



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
 РОССИЙСКОЙ ФЕДЕРАЦИИ
 ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
 ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
 «ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
 (ДГТУ)

Факультет «Информатика и вычислительная техника»

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

ОТЧЕТ

по учебной практике по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности

на ФГБОУ ВО «Донской государственный технический университет»

кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

Обучающийся *[подпись]* 06.06.20

Л.Р. Гвоздев

подпись, дата

Обозначение отчета УП.920000.000

Группа ВМО21

Направление 02.03.03 Математическое обеспечение и администрирование информационных систем

Профиль Математическое обеспечение и администрирование информационных систем

Руководитель практики:

от кафедры доцент

[подпись] 06.06.2020

Т.А. Медведева

должность

подпись, дата

имя, отчество, фамилия

Оценка отлично

06.06.2020

[подпись]

дата

подпись преподавателя

Ростов-на-Дону

2020



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)

Факультет «Информатика и вычислительная техника»

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

ЗАДАНИЕ

по учебной практике по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности

на ФГБОУ ВО «Донской государственный технический университет»
кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»
в период с «10» февраля 2020 г. по «06» июня 2020 г.

Обучающийся Гвоздев Леонид Русланович

Обозначение отчета УП.920000.000 Группа ВМО21

Срок представления отчета на кафедру «06» июня 2020 г.

Содержание индивидуального задания

Моделирование растительных объектов с помощью элементов фрактальной геометрии



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)**

Факультет «Информатика и вычислительная техника»

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

Рабочий график (план) проведения практики

№	Мероприятие	Срок выполнения
1	Прохождение вводного и первичного инструктажа по охране труда на рабочем месте и инструктажа по пожарной безопасности на объекте	С 10.02.2020 по 13.02.2020
2	Получение индивидуального задания и постановка задачи	С 14.02.2020 по 18.02.2020
3	Теоретическое изучение фрактальной геометрии	С 19.02.2020 по 17.03.2020
4	Изучение алгоритмов фрактальной графики	С 18.03.2020 по 07.04.2020
5	Выбор среды разработки программы и ее реализация	С 07.04.2020 по 28.04.2020
6	Тестирование программного средства и составление отчета о проделанной работе	С 29.04.04 по 25.05.2020
7	Защита итогового отчета по практике	С 26.05.2020 по 06.06.2020

Ростов-на-Дону

2020

ДНЕВНИК ПРОХОЖДЕНИЯ ПРАКТИКИ

Дата	Место работы	Выполняемые работы	Оценка руководителя
с 10.02.2020 по 11.02.2020	ДГТУ	Знакомство с предприятием, прохождение вводного инструктажа	Всн. Мед
с 12.02.2020 по 13.02.2020	ДГТУ	Ознакомление с территорией предприятия, прохождение первичного инструктажа по ТБ, ПБ	Всн. Мед
с 14.02.2020 по 18.02.2020	ДГТУ	Получение индивидуального задания	Всн. Мед
с 19.02.2020 по 17.03.2020	ДГТУ	Изучение теории фрактальной геометрии и L-систем для моделирования растительных объектов	Всн. Мед
с 18.03.2020 по 07.04.2020	ДГТУ	Алгоритмическое конструирование	Всн. Мед
с 08.04.04 по 28.04.2020	ДГТУ	Программное конструирование и отладка программы	Всн. Мед
с 29.04.2020 по 25.05.2020	ДГТУ	Тестирование программного средства, подробное описание работы программы, написание отчета о проделанной работе	Всн. Мед
с 26.05.2020 по 06.06.2020	ДГТУ	Защита итогового отчета по практике	Всн. Мед

ОТЗЫВ - ХАРАКТЕРИСТИКА

Обучающийся Гвоздев Леонид Русланович

2 курса группы ВМО21 кафедра «ПОВТиАС»

Вид практики учебная

Наименование места практики ФГБОУ ВО ДГТУ

Обучающийся выполнил задания программы практики

К выполнению работы подошел ответственно. Все этапы прохождения учебной практики выполнены в срок.

Дополнительно ознакомился/изучил

Инструктаж по технике безопасности.

Заслуживает оценки

отлично

Руководитель практики от
кафедры

Т.А. Медведева Т.А. Медведева
« 06 » 06 20 20 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	7
1 Теоретический обзор	8
1.1 Фрактальная геометрия и L-системы	8
1.2 Геометрическая интерпретация L-систем	9
1.3 Моделирование растительных объектов с помощью DOL-систем	12
1.4 Стохастические системы Линденмайера	14
1.5 Выводы по главе	15
1.6 Постановка задачи	15
2 Алгоритмическое конструирование	16
2.1 Схема реализации DOL-систем	16
2.2 Схема реализации стохастических контекстно-свободных L-систем	17
2.3 Выводы по главе	17
3 Программное конструирование	18
3.1 Выбор среды и языка программирования	18
3.2 Описание программы	18
3.3 Выводы по главе	19
4 Тестирование программы	20
4.1 Результаты работы программного средства	20
4.2 Выводы по главе	25
ЗАКЛЮЧЕНИЕ	26
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	27
ПРИЛОЖЕНИЕ А Исходный код программного средства	28

				УП.920000.000		
Имя	Лист	Матрица	Подпись	Моделирование растительных объектов с помощью элементов фрактальной геометрии		
Листов	Генератор 2.7		<i>Игорь...</i>			
Руководител	Матрица 1.4		<i>Игорь...</i>		Лист	Листов
					6	40
Примечание				ДИТУ кафедра «ПОВТнАС»		
Утвержд.						

ВВЕДЕНИЕ

Многообразие природы всегда привлекало исследователей. Активной всего учеными изучались геометрические свойства растений, такие как симметрия листьев и цветов, расположение семечек в шишках. Во время роста живых организмов, особенно растений, можно видеть регулярно повторяющиеся многоклеточные структуры.

Сложную геометрическую фигуру, обладающую свойством самоподобия, т.е. составленной из нескольких частей, каждая из которых подобна целой фигуре, называют фракталом.

В настоящее время ученые находят все больше областей для применения теории фракталов. С их помощью можно анализировать колебания котировок на бирже, исследовать всевозможные естественные процессы, такие как колебание численности видов или моделирование динамики потоков.

Для описания строения и развития сложных растительных структур достаточно знать простой алгоритм ветвления. Форму растения, как правило нельзя описать заранее – она подвержена вариации, невозможно узнать какую в точности форму приобретет растение. Алгоритмический подход является решением этой проблемы. С помощью описания и построения алгоритмов, есть возможность изучать и моделировать формы сложных организмов.

					Лист
					7
Изм.	Лист	№ докум.	Подпись	Дата	

1 Теоретический обзор

1.1 Фрактальная геометрия и L-системы

Аристид Линденмайер – венгерский биолог и ботаник в 1968 году предложил L-системы для описания поведения клеток и моделирования развития растений. Изначально системы Линденмайера могли описывать развитие таких простых многоклеточных организмов как дрожжи, нитевидные грибы, различные виды водорослей и связи между соседними клетками растений.

L-система – это вид формальной грамматики, имеющая рекурсивный характер и основанная на параллельном переписывании строк.

Система Линденмайера состоит из трёх элементов:

$$L = (V, \omega_0, R), \text{ где}$$

V – алфавит символов, множество состоящее из переменных (заменяемых символов) и констант (незаменяемых символов),

ω_0 – аксиома, строка символов из V , определяющая начальное состояние системы,

R – множество порождающих правил, определяющих, каким образом переменные могут быть заменены комбинациями других переменных и констант, присутствующих в алфавите.

Применяя множество порождающих правил к аксиоме, можно получить грамматику L-системы, представляющую собой простую строку из алфавита символов. От формального языка, генерируемого по формальной грамматике, L-системы отличаются тем фактом, что на каждой итерации применяется как можно больше правил.

Аксиома – первое состояние предложения, для которого можно многократно применять порождающие правила, моделируя эволюцию и рост системы. Порождающие правила применяются путем замены одиночных

					УП.920000.000	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

символов на цепочку символов в соответствии с правилами. Если конкретный символ не подходит под заданные правила, то он заменяется самим собой. В качестве простого примера приведена система с аксиомой А и правилом $A \rightarrow ABA$.

Аксиома и первое предложение: А

После первой итерации предложение сменится на: АВА

После второй итерации предложение будет иметь вид: АВАВАВА

Саморасширяющееся предложение становится аналогом деления клеток в растениях и других биологических процессов.

Можно выделить несколько основных типов L-систем:

- *контекстно-зависимые и контекстно-независимые L-системы* - L-система является контекстно-независимой, если подстановка выполняется независимо от ее окружения; в контекстно-зависимой L-системе применение правила зависит не только от единичного символа, но и от соседей;
- *детерминированные и стохастические L-системы* – если существует в точности одно правило для каждого символа алфавита, то система называется детерминированной; если же существует несколько правил для символов и с каждой итерацией правило выбирается с некоторой долей вероятности, то это стохастическая система;
- *параметрические и непараметрические L-системы* – в параметрической системе правила могут выполняться при соблюдении определенных условий, в такой системе порождающее правило состоит из предшествующего элемента, условия, и последующего элемента.

1.2 Геометрическая интерпретация L-систем

Многие фракталы можно представить как последовательности простых элементов – отрезков. Однако длина сегментов и углы между ними имеют

					УП.920000.000	Лист
Изм.	Лист	№ докум.	Подпись	Дата		9

большое значение. Для создания фракталов, строки, генерируемые L-системами, должны содержать необходимую информацию о геометрии фигуры. Этот стиль визуализации называется графикой с использованием относительных команд, основанный на понятии «черепашки в стиле LOGO». Эта интерпретация была первоначально предложена Сцилардом и Куинтоном.

