Учреждение образования

«Гомельский Государственный технический университет имени П.О.Сухого»

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

Отчет по лабораторной работе №3

«Реализация программ отсечения»

По дисциплине «Алгоритмические основы современной компьютерной графики»

Выполнил студент гр.ИТЗ-21

Касаткин Д.О.

Проверил преподаватель

Комракова М.В.

Гомель 2019

**Цель работы:** получить теоретические и практические знания выполнения сечений геометрических двумерных фигур.

**Задание.** Разработать программу, выполняющую рассечение заданной геометрической фигуры на части прямой линией, задаваемой пользователем в виде отрезка.

Листинг программы:

using SharpGL;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab\_3

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

create\_osxy();

}

List<Osxy> osxy\_rhombus = new List<Osxy>();

List<Osxy> kub = new List<Osxy>();

List<Osxy> line = new List<Osxy>();

List<Osxy> down\_buffer = new List<Osxy>();

List<Osxy> down\_buffer\_kub = new List<Osxy>();

private void create\_osxy()

{

osxy\_rhombus.Add(new Osxy(0f, 6f));

osxy\_rhombus.Add(new Osxy(3f, 0f));

osxy\_rhombus.Add(new Osxy(0f, -6f));

osxy\_rhombus.Add(new Osxy(-3f, 0f));

kub.Add(new Osxy(2f, 2f));

kub.Add(new Osxy(2f, -2f));

kub.Add(new Osxy(-2f, -2f));

kub.Add(new Osxy(-2f, 2f));

line.Add(new Osxy(5, -8));

line.Add(new Osxy(-4, 7));

}

private void Button1\_Click(object sender, EventArgs e)

{

drow\_figures(true);

float a, b, c, t, d;

int x1, x2, x3, y1, y2, y3;

}

private void drow\_figures(bool up)

{

OpenGL gl = this.openGLControl1.OpenGL;

// Очистка экрана и буфера глубин

gl.Clear(OpenGL.GL\_COLOR\_BUFFER\_BIT | OpenGL.GL\_DEPTH\_BUFFER\_BIT);

// Сбрасываем модельно-видовую матрицу

gl.LoadIdentity();

gl.Translate(0.0f, 0.0f, -20.0f);

if (up)

{

gl.Begin(OpenGL.GL\_LINE\_LOOP);

gl.Color(0f, 1f, 0f);

foreach (Osxy figures in osxy\_rhombus)

{

gl.Vertex(figures.x, figures.y);

}

gl.End();

gl.Begin(OpenGL.GL\_LINE\_LOOP);

gl.Color(0f, 1f, 0f);

foreach (Osxy figures in kub)

{

gl.Vertex(figures.x, figures.y);

}

gl.End();

}

bool bolean\_f = false; Osxy temp = new Osxy(0, 0);

foreach (Osxy figures in osxy\_rhombus)

{

if (bolean\_f)

{

are\_crossing1(figures.x, figures.y, temp.x, temp.y);

temp = new Osxy(figures.x, figures.y);

}

else

{

bolean\_f = true;

temp = new Osxy(figures.x, figures.y);

}

}

are\_crossing1(osxy\_rhombus[0].x, osxy\_rhombus[0].y, temp.x, temp.y);

gl.Begin(OpenGL.GL\_LINE\_LOOP);

// Указываем цвет вершин

if (up)

gl.Color(0f, 0f, 0f);

else

gl.Color(1f, 0f, 0f);

foreach (Osxy figures in down\_buffer)

{

gl.Vertex(figures.x, figures.y);

}

gl.End();

temp\_b\_fl = false;

bolean\_f = false;

foreach (Osxy figures in kub)

{

if (bolean\_f)

{

are\_crossing2(figures.x, figures.y, temp.x, temp.y);

temp = new Osxy(figures.x, figures.y);

}

else

{

bolean\_f = true;

temp = new Osxy(figures.x, figures.y);

}

}

are\_crossing2(kub[0].x, kub[0].y, temp.x, temp.y);

gl.Begin(OpenGL.GL\_LINE\_LOOP);

if (up)

gl.Color(0f, 0f, 0f);

else

gl.Color(1f, 0f, 0f);

foreach (Osxy figures in down\_buffer\_kub)

