

Федеральное государственное бюджетное образовательное учреждение высшего  
образования «Волгоградский государственный технический университет»

На правах рукописи



Нikitin Никита Андреевич

**МОДЕЛИ, МЕТОДЫ И СРЕДСТВА КОМПЬЮТЕРНОГО  
СИНТЕЗИРОВАНИЯ МУЗЫКИ ПО ЦВЕТОВОМУ ИЗОБРАЖЕНИЮ**

Специальность 2.3.1. «Системный анализ, управление и обработка информации»

**ДИССЕРТАЦИЯ**  
на соискание ученой степени  
кандидата технических наук

Научный руководитель:  
доктор технических наук, доцент  
Орлова Юлия Александровна

Волгоград – 2022

## Оглавление

Введение.....	4
1 Современные подходы к автоматизированному синтезу звуковых последовательностей по цветовому изображению .....	10
1.1 Компьютерные средства генерирования музыкальных композиций по цветовому изображению.....	10
1.2 Алгоритмические методы построения звуковых последовательностей .	19
1.3 Методы синтеза звуков.....	29
1.4 Выбор нейронной сети для генерации музыкальных композиций.....	33
1.5 Выводы по первой главе.....	36
2 Комплексный метод синтезирования цельной музыкальной композиции по цветовому изображению с использованием нейронных сетей .....	38
2.1 Метод структурированного представления музыкальной композиции..	39
2.2 Моделирование музыкальных композиций и жанров.....	44
2.3 Метод воспроизведения звуков музыкальных инструментов по структурированному представлению музыкальной композиции .....	57
2.4 Выводы по второй главе.....	58
3 Алгоритмическое обеспечение системы синтезирования музыкальной композиции по цветовому изображению.....	60
3.1 Алгоритмы анализа изображений .....	60
3.2 Алгоритмы структурированного представления музыкальной композиции .....	64
3.3 Алгоритм синтеза звуков по структурированному описанию музыкальной композиции .....	74
3.4 Выводы по третьей главе .....	85
4 Реализация системы синтезирования музыкальной композиции по цветовому изображению.....	86
4.1 Архитектура системы .....	86

4.2 Интерфейс и функции системы .....	88
4.3 Язык программирования и используемые библиотеки.....	90
4.4 Процесс проектирования в нотации UML .....	92
4.5 Выводы по четвёртой главе .....	96
5 Вычислительные эксперименты по оцениванию музыкальных композиций и компьютерной системы .....	97
5.1 Тестирование системы.....	97
5.2 Модельный массив тестовых изображений .....	108
5.3 Экспертное оценивание синтезированных музыкальных композиций.	115
5.4 Анализ результатов тестирования и валидации .....	119
5.4.1 Анализ результатов тестирования.....	119
5.4.2 Анализ результатов валидации.....	120
5.5 Выводы по пятой главе.....	123
Заключение .....	124
Литература .....	127
ПРИЛОЖЕНИЕ А. СХЕМЫ СООТНЕСЕНИЯ ЦВЕТОВ И НОТ .....	139
ПРИЛОЖЕНИЕ Б. СВИДЕТЕЛЬСТВО О РЕГИСТРАЦИИ ПО .....	140
ПРИЛОЖЕНИЕ В. АКТ ВНЕДРЕНИЯ .....	141

## Введение

**Актуальность темы исследования.** Несмотря на все достижения в понимании творческих процессов, создание музыки не может проходить автоматически. Роль пользователя-композитора очень высока и можно лишь говорить об автоматизации этого процесса [1]. Передаваемая музыкой и картинами эмоциональность сложно распознаваема [2]. Сам процесс создания музыки, на данный момент, не поддаётся чёткой формализации, хотя и основывается на чётко определённых музыкальных правилах. Наибольших успехов автоматизация процесса написания и создания музыки достигла сравнительно недавно (в последние десятилетия), однако по большей части связана с изучением и повторением различных музыкальных стилей [3, 4].

В рамках данной работы предполагается автоматизация процесса создания музыки, путём автоматизированной генерации звуков по изображению. Другими словами, генерация звуков по изображению есть процесс преобразования изображения в одну или несколько последовательностей нот, с определённым основным тоном и длительностью [3, 5].

**Научная новизна** диссертационного исследования заключается в разработке оригинальных моделей, методов и средств компьютерного синтезирования цельной музыкальной композиции по цветовому изображению, включающие:

1. Новый комплексный метод синтезирования цельной музыкальной композиции по цветовому изображению, основанный на совместном использовании методов машинного обучения для создания мелодии и правилах построения музыкальной композиции для получения гармонии произведения, включающий модель соотнесения цветовых и музыкальных характеристик, модель структуры музыкальной композиции, модель жанров музыкальных композиций;

2. Новый метод структурированного представления музыкальной композиции, основанный на сопоставлении различных цветовых (оттенок, цветовая группа, яркость, насыщенность) и музыкальных (нота, музыкальный лад, октава, длительность) характеристик, позволяющий определить характеристики результирующей композиции по изображению;

3. Новый метод воспроизведения звуков музыкальных инструментов по структурированному представлению музыкальной композиции, использующий собранную автором коллекцию образцов записанных отдельных нот и позволяющий синтезировать музыку с высокой степенью реалистичности.

**Теоретическая значимость** работы заключается в том, что в результате анализа различных методов и подходов к автоматизированной генерации музыкальных композиций был разработан метод, основанный на традиционном и современном подходах, что позволяет учитывать как прошлые достижения в области алгоритмических композиций, так и использовать современные методы анализа данных в виде машинного обучения. Таким образом, разработанные алгоритмы и методы представляют собой некую финализацию развития данного направления, а также предлагают оригинальный метод создания композиций.

**Практическая значимость** работы заключается в том, что данную систему можно использовать для автоматизированной генерации фоновых композиций для рекламы, фильмов и компьютерных игр [3].

**Объектом исследования** является процесс компьютерного синтезирования музыкальных композиций [3].

**Предметом исследования** являются модели, методы и средства автоматизированного синтезирования музыкальной композиции по цветовому изображению [3].

**Гипотеза исследования** состоит в том, что автоматизация сочинения музыкальных композиций по цветовому изображению, основанная на совместном использовании методов искусственного интеллекта и теории музыки, позволит повысить мелодичность и гармоничность музыкальных композиций [3].

