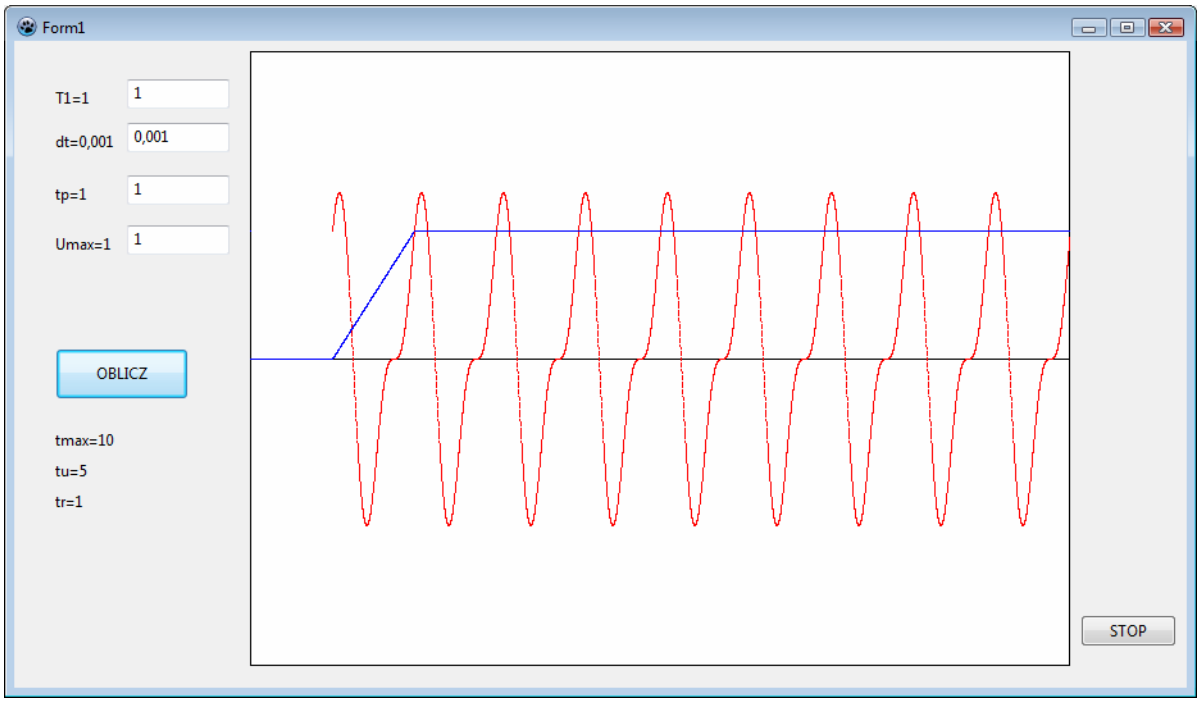
określa on wartości funkcji Us w każdym z rozpatrywanych rozdziałów, tzn 0-tp, tp-tr, tr-tmax. W celu narysowania tego przebiegu należy dodatkowo wprowadzić polecenie:

Shape1.Canvas.Pixels[round(640/tmax*czas),240-round(us*100)]:=clblue;

Zapis w edytorze źródeł wygląda następująco:



a program po przeliczeniu:

