Министерство образования и науки Российской Федерации Федеральное агентство по образованию Федеральное государственное бюджетное образовательное учреждение высшего образования «Вятский государственный университет»

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Лабораторная работа №5

по курсу «Компьютерная графика»

# «Реализация двумерного алгоритма отсечения отрезка Сазерленда-Коэна»

Выполнил студент группы ИВТ-21\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Птахова А.М/

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

Киров 2021

Цель работы: закрепить лекционный материал по изучению алгоритмов отсечения невидимых областей изображения. Реализовать двумерный алгоритм Сазерленда - Коэна для отсечения отрезка окном прямоугольной формы.

Краткие теоретические сведения

Вся плоскость вывода разбивается прямыми, образующими прямоугольник на девять подплоскостей; каждой из них присваивается четырехбитовый код, в котором единица

в нулевом бите означает, что конец отрезка лежит левее окна,

в первом бите - выше окна,

во втором бите - правее окна,

в третьем бите - ниже окна.

Описание алгоритма Сазерленда - Коэна:

Дано: прямоугольное окно с координатами (xl,ya),(xr,yb) верхнего левого и нижнего правого углов прямоугольника; отрезок с координатами концов (x1,y1) и (x2,y2).

Найти: новые координаты отрезка (x1,y1) и нарисовать его целиком или частично в случае попадания в область окна и не рисовать в противном случае.

1. Присвоить коды (р1 и р2) концам отрезка.
2. Если отрезок полностью невидим(p1 and p2 <> 0), идти к 9, иначе идти к 3.
3. Если отрезок видим полностью (p1 = 0 and p2 = 0) - нарисовать его (провести линию от точки с координатами (х1,y1) до точки - (x2,y2) и идти к 9, иначе идти к 4.
4. Если р1=0 - поменять местами точки (x1,y1) и (x2,y2),чтобы обработка шла каждый раз с точки, находящейся вне окна: х=х1,х1=х2,х2=х,y=y1,y1=y2,y2=y,p1=p2 и идти к 5, иначе идти к 5.
5. Если точка (x1,y1) - слева от окна - найти ее новые координаты по формулам: y1=y1+(y2-y1)\*(xl-х1)/(х2-х1), x1=xl и идти к 1, иначе идти к 6.
6. Если точка (x1,y1) - выше окна - найти ее новые координаты по формулам: х1=х1+(х2-х1)\*(ya-y1)/(y2-y1), y1=ya и идти к 1, иначе идти к 7.
7. Если точка (x1,y1)- справа от окна - найти ее новые координаты по формулам: y1=y1+(y2-y1)\*(xr-х1)/(х2-х1), х1=xr и идти к 1, иначе идти к 8.
8. Если точка (x1,y1) - ниже окна - найти ее новые координаты по формулам: х1=х1+(х2-х1)\*(yb-y1)/(y2-y1),y1=yb и идти к 1, иначе идти к 1.
9. Закончить.

**Разработка программы**

Пользователь может увидеть отсечение заданной линии, нажав на кнопку Button1.

Схемы алгоритмов приведены в приложении А, листинг программ приведет в приложении Б, экранные формы – в приложении В.

**Вывод:**

В ходе выполнения лабораторной работы были получены знания об алгоритме отсечения отрезка Сазерленда-Коэна. В результате была написана программа, реализующая изученный метод.