Состояние графического устройства определяется как тройка вида (x, y, α) , где декартовы координаты (x, y) представляют положение в данный момент, α – угол, называемый направлением поворота. Учитывая размер шага d , и приращение угла δ , устройство может реагировать на команды, представленные следующими символами:

- F – продвижение на шаг длины d , с прорисовкой отрезка;
- f – продвижение на шаг длины d , без прорисовки отрезка;
- + – повернуть направо на угол δ ;
- - – повернуть налево на угол δ .

Все остальные символы игнорируются. Пример представлен на рисунке 1.1.

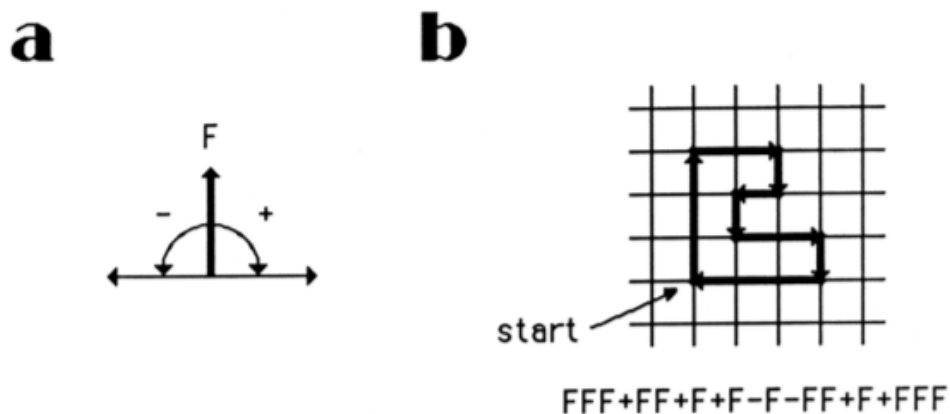


Рисунок 1.1 – (а) Представление символов F, +, -. (б) Представление строки.

Приращение угла δ равно 90° .

Имея строгий метод для отображения строк, можно применить его для представления генерируемых L-системами строк. На рисунке 1.2 представлены четыре приближения «квадратного острова Коха». Эти фигуры получены путем отображения строк, генерируемых следующей L-системой:

$$\omega: F + F + F + F;$$

$$p: F \rightarrow F + F - F - FF + F + F - F.$$

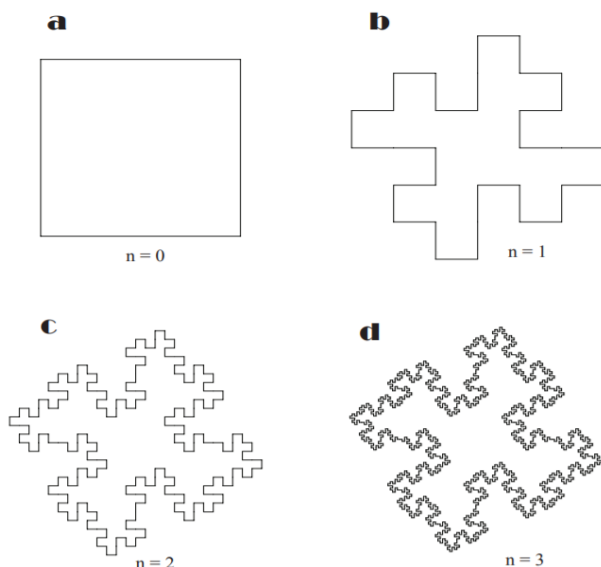
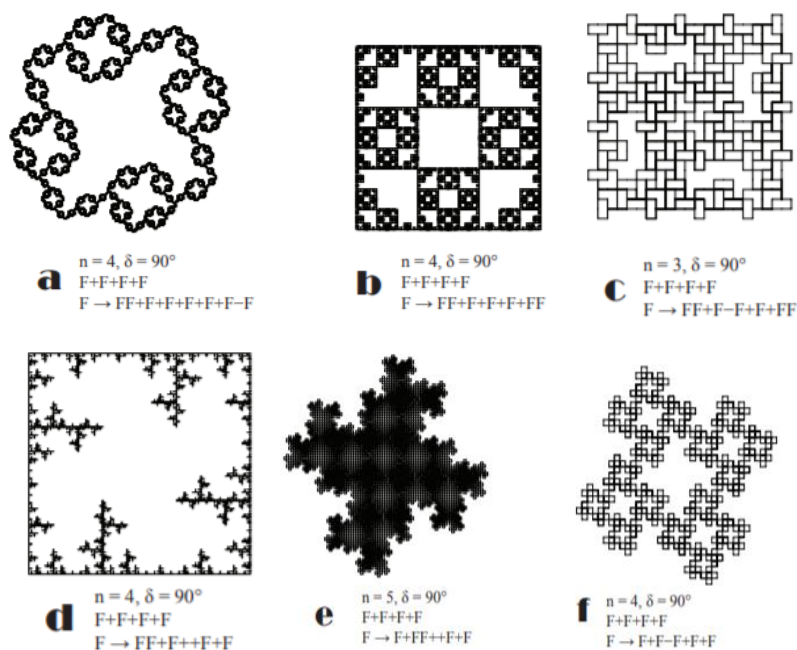


Рисунок 1.2 – Создание квадратичного острова Коха

Длина шага d уменьшается в четыре раза между последующими итерациями, что делает расстояние между конечными точками фигуры равными по длине. L-системы, указанные таким образом, могут восприниматься как кодировки конструкций Коха. Простота L-систем делает их удобными для разработки новых кривых Коха, например, можно начать с определенной L-системы и наблюдать за результатами вставки, удаления или замены некоторых символов. Некоторые фракталы полученные таким образом представлены на рисунке 1.3.



Изм.	Лист	№ докум.	Подпись	Дата

Рисунок 1.3 – Кривые Коха, полученные путем изменения порождающих правил
 Кривая дракона, показанная на рисунке 1.4, может быть сгенерирована путем многократной замены сегментов линий парами линий, образующими поворот вправо или влево. Это описывается следующей L-системой:

$$\omega: F1$$

$$p_1: F1 \rightarrow F1 + rF +$$

$$p_2: rF \rightarrow -F1 - rF$$



Рисунок 1.4 – Кривая дракона

Базовые L-системы имеют аналогичную структуру, как у кривой дракона. В рассмотренных ранее примерах символы «+» и «-» истолкованы, как поворот направо и налево на 90° . Другие значения поворота угла δ также могут привести к интересным изображениям.

1.3 Моделирование растительных объектов с помощью DOL-систем

Отображение символьной строки представляет собой последовательность отрезков, соединенных «началом к концу» друг с другом. В зависимости от длины сегмента и углов между ними, получающаяся линия самопересекающаяся или нет, может быть более или менее извилистой, причем некоторые сегменты прорисованы много раз, а другие сделаны невидимыми, но она всегда остается одной линией.

Изм.	Лист	№ докум.	Подпись	Дата

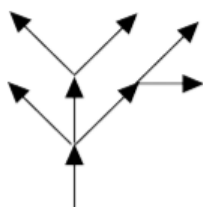
В статье 1968 года Линденмайер впервые предложил использование скобок для представления графических деревьев. Идея состояла в описании ветвящихся структур, присутствующих во многих растериях, от водрослей до деревьев, используя L-системы.

Для этого необходимо введение двух новых символов:

[- запись текущего состояния в стек. Информация, хранящаяся в стеке, содержит сведения о положении и ориентации графического устройства;

] – извлечение состояния из стека и преобразование состояния в текущее. Линия не проводится, хотя в целом положение меняется.

На рисунке 1.5 представлен пример системы Линденмайера, которая описывает ветвящуюся структуру растения.



$F[+F][-F[-F]F]F[+F][-F]$

Рисунок 1.5 – Графическое представление строки с скобках

Примеры двумерных ветвящихся структур, генерируемых OL-системами показаны на рисунке 1.6.

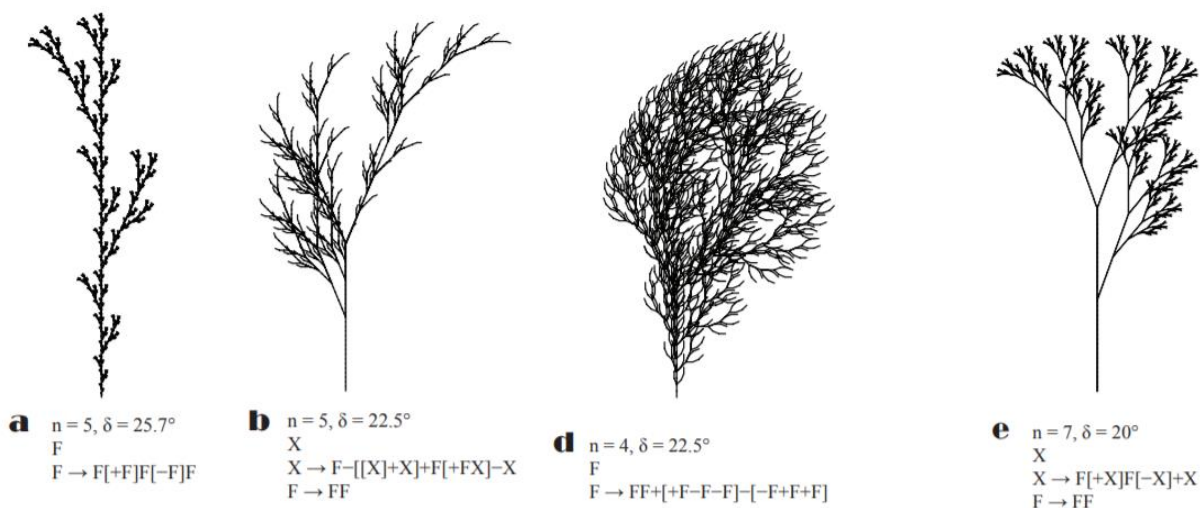


Рисунок 1.6 – Примеры ветвистых структур, созданных OL-системами

Рассмотренные примеры демонстрируют возможности использования L-систем для моделирования растительных объектов.