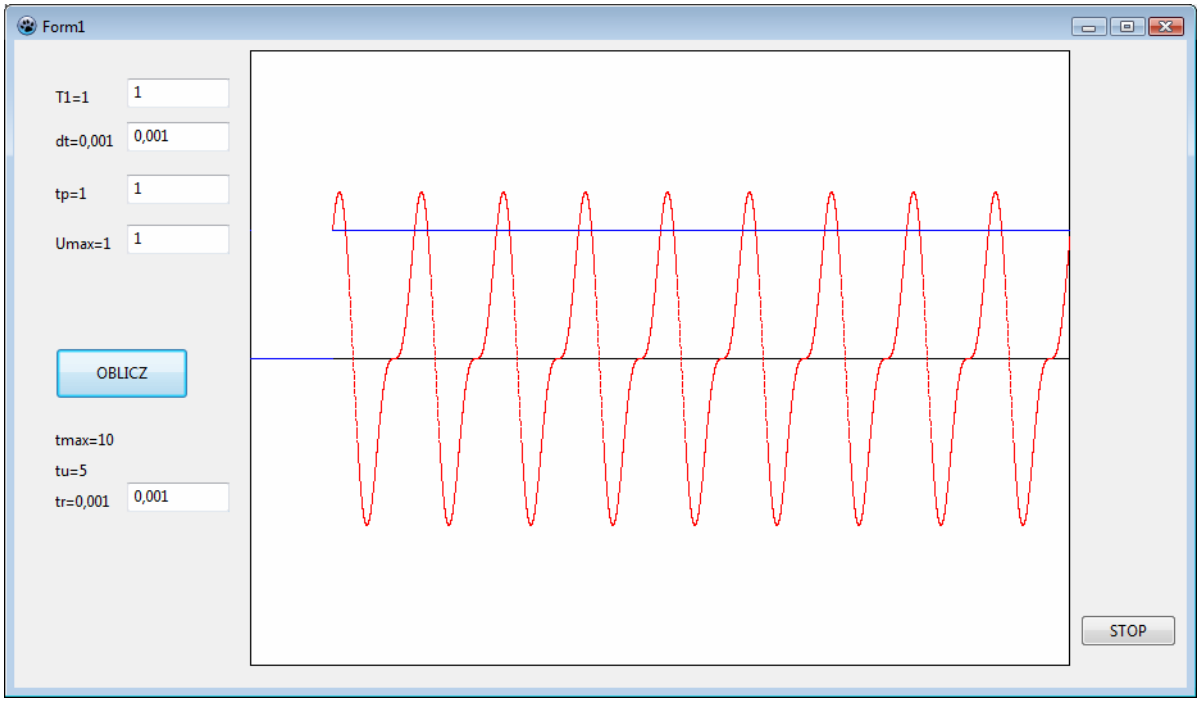


Jeżeli chcemy mieć wpływ na czas narastania funkcji Us, należy zamienić stałą tr na zmienną z wartością początkową. Dokonujemy tego usuwając definicję tr z obszaru const i przenosimy do obszaru var, uzupełniając ją do następującej formy.

tr:real=0.001;

następnie do formularza dodajemy obiekt Edit5 i umożliwiamy zmiany parametru tr podobnie jak w przypadku obiektów Edit od 1 do 4.

Ze względu na zależności matematyczne tr nie może przyjmować wartości zero, zatem minimalna wartość dla tr może wynosić tyle co dt i taką wartość przyjmiemy za początkową. Program pokaże wówczas następujące wykresy:



Układy inercyjne II rzędu i oscylacyjne

Zapis tego typu układów może być dwojaki:

$$T_{i1} \cdot \frac{d^2y}{dt^2} + T_{i2} \cdot \frac{dy}{dt} + y = k_i \cdot U_s$$

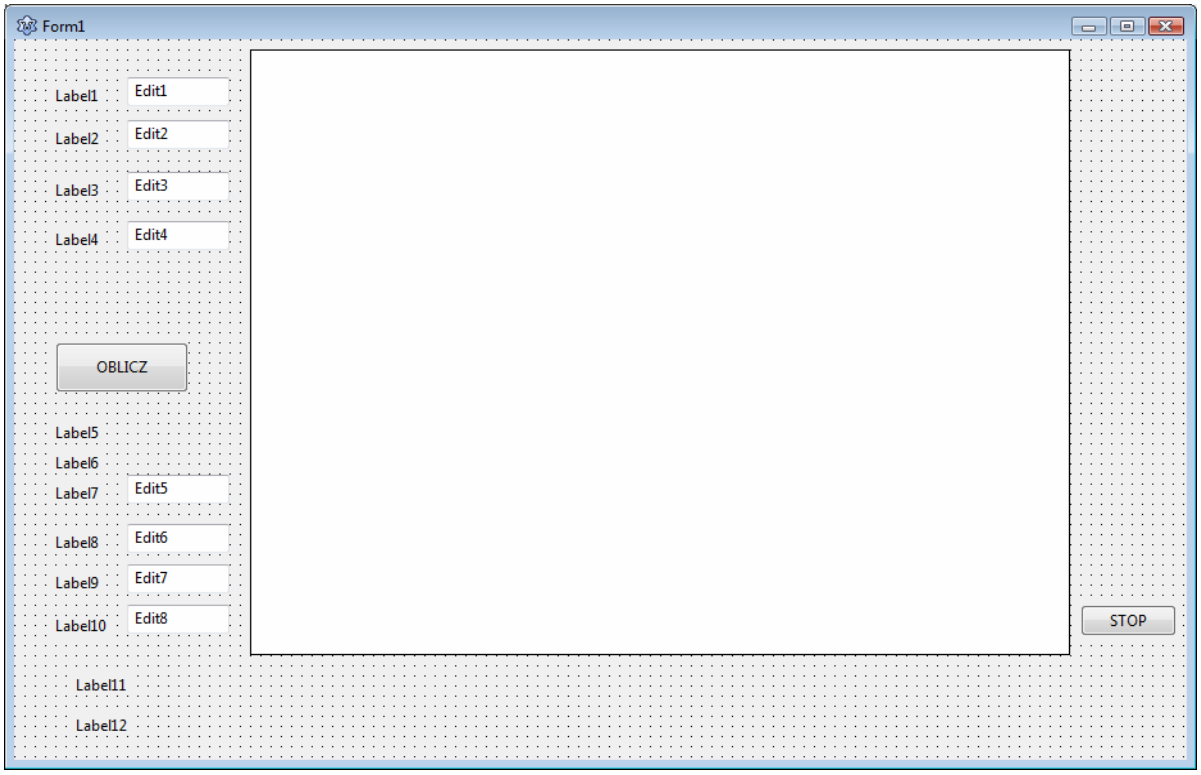
$$\frac{d^2y}{dt^2} + 2 \cdot \omega \cdot \xi \cdot \frac{dy}{dt} + \omega^2 \cdot y = k_i \cdot \omega^2 \cdot U_s$$

