МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«**Вятский государственный университет**»

**(ФГБОУ ВО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра ЭВМ

Отчёт

### Лабораторная работа № 7 по дисциплине

«Программирование»

«Исследование фракталов»

Выполнил студент группы ИВТб-1301 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Жеребцов К. А.

## Проверил преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Чистяков Г.А.

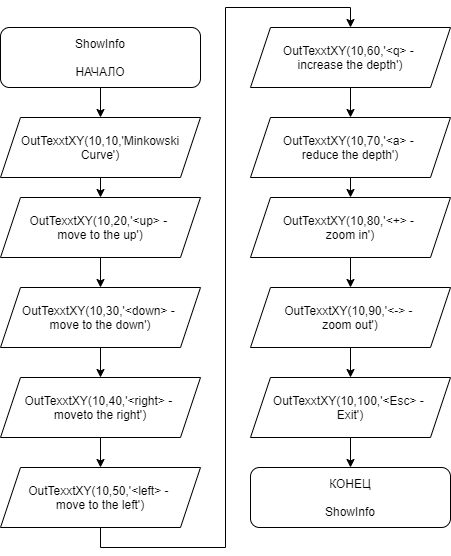
Киров 2021

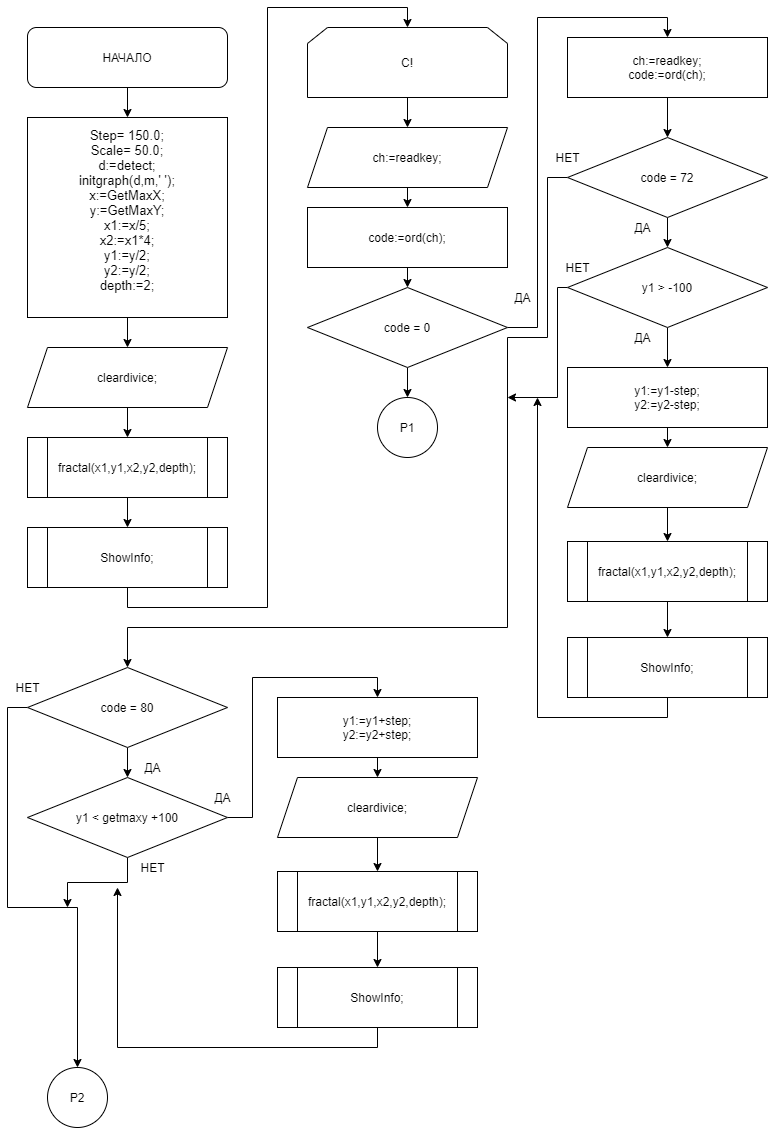
***Цель работы:*** получение навыков реализации алгоритмов с рекурсивными вычислениями, знакомство с фракталами.

