

Rīgas Tehniskā Universitāte  
Datorzinātnes un Informācijas Tehnoloģijas fakultāte  
Lietišķo datorsistēmu institūts

**„Datorgrafikas un attēlu apstrādes pamati”  
Kontroldarbs Nr.6  
„2D pārveidojumi”**

D.I.T.F  
RDB 1.kurss 14.grupa  
Sergejs Terentjevs  
Studenta apl. 061RDB140

## 1.Darba uzdevums

Konstruēt četrstūri, veikt tā pagriešanu un palielināšanu.

## 2.Teorētiskais pamatojums

1. Konstruējam četrstūri, pielietojot līnijas veidošanas algoritmu:

**Taisne**(m[1]+100,m[3],m[2]+100,m[4]);

**Taisne**(m[5]+100,m[7],m[6]+100,m[8]);

**Taisne**(m[1]+100,m[3],m[6]+100,m[8]);

**Taisne**(m[5]+100,m[7],m[2]+100,m[4]);

2. pielietojām papildus mainīgos:

**increase** – mainīgais, kurš atbilst par palielinājumu;

**lenkis** – mainīgais, kurš apraksta mūsu definēto leņķi (**lenkis**= **m[9]\*Pi/180**);

mainīgie **a11, a12, a21, a22** – nepieciešami, lai veiktu operācijas ar mūsu četrstūri, sareizinot četrstūra punktus ar šiem mainīgajiem mēs iegūstām palielinājumus un pagriezienu.

**xs:=round(m[1]\*a11+m[3]\*a21);**

**ys:=round(m[1]\*a12+m[3]\*a22);**

**xb:=round(m[2]\*a11+m[4]\*a21);**

**yb:=round(m[2]\*a12+m[4]\*a22);**

**x2s:=round(m[5]\*a11+m[7]\*a21);**

**y2s:=round(m[5]\*a12+m[7]\*a22);**

**x2b:=round(m[6]\*a11+m[8]\*a21);**

**y2b:=round(m[6]\*a12+m[8]\*a22);**

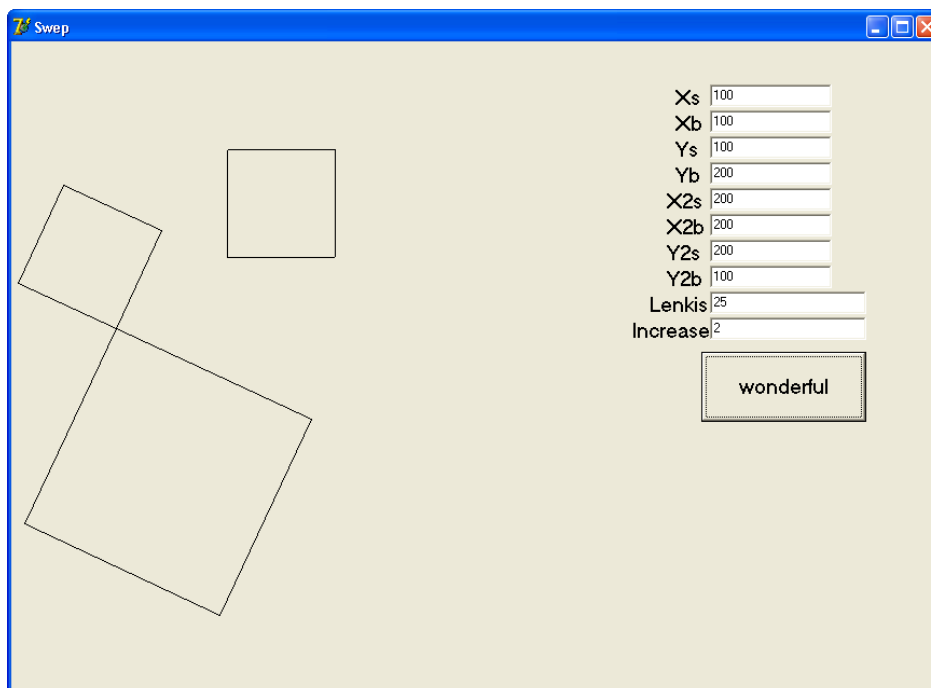
3. Attēlojām pagriezto četrstūri:

**Taisne(xs, ys, xb ,yb);**

**Taisne(xb,yb,x2s,y2s);**

**Taisne (x2s,y2s,x2b,y2b);**

**Taisne (x2b,y2b,xs,ys);**



### 3. Programmas kods

```
procedure TForm1.Button1Click(Sender: TObject);
    var m:array[1..9] of integer;
    i,xs,ys,x2s,y2s,xb,yb,x2b,y2b:integer;
    a11,a21,a12,a22,lenkis,increase:real;
begin
    increase:=StrToInt(Edit10.Text);
    For i:=1 to 9 do
        m[i]:=StrToInt(TEdit(FindComponent('Edit'+IntToStr(i))).Text);
        lenkis:=m[9]*Pi/180;

    a11:=cos(lenkis)*increase;
    a21:=-sin(lenkis)*increase;
    a12:=Sin(lenkis)*increase;
    a22:=Cos(lenkis)*increase;

    xs:= round(m[1]*a11+m[3]*a21);
    ys:= round(m[1]*a12+m[3]*a22);
    xb:= round(m[2]*a11+m[4]*a21);
    yb:= round(m[2]*a12+m[4]*a22);
    x2s:=round(m[5]*a11+m[7]*a21);
    y2s:=round(m[5]*a12+m[7]*a22);
    x2b:=round(m[6]*a11+m[8]*a21);
    y2b:=round(m[6]*a12+m[8]*a22);

    Taisne(xs,ys,xb,yb);
    Taisne(xb,yb,x2s,y2s);
    Taisne (x2s,y2s,x2b,y2b);
    Taisne (x2b,y2b,xs,ys);

    Taisne(m[1]+100,m[3],m[2]+100,m[4]);
    Taisne(m[5]+100,m[7],m[6]+100,m[8]);
    Taisne(m[1]+100,m[3],m[6]+100,m[8]);
    Taisne(m[5]+100,m[7],m[2]+100,m[4]);

end;
```

### 4. secinājumi

Programmas algoritmu bija ļoti izdevīgi realizēt ar papildus mainīgiem , kuri ietvēra matemātiskas darbības, tādejādi mēs ieguvām kompakto un atvieglotāko kodu.