Приложение А

Схемы алгоритмов

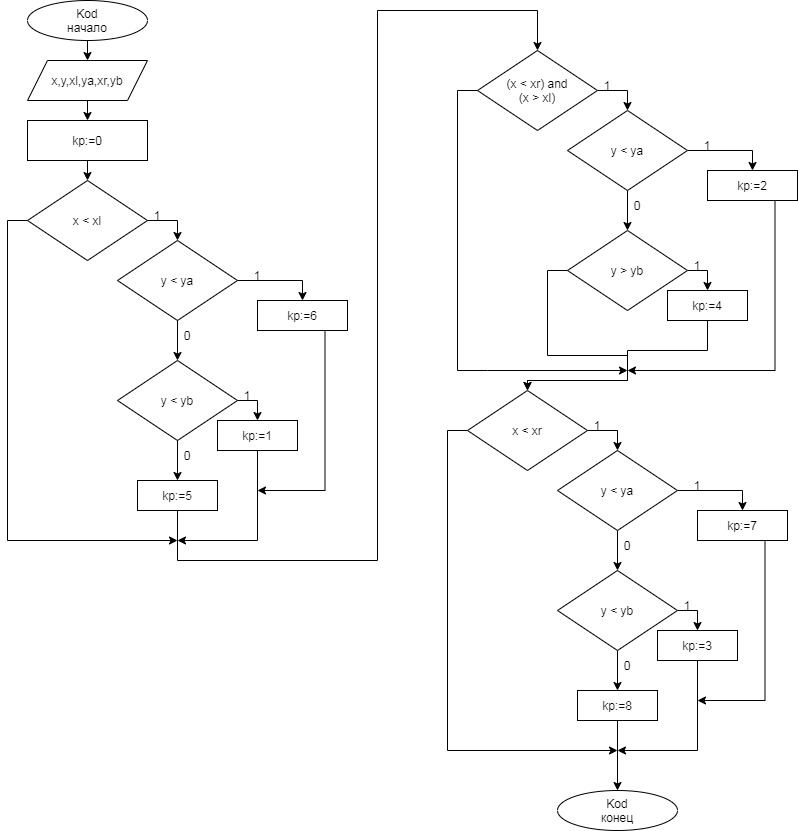


Рисунок 1 – схема алгоритма процедуры, реализующей присвоение кода концу отрезка

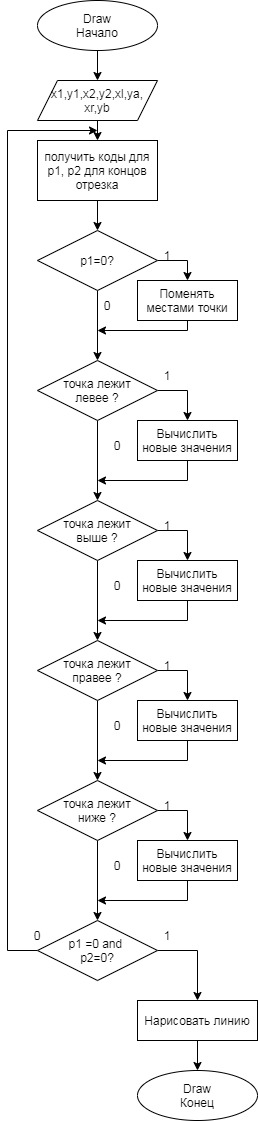


Рисунок 2 – схема алгоритма процедуры, реализующей отсечение отрезка

Приложение Б

Листинг программы

**function** kod(x,y,xl,ya,xr,yb:longint):byte;

**var** kp:byte;

**begin**

kp:=0;

**if** x < xl **then**

**if** y < ya **then** kp:=6

**else if** y < yb **then** kp:=1

**else** kp:=5;

**if** (x < xr) **and** (x > xl) **then**

**if** y < ya **then** kp:=2

**else if** y > yb **then** kp:=4;

**if** (x > xr) **then**

**if** y < ya **then** kp:=7

**else if** y < yb **then** kp:=3

**else** kp:=8;

kod:=kp;

**end**;

**procedure** Draw(x1,x2,y1,y2,xl,ya,xr,yb:longint);

**var** p1,p2,x,y:byte;

**begin**

**repeat**

p1:=kod(x1,y1,xl,ya,xr,yb);

p2:=kod(x2,y2,xl,ya,xr,yb);

**if** p1=0 **then begin**

x:=x1;

x1:=x2;

x2:=x;

y:=y1;

y1:=y2;

y2:=y;

p1:=p2;

