

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ  
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»



Направление подготовки/специальность  
09.03.01 Информатика и вычислительная техника  
направленность (профиль)/специализация  
«Технологии разработки программного обеспечения»

**Выпускная квалификационная работа**  
«Разработка веб-приложения для изучения фракталов»

Обучающегося 4 курса

Очной формы обучения

Ал-Обайди Лина Моханадовна

Руководитель выпускной квалификационной работы:

Власова Елена Зотиковна, кандидат пед. наук, доцент

Санкт-Петербург, 2025 год

## Оглавление

Введение.....	3
Теоретическая часть.....	5
История и теоретические основы возникновения фракталов .....	5
Определение фракталов.....	7
Понятие фрактальная графика .....	10
Практическая часть .....	18
Постановка задачи и цели разработки .....	18
Выбор технологий и обоснование .....	18
Архитектура проекта и структура сайта .....	22
Этапы разработки сайта .....	24
Хостинг и публикация .....	27
Тестирование и отладка .....	<b>Error! Bookmark not defined.</b>
Оценка результата и возможности развития .....	27
Литература .....	31

## Введение

В современном мире для изучения чего-либо, используют веб-ресурсы: приложения, сайты, электронные справочники. Сейчас, все меньше применяют физические носители информации, так как они не всегда находятся в шаговой доступности, а интернетом пользуется почти каждый человек. Поэтому для работы было выбрано создание веб-сайта. Для изучения была взята фрактальная графика. Она сочетает в себе математику, искусство и программирование. Фракталы предоставляют уникальные возможности и могут быть использованы в разных сферах жизни. Фрактальная графика уникальна предоставляемыми возможностями: они визуально притягательны и часто используются в моделировании природных форм.

Но для изучения ее, на просторах интернета предоставлено не так много ресурсов. Большинство веб-сервисов хранят устаревшие данные или предоставляют ограниченную информацию. Поэтому, чтобы изучить фракталы, пользователю придется потратить много времени для понимания, как пользоваться и для чего они нужны. Решено создать удобный и адаптивный сайт по изучению фрактальной графики. Это является актуальной задачей.

**Актуальность** работы связана с малой доступностью информации к фракталам.

**Предметом исследования** является проектирование и создания сайта для изучения фракталов.

**Теоретическая значимость** заключается в предоставлении знаний о фрактальной графике. Сбор информации и систематизировании ее.

**Целью** выпускной квалификационной работы стала разработка удобного веб-приложения по изучению фракталов.

Для достижения поставленной цели были поставлены следующие задачи:

Проанализировать фрактальную графику, узнать какие бывают виды, для чего они могут быть использованы, ознакомиться с историей создания. Выделить плюсы и минусы использования того или иного вида.

- Выбрать ресурс для создания веб-сайта. Расписать, почему был выбран именно он.
- Спроектировать и разработать сайт.
- Протестировать, оценить, что получилось реализовать и написать заключение о проделанной работе.

Результатом бакалаврской выпускной квалификационной работы, является готовый сайт к использованию и изучению фракталов.

Структура бакалаврской выпускной квалификационной работы. Выпускная квалификационная работа состоит из введения, двух глав, заключения, списка литературы и приложения. Она содержит 35 страниц, 20 изображений и 10 таблицы.

## Теоретическая часть

### История и теоретические основы возникновения фракталов

Термин «фрактал» был введен в 1975 году Бенуа Мандельброта, и был описан им примерно так: Фрактал- это структура, обладающая самоподобием и дробной размерностью. Это определение дает нам понятие того, что фракталы окружают нас в природе повсюду от коры деревьев до кровеносно-сосудистой системы, ведь основной принцип построения является рекурсия, когда одна и та же операция повторяется на каждом уровне масштаба. Стоит также отметить, что первые объекты, обладающие фрактальными свойствами, были описаны еще в XXв. Кантором, Кохом и Жюлиа.

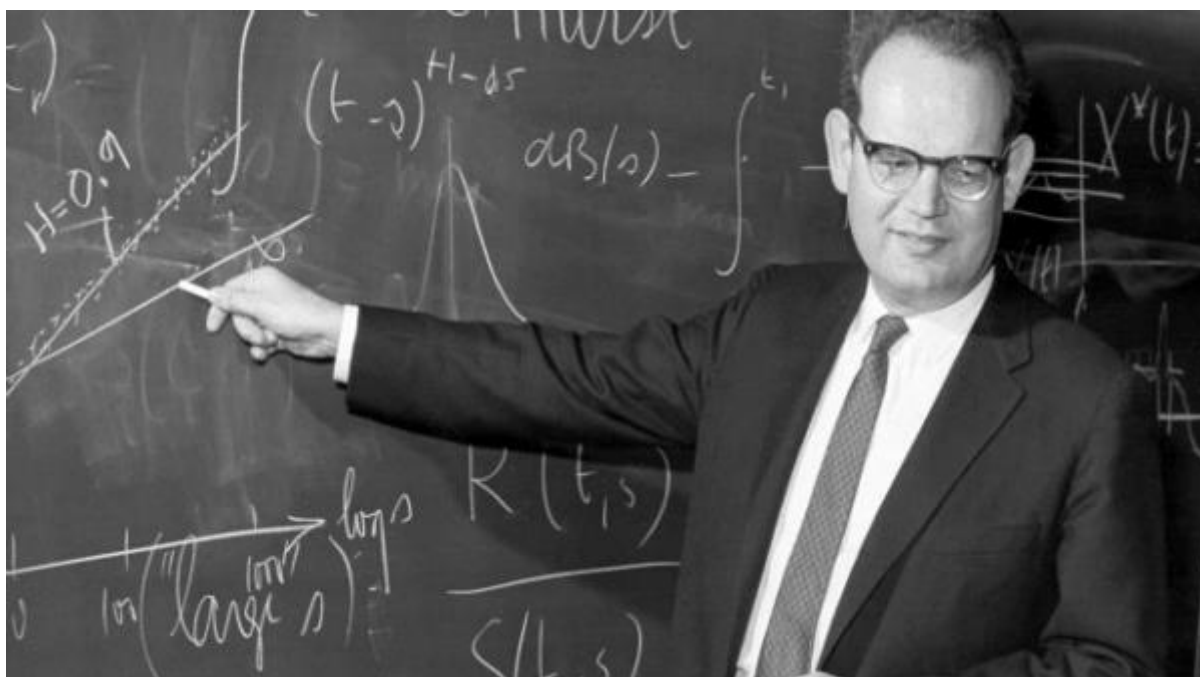


Рисунок 1. Бенуа Мандельброт

Одна из причин, почему не могли раньше выделить в том, что развиваться фракталы могли только тогда, когда в обиход вошли первые электронные вычислительные устройства. В то время Бенуа работал в исследовательском центре компании IBM. Тогда он и несколько других сотрудников трудились над передачей данных на расстояние. В тот момент возникла проблема больших потерь, возникающих из-за шумовых помех. Перед Бенуа стояла важная и непростая задача