Изм.	Лист	№ докум.	Подпись	Дата

1.4 Стохастические системы Линденмайера

Главным недостатком DOL-систем является однозначность внешнего вида. Если рассматривать рост объектов растительного мира, то представители одного типа, при схожей структуре, могут принимать различные формы. Это зависит от влияния внешней среды, которое носит стохастический характер.

Если правила можно выбрать с определенной степенью свободы, система называется стохастической L-системой, обозначаемой в виде четверки вида (V, ω, R, π) . Функция π отображает вероятность применения определенного правила. Общая сумма всех вероятностей должна быть равна единице. Правила подстановки могут быть обозначены следующим образом:

$$P_1 = a \rightarrow^{0.2} aba;$$

$$P_2 = a \rightarrow^{0.3} bab;$$

$$P_3 = a \rightarrow^{0.5} abb.$$

Подмена выполняется с вероятностью 0.2 для P_1 , 0.3 для P_2 и 0.5 для P_3 .

При применении стохастической L-системы получаются объекты очень похожие на природные – несимметричные деревья, изрезанные береговые линии и так далее.

Например можно построить объект со следующей структурой L-системы:

$$\omega: F$$

$$P_1 = F \rightarrow^{0.4} F [+F] F [-F] F$$

$$P_2 = F \rightarrow^{0.3} F [+F] F$$

$$P_3 = F \rightarrow^{0.3} F [-F] F$$

На рисунке 1.10 представлена графическая интерпретация стохастической системы Линденмайера при различных запусках.



Рисунок 1.10 – Стохастические L-системы с одинаковыми правилами и вероятностями применения

Рассматривая результаты различных запусков, можно заметить, как стохастический характер влияет на внешний вид и структуру растительного объекта.

1.5 Выводы по главе

В данной главе рассмотрены элементы фрактальной геометрии и системы Линденмайера. Дана геометрическая интерпретация L-систем, подробно описано моделирование роста растений, и придание моделям выразительности с помощью стохастических систем.

1.6 Постановка задачи

Изучить теоретический материал по фрактальной геометрии и L-системам, а также особенности их моделирования. Разработать программное средство для визуализации DOL и стохастических L-систем. Тестировать программу с указанием различных аксиом, правил и начальных параметров.

Изм.	Лист	№ докум.	Подпись	Дата

2 Алгоритмическое конструирование

2.1 Схема реализации DOL-систем

На рисунке 2.1 продемонстрирована реализация алгоритма работы DOL – систем.

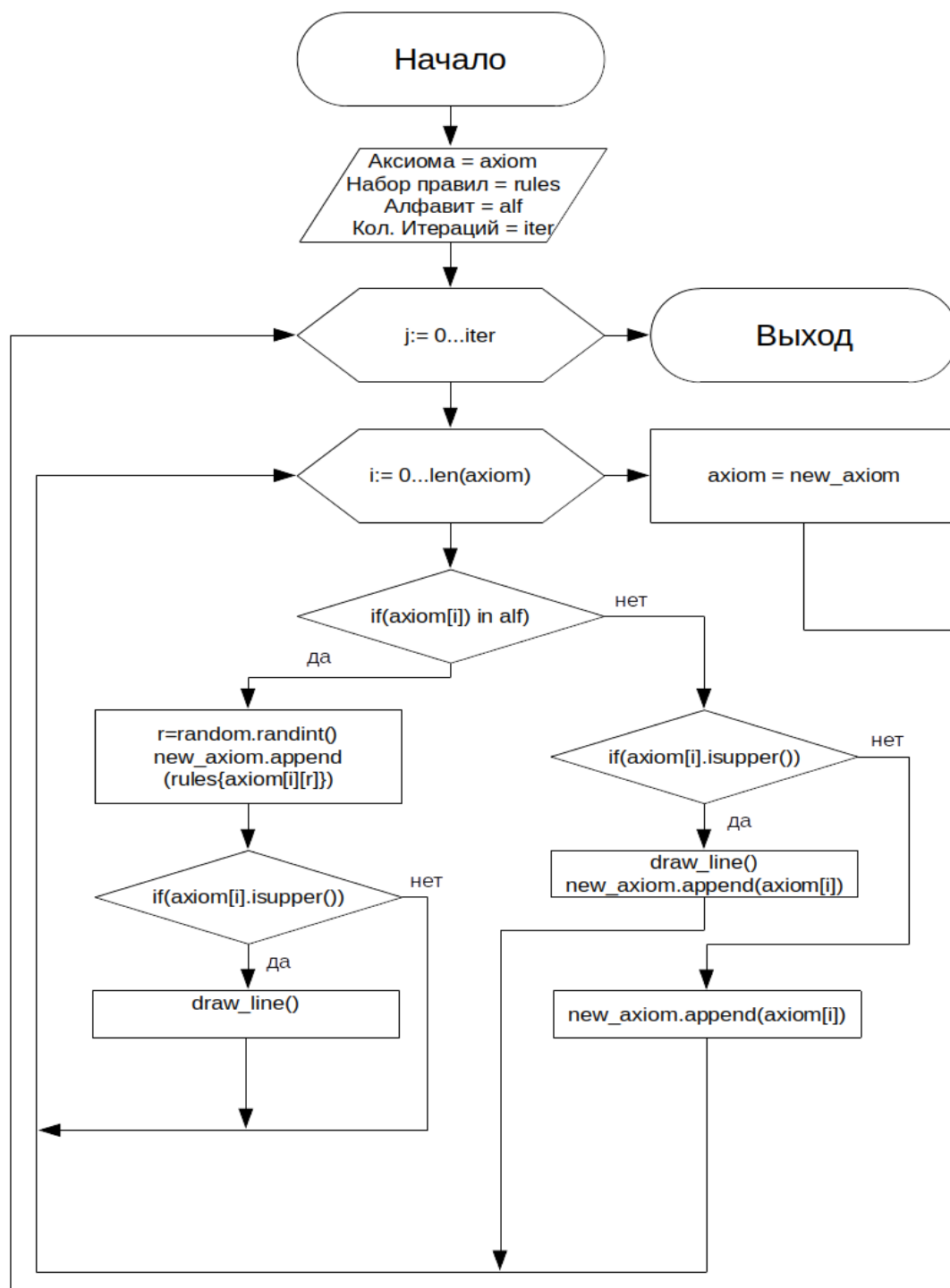


Рисунок 2.1 – Схема реализации DOL-системы

Изм.	Лист	№ докум.	Подпись	Дата

2.2 Схема реализации стохастических контекстно-свободных L-систем

На рисунке 2.2 представлена реализация алгоритма работы стохастических систем Линденмайера.