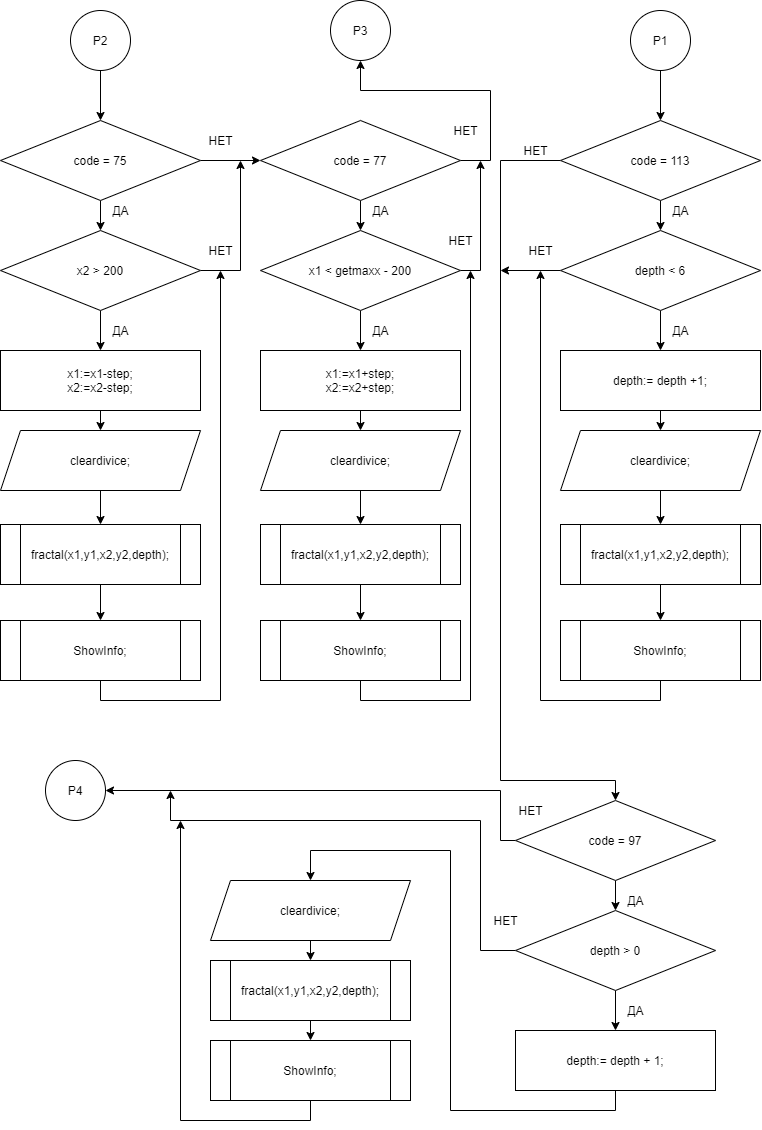
***Задания:***

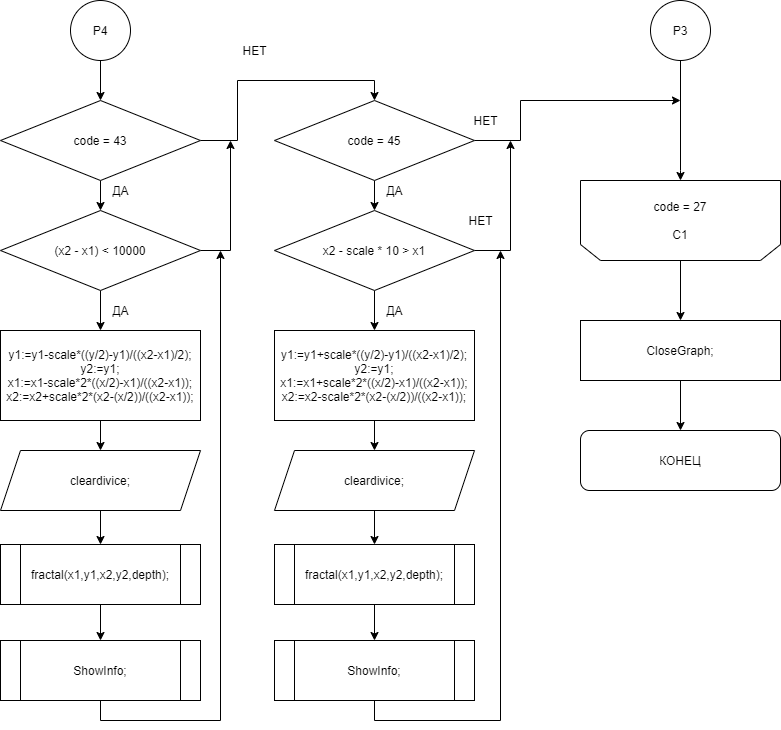
1. Написать программу для визуализации фрактала «Кривая Минковского».
2. Предусмотреть возможность масштабирования, Изменения глубины прорисовки и перемещения полученной фигуры.
3. Построение множества ломанных, образующих фрактал, должно осуществляться в отдельном модуле.