— выяснить, каким образом можно предсказать появление шумовых помех в электронных схемах в тех случаях, когда традиционные статистические методы не давали точных результатов. Наблюдая за поведением шумов он заметил, что помехи в электронных сигналах не являются полностью случайными. В них присутствовали самоподобные структуры, а именно фрагменты, которые повторяются на разных масштабах. Стоит отметить, что Бенуа Мандельброт при жизни признался, что не занимается формулами, а переносит все в визуальную модель в голове. Именно благодаря богатому пространственному воображению, он смог соединить звуковые волны и фрактальную графику. Он предположил, что классическая математика не подходит для обычного решения и поэтому ввел геометрические фигуры, которые и назвал фракталами. Они повторяются при увеличении масштаба, описывают сложные формы природы и данные, включая шумы. Таким образом, Бенуа смог смоделировать поведение сигналов и помех, оценить степень их сложности и предсказать возникновение шумов лучше, чем с помощью классических моделей.

Фрактальный рисунок не содержит одинаковых фрагментов, но обладает подобностью в любом масштабе. До создания ЭВМ построить изображение с высокой детализацией вручную было почти невозможно. Но, ученые, даже не смотря на сложность стремились приблизиться к пониманию сути фракталов. Так, французский математик Пьер Жозе Луи Фату, смог описать данное множество еще до десятилетия открытия Бенуа Мандельбротом. В частности, он исследовал поведение итерационных процессов, то есть последовательностей, в которых результат каждой итерации подставляется обратно в исходную функцию. Фату заложил фундамент для будущей визуализации фракталов, однако в его время не было технической возможности построить изображение этих множеств — для этого потребовались бы миллионы итераций, что стало возможным только с появлением компьютеров. Благодаря работам его и других ученых, Бенуа смог при помощи компьютера обработать последовательность, состоящую из большого количества значений, перенести результаты на график (Рис 2).

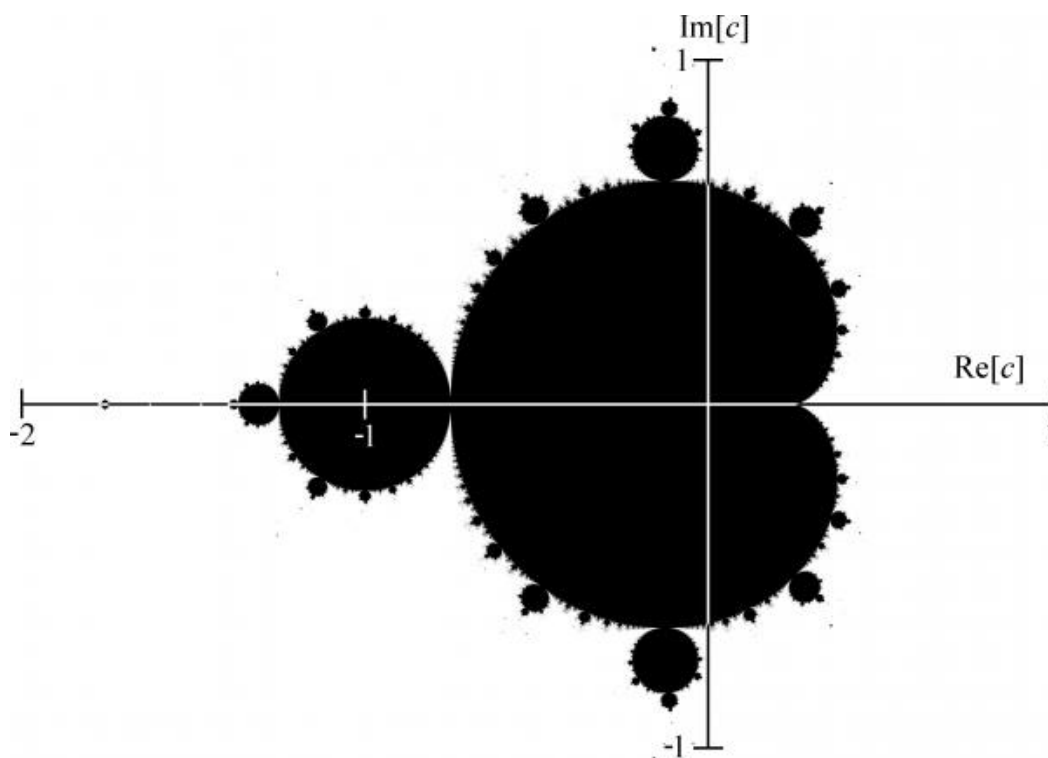


Рисунок 2. Фрактал Мандельброта

### Определение фракталов

До того, как начать изучать фрактальную графику, стоит глубже рассмотреть такое понятие, как фрактал. Ключевые свойства фрактала это: самоподобие, дробная размерность, итеративность, бесконечная детализация и нерегулярность. Разберем каждый пункт отдельно:

1) Самоподобие. Одно из основных черт фракталов. Оно означает, что объект состоит из частей, каждая из которых подобна всей структуре целиком. Самоподобие может быть:

- Точным, когда фрагменты являются уменьшенными копиями объекта;
- Статическим, потому что части похожи на целое по статистическим характеристикам (например, береговая линия или облака);
- Квази – самоподобным означает, что структура сохраняет общую форму, но фрагменты отличаются по виду (например дерево или сосудистая система человека).

- 2) Дробная размерность. Дробная размерность показывает, насколько "сложным" и "заполненным" является фрактал в пространстве, и при этом может быть нецелым числом (отсюда и название — *дробная*). Например, обычная размерность точки - 0, а куба -3. Но фракталы, благодаря своей *самоподобной* и *сложной* структуре, занимают промежуточное положение.

