МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Вятский государственный университет»**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Реализация двумерного алгоритма отсечения отрезка Сазерленда-Коэна

Отчет по лабораторной работе №6

по дисциплине «Компьютерная графика»

Выполнил студент группы ИВТ-21\_\_\_\_\_\_\_\_\_\_\_\_/Коротаев Р.С./

Проверил преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Коржавина А.С./

Киров 2020

Цель работы: научиться применять математический аппарат проекций для визуализации объемных геометрических тел.

Задание:

1. Описать брусок в приборной системе координат.
2. Вывести на экран три его ортогональные проекции (вид спереди, сверху, сбоку).
3. Продемонстрировать три прямоугольные аксонометрические проекции данного бруска (изометрию, диметрию, тpиметpию).
4. Постpоить две косоугольные аксонометрические проекции бруска (кавалье, кабине).
5. Показать одноточечную центральную проекцию бруска.

Схема алгоритма

Схема алгоритма Сазерленда-Коэна представлена на рисунке 1.

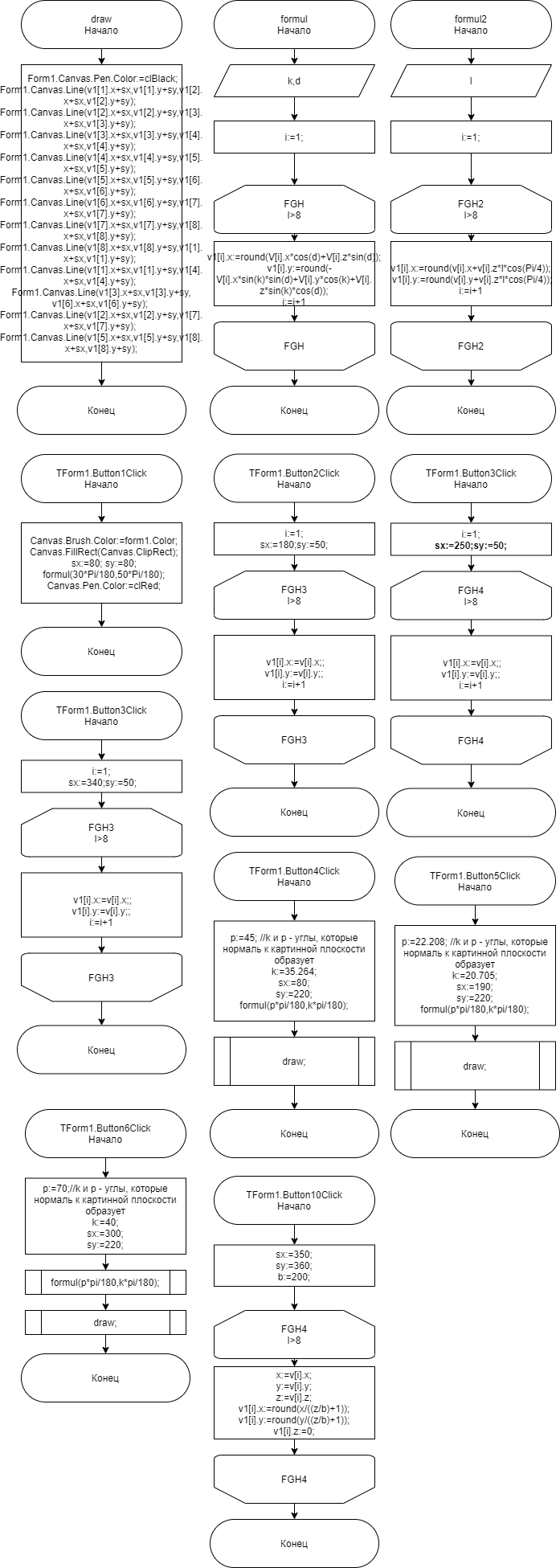


Рисунок 1 – Схема алгоритма поиска новых координат

Экранные формы работы программы

Экранные формы программы представлены в приложении А.

Код программы

Листинг программы представлен в приложении Б.

Вывод

В ходе лабораторной работы были реализованы матрицы проецирования, такие как одноточечные, ортографические, аксонометрические прямоугольные параллельные проекции и косоугольная параллельная аксонометрия.