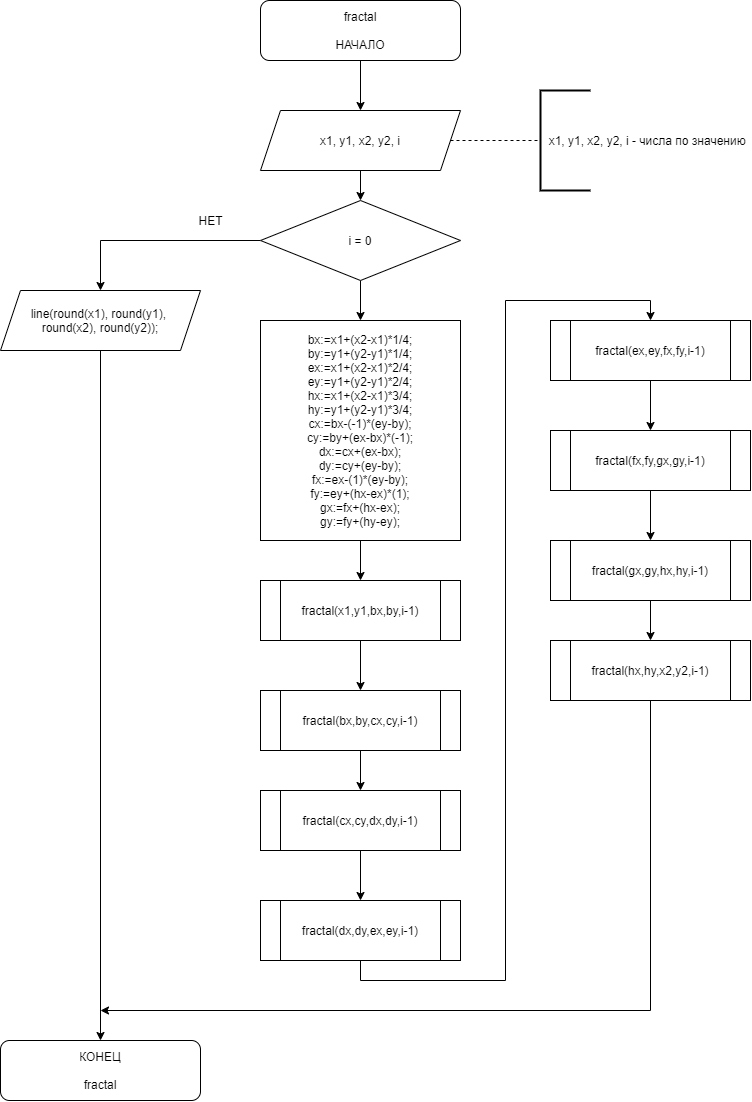
***Схемы алгоритмов:***



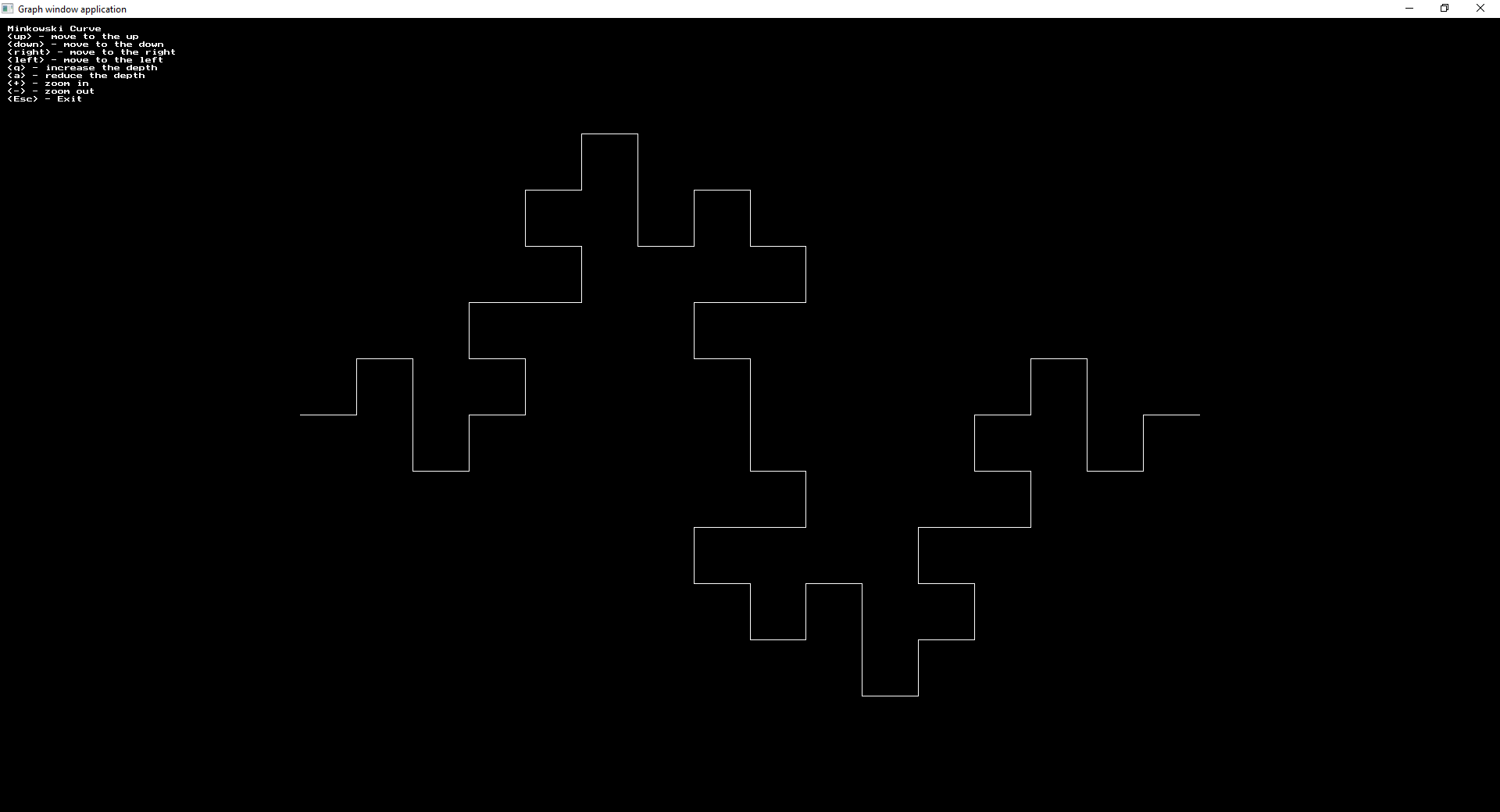


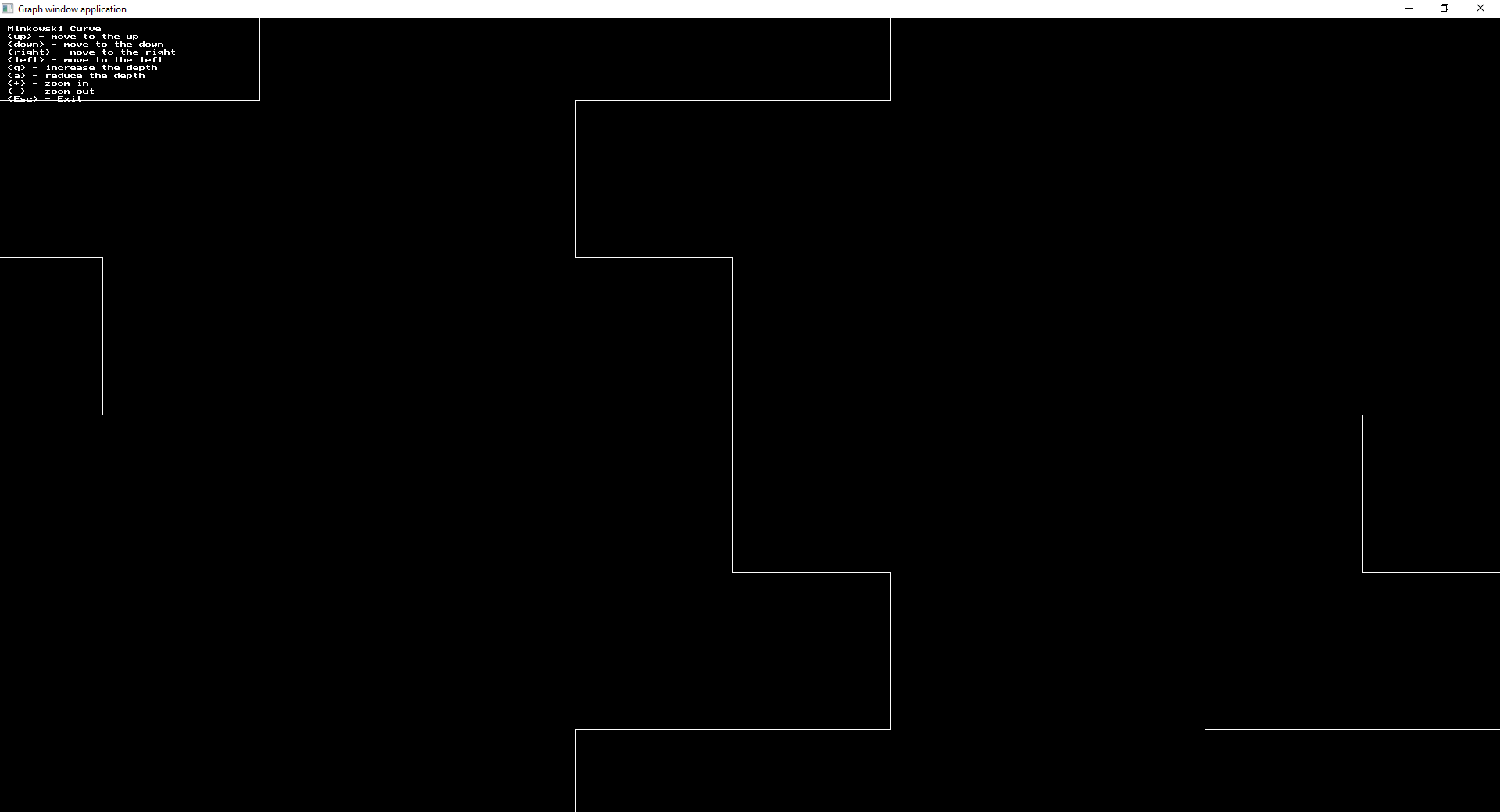


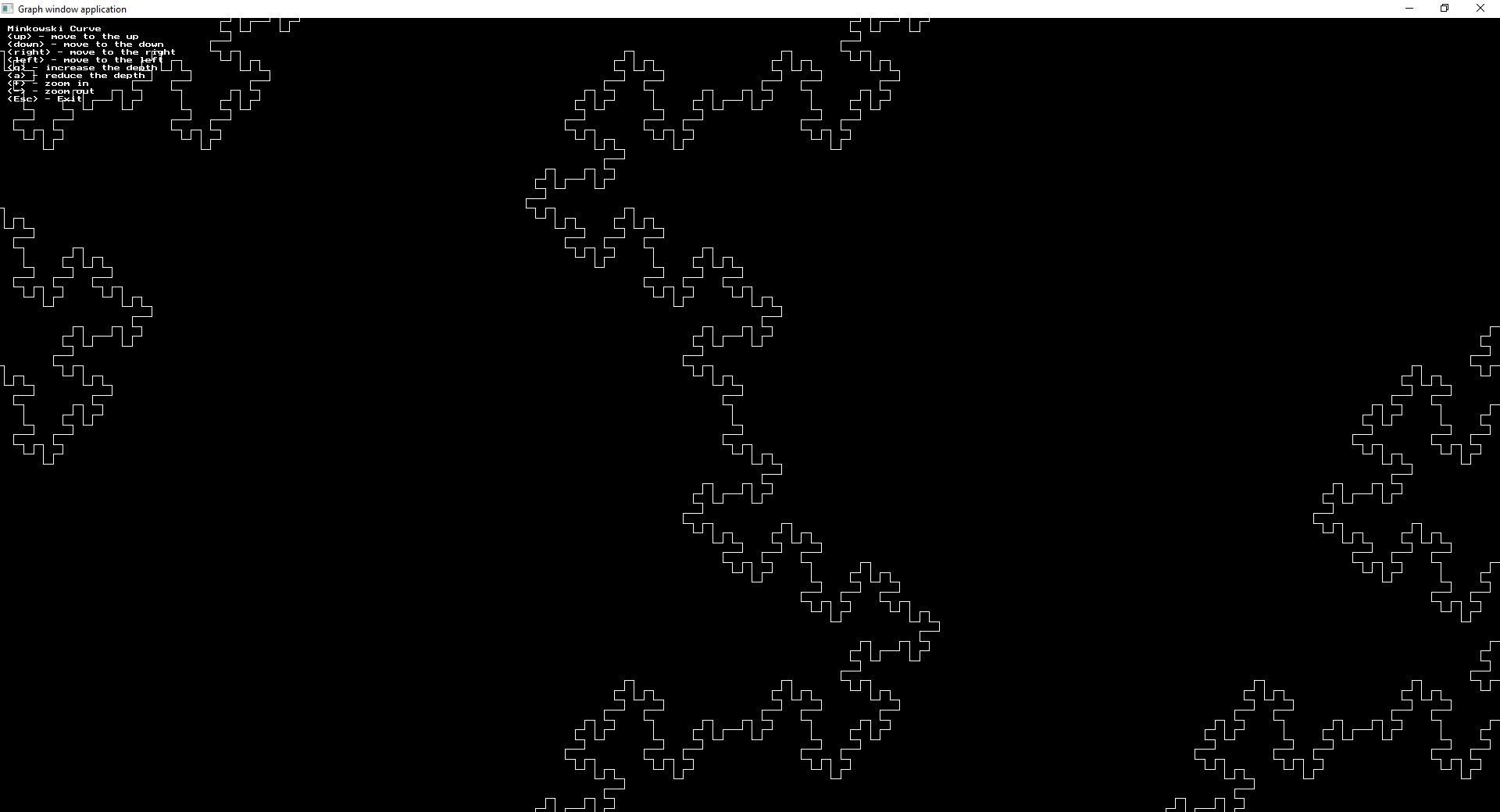


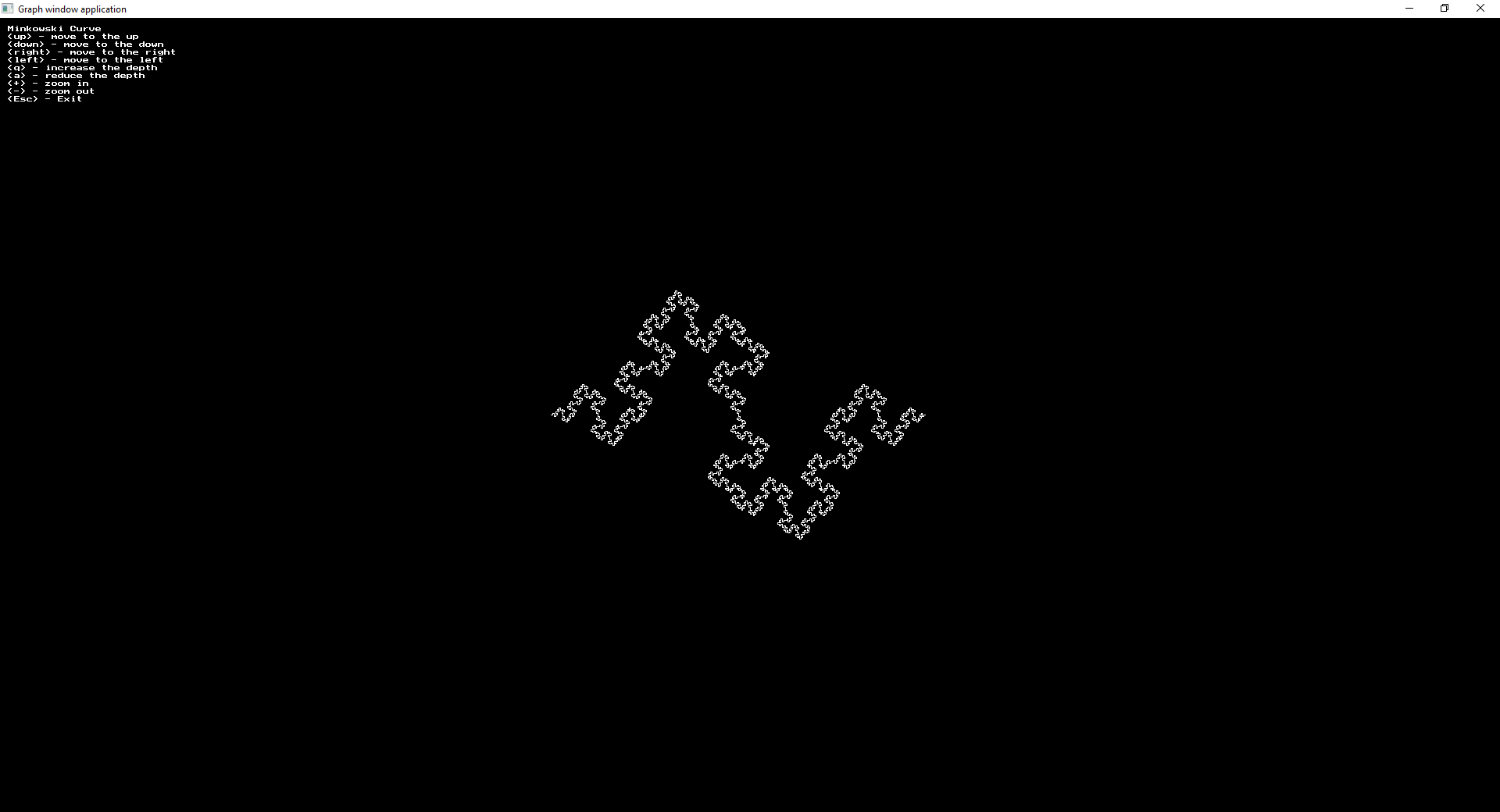


***Экранные формы:***









***Листинг:***

*Основная программа:*

program lab7;

uses

graph,math,wincrt,fractal;