Например:

- Кривая Коха: выглядит как линия, но её размерность  $\approx 1.26$
  - Множество Мандельброта: размерность  $\approx 2$
  - Ковёр Серпинского: размерность  $\approx 1.89$
- 3) Итеративность. Данный процесс означает, что фрактал создаётся пошагово, через итерации, где каждая новая итерация добавляет детали и усложняет структуру фигуры.
- 4) Бесконечная детализация. По названию становится понятно, что фрактал может быть бесконечным и при увеличении можно увидеть новые детали.
- 5) Нерегулярность. Означает, что фракталы не могут приравняться к обычным геометрическим правилам и имеют разную структуру.

Фракталы классифицируются по способу их построения и происхождения. Основные категории геометрические, итерационные (алгебраические), стохастические и естественные (натуральные) фракталы.

### 1) Геометрические фракталы

Эти фракталы строятся по чётким геометрическим правилам, благодаря чему они имеют точное самоподобие и создаются с помощью пошаговой (итеративной) замены частей фигуры.

Примеры:

- Кривая Коха — строится из треугольников, итеративно добавляемых к каждой стороне.
- Треугольник Серпинского — создаётся удалением центральной части из треугольника, оставляя три равных фрагмента.



- Ковёр Серпинского – аналогичен, но используется квадрат вместо треугольника.

Особенности геометрических фракталов это: чёткие формулы построения, высокая степень самоподобия и легкость для математического описания.

## 2) Итерационные или алгебраические фракталы

Алгебраические фракталы строятся в результате повторений математических операций и исходя из этого, можно сделать вывод, что их можно превратить в геометрические, если построить графики на координатной плоскости.

Примеры данных фракталов:

- Множество Мандельброта – одно из самых известных фрактальных множеств, созданных с помощью итерации комплексных чисел.
- Множество Жюлиа – связано с множеством Мандельброта, но при фиксированных значениях параметров.
- Дракон Хартера–Хейтауэя – ещё один пример итерационного построения фрактала.

Одни из главных свойств состоят в том что они используют рекурсивные математические формулы, часто обладают квази-самоподобием и один из критериев является их красочность.

## 3) Стохастические фракталы

Стохастические фракталы (от греческого слова *στοχαστικός* — «предполагать», «угадывать») интересны тем, что строятся с элементами случайности. Данная способность делает их похожими на реальные объекты. Данный вид имеет статический характер, а не точный, поэтому их удобно использовать в моделировании природных условий, например, моделирование облаков, береговой линии или горных ландшафтов гор.

Можно выделить такие особенности:

- Используются генераторы случайных чисел.
- Применяются в компьютерной графике для создания реалистичных текстур и пейзажей.
- Обладают фрактальными свойствами, но выглядят как что-то из реального мира.

#### 4) Естественные фракталы

Самый наглядный класс фрактала, так как они являются объектами природы, которые естественным образом образуются в живой или неживой природе. Например, папоротники, деревья, сосудистая система человека, снежинки и многое другое. Их главное отличие, что природа сама может их создать, без помощи человека.

Структурирование фракталов помогает упростить визуализацию и разработку фрактальной графики. Это позволяет разобраться, какой именно фрактал подойдет для выданной задачи. Конечно, есть и другие виды, но обозначив основные можно начать работу.

### Понятие фрактальная графика

Фрактальная графика — это область компьютерной графики, основанная на использовании математических алгоритмов, описывающих фрактальные структуры. В отличие от традиционной графики, которая строится из базовых геометрических фигур (точек, линий, полигонов), фрактальная графика создаёт изображения, обладающие самоподобием, бесконечной детализацией и нерегулярной формой. Данные свойства имеются у фракталов и на основе них и строится графика.

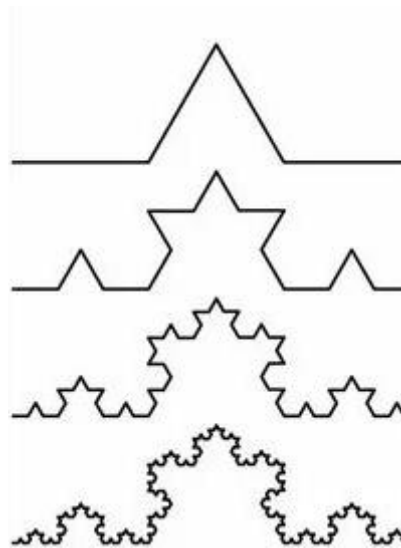
Ниже перечислены способы построения на основе фракталов:

#### 1) Геометрические фракталы:

Метод был разработан Аристидом Линденмайнером в 1968 году для моделирования роста растений. С того времени L- системы активно развивались и

стали одним из инструментов в компьютерной графике для генерации фрактальных структур. Суть метода заключается в рекурсивной замене символов по заданным правилам, что приводит к построению всё более сложных строк. Каждая строка интерпретируется как команда для "черепаховой графики" (Turtle graphics) — способ рисования на плоскости с помощью движений и поворотов. Если разбирать по шагам, то выглядит это так: у нас есть первая итерация 0 – A начальная строка. Итерация 1: AB (заменяли A на AB)- Итерация 2: ABA ( $A \rightarrow AB, B \rightarrow A$ )- Итерация 3: ABAAB (и так далее...). Видно, что на каждой итерации мы переписываем всю строку, применяя правила ко всем символам одновременно. Перейдем к более наглядным примерам.

Кривая Коха или как еще называют, «Снежинка Коха», строится по такому принципу:  $F \rightarrow F+F-F+F$ , где F — нарисуй линию вперед, «+» — поверни направо, «-» — поверни налево.



Снежинка Коха.

Индивидуальной особенностью метода является простота реализации, четкая формальная структура, а также позволяет моделировать биологический рост и фрактальные структуры.

Опишем еще один пример, «Треугольник Серпинского». Треугольник Серпинского — это один из наиболее известных и наглядных примеров

геометрического фрактала. Он иллюстрирует принцип самоподобия, при котором структура повторяется на каждом уровне масштабирования.

Фрактал строится следующим образом: вначале берётся равносторонний треугольник. Затем этот треугольник делится на четыре меньших равносторонних треугольника путём соединения середин сторон. Центральный треугольник удаляется, оставляя три внешних. Далее эта операция повторяется рекурсивно для каждого из оставшихся треугольников. С каждым шагом структура становится всё более сложной, но сохраняет свой общий треугольный силуэт.

