МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Вятский государственный университет»**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Представление и проецирование трехмерных объектов

Отчет по лабораторной работе №9

по дисциплине «Компьютерная графика»

Выполнил студент группы ИВТ-21 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Козюков М.Д./

Проверил преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Клюкин В.Л./

Киров 2018

1 Цель работы

Научиться применять математический аппарат проекций для визуализации объемных геометрических тел.

2 Постановка задачи

Написать на языке PASCAL программу:

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. Построить две косоугольные аксонометрические проекции бруска (кавалье, кабине).
5. Показать одноточечную центральную проекцию бруска.

3 Алгоритм работы программы

Алгоритм работы программы представлен на рисунке 1.



Рисунок 1 – Алгоритм работы программы

4 Экранная форма работы программы

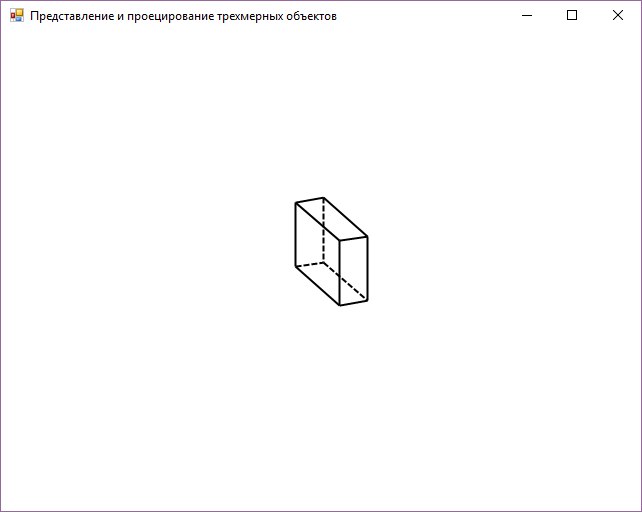


Рисунок 2 – Экранная форма работы программы

5 Код программы

Код программы представлен в приложении А.

6 Вывод

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

Приложение А

(обязательное)

Код программы

**uses GraphABC;**

**const**

**L = 200;**

**sTime = 1;**

**dRot = 60;**

**type**

**pix3D = record**

**X,Y,Z : Real;**

**constructor; begin (X, Y, Z) := (0, 0, 0); end;**

**constructor (X, Y, Z : Real); begin (Self.X, Self.Y, Self.Z) := (X, Y, Z); end;**

**function rotationX(angle : Real) := new pix3D(X, Y\*cos(angle)+Z\*sin(angle), -Y\*sin(angle)+z\*cos(angle));**

**function rotationY(angle : Real) := new pix3D(X\*cos(angle)-z\*sin(angle), Y, X\*sin(angle)+Z\*cos(angle));**

**function rotationZ(angle : Real) := new pix3D(X\*cos(angle)+Y\*sin(angle), -X\*sin(angle)+Y\*cos(angle), Z);**

**function winX := Window.Center.X + Round(X);**

**function winY := Window.Center.Y + Round(Y);**

**end;**

**var pix : array of pix3D = (**

**new pix3D(+L,+L,+L), new pix3D(+L,+L,-L), new pix3D(+L,-L,+L), new pix3D(+L,-L,-L),**

**new pix3D(-L,+L,+L), new pix3D(-L,+L,-L), new pix3D(-L,-L,+L), new pix3D(-L,-L,-L)**

**);**

**flag:boolean;**

**procedure Li(a, b : pix3D) := Line(a.winX, a.winY, b.winX, b.winY);**

**procedure DrawCube(angleX, angleY, angleZ : Real);**

**begin**

**var r : array of pix3D;**

**SetLength(r, 8);**

**for var i := 0 to 7 do r[i] := pix[i].rotationX(angleX).rotationY(angleY).rotationZ(angleZ);**

**Li(r[0],r[1]); Li(r[0],r[2]); Li(r[1],r[3]); Li(r[2],r[3]); Li(r[0],r[4]); Li(r[1],r[5]);**

**Li(r[4],r[5]); Li(r[2],r[6]); Li(r[4],r[6]); Li(r[3],r[7]); Li(r[5],r[7]); Li(r[6],r[7]);**

**end;**

**procedure KeyDown(Key: integer);**

**var i, t : integer;**

**rotX, rotY, rotZ : real;**

**begin**

**flag := false;**

**case Key of**

**VK\_Q : begin**

**(rotX, rotY, rotZ) := (-pi/4,pi/4,pi/2);**

**flag := true;**

**end;**

**VK\_W : begin**

**(rotX, rotY, rotZ) := (pi/6,pi/6,pi/2);**

**LockDrawing;**

**repeat**

**Window.Clear;**

**rotX -= 1/dRot;**

**DrawCube(rotX, rotY, rotZ);**

**Redraw;**

**Sleep(sTime);**

**until flag=true;**

**end;**

**VK\_E : begin**

**(rotX, rotY, rotZ) := (0,0,0);**

**flag := true;**

**end;**

**VK\_R : begin**

**(rotX, rotY, rotZ) := (pi/2,0,pi/2);**

**flag:=true;**

**end;**

**VK\_T : begin**

**(rotX, rotY, rotZ) := (pi/6,pi/6,pi/2);**

**end;**

**VK\_Y : begin**

**(rotX, rotY, rotZ) := (pi/6,pi/4,pi/2);**

**end;**

**VK\_Escape : begin**

**Window.Close;**

**end;**

**VK\_U : begin**

**(rotX, rotY, rotZ) := (pi/3,pi/6,pi/2);**

**end;**

**VK\_Left : begin**

**(rotX, rotY, rotZ) := (pi/6,pi/6,pi/2);**

**for t := 0 to 7 do begin**

**pix[t].X := pix[t].X - 5;**

**pix[t].Y := pix[t].Y - 5;**

**end;**

**end;**

**VK\_Right : begin**

**(rotX, rotY, rotZ) := (pi/6,pi/6,pi/2);**

**for t := 0 to 7 do begin**

**pix[t].X := pix[t].X + 5;**

**pix[t].Y := pix[t].Y + 5;**

**end;**

**end;**

**VK\_Up : begin**

**(rotX, rotY, rotZ) := (pi/6,pi/6,pi/2);**

**for t := 0 to 7 do begin**

**pix[t].Z := pix[t].Z - 5;**

**end;**

**end;**

**VK\_Down : begin**

**(rotX, rotY, rotZ) := (pi/6,pi/6,pi/2);**

**for t := 0 to 7 do begin**

**pix[t].Z := pix[t].Z + 5;**

**end;**

**end;**

**end;**

**FillRect(0, 0, Window.Width, Window.Height);**

**DrawCube(rotX, rotY, rotZ);**

**end;**

**begin**

**Window.Width := 800;**

**Window.Height := 800;**

**Window.CenterOnScreen;**

**OnKeyDown := KeyDown;**

**SetPenColor(clRed);**

**SetPenWidth(3);**

**DrawCube(pi/6,pi/6,pi/2);**

**end.**