Описание работы по классификаторам:

ACM: CCS → Applied Computing → Arts and humanities → Sound and music computing

IEEE: Computers and information processing → Computer applications → Computer generated music

РФФИ: 07-3 АЛГОРИТМИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ → 07-346 Мультимедийные информационные системы

Сколково: Новые системы поиска, распознавания и обработки аудио, видео и графической информации → Новые решения для поиска, распознавания и обработки в аудио, видео и изображениях.

**Степень разработанности темы.** Исследованием данного направления занимались многие известные учёные, например, Ньютон, предложивший в 1704 году свою схему соотнесения цветов и нот [6], основанной на соотнесении длины волны звука и цвета. Также были подходы, основанные на интуиции (Л. Кастель 1730 год [6]), психологии, психофизике, и физиологии. Одной из первых комплексных работ в данном направлении стала работа J. L. Caivano в 1994 году, который помимо прямого соотнесения цветов и нот, выделил схему соотнесения цветовых и музыкальных характеристик [1]. Следующим развитием данного направления стала работа отечественного композитора и художника В. В. Афанасьева, который предложил связать отношения звуков и цветов в зависимости от того, в какой плоскости они представлены: мелодия, гармония или тональность [7]. Также, Xiaoying Wu по данному направлению была защищена магистерская диссертация в Канаде в 2008 году [5]. Затем, был целый ряд работ, направленных на автоматизацию создания музыкальных произведений с помощью методов машинного обучения. Например, в 2016 году была представлена работа Ф. Лиангом, которая описывает использование рекуррентных нейронных сетей для задачи генерации композиций, имитирующих стиль произведений Баха [8]. В 2017 была представлена разработка команды из Google Brains под названием Magenta, которая также описывает возможность использования сетей с долгой краткосрочной памятью для задачи генерации музыкальных композиций [9]. В 2020 году Ф. Роше была представлена

диссертация, описывающая различные аспекты генерации композиций с использованием методов машинного обучения [10].

В связи с этим **целью** является повышение эффективности компьютерного синтеза музыки по цветовому изображению за счет разработки моделей, методов и средств автоматизированного создания мелодичных и гармоничных музыкальных композиций с помощью машинного обучения [11].

Для достижения поставленной цели были определены следующие **задачи** [12]:

1. Провести анализ современных подходов и методов автоматизированного синтеза звуковых последовательностей по цветовому изображению.
2. Предложить критерии оценки и сравнения компьютерных средств генерирования музыкальных композиций по цветовому изображению, алгоритмических методов построения звуковых последовательностей, методов воспроизведения звучания музыкальных инструментов.
3. Разработать метод структурированного представления музыкальной композиции и модель соотнесения цветовых и музыкальных характеристик для получения параметров музыкальной композиции.
4. С использованием машинного обучения построить модельную базу композиций с различными музыкальными жанрами для выбора жанра музыкальной композиции.
5. Разработать метод воспроизведения звуков музыкальных инструментов по структурированному представлению музыкальной композиции.
6. Разработать комплексный метод синтезирования цельной музыкальной композиции по цветовому изображению с использованием нейронных сетей.
7. Разработанные модели, методы и алгоритмы реализовать в виде компьютерной системы, автоматизирующей процесс создания музыкальной композиции.

8. Провести тестирование и экспертное оценивание мелодичности и гармоничности синтезированных музыкальных композиций.

**Методы исследования**, использованные в работе: системный и сравнительный анализ, интеллектуальный анализ данных, метод перехода от абстрактного к конкретному, метод формализации, экспертные оценки, экспериментирование.

**Обоснованность и достоверность.** Результаты данной работы были опубликованы в различных рецензируемых научных журналах и представлены на соответствующих тематике конференциях. Кроме того, разработанные модели и методы легли в основу разработанной системы – веб-сайта для автоматизированной генерации композиции по цветовой гамме изображения.

**Апробация работы.** Работа была представлена на следующих научных конференциях: international scientific Conference on Artificial Intelligence and Digital Technologies in Technical Systems II-2021 (AIDTTS II-2021), Волгоград, 2021; Fourth International Scientific Conference «Intelligent Information Technologies for Industry» (IITI'19), Острава, Чехия, 2019; Third International Scientific Conference «Intelligent Information Technologies for Industry» (IITI'18), Сочи, 2018; XXIII Региональная конференция молодых исследователей Волгоградской области, Волгоград, 2018; International Conference «Information Technology and Nanotechnology», Самара, 2018.

По разработанным программным решениям было получено свидетельство о государственной регистрации программы для ЭВМ №2017616834.

Результаты диссертации были использованы при выполнении проектов Российского фонда фундаментальных исследований и администрации Волгоградской области № 18-07-00220, 19-07-00020, 19-47-343001, 19-47-340003, 19-47-340009, 19-47-340013, 19-47-343002, 19-37-90060, 20-07-00502

**Положения, выносимые на защиту**, по результатам работы:

1. Комплексный метод синтезирования цельной музыкальной композиции по цветовому изображению с использованием нейронных сетей, включающий модель соотнесения цветовых и музыкальных характеристик

музыкальной композиции, модель структуры музыкальной композиции, модель жанров музыкальных композиций.

2. Метод структурированного представления музыкальной композиции.

3. Метод воспроизведения звуков музыкальных инструментов по структурированному представлению музыкальной композиции.

4. Архитектура и программная реализация системы синтезирования музыкальной композиции по цветовому изображению.

5. Результаты тестирования компьютерной системы и экспертного оценивания синтезированных музыкальных композиций.

В первой главе диссертации рассмотрены современные подходы к автоматизированному синтезу звуковых последовательностей: обзор существующих систем для синтеза музыкальных композиций, обзор различных алгоритмических методов построения звуковых последовательностей, а также обзор методов синтеза звуков. Во второй главе описан разработанный комплексный метод синтезирования цельной музыкальной композиции по цветовому изображению. В третьей главе описано алгоритмическое обеспечение разрабатываемой системы. В четвёртой главе описана программная реализация системы синтезирования музыкальной композиции по цветовому изображению. В пятой главе приведено описание процесса тестирования системы, а также результаты проведения экспериментов. В заключении работы описаны основные выводы и результаты.

# 1 Современные подходы к автоматизированному синтезу звуковых последовательностей по цветовому изображению

## 1.1 Компьютерные средства генерирования музыкальных композиций по цветовому изображению

### Music in image

Единственным прямым аналогом создаваемой системы на данный момент является веб-приложение Music in image [13]. Сервис создан в 2015 году Американским программистом Arjun Gupta. На вход приложения подаётся изображение, затем программа построчно анализирует каждый пиксель (или какую-то группу пикселей), и воспроизводит соответствующие данному набору пикселей звуки.