Треугольник Серпинского обладает рядом интересных математических свойств. Его площадь стремится к нулю по мере увеличения количества итераций, но периметр остаётся бесконечным. Это демонстрирует фрактальную природу объекта — бесконечную сложность, содержащуюся в конечной форме.

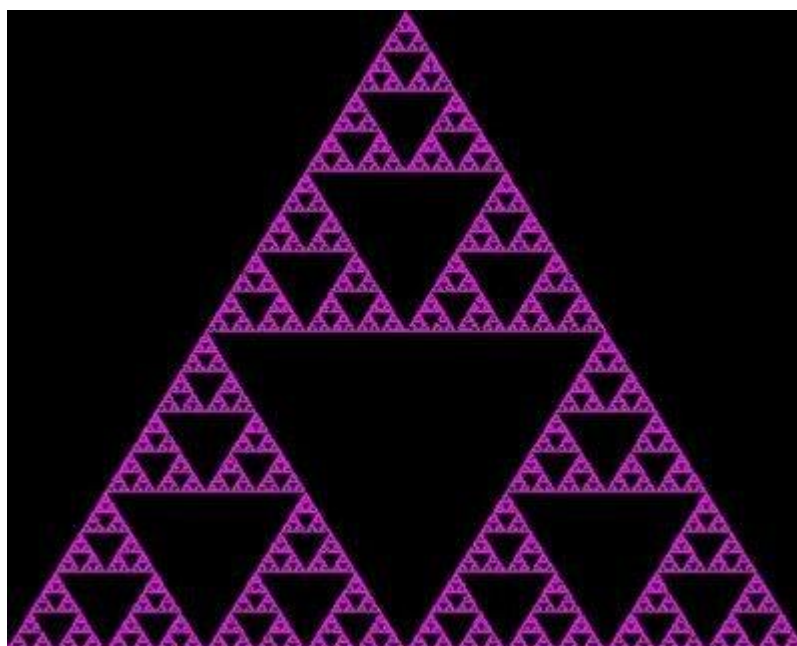


Рисунок . Треугольник Серпинского

## 2) Алгебраические фракталы:

По названию, можно понять, что метод реализуется при помощи математических или программных алгоритмов. Они формируются с использованием числовых вычислений, рекурсивных процедур и итераций формул.

Перечислим примеры использования данного метода. Начнем с множества Мандельброта. Оно является одно из самых известных фракталов не только потому, что создатель являлся тем, кто ввел в обиход термин «фрактал». Фрактал Мандельброта лежит в основе фрактального арта: художники дизайнеры и разработчики используют его для создания абстрактных фонов, эстетических визуальных эффектов и текстур в 2D и 3D – графике.

Строится по такому алгоритму:

1. Перебираем пиксели экрана.
  2. Для каждого пикселя берём его координаты как комплексное число  $c$ .  
Начинаем с  $z = 0$  и итеративно считаем по формуле:
- $$z_{n+1} = z_n^2 + c$$
3. Если модуль  $|z|$  становится больше 2, выходим.
  4. Количество итераций до выхода — это значение, по которому окрашиваем пиксель.

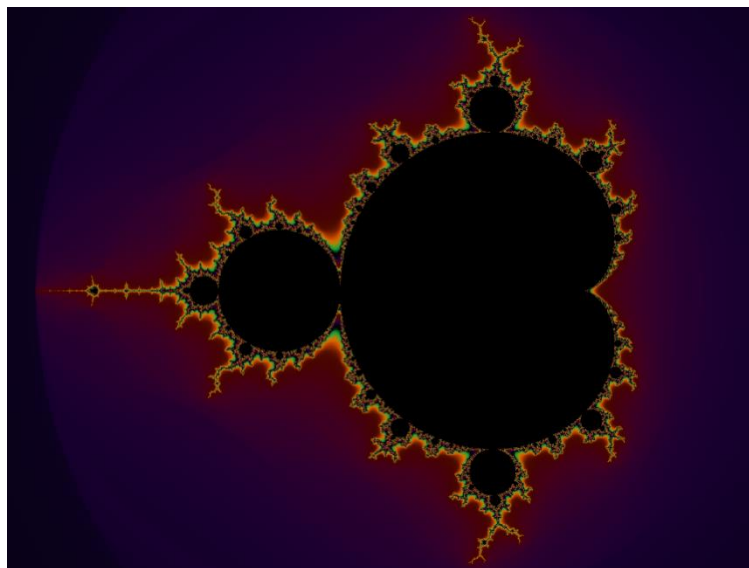


Рис. Фрактал Мандельброта

Следует привести еще один пример — множество Жюлиа. Это множество, разработанное французским математиком Гастоном Жюлиа, также основывается на

формуле Фату. Хотя оно напоминает фрактал Мандельброта, у него есть определенные математические отличия, которые влияют на конечный результат. В отличие от множества Мандельброта, которое на каждой итерации использует новое значение параметра  $C$ , в множестве Жюлиа это значение остается фиксированным на протяжении всех циклов. Таким образом, при различных значениях  $C$  фрактал Жюлиа может быть визуализирован по-разному.

