

Przesunięcie (translacja):

$$\begin{bmatrix} a \\ b \\ c \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & v \\ 0 & 1 & 0 & w \\ 0 & 0 & 1 & u \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Skalowanie:

$$\begin{bmatrix} a \\ b \\ c \\ 1 \end{bmatrix} = \begin{bmatrix} sx & 0 & 0 & 0 \\ 0 & sy & 0 & 0 \\ 0 & 0 & sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Obrót wokół osi X:

$$\begin{bmatrix} a \\ b \\ c \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Obrót wokół osi Y:

$$\begin{bmatrix} a \\ b \\ c \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \alpha & 0 & \sin \alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \alpha & 0 & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Obrót wokół osi Z:

$$\begin{bmatrix} a \\ b \\ c \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 & 0 \\ \sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Do wyznaczania obrazów przekształceń stosuje się macierze 4-wierszowe w tzw. zapisie jednorodnym

https://pl.wikipedia.org/wiki/Wsp%C3%B3%C5%82rz%C4%99dne_jednorodne

Punkt, który ma współrzędne (x, y, z) jest przedstawiany w postaci jednorodnej (x, y, z, 1).

PRZEKSZTAŁCENIA 3D
(x, y, z) \longrightarrow (a, b, c)

wzory macierzowe

-
- 1) Przesunięcie o wektor (v, w, u)
 - 2) Skalowanie o skali (sx, sy, sz)
 - 3) Obroty o kąt α

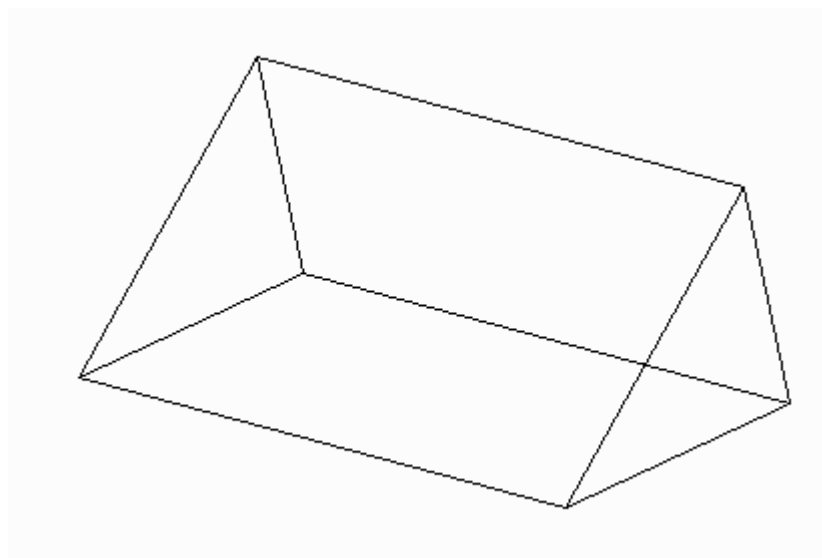
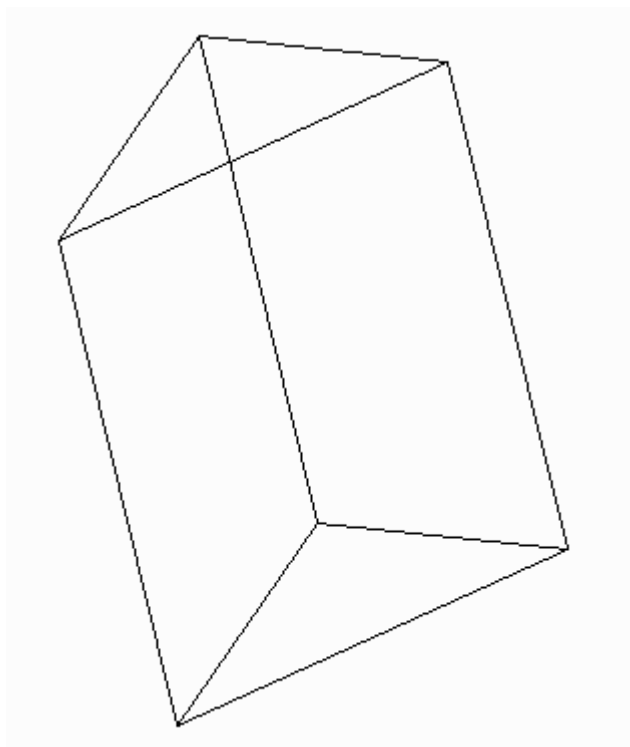
Przykład zastosowania: Obroty 3D (kod w języku Pascal w trybie graficznym)

Zamiast modułu **Crt** zastosowano bibliotekę **WinCrt** w celu zapewnienia właściwego działania warunku **KeyPressed**. Konieczne jest użycie standardowego modułu **Graph**.