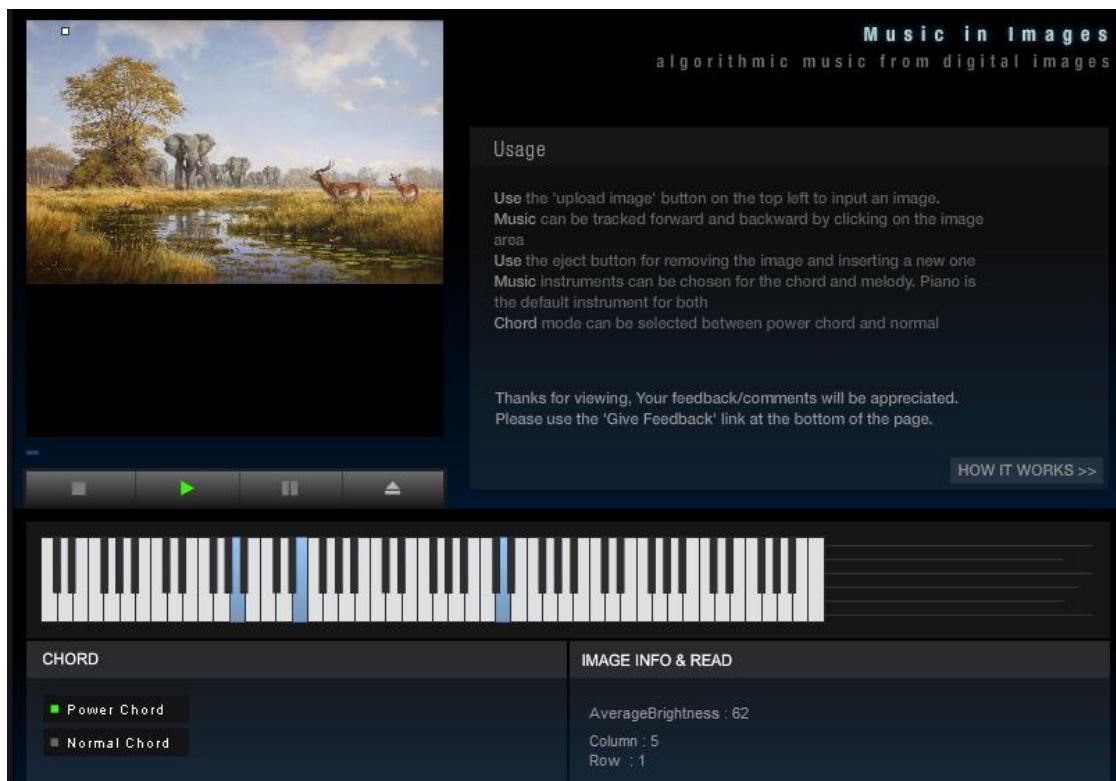


Рисунок 1 – Веб-сервис Music in image

На выходе получаем набор звуков, практически не связанных, однако диссонансов при этом не наблюдается.

Как уже было сказано программа последовательно анализирует отдельный набор пикселей, учитывая такие характеристики изображения как: цвет пикселя, яркость и насыщенность. При этом в данной программе каждый пиксель (или набор пикселей) несёт в себе информацию о гармонии и мелодии, таким образом получается, что при анализе монотонного изображения (например, изображение моря, где на протяжении всего изображения имеется один цвет – синий) даёт в гармонии одинаковый аккорд.

При анализе более наполненных изображений, получаем более интересную гармонию, однако смена аккордов полностью определяет цвет (таким образом может получиться, что половину песни будет звучать один аккорд, половину песни другой).

Таким образом, *Music in image* достаточно хорошо анализирует изображение, учитывая многие его параметры, однако полученная в итоге композиция не всегда является цельным произведением, правильно построенным с точки зрения теории музыки.

### UPIC.

UPIC – это компьютеризированный инструмент создания музыкальных композиций, разработанный композитором Iannis Xenakis [14].

Физически UPIC – это графический планшет, соединённый с компьютером, имеющий векторный дисплей. Пользователь рисует на планшете различные кривые и фигуры, которые передаются на компьютер, обрабатываются и озвучиваются. Таким образом, в зависимости от того, какие фигуры или кривые рисует пользователь, от их размера и места расположения, на выходе компьютером воспроизводятся различные звуки (ось X представляет время, ось Y - шаг) [14].

В современное время данный инструмент используется различными музыкантами и композиторами, включающими UPIC в своё произведение, а также для обучения детей младших классов азам музыки.



Рисунок 2 – UPIC

### MetaSynth

MetaSynth – это приложение для операционной системы Mac OS, позволяющее пользователю «рисовать» звуки [15]. В основе данного приложения лежит быстрое преобразование Фурье (FFT) [15]. Данное приложение является двукратным лауреатом Electronic Musician's Prestigious Editor's Choice Award. Программа имеет шесть подпрограмм, называемых комнатами (Room).

Image Synth. Это «Комната», позволяющая непосредственно рисовать звуки. Принцип очень напоминает систему UPIC, однако более современную, имеющую приятный дизайн и не требующий графического планшета.

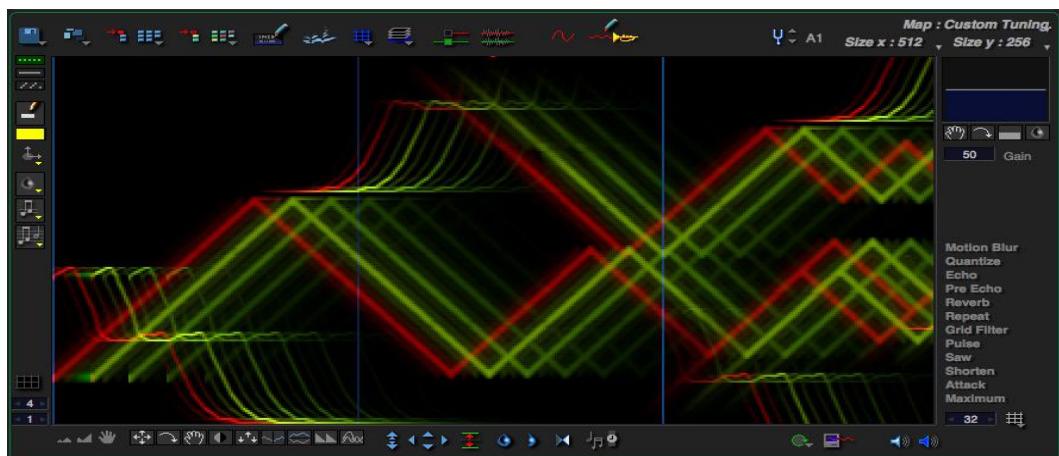


Рисунок 3 – Image Synth

Image Filter Room. Данная «Комната» позволяет накладывать различные фильтры на изображение, нарисованное в предыдущей комнате. Фильтры позволяют разнообразить звуки.