Приложение А

Экранные формы

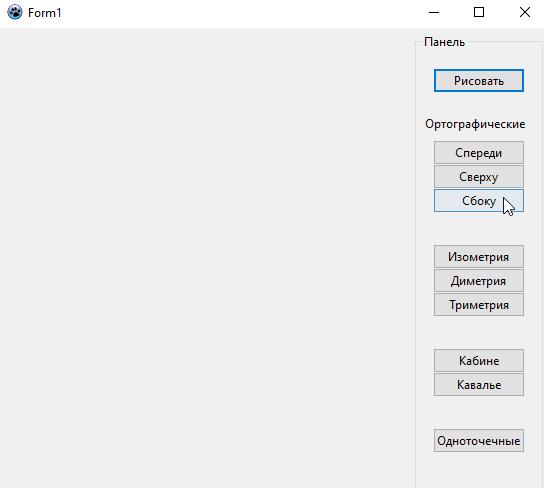


Рисунок 2 – экранная форма работы программы до отсечения

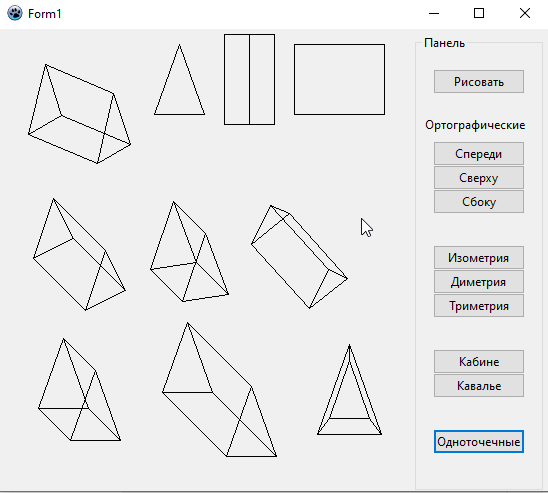


Рисунок 3 – экранная форма работы программы после отсечения

Приложение Б

Листинг программы

**unit** Unit1;

{$mode objfpc}{$H+}

**interface**

**uses**

Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs, ExtCtrls,

Buttons, StdCtrls;

**type**

{ TForm1 }

TForm1 = **class**(TForm)

Button1: TButton;

Button10: TButton;

Button2: TButton;

Button3: TButton;

Button4: TButton;

Button5: TButton;

Button6: TButton;

Button7: TButton;

Button8: TButton;

Button9: TButton;

GroupBox1: TGroupBox;

Label1: TLabel;

**procedure** Button10Click(Sender: TObject);

**procedure** Button1Click(Sender: TObject);

**procedure** Button2Click(Sender: TObject);

**procedure** Button3Click(Sender: TObject);

**procedure** Button4Click(Sender: TObject);

**procedure** Button5Click(Sender: TObject);

**procedure** Button6Click(Sender: TObject);

**procedure** Button7Click(Sender: TObject);

**procedure** Button8Click(Sender: TObject);

**procedure** Button9Click(Sender: TObject);

**private**

{ private declarations }

**public**

{ public declarations }

**end**;

**var**

Form1: TForm1;

**implementation**

**type**

TXYZ=**record**

x,y,z:integer;

**end**;

**const** V:**array**[1..8] **of** TXYZ=

((x:-25;y: 35;z: -45), //нижний левый

(x:0;y: -35;z: -45), //верхний левый

(x:0;y: -35;z: -45),//верхний правый

(x:25;y: 35;z: -45), //нижний правый

(x:25;y: 35;z: 45), //нижний правый

(x:0;y: -35;z: 45), //верхний правый

(x:0;y: -35;z: 45), //верхний левый

(x:-25;y: 35;z: 45)); //нижний левый

**var**

v1:**array**[1..8] **of** TXYZ;

i,sx,sy:integer;

{$R \*.lfm}

{ TForm1 }

**procedure** draw;

**begin**

Form1.Canvas.Pen.Color:=clBlack;

Form1.Canvas.Line(v1[1].x+sx,v1[1].y+sy,v1[2].x+sx,v1[2].y+sy);

Form1.Canvas.Line(v1[2].x+sx,v1[2].y+sy,v1[3].x+sx,v1[3].y+sy);

Form1.Canvas.Line(v1[3].x+sx,v1[3].y+sy,v1[4].x+sx,v1[4].y+sy);

Form1.Canvas.Line(v1[4].x+sx,v1[4].y+sy,v1[5].x+sx,v1[5].y+sy);

Form1.Canvas.Line(v1[5].x+sx,v1[5].y+sy,v1[6].x+sx,v1[6].y+sy);

Form1.Canvas.Line(v1[6].x+sx,v1[6].y+sy,v1[7].x+sx,v1[7].y+sy);

Form1.Canvas.Line(v1[7].x+sx,v1[7].y+sy,v1[8].x+sx,v1[8].y+sy);

Form1.Canvas.Line(v1[8].x+sx,v1[8].y+sy,v1[1].x+sx,v1[1].y+sy);

Form1.Canvas.Line(v1[1].x+sx,v1[1].y+sy,v1[4].x+sx,v1[4].y+sy);