Użyte oznaczenia to odpowiednio:

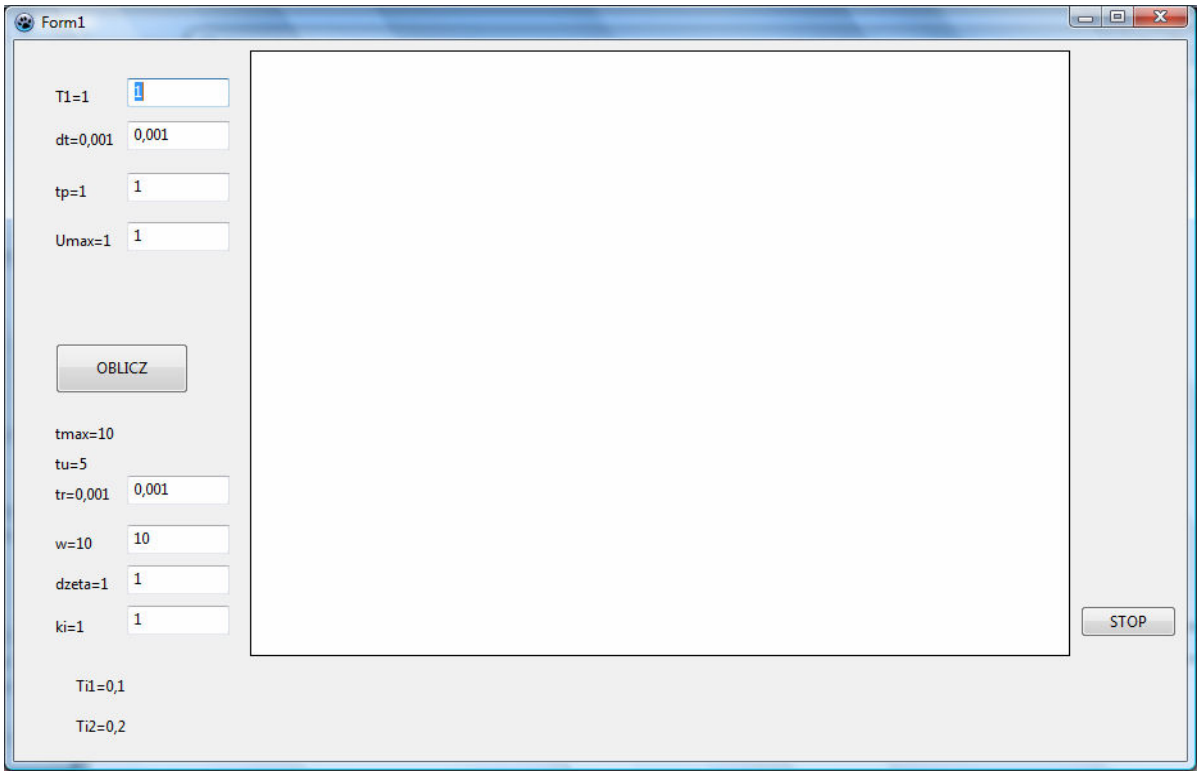
Ti1 i Ti2	stałe czasowe
k_i	współczynnik wzmocnienia
ω	częstość drgań własnych nietłumionych
ξ	współczynnik tłumienia względnego

$$T_{i1} = \frac{1}{\omega} \qquad T_{i2} = 2 \cdot T_{i1} \cdot \xi$$

Ponieważ będziemy korzystać z drugiego wzoru, zatem należy wprowadzić odpowiednie elementy na Form1 zgodnie z rys. poniżej