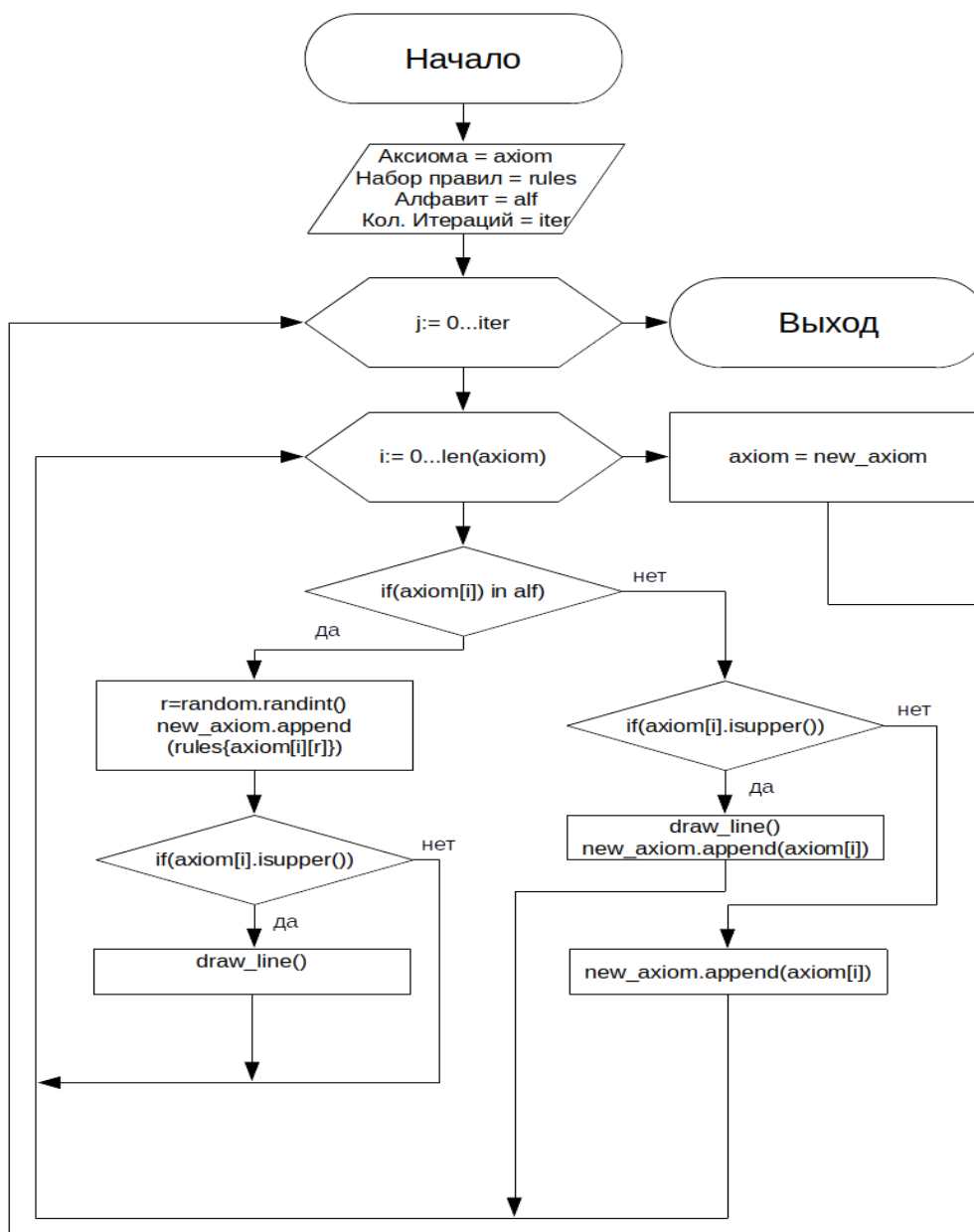


Рисунок 2.2 – Схема реализации стохастической контекстно-свободной L-системы

2.3 Выводы по главе

Представленные алгоритмы позволяют решить поставленную задачу, а именно использовать различные виды L-систем и визуализировать их.

Изм.	Лист	№ докум.	Подпись	Дата

3 Программное конструирование

3.1 Выбор среды и языка программирования

Для решения поставленной задачи выбран язык Python, который является языком высокого уровня и позволяет быстро, а также эффективно создавать приложения, графики и расчёты.

К достоинствам данного языка можно отнести:

- наличие большого числа подключаемых к программе модулей;
- эффективность, скорость разработки;
- кроссплатформенность.

В качестве среды разработки выбран PyCharm Professional Edition, это продукт от компании JetBrains, который включает в себя интегрированную среду разработки программного обеспечения и ряд других преимуществ. Данный продукт позволяет разрабатывать как консольные приложения, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения и веб-службы.

PyCharm содержит редактор и отладчик исходного кода. В остальные встраиваемые инструменты включены: контроль версий, удалённая разработка, дизайнер классов и дизайнер схем баз данных, возможность работы с библиотеками для научных вычислений.

PyCharm является одним из лидирующих продуктов на рынке программных средств, используемых для разработки программного обеспечения.

3.2 Описание программы

Программный код написан и оформлен на языке высокого уровня – Python в среде разработки PyCharm Professional Edition.

Для моделирования растительных объектов с помощью фрактальной геометрии, в качестве входных данных, вводятся (выбираются) следующие значения:

- угол поворота от 1° до 359° ;
- ширина пера от 1 px до 20px;
- количество итераций от 1 до 10;
- длина линии от 30 px до 500 px;
- аксиома;
- набор правил;
- начальные координаты (x,y);
- изменение длины линии;
- выбор цвета.

Рисунок 3.1 – Перечень вводимых значений.

После ввода всех данных необходимо нажать кнопку «Начать». Далее выполняется отрисовка системы, начиная с аксиомы, являющейся её начальным состоянием, параллельно применяя множество порождающих правил над аксиомой с каждым шагом. При этом с каждой последующей итерацией возможно изменение длины линий на 0%, 25%, 50%, 75%.

В результате ввода правил и аксиомы L-системы, исходных параметров и последующего выполнения программы на экран выводятся: номер итерации, изображение, соответствующее эволюции системы на данном этапе итерации.

3.3 Выводы по главе

Реализованная программа позволяет интерпретировать различные системы в геометрические объекты с заданной шириной пера, длиной линии и цветом.

4 Тестирование программы

4.1 Результаты работы программного средства

Для проверки работы программного средства рассмотрены примеры восьми различных растительных объектов:

1. Для моделирования D0L-системы с аксиомой F и правилом $F \rightarrow FF-[-F+F+F]+[+F-F-F]$. Построение проводится со следующими параметрами: углом поворота 21° , шириной пера $2r_x$, количеством итераций 4, длиной линий $30r_x$, цвет - зелёный. Результат построения данной системы продемонстрирован на рисунках 4.1 - 4.4.