**end**;

**if** (p1=6) **or** (p1=1) **or** (p1=5) **then begin**

y1:=round(y1+(y2-y1)\*(xl-x1)/(x2-x1));

x1:=xl;

**end**;

**if** (p1=2) **then begin**

x1:=round(x1+(x2-x1)\*(ya-y1)/(y2-y1));

y1:=ya;

**end**;

**if** (p1=7) **or** (p1=3) **or** (p1=8) **then begin**

y1:=round(y1+(y2-y1)\*(xr-x1)/(x2-x1));

x1:=xr;

**end**;

**if** (p1=4) **then begin**

x1:=round(x1+(x2-x1)\*(yb-y1)/(y2-y1));

y1:=yb;

**end**;

**until** (p1=0) **and** (p2=0);

Form1.PaintBox1.Canvas.Line(x1,y1,x2,y2);

//Form1.paintBox1.Canvas.Rectangle(100,100,200,200);

**end**;

Приложение В

Экранные формы

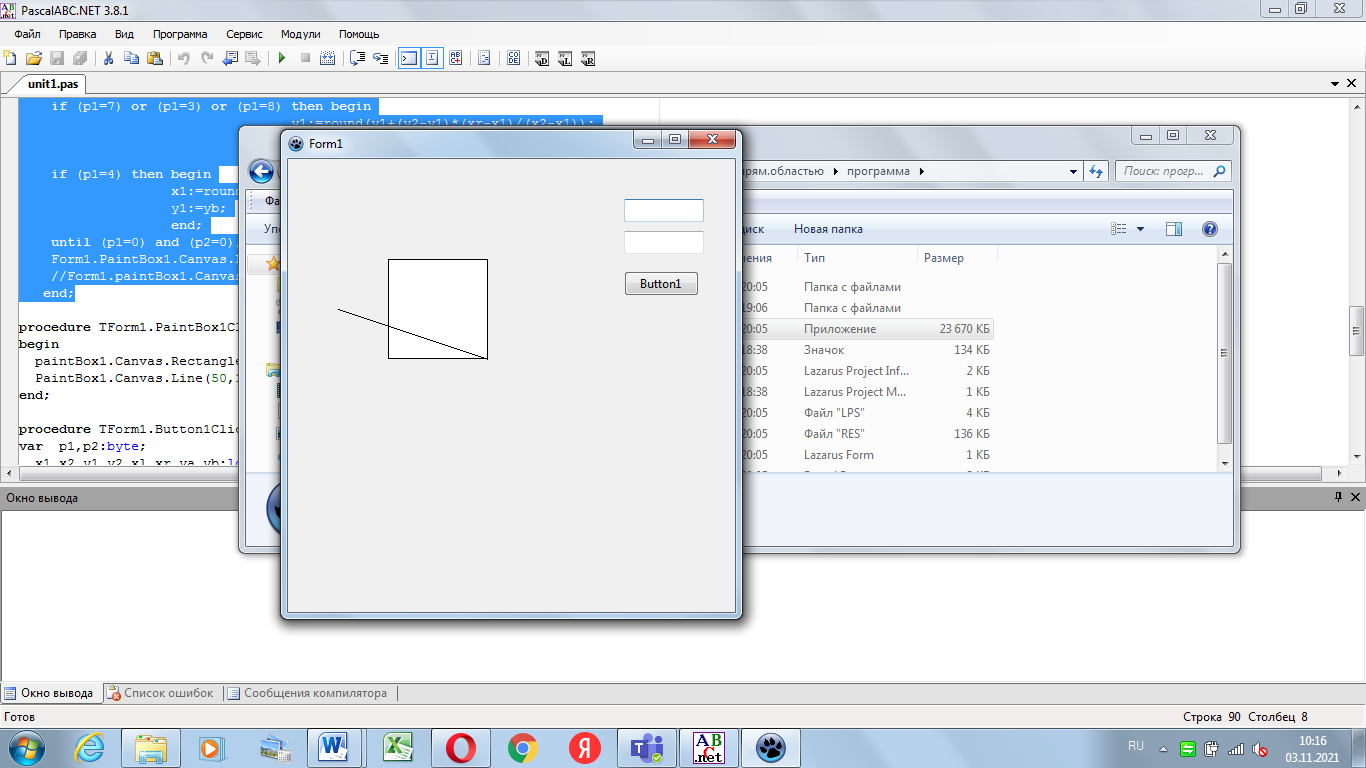


Рисунок 3 – первоначальный вид программы

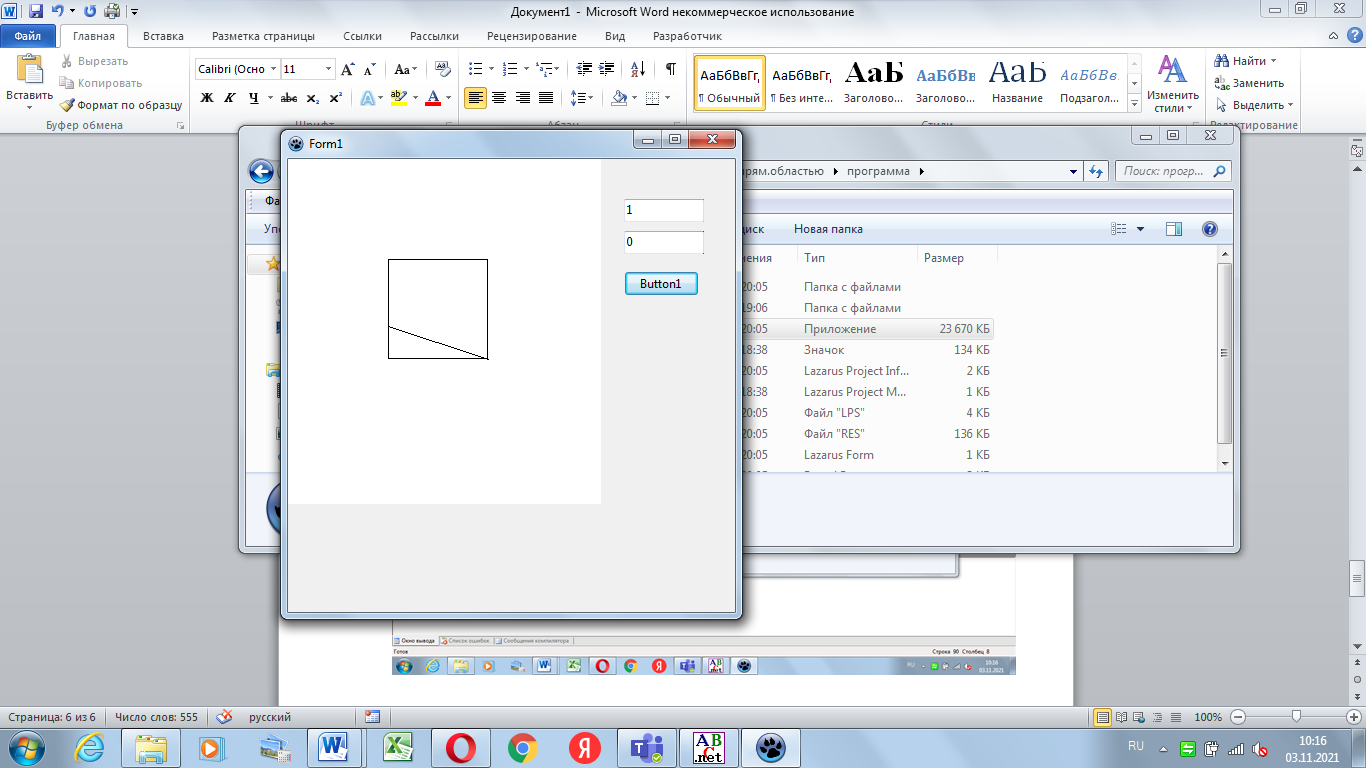


Рисунок 4 – отсечение отрезка