Oraz skojarzyć z nimi (znanym już sposobem) wszystkie parametry (zgodnie z rys. poniżej



Ponieważ w zapisie jest druga pochodna, należy rozbić układ na dwa równania, wprowadzając dodatkową zmienną y1

$$\frac{dy}{dt} = y_1$$

$$\frac{dy_1}{dt} + 2 \cdot \omega \cdot \xi \cdot y_1 + \omega^2 \cdot y = k_i \cdot \omega^2 \cdot U_s$$

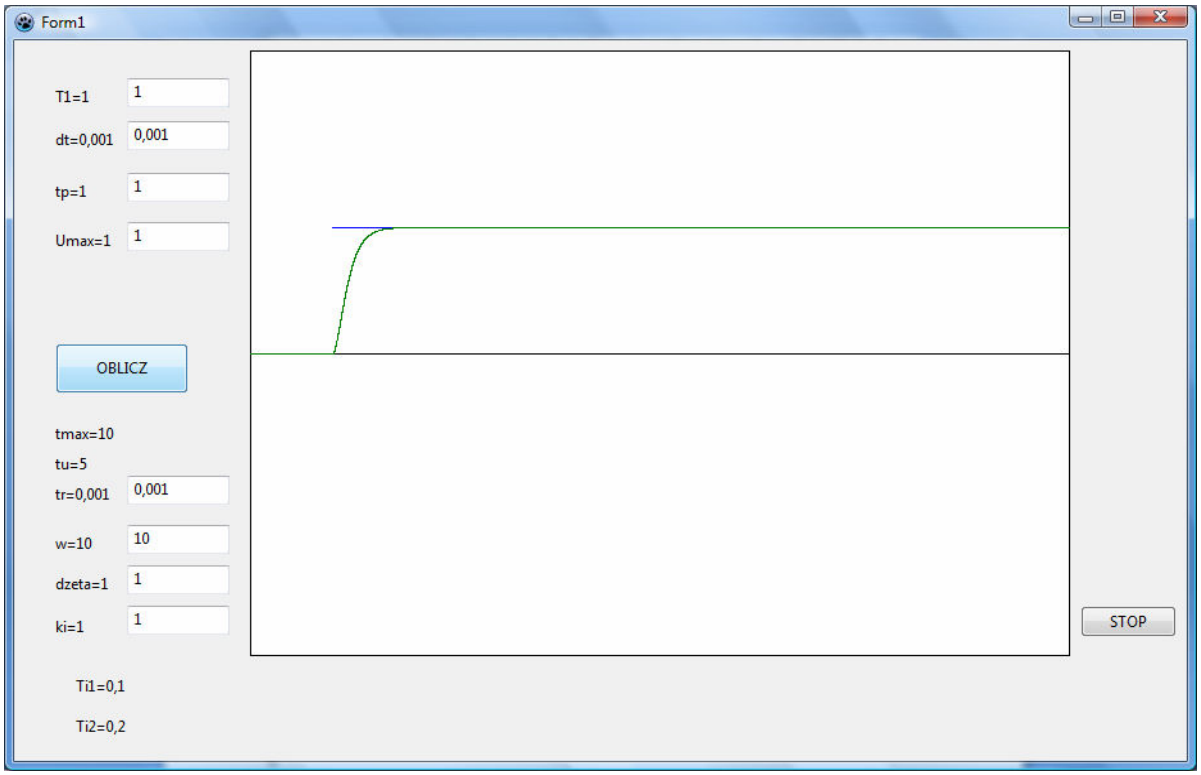
Uporządkowujemy zapis do przestrzeni zmiennych stanu:

$$\frac{dy}{dt} = y_1$$

$$\frac{dy_1}{dt} = -2 \cdot \omega \cdot \xi \cdot y_1 - \omega^2 \cdot y + k_i \cdot \omega^2 \cdot U_s$$

Wyznaczamy warunki początkowe (zerowe dla y, y1 i pochodnych) oraz przyjmujemy Us jako wymuszenie skokowe.

