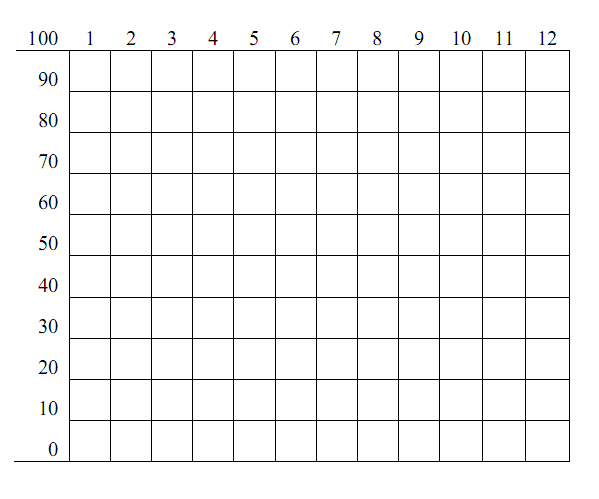
**Министерство образования и науки РФ**

**ФГБОУ ВПО «Уфимский государственный авиационный технический университет»**

**Кафедра** Высокопроизводительных вычислительных технологий и систем

****

**Реализация алгоритма поиска расстояний и пути от каждой вершины до остальных путем удалении n-2 вершин и добавлении их обратно**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовой работе по дисциплине**

**«»**

|  |
| --- |
| ... ПЗ |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Группа МКН-308 | Фамилия И.О. | Подпись | Дата | Оценка |
| Студент | Ибатуллин И.Т. |  |  |  |
| Консультант | Житников В.П. |  |  |  |
| Принял | Газизов Р.К. |  |  |  |

**Уфа 2014**

ФГБОУ ВПО «Уфимский государственный авиационный технический университет»

Кафедра ВВТиС

**ЗАДАНИЕ**

на курсовую работу по дисциплине

**«»**

Студент: Ибатуллин Ильнур Тимерьярович Группа МКН-308

Консультант: Житников Владимир Павлович

**1. Тема курсовой работы**

Реализация алгоритма поиска расстояний и путей от каждой вершины до остальных.

**2. Основное содержание**

* 1. Требуется написать алгоритм, который путем удаления вершин, не теряя целостность и «истинность», а затем путем добавления находит полную матрицу смежности. Данный алгоритм предназначен для поиска кратчайшего расстояния и пути в графах, полученных из реальных городов.
  2. Выполнить программную реализацию разработанного алгоритма на языке С++ в среде разработки Microsoft Visual Studio 2013(или 2012).
  3. Оформить пояснительную записку к курсовой работе.

**3. Требования к оформлению материалов работы**

3.1. Требования к программе

* Т
* В

3.2. Требования к оформлению пояснительной записки

Пояснительная записка к курсовой работе должна быть оформлена в текстовом процессоре Microsoft Word в соответствии с требованиями стандарта СТО УГАТУ 016-2007 и содержать

* титульный лист,
* задание на курсовую работу,
* содержание,
* введение,
* главу, описывающую особенности программной реализации алгоритма,
* главу, описывающую правила работы с программой и примеры ее работы,
* заключение,
* список литературы,
* приложение, содержащее листинг разработанной программы.

**4.** **Список литературы**

1) А.Д., М. (2002). *Введение в теорию фракталов.* Москва, Ижевск.

2) Квант. (1987). *№ 11*, 21.

3) Фролов, А. Ф. (2003). *Язык С#. Самоучитель.*

Дата выдачи задания Дата окончания

\_\_ \_\_\_\_\_\_\_\_\_ 201\_ г. \_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ 201\_ г.

Консультант \_\_\_\_\_\_\_\_\_ Житников В.П.

Принял \_\_\_\_\_\_\_\_\_ Газизов Р.К.