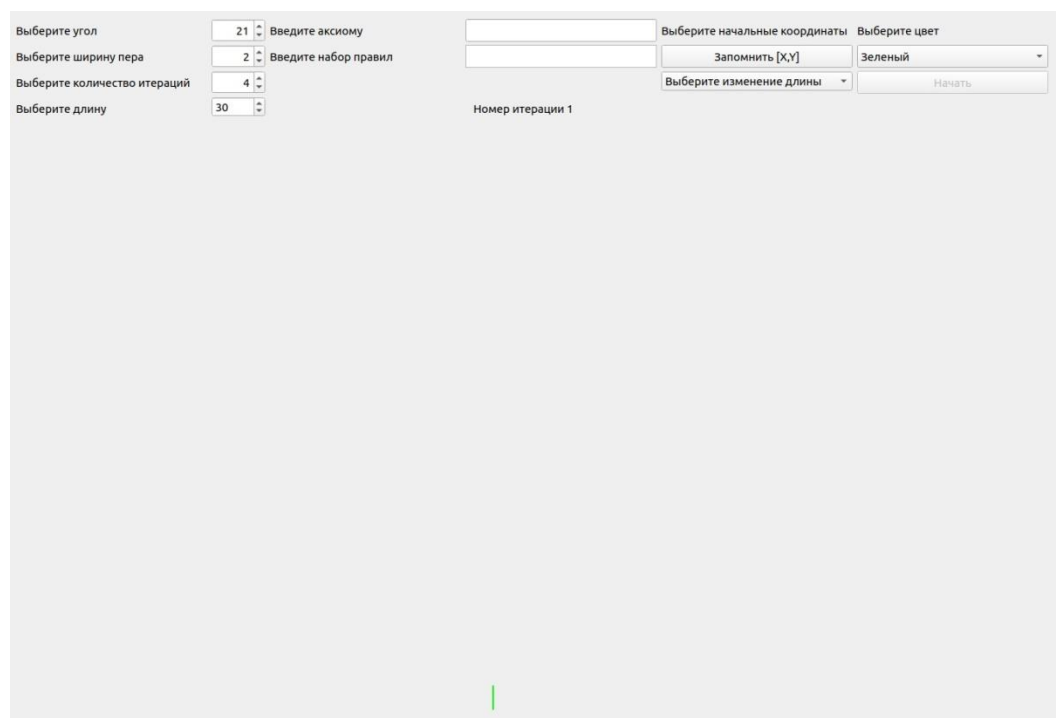


Рисунок 4.1 – Итерация №1

Изм.	Лист	№ докум.	Подпись	Дата

УП.920000.000

Лист

20

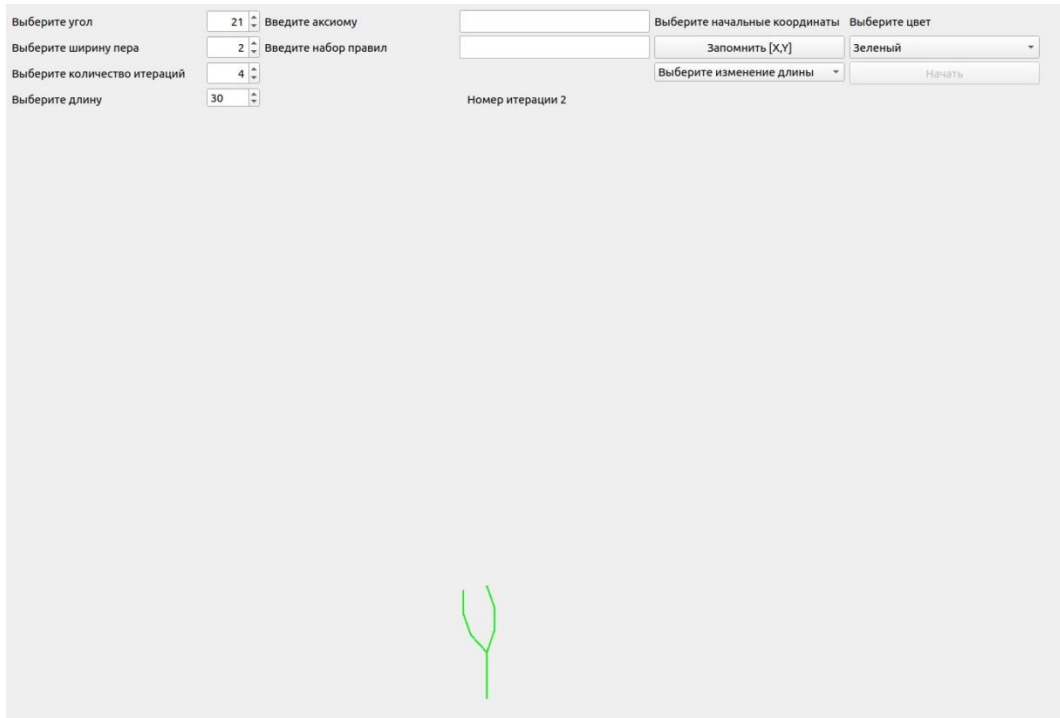


Рисунок 4.2 – Итерация №2

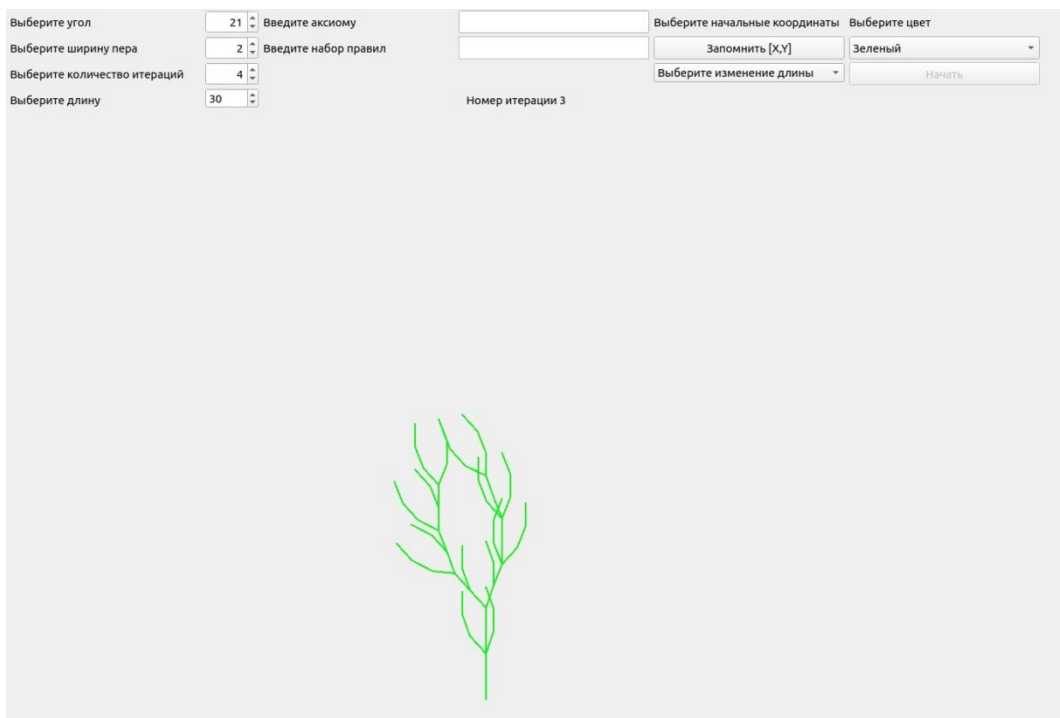


Рисунок 4.3 – Итерация №3

Изм.	Лист	№ докум.	Подпись	Дата

УП.920000.000

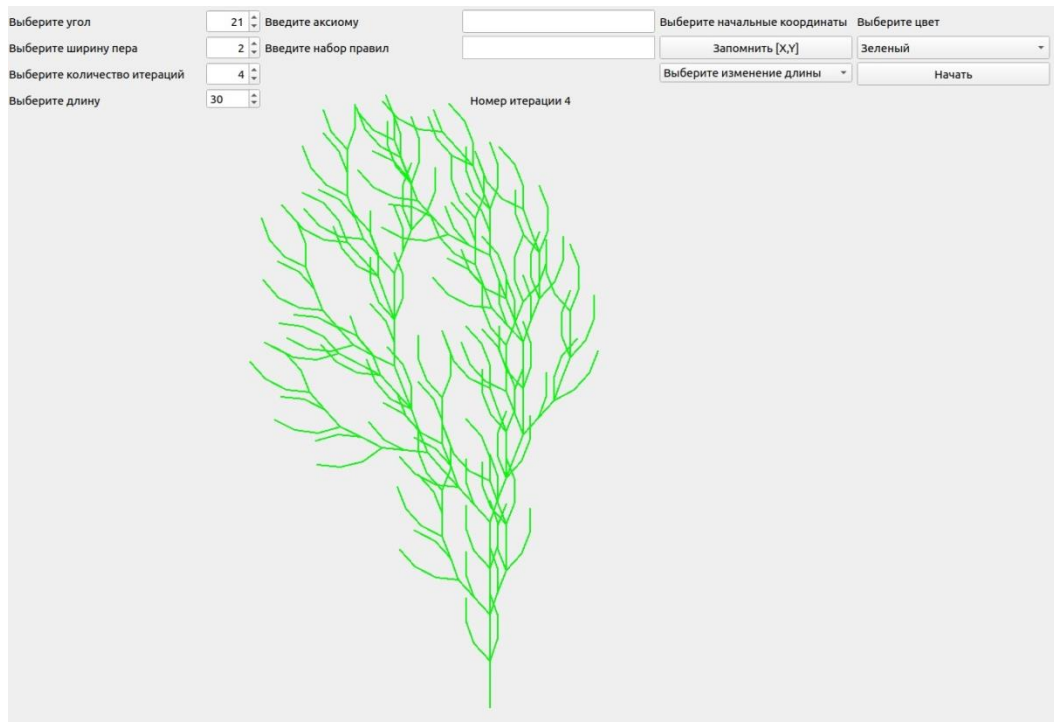


Рисунок 4.4 – Итерация №4

Наблюдая за эволюцией аксиомы под действием правил можно заметить, что рост и ветвление данной структуры происходят в экспоненциальном порядке.

- Для моделирования D0L-системы «дерево Пифагора» с аксиомой X и правилами: $F \rightarrow FF$; $X \rightarrow F[+X]-X$. Построение проводится со следующими параметрами: углом поворота 45° , шириной пера $3r_x$, количеством итераций 8, длиной линий $50r_x$, чёрного цвета, с уменьшением длины линий на 25%. Результат построения дерева Пифагора продемонстрирован на рисунке 4.5.

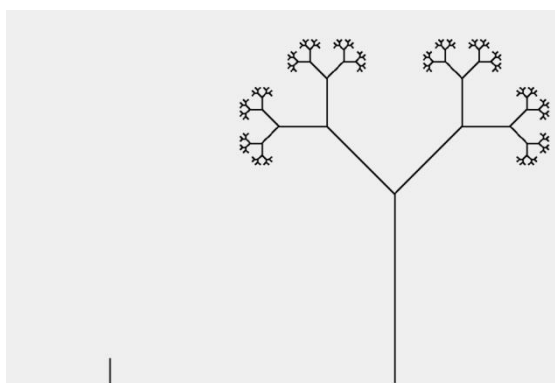


Рисунок 4.5 – Итерации №1,8

Наблюдая за эволюцией аксиомы под действием правил можно заметить, что происходит не только ветвление, но и рост системы в целом.

Изм.	Лист	№ докум.	Подпись	Дата

УП.920000.000

Лист

22

3. Для моделирования D0L-системы с аксиомой X и правилами: $F \rightarrow FF$; $X \rightarrow F[+X][-X]FX$. Построение проводится со следующими параметрами: углом поворота 20° , шириной пера $2r_x$, количеством итераций 5, длиной линий $30r_x$, синего цвета, с уменьшением длины линий на 25%. Результат построения данной системы продемонстрирован на рисунке 4.6.

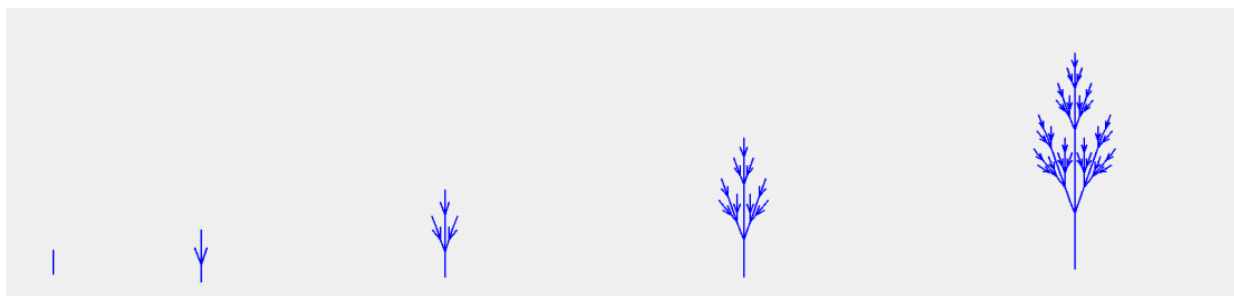


Рисунок 4.6 – Итерации №1-5

Наблюдая за эволюцией аксиомы под действием правил можно заметить, что ветвление в данном примере происходит зеркально.

4. Для моделирования D0L-системы с аксиомой X и правилами: $F \rightarrow FF$; $X \rightarrow F-[[X]+X]+F[+FX]-X$. Построение проводится со следующими параметрами: углом поворота 20° , шириной пера $2r_x$, количеством итераций 5, длиной линий $50r_x$, цвета радуги, с уменьшением длины линий на 25%. Результат построения данной системы продемонстрирован на рисунке 4.7.



Рисунок 4.7 – Итерации №1-5

Данное цветовое решение позволяет наблюдать, как растёт размер цепочки букв при каждом шаге работы программы.