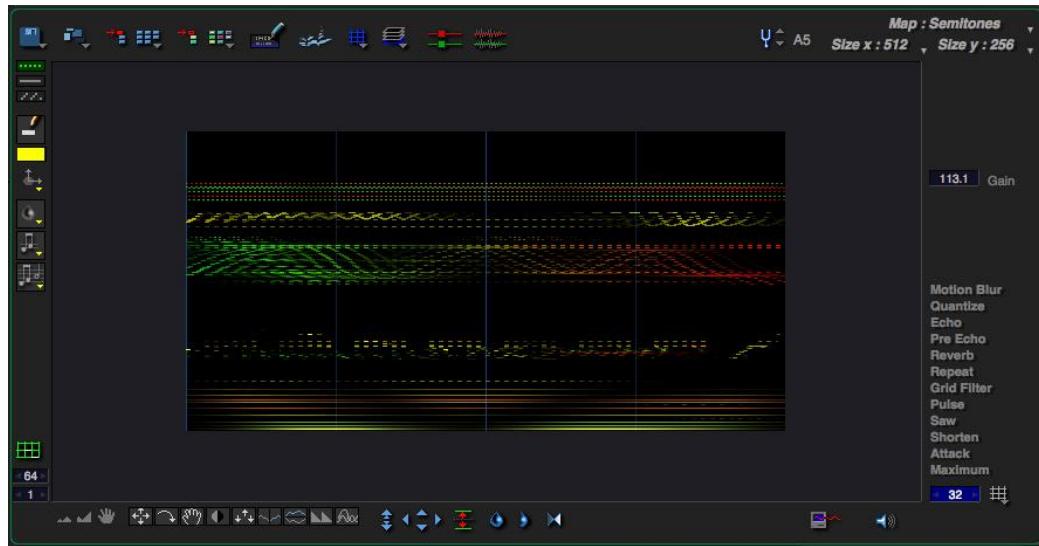


Рисунок 4 – Image Filter Room

The Spectrum Synth Room позволяет, путём анализа звуков, извлекать из них оригинальные гармоники, для наложения фильтров и создания новых звуков.

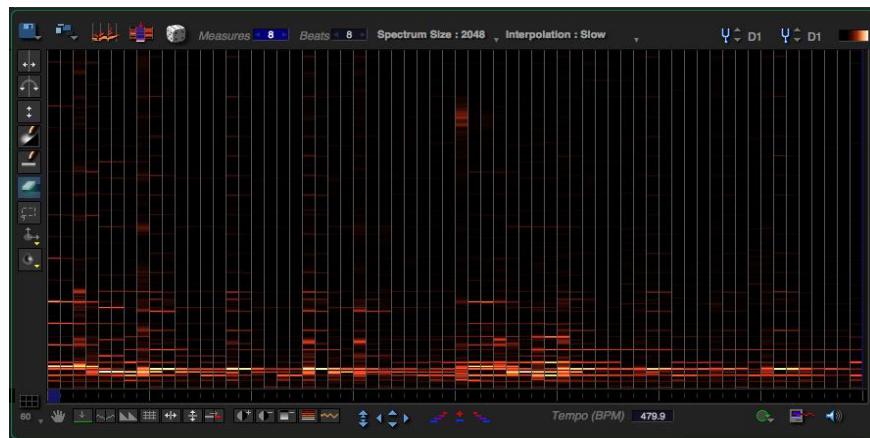


Рисунок 5 – The Spectrum Synth Room

Effects Room. Данная «Комната» позволяет накладывать различные фильтры (например, chorus, reverb, echo и другие.) на звуки, полученные в предыдущей «комнате».



Рисунок 6 – Effects Room

Sequencer Room. Это лёгкий, не MIDI секвенсор, по функционалу похожий на стандартные программы создания музыки (Fl Studio), также позволяет сохранять звуки в изображения.

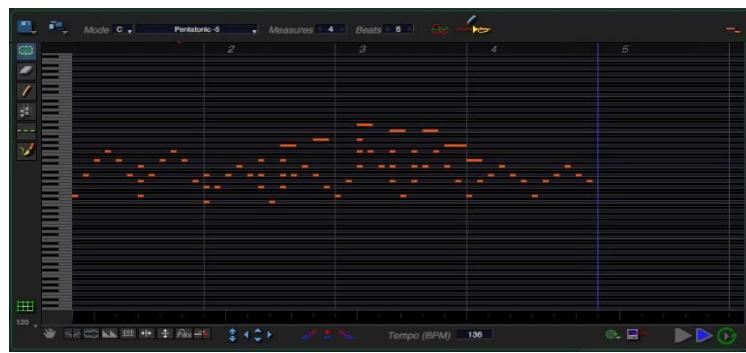


Рисунок 7 – Sequencer Room

Montage Room. Это 24-х дорожечный микшер, позволяющий смешивать звуки, для получения полноценной композиции.

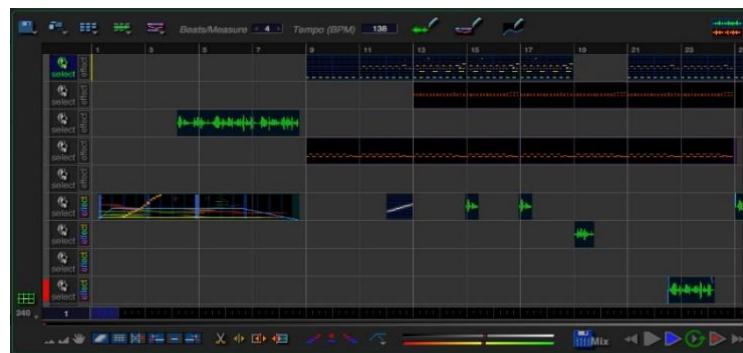


Рисунок 8 – Montage Room

### Iamascope

Iamascope – это программа, частью которой является экран, на котором отображаются абстрактные изображения (калейдоскоп) и видео камера, которая отслеживает движения пользователя. Программа отслеживает действия пользователя и генерирует калейдоскоп (абстрактное изображение), затем полученное изображение передаётся подсистеме генерации звуков, которая по данному изображению создаёт «абстрактное» музыкальное произведение, которым, благодаря действиям пользователя, можно управлять в режиме реального времени [16].