{

gl.Vertex(figures.x, figures.y);

}

gl.End();

gl.Begin(OpenGL.GL\_LINES);

gl.Color(0f, 1f, 1f);

foreach (Osxy figures in line)

{

gl.Vertex(figures.x, figures.y);

}

gl.End();

down\_buffer\_kub.Clear();

down\_buffer.Clear();

}

private void Button2\_Click(object sender, EventArgs e)

{

drow\_figures(false);

}

bool temp\_b\_fl = false;

public void are\_crossing1(double x1, double y1, double x\_temp, double y\_temp)

{

double a1, b1, c1, a2, b2, c2;

a2 = line[1].y - line[0].y;

b2 = line[0].x - line[1].x;

c2 = -line[0].x \* (line[1].y - line[0].y) + line[0].y \* (line[1].x - line[0].x);

double znak = a2 \* x1 + b2 \* y1 + c2;

if (znak < 0)

{

if (temp\_b\_fl != true)

{

a1 = y\_temp - y1;

b1 = x1 - x\_temp;

c1 = -x1 \* (y\_temp - y1) + y1 \* (x\_temp - x1);

double x\_new = -(c1 \* b2 - c2 \* b1) / (a1 \* b2 - a2 \* b1);

double y\_new = -(a1 \* c2 - a2 \* c1) / (a1 \* b2 - a2 \* b1);

down\_buffer.Add(new Osxy(x\_new, y\_new));

down\_buffer.Add(new Osxy(x1, y1));

temp\_b\_fl = true;

}

else

{

down\_buffer.Add(new Osxy(x1, y1));

}

}

else

{

if (temp\_b\_fl)

{

a1 = y\_temp - y1;

b1 = x1 - x\_temp;

c1 = -x1 \* (y\_temp - y1) + y1 \* (x\_temp - x1);

double x\_new = -(c1 \* b2 - c2 \* b1) / (a1 \* b2 - a2 \* b1);

double y\_new = -(a1 \* c2 - a2 \* c1) / (a1 \* b2 - a2 \* b1);

down\_buffer.Add(new Osxy(x\_new, y\_new));

temp\_b\_fl = false;

}

}

}

public void are\_crossing2(double x1, double y1, double x\_temp, double y\_temp)

{

double a1, b1, c1, a2, b2, c2;

a2 = line[1].y - line[0].y;

b2 = line[0].x - line[1].x;

c2 = -line[0].x \* (line[1].y - line[0].y) + line[0].y \* (line[1].x - line[0].x);

double znak = a2 \* x1 + b2 \* y1 + c2;

if (znak < 0)

{

if (temp\_b\_fl != true)

{

a1 = y\_temp - y1;

b1 = x1 - x\_temp;

c1 = -x1 \* (y\_temp - y1) + y1 \* (x\_temp - x1);

double x\_new = -(c1 \* b2 - c2 \* b1) / (a1 \* b2 - a2 \* b1);

double y\_new = -(a1 \* c2 - a2 \* c1) / (a1 \* b2 - a2 \* b1);

down\_buffer\_kub.Add(new Osxy(x\_new, y\_new));

down\_buffer\_kub.Add(new Osxy(x1, y1));

temp\_b\_fl = true;

}

else

{

down\_buffer.Add(new Osxy(x1, y1));

}

}

else

{

if (temp\_b\_fl)

{

a1 = y\_temp - y1;

b1 = x1 - x\_temp;

c1 = -x1 \* (y\_temp - y1) + y1 \* (x\_temp - x1);

double x\_new = -(c1 \* b2 - c2 \* b1) / (a1 \* b2 - a2 \* b1);

double y\_new = -(a1 \* c2 - a2 \* c1) / (a1 \* b2 - a2 \* b1);

down\_buffer\_kub.Add(new Osxy(x\_temp, y\_temp));

down\_buffer\_kub.Add(new Osxy(x\_new, y\_new));

temp\_b\_fl = false;

}

}

}

private void button3\_Click(object sender, EventArgs e)

