

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование кафедры)

01.03.02 Прикладная математика и информатика

(код и наименование направления подготовки, специальности)

Системное программирование и компьютерные технологии

(направленность (профиль)/специализация)

БАКАЛАВРСКАЯ РАБОТА

на тему Компьютерная реализация фрактального ряда

Студент

А.В. Цель

(И.О. Фамилия)

(личная подпись)

Руководитель

Г.А. Тырыгина

(И.О. Фамилия)

(личная подпись)

Консультанты

Н.В. Андрюхина

(И.О. Фамилия)

(личная подпись)

Допустить к защите

Заведующий кафедрой к.т.н., доцент, А.В. Очеповский

(ученая степень, звание, И.О. Фамилия)

(личная подпись)

« » 20 г.

АННОТАЦИЯ

Тема бакалаврской работы: «Компьютерная реализация фрактального ряда».

Работа выполнена студентом Тольяттинского государственного университета, института математики, физики и информационных технологий, группы ПМИБ-1502, Цель Антоном Вадимовичем.

Выпускная квалификационная работа посвящена исследованию методов генерации и анализа фрактальных временных рядов и разработке программного кода, реализующего их.

В данной работе используются следующие методы: метод смещения средней точки для генерации фрактального ряда, и метод Минковского для анализа фрактальной размерности.

Объект исследования – фрактальные временные ряды.

Предмет исследования – компьютерное моделирование фрактальных временных рядов и анализ фрактальной размерности.

Цель бакалаврской работы – разработать программную реализацию задачи моделирования фрактального временного ряда.

Для достижения цели работы необходимо решить следующие задачи:

1. Выявить алгоритмы генерации фрактальных временных рядов;
2. Разработать программу для моделирования и анализа фрактального временного ряда;
3. Протестировать получившуюся программную реализацию на реально существующих данных.

Бакалаврская работа представлена на 47 страницах, включает 24 иллюстраций, 34 формулы, список используемой литературы, состоящий из 20 источников.

ABSTRACT

The title of the graduation work is «Computer implementation of the fractal series».

The object of this work is the fractal time series.

The subject of this work is the computer simulation of fractal time series and analysis of fractal dimension.

The key issue of the graduation work is to model a fractal time series.

The graduation work consists of an introduction, three chapters, conclusions and references. Much attention is paid to the description of the generation of a fractal time series by the midpoint displacement method and analysis of the fractal coefficient by Minkowski's method.

We start with the statement of the problem, then follow through with its possible solutions, and give full coverage to generation of a fractal time series by the midpoint displacement method and analysis of the fractal coefficient by Minkowski's method.

The first part describes fractal dimension determination problem, its history and algorithms for the solution of this problem, all existing solutions.

The second part considers the development of the application that solves the fractal dimension determination problem and modeling fractal curve. It contains the implementation of the algorithm, the application interface.

The third part of the work describe the test results and analysis of methods that have been implemented.

As a result, the program code was developed, described and tested to solve the problem fractal dimension determination.

The graduation work is on 47 pages, including 34 figures, a list of 20 references including 3 foreign sources.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
ГЛАВА 1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ФРАКТАЛЬНЫХ МНОЖЕСТВ	7
1.1 Свойства фрактальных множеств	7
1.2 Фрактальная размерность	9
1.3 Показатель Хёрста	16
1.4 Фрактальные временные ряды	19
1.5 Методы модельного представления фрактальных временных рядов	20
1.5.1 Метод смещения средней точки	21
1.5.2 Метод Фурье	22
1.5.3 Метод обобщённого броуновского движения	22
1.6 Методы анализа размерности	23
1.6.1 Метод Ричардсона	24
1.6.2 Метод Минковского	25
1.6.3 Метод R/S анализа	26
ГЛАВА 2 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ	29
2.1 Описание структуры программы	29
2.2 Реализация метода генерации фрактального временного ряда	31
2.3 Реализация метода анализа фрактальной размерности	35
2.4 Интерфейс приложения	41
ГЛАВА 3 ТЕСТИРОВАНИЕ ПРОГРАММНОЙ ЧАСТИ	44
ЗАКЛЮЧЕНИЕ	50
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	51
ПРИЛОЖЕНИЕ А	53
ПРИЛОЖЕНИЕ Б	54
ПРИЛОЖЕНИЕ В	55

ВВЕДЕНИЕ

Издревле мысль о внутреннем подобии объектов и процессов занимала человеческие умы. Сама идея о том, что великое может быть вложено в малое, заставляла учёных пересматривать устоявшиеся принципы геометрии. Как правило, чтобы описать геометрические свойства объектов и процессов можно ограничиться простейшими фигурами: линией, плоскостью, многоугольником, кругом, сферой, квадратичной поверхностью и коническим сечением, либо их комбинациями. Но есть системы, предсказания поведения которых, казалось бы, невозможно предсказать из-за беспорядочных скачков в разных направлениях, из-за чего они никогда не упорядочатся в некую закономерную последовательность. В качестве примера таких систем можно привести прогнозирование погоды, поведение рынка финансов, ритму сердца и мозга, наблюдаемые на электрокардиограмме и другие. Такие системы принято называть хаотичными.

Для описания хаотичных множеств, обладающих свойствами самоподобия, используются фракталы. Само слово «фрактал» в переводе с латинского «fractus» означает дробленный, сломанный, разбитый на куски, а термин же был придуман Б. Б. Мандельбротом ещё в 1975 году. Математическое понятие фрактала выделяет объекты, которые обладают структурами разных масштабов (как больших, так и малых), тем самым отражая принцип организации.

Благодаря тому, что развитие фрактальной геометрии происходит достаточно интенсивно, появление новых определений фракталов, от строго математических, до метафорических вполне закономерно. Для обработки сигналов и изображений количественным показателем часто используется фрактальная размерность, как показатель нерегулярности графика фрактала. Динамические процессы так же имеют фрактальную природу. Всё это указывает на **актуальность** темы исследования.

Объект исследования: фрактальные временные ряды.

Предмет исследования: компьютерное моделирование фрактальных временных рядов и анализ фрактальной размерности.

Цель данной работы — разработать программную реализацию задачи моделирования фрактального временного ряда.

Для реализации поставленной цели, необходимо сформировать следующие задачи:

Задачи:

- 1) Выявить алгоритмы генерации фрактальных временных рядов.
- 2) Разработать программу для моделирования и анализа фрактального временного ряда.
- 3) Протестировать получившуюся программную реализацию на реально существующих данных.

Описание глав:

1. В первой главе описаны основные свойства фрактальных множеств, а также способы генерации фрактальных временных рядов, методы анализа фрактальной размерности этих рядов.
2. Во второй главе была описана структура разработанной программной реализации: выбранный паттерн разработки, описание классов и их методов, технологии разработки, демонстрация графического интерфейса.
3. В третьей главе был произведён тест программной реализации на реальных временных рядах для определения фрактальных свойств.

ГЛАВА 1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ФРАКТАЛЬНЫХ МНОЖЕСТВ

1.1 Свойства фрактальных множеств

Понятие фрактала было введено учёным Бенуа Мандельбротом в 1975 году, но известность он получил только в 1977 году с выходом его книги «Фрактальная геометрия природы». Самого же Мандельброта можно считать основателем фрактальной геометрии.

Суть фрактальной геометрии, как говорил сам Мандельброт, — описание геометрии объектов, которые, по сути, не имеют своей геометрии.

Фрактал — это сложное математическое понятие, которое базируется на «размерности» природного объекта. Удивительно, но в природе существует множество объектов, обладающими этими свойствами — снежинки, облака, корни деревьев и растений, горный рельеф, река, береговая линия, бронхи лёгких и другие.

Мандельброт писал: «Фракталы — это математические, природные или созданные человеком объекты, которые мы называем неправильными, шероховатыми, пористыми или раздробленными, причём этими свойствами обладают фракталы в одинаковой степени в любом масштабе. Можно сказать, что форма этих объектов не изменятся от того, смотрим ли мы на них вблизи или издали» [9]. Легко уловить то, что одним из основных свойств фрактала Мандельброт выделял масштабную инвариантность.

Простейший пример инвариантности — это самоподобие. Объекты, обладающие таким свойством, состоят из множества частей, которые получаются из преобразованных частей целого объекта, подобных целому. При всём при этом в качестве преобразований над объектом могут выступать переносы, вращения и осевые симметрии, что допускает наложение малой части объекта на большую, либо на весь объект в целом.

Самым известным в математике объектом, обладающим свойством инвариантности является снежинка Коха. На рисунке 1.1 представлены первые шаги построения снежинки Коха: берётся равнобедренный треугольник, каждая его грань разделяется на три равные части и средняя часть заменяется подобным треугольником меньшего размера. Данный алгоритм может повторяться бесконечное количество раз. На рисунке 1.1 продемонстрированы первые четыре шага алгоритма.

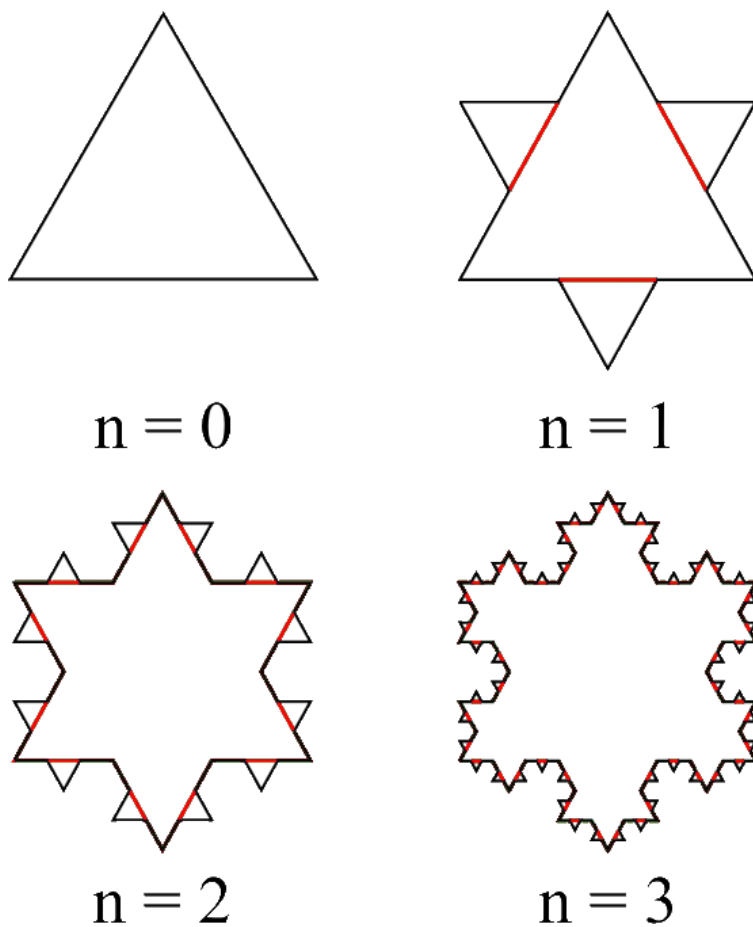


Рисунок 1.1 — Первые четыре шага построения снежинки Коха

Как правило, самоподобные фрактальные объекты являются искусственными (их так же называют регулярными), так как в природе они не встречаются.

Реальные фрактальные объекты обладают более сложным свойством самоподобия — самоаффинностью [10]. Они отличаются от искусственных тем, что не являются строго самоподобными, а также не имеют свойства

бесконечной делимости, то есть не обладают бесконечно большими и бесконечно малыми частями. Фрактал обладает самоаффинностью в том случае, если при «разложении» его на части вдоль пространственных осей, построение происходит с разными коэффициентами. Это означает, что оси между собой не эквивалентны, а значит и отвечают разным физическим величинам. Типичным примером самоаффинных фракталов являются временная зависимость броуновской частицы, фрактальные временные ряды и так далее.

Так же фракталы обладают свойствами дробной размерности, которая должна быть большей, чем топологическая [1]. Это свойство относится к фрактальной размерности.