Оглавление

[Аннотация 5](#_Toc346160249)

[1. Теоритические основы 6](#_Toc346160250)

[2. Особенности программной реализации алгоритма 10](#_Toc346160251)

[2.1. Описание классов. 10](#_Toc346160252)

[2.2. Алгоритм работы программы. 10](#_Toc346160253)

[3. Правила работы с программой и примеры ее использования 11](#_Toc346160254)

[Результаты и выводы 14](#_Toc346160255)

[литература 15](#_Toc346160256)

## Аннотация

Курсовая работа состоит из следующих разделов:

* в первом разделе содержатся теоретические сведения кривых, фракталах и основные определения
* второй раздел содержит программную реализацию используемых методов – листинг программы, написанной на языке C#, с подробными комментариями.
* третий раздел включает визуализацию результатов полученных значений, составленную средствами разработки Microsoft Visual Studio .NET.
* результаты и выводы

## Теоритические основы

**Фракталы. Степень изгибания кривой**

В математике существует несколько различных определений размерности, наиболее известна топологическая размерность. Идея определения размерности была высказана ещё А.Пуанкаре. Размерность пустого множества полагается равной «-1» и далее по индукции. Если мы знаем, что такое размерность до n-1, то размерность n некоторого множества означает, что его можно разбить на сколь угодно мелкие части множествами размерности n-1 и нельзя этого сделать множествами размерности n-2. Точка, линия, поверхность имеют соответственно, топологические размерности 0, 1, Более точное понятие топологической размерности ввел нидерландский математик Брауэр (1881-1966). Другие математики (Хаусдорф, Безикович, Колмогоров) определили размерность по-другому. Их определения необязательно дают целые размерности.

Вернемся к эксперименту Ричардсона. Мы выдираем произвольно малую единицу измерения «а», линейку. Затем измеряем длину кривой линии, заменяя ее ломаной линией, составленной из равных отрезков длины «а». Если линейка используется N раз, то общая измеряемая длина равна N\*a. Далее, в соответствии с определением Мандельборта, «фрактальная размерность» ломаной линии равна:

(2.3)

Назовем D «степенью изгибания» кривой линии(границы). В некоторых случаях дробь (2.3) имеет постоянные значения на каждом шаге. Тогда

(2.4)

или

(2.5)

Если обозначить S=Na, то получим

(2.6)

Эта формула показывает, как измеряемая длина увеличивается при уменьшении единицы измерения

**Кривая Коха**

В 1904 году математик Кох дал пример кривой, которая нигде не имеет касательной. Представьте кривую, состоящую из частей, каждая из которых бесконечной длины. Рис 1.0 является хорошим приближением кривой Коха. Построение кривой Коха похоже на построение множества Кантора. Начнем с отрезка-основы: удаляем его среднюю третью часть и заменяем ее сторонами равностороннего треугольника (см рис 1.1)

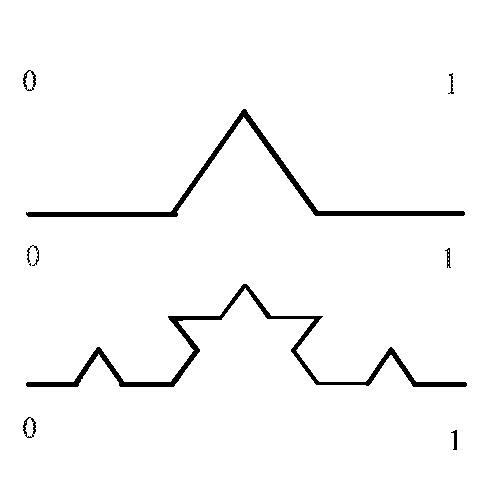
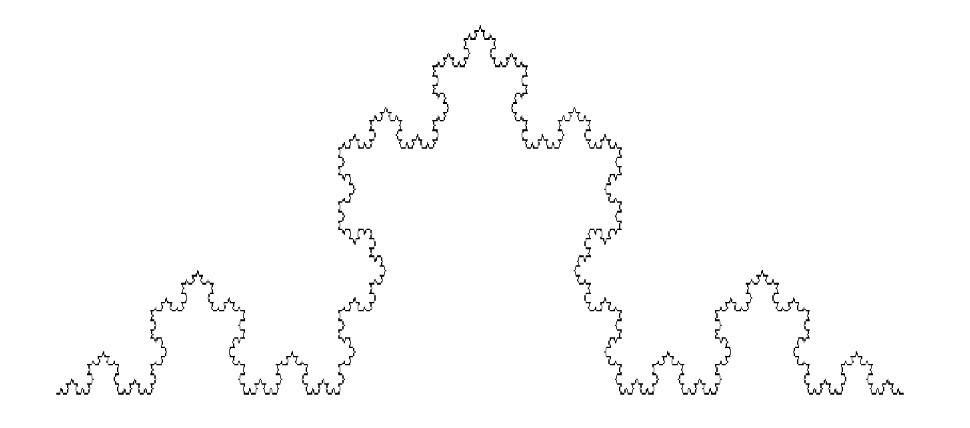