Изм.	Лист	№ докум.	Подпись	Дата

5. Для моделирования стохастической системы с аксиомой F и правилами: $F \rightarrow F[+F]F[-F]F$ с вероятностью применения 40%; $F \rightarrow F[+F]F$ – 30%; $F \rightarrow F[-F]F$ – 30%. Данная система при каждом запуске программы демонстрирует новые результаты. Построения проводятся со следующими параметрами: углом поворота 20° , шириной пера $2r_x$, количеством итераций 5, длиной линий $50r_x$, цвета: синий, фиолетовый, радуги, для каждого из запусков, соответственно, с уменьшением длины линий на 25%. Результаты построения данной системы продемонстрированы на рисунке 4.8 для трёх запусков.

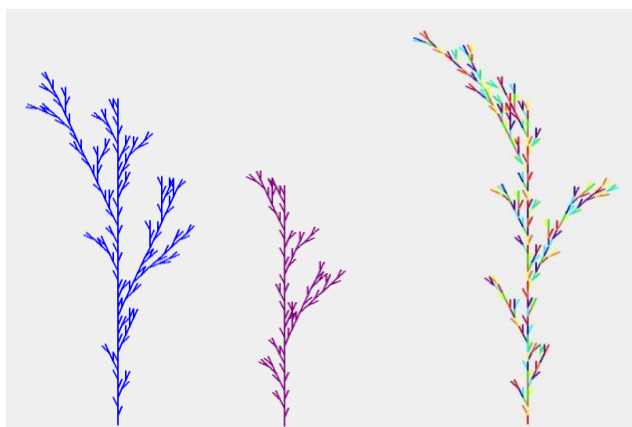


Рисунок 4.8 – Результат итерации №5 для запусков № 1-3

Так как в представленном примере правила применяются с определённой вероятностью при каждом замещении, результаты итераций имеют схожую структуру, но существенно различаются в деталях.

6. Для моделирования стохастической системы с аксиомой F и правилами: $F \rightarrow F[+F]F[-F]F$ с вероятностью применения 20%; $F \rightarrow F[+F]F$ – 40%; $F \rightarrow F[-F]F$ – 40%. Данная система при каждом запуске программы демонстрирует новые результаты. Построения проводятся со следующими параметрами: углом поворота 20° , шириной пера $2r_x$, количеством итераций 5, длиной линий $50r_x$, цвета: синий, фиолетовый, радуги, для каждого из запусков, соответственно, с уменьшением длины линий на 25%. Результаты построения данной системы продемонстрированы на рисунке 4.9 для трёх различных запусков.

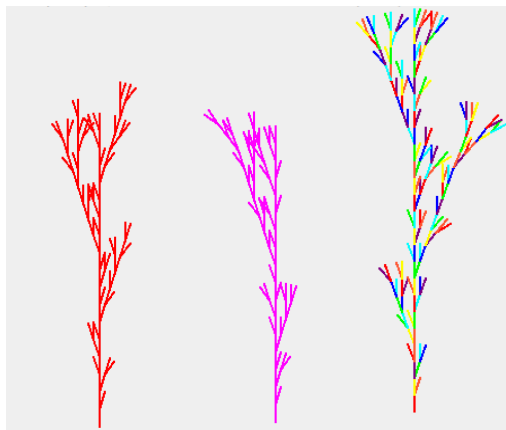


Рисунок 4.9 – Результат итерации №5 для запусков № 1-3

Наблюдая результаты нескольких запусков программного средства, можно сделать вывод о том, как стохастический характер системы изменяет внешний вид растительной структуры, делая её более реалистичной.

4.2 Выводы по главе

Протестировав программу на различных L-системах, можно сделать вывод о том, что варьируя правила и параметры системы, можно моделировать разнообразные растительные структуры.

Изм.	Лист	№ докум.	Подпись	Дата

УП.920000.000

Лист

25

ЗАКЛЮЧЕНИЕ

В ходе выполнения задания учебной практики рассмотрены различные фракталы и L-системы, реализованы алгоритмы решения поставленной задачи и разработано соответствующее программное средство.

Можно заметить, что различные виды L-систем имеют свои преимущества и недостатки. Детерминированные контекстно-свободные системы Линденмайера (DOL) являются более простым и удобным средством моделирования, но в то же время выразительные возможности этих систем несколько ограничены.

При моделировании растительных объектов нужно учитывать, что растения принадлежащие одному виду, схожи по структуре, но могут иметь существенные различия внешнего вида в целом или в некоторых деталях. Так как влияние внешней среды носит стохастический характер, для придания моделям большей реалистичности вводят разные правила для одной и той же буквы применяемые с разной долей вероятности.

Изучив теоретическую информацию о системах Линденмайера, разработав и протестировав программное средство на различных L-системах, можно сделать вывод о том, что с помощью небольшого числа правил, возможно моделирование сложных фрактальных объектов.

					УП.920000.000	Лист
Изм.	Лист	№ докум.	Подпись	Дата		26

СПИСОК ИСОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Балханов В.К. Основы фрактальной геометрии и фрактального исчисления – Улан-Удэ: Изд-во Бурятского госуниверситета, 2013. - 224 с.
2. Ершов Н. М. Системы Линденмайера. Естественные модели параллельных вычислений [Электронный ресурс] / Н. М. Ершов, Н. Н. Попова. – Режим доступа: [http:// naturalmodels.blogspot.com/2013/03/4.html](http://naturalmodels.blogspot.com/2013/03/4.html)
3. Нирхаус Г. Хаос и самоподобие. // Нелинейная динамика. - № 1. - том 7, 2011. - С. 153-175.
4. Developmental models of herbaceous plants [Электронный ресурс] – Режим доступа: [http:// algorithmicbotany.org/papers/abop/abop-ch3.pdf](http://algorithmicbotany.org/papers/abop/abop-ch3.pdf)
5. Prusinkiewicz P. Lindenmayer Systems, Fractals, and Plants / P. Prusinkiewicz, J. Hanan – New York: Springer-Verlag, 2016. – 125 p.
6. Prusinkiewicz P. The Algorithmic Beauty of Plants / P. Prusinkiewicz, A. Lindenmayer. – New York: Springer-Verlag, 2004. – 240 p.
7. Smith A.R. Plants, fractals, and formal languages // Computer Graphics, 1984, vol. 18, pp. 1–10.

ПРИЛОЖЕНИЕ А

Исходный код программного средства

```
from PyQt5 import QtGui,QtWidgets,QtCore
import sys
import time
import math
import random
import re

re_var = re.compile("\w+")
re_multi = re.compile("\w+*\w+")
re_division = re.compile("\w+/\w+")
re_plus = re.compile("\w+\+\w+")
re_minus = re.compile("\w+-\w+")
hooks = re.compile("\w\(.+\)")

width = 1400
height = 1000

dic = dict()
dic_chance = dict()
dic_options = dict()
dic_terms = dict()
dic_len_options = dict()

doc = """Программный модуль рисует все буквы в верхнем регистре и не рисует
в нижнем, задать правило можно написав { букву -> правило }.
Для изменения угла перед буквой нужно написать "+" или "-", для
запоминания координат используются квадратные скобки.
Задать вероятность можно в правиле, для этого нужно дописать "|" и число от
1 до 100 - означающее соответствующую вероятность, например { букву ->
правило | 73 }.
"""
```

					УП.920000.000	Лист
Изм.	Лист	№ докум.	Подпись	Дата		28

```

class Window(QtWidgets.QWidget):
    def __init__(self):
        QtWidgets.QWidget.__init__(self)
        self.paint_event = False
        self.set_position = False
        self.width_brush = 1
        self.stack_paint = []
        self.start_position = [width // 2, height // 2]
        self.number_color = 0
        self.color = 0
        self.length = 0
        self.change_length = 0
        self.position = [0, 0]
        self.resize(width, height)
        self.setWindowTitle("Фрактальная геометрия ")
        self.doc_label = QtWidgets.QLabel(doc)
        self.first_angle_label = QtWidgets.QLabel("Выберите угол")
        self.first_width_label = QtWidgets.QLabel("Выберите ширину пера")
        self.first_iteration_label = QtWidgets.QLabel("Выберите количество
итераций")
        self.first_rule_label = QtWidgets.QLabel("Введите набор правил")
        self.first_axiom_label = QtWidgets.QLabel("Введите аксиому")
        self.first_choise_position_label = QtWidgets.QLabel("Выберите начальные
координаты")
        self.first_choise_color_label = QtWidgets.QLabel("Выберите цвет")
        self.first_length_label = QtWidgets.QLabel("Выберите длину")
        self.first_iteration_number_label = QtWidgets.QLabel()
        self.first_angle_box = QtWidgets.QSpinBox()
        self.first_width_box = QtWidgets.QSpinBox()
        self.first_length_box = QtWidgets.QSpinBox()
        self.first_iteration_box = QtWidgets.QSpinBox()