Последним, но не менее важным свойством фракталов является способ построения: любой фрактал может быть построен при помощи рекурсивных процедур.

Для количественного описания фракталов необходимо и достаточно всего лишь одной величины — размерности Хаусдорфа (показатель скейлинга). Данный показатель описывает сохраняемость геометрии или статистических характеристик при изменении масштаба. Однако в биологии, химии, физики и даже в телекоммуникационных отраслях нередко встречаются явления, требующие распространения понятия фрактала на более сложные структуры, которые имеют более одного показателя скейлинга. Данные структуры, как правило, характеризуются сразу спектром показателей, где размерность Хаусдорфа лишь одна из многих величин. Для таких фракталов был придуман термин «мультифракталы» [3, 19].

1.2 Фрактальная размерность

Прежде чем описывать фрактальную размерность, необходимо пояснить понятия топологической размерности и меры.

Топологическая размерность геометрической фигуры — это число координат, необходимых для определения положения, лежащей на этой фигуре

точке. Можно просто определить, что для определения размерности точки необходимо только одна координата, для плоскости две, а для объёмной фигуры три. Из этого можно сделать простой вывод: размерность некой n -мерной фигуры всегда выражена целым числом, равным n .

Метрической размерностью называют число, выражающее связь между измеряемым размером фигуры и единицей, лежащей в основе измерения. По факту, метрическая размерность некой фигуры $N(\delta)$ будет равна количеству вмещающихся в неё объектов δ . Для примера возьмём прямую: для расчёта её метрической размерности определяют количество сантиметров, метров, миллиметров и так далее, полностью её покрывающих. Для расчёта площади измеряется количество единичных кубов, помещающихся в объёмную фигуру. Но при уменьшении единиц измерения, метрическая размерность всегда увеличивается, и наоборот: при увеличении единиц измерения, метрическая размерность уменьшается. В простейшем случае можно вывести формулу расчёта размера фигуры

$$N \delta = (1/\delta)^D, \quad (1)$$

где δ — количество покрывающих фигуру метрических объектов;

D — метрическая размерность.

То есть если, например, начать измерять плоскую фигуру, допустим, в метрах ($\delta = 1$), а затем последовательно заменять метры дециметрами, сантиметрами, миллиметрами и так далее, то и δ будет менять значения на $\delta = 1/10$, $\delta = 1/100$ и так далее. При этом, например, для линии всегда будет выполняться равенство $N(\delta) = (1/\delta)^1$, для плоскости $N(\delta) = (1/\delta)^2$ и так далее. Таким образом, можно легко выделить метрическую размерность для разных объектов: для линии размерность будет равна единице, для плоскости двойке, для объёмных же фигур трём и так далее [18].

Если размерность некого объекта выражается целым числом в одном масштабе, то при переходе к более маленьким или большим значениям δ ,

размер объекта останется неизменным. Но для фракталов данное правило не действует.

Для примера можно рассмотреть береговую линию, потому что она очень извилистая и имеет фрактальную природу. Если мы замерим длину всех её извилин, то это значение изменится при увеличении масштаба, потому что проявятся новые более мелкие извилины. К примеру, если мы замерим длину некой береговой линии, то изменение масштаба приведёт к тому, что это значение изменится, а значит, длина береговой линии в разных масштабах всегда будет разной.

Данный эффект возникает из-за того, что при увеличении масштаба мы способны учитывать лишь те извилины, которые возможно сопоставить с единицей измерения. Это означает, что длина некой береговой линии в метрах, к примеру, будет равняться 126 метрам, при этом же при увеличении масштаба (уменьшении единицы измерения) длина этой же линии будет равна, к примеру, 134213 миллиметрам.

Линии, меняющие свой размер в зависимости от масштаба, являются чем-то промежуточным между математической линией и плоскостью, а их размерность лежит в диапазоне $1 < D < 2$. Если единица измерения δ стремится к нулю, то размер фигуры $N(\delta)$ асимптотически приближается к величине u^{-D} , что можно записать следующим образом:

$$N(\delta) \sim (1/\delta)^D. \quad (2)$$

Размерность D имеет следующий математический смысл: она указывает на то, как равномерно элементы множества заполняют собой всё пространство [15]. Например, для линии на плоскости значение фрактальной размерности D стремиться к единице, что означает «гладкость» множества, а при ярко выраженной извилистости и изрезанности некого множества в разных масштабах указывает на то, что D стремиться к двойке.

Таким образом, можно утверждать, что размерность D является методом определения размерности фрактального множества в метрическом пространстве.

Существует не один подход для расчёта размерности фрактальных множеств, но все они так или иначе опираются на размерность Хаусдорфа, которая рассчитывается по формуле

$$D = -\lim_{\delta \rightarrow 0} \frac{\ln N \delta}{\ln 1/\delta}, \quad (3)$$

где $N(\delta)$ — минимальное количество n -мерных условных единиц (квадраты, круги, шары и так далее) полностью покрывающих множество;

δ — размер отдельно взятой условной единицы. Данную зависимость Феликс Хаусдорф вывел ещё в далёком 1919 году [6].

Пример вычисления размерности Хаусдорфа для двумерного множества представлен на рисунке 1.2.

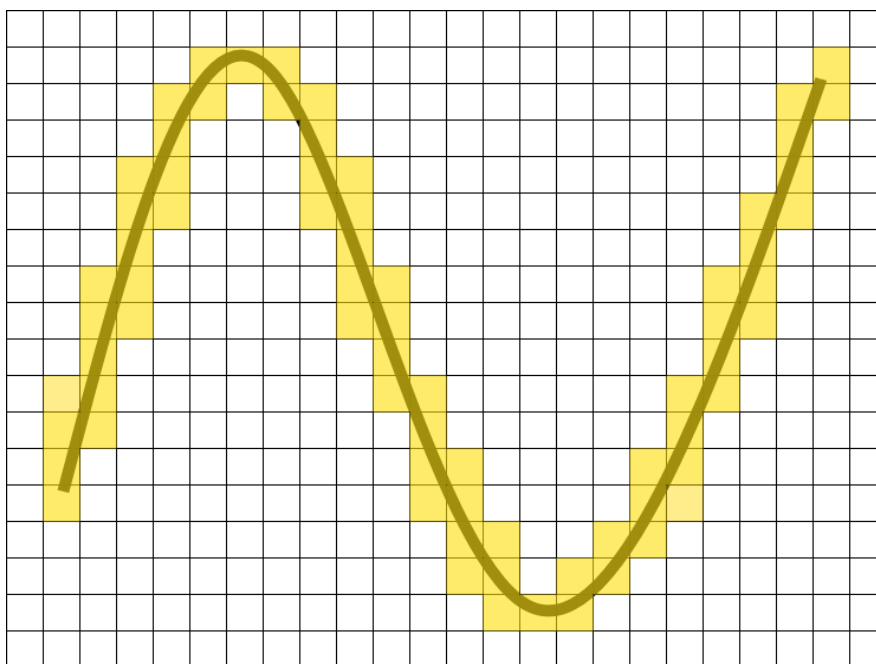


Рисунок 1.2 — Пример вычисления размерности Хаусдорфа для двумерного множества

При изменении масштаба изменяется и сетка построения, а значит, изменяется и количество единичных кубов, лежащих на множестве. Поэтому для

нахождения более точного значения размерности необходимо иметь минимально возможный масштаб сетки.

Существуют и другие виды размерности как, например, размерность Минковского. Данная размерность так же не редко обозначается как «грубая размерность» ограниченного множества в метрическом пространстве. Расчёт размерности происходит по формуле

$$D = - \lim_{\delta \rightarrow 0} \frac{\ln N_{\delta}}{\ln \delta}, \quad (4)$$

где N_{δ} — минимальное число множеств диаметром δ , которыми можно покрыть множество.

Если же предел не существует, то в таком случае необходимо рассматривать верхний и нижний пределы и говорить соответственно о верхней и нижней размерности Минковского.

Данная размерность тесно связана с размерностью Хаусдорфа, хоть и существуют множества, для которых они различаются, во многих случаях эти размерности совпадают, поэтому говоря о размерностях, чаще всего не имеет значение, какая размерность имеется в виду [7].

Иногда размерность Минковского называют дробной размерностью, однако дробность D не является единственным необходимым условием фрактальности. Существуют фрактальные объекты, размерность которых выражается целым числом. Как говорилось ранее, Мандельброт выделил ключевое условие определения фрактала: размерность D должна быть строго больше топологической размерности [9].

Минковский предложил теорию с обобщением размерности для компактного множества $A \subset R^n$. Данную теорию легко показать наглядно при помощи аппроксимации A через объединение шаров и последующим суммированием их объёмов и мер (рисунок 1.3).

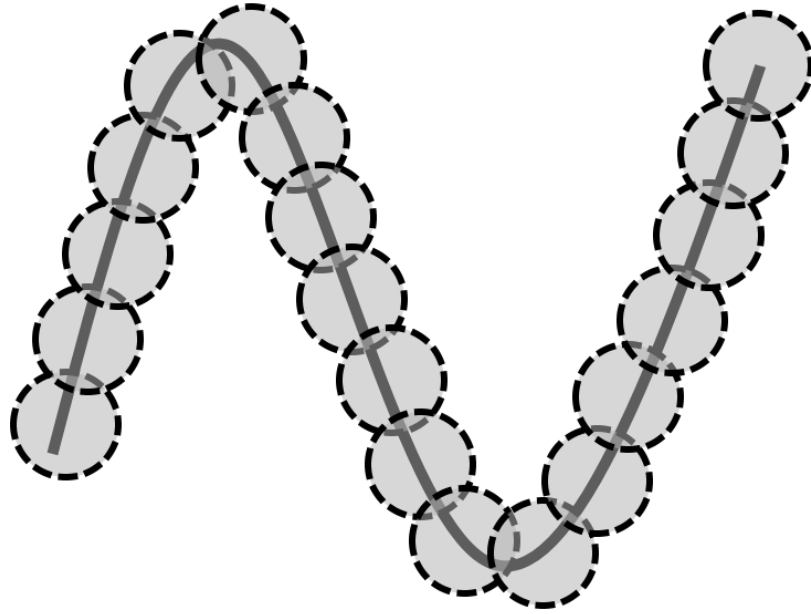


Рисунок 1.3 — Аппроксимация A через объединение шаров размерностью Минковского

Пусть $N(\varepsilon)$ — минимальное число шаров радиуса ε , необходимых для покрытия компактного множества A . Тогда мера D множества A , которую мы обозначим через $B_d(A)$, должна приближённо удовлетворять условию

$$B_d A \propto N \varepsilon \varepsilon^D. \quad (5)$$

Полагая, что $B_d A > 0$ мы можем предположить, что для некоторого $c > 0$ будет правдиво равенство

$$N \varepsilon \approx \frac{c}{\varepsilon^D}. \quad (6)$$

Логарифмируя левую и правую части, получим следующее выражение

$$\log N \varepsilon \approx \log c - D \log \varepsilon, \quad (7)$$

которое мы можем записать как

$$D = -\frac{\log N \varepsilon}{\log \varepsilon} + \frac{\log c}{\log \varepsilon}. \quad (8)$$

Так как $\log \varepsilon \rightarrow -\infty$ при $\varepsilon \rightarrow 0$, то размерность Минковского для множества A должна удовлетворять равенству (9).

$$D = -\lim_{\varepsilon \rightarrow 0} \frac{\log N \varepsilon}{\log \varepsilon}. \quad (9)$$

Если предел существуют, то говорят, что выражение (9) определяет размерность Минковского для множества A [11].

В качестве примера измерим размерность Хаусдорфа для снежинки Коха, продемонстрированной на рисунке 1.1, где n — это шаг построения снежинки. За единицу измерения возьмём сторону многоугольника, которая будет получаться на каждом шаге построения.

На рисунке 1.4 продемонстрирована пятая итерация кривой Коха. Для расчёта размерности будем использовать три различных масштаба.

Для верхней сетки на картинке 1.4 длина отрезка равна единицы (так как мы взяли всю кривую за единичный отрезок), а разбита она на девять частей, поэтому $\delta = 1/9$. Всего закрашено 12 квадратов, поэтому после подстановки $D \approx 1.1309$.