Рисунок 9 – Iamascope

### Magenta

Magenta — это музыкальный проект с открытым исходным кодом от Google. Проект стал открытым в июне 2016 года и в настоящее время реализует рекуррентную нейронную сеть. На данный момент позволяет генерировать одноголосное музыкальное произведение на основе обучения по midi файлам [3].

Программа использует midi музыкальные инструменты, что позволяет пользователю генерировать музыкальные произведения с различными музыкальными инструментами.

### BachBot

Исследовательский проект Фейнмана Ляна из Кембриджского Университета. Использующий LSTM (долгая краткосрочная память – разновидность архитектуры рекуррентных нейронных сетей). Для программы

автор использовал хоровые композиции Баха. Цель работы состоит в том, чтобы создавать и гармонизировать хоровые композиции, неотличимые от собственной работы Баха. Веб-сайт программы предлагает тест, в котором можно прослушать две композиции и угадать, какая из них является настоящей композицией Баха [3].

Исследования показали, что людям трудно отличить сгенерированную композицию от настоящих произведений Баха. Кроме того, это одна из лучших работ в обработке полифонических композиций, поскольку алгоритм может обрабатывать до четырех голосов.

### **FlowMachines**

Проект французских разработчиков. Является программой, помогающей композиторам сочинять поп-музыку, на основе шаблонов существующих композиций, записанных в базе данных. На данный момент база данных состоит из 13 000 шаблонов.

Создание композиций с использованием данной программы является трудоёмким процессом и требует специализированных знаний от пользователя-композитора, однако выходная музыкальная композиция неотличима от произведений, написанных композиторами вручную.

### **DeepJaz**

Данная система позволяет генерировать джазовые композиции на основе анализа midi файлов. Она использует двухслойную LSTM, также как Magenta и BachBot. На выходе также получается midi файл, который требует многочасовой постобработки для того, чтобы сделать композицию похожую на произведение, написанное человеком. Для генерации композиции используются такие библиотеки, как Keras, Theano и Tensorflow.

### **Сравнение систем**

Программа Music in image достаточно хорошо анализирует изображение, учитывая многие его параметры, однако полученная в итоге композиция не всегда является цельным произведением, правильно построенным с точки зрения теории музыки, в отличии от создаваемой системы.

Система UPIC является программно – аппаратной платформой создания музыкальных произведений. Пользователь, посредством графического планшета рисует кривые и фигуры, которые передаются на компьютер, анализируются и озвучиваются. Таким образом на выходе получаем просто набор звуков, полностью зависящий от нарисованных пользователем изображений, т.е. пользователями данной системы являются композиторы.

Система MetaSynth в целом похожа на систему UPIC, более современную и не использующую графический планшет. Часть функций данной системы также аналогична функциям секвенсоров, например, FL Studio.

Система Iamascopre является примером real-time управления музыкальным произведением посредством жестов. Пользователь, стоит перед камерой, которая захватывает его движения и изменяет, в зависимости от их характера, музыкальное произведение.

Программы Magenta, BachBot, FlowMachines и DeepJaz позволяют автоматизировано генерировать музыкальные последовательности, однако не позволяют генерировать звуки по изображению [17].

Аналоги сравнивались по следующим критериям:

- удобство использования программы (интерфейс) – 1/неудобно, 2/скорее неудобно, 3 /нейтрально, 4/скорее удобно, 5/удобно;
- возможность загрузки готового изображения – 1/не работает с изображениями, 2/работает с изображениями очень слабо, 3/работает с собственными абстрактными фигурами; 4/позволяет нарисовать любое изображению с использованием вспомогательных средств; 5/работает с широким спектром изображений;
- цельность композиции – 1/не цельная, звучит как отдельные воспроизведимые звуки, 2 скорее не цельная, какие-то части звучат цельно, но большая часть звуков несвязанная, 3/средняя, какие-то части композиции звучат цельно, какие-то нет, 4/скорее цельная, есть отдельные выбивающиеся ноты, но в целом звучит связано, 5/цельная, вся композиция звучит как единое целое;

- мелодичность композиции – 1/не мелодичная, присутствует большое количество диссонирующих нот, 2/скорее не мелодичная, отдельные участки звучат мелодично, но большая часть композиции представляет полную смесь нот, 3/средне мелодичная, часть композиции звучит мелодично, часть нет, 4/скорее мелодичная, в целом композиция мелодична, но встречаются диссонирующие и не мелодичные ноты, 5/полностью мелодичная;
- учитывает музыкальные правила построения композиций – 1/совсем не учитывает, 2/скорее не учитывает, 3/более-менее учитывает, 4/скорее учитывает, 5/учитывает;
- анализирует изображение как целое – 1/не анализирует изображение, 2/анализирует каждый пиксель отдельно, 3/анализирует несколько рядом стоящих пикселей, 4/анализирует большой контекст пикселей, не обязательно стоящих рядом, 5/анализирует весь контекст изображения.

Результат сравнения аналогов по предложенным критериям представлен в таблице 1.

Таблица 1 – Сравнение компьютерных средств генерирования музыкальных композиций по цветовому изображению

	1*	2*	3*	4*	5*	6*	7*	8*
Интерфейс	5	4	3	3	5	3	3	3
Возможность загрузки готового изображения	5	1	1	2	2	1	1	1
Цельность композиции	3	3	3	2	4	4	3	3
Мелодичность композиции	4	3	3	2	4	4	3	3
Учёт музыкальных правил построения композиций	3	3	2	2	5	4	3	4
Анализ изображения - как целого	3	5	5	5	1	1	1	1

где 1 – Music in image, 2 – UPIC, 3 – MetaSynth, 4 – Iamascope, 5 – Magenta, 6 – BachBot, 7 – FlowMachines, 8 – DeepJaz.