Рис 1.1

Мысленно мы можем представить кривую Коха как предел таких операций. Если основа имеет длину 1, то фрагмент будет состоять из четырех отрезков, каждый длины 4/3. На следующем шаге получаем ломанную,

Рис 1.0

состоящую из 16 отрезков, и имеющую общую длину 16/9 или , и так далее. Так как на каждом шаге (2.7)

то можно применить формулу (2.4). Тогда фрактальная размерность

(2.8)

Кривая Коха самоподобна: каждая часть является миниатюрной копией целого. Можно попробовать самим написать программу построения кривой Коха для случаев, когда основа – отрезок или многоугольник. Для облегчения этого задания дадим анализ построения кривой Коха.

Фиксируем степень приближения *p*. Это означает, что мы будем применять «*p»* преобразований к «*основе».* Если основа – это отрезок, то результатом будет ломаная линия, состоящая из отрезков равной длины . Будем нумеровать отрезки от 0 до включительно. Для каждого шага (соответствующего индексу n) должен нарисоваться отрезок, точнее говоря, вектор. Направление вектора определяется следующим образом.

Запишем индекс n отрезка в четверичной системе. Например, для отрезка с номером 482 = 1\*256+3\*64+2\*16+0\*4+2, то есть, 482=13202 в четверичной системе счисления. Каждое из четырех возможных направлений (на самом деле можно говорить о двух) определяется числом, как показано на рисунке (1.2). Тогда мы найдем направление отрезка с *n*=482:

.

Общая формула имеет вид:

(2.9)

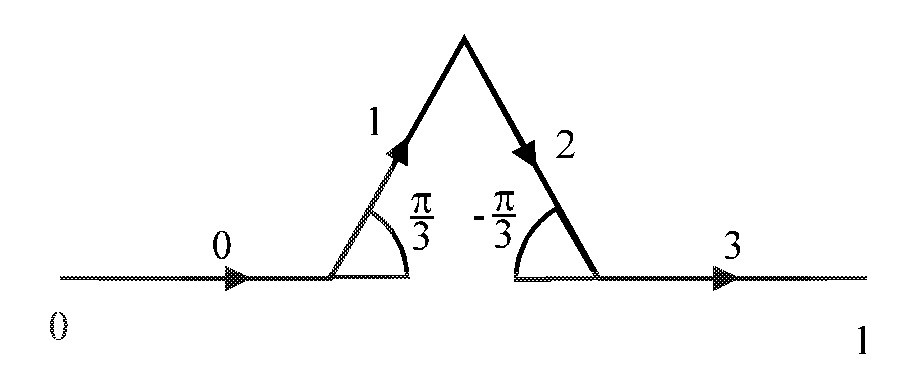


Рис 1.2

В нашем примере a(0)=a(3)=0, a(1) = , a(2) = -

На рис. 1.3 показано, как отрезок с *n*=482 получается из первоначальной основы. Здесь мы пренебрегли с масштабом. Удобнее использовать для *n* в формуле (2.9) не целые числа, а четвертичные дроби, имеющие *p* цифр после запятой. Для этого достаточно поделить номер на . Если , то длина каждого отрезка стремится к нулю, то есть отрезок превращается в точку. Тогда кривая Коха является образом единичного отрезка [0,1].