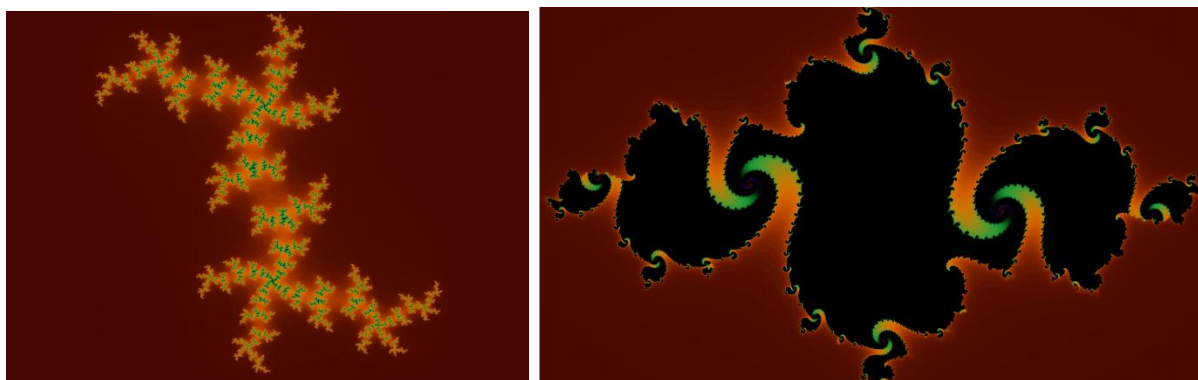


Рис. Метод Жюлиа

### 3) Стохастические метод.

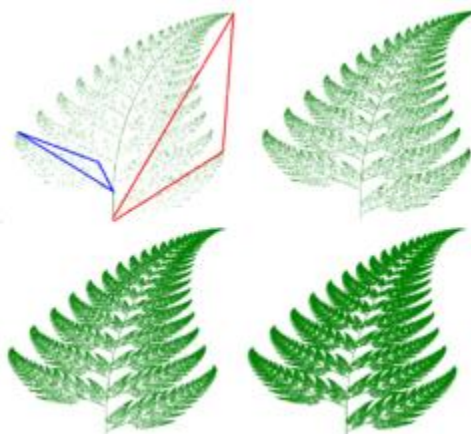
Данный метод интересен тем, что строится, используя случайные числа и вероятности. Он способен не повторять точную форму на каждом уровне и сохранять общее фрактальное поведение и самоподобие на статистическом уровне. Сферы применения таких фракталов обширна. Их используют для компьютерной графики (визуализации природы, спецэффекты), моделирование ландшафтов, погоды и создания реалистичных текстур и поверхностей в 3D играх.

Для лучшего понимания предоставим пример, а именно папоротник Барнсли. Данный график имитирует настоящий папоротник и поэтому задается четырьмя преобразованиями, каждое из которых отвечает за отдельную часть листа. Представим данные и вычисления в виде таблицы:

№	Формула (a, b, c, d, e, f)	Вероятность
1	(0, 0, 0, 0.16, 0, 0)	1%

2	$(0.85, 0.04, -0.04, 0.85, 0, 1.6)$	85%
3	$(0.2, -0.26, 0.23, 0.22, 0, 1.6)$	7%
4	$(-0.15, 0.28, 0.26, 0.24, 0, 0.44)$	7%

Выполнив несколько тысяч итераций, мы получим узнаваемый рисунок папоротника.



Изучив метод, можем выделить такие преимущества, как:

- Генерирует изображения с высокой степенью самоподобия.
- Весь фрактал задаётся лишь несколькими формулами и коэффициентами.

Подходит для фрактального сжатия изображений, так как позволяет хранить картинку не как пиксели, а как набор преобразований.

Фракталы находят широкое и разнообразное применение в науке, цифровых технологиях, дизайне, архитектуре и медицине. Благодаря своей уникальной природе — свойству самоподобия и способности к бесконечному масштабированию — фракталы позволяют описывать и моделировать сложные системы, как природные, так и искусственные.

В науке фракталы используются для моделирования объектов природы: деревьев, горных ландшафтов, береговых линий и облаков. Такие формы сложно описывать с помощью классической геометрии, однако фрактальные алгоритмы

позволяют создавать очень реалистичные модели. Также фракталы играют важную роль в теории хаоса и изучении динамических систем: например, при визуализации аттракторов или исследовании поведения нелинейных процессов.

В цифровом искусстве фракталы являются мощным инструментом для генеративной графики. С их помощью создаются захватывающие абстрактные изображения и анимации, которые находят применение в видеодизайне, визуальных инсталляциях и интро-роликах. Кроме того, фрактальные узоры используются в оформлении пользовательских интерфейсов и как элементы декоративного дизайна.

Фрактальные алгоритмы активно применяются и в компьютерных играх. Процедурная генерация ландшафтов, деревьев и других объектов игровых миров позволяет разработчикам создавать масштабные и детализированные сцены с минимальными затратами ресурсов. Фрактальные методы также помогают в сжатии данных, что повышает производительность и уменьшает объём хранимой информации.

В архитектуре фракталы становятся источником вдохновения для построения параметрических и бионических форм. Современные архитекторы создают сложные фасады и декоративные элементы, опираясь на повторяющиеся геометрические узоры. Это позволяет сочетать эстетическую привлекательность с функциональностью и инновационностью.

Наконец, в медицине и биологии фрактальный анализ применяется для изучения сосудистых систем, структуры лёгких, ДНК, а также в диагностике патологий. Фрактальные характеристики биологических тканей помогают исследовать их нормальное и аномальное состояние, например, при анализе медицинских изображений.

Таким образом, фракталы — это не только математическая абстракция, но и мощный инструмент, позволяющий исследовать, визуализировать и создавать сложные структуры в самых разных областях человеческой деятельности.



В процессе анализа были изучены фракталы и фрактальная графика. Они представляют собой уникальные математические объекты, обладающие определенными свойствами: самоподобием, итеративной структурой, дробной размерностью, бесконечной детализацией и нерегулярностью.

Были изучены виды фракталов. Каждый из них имел свои особенности, алгоритмы построения и области применения. Методы генерации фрактальной графики позволяют создавать сложные, но математически строгие изображения, что делает фрактальную графику применимой как в научных, так и художественных целях.

## Практическая часть

### Постановка задачи и цели разработки

Основной целью разработки является создание информативного, наглядного и доступного веб-сайта, посвящённого фрактальной графики. Сайт должен представлять теоретическую информацию о фракталах, методах их генерации, а также демонстрировать визуальные примеры, позволяющие пользователю лучше понять принципы построения фрактальных структур. Сайт разрабатывается как часть учебного проекта и предназначен для популяризации математических и визуальных основ фракталов.

Цель данного сайта предоставить понятную структурированную информацию, чтобы ее мог понять не только специалист, но и обычный обыватель. Главная цель - сделать удобный и функциональный сайт, чтобы его можно было бы просматривать и в компьютере, и в телефоне.

### Выбор технологий и обоснование

Данный сайт был реализован в среде для разработки Visual Studio Code. Основные причины выбора это: легкость, гибкость и большой выбор расширений, поддерживающих современные веб-технологии. Это кроссплатформенное приложение с множеством функций, включая подсветку синтаксиса, автодополнение, встроенный терминал и интеграцию с Git.

Современные программисты часто используют сторонние инструменты, одни из которых фреймворки. Это уже готовый набор инструментов в программировании, а именно готовые решения для часто используемых задач, на основе которых можно написать свой код. Для данного же сайта был выбран фреймворк ASTRO. Главные преимущества являются:

- Статическая генерация: Astro генерирует полностью статичные HTML-страницы, что повышает скорость загрузки и улучшает SEO.