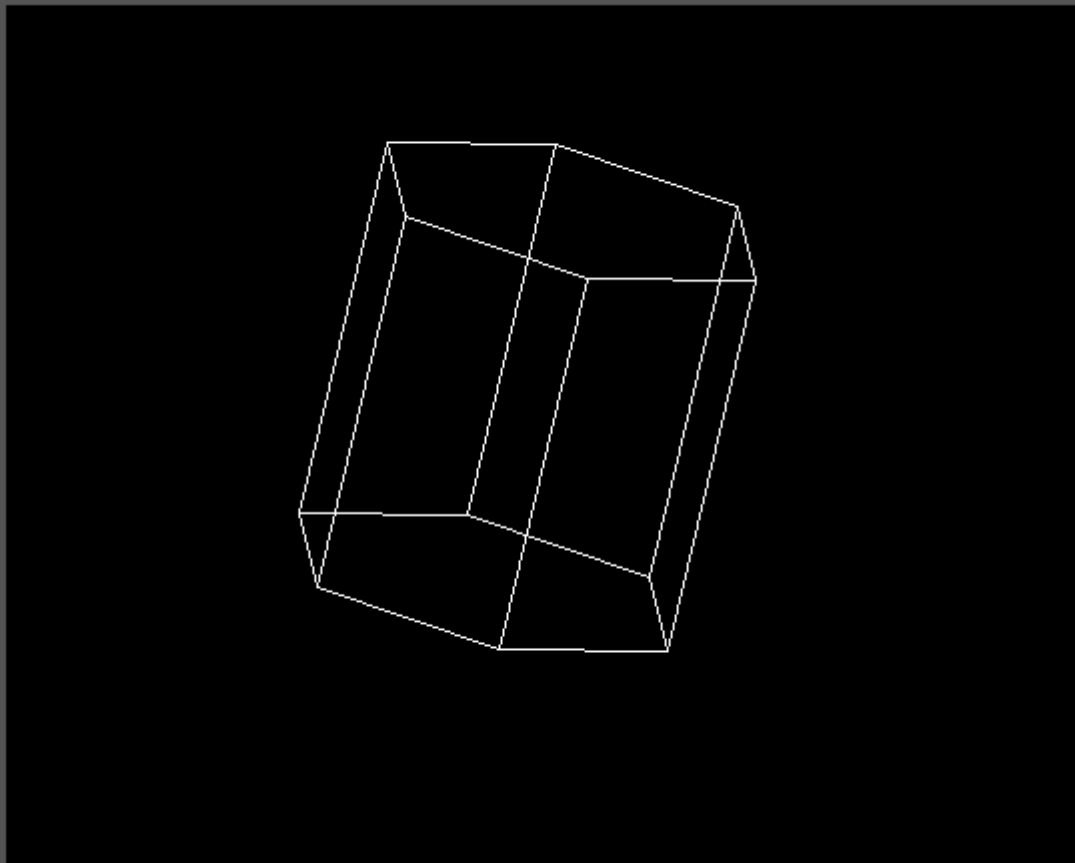
Program pozwala na obracanie graniastosłupa trójkątnego przy użyciu klawiszy strzałek.

Rozpoczęcie wyświetlania następuje po naciśnięciu klawisza <ENTER> lub <SPACJA>.

Zakończenie programu po naciśnięciu klawisza <ESC>.



Drobne modyfikacje programu dotyczące ilości wierzchołków i współrzędnych początkowych układu XYZ umożliwiają uzyskanie rysunków innych wielościanów z możliwością ich obrotów.



```

program bryla_4;
uses WinCrt,Graph;
type
  macierz=array[1..4,1..4] of real;
  typ=array [0..6] of macierz;
var
  znak :char;
  kod :integer;
  obrotx,obroty,obrotz :integer;
  xp,yp,zp :real;
  figura :typ;
  T,Rx,Ry,Rz,Skala,A,B :macierz;

{***** BUFOR *****}
procedure bufor;
begin
  while keypressed do znak:=readkey;
end;

{***** ILOCZYN MACIERZY *****}
procedure iloczyn(m,n,p :integer;A,B :macierz;var C :macierz);
var i,j,k :integer;
    suma :real;
begin
  for i:=1 to m do
    for j:=1 to p do
      begin

```

```

    suma:=0;
    for k:=1 to n do
        suma:=suma+A[i,k]*B[k,j];
    C[i,j]:=suma;
    end;
end;

```

```

{***** KRAWEDZ *****}
procedure krawedz(var fig3d :typ; n1,n2 :byte;x,y,kolor :integer);
var x1,y1,x2,y2 :integer;
begin
    setcolor(kolor);
    x1:=round(fig3d[n1][1,1]);
    y1:=round(fig3d[n1][2,1]);
    x2:=round(fig3d[n2][1,1]);
    y2:=round(fig3d[n2][2,1]);
    line(x1+x,y1+y,x2+x,y2+y);
end;

