МИНИСТЕРСТВО НАУКИ И ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное образовательное учреждение

Высшего профессионального образования

**«Вятский государственный университет»**

**(ФГБОУ ВПО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Исследование фракталов.

Отчет по лабораторной работе №7 дисциплины

«Программирование»

Выполнил студент группы ИВТб-11\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Бурдукова А.А./

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

Киров 2019

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

**Задание:**

**1. Написать программу для визуализации фрактала «Кривая Госпера».**

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

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

**Схема алгоритмов:**

**Изображение выглядит как текст, карта

Автоматически созданное описание**

**Изображение выглядит как текст, карта

Автоматически созданное описание**

**Исходный текст программы:**

**unit peano;**

**interface**

**uses WINCRT, GRAPH;**

**procedure Draw(x, y, l, u : real; t, q : integer);**

**implementation**

**Procedure Draw2(Var x, y: Real; l, u : Real; t, q : Integer);**

**Begin**

**Draw(x, y, l, u, t, q);**

**x := x + l\*cos(u);**

**y := y - l\*sin(u)**

**End;**

**procedure Draw(x, y, l, u : real; t, q : integer);**

**Begin**

**If t > 0 Then**

**Begin**

**If q = 1 Then**

**Begin**

**x := x + l\*cos(u);**

**y := y - l\*sin(u);**

**u := u + pi**

**End;**

**u := u - 2\*pi/19;**

**l := l/sqrt(7);**

**Draw2(x, y, l, u, t-1, 0);**

**Draw2(x, y, l, u+pi/3, t-1, 1);**

**Draw2(x, y, l, u+pi, t-1, 1);**

**Draw2(x, y, l, u+2\*pi/3, t-1, 0);**

**Draw2(x, y, l, u, t-1, 0);**

**Draw2(x, y, l, u, t-1, 0);**

**Draw2(x, y, l, u-pi/3, t-1, 1)**

**End**

**Else**

**Line(Round(x), Round(y), Round(x + cos(u)\*l), Round(y -sin(u)\*l))**

**End;**

**end.**

**uses peano, WINCRT, GRAPH;**

**type**

**num = integer;**

**var**

**Gdrr, Gmrr: num;**

**x, y, depth, MAX\_X, MAX\_Y : integer;**

**k : real;**

**key: char;**

**begin**

**Gdrr:=Detect;**

**Gmrr:=GetMaxMode;**

**InitGraph(Gdrr, Gmrr, '');**

**SetBkColor (15);**

**SetColor (1);**

**SetFillStyle (1,1);**

**ClearViewPort();**

**MAX\_X := GetMaxX; MAX\_Y := GetMaxY;**

**x := 500; y := 400; depth := 3; k := 1.0;**

**Draw(x, y, 400, 0, depth, 0);**

**OutTextXY (10,10, 'Arrows: moving');**

**OutTextXY (10,20, 'Pg Up, Pg Dn: scaling');**

**OutTextXY (10,30, 'Home, End: recursion depth');**

**repeat**

**key:=WINcrt.ReadKey();**

**if key=#0 then**

**begin**

**key:=WINcrt.ReadKey();**

**case key of**

**#72: //up**

**begin**

**if y >= 20 then**

**begin**

**ClearViewPort;**

**y:=y-20;**

**Draw(x, y, 400 \* k, 0, depth, 0);**

**end;**

**end;**

**#80: //down**

**begin**

**if y <= MAX\_Y - 20 then**

**begin**

**ClearViewPort;**

**y:=y+20;**

**Draw(x, y, 400 \* k, 0, depth, 0);**

**end;**

**end;**

**#75: //left**

**begin**

**if x >= 20 then**

**begin**

**ClearViewPort;**

**x:=x-20;**

**Draw(x, y, 400 \* k, 0, depth, 0);**

**end;**

**end;**

**#77: //up**

**begin**

**if x <= MAX\_X - 20 then**

**begin**

**ClearViewPort;**

**x:=x+20;**

**Draw(x, y, 400 \* k, 0, depth, 0);**

**end;**

**end;**

**#71: //home**

**begin**

**if depth <= 5 then**

**begin**

**ClearViewPort;**

**depth:=depth+1;**

**Draw(x, y, 400 \* k, 0, depth, 0);**

**end;**

**end;**

**#79: //end**

**begin**

**if depth >= 1 then**

**begin**

**ClearViewPort;**

**depth:=depth-1;**

**Draw(x, y, 400 \* k, 0, depth, 0);**

**end;**

**end;**

**#73: //pg up**

**begin**

**if k < 2 then**

**begin**

**ClearViewPort;**

**k := k \* 1.1;**

**Draw(x, y, 400 \* k, 0, depth, 0);**

**end;**

**end;**

**#81: //pg down**

**begin**

**if k > 0.5 then**

**begin**

**ClearViewPort;**

**k := k / 1.1;**

**Draw(x, y, 400\*k, 0, depth, 0);**

**end;**

**end;**

**end;**

**OutTextXY (10,10, 'Arrows: moving');**

**OutTextXY (10,20, 'Pg Up, Pg Dn: scaling');**

**OutTextXY (10,30, 'Home, End: recursion depth');**

**end;**

**until key=#27;**

**end.**

**Скриншоты выполнения программы**

**Вывод**

В ходе данной лабораторной работы были получены знания для работы с текстовыми файлами. Также были получены базовые сведения об основных алгоритмах сортировки. Например, были изучены такие алгоритмы сортировки как сортировка “вставками” и быстрая сортировка. Каждая сортировка была проверена на определённом количестве входных данных, в частности на количестве элементов массива. При большом количестве чисел быстрая сортировка показывает намного лучшие результаты. Например, при 100 000 рандомных чисел быстрая сортировка выполняется за 300 миллисекунд, а “вставками”-22 секунды. При миллионе элементов: быстрая-1,5 секунды, “вставками”-почти час. Входные данных находились в текстовом файле input.txt, исходный отсортированный массив находится в выходном текстовом файле output.txt.