- Zero JavaScript by default: если JS не нужен — он не загружается, сайт быстрее и легче.
- Интеграция с компонентами из разных фреймворков: можно использовать React, Vue, Svelte и другие в одном проекте.
- Простота и минимализм: низкий порог вхождения, особенно если знаком с HTML/CSS/JS.

Таким образом ASTRO оказалось идеальным инструментом для создания сайта на данную тематику, потому что для кода были использованы в основном с HTML/CSS/JS, а также чтобы легко автоматизировать ввод статей был использован Markdown.

Выбор технологий HTML, CSS и JavaScript был обусловлен спецификой проекта и особенностями самого фреймворка **Astro**, который ориентирован на создание быстрых, статичных и удобных в разработке сайтов. Эти языки являются основополагающими для веб-разработки:

- HTML (HyperText Markup Language) был использован для структурирования контента страниц. Он позволяет логически организовать разделы, заголовки, изображения, списки и другие элементы, что особенно важно для фрактальной тематики сайта с множеством текстов и графики.
- CSS (Cascading Style Sheets) обеспечил стилизацию проекта: выбор цветовой схемы, типографики, отступов, анимаций и адаптивной верстки. Использование Tailwind CSS или кастомных модулей позволило гибко управлять внешним видом без излишней сложности.
- JavaScript применялся для создания интерактивных элементов, таких как визуализация фракталов на canvas, анимации, переключение тем (светлая/тёмная), а также динамическая подгрузка или фильтрация данных.

Дополнительно, Markdown был выбран как формат для хранения и отображения статей. Это облегчает работу с текстом, делая написание и

редактирование контента удобным, особенно для не-технических пользователей. Astro предоставляет нативную поддержку Markdown, позволяя автоматически преобразовывать его в полноценные страницы с применением заданного шаблона.

Таким образом, совокупность этих технологий обеспечила простоту разработки, поддержку интерактивности и лёгкое масштабирование сайта.

Для управления кодом и организации процесса разработки использовалась интеграция с GitHub и Netlify. GitHub выступал в роли системы контроля версий и хостинга репозитория, что значительно упростило командную работу, отслеживание изменений и обеспечение надежного резервного копирования проекта.

После загрузки проекта на GitHub, репозиторий был подключен к платформе Netlify, которая обеспечивает автоматическую сборку и деплой сайта при каждом новом коммите. Это позволяет мгновенно вносить изменения и видеть результат в реальном времени без необходимости вручную пересобирать и загружать сайт.

Кроме автоматического деплоя, Netlify предоставляет ряд дополнительных возможностей:

- Бесплатный SSL-сертификат для безопасного соединения по HTTPS,
- Простая настройка пользовательского домена,
- Поддержка обработки форм без серверной логики,
- Гибкие настройки редиректов и других параметров конфигурации проекта.

Эта связка GitHub + Netlify позволяет выстроить удобный и современный процесс разработки и публикации сайта. А их доступность делают данные инструменты идеальными.

## Выбор методологии

В рамках дипломной работы была использована методология Agile, которая наилучшим образом подходит для итеративного подхода, постоянной адаптации и улучшения продукта в процессе работы. Благодаря своей поэтапной структуре

Работа над дипломным проектом велась с использованием гибкой методологии Agile, которая доказала свою эффективность при создании современного веб-приложения. Agile — это не просто набор техник, а философия, предполагающая поэтапное выполнение работы, её постоянную проверку и корректировку на каждом шаге. Такой подход оказался особенно полезным в процессе разработки сайта, посвящённого фрактальной графике, поскольку проект требовал как технической гибкости, так и творческого подхода.

Проект был разбит на несколько итераций (спринтов), каждый из которых решал конкретные задачи. На первом этапе было изучение возможных инструментов, выбор и знакомство с документацией Astro — современного фронтенд-фреймворка, выбранного как основа для реализации. После этого были определены маршруты, подготовлены шаблоны шапки и подвала, что позволило единообразно стилизовать все страницы.

Следующим шагом стало создание главной страницы и разделов, в которых размещались статьи о типах фракталов. Особенностью Agile-подхода стала возможность гибко адаптироваться к новым идеям. Например, в процессе работы было решено не просто разместить текст, а добавить интерактивные визуализации фракталов через canvas и JavaScript. Это не было запланировано на старте, но гибкость методологии позволила органично включить эти элементы без нарушения общей архитектуры проекта.

Каждый спринт завершался работающим фрагментом сайта, который я мог сразу опубликовать через Netlify. Это обеспечивало регулярное тестирование и возможность получать обратную связь. После развёртывания проекта в интернете, я добавил описание о себе и использованных технологиях, реализовал светлую и

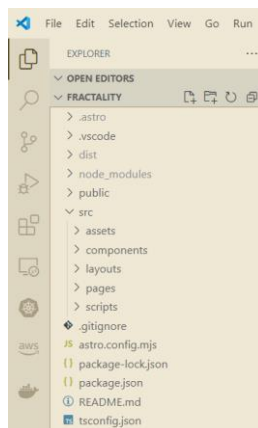
тёмную темы, переработал визуальный стиль подвала и секции преимуществ фракталов на основе пользовательских предпочтений.

Для отслеживания задач и управления версией проекта использовался Git с удалённым репозиторием на GitHub. Это позволяло в любой момент вернуться к предыдущей версии проекта, проверить историю изменений и работать поэтапно, не теряя прогресса.

В целом, применение Agile позволило организовать работу над дипломом эффективно, с чётким пониманием этапов и гибкостью в решениях. Благодаря этому методологическому подходу, сайт получился функциональным, адаптивным и современным, соответствующим как техническим, так и эстетическим требованиям. Agile обеспечил не только техническую базу, но и комфортный творческий процесс.

### Архитектура проекта и структура сайта

Любой, кто начинает работать с кодом, хочет видеть понятную структуру файловой системы. Поэтому стоит начать с описания, расположения и причины создания папок.