## 1.2 Алгоритмические методы построения звуковых последовательностей

Существует большое количество математических методов для генерации музыкальных композиций с использованием компьютеров. Условно все методы алгоритмической композиции можно разделить на следующие группы [18, 19]:

- вероятностные методы;
- грамматические методы;
- биологические методы;
- методы искусственного интеллекта;

### Марковские модели

Применение Марковских процессов в генерации музыкальных конструкций впервые было рассмотрено Гарри Ф. Олсоном в 1950 году. Вместе с Генри Беларом он разработал “Электронный музыкальный синтезатор” в 1955 году, первом устройстве, которое называлось синтезатором. Олсон проанализировал одиннадцать мелодий Стивена Фостера и сконструировал Марковские модели первого и второго порядка в отношении нот и ритмов. Также, марковские модели для генерации музыкального материала использовали Леджарен Хиллер и Леонард Исааксон в своём произведении «Сюита Иллиака». Янн Ксенакис использовал Марковские модели для генерации звукового материала в 1958 году в композиции “Analogique A” [18, 20].

Структурно Марковские модели построены таким образом, что предоставляют только описание контекста через вероятности переходов, притом позволяя проследить контекст только в прямой последовательности. Другими словами, данные модели больше подходят для моделирования одномерных символьных последовательностей, что согласуется с тем фактом, что они изначально были разработаны для моделирования процесса обработки естественного языка. Однако, данная особенность делает сложным применение обычных Марковских моделей для задачи генерации музыки, поскольку музыкальная информация зависит от всего предыдущего контекста, кроме того,

добавляя несколько вертикальных измерений – взаимосвязанные голоса в полифонической композиции. Одним из возможных способов решения данной проблемы является применение скрытых Марковских моделей, однако при таком подходе строится большая матрица переходов, что приводит к практически воспроизведению целых кусков композиций, а не генерации новых [19, 21].

Несмотря на свои недостатки, Марковские модели хорошо подходят для определённых музыкальных задач, например, для задачи имитации стиля. Тип и качество выходной композиции будет в значительной степени зависеть от свойств обучающего набора данных, и может быть хорошо предсказано, в отличие от других методов, например нейронных сетей или клеточных автоматов [18, 21].

### Порождающая грамматика.

Порождающая грамматика — формализм генеративной лингвистики, связанный с изучением синтаксиса. В рамках подхода порождающей грамматики формулируется система правил, при помощи которых можно определить, какая комбинация слов оформляет грамматически правильное предложение. Термин впервые введён в научный оборот американским лингвистом Ноамом Хомским в конце 1950–х годов. Цель лингвистической теории по Хомскому заключается в том, чтобы объяснить факт поразительно быстрого усвоения родного языка ребенком на основе явно недостаточного внешнего стимула, то есть той информации, которая может быть извлечена из речи окружающих [18].

Применение порождающих грамматик для задачи генерации музыкальных композиций является естественной процедурой, поскольку именно данный подход связан с формализацией, в том или ином виде, правил построения музыкальных композиций. В рамках данного подхода предполагается построение контекстно-зависимой грамматики по какому-либо обучающему набору. После построения грамматики, необходимо задать какую-то начальную мелодию и затем выводить композицию на основе построенных ранее правил. Для определения нот вполне естественно использовать их обозначение латиницей (С – нота до, D – ре и т.д.), однако, помимо этого, ещё необходимо определять высоту, например,

указывать цифровое обозначение октавы. Однако такой подход сильно ограничен, например невозможно генерировать полифонические мелодии, кроме того, построение грамматических правил не гарантирует соблюдение всех музыкальных норм построения композиций, поскольку зачастую они гораздо сложнее чем прямая зависимость от последовательности предшествующих нот. Второй проблемой грамматических методов является проблема, схожая с Марковскими моделями – построенная система правил и последующий вывод по ней композиции может привести к тому, что будут воспроизводиться целые участки композиций, по которым была построена данная грамматика. Кроме того, построение правил требует достаточно серьёзного подхода в том смысле, что в музыкальной теории существует большое количество различных жанров, под которые существуют разные правила, что может приводить к некорректному построению грамматики [22].

Таким образом, использование грамматических методов в задаче генерации музыкальных композиций кажется естественным продолжением попытки формализации музыкальных правил, однако при глубоком рассмотрении приводит к невозможности генерации сложных конструкций, а также генерации похожих композиций, которые только повторяют данные, на которых эти правила были получены. Однако, грамматические методы становятся очень полезными для задачи анализа музыкальных произведений, поскольку так или иначе позволяют построить систему правил и делать по ней выводы.

Сети Петри.

Сети Петри — это математический аппарат, который был создан для моделирования событийно-управляемых систем и процессов. Впервые описаны Карлом Петри в 1962 году. Сеть Петри представляет собой двудольный ориентированный граф, узлы которого могут состоять из позиций и переходов, соединённых между собой дугами [18]. В позициях могут быть указаны метки, которые могут перемещаться по сети при наступлении определённых событий. События — это выполнение условия перехода в сети, при этом метки

перемещаются по сети из входных позиций в выходные. При этом для первоначальной работы необходимо проинициализировать сеть начальными состояниями [23].

В алгоритмической композиции, сети Петри впервые применили Гоффредо Хаус и Альберто Саметти, разработав систему ScoreSynth. С помощью взаимосвязи позиций в графе, которые описывают некоторые музыкальные объекты с функциями перехода, система может генерировать и манипулировать данными в формате MIDI. Музыкальные объекты состоят из последовательностей нот с соответствующей информацией о названии ноты, её длительности, скорости и канале MIDI. Поскольку в традиционной сети Петри временное структурирование последовательностей не кодируется (поскольку переходы срабатывают в тот момент, когда они связаны с отмеченным местом на входной стороне), места снабжены счетчиком, который обеспечивает доступ к информации соответствующего «музыкального объекта» только после определенного периода времени [18, 19].

### Хаос и самоподобие.

Теория хаоса стала популярной в 80-х годах двадцатого века с распространением так называемого эффекта бабочки. Данная теория описывает некоторые нелинейные системы, поведение которых кажется непредсказуемым, хотя может опираться и на определённо заданную модель [19].

Фрактал – это геометрическая фигура, которая показывает высокую степень самоподобия. В контексте визуального представления, это означает, что определённые графические элементы могут встречаться в шаблоне много раз в идентичных или очень близких формах [19].

Теория хаоса и самоподобие является достаточно широкой областью, поэтому в контексте алгоритмической композиции существует большое количество различных применений данных теорий. Например, часто применяемой формой генерации музыкальных структур из хаоса является теория, известная как фрактальный шум. Сутью данного способа является некое подобие генерации

музыки по шуму, при этом используется теория, называемая «цвета шума». В данной системе терминов описываются определения, при которых некоторым видам шума приписываются определённые цвета исходя из их спектральной аналогии [24].