const

step=150.0; // шаг при перемещении фрактала

scale=50.0; // изменение масштаба

var

d,m:integer;

x,y:real;

x1,x2,y1,y2:real;

ch:char;

code:integer;

depth:integer;

procedure ShowInfo(); // процедура, выводящая информацию об управлении

begin

OutTextXY(10,10,'Minkowski Curve');

OutTextXY(10,20,'<up> - move to the up');

OutTextXY(10,30,'<down> - move to the down');

OutTextXY(10,40,'<right> - move to the right');

OutTextXY(10,50,'<left> - move to the left');

OutTextXY(10,60,'<q> - increase the depth');

OutTextXY(10,70,'<a> - reduce the depth');

OutTextXY(10,80,'<+> - zoom in');

OutTextXY(10,90,'<-> - zoom out');

OutTextXY(10,100,'<Esc> - Exit');

end;

begin

d:=detect;

initgraph(d,m,'');

x:=getMaxX;

y:=getMaxY;

x1:=x/5; // определение начальных координат

x2:=x1\*4;

y1:=y/2;

y2:=y/2;

depth:=2; // определение начальной глубины

cleardevice;

fractal(x1, y1, x2 ,y2,depth); // вызов процедуры отрисовки фрактала

ShowInfo();

repeat // цикл обработки клавиш управления

ch:=readkey();

code:=ord(ch);

if code = 0 then

begin

ch:=readkey();

code:=ord(ch);

case code of

72 : // сдвиг вверх

begin

if (y1 > -100) then

begin

y1:=y1-step;

y2:=y2-step;

cleardevice;

fractal(x1, y1, x2 ,y2,depth);

ShowInfo();

end;

end;

80 : // сдвиг вниз

begin

if y1 < getmaxy()+100 then

begin

y1:=y1+step;

y2:=y2+step;

cleardevice;

fractal(x1, y1, x2 ,y2,depth);