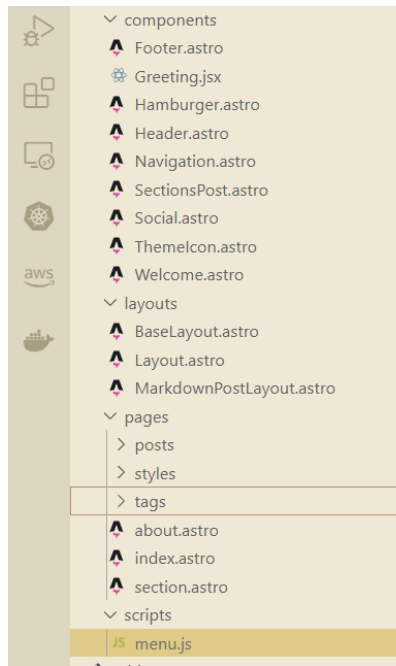
Файловая система

Astro по умолчанию создаёт несколько служебных папок, которые не требуют ручного редактирования. Они содержат установленные зависимости и внутренние ресурсы, используемые фреймворком. Основная разработка проекта обычно

ведётся в директориях `src/` и `public/`, поэтому именно их структура будет подробно рассмотрена ниже.

В `public/` расположены все изображения и логотип. `src/` основная директория, где размещаются все пользовательские ресурсы и компоненты. Она отвечает за структуру, логику и оформление интерфейса сайта. Содержит:

- `components/` — переиспользуемые элементы интерфейса (например, `Header`, `Footer`, `Social`, `Welcome`). Компоненты оформлены в виде `.astro` или `.jsx` файлов и инкапсулируют разметку, стили и поведение.
- `layouts/` — шаблоны страниц. Используются для задания общей структуры документа: шапки, подвала, секций контента. Это своего рода "скелет" для всех страниц (`BaseLayout.astro`, `MarkdownPostLayout.astro` и `Layout.astro`).
- `pages/` — файлы, определяющие маршруты сайта. Каждый `.astro` файл становится доступным по соответствующему URL. Например, `pages/about.astro` будет доступна по пути `/about`. Там расположены главные страницы сайта.
- `assets/` — папка для вспомогательных ресурсов (например, иконок, изображений или шрифтов), используемых внутри компонентов.
- `scripts/` — используются для хранения вспомогательных JavaScript-функций, конфигураций, анимаций и т.п., а именно `menu.js`, в которой расположена меню для режима телефона.



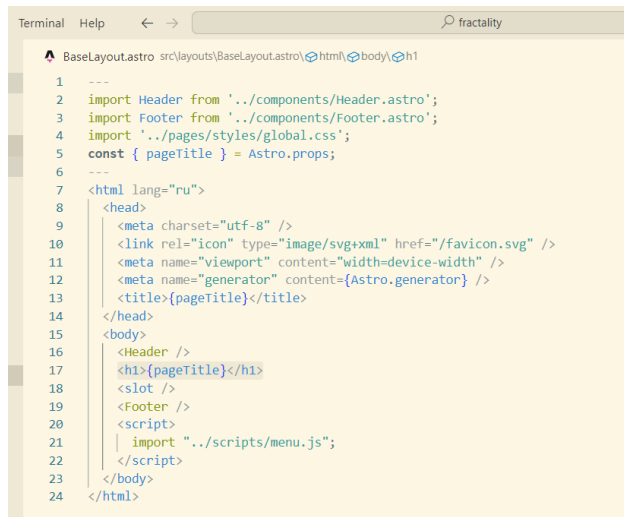
Директория src/

## Этапы разработки

На начальном этапе была изучена официальная документация Astro и пройдены обучающие уроки, представленные на их сайте. Это позволило быстро освоиться с подходами фреймворка и понять его основные принципы работы: компонентность, маршрутизация и использование макетов (layouts).

После настройки окружения была создана структура проекта с основной директорией src/. Для упрощения работы, чтобы не приходилось заново создавать страницу с нуля был разработан единый шаблон для каждой страницы, который по мере необходимости можно изменять. В него вынесены общие части сайта — шапка, навигация и подвал, что позволило обеспечить единый стиль для всех страниц(рис.). Отдельно же были разработаны стили, в общем файле “global.css” (рис.). Туда же по ходу работы будут записываться остальные стили для страниц кнопок и всего остального.



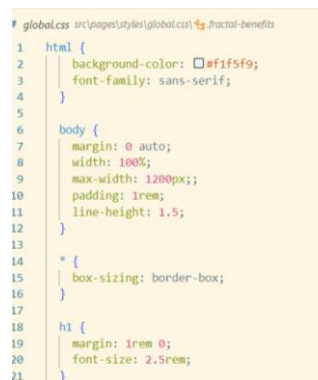


```

1  ---
2  import Header from '../components/Header.astro';
3  import Footer from '../components/Footer.astro';
4  import '../pages/styles/global.css';
5  const { pageTitle } = Astro.props;
6  ---
7  <html lang="ru">
8  <head>
9    <meta charset="utf-8" />
10   <link rel="icon" type="image/svg+xml" href="/favicon.svg" />
11   <meta name="viewport" content="width=device-width" />
12   <meta name="generator" content={Astro.generator} />
13   <title>{pageTitle}</title>
14 </head>
15 <body>
16   <Header />
17   <h1>{pageTitle}</h1>
18   <slot />
19   <Footer />
20   <script>
21     import '../scripts/menu.js';
22   </script>
23 </body>
24 </html>

```

Рисунок. Шаблон страниц



```

1  html {
2    background-color: #f1f5f9;
3    font-family: sans-serif;
4  }
5
6  body {
7    margin: 0 auto;
8    width: 100%;
9    max-width: 1200px;
10   padding: 1rem;
11   line-height: 1.5;
12 }
13
14 * {
15   box-sizing: border-box;
16 }
17
18 h1 {
19   margin: 1rem 0;
20   font-size: 2.5rem;
21 }

```

Рисунок. Стили

Затем была разработана главная страница, на которой размещено краткое описание проекта и реализована галерея фрактальных изображений, чтобы пользователь сразу мог визуально познакомиться с темой.

Для информирования о технологиях, использованных при разработке, был создан отдельный раздел «О сайте». В нем размещены сведения об авторе и инструментах, применённых в ходе реализации: Astro, CSS, интерактивные скрипты на JavaScript и анимации с использованием Canvas API. Напимер на сатие была реализована анимация стохастического дерева.