```

```

{***** OBROT X *****}
procedure mac_Rx(alfa: real);
var i,j :integer;
    kat :real;
begin
    for i:=1 to 4 do
    begin
        for j:=1 to 4 do Rx[i,j]:=0;
        Rx[i,i]:=1;
        end;
        kat:=alfa*pi/180;
        Rx[2,2]:=cos(kat);
        Rx[3,3]:=Rx[2,2];
        Rx[2,3]:=sin(kat);
        Rx[3,2]:=-Rx[2,3];
    end;
end;

```

```

{***** OBROT Y *****}
procedure mac_Ry(alfa: real);
var i,j :integer;
    kat :real;
begin
    for i:=1 to 4 do
    begin
        for j:=1 to 4 do Ry[i,j]:=0;
        Ry[i,i]:=1;
        end;
        kat:=alfa*pi/180;
        Ry[1,1]:=cos(kat);
        Ry[3,3]:=Ry[1,1];
        Ry[3,1]:=sin(kat);
        Ry[1,3]:=-Ry[3,1];
    end;
end;

```

```

{***** OBROT Z *****)
procedure mac_Rz(alfa: real);
var i,j :integer;
    kat :real;
begin
for i:=1 to 4 do
begin
for j:=1 to 4 do Rz[i,j]:=0;
Rz[i,i]:=1;
end;
kat:=alfa*pi/180;
Rz[1,1]:=cos(kat);
Rz[2,2]:=Rz[1,1];
Rz[1,2]:=sin(kat);
Rz[2,1]:=-Rz[1,2];
end;

{***** WSPOLRZEDNE *****)
procedure wsp(var fig3d :typ; n :byte; x,y,z :real);
begin
fig3d[n][1,1]:=x;
fig3d[n][2,1]:=y;
fig3d[n][3,1]:=z;
fig3d[n][4,1]:=1;
end;

{***** WSPOLRZEDNE POZATKOWE *****)
procedure wsp_pocz;
begin
wsp(figura,0,-40,-140,-100);
wsp(figura,1,110,-140,-100);
wsp(figura,2,0,-140,100);
wsp(figura,3,-40,140,-100);
wsp(figura,4,110,140,-100);
wsp(figura,5,0,140,100);
wsp(figura,6,0,-140,-100);
end;

{***** RYSUJ *****)
procedure rysuj(x,y :integer);
var i,j :integer;
begin
krawedz(figura,0,1,x,y,0);
krawedz(figura,1,2,x,y,0);
krawedz(figura,2,0,x,y,0);
krawedz(figura,3,4,x,y,0);
krawedz(figura,4,5,x,y,0);
krawedz(figura,5,3,x,y,0);
krawedz(figura,0,3,x,y,0);
krawedz(figura,1,4,x,y,0);
krawedz(figura,2,5,x,y,0);
end;

```

```

{***** GRAFIKA *****}
procedure grafika;
var karta, tryb :integer;
begin
  karta:=detect;
  InitGraph(karta, tryb, '');
end;

{***** RYSUNEK *****}
procedure rysunek;
var k: integer;
begin
  bar(80,25,614,454);
  rysuj(340,240);
  delay(10);
  for k:=0 to 6 do
  begin
    iloczyn(4,4,1,B,figura[k],A);
    figura[k]:=A;
  end;
  delay(100);
end;

{***** MAIN *****}
begin
  wsp_pocz;

  grafika;
  cleardevice;
  setfillstyle(1,15);
REPEAT
  bufor;

  repeat
    mac_Rz(0);
    mac_Ry(0);
    mac_Rx(0);
    iloczyn(4,4,4,Ry,Rx,B);
    repeat
      until keypressed;
      kod:=ord(readkey);
      until ((kod=75) or (kod=72) or (kod=77) or (kod=80)
        or (kod=13) or (kod=32) or (kod=27));
    case kod of
      72: begin
          obrotx:=5;
          mac_Rx(obrotx);
          iloczyn(4,4,4,Ry,Rx,B);
          rysunek;
        end;
      80: begin
          obrotx:=-5;
          mac_Rx(obrotx);
          iloczyn(4,4,4,Ry,Rx,B);
          rysunek;
        end;
      77: begin

```

```
    obroty:=5;
    mac_Ry(obroty);
    iloczyn(4,4,4,Ry,Rx,B);
    rysunek;
    end;
75: begin
    obroty:=-5;
    mac_Ry(obroty);
    iloczyn(4,4,4,Ry,Rx,B);
    rysunek;
    end;

27: begin
    cleardevice;
    halt;
    end;
end; {case}
    rysunek;
UNTIL 1=2;
READLN;
CloseGraph;
end.
```