Рассмотрим среднюю сетку рисунка 1.4. Теперь значение $\delta = 1/18$, а количество «закрашенных» квадратов равно 30. Отсюда получаем значение $D \approx 1.1767$.

Рассмотрим последнюю сетку рисунка 1.4. Значение $\delta = 1/36$, количество «занятых» фракталом клеток равно 72. Итого получаем значение $D \approx 1.1934$.

Показатель Хёрста является крайне устойчивым, благодаря чему взыскал свою популярность. Коэффициент содержит минимальное число предположений об изучаемой системе, а также позволяет классифицировать временные ряды — отличать случайные фрактальные временные ряды от неслучайных.

Показатель степени Хёрста определяется в терминологии асимптотического поведения некоего масштабированного диапазона как функции отрезка времени некоего временного ряда по отношению

$$E \frac{R(n)}{S(n)} = Cn^H, n \rightarrow \infty, \quad (10)$$

где $R(n)$ — размах накопленных отклонений первых n значений от среднего значения ряда;

$S(n)$ — стандартное отклонение;

$E[x]$ — математическое ожидание;

n — величина промежутка времени (количество делений в отрезке временного ряда);

C — константа.

Точность коэффициента Хёрста напрямую зависит от длины временного ряда: чем ряд длиннее, тем точнее будет рассчитан коэффициент.

Показатель Хёрста напрямую связан с фрактальной размерностью D по формуле

$$D = 2 - H, \quad (11)$$

где D — оценка степени изломанности ряда (фрактальная размерность);

H — показатель Хёрста.

Мандельброт в своей книге показал, что при помощи коэффициента Хёрста так же можно вывести размерность пространства вероятностей, которую он обозначил как A и предложил рассчитывать по формуле

$$A = 1/H. \quad (12)$$

На основе этих данных возможно предположить вариацию фрактальной размерности:

- Если $H \approx 0$, $D \approx 2$ и $A \rightarrow \infty$, то это прямая линия;
- Если $H = 0.5$, $D = 1.5$ и $A = 2$, то это случайный ряд;
- Если $H = 1$, $D = 1$ и $A = 1$, то это бесконечный линейный тренд.

Каждое наблюдение хранит в себе память о всех предыдущих событиях, всё что происходит в момент времени, напрямую отражается в будущем [10]. Здесь время становится важнейшим фактором, а так как время итеративно, то мы можем вывести корреляционное соотношение по формуле

$$C = 2^{2H-1} - 1, \quad (13)$$

где C — это мера коллекции;

H — коэффициент Хёрста.

Показатель Хёрста может быть определён на отрезке значений $[0; 1]$ и рассчитан в следующих пределах:

- Если $0 \leq H < 0.5$, то это означает, что система подвержена переменам. Такую ещё обзывают как «возврат к среднему», что означает обратную зависимость текущего состояния от времени: если в системе произошёл рост, то в следующем периоде начнётся спад и наоборот. Устойчивость всей системы зависит от близости показателя Хёрста к нулю: чем ближе значение C в уравнении (12) к значению -0.5 (отрицательной корреляции), тем устойчивее система [5];

- Если $H = 0.5$ — ряд является случайным. Это означает, что все события случайны и некоррелированные и текущие значения никак не влияют на будущее, то есть происходит нормальное распределение;

- Если $0.5 < H \leq 1$, то это означает, что работа идёт с трендоустойчивым рядом. То есть, если временной ряд убывал на промежутке, то и дальше он будет убывать [12, 13].

1.4 Фрактальные временные ряды

Фрактальность пространства и времени имеет глубокие физические причины. Связь между фрактальными структурами пространства и времени образуют «стрелу пространство-время», которая определяет эволюцию самоорганизующейся системы.

Фрактальность времени можно изучать при помощи различных «фрактальных» подходов.

Временной ряд состоит из последовательности значений некой величины, которые были измерены в последовательные моменты времени — как правило, в равные промежутки [4]. Временной ряд характеризует случайные изменения величин. Наиболее популярными примерами являются временные изменения неких экономических показателей, таких как, например, колебания обменных курсов валют; временные характеристики метеорологических данных — изменение температуры воздуха, количества осадков, скорости ветра и другие; некие стохастические процессы в медицине, как, например, электрокардиограмма и так далее.

Традиционные разделы стохастического анализа временных рядов основывается на предположении, что некая изучаемая система является случайной (процесс, порождающий временной ряд, состоит из множества составных частей и степеней свободы). Но многие временные ряды обладают инвариантностью относительно масштабных преобразований, что делает статические методы анализа несостоятельными, а вопрос анализа перекладывается на изучения фрактального множества, который описывает временной ряд [2].

Фрактальные временные ряды бывают случайными (при значении показателя Хёрста $H = 0.5$). В таком случае временной ряд состоит из несвязанных между собой событий: настоящее никак не влияет на будущее. Но степенной показатель H , как правило, больше 0.5, а вероятностные распределения не являются нормальными.

Фрактальные временные ряды могут быть персистентными (трендоустойчивыми), если показатель Хёрста лежит в диапазоне $0.5 < H < 1$. В таком случае временной ряд склонен сохранять своё состояние некоторое время в будущем. Чем ближе значение коэффициента H к единице, тем очевидней трендоустойчивость ряда, а если коэффициент близится к значению 0.5, то говорят, что ряд сильно зашумлён (присутствует белый шум).

Персистентный ряд — это обобщённое броуновское движение или смещённые случайные блуждания.

Обратным персистентному фрактальному временному ряду является антиперсистентные ряды (так же их называют эргодическими рядами). В таком ряде происходит «возврат к среднему», то есть система пытается изменить своё состояние на противоположное. Устойчивость антиперсистентного ряда определяется значением коэффициента Хёрста: чем ближе он к нулю, тем устойчивее система [14].

В жизни персистентные временные ряды встречаются гораздо чаще, чем антиперсистентные.

1.5 Методы модельного представления фрактальных временных рядов

Чтобы иметь возможность проанализировать фрактальный временной ряд, необходимо для начала его смоделировать. Данная задача является достаточно тривиальной, так как существует не один способ генерации фрактального временного ряда.

Каждый из таких методов отличается простотой компьютерной реализации, скоростью работы, а также качеством сгенерированного фрактального множества.

Стоит отметить, что все фрактальные ряды, смоделированные методами, описанными далее, являются однозначными и не самоаффинными.

1.5.1 Метод смещения средней точки

Первым и наиболее популярным алгоритмом является алгоритм смещения средней точки. Свою популярность данный метод взыскал за свою простоту реализации на ЭВМ в качестве компьютерного алгоритма. Но говоря о плюсах, стоит сразу же указать на минус данного метода — сложность вычислений возрастает в геометрической прогрессии, поэтому для больших входных данных время работы алгоритма может сильно возрасти.

Сам же алгоритм заключается в разбиении единичного отрезка X по оси абсцисс на $N = 2^m$ частей, где m — любое положительное число. В реализации данного алгоритма можно выделить следующие шаги:

1. Берётся единичный отрезок по оси X произвольной длины;
2. Для выбранного промежутка определяются начальные значения y_{left} и y_{right} как случайные величины;
3. Следующим шагом происходит разбиение выбранного единичного отрезка пополам, рассчитав середину x_{middle} как среднее значение между двумя границами отрезка x_{left} и x_{right} .

Значение по оси ординат будет считаться исходя из значений y_{left} и y_{right} по следующей формуле:

$$y_{middle} = \frac{y_{left} + y_{right}}{2} + h, \quad (14)$$

где h — случайная величина, распределённая по нормальному гауссову закону с нулевым средним и дисперсией σ_h , которая определяется соотношением:

$$\sigma_h = r^H, \quad (15)$$

где r — расстояние от средней точки исследуемого отрезка x_{middle} , которое определяется выражением:

$$r = \frac{x_{right} - x_{left}}{2}. \quad (16)$$

1.5.2 Метод Фурье

Вторым чуть менее популярным методом генерации фрактальных временных рядов является метод Фурье. Данный метод получил широкое признание благодаря тому, что в основе, как это не сложно догадаться, лежит разложение функции в ряд Фурье.

Исходную временную фрактальную кривую $y(t)$ записывают в виде:

$$y(t) = \sum_{f=f_{min}}^{f_{max}} c(f) \sin(2\pi f t + \varphi(f)), \quad (17)$$

где $f \in [f_{min}, f_{max}]$ — набор частот, входящих в разложение Фурье;

$c(f)$ и $\varphi(f)$ — амплитуда и фаза соответствующих гармоник.

Для каждой гармоники f определяется фаза φ как случайное значение, лежащее в пределах диапазона $0, 2\pi$.

Зависимость амплитуды c от частоты f задаётся скейлинговым соотношением:

$$c(f) = k f^{-\beta}, \quad (18)$$

где k — общий амплитудный коэффициент;

β — фрактальная размерность, которая по смыслу и значению полностью совпадает с показателем Херста H . Это означает, что расчёт фрактальной размерности полностью идентичен методу средней точки, за исключением вычитаемого из двойки коэффициента, и производится по формуле $D = 2 - \beta$.

1.5.3 Метод обобщённого броуновского движения

Метод обобщённого броуновского движения, как это понятно из названия, полностью основан на модели описания движения некой смоделированной броуновской частицы, совершающей случайные перемещения в пространстве.

Принято считать, что, как и в классической модели броуновского движения, перемещения частицы являются полностью хаотическими, не зависимыми от предыдущих перемещений. Однако данная модель имеет различие с классическим представлением — перемещение броуновской частицы полностью зависит от всех перемещений, совершённых ей во всех предыдущих отрезках времени [18]. Для зависимости перемещения обобщённой броуновской частицы вдоль одной из пространственных координат от времени можно записать соотношение:

$$B_H t = \frac{1}{\Gamma(H + 1/2)} \int_{-\infty}^t K(t - t') dB t', \quad (19)$$

где $\Gamma(z)$ — это гамма функция;

H — показатель Хёрста, определяющий фрактальную размерность, как соотношению $D = 2 - H$ и отвечающее условию $0 \leq H \leq 1$; $dB t'$ — независимые случайные перемещения, обычной броуновской частицы в момент времени t' ;

$K(t - t')$ — функция, отвечающая за память системы и определяющаяся соотношением:

$$K(t - t') = \begin{cases} t - t'^{-H-1/2}, & 0 \leq t' \leq t \\ t - t'^{-H-1/2} - -t'^{-H-1/2}, & t < 0 \end{cases} \quad (20)$$

1.6 Методы анализа размерности

Далеко не всегда точно можно сказать, посмотрев на систему, является ли она фракталом, поэтому оценка фрактальных свойств таких систем является важным вопросом в изучении фракталов. Для решения данной проблемы существуют множество методов определения фрактальной природы, каждый из которых имеет свои ограничения по применению: одни методы могут быть использованы только к самоподобным фракталам, другие же наоборот — к самоаффинным, а для самоподобных являются не применимыми, третьи же применяются только к однозначным системам и так далее. Стоит подчеркнуть,

что метод определения фрактальной размерности полностью зависит от типа природы фрактала, поэтому для разных типов может меняться точность расчётов. Из-за чего не редко на практике стоит вопрос о лучшем методе, для решения задачи, либо используются сразу несколько способов, чтобы более точно оценить фрактальную размерность.

1.6.1 Метод Ричардсона

Первый рассматриваемый метод является представителем первых алгоритмов определения фрактальной размерности, применимый только к самоподобным фрактальным множествам, называется метод Ричардсона. Его суть достаточно проста, потому как фрактальность некой фрактальной кривой длины L измеряется путём многократного наложения эталонного отрезка r вдоль кривой. Благодаря своей простоте данный метод имеет повышенную точность решения.

Алгоритм данного метода достаточно простой: берётся первая точка фрактальной кривой, представленной в виде последовательностей координат (x, y) с наибольшей точностью, и рассчитывается расстояние пересечения кривой и заданным набором отрезков с окружностями заранее заданного радиуса r . Для точки пересечения кривой и окружности действия повторяются: так же рассчитывается новое пересечение окружности и ранее отобранной точкой [17]. Данный алгоритм повторяется для всего фрактального множества.

Длина фрактальной кривой L рассчитывается как суммарная длинна всех N полученных отрезков длины r и неполного отрезка r' от последней найденной точки пересечения до конечной точки кривой:

$$L r = N r \cdot r + r'. \quad (21)$$

Для определения фрактальной размерности D исследуемого временного ряда строится зависимость $\log L(\log r)$ и проводится линейная аппроксимация.

Тангенс угла наклона β линейной аппроксимации связан с размерностью отношением $D = 1 - \beta$.

1.6.2 Метод Минковского

Метод Минковского является наиболее точным методом анализа фрактальной размерности для самоаффинных кривых. Его принцип заключается в том, чтобы покрыть фрактальную кривую окружностями и рассчитать зависимость полученного покрытия S от радиуса окружности r .

Частным случаем расчёта фрактальной размерности Минковского является в покрытие фрактального временного ряда не кругами, а квадратами, что облегчает реализацию компьютерного алгоритма. В таком случае поиск фрактальной размерности сводится к поиску отношения количества квадратов $N(\varepsilon)$ размером ε , покрывающих всю фрактальную кривую, к размеру одной ячейки ε .

Вычисление размерности Минковского d обычно упирается в следующее выражение:

$$\log N \varepsilon = \log c - d \cdot \log \varepsilon, \quad (23)$$

где ε — сторона единичного квадрата;

$N \varepsilon$ — наименьшее число клеток стороной ε , необходимых для покрытия фрактальной кривой.

Не сложно заметить, что график зависимости (формула 23) от $\log N \varepsilon$ и $\log \varepsilon$ является прямой, где c — константа, а d — угловой коэффициент.

Если использовать клетки только двух размеров ε_1 и ε_2 , то неизвестные переменные c и d можно вычислить из системы уравнений:

$$\begin{aligned} \log N \varepsilon_1 &= \log c - d \cdot \log \varepsilon_1 \\ \log N \varepsilon_2 &= \log c - d \cdot \log \varepsilon_2 \end{aligned} \quad (24)$$

В таком случае, из-за того, что величины $N(\varepsilon)$ могут быть найдены лишь приближённо, поэтому имеет смысл оценить $N(\varepsilon)$ для большего числа различных значений ε . В таком случае мы получим систему, в которой число

уравнений будет больше числа неизвестных и которая, скорее всего, не будет иметь точного решения. В таких случаях стандартным решением является найти значения $\log c$ и d минимизирующие сумму квадратов отклонений (метод наименьших квадратов) [20].

Рассмотрим пример, когда необходимо интерполировать $\xi_i, \eta_i, i = 1, \dots, n$ прямой линией. Предположим, что $\eta_i = b + m\xi_i$, тогда прямая $y = b + mx$ будет называться наилучшим приближением к $\eta(\xi)$, если сумма квадратов отклонений $y_i - \eta_i$ минимальна. Говоря другими словами, необходимо найти значения b и m , при которых бы уравнение будет достигать свой минимум

$$E = \sum_{i=1}^n (y_i - \eta_i)^2 = \sum_{i=1}^n (b - m\xi_i - \eta_i)^2. \quad (25)$$

Значения этих коэффициентов найдутся при решении системы уравнений

$$\begin{aligned} \frac{\partial E}{\partial b} &= 0 \\ \frac{\partial E}{\partial m} &= 0 \end{aligned}. \quad (26)$$

В матричной системе это можно записать в следующем виде:

$$\begin{matrix} n & & n \\ & \xi_i & \eta_i \\ n & \sum_{i=1}^n & b \\ & \xi_i & m \\ & \sum_{i=1}^n \xi_i^2 & \sum_{i=1}^n \xi_i \eta_i \end{matrix} = \begin{matrix} n \\ \sum_{i=1}^n \eta_i \\ \sum_{i=1}^n \xi_i \eta_i \end{matrix}. \quad (27)$$

1.6.3 Метод R/S анализа

R/S анализ по праву считается одним из самых популярных методов расчёта фрактальной размерности. Первоначально данный метод предложил Хёрст для анализа зависимости стока воды из различных озёр и рек. Но как оказалось, данный метод так же можно использовать для расчёта фрактальной размерности D фрактальных временных рядов, а так же самоаффинных и однозначных кривых.

Фрактальный временной ряд преобразуется в дискретный набор данных M длиной M . Затем ряд M преобразуется во временной ряд N длины $N = M - 1$ исходя из логарифмических отношений:

$$N_i = \ln \frac{M_{i+1}}{M_i}, i = 1, 2, 3, \dots, M - 1. \quad (29)$$

Следующим шагом рассматриваем период времени разделённый на A смежных подпериодов длины n , так что $A \cdot n = N$. Каждый такой период помечен как $I_a, a = 1, 2, 3, \dots, A$. Каждый из элементов в I_a помечен как N_k , при этом $k = 1, 2, 3, \dots, n$. Для каждого I_a длины n определяется среднее значение согласно соотношению:

$$e_a = \frac{1}{n} \cdot \sum_{k=1}^n N_k, \quad (30)$$

где e_a — среднее значение N_k , содержащееся в подпериоде I_a длины n .

Теперь необходимо рассчитать временной ряд накопленных отклонений X_k от среднего значения для каждого подпериода I_a :

$$X_{k,a} = \sum_{k=1}^n N_{k,a} - e_a, k = 1, 2, 3, \dots, n. \quad (31)$$

В дальнейшем диапазон определяется как максимальное значение за вычетом минимального значения $X_{k,a}$ в пределах каждого подпериода I_a :

$$R_{I_a} = \max X_{k_a} - \min X_{k_a}, 1 \leq k \leq n. \quad (32)$$

Определяется выборочное стандартное отклонение, рассчитываемое для каждого под периода I_a :

$$S_{I_a} = \left(\frac{1}{n} \cdot \sum_{k=1}^n (N_{k,a} - e_a^2) \right)^{0.5}. \quad (33)$$

Теперь каждый диапазон R_{I_a} нормализуется путём деления на соответствующие значения S_{I_a} , поэтому повторно нормированный размах в течении каждого I_a равен R_{I_a}/S_{I_a} . При помощи двух смежных подпериодов длины n вычисляется среднее значение R/S для длины n :

$$\frac{R}{S}_n = \frac{1}{A} \cdot \prod_{a=1}^A \frac{R_{I_a}}{S_{I_a}}. \quad (34)$$

Длина n увеличивается до следующего, более высокого значения, а $\frac{M-1}{n}$ является целочисленным значением. Описанная ранее процедура подсчёта повторяется пока не достигнута равенство $n = \frac{M-1}{2}$. Затем выполняется линейная регрессия методом наименьших квадратов на $\ln(n)$ как независимой переменной, и $\ln(R/S)_n$ — как зависимой переменной. Отрезок, отсекаемый на координатной оси, является оценкой $\ln(c)$ и константой. Показатель Хёрста H является наклоном прямой, и он связан с фрактальной размерностью D фрактальной кривой соотношением $D = 2 - H$.

ГЛАВА 2 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

2.1 Описание структуры программы

Разработка программной части выпускной квалификационной работы производилась на языке программирования Java версии 8.

Java — строго типизированный объектно-ориентированный язык программирования, разработанный компанией «Sun Microsystems» (ныне «Oracle»). Благодаря своей строгой типизации данный язык программирования как никогда кстати подходит для разработки программы, так как позволяет строго структурировать и ограничить слои работы программы.

Так же в ходе разработки были использованы технологии JavaFX для создания GUI (графического интерфейса пользователя); Junit версии 4.11 для тестирования разрабатываемой программы; Log4j версии 1.7.26 для введения логгирования программы, для её последующей отладки.

Из-за использования технологии JavaFX был применён паттерн разработки Model-View-Controller (MVC). Он позволяет чётко определить слои приложения и назначить каждому слою свою задачу:

- Слой Model — отвечает за логику программы. Реализует все необходимые расчёты программы, а затем передаёт итоговые расчёты в следующий слой.

- Слой View — отвечает за реализацию графического интерфейса программы (GUI). Так как технология JavaFX позволяет использовать файлы FXML для описания графического интерфейса (FXML — язык разметки, основанный на языке XML, разработанный специально для описания графических компонентов), то и всё описание графического слоя было перенесено в файлы с расширением *fxml*.

- Слой Controller — отвечает за взаимодействие пользователя и графического интерфейса. Данный слой реализует обработку нажатий кнопок и представление слою view всю необходимую информацию из слоя model.

Полная UML-диаграмма классов проекта представлена в приложении Б.

Функция `publicstaticvoidmain()` реализована в классе `App` (UML-диаграмма класса представлена на рисунке 2.1).

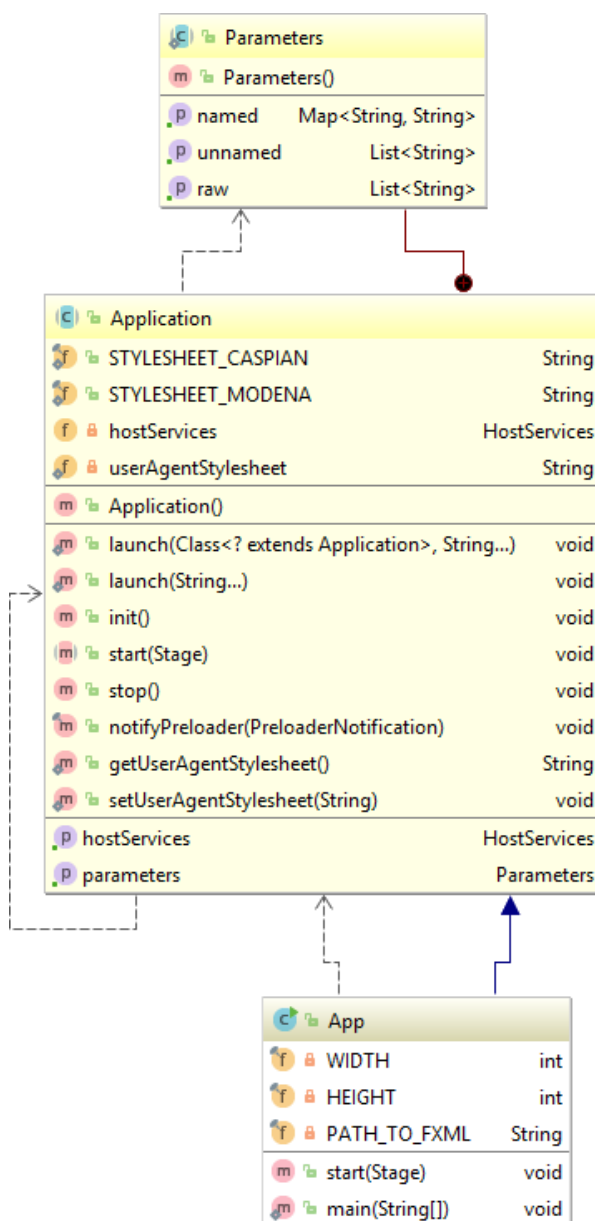


Рисунок 2.1 — UML-диаграмма класса `App`

Благодаря паттерну `dependency injection` (внедрение зависимостей), класс `App` не связана ни с каким другим классом, а он лишь реализует интерфейс `Application` и его метод `public void start(Stage primaryStage)`, который позволяет запустить графический интерфейс и контроллеры, обрабатывающие его.

2.2 Реализация метода генерации фрактального временного ряда

Для генерации фрактального временного ряда был выбран метод смещения средней точки. Математическая реализация была описана в главе 1.5.1. Алгоритм программной реализации представлен на рисунке 2.2.

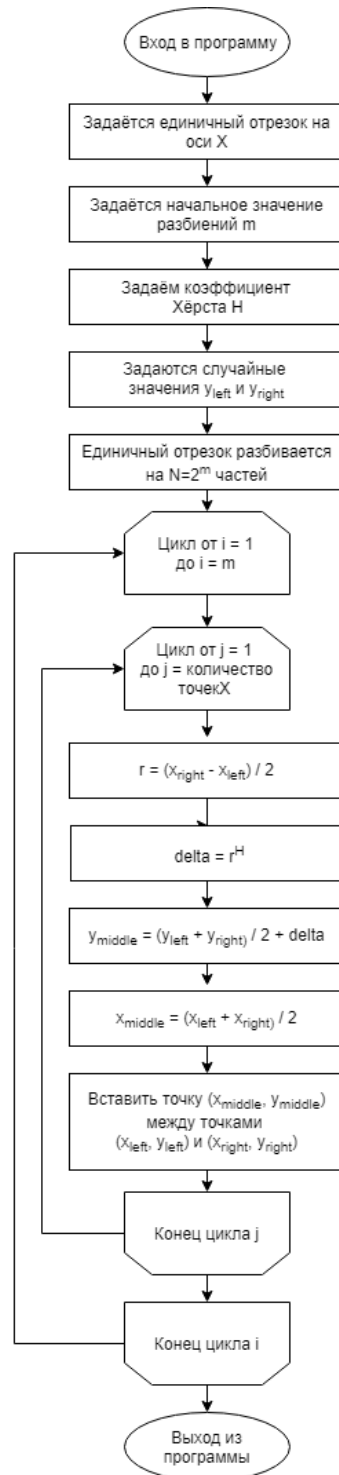


Рисунок 2.2 — Блок-схема реализации метода смещения средней точки

Для реализации данного метода был реализован класс *GenerateGraphModel*, UML-диаграмма которого представлена на рисунке 2.3.

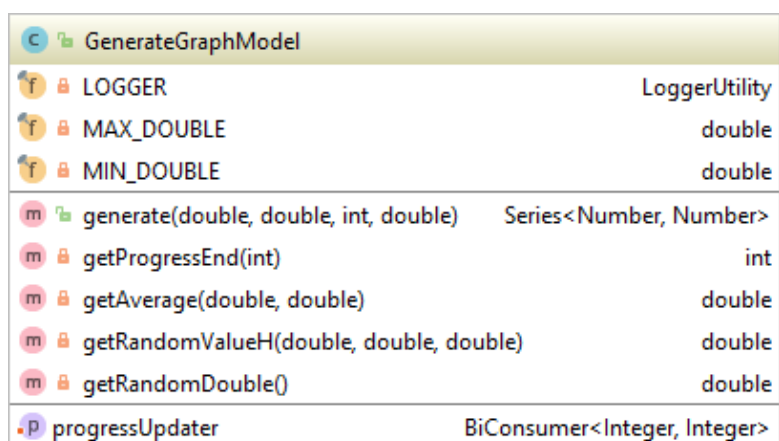


Рисунок 2.3 — UML-диаграмма класса *GenerateGraphModel*

Данный класс имеет три поля: *LOGGER* реализует класс *LoggerUtility*, который позволяет логгировать все методы класса; *MAX_DOUBLE* и *MIN_DOUBLE* имеют тип *double* и являются верхней и нижней границей случайных значений по оси ординат *Y*.

Класс имеет параметр *progressUpdater*, который позволяет обновлять полосу прогресса выполнения алгоритма.

Так же класс реализует четыре приватных и один публичный методы:

Метод *setProgressUpdater* является «сеттером» и служит для передачи значений для *progressUpdater*.

Метод *getProgressEnd* получает на вход целое число *m* типа *int*, которое является степенью двойки. Данный метод рассчитывает количество атомарных операций для выполнения генерации фрактального временного ряда и возвращает целое число типа *int* — количество таких операций.

Метод *getAverage* получает два числа типа *double* — *leftValue* и *rightValue*. Он рассчитывает среднее арифметическое между двумя входными параметрами и возвращает результат в виде числа типа *double*.

Метод *getRandomDouble* возвращает случайное число типа *double* в диапазоне [*MIN_DOUBLE*; *MAX_DOUBLE*].

Метод *getRandomValueH* принимает на вход два числа типа *double* — *xLeft* и *xRight*, которые являются правой и левой границей очередного рассматриваемого отрезка $[x_{left}, x_{right}]$, а так же параметр типа *doubleH*, который является ожидаемой фрактальной размерностью моделируемого графика. Сам же метод вычисляет значение y_{middle} по формуле 14.

Публичный метод *generate* получает на вход начальные параметры: *left* и *right* типа *double*, которые являются левой и правой границей единичного отрезка соответственно; параметр *numDiv* типа *int*, отвечающий за количество делений единичного отрезка; значение *H* типа *double*, которое является ожидаемым показателем Хёрста моделируемого фрактального временного ряда.

На выходе метод возвращает координаты точек в виде класса-коллекции *XYChart.Series*.

Сама работа метода происходит следующим образом:

1. Создаётся коллекция точек фрактального ряда типа *XYChart.Series* и в нём создаются две пары значений.

2. В первую пару помещается значение переменной *left* и случайное число типа *double*, генерируемое функцией *getRandomDouble*, а во вторую значение переменной *right* и, так же как для первой пары, случайное число типа *double*.

3. Начинается цикл от нуля до значения переменной *numDiv*. На каждой итерации производится обход массива начиная с первого до предпоследнего элемента ($series.length - 1$). На каждом шаге итерации вложенного цикла создаётся новая точка со средним значением x_{middle} , рассчитанное при помощи функции *getAverage* для двух значений x_i и x_{i+1} . Второе значение точки задаётся как среднее значение двух элементов массива y_i и y_{i+1} через функцию *getAverage*, умноженное на значение функции *getRandomValueH*. Полученная точка помещается между двумя элементами массива с индексами i и $i + 1$ так, чтобы длина массива увеличилась на единицу.

4. После завершения обхода массива значение счётчика первого цикла увеличивается на единицу. Далее происходит деление ещё раз, как описано в пункте 3, вплоть до достижения счётчика цикла значения равное $numDiv$. Таким образом, итоговая коллекция включает в себя 2^{numDiv} точек.

5. После выполнения всех предыдущих пунктов итоговый массив точек передаётся в слой *controller*, откуда передаётся в слой *view*.

Результаты работы класса *GenerateGraphModel* представлены в виде графиков фрактальных временных кривых на рисунках 2.4 — 2.6.



Рисунок 2.4 — Результат работы программы при $x_{left} = 0$, $x_{right} = 10$,
 $H = 0.75$, $numDiv = 10$



Рисунок 2.5 — Результат работы программы при $x_{left} = -10$, $x_{right} = 10$,
 $H = 0.15$, $numDiv = 13$



Рисунок 2.6 — Результат работы программы при $x_{left} = -10$, $x_{right} = 0$,
 $H = 0.5$, $numDiv = 15$

2.3 Реализация метода анализа фрактальной размерности

В качестве метода расчёта фрактальной размерности был выбран метод Минковского из-за высокой точности расчёта и гибкости реализации. Описание алгоритма приведено в главе 1.6.3.

Для реализации данного метода необходимо реализовать следующие программные модули:

- Создание битовой сетки;
- Сумматор битовой сетки;
- Интерполирование значений для поиска углового коэффициента;
- Функция построения графика.

Для реализации алгоритма Минковского был создан класс *MinkovskyAnalyzer*. UML-диаграмма данного класса, его поля и методы представлены на рисунке 2.7.

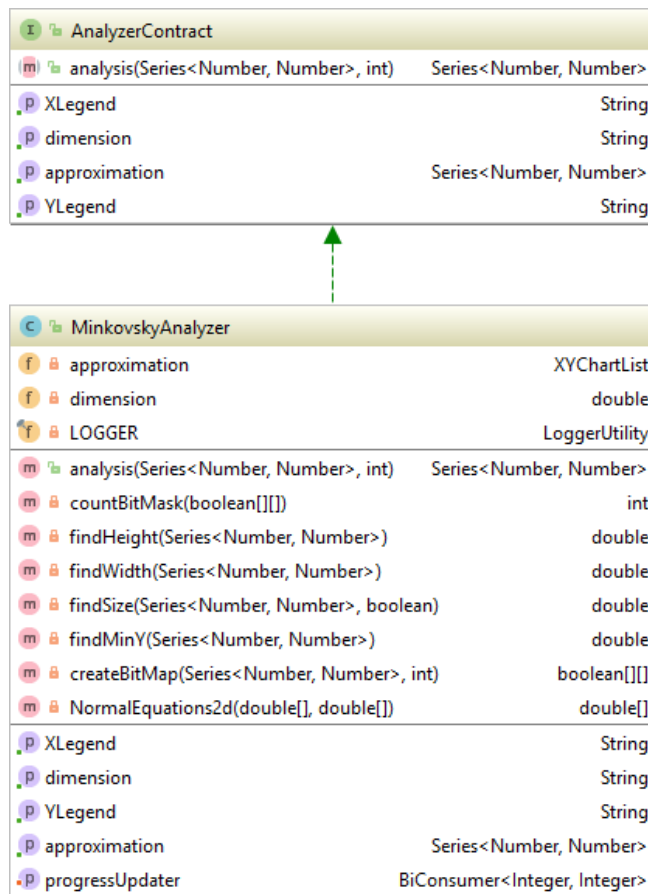


Рисунок 2.7 — UML-диаграмма класса *MinkovskyAnalyzer*

Данный класс реализует интерфейс *AnalyzerContract*, который содержит следующие методы:

- *XLegend* для отображения легенды по оси абсцисс на графике;
- *YLegend* для отображения легенды по оси ординат на графике;
- *dimension* для отображения рассчитанной фрактальной размерности и показателя Хёрста;
- *progressUpdater* для отображения процента выполнения программы в статус баре графического интерфейса;
- *approximation* для вывода на график интерполирующей прямой.

Сам класс имеет следующие поля: *LOGGER* для логирования методов, являющийся объектом класса *LoggerUtility*; параметр *dimension* типа *double*, который хранит в себе значение рассчитанной фрактальной размерности;

коллекция *XYChartList* точек *approximation* для хранения точек интерполирующей функции.

Помимо вышеперечисленных методов, существуют дополнительные приватные методы для логического разделения кода:

- *findSize* — метод для поиска расстояния между двумя точками;
- *findWidth* — метод поиска расстояние между двумя точками на оси абсцисс X , вызывающий в своём теле метод *findSize*;
- *findHeight*, как и метод *findWidth*, основывается на методе *findSize*, только расстояние находится между точками на оси ординат;
- *findMinY* — метод поиска наименьшего значения Y всех точек фрактального временного множества.

Далее опишем основные функции, используемые для анализа фрактальной размерности временного ряда:

Первый метод *createBitMap*, созданный для создания битовой сетки фрактального множества точек. На вход поступают два параметра — коллекция *XYChart* всех точек фрактального временного ряда; *blockSize* типа *int*, который указывает количество используемых клеток по оси абсцисс.

Сам же метод создаёт сетку из квадратов размером *blockSize* и «закрашивает» (устанавливает не нулевое значение) в тех точках, где находится хотя бы одна точка фрактального временного ряда. Для удобства реализации был использован двумерный массив (матрица) типа *boolean*.

Блок-схема данного алгоритма представлена на рисунке 2.8.

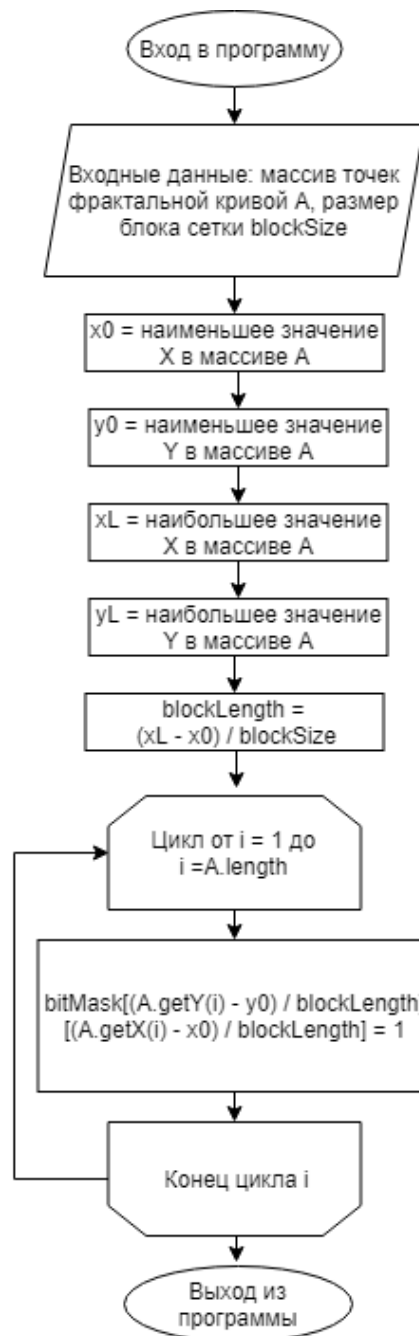


Рисунок 2.8 — Блок-схема метода *createBitMap*

Метод *countBitMask* принимает на вход двумерный массив типа *boolean* (битовая сетка, сгенерированная методом *createBitMap*). Далее он совершает обход всего массива и считает количество точек, имеющих значение ячейки равное *true*.

Блок-схема данного метода представлена на рисунке 2.9.

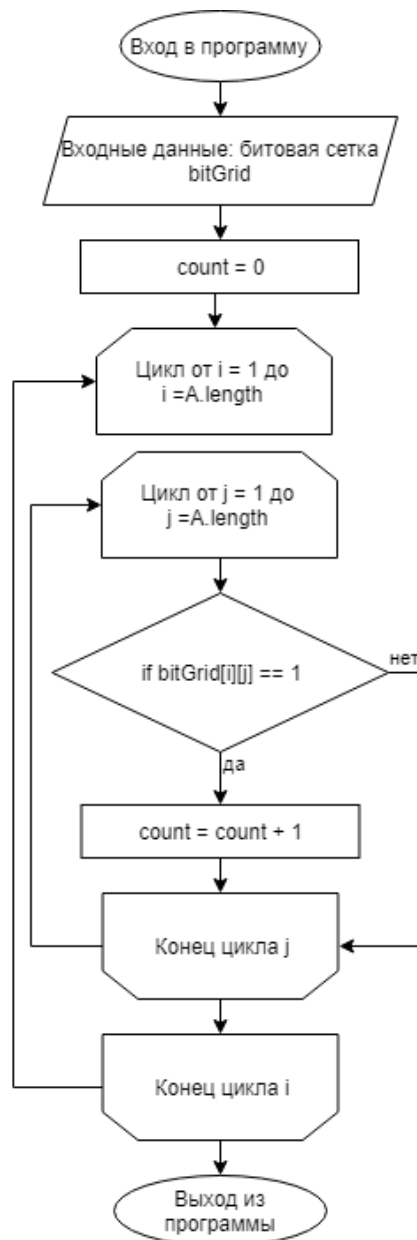


Рисунок 2.9 — Блок-схема метода *countBitMask*

Для интерполяции полученных значений в ходе выполнения метода Минковского, реализован метод *NormalEquations2d*. Данный метод занимается решением матрицы, описанной в формуле 27.

Блок-схема алгоритма наименьших квадратов, реализованная в данном методе, представлена в приложении В.

Метод *analysis* объединяет все подпрограммы данного класса в алгоритм поиска фрактальной размерности методом Минковского. Данный метод

принимает на вход параметры *maxSquares* — количество квадратов по одной из граней сетки, и коллекцию точек *series*.

Блок-схема алгоритма представлена на рисунке 2.10.

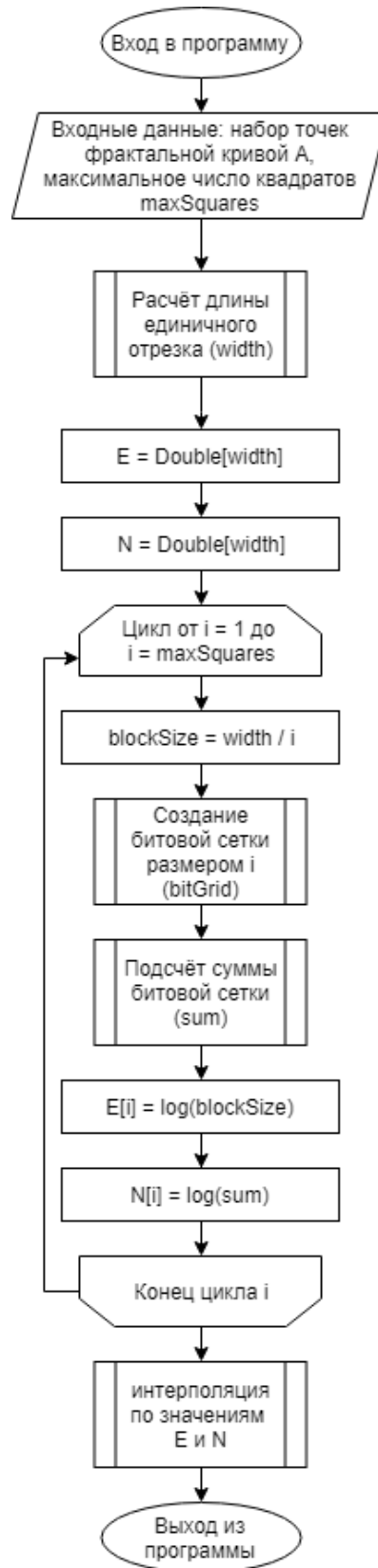


Рисунок 2.10 — Блок-схема метода *analysis*

Результатом работы алгоритма является график точек логарифмических отношений количества квадратов размера $blockSize$, полностью покрывающих фрактальную кривую, и размера единичного блока $blockSize$, а так же прямой интерполирующей данные точки. Угол наклона прямой будет считаться фрактальной размерностью исследуемой фрактальной кривой. Показатель Хёрста можно вычислить из отношения $D = 2 - H$.

Результирующий график продемонстрирован на рисунке 2.11.

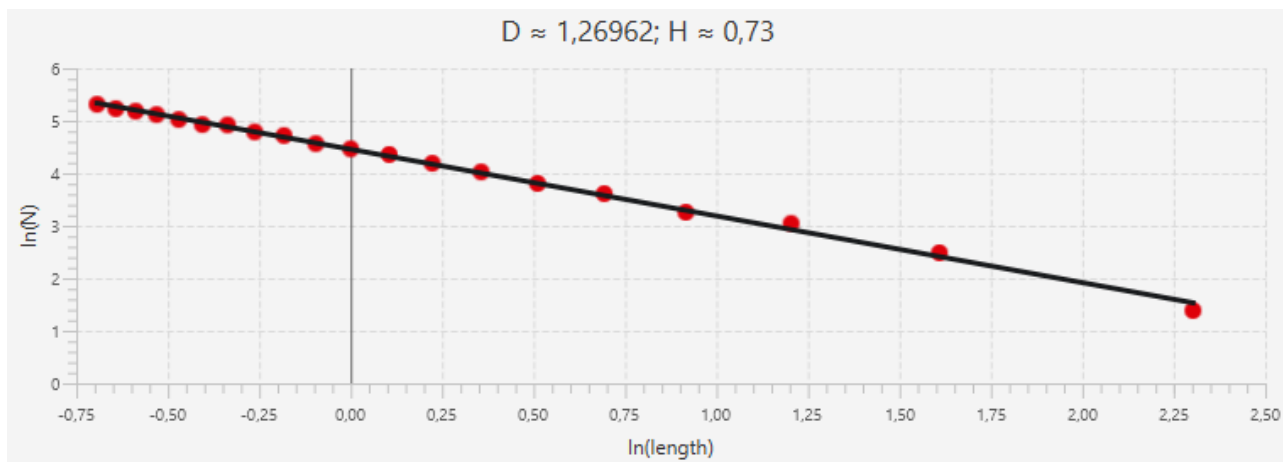


Рисунок 2.11 — Результат работы класса *MinkovskyAnalyzer* при $x_{left} = 0$, $x_{right} = 10$, $H = 0.75$, $numDiv = 10$, $maxSquares = 20$

2.4 Интерфейс приложения

Как уже говорилось, программа имеет графический интерфейс, реализованный при помощи технологии JavaFX.

Интерфейс состоит из одного главного окна, двух графиков типа *lineChart*, пять полей ввода данных, две кнопки выбора входных данных и две кнопки подтверждения.

Работа графического интерфейса работает через класс-контроллер *MainWindowController*.

Само окно и компоненты на нём пронумерованы и продемонстрированы на рисунке 2.12.



Рисунок 2.12 — Графический интерфейс пользователя с пронумерованными компонентами

Графический интерфейс состоит из следующих компонентов:

1. График вывода фрактального временного ряда. Данный компонент через класс-контроллер получает данные из классов *GenerateGraphModel* или *GeneratorDataFromFile*.

2. График интерполяции. Данный график получает данные через класс-контроллер из метода *MinkovskyAnalyzer*.

3. Компоненты *TextField* служат для ввода данных. Каждое поле, прежде чем попадают в слой *model* проходят проверку при помощи класса *Validator*.

Для отлова соответствующих ошибок были реализованы специальные классы наследующие класс *Exception*: *NotFileFoundException*, *SeriesNullPointerException*, *TextFieldNotValidateException*.

Для обработки ошибки ввода неверных значений вызывается класс-ошибка *TextFieldNotValidateException*.

4. Кнопки типа *RadioButton* нужны для того, чтобы можно было выбрать способ генерации: либо фрактальный временной ряд генерируется случайно, либо данные загружаются через файлы с расширением *xls*. В зависимости от выбранного поля используется класс *GenerateGraphModel*, либо *GeneratorDataFromFile*.

5. Кнопка «Выбрать файл» необходима для загрузки исходного файла для последующего построения фрактального ряда. Для того, чтобы файл был принят, необходимо чтобы *RadioButton* была установлена в соответствующее положение.

6. Компонент типа *Label* необходим для вывода сообщений возникших ошибок пользователю. Все сообщения хранятся в классе *ErrorMessage*, поля которого являются переменными типа *final String*.

7. Статус бар необходим для отображения хода выполнения генерации и анализа фрактального временного ряда в процентах.

8. Кнопка «Построить и проанализировать фрактальный временной ряд» выполняет генерацию ряда, в соответствии с выбранной кнопкой в пункте четыре, после чего анализирует его, а результаты выводятся в графики под номером один и два.

Для обработки взаимодействия пользователя с компонентами графического интерфейса был реализован класс *MainWindowController*. Данный класс имеет поля с аннотацией «@FXML», которые перенимают состояния соответствующих компонентов графического интерфейса.

Из особенностей данного класса стоит отметить, что для выполнения генерации фрактального временного ряда, чтения из файла и последующего анализа, был реализован отдельный поток, параллельный основному, который позволяет не останавливать программу во время расчётов.

Листинг всего кода представлен в приложении А.

ГЛАВА 3 ТЕСТИРОВАНИЕ ПРОГРАММНОЙ ЧАСТИ

Для тестирования полученной в ходе разработки программы, были использованы временные ряды реальных временных зависимостей.

Полученная программа не может точно предсказать будущие изменения некой системы, но может оценить имеющийся временной ряд и дать оценку фрактальной размерности в виде показателя Хёрста, что даст нам право предположить по последующем состоянии системы.

Все данные были взяты из официального портала центрального банка российской федерации, располагающийся по адресу www.cbr.ru.

В качестве первого теста были взяты данные учётных цен на золото в рублях. Диапазон рассмотрения выбран исходя из 02.07.2008 по 20.06.2019, то есть в целом одиннадцать лет. Разбиение происходит по дням, а значит исследуется около четырёх тысяч точек временного ряда.

На официальном сайте центрального банка российской федерации можно получить график изменения цены золота в рублях, поэтому для тестирования программы будем ориентироваться на него. Сам граф представлен на рисунке 3.1.



Рисунок 3.1 — График изменения отношения рубля к грамму золота от 02.07.2008 по 20.06.2019

Данные были организованы в один файл с расширением *xls* (электронная таблица MSExcel) на официальном сайте центрального банка российской федерации и были переданы в программу как исходные данные.

Загруженные данные сразу были проанализированы на фрактальную размерность методом Минковского при максимальном делении сетки на десять квадратов. Результат выполнения данной операции продемонстрирован на рисунке 3.2.

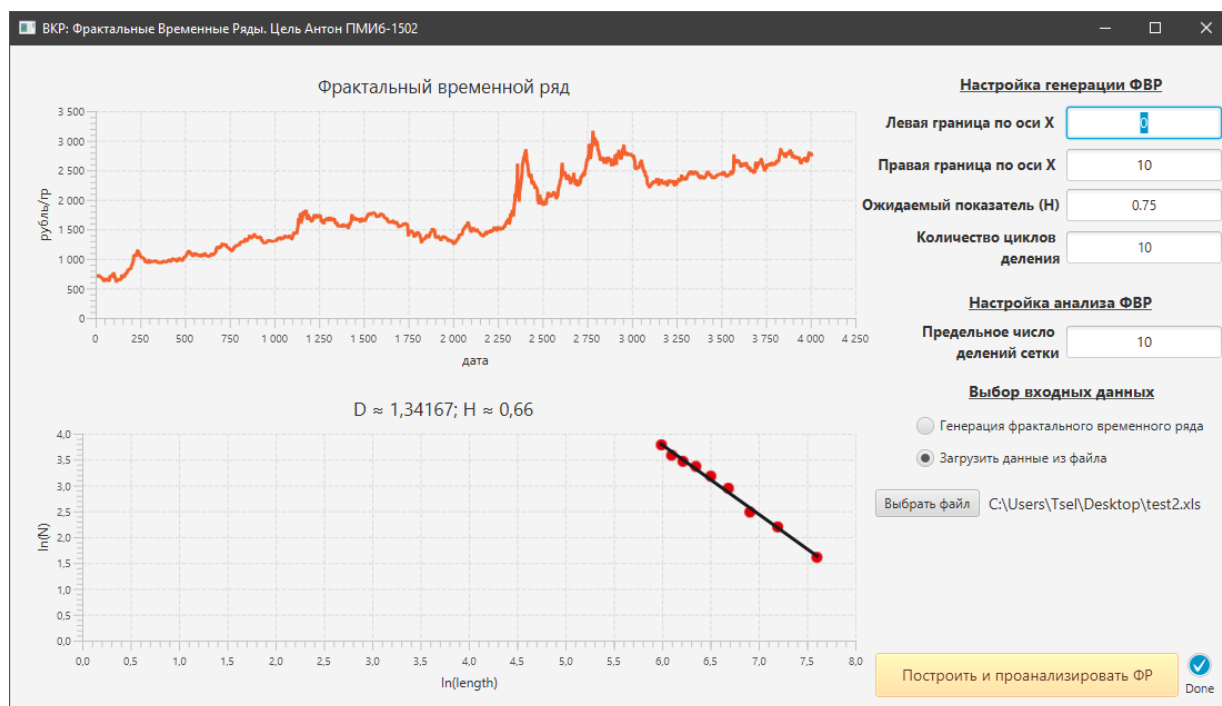


Рисунок 3.2 — Выполнение программы с загруженными данными стоимости грамма золота в рублях за период от 02.07.2008 по 20.06.2019

Как сразу можно заметить из рисунка 3.2 временной ряд, построенный программой и сгенерированный порталом центрального банка (рисунок 3.1) точно сходятся.

На нижнем графике рисунка 3.2, интерполирующая функция выглядит достаточно малой, потому что было использовано всего десять делений.

Показатель фрактальной размерности $D \approx 1.34167$, а показатель Хёрста $H \approx 0.66$. Из главы 1.3 данной выпускной квалификационной работы можно сделать вывод, что фрактальный ряд будет трендоустойчивым из-за того, что

$0.5 < H \leq 1$. Но пока нельзя утверждать трендоустойчивость фрактального временного ряда, так как была выбрана слишком малое значение предельного числа деления сетки. Поэтому повторим анализ данного ряда при предельных значениях делений сетки равным ста, двумстам и трёмстам (рисунки 3.3, 3.4 и 3.5 соответственно).

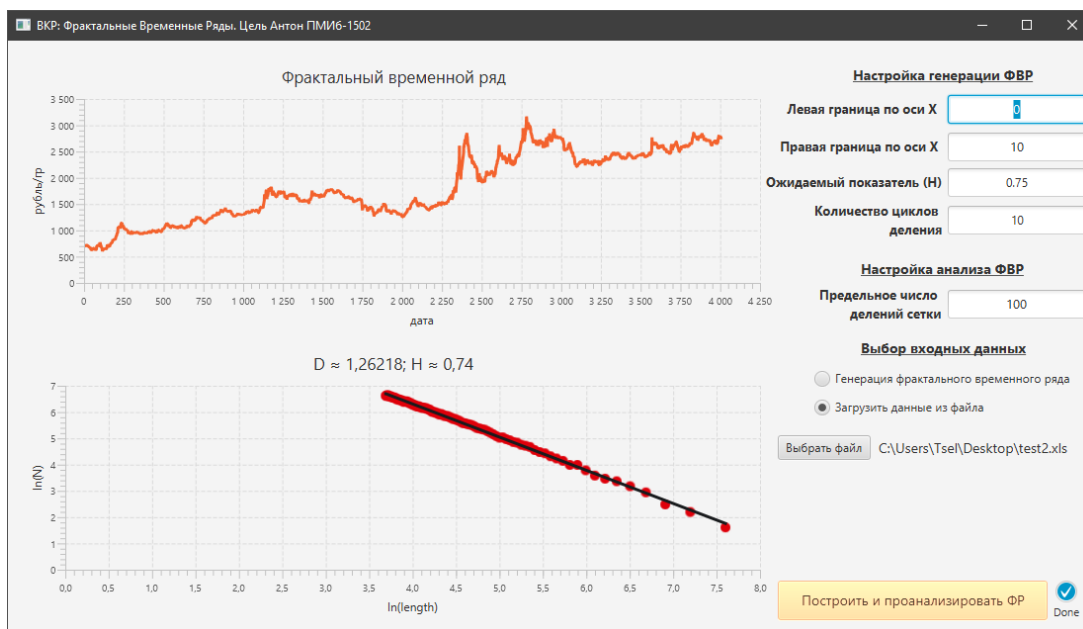


Рисунок 3.3 — Анализ данных стоимости грамма золота в рублях за период от 02.07.2008 по 20.06.2019 с предельным числом делений сетки равным 100

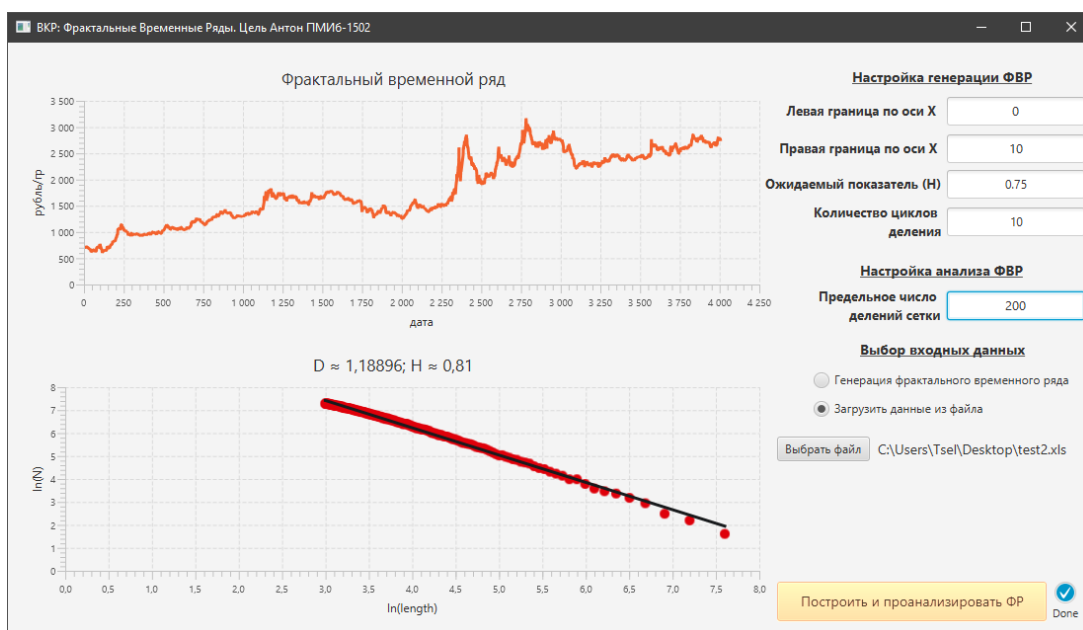


Рисунок 3.4 — Анализ данных стоимости грамма золота в рублях за период от 02.07.2008 по 20.06.2019 с предельным числом делений сетки равным 200

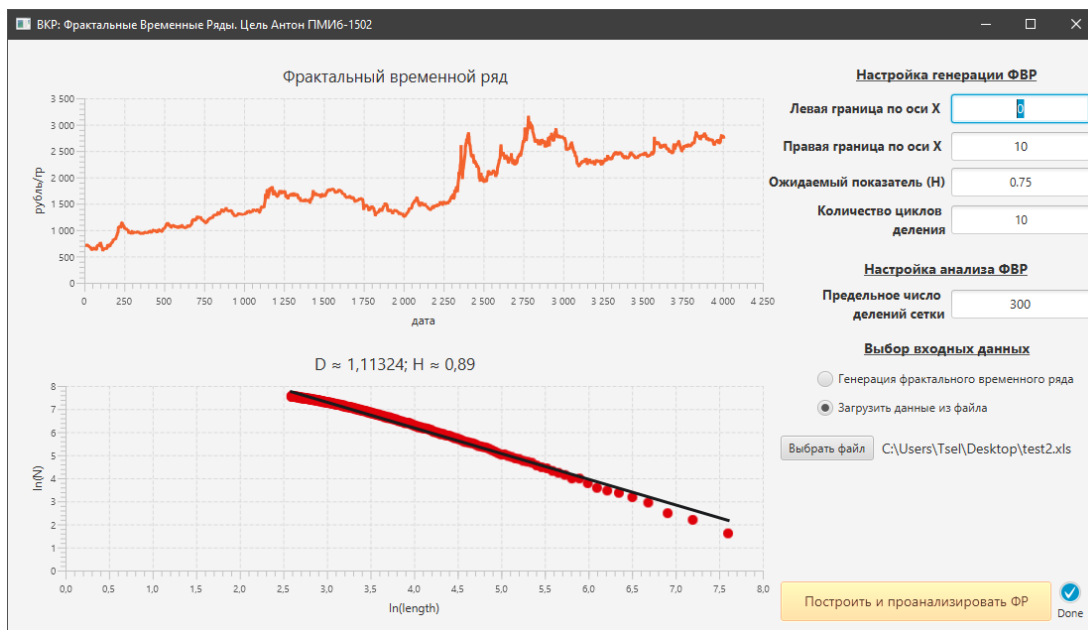


Рисунок 3.5 — Анализ данных стоимости грамма золота в рублях за период от 02.07.2008 по 20.06.2019 с предельным числом делений сетки равным 300

Из рисунков 3.3-3.5 видно, что при увеличении предельного числа делений сетки в методе Минковского для анализа данных стоимости грамма золота в рублях, значение показателя Хёрста H стремится к единице ($H \approx 0.74$, $H \approx 0.81$, $H \approx 0.89$). Это обозначает, что данный временной ряд с большой долей вероятности является трендоустойчивым, а значит, как это было уже рассказано в главе 1.3, временной ряд стремиться быть бесконечным линейным трендом.

Следующими данными были рассмотрены стоимость доллара США в российских рублях за период от 01.07.1999 года по 20.06.2019. Период, рассматриваемый в данном наборе данных, примерно равен двадцати годам, что соответствует около 7300 рассматриваемых точек.