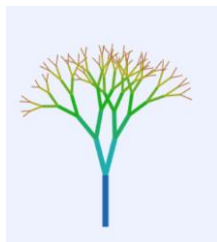


Рисунок. Стохастического дерева

```
<canvas id="treeCanvas" width="500" height="400" style="display: block; margin: auto;
background: #eef2ff; border-radius: 10px;"></canvas>
```

```
<script is:inline>
  const canvas = document.getElementById("treeCanvas");
  const ctx = canvas.getContext("2d");

  function drawBranch(x, y, length, angle, depth) {
    if (depth === 0) return;
    const x2 = x + length * Math.cos(angle);
    const y2 = y - length * Math.sin(angle);
    ctx.beginPath();
    ctx.moveTo(x, y);
    ctx.lineTo(x2, y2);
    ctx.strokeStyle = `hsl(${depth * 30}, 70%, 40%)`;
    ctx.lineWidth = depth;
    ctx.stroke();

    const newLength = length * (0.7 + Math.random() * 0.15);
    const angleVariation = 0.2 + Math.random() * 0.3;

    drawBranch(x2, y2, newLength, angle - angleVariation, depth - 1);
    drawBranch(x2, y2, newLength, angle + angleVariation, depth - 1);
  }

  function drawTree() {
    ctx.clearRect(0, 0, canvas.width, canvas.height);
    drawBranch(250, 380, 60, Math.PI / 2, 7);
  }

  drawTree();
  setInterval(drawTree, 3000); // обновление каждые 3 секунды
</script>
```

Особое внимание было уделено разделу «Разделы», где собраны тематические статьи: от истории фракталов до описания их разновидностей и принципов генерации. Каждая статья — это отдельный Markdown-файл, отображаемый через общий шаблон, что упрощает добавление и поддержку новых материалов.

И заканчиваются разделы на странице «Сайты для построения фракталов». Данная страница поможет сэкономить время для по поиску доступных сайтов. На ней описаны плюсы и минусы того или иного сервиса.

### Хостинг и публикация

После завершения разработки сайта важным этапом стало его размещение в интернете и организация командной работы. Весь код проекта с самого начала вёлся под системой контроля версий Git. Это позволило, отслеживать изменения на каждом этапе, откатываться к предыдущим версиям при необходимости, вести чистую и понятную историю коммитов. Как говорилось ранее, Netlify был выбран как простая и быстрая платформа для деплоя:

- Репозиторий был подключён напрямую через интерфейс Netlify.
- Была указана команда сборки (`npm run build`) и директория вывода (`dist/`).
- После настройки проект автоматически деплоился при каждом новом коммите в ветку `main`.

Это позволило мгновенно видеть изменения на продакшене и упростило процесс обновлений.

Для сайта был подключён кастомный домен — `fractaline.netlify.app` с возможностью дальнейшей привязки собственного доменного имени. Были выполнены:

- базовые настройки DNS,
- активация HTTPS-сертификата через Let's Encrypt (Netlify подключает автоматически),
- редиректы и SEO-настройки для лучшей индексации.

### Оценка результата и возможности развития

Основной целью дипломного проекта было создание современного, визуально привлекательного и содержательно наполненного сайта, посвящённого

Санкт-Петербург, 2025 год

фрактальной графике. Все поставленные задачи были успешно реализованы. Сайт построен на фреймворке Astro, что обеспечило высокую производительность, модульность и удобство в работе с контентом.

В рамках проекта были достигнуты следующие ключевые цели:

- Разработана структура сайта с удобной навигацией и разделами, охватывающими историю, теорию и примеры фракталов.
- Реализована система шаблонов, обеспечивающая единый стиль и быстрое добавление новых материалов.
- Включены визуализации фракталов с использованием HTML5 Canvas и JavaScript, что усилило наглядность контента.
- Обеспечена адаптивность и кроссбраузерность сайта.
- Настроен хостинг и деплой через платформу Netlify, включая подключение собственного домена.
- Использован Git и GitHub для управления версиями, что позволило отслеживать ход разработки и обеспечило надёжное хранение кода.

Таким образом, все запланированные функции были реализованы в полном объёме, а итоговый результат соответствует современным требованиям к образовательным и визуальным веб-проектам.

Хотя текущая версия сайта является полноценной, ей смело можно пользоваться, в дальнейшем возможно расширение функциональности и улучшение пользовательского опыта. Вот некоторые направления для развития:

1. Добавление новых разделов и статей — можно расширить тематику за счёт включения фракталов в природе, биологии и физике, а также более глубокого математического анализа.
2. Интерактивные обучающие модули — создание пошаговых симуляторов построения фракталов и задач для закрепления знаний.

3. Генератор фракталов — интерфейс, в котором пользователь сможет задавать параметры и наблюдать визуализацию фракталов в реальном времени.
4. Личный кабинет — возможность зарегистрироваться, сохранять избранные фракталы или свои визуализации.
5. Мультиязычность — добавление английской версии сайта для расширения аудитории.
6. Интеграция с API — подключение внешних библиотек или сервисов для генерации сложных математических объектов.
7. Аналитика и обратная связь — подключение инструментов аналитики и форм для предложений/комментариев от пользователей.

## Заключение

В ходе выполнения дипломного проекта была разработана и успешно реализована веб-платформа, посвящённая теме фрактальной графики. Проект охватывает как теоретические основы, так и визуальные примеры, что делает его полезным как для ознакомления с фракталами, так и для более глубокого изучения их свойств и применения.

Основой технической реализации стал современный фреймворк Astro, который обеспечил высокую производительность, гибкость архитектуры и удобство работы с компонентами и шаблонами. Использование языков HTML, CSS и JavaScript позволило эффективно управлять визуализацией фракталов и создавать интерактивный пользовательский интерфейс. Для структурирования контента был применён формат Markdown, обеспечивающий простоту добавления и редактирования статей.

Сайт успешно прошёл этапы дизайна, программной реализации, тестирования и развёртывания на платформе Netlify. Также была настроена система контроля версий через Git и организована публикация кода на GitHub, что позволяет в будущем легко вносить улучшения и работать в команде.

В результате был создан современный, эстетически оформленный и функциональный веб-ресурс, который может быть использован как учебное пособие, визуальный справочник или основа для дальнейшего развития образовательного проекта. Работа над сайтом позволила автору не только углубить знания в области веб-разработки, но и применить на практике методологию Agile, обеспечив гибкость и последовательность в реализации целей.

Таким образом, дипломный проект полностью соответствует поставленным задачам и демонстрирует как технические, так и креативные навыки автора. Возможности дальнейшего развития ресурса открывают простор для совершенствования и расширения функциональности с целью привлечения более широкой аудитории.

## Литература

- 1) <https://3dnews.ru/754657>
- 2) <https://ru.wikipedia.org/wiki/L-система>