```

					УП.920000.000	Лист
Изм.	Лист	№ докум.	Подпись	Дата		29

```

self.first_rules_edit = QtWidgets.QTextEdit()
self.first_axiom_edit = QtWidgets.QLineEdit()
self.first_choise_position_btn = QtWidgets.QPushButton("Запомнить
[X,Y]")
self.first_choise_color_box = QtWidgets.QComboBox()
self.first_change_length_box = QtWidgets.QComboBox()
self.button_start = QtWidgets.QPushButton("Начать")
self.first_angle_label.setFixedSize(250, 30)
self.first_width_label.setFixedSize(250, 30)
self.first_iteration_label.setFixedSize(250, 30)
self.first_rule_label.setFixedSize(250, 30)
self.first_axiom_label.setFixedSize(250, 30)
self.first_iteration_number_label.setFixedSize(250, 30)
self.first_choise_position_label.setFixedSize(250, 30)
self.first_angle_box.setFixedSize(70, 30)
self.first_width_box.setFixedSize(70, 30)
self.first_iteration_box.setFixedSize(70, 30)
self.button_start.setFixedSize(250, 30)
self.first_rules_edit.setFixedSize(250, 30)
self.first_axiom_edit.setFixedSize(250, 30)
self.first_choise_position_btn.setFixedSize(250, 30)
self.first_choise_color_label.setFixedSize(250, 30)
self.first_choise_color_box.setFixedSize(250, 30)
self.first_length_label.setFixedSize(250, 30)
self.first_angle_box.setAlignment(QtCore.Qt.AlignRight)
self.first_width_box.setAlignment(QtCore.Qt.AlignRight)
self.first_iteration_box.setAlignment(QtCore.Qt.AlignRight)
self.first_angle_box.setWrapping(True)
self.first_width_box.setWrapping(True)
self.first_iteration_box.setWrapping(True)

```

Изм.	Лист	№ докум.	Подпись	Дата

```

self.first_angle_box.setRange(1, 359)
self.first_width_box.setRange(1, 20)
self.first_iteration_box.setRange(1, 10)
self.first_length_box.setRange(10, 500)
self.first_angle_box.setValue(1)
self.first_width_box.setValue(1)
self.first_iteration_box.setValue(2)
self.first_length_box.setValue(30)
self.first_choose_color_box.addItem(
    ["Черный", "Серый", "Красный", "Зеленый", "Синий", "Голубой",
    "Фиолетовый", "Розовый", "Желтый", "Радуга",
    "Произвольный"])
self.first_change_length_box.addItem(
    ["Выберите изменение длины", "Не изменяется", "Каждая итерация
1/4", "Каждая итерация 1/2",
    "Каждая итерация 3/4"])
self.first_grid = QtWidgets.QGridLayout()
self.first_grid.setAlignment(QtCore.Qt.AlignLeft | QtCore.Qt.AlignTop)
self.first_grid.addWidget(self.first_angle_label, 0, 0)
self.first_grid.addWidget(self.first_width_label, 1, 0)
self.first_grid.addWidget(self.first_iteration_label, 2, 0)
self.first_grid.addWidget(self.first_angle_box, 0, 1)
self.first_grid.addWidget(self.first_width_box, 1, 1)
self.first_grid.addWidget(self.first_iteration_box, 2, 1)
self.first_grid.addWidget(self.first_rule_label, 1, 2)
self.first_grid.addWidget(self.first_rules_edit, 1, 3)
self.first_grid.addWidget(self.first_axiom_label, 0, 2)
self.first_grid.addWidget(self.first_axiom_edit, 0, 3)
self.first_grid.addWidget(self.button_start, 2, 5)
self.first_grid.addWidget(self.first_iteration_number_label, 3, 3)

```

Изм.	Лист	№ докум.	Подпись	Дата

```

self.first_grid.addWidget(self.first_choise_position_label, 0, 4)
self.first_grid.addWidget(self.first_choise_position_btn, 1, 4)
self.first_grid.addWidget(self.first_choise_color_label, 0, 5)
self.first_grid.addWidget(self.first_choise_color_box, 1, 5)
self.first_grid.addWidget(self.first_length_label, 3, 0)
self.first_grid.addWidget(self.first_length_box, 3, 1)
self.first_grid.addWidget(self.first_change_length_box, 2, 4)
self.first_vertical_layout = QtWidgets.QVBoxLayout()
self.first_vertical_layout.addLayout(self.first_grid)
self.first_vertical_layout.addWidget(self.doc_label)
self.setLayout(self.first_vertical_layout)
self.button_start.clicked.connect(self.start)
self.first_choise_position_btn.clicked.connect(self.remember_position)
self.show()

```

def start(self):

```

self.doc_label.setText("")
self.button_start.setDisabled(True)
angle = int(self.first_angle_box.text())
self.width_brush = int(self.first_width_box.text())
iterations = int(self.first_iteration_box.text())
axiom = self.first_axiom_edit.text()
self.color = self.first_choise_color_box.currentIndex()
self.length = int(self.first_length_box.text())
self.change_length = self.first_change_length_box.currentIndex()
rules = self.first_rules_edit.toPlainText()
dic.clear()
dic_chance.clear()
dic_options.clear()
dic_terms.clear()
dic_len_options.clear()

```

Изм.	Лист	№ докум.	Подпись	Дата

УП.920000.000

Лист

32


```

try:
    self.get_rules(rules)
    self.repaint()
except:
    pass
self.paint(angle, axiom, iterations)
self.button_start.setDisabled(False)
def paint(self, angle, axiom, iterations):
    stack = []
    stack_angle = []
    self.new_position = [0, 0]
    new_axiom = dict()
    for i in range(iterations):
        self.position = self.start_position[:]
        turn = -90
self.first_iteration_number_label.setText(" Номер итерации {}".format(i + 1))
        self.stack_paint = []
        step = 0
        while step < len(axiom):
            if axiom[step] == '+':
                if turn + angle <= 360:
                    turn += angle
                else:
                    turn = (turn + angle) - 360
                new_axiom[step] = axiom[step]
            elif axiom[step] == '-':
                if turn - angle >= 0:
                    turn -= angle
                else:
                    turn = (turn - angle) + 360
                new_axiom[step] = axiom[step]

```

```

elif axiom[step] == '[':
    stack.append(self.position[:])
    new_axiom[step] = axiom[step]
    stack_angle.append(turn)
elif axiom[step] == ']':
    self.position = stack.pop()
    new_axiom[step] = axiom[step]
    turn = stack_angle.pop()
else:
    if axiom[step] not in dic_chance.keys() and axiom[step] in dic.keys():
        new_axiom[step] = dic[axiom[step]]
    elif axiom[step] in dic_chance.keys():
        number = random.randint(0,99)
        for k in range(len(dic_chance[axiom[step]])):
            if number in dic_chance[axiom[step]][k]:
                new_axiom[step] = dic[axiom[step]][k]
    else:
        change = False
        for w in dic.keys():
            remember = w[:]
            w = w.split('<')
            if len(w) == 1:
                w = w[0].split('>')
            if len(w) == 2:
                if w[0] == axiom[step]:
                    if len(axiom)-(step+1):
                        change = remember
                    for x in range(len(w[1])):
                        if w[1][x] != axiom[step+(x+1)]:
                            change = False

```

Изм.	Лист	№ докум.	Подпись	Дата

УП.920000.000

Лист

34

```

elif len(w) == 2:
    new_w = w[1].split('>')
    if len(new_w) == 2:
        w[1] = new_w[0]
        w.append(new_w[1])
        if axiom[step] == w[1]:
            if len(w[0])-1 <= step and len(w[2]) <= len(axiom)-(step+1):
                change = remember
                for x in range(len(w[0])):
                    if w[0][len(w[0])-(x+1)] != axiom[step-(x+1)]:
                        change = False
                for x in range(len(w[2])):
                    if w[2][x] != axiom[step+(x+1)]:
                        change = False
            else:
                w[1] = new_w[0]
                if axiom[step] == w[1]:
                    if len(w[0])-1 <= step and step!=0:
                        change = remember
                        for x in range(len(w[0])):
                            if w[0][len(w[0])-(x+1)] != axiom[step-(x+1)]:
                                change = False
                    if change == remember:
                        break
            if change:
                new_axiom[step] = dic[change]
            else:
                new_axiom[step] = axiom[step]
self.new_position[0] = self.position[0] + self.length * math.cos(math.radians(turn))
self.new_position[1] = self.position[1] + self.length * math.sin(math.radians(turn))
if axiom[step].isupper():

```

					УП.920000.000	Лист
Изм.	Лист	№ докум.	Подпись	Дата		35

```

        self.stack_paint.append(
            [self.position[0], self.position[1], self.new_position[0],
self.new_position[1]])
        self.paint_event = True
        self.repaint()
        self.paint_event = False
        self.position = [self.new_position[0], self.new_position[1]]
        step+=1
axiom = ""
if self.change_length == 2:
    self.length = int(self.length * 0.25)
elif self.change_length == 3:
    self.length = int(self.length * 0.5)
elif self.change_length == 4:
    self.length = int(self.length * 0.75)
for j in new_axiom.values():
    axiom += j
if i == iterations - 1:
    self.paint_event = True
    self.repaint()
else:
    time.sleep(5)
    self.repaint()
    self.paint_event = False
def remember_position(self):
    self.set_position = True
def get_rules(self, rules):
    rules = rules.split("\n")
    chance = [x for x in range(100)]
    for i in range(len(rules)):
        choice = []