Rozwiązanie równań polega na całkowaniu numerycznym metodą Eulera (teoretycznie znaną z modelowania).
Wprowadzenie tego modelu do programu spowoduje pojawienie się następujących przebiegów:



Do uzyskania takiego ekranu niezbędne są zapisy wytłuszczone w listingu procedur przedstawionych poniżej:

```
var
  Form1: TForm1;
  T1:real=1; dt:real=0.001;  tp:real=1; Umax:real=1;
  czas, Us, x1, Ti1,Ti2:real;
  dy1,dy,y1,y:real;
  tr:real=0.001;
  dzeta:real=1;    omega:real=10;     ki:real=1;
const
  tmax=10;  tu=5;

implementation

{$R *.lfm}

{ TForm1 }

procedure TForm1.Button1Click(Sender: TObject);
begin
 halt;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
shape1.canvas.rectangle (0,0,640,480);
shape1.canvas.moveto(0,240);
```

```
Shape1.canvas.lineto(640,240);
T1:=strtofloat(edit1.text);
dt:=strtofloat(edit2.text);
tp:=strtofloat(edit3.text);
Umax:=strtofloat(edit4.text);
label1.caption:= 'T1='+floattostr(T1);
label2.caption:= 'dt='+floattostr(dt);
label3.caption:= 'tp='+floattostr(tp);
label4.caption:= 'Umax='+floattostr(Umax);
tr:=strtofloat(edit5.text);
label7.caption:= 'tr='+floattostr(tr);
    omega:=strtofloat(edit6.text);
    label8.caption:= 'w='+floattostr(omega);
    dzeta:=strtofloat(edit7.text);
    label9.caption:= 'dzeta='+floattostr(dzeta);
    ki:=strtofloat(edit8.text);
    label10.caption:= 'ki='+floattostr(ki);

    Ti1:=1/omega;   Ti2:=2*Ti1*dzeta;
    label11.caption:= 'Ti1='+floattostr(Ti1);
    label12.caption:= 'Ti2='+floattostr(Ti2);
czas:=0; x1:=0;  y:=0;dy:=0;y1:=0;dy1:=0;
repeat
  if czas>0 then Us:=0;
  if czas>tp then Us:= Umax*(czas-tp)/tr;
  if czas>(tp+tr) then Us:=Umax;

  dy:=y1;
  dy1:=-2*omega*dzeta*y1-omega*omega*y+ki*omega*omega*Us;
  y1:=y1+dy1*dt;
  y:=y+dy*dt;
  //if czas>=tp then x1:=cos(2*pi/T1*(czas-tp))+sin(2*pi/T1*(czas-tp))*cos(2*pi/T1*(czas-tp));
  //Shape1.Canvas.Pixels[round(640/tmax*czas),240-round(x1*100)]:=clred;
  Shape1.Canvas.Pixels[round(640/tmax*czas),240-round(us*100)]:=clblue;
  Shape1.Canvas.Pixels[round(640/tmax*czas),240-round(y*100)]:=clgreen;
czas:=czas+dt;
until czas>tmax;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
 label1.caption:= 'T1='+floattostr(T1);
 Edit1.Text:=floattostr(T1);
 label2.caption:= 'dt='+floattostr(dt);
 Edit2.Text:=floattostr(dt);
 label3.caption:= 'tp='+floattostr(tp);
```

```
 Edit3.Text:=floattostr(tp);
 label4.caption:= 'Umax='+floattostr(Umax);
 Edit4.Text:=floattostr(Umax);
 label5.caption:= 'tmax='+floattostr(tmax);
 label6.caption:= 'tu='+floattostr(tu);
 Edit5.Text:=floattostr(tr);
 label7.caption:= 'tr='+floattostr(tr);
    Edit6.Text:=floattostr(omega);
    label8.caption:= 'w='+floattostr(omega);
    Edit7.Text:=floattostr(dzeta);
    label9.caption:= 'dzeta='+floattostr(dzeta);
    Edit8.Text:=floattostr(ki);
    label10.caption:= 'ki='+floattostr(ki);
    Ti1:=1/omega;   Ti2:=2*Ti1*dzeta;
    label11.caption:= 'Ti1='+floattostr(Ti1);
    label12.caption:= 'Ti2='+floattostr(Ti2);
end;
```

Po zmianie pewnych parametrów otrzymamy: