Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №7**

**«МДК 05.02 Разработка кода информационных систем»**

Выполнил: студент учебной группы

ИСПк-203-52-00

Кононов Степан Андреевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

Цель работы

Получить навыки реализации алгоритмов с рекурсивными вычислениями, знакомство с фракталами.

Формулировка задания

Вариант 12.

1. Написать программу для визуализации фрактала “Кривая миновского”.

2. Предусмотреть возможности масштабирования, изменения глубины прорисовки и перемещения полученной фигуры.

3. Построение множества ломанных, образующих фрактал, должно осуществляться в отдельном модуле.

Описание алгоритма

Данный алгоритм создаёт фрактал “Кривая миновского” и позволяет взаимодействовать с ним: двигать фигуру, увеличивать/уменьшать масштаб, увеличивать/уменьшать глубину прорисовки.

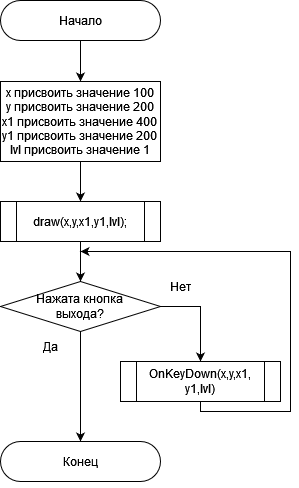


Рисунок 1 – Схема главного алгоритма

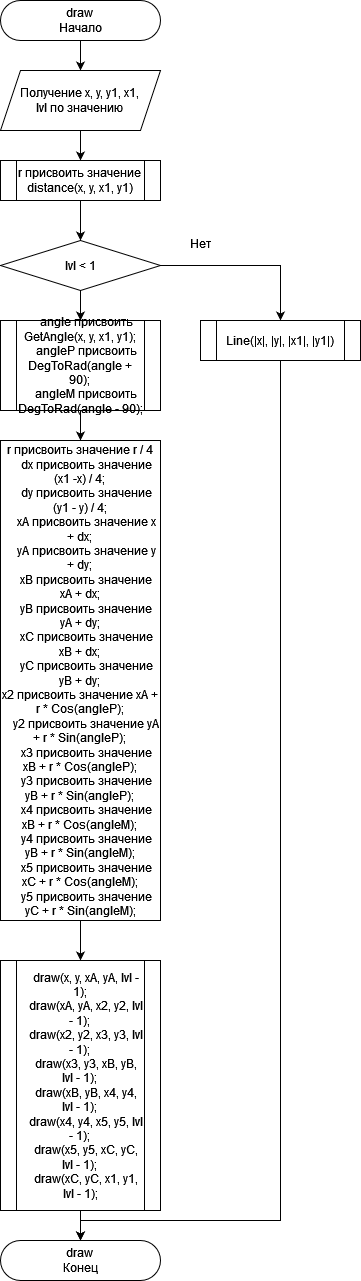


Рисунок 2 – Схема алгоритма процедуры Draw

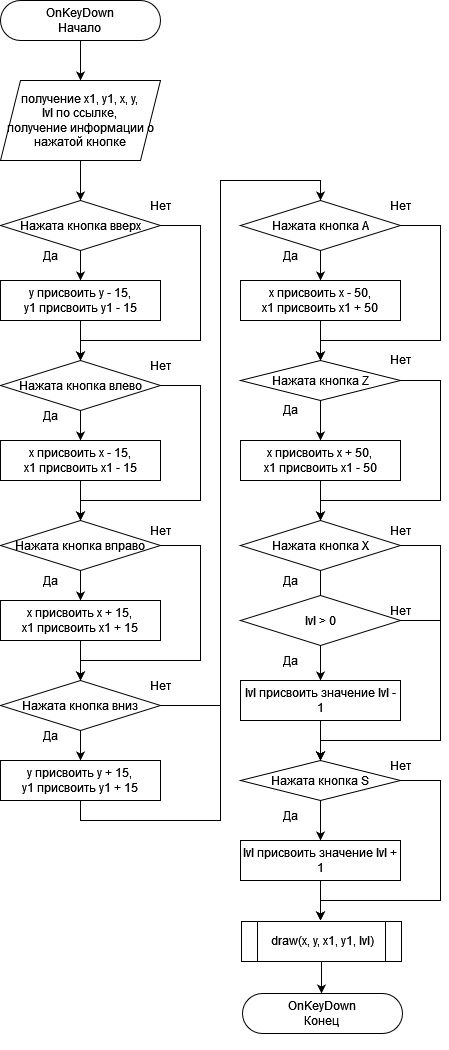


Рисунок 3 – Схема алгоритма процедуры KeyDown

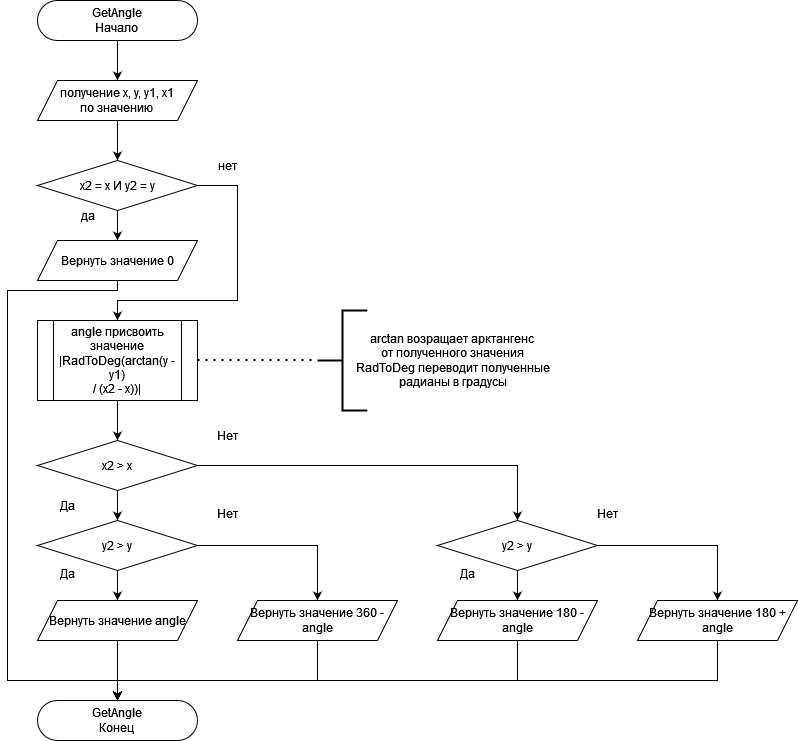


Рисунок 4 – Схема алгоритма процедуры GetAngle

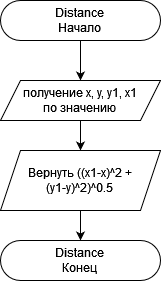


Рисунок 5 – Схема алгоритма процедуры Distance

Код программы

**uses** GraphABC, math;

**var** x, y, x1, y1, lvl: integer;

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

**begin**

TextOut(0,0,'X - уменьшить глубину прорисовки');

TextOut(0,15,'S - увеличить глубину прорисовки');

TextOut(0,30,'Z - уменьшить масштаб');

TextOut(0,45,'A - увеличить масштаб');

TextOut(0,60,'Стрелки - двигать фигуру');

**case** key **of**

VK\_Up: **begin** y := y - 15; y1 := y1 - 15 **end**;

VK\_Down: **begin** y += 15; y1 += 15 **end**;

VK\_Left: **begin** x := x - 15; x1 := x1 - 15 **end**;

VK\_Right: **begin** x := x + 15; x1 := x1 + 15 **end**;

VK\_A: **begin** x := x - 50; x1 +=50; **end**;

VK\_Z: **begin** x := x + 50; x1 -=50; **end**;

vk\_x: **if** lvl > 0 **then** lvl -= 1;

vk\_s: lvl += 1;

**end**;

Window.Clear;

draw(x, y, x1, y1, lvl);

TextOut(0,0,'X - уменьшить глубину прорисовки');

TextOut(0,15,'S - увеличить глубину прорисовки');

TextOut(0,30,'Z - уменьшить масштаб');

TextOut(0,45,'A - увеличить масштаб');

TextOut(0,60,'Стрелки - двигать фигуру');

redraw;

**end**;

**begin**

LockDrawing;

x := 100;

y := 200;

x1 := 400;

y1 := 200;

lvl:= 1;

draw(x, y, x1, y1, lvl);

redraw;

onKeyDown := keydown;

**end**.

Дополнительный блок:

**Unit** math;

**uses** GraphABC;

**procedure** RLine(x, y, x1, y1: real) := Line(Round(x), Round(y), Round(x1), Round(y1));

**function** GetAngle(x, y, x2, y2: real): real;

**begin**

**if** (x2 = x) **and** (y2 = y) **then**

Result := 0

**else begin**

**var** angle := Abs(RadToDeg(ArcTan((y2 - y) / (x2 - x))));

**if** x2 > x **then**

**if** y2 > y **then** Result := angle **else** Result := 360 - angle

**else**

**if** y2 > y **then** Result := 180 - angle **else** Result := 180 + angle;

**end**;

**end**;

**function** Distance(x, y, x1, y1: real) := Sqrt(Sqr(x1 - x) + Sqr(y1 - y));

**procedure** Draw(x, y, x1, y1, lvl: real);

**begin**

**var** r := Distance(x, y, x1, y1);

**if** lvl < 1 **then**

RLine(x, y, x1, y1)

**else**

**begin**

**var** angle := GetAngle(x, y, x1, y1);

**var** angleP := DegToRad(angle + 90);

**var** angleM := DegToRad(angle - 90);

r /= 4;

**var** dx := (x1 - x) / 4;

**var** dy := (y1 - y) / 4;

**var** xA := x + dx;

**var** yA := y + dy;

**var** xB := xA + dx;

**var** yB := yA + dy;

**var** xC := xB + dx;

**var** yC := yB + dy;

**var** x2 := xA + r \* Cos(angleP);

**var** y2 := yA + r \* Sin(angleP);

**var** x3 := xB + r \* Cos(angleP);

**var** y3 := yB + r \* Sin(angleP);

**var** x4 := xB + r \* Cos(angleM);

**var** y4 := yB + r \* Sin(angleM);

**var** x5 := xC + r \* Cos(angleM);

**var** y5 := yC + r \* Sin(angleM);

Draw(x, y, xA, yA, lvl - 1);

Draw(xA, yA, x2, y2, lvl - 1);

Draw(x2, y2, x3, y3, lvl - 1);

Draw(x3, y3, xB, yB, lvl - 1);

Draw(xB, yB, x4, y4, lvl - 1);

Draw(x4, y4, x5, y5, lvl - 1);

Draw(x5, y5, xC, yC, lvl - 1);

Draw(xC, yC, x1, y1, lvl - 1);

**end**;

**end**;

**begin**

**end**.

Результат выполнения программы

На рисунке 6 представлен результат выполнения программы.

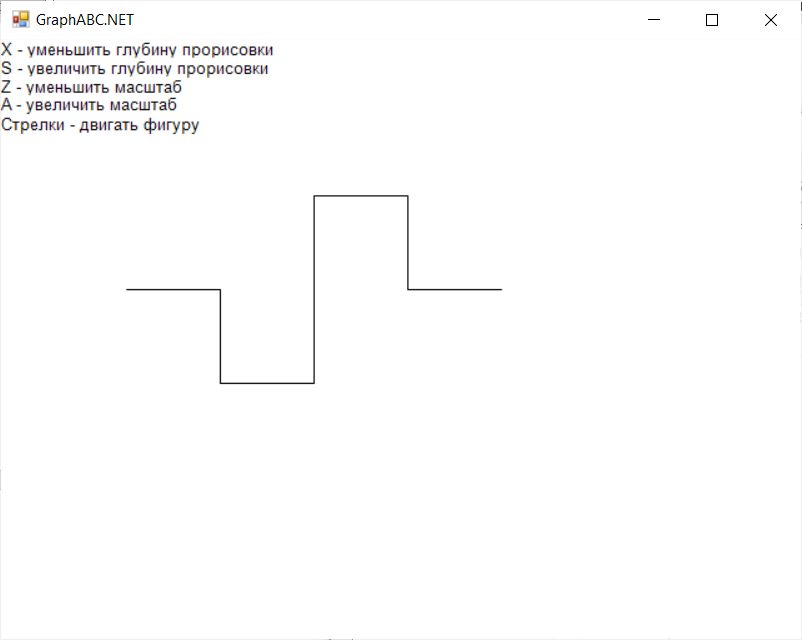


Рисунок 6 - Результат выполнения программы

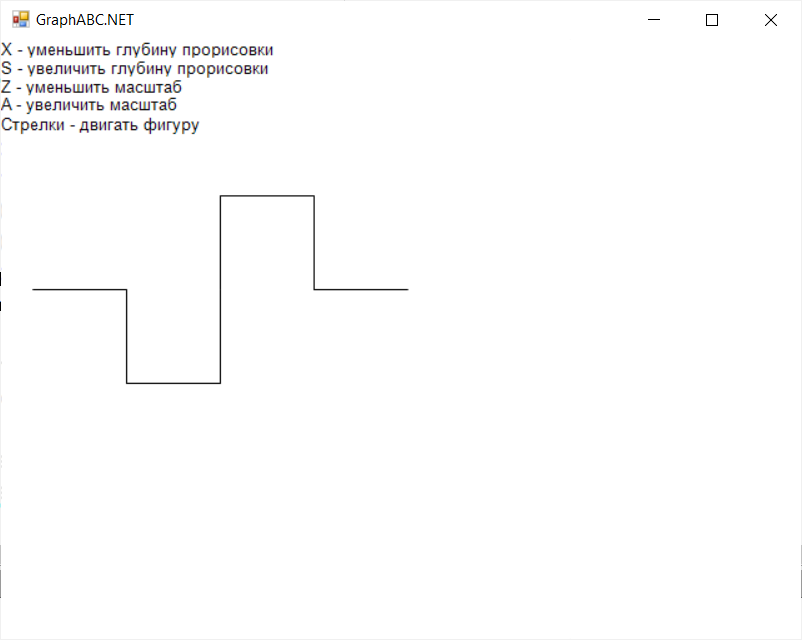


Рисунок 7 - Результат выполнения программы



Рисунок 8 - Результат выполнения программы

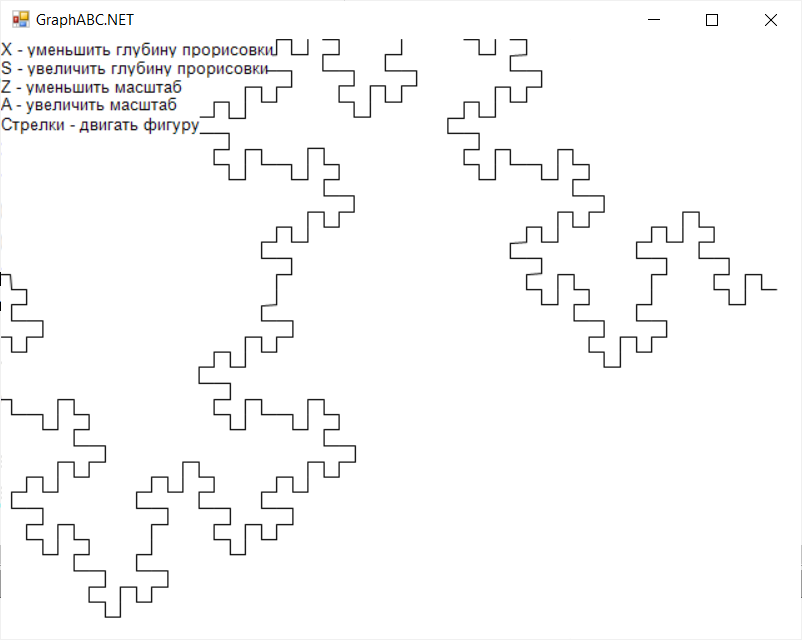


Рисунок 9 - Результат выполнения программы

Вывод

В процессе выполнения этой работы мы успешно применили знания, полученные на лабораторных занятиях, в практических задачах. Мы использовали процедуры, условные операторы, циклы while, repeat и for, создавали переменные и использовали операторы ввода-вывода. Кроме того, мы построили алгоритм с помощью онлайн-сервиса app.diagrams.net.

В результате выполнения работы мы освоили основные методы сортировки данных, изучили работу с текстовыми файлами и успешно выполнили практическое задание, применяя полученные знания.

Таким образом, выполнение контрольной работы №7 принесло нам ценный опыт и знания, которые будут полезны в дальнейшем образовании. Мы изучили применение базовых конструкций языка программирования и разработку алгоритмов для решения задач. Этот опыт будет полезен при решении новых задач и работе над реальными проектами в будущем.