Form1.Canvas.Line(v1[3].x+sx,v1[3].y+sy,v1[6].x+sx,v1[6].y+sy);

Form1.Canvas.Line(v1[2].x+sx,v1[2].y+sy,v1[7].x+sx,v1[7].y+sy);

Form1.Canvas.Line(v1[5].x+sx,v1[5].y+sy,v1[8].x+sx,v1[8].y+sy);

**end**;

**procedure** formul(k,d:real);

**begin**

**for** i:=1 **to** 8 **do begin**

v1[i].x:=round(V[i].x\*cos(d)+V[i].z\*sin(d));

v1[i].y:=round(-V[i].x\*sin(k)\*sin(d)+V[i].y\*cos(k)+V[i].z\*sin(k)\*cos(d));

**end**;

**end**;

**procedure** formul2(l:real);

**begin**

**for** i:=1 **to** 8 **do**

**begin**

v1[i].x:=round(v[i].x+v[i].z\*l\*cos(Pi/4)); //pi=3.14159 Для получения координат проекции любой точки изображения необходимо

v1[i].y:=round(v[i].y+v[i].z\*l\*cos(Pi/4)); //pi=3.14159 исходные координаты этой точки перемножить с соответствующей матрицей.

**end**;

**end**;

**procedure** TForm1.Button1Click(Sender: TObject);

**begin**

Canvas.Brush.Color:=form1.Color;

Canvas.FillRect(Canvas.ClipRect);

sx:=80; sy:=80;

formul(30\*Pi/180,50\*Pi/180);

Canvas.Pen.Color:=clRed;

draw;

**end**;

**procedure** TForm1.Button2Click(Sender: TObject);//Фронтальная проекция(z)

**begin**

sx:=180;sy:=50;

**for** i:=1 **to** 8 **do**

**begin**

v1[i].x:=v[i].x; //При построении вида спереди координаты z точек проекции отбрасываются

v1[i].y:=v[i].y;

**end**;

draw;

**end**;

**procedure** TForm1.Button3Click(Sender: TObject);//(y)

**begin**

sx:=250;sy:=50;

**for** i:=1 **to** 8 **do**

**begin** //сверху

v1[i].x:=v[i].x; //x - координата остается без изменения. y - координату заменяют на z

v1[i].y:=v[i].z;

**end**;

draw;

**end**;

**procedure** TForm1.Button9Click(Sender: TObject); //(x)

**begin**

sx:=340;sy:=50;

**for** i:=1 **to** 8 **do**

**begin**

v1[i].x:=v[i].z; //При построении вида сбоку х-координату точки проекции заменяют координатой z. y - координата остается без изменения

v1[i].y:=v[i].y;

**end**;

draw;

**end**;

**procedure** TForm1.Button4Click(Sender: TObject);//изометрия

**var** p,k:real;

**begin**

p:=45; //k и p - углы, которые нормаль к картинной плоскости образует

k:=35.264;

sx:=80;

sy:=220;

formul(p\*pi/180,k\*pi/180);

draw;

**end**;

**procedure** TForm1.Button5Click(Sender: TObject);//диметpия

**var** p,k:real;

**begin**

p:=22.208; //k и p - углы, которые нормаль к картинной плоскости образует

k:=20.705;

sx:=190;

sy:=220;

formul(p\*pi/180,k\*pi/180);

draw;

**end**;

**procedure** TForm1.Button6Click(Sender: TObject);//триметрия

**var** p,k:real;

**begin**

p:=70;//k и p - углы, которые нормаль к картинной плоскости образует

k:=40;

sx:=300;

sy:=220;

formul(p\*pi/180,k\*pi/180);

draw;

**end**;

**procedure** TForm1.Button7Click(Sender: TObject);//кабине

**var** l:real;

**begin**

l:=0.5; //из матрицы для этой проекции l=0.5

sx:=80;

sy:=360;

formul2(l);

draw;

**end**;

**procedure** TForm1.Button8Click(Sender: TObject);//кавалье

**var** l:real;

**begin**

l:=1; //из матрицы для этой проекции l = 1 для кавалье

sx:=220;

sy:=360;

formul2(l);

draw;

**end**;

**procedure** TForm1.Button10Click(Sender: TObject);//одноточечные

**var** x,y,z,b:integer;

**begin**

sx:=350;

sy:=360;

b:=200;

**for** i:=1 **to** 8 **do begin** //Для получения проекции точки в пространстве с координатами (x,y,z,1) необходимо найти ее новые однородные, а затем - новые координаты (x'y')

x:=v[i].x;

y:=v[i].y;

z:=v[i].z;

v1[i].x:=round(x/((z/b)+1));

v1[i].y:=round(y/((z/b)+1));

v1[i].z:=0;

**end**;

draw;

**end**;

**end**.