```

Изм.	Лист	№ докум.	Подпись	Дата

УП.920000.000

Лист

36

```

string = ""
for j in range(len(rules[i])):
    if rules[i][j]!=' ':
        string+=rules[i][j]
rules[i] = string[:]
rule = rules[i].split("->")
rule[1] = rule[1].split("|")
if len(rule[1]) == 1:
    hook = hooks.findall(rule[0])
    if hook:
        rule[0] = rule[0].split(':')
        if len(rule[0]) == 1:
            dic_options[rule[0][0]] = rule[1][0]
            dic_len_options[rule[0][0][0]] = len(rule[0][0])-1
        else:
            if rule[0][0] not in dic_options.keys():
                dic_options[rule[0][0]] = [rule[1][0]]
                dic_terms[rule[0][0]] = [rule[0][1]]
                dic_len_options[rule[0][0][0]] = len(rule[0][0])-1
            else:
                dic_options[rule[0][0]].append(rule[1][0])
                dic_terms[rule[0][0]].append(rule[0][1])
        else:
            dic[rule[0]] = rule[1][0]
    else:
        if len(rule[0]) == 1:
            if rule[0][0] not in dic.keys():
                dic[rule[0]] = [rule[1][0]]
                for j in range(int(rule[1][1])):
                    number = random.choice(chance)
                    chance.remove(number)

```

Изм.	Лист	№ докум.	Подпись	Дата

УП.920000.000

Лист

37

```

        choice.append(number)
        dic_chance[rule[0]] = [choice]
    else:
        dic[rule[0]].append(rule[1][0])
        for j in range(int(rule[1][1])):
            number = random.choice(chance)
            chance.remove(number)
            choice.append(number)
        dic_chance[rule[0]].append(choice)

def set_color(self):
    if self.color == 0:
        self.pen = QtGui.QPen(QtCore.Qt.black, self.width_brush)
    elif self.color == 1:
        self.pen = QtGui.QPen(QtCore.Qt.gray, self.width_brush)
    elif self.color == 2:
        self.pen = QtGui.QPen(QtCore.Qt.red, self.width_brush)
    elif self.color == 3:
        self.pen = QtGui.QPen(QtCore.Qt.green, self.width_brush)
    elif self.color == 4:
        self.pen = QtGui.QPen(QtCore.Qt.blue, self.width_brush)
    elif self.color == 5:
        self.pen = QtGui.QPen(QtCore.Qt.cyan, self.width_brush)
    elif self.color == 6:
        self.pen = QtGui.QPen(QtCore.Qt.darkMagenta, self.width_brush)
    elif self.color == 7:
        self.pen = QtGui.QPen(QtCore.Qt.magenta, self.width_brush)
    else:
        self.pen = QtGui.QPen(QtCore.Qt.yellow, self.width_brush)

def rainbow(self, number_color):
    if number_color == 0 or number_color == "красный":
        self.pen = QtGui.QPen(QtCore.Qt.red, self.width_brush)

```

					УП.920000.000	Лист 38
Изм.	Лист	№ докум.	Подпись	Дата		

```

elif number_color == 1 or number_color == "оранжевый":
    self.pen = QtGui.QPen(QtGui.QColor.fromRgb(255, 87, 51),
self.width_brush)

elif number_color == 2 or number_color == "желтый":
    self.pen = QtGui.QPen(QtGui.QColor.yellow, self.width_brush)

elif number_color == 3 or number_color == "зеленый":
    self.pen = QtGui.QPen(QtGui.QColor.green, self.width_brush)

elif number_color == 4 or number_color == "голубой":
    self.pen = QtGui.QPen(QtGui.QColor.cyan, self.width_brush)

elif number_color == 5 or number_color == "синий":
    self.pen = QtGui.QPen(QtGui.QColor.blue, self.width_brush)

elif number_color == 6 or number_color == "фиолетовый":
    self.pen = QtGui.QPen(QtGui.QColor.darkMagenta, self.width_brush)

def paintEvent(self, QPaintEvent):
    self.painter = QtGui.QPainter()
    if self.paint_event:
        self.number_color = 0
        self.painter.begin(self)
        if self.color <= 8:
            self.set_color()
            self.painter.setPen(self.pen)
            for i in range(len(self.stack_paint)):
                if self.stack_paint[i][0] > 0 and self.stack_paint[i][0] < width and
self.stack_paint[i][1] > 150 and self.stack_paint[i][1] < height:
                    self.painter.drawLine(self.stack_paint[i][0], self.stack_paint[i][1],
self.stack_paint[i][2],
                                self.stack_paint[i][3])
            elif self.color == 9:
                for i in range(len(self.stack_paint)):
                    self.rainbow(self.number_color)
                    self.number_color += 1

```

Изм.	Лист	№ докум.	Подпись	Дата

```

        if self.number_color == 7:
            self.number_color = 0
            self.painter.setPen(self.pen)
            if self.stack_paint[i][0] > 0 and self.stack_paint[i][0] < width and
self.stack_paint[i][
                1] > 150 and self.stack_paint[i][1] < height:
                    self.painter.drawLine(self.stack_paint[i][0], self.stack_paint[i][1],
self.stack_paint[i][2],
                                self.stack_paint[i][3])
        else:
            for i in range(len(self.stack_paint)):
                self.color = random.randint(0, 8)
                self.set_color()
                self.painter.setPen(self.pen)
                self.color = 10
                if self.stack_paint[i][0] > 0 and self.stack_paint[i][0] < width and
self.stack_paint[i][
                    1] > 150 and self.stack_paint[i][1] < height:
                        self.painter.drawLine(self.stack_paint[i][0], self.stack_paint[i][1],
self.stack_paint[i][2],
                                    self.stack_paint[i][3])
            self.painter.end()
    def mousePressEvent(self, QMouseEvent):
        if self.set_position:
            self.start_position[0] = QMouseEvent.x()
            self.start_position[1] = QMouseEvent.y()
            self.set_position = False
app = QtWidgets.QApplication(sys.argv)
window = Window()
sys.exit(app.exec_())

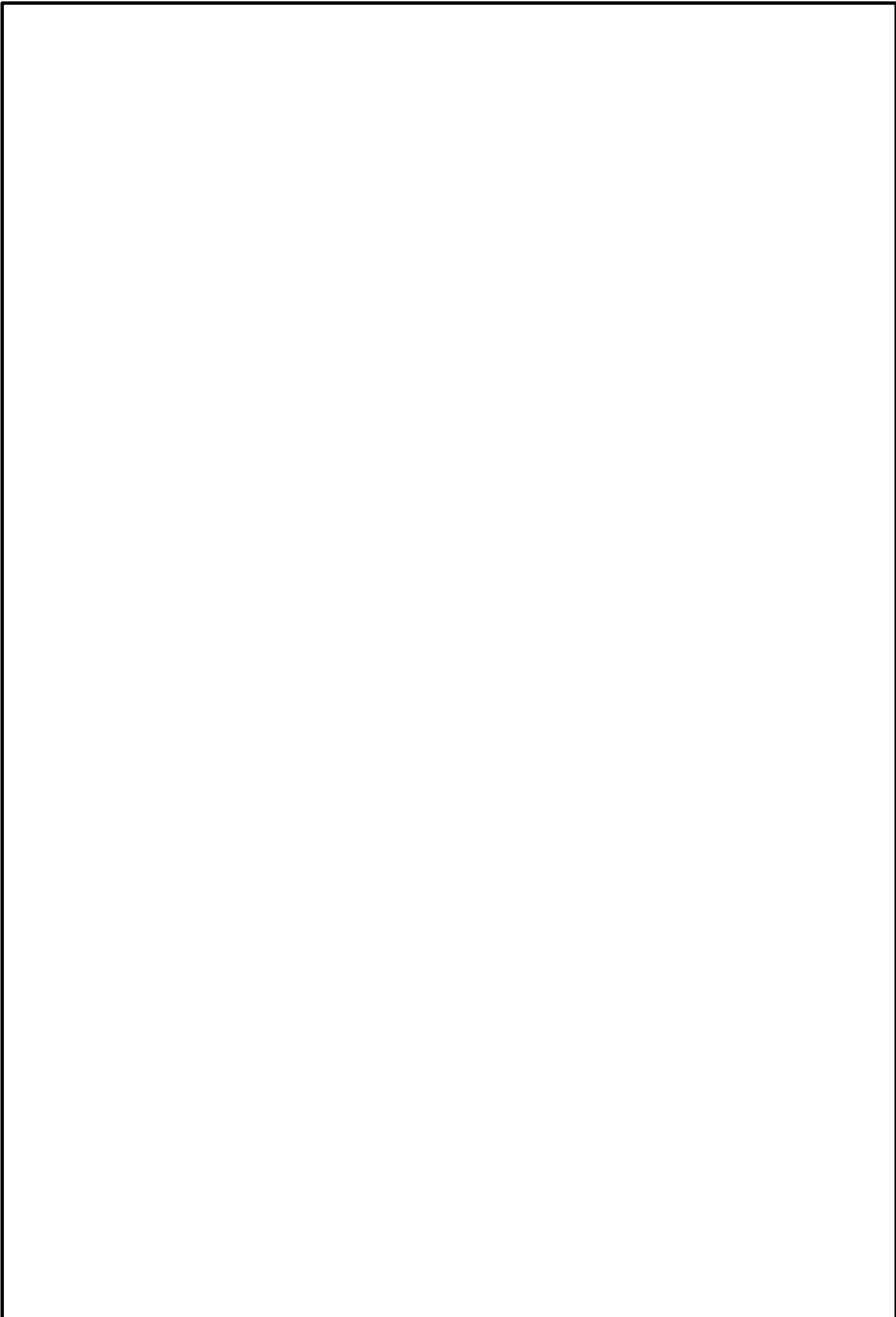
```

Изм.	Лист	№ докум.	Подпись	Дата

УП.920000.000

Лист

40



					УП.920000.000	Лист
Изм.	Лист	№ докум.	Подпись	Дата		41