В 1975 году ученые кафедры физики Калифорнийского Университета Ричард Фосс и Джон Кларк проанализировали несколько записей музыкальных произведений (от Баха до пианиста Скотта Джоплина) и пришли к выводу, что мощность звуковых частот композиций очень близка к белому шуму. Стоит отметить, что ученые для общего анализа использовали не отдельные произведения, а склеенные, иногда кардинально различающиеся по стилю, произведения, перемежающиеся речевыми комментариями и вставками — продолжительность аудиоматериала могла достигать 12 часов. Ученые считали, что нужно как можно более обще подойти к количеству анализируемого материала, а не использовать отдельные несопоставимые выборки отдельных произведений. Таким образом, в долгосрочной перспективе, розовый шум оказался наиболее близок к музыке — он не имеет непредсказуемости белого шума, ровно так же, как и предсказуемости коричневого, и логарифмически коррелирует с прошлыми значениями. Иными словами, средняя активность последних десяти событий настолько же сильно влияет на текущее значение, как и последние сто или тысяча событий [25].

### Системы Линденмайера.

Системы Линденмайера или Л–системы были разработаны в 1968 году ботаником Аристидом Линденмайером для изучения развития водорослей. Позже данные системы были расширены путём введения систем графического представления, которые впоследствии использовались для моделирования процессов роста различных растений. Впервые для решения задач автоматической генерации музыки их применил в своей диссертации 1996 года американский программист и композитор Люк Дюбуа [18].

В основе работы Л–систем лежит набор правил замещения, а принцип работы схож с грамматическими методами. Начиная с начального элемента применяются продукционные правила для получения дальнейшей композиции. В отличие от традиционных грамматических методов, данные системы не различают терминальные и нетерминальные символы, таким образом продукционные правила и правила замещения применяются вместе. Правила замещения определяют, как должен быть перемещен каждый конкретный символ в текущем поколении [18].

Применение Л–систем для генерации алгоритмических композиций подразумевает использование вместо символов определенных музыкальных параметров. Например, профессор Педро Пестана использовал алфавит из семи символов — нот, входящих в гамму до мажора, а правила замещения определяются исходя из матрицы переходов [18, 26].

Другим воплощением Л–систем в алгоритмической композиции является программа LMUSE Дэвида Шарпа, в которой различные музыкальные параметры, такие, как высота, продолжительность и громкость могут быть назначены различным компонентам положения, толщины линии и т.д. В данную программу заложено около 20 команд направлений, 10 команд движений, а также около 10 команд, относящихся только к музыкальному воплощению. Например, высота ноты определяется исходя из текущего положения черепахи в данный момент, продолжительность ноты — из длины нарисованной линии, динамика — из вектора поступательного движения [18, 26].

Генетический алгоритм.

Генетический алгоритм — это алгоритм поиска решения, основанный на биологических принципах эволюции. Генетический алгоритм хорошо подходит для решения задач, о которых на начальном этапе мало известно. Он использует принцип отбора и эволюции для поиска нескольких различных вариантов решения заданной задачи [18, 27].

Входом генетического алгоритма является набор потенциальных решений проблемы, которые закодированы каким-либо образом, в зависимости от самой задачи. Также задаётся так называемая функция пригодности, которая позволяет качественно оценить кандидата на соответствие условиям задачи. Данные кандидаты могут быть уже известными решениями, например если целью алгоритма является их улучшение, либо же они могут быть заданы случайным образом. Затем, алгоритм оценивает каждого кандидата в соответствии с функцией пригодности, чем больше значение функции для кандидата, тем лучше он удовлетворяет условиям задачи, а значит является искомым решением [27].

Одним из первых применений генетического алгоритма для задачи создания музыки можно считать работу профессора кафедры компьютерных наук Гонг–Конгского университета Эндрю Хорнера и профессора кафедры индустриальной инженерии университета Иллинойса Дэвид Голдберг в 1991 году [18]. Описанная ими техника использует некий начальный паттерн, модифицирует его с помощью функции и затем сравнивает результат с использованием функции пригодности. Для генерации композиции авторы использовали шесть таких циклов для того, чтобы произвести многоголосную музыкальную композицию. В их процессе входной шаблон может быть задан в виде латинского обозначения нот, и весь процесс подбора происходит с такими данными. Функция пригодности сравнивает соответствие высоты тона выхода с эталонным значением, а также с требуемым значением [28].

### Клеточные автоматы.

Клеточные автоматы представлены в виде  $n$ -мерной равномерной сетки, где пространства ячеек представлены векторами или  $n$ -мерными матрицами, при этом предполагается, что размерность самой сетки потенциально бесконечно. Внутри пространства есть ячейки, которые могут принимать конечное число состояний. Состояние ячейки определяется как ещё собственным состоянием, так и состоянием соседних ячеек в предыдущий момент времени. При этом определяются правила перехода состояний, которым должна следовать каждая

ячейка сетки, а новое состояние получается в результате применения правил ко всем ячейкам пространства [29].

Первым исследователем, который применил клеточные автоматы для задачи генерации музыки стал Питер Бейлис. В качестве дополнения к традиционным подходам, Бейлис определил дополнительные методы для расчёта соседних ячеек, участвующих в определении состояния. Кроме того, он предложил использовать большее количество предыдущих состояний ячейки для вычисления следующего состояния, таким образом получая лучший результат. В своей более поздней работе Бейлис описал программу CA Explorer, которая, наряду с одномерными клеточными автоматами, использовала системы Линдемайера и генетический алгоритм, таким образом описывая одну из первых программ, использующую сразу несколько методов алгоритмической композиции [19].

Однако самой известной программой, использующей клеточные автоматы стала CAMUS, разработанная в 1993 году бразильским композитором Р. Мирандой. Данная программа использует два клеточных автомата для генерации музыкальной информации. В системе используются два ранее разработанных клеточных автомата: Игра «Жизнь» математика Джона Конвея и циклический клеточный автомат Дэвида Гриффита. В циклическом автомате постоянно генерируются сложные шаблоны, которые затем циклически распространяются по всей сетке. В этом автомате ячейки инициализируются случайными значениями в заданном диапазоне, при этом используются специальные правила для вычисления состояния ячейки по состоянию ближайших соседей [19].

### Искусственные нейронные сети.

Искусственные нейронные сети (ИНС) – вычислительная модель, построенная на основе биологического принципа работы нейронов. Она состоит из связанного набора искусственных нейронов – очень простых вычислительных узлов, которые агрегируют несколько числовых входов в один выход. Процесс обучения нейронной сети построен на настройке весов между нейронами посредством анализа заранее заданного обучающего набора данных.