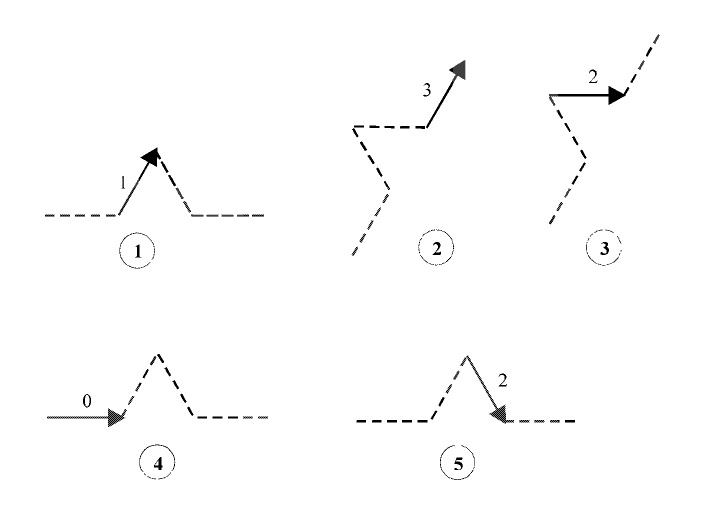
**

Рис 1.3

## 2. Особенности программной реализации алгоритма

### 2.1. Описание классов.

Описание основных функций программы

Выделяется два основных класса – первый(class Line) визуализирует кривую Коха, второй(class surface) поверхностный фрактал рисует.

Class Line – рисует на форме кривую Коха с заданной точностью. Перед рисованием по умолчанию происходит очистка экрана. Поэтому можно менять точность и заново рисовать, что приведет нас к визуализации анимации.

Class surface – рисует поверхность, где для каждой точки определяется значение непрерывной функции, которая влияет на её яркость цвета. Так же имеется метод zoom – непрерывное увеличение изображения.

### 2.2. Алгоритм работы программы.

Class Line

Координаты кривой Коха хранятся в одномерных массивах. Для нахождения следующей итерации используется каждая пара соседних точек текущей кривой. Рассматривая эту пару строго в одном определенном направлении, мы можем использовать их в качестве вектора. При помощи простых математических операций, для одного вектора мы получаем 4 вектора. Суть этого метода заключается в том, что независимо направления вектора, от его длины, мы можем получить всегда необходимую ломаную Коха.

Пусть имеется вектор a, закрепленный на точке (fx,fy), имеющий координаты (lx-fx,ly-fy). Очевидно, что конечной точкой является (lx,ly). Проделаем следующие операции:

float C = 3;

float K = 0.2887f;

float mx = fx + (lx - fx) / 2;

float my = fy + (ly - fy) / 2;

float cx = fx + (lx - fx) / C, cy = fy + (ly - fy) / C;

float dx = lx - (lx - fx) / C, dy = ly - (ly - fy) / C;

float ax = ((lx - fx) \* K), ay = ((ly - fy) \* K);

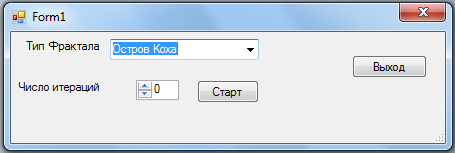
float ex = mx + ay, ey = my - ax;

Где отрезки (fx,fy,cx,cy)(I) , (cx,cy,ex,ey)(II), (ex,ey,dx,dy)(III), (dx,dy,lx,ly)(IV) – являются разбиением отрезка (fx,fy,lx,ly). В качестве основы для острова Коха было выбрано три вершины – A(300,40), B(475,300), C(125,300) и четвертую точку приравниваем первой, для замыкания.

Для поверхностного фрактала все вычисления проводятся простыми математическими операциями, не требующей специальных знаний.