Все данные, как и в предыдущем примере, были взяты с официального сайта центрального банка российской федерации.

Для последующего фрактального анализа загрузим данные в программе в формате *xls* и сделаем выборку предельного числа делений сетки равными 100, 300 и 500 (рисунки 3.6, 3.7 и 3.8 соответственно).

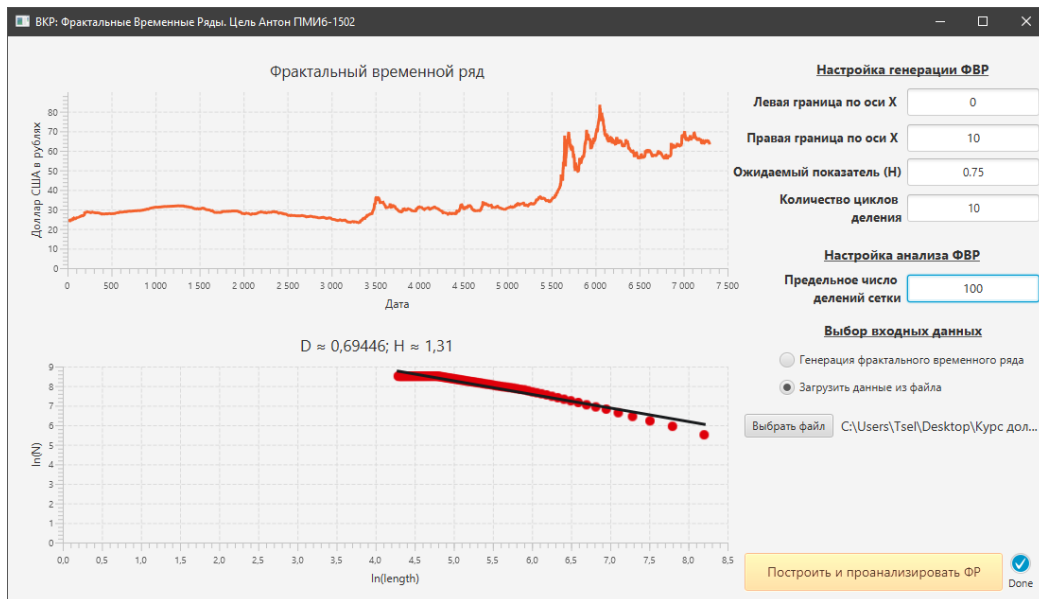


Рисунок 3.6 — Анализ курса долларов США в рублях за период от 01.07.1999 по 20.06.2019 с предельным числом делений сетки равным 100

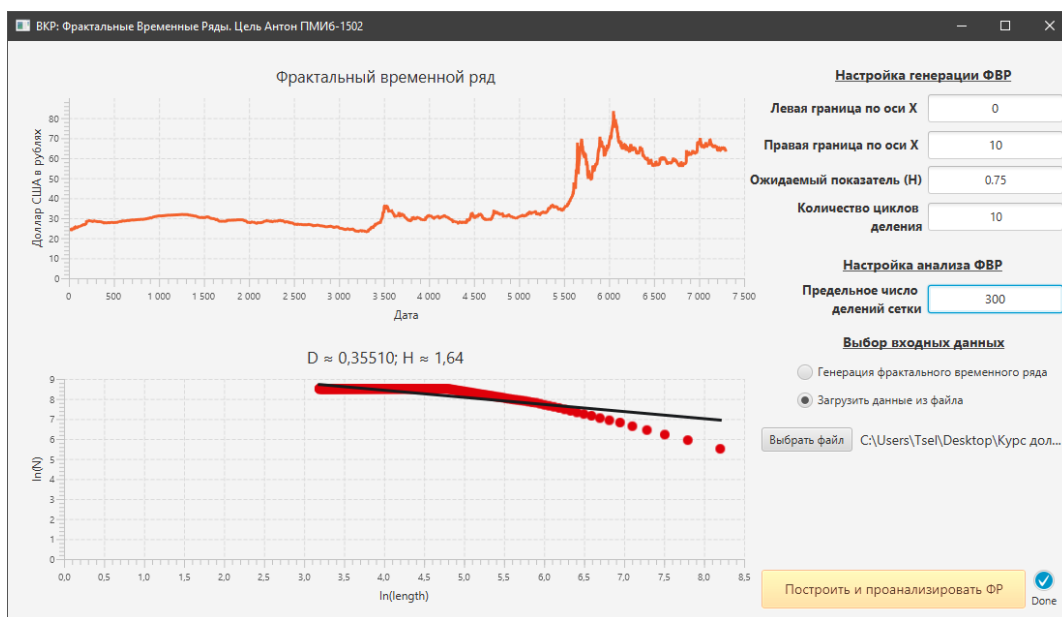


Рисунок 3.7 — Анализ курса долларов США в рублях за период от 01.07.1999 по 20.06.2019 с предельным числом делений сетки равным 200

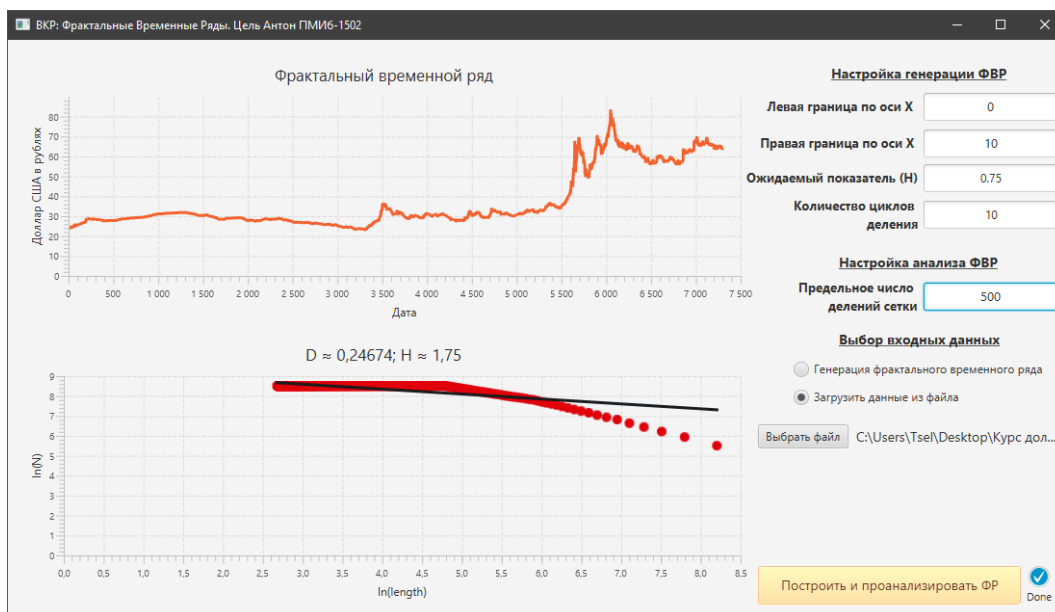


Рисунок 3.8 — Анализ курса долларов США в рублях за период от 01.07.1999 по 20.06.2019 с предельным числом делений сетки равным 300

Из результатов, представленных на рисунках 3.6, 3.7 и 3.8, можно легко заметить, что фрактальная размерность D стремится к нулю, а, следовательно, и показатель Хёрста H стремится к двойке исходя из отношения $D = 2 - H$.

Так как одним из главных условий фрактальности множества, описанных в главе 1.1 и 1.2, фрактальная размерность D должна быть больше, чем топологическая размерность. Для линии топологическая размерность равна единицы. А это означает, что данный временной ряд не является фракталом, так как не обладает свойствами фрактальных множеств.

ЗАКЛЮЧЕНИЕ

Цель работы состояла в разработке компьютерной реализации фрактального временного ряда.

Для этого были решены следующие задачи:

1. Выявлены основные алгоритмы генерации фрактальных временных рядов.

2. Была разработана структура программы, а затем по ней и сама программа, реализующая генерацию фрактальных временных рядов, возможность ввода фрактального ряда, а также анализа размерности полученного ряда.

3. Проведено тестирование реального временного ряда на наличие фрактальных свойств.

Были пошагово реализованы метод генерации фрактального временного ряда методом смещения средней точки, а также метод анализа фрактальной размерности методом Минковского на языке объектно-ориентированном языке программирования Java, с использованием пользовательского графического интерфейса.

Проведены вычислительные эксперименты, состоящие из вычисления фрактальной размерности реально существующих временных рядов с целью определения свойств фрактальности временного ряда и анализа фрактальной размерности.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

Научная и методическая литература

1. Андриенко В.М. Фрактальные свойства фондовых рынков / В.М. Андриенко, О.В. Варчук // Одесский национальный политехнический университет. – Одесса, Украина. – 2010, 4 с.
2. Барабаш Т.К. Компьютерное моделирование фрактальных временных рядов / Т.К. Барабаш, А.Г. Масловская // Вестник Амурского государственного университета, Благовещенск. – 2010. – №49. – с. 37 – 38.
3. Божокин, С.В. Фракталы и мультифракталы / С.В. Божокин, Д.А. Паршин. – Ижевск: НИЦ «Регулярная и хаотическая динамика», 2001. – 126 – 128 с.
4. Борисов, В.Д. Метод фрактального анализа временных рядов. / В.Д. Борисов, Г.С. Садовой//Автометрия, 2000, т.6, С. 1019.
5. Боровской И.Г. Исследование прогностной способности показателя Херста применительно к российскому фондовому рынку / И.Г. Боровской, А.О. Жучков // Доклады ТУСУР. – 2017, №2. – 3 с.
6. Зельдович Я.Б. Фракталы. Подобие, промежуточная асимптотика / Я.Б. Зельдович, Д.Д. Соколов // УФН, 1985, т.146, вып.3, стр. 493-506.
7. Кроновер Р.М. Фракталы и хаос в динамических системах. Основы теории / Р.М. Кроновер // М.: Постмаркет, 2000. — 123 – 142 с.
8. Мандельброт Б. Фракталы, случай и финансы // М.: Регулярная и хаотическая динамика, 2004.
9. Мандельброт Б. Фрактальная геометрия природы. – М.: Институт компьютерных исследований, 2002. – 656 с.
10. Мандельброт Б. Самоаффинные фрактальные множества, «Фракталы в физике». – М.: Мир, 1988, 672 с.
11. Масловская А.Г. Применение фрактальных методов для анализа динамических данных / А.Г. Масловская, Т.Р. Осокина, Т.К. Барабаш // Вестник

Амурского государственного университета, Благовещенск. – 2010. – №51. – с. 13 – 20.

12. Некрасова И.В. Показатель Херста как мера фрактальной структуры и долгосрочной памяти финансовых рынков // Международный научно-исследовательский журнал. - №7(38). – 2015, с. 87 – 91.

13. Петерс Э. Фрактальный анализ финансовых рынков. Применение теории хаоса в инвестициях и экономике // М.: Интернет-трейдинг, 2004. — 304 с.

14. Старченко Н.В. Локальный анализ хаотических временных рядов с помощью индекса фрактальности // Автореферат диссертации на соискание ученой степени кандидата физико-математических наук, Москва, 2005.

15. Федер Е. Фракталы // М.: Мир, 1991. – 262 с

16. Шайдук А.М. Влияние фрактальной размерности сигнала на распределение энергии в его спектре / А.М. Шайдук, С.А. Останин // Журнал радиоэлектроники. - 2016. - №2.

Иностранная литература

17. Feder J. Fractals. // New York: Plenum Press, 1988.

18. Provata A. Abstract Phase-space Networks Describing Reactive Dynamics / A. Provata, E.Panagakou // National Center for Scientific Research «Demokritos». – Greece. – 2018, 10 с.

19. Theiler J. Estimation fractal dimension // LiLincoln Laboratory, Massachusetts Institute of Technology, Lexington, Massachusetts. – Vol. 7. - №6. – 1990, 19 с.

Электронные ресурсы

20. Вычисление фрактальной размерности Минковского для плоского изображения [Электронный ресурс:] - Режим доступа: <https://habr.com/ru/post/208368/>

Программная реализация прикреплена на диске.