{

OpenGL gl = this.openGLControl1.OpenGL;

gl.Clear(OpenGL.GL\_COLOR\_BUFFER\_BIT | OpenGL.GL\_DEPTH\_BUFFER\_BIT);

gl.LoadIdentity();

gl.Translate(0.0f, 0.0f, -20.0f);

gl.Begin(OpenGL.GL\_LINE\_LOOP);

gl.Color(1f, 1f, 1f);

foreach (Osxy figures in osxy\_rhombus)

{

gl.Vertex(figures.x, figures.y);

}

gl.End();

gl.Begin(OpenGL.GL\_LINE\_LOOP);

gl.Color(1f, 1f, 1f);

foreach (Osxy figures in kub)

{

gl.Vertex(figures.x, figures.y);

}

gl.End();

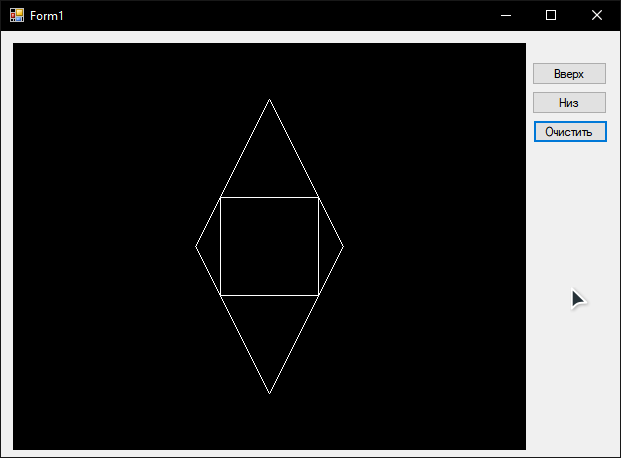
down\_buffer\_kub.Clear();

down\_buffer.Clear();

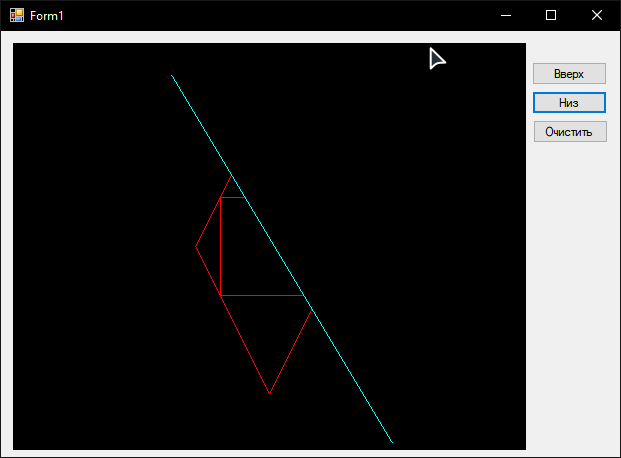
}

}

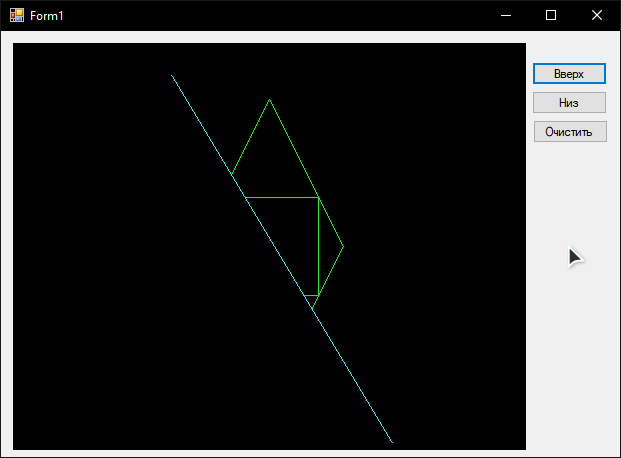
}



Скриншот 1 : запуск программы.



Скриншот 2 : отсечение верха фигуры



Скриншот 3: отсечение низа фигуры

**Вывод:** мы научились выполнять рассечение заданной геометрической фигуры на части прямой линией, задаваемой пользователем в виде отрезка