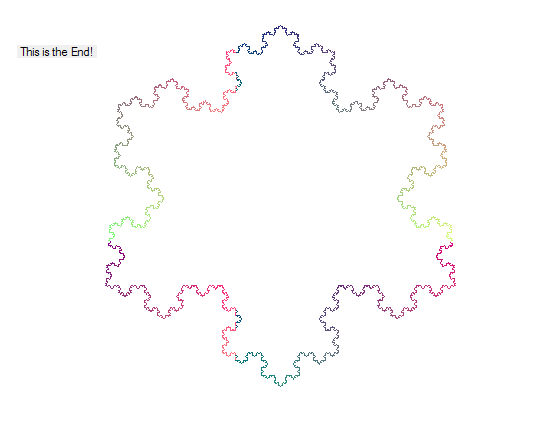
## 3. Правила работы с программой и примеры ее использования

При запуске открывается начальная форма, где можно выбрать, какой из фракталов необходимо нарисовать



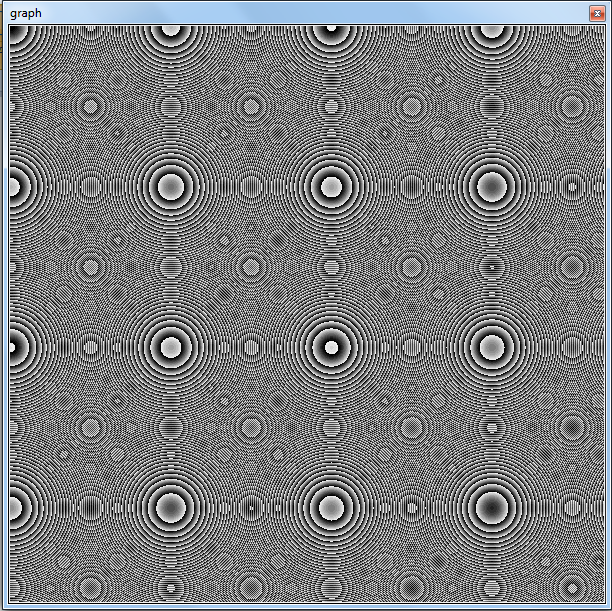
Затем, нажав на кнопку Старт, открывается новое окно, которое не позволяет менять размер окна, где прорисовывается фигура.

Выбрав Остров Коха для 5 итераций, можно увидеть следующее:

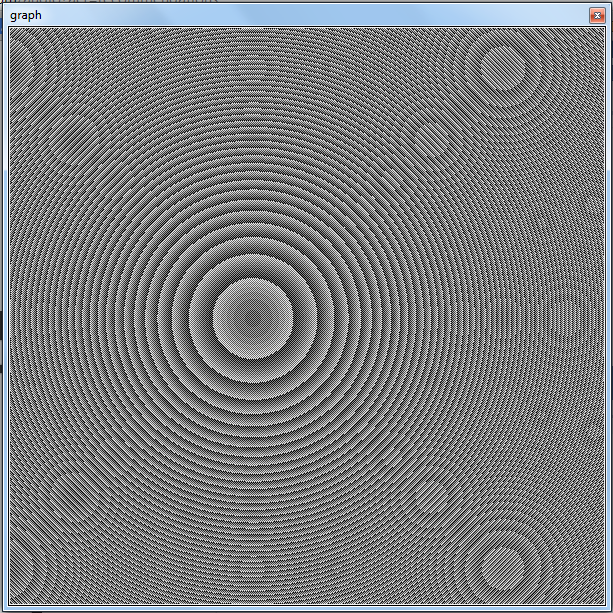


Как видим, отрезки разноцветные, что несколько украшает ломанную.

При выборе пункта «Поверхностный фрактал», можно после запуска увидеть долгую прорисовку точек поверхности, которые рисуются по закону f(x,y)=(x^2+y^2)/S %K, где x и y – это координаты точки, S=20,K=16.



После дорисовки можно выделить прямоугольную область мышкой, которая в дальнейшем прорисуется на полный размер окна.



## Результаты и выводы

При написании данной работы выполнены следующие задания:

1. Разработана программа на С# для визуализации кривой Коха и поверхностного фрактал.
2. Выполнена программная реализация на языке С# в среде разработки Microsoft Visual Studio .NET. Программа содержит следующие обязательные элементы работы с классами:

* конструкторы классов;
* статические компонентные данные и функции;

## литература

А.Д., М. (2002). *Введение в теорию фракталов.* Москва, Ижевск.

Квант. (1987). *№ 11*, 21.

Фролов, А. Ф. (2003). *Язык С#. Самоучитель.*