ShowInfo();

end;

end;

75 : // сдвиг влево

if x2 > 200 then

begin

x1:=x1-step;

x2:=x2-step;

cleardevice;

fractal(x1, y1, x2 ,y2,depth);

ShowInfo();

end;

77 : // сдвиг вправо

begin

if x1 < getmaxx()-200 then

begin

x1:=x1+step;

x2:=x2+step;

cleardevice;

fractal(x1, y1, x2 ,y2,depth);

ShowInfo();

end;

end;

end;

end

else

case code of

113: // увеличение глубины прорисовки

begin

if depth<6 then

begin

depth:=depth+1;

cleardevice;

fractal(x1, y1, x2 ,y2,depth);

ShowInfo();

end;

end;

97: //уменьшение глубины прорисовки

begin

if depth>0 then

begin

depth:=depth-1;

cleardevice;

fractal(x1, y1, x2 ,y2,depth);

ShowInfo();

end;

end;

43: // увеличение

begin

if(x2-x1)<10000 then

begin

y1:=y1-scale\*((y/2)-y1)/((x2-x1)/2);

y2:=y1;

x1:=x1-scale\*2\*((x/2)-x1)/((x2-x1));

x2:=x2+scale\*2\*(x2-(x/2))/((x2-x1));

cleardevice;

fractal(x1, y1, x2 ,y2,depth);

ShowInfo();

end;

end;

45: // уменьшение

begin

if x2-scale\*10>x1 then

begin

y1:=y1+scale\*((y/2)-y1)/((x2-x1)/2);

y2:=y1;

x1:=x1+scale\*2\*((x/2)-x1)/((x2-x1));

x2:=x2-scale\*2\*(x2-(x/2))/((x2-x1));

cleardevice;

fractal(x1, y1, x2 ,y2,depth);

ShowInfo();

end;

end;

until code = 27;

CloseGraph;

end.

*Модуль:*

unit fractal;

interface

uses graph;

procedure fractal(x1,y1,x2,y2:real;i:integer);

implementation

procedure fractal(x1,y1,x2,y2:real;i:integer);

var

bx,by:real;

cx,cy:real;

dx,dy:real;

ex,ey:real;

fx,fy:real;

gx,gy:real;

hx,hy:real;

begin

if i=0 then // минимальная глубина, при которой выведется фрактал

begin

line(round(x1),round(y1),round(x2),round(y2));

end

else

begin

bx:=x1+(x2-x1)\*1/4; // вычисление координат при изменении глубины

by:=y1+(y2-y1)\*1/4;

ex:=x1+(x2-x1)\*2/4;

ey:=y1+(y2-y1)\*2/4;

hx:=x1+(x2-x1)\*3/4;

hy:=y1+(y2-y1)\*3/4;

cx:=bx-(-1)\*(ey-by);

cy:=by+(ex-bx)\*(-1);

dx:=cx+(ex-bx);

dy:=cy+(ey-by);

fx:=ex-(1)\*(ey-by);

fy:=ey+(hx-ex)\*(1);

gx:=fx+(hx-ex);

gy:=fy+(hy-ey);

fractal(x1,y1,bx,by,i-1); // рекурсивный вызов процедуры для изображения

fractal(bx,by,cx,cy,i-1); // кривой

fractal(cx,cy,dx,dy,i-1);

fractal(dx,dy,ex,ey,i-1);

fractal(ex,ey,fx,fy,i-1);

fractal(fx,fy,gx,gy,i-1);

fractal(gx,gy,hx,hy,i-1);

fractal(hx,hy,x2,y2,i-1);

end;

end;

end.

***Вывод:***

Были получены навыки реализации алгоритмов с рекурсивными вычислениями. Была написана программа для визуализации фрактала «Кривая Минковского», предусмотрена возможность масштабирования, изменения глубины прорисовки и перемещения полученной фигуры.

Построение множества ломанных, образующих фрактал осуществляется в отдельном модуле.

Модуль имеет два основных радела: Interface и Implementation. В первом разделе описываются имена типов, процедур и функций, которые могут заимствоваться при подключении данного модуля. Во втором – описание тех процедур и функций. После подключения модуля в основной программе можно использовать подпрограммы, описанные в модуле, без их описания в основном коде программы.