Впоследствии эти веса используются для «предсказания» выходов по неразмеченным данным [10].

На заре применения нейронных сетей в алгоритмической композиции, в 1970-х и 1980-х годах, они использовались для анализа музыкальных композиций, создавая модели когнитивных теорий музыки. Одним из первых комплексных примеров нейронных сетей для генерации музыкальных структур стала работа Германа Хильда, Йоханнеса Фельнера и Вольфрама Менцеля, которые разработали программу HARMONET. Данная программа могла гармонизировать мелодии в стиле Баха на основе рекуррентной нейронной сети и системе, основанной на правилах. Более поздним подходом к построению систем алгоритмической композиции основанным на нейронных сетях стала программа CONCERT Майкла Мозера. Данная система способна генерировать мелодии по некоторой гармонической партии. CONCERT была обучена на хоровых партиях, фолк композициях и гармонических последовательностях различных вальсов [19].

Сейчас наиболее часто используемой архитектурой нейронной сети для генерации музыкальных композиций является рекуррентная искусственная нейронная сеть, которая используется в Magenta, BachBot и DeepJazz [3].

#### Системы, основанные на знаниях.

Под системами, основанных на знаниях, понимаются различные системы, основанные на правилах, которые используют представления знаний в виде более или менее структурированных символов. Поскольку знания о музыкальных композициях традиционно были структурированы, как наборы более или менее формализованных правил обработки музыкальных символов, то системы, основанные на знаниях и правилах, являются естественным способом реализации алгоритмической композиции. Наиболее известной ранней работой по алгоритмической композиции с использование методов, основанных на знаниях, является произведение «Сюита Иллиака», в которой использовались знания, для описания классических правил контрапункта. Из-за этого этот раздел в основном ограничивается описанием систем с сильным фундаментом в искусственном

интеллекте (как экспертных систем), обойдя в известной степени работы композиторов, которые трудно классифицировать, из-за нерегулярности их подходов [10, 18].

S. Gill представил первое применение классической AI-эвристики к алгоритмической композиции: он использовал иерархический поиск с обратным отслеживанием для обработки набора композиционных правил. M. T. Thomas разработала основанную на правилах систему для согласования хоровых композиций, состоящих из четырех частей, реализованную с использованием Lisp, с целью разъяснения музыкальных правил, которые она преподавала своим ученикам. Позже она разработала еще одну систему правил для простого создания мелодии в индийском стиле регги [18, 30].

Сейчас, методы, основанные на знаниях, широко используются для генерации музыкальных композиций, однако, как правило, лучшие результаты получаются при совместном использовании таким методов с другими методами алгоритмической композиции [18, 31].

#### Сравнение и анализ методов.

В результате выбора метода алгоритмической музыкальной композиции были рассмотрены различные методы. Выделим основные, представляющие интерес в данной работе: Марковские модели (ММ); порождающая грамматика (ПГ); сети Петри (СП); хаотические системы и фрактальная музыка (ХС); системы Линденмайера (СЛ); генетический алгоритм (ГА); клеточные автоматы (КА); искусственные нейронные сети (ИНС); системы, основанные на знаниях (СОЗ).

Они были сравнены по следующим критериям:

- возможность моделирования многомерных последовательностей – 1 (поддерживает только одномерные последовательности), 2 (поддерживает многомерные последовательности только с совместным использованием дополнительных методов), 3 (поддерживает многомерные последовательности только с существенной доработкой), 4 (скорее поддерживает многомерные

последовательности), 5 (полностью поддерживает многомерные последовательности);

- простота обучения на готовых композициях – 1 (обучение или построение правил производится вручную), 2 (поддерживает обучение на готовых композициях с совместным использованием других методов), 3 (поддерживает обучение на готовых композициях с существенным преобразованием данных), 4 (поддерживает обучение на готовых композициях с преобразованием данных), 5 (поддерживает обучение на готовых композициях);

- возможность изменения условий – 1 (можно менять только начальные условия), 5 (можно менять условия в процессе);

- простота реализации – 1 (реализовать крайне сложно или невозможно), 2 (сложно реализовать), 3 (можно реализовать с достаточным количеством усилий), 4 (относительно просто реализовать), 5 (очень просто реализовать).

Результаты сравнения методов представлены в таблице 2.

Таблица 2 – Сравнение алгоритмических методов построения звуковых последовательностей

	ММ	ПГ	СП	ХС	СЛ	ГА	КА	ИНС	СОЗ
Моделирование многомерных последовательностей	1	1	1	5	1	5	1	5	5
Простота обучения на композициях	3	2	2	3	2	3	2	4	3
Возможность изменения условий	5	5	5	1	5	5	5	5	5
Простота реализации	4	4	4	2	4	3	3	4	2

### 1.3 Методы синтеза звуков

Описание методов синтеза звуков.

Существует большое количество различных методов синтеза звуков, первые из них разрабатывались ещё в прошлом веке для включения в физические

синтезаторы, например фирмами Casio, Yamaha и другими. Другие методы более современные и имеют более современное звучание инструментов. Далее представлены основные методы и описан выбор одного из них для реализации системы.

#### Additive synthesis (Аддитивный синтез).

Метод основан на теоретических работах математика и физика Фурье. Он предполагает получение любого звука путём сложения необходимого количества синусоидальных колебаний с определенной высотой и громкостью [33].

#### FM-synthesis (FM-синтез).

Базовый смысл FM-синтеза – это создание сложных сигналов путём модуляции простых. Если взять простой звук и удалить из него все обертоны, то получится только основная частота, представленная синусоидой. Точно также, если выделить из простого звука только один частный обертон, то получится синусоида. Таким образом, все формы звука состоят из набора простых синусоид. Именно данную логику использует данный метод – создание сложных звуков из простых путём их модуляции: простой сигнал используется для модуляции другого простого сигнала, тот может модулировать другой и так далее [34].

#### Sample playback (Сэмплинг).

Сэмплинг – единственный из видов синтеза, в котором звук не генерируется путём низкоуровневой работы с сигналами, а воспроизводится, т.е. образцы звуков хранятся в памяти и воспроизводятся с нужной высотой и громкостью, либо вообще хранятся отдельные звуки для разных октав. Один из наиболее реалистичных способов воспроизведения реальных звуков, достаточно прост и дешев, но очень сильно ограничен в возможностях изменения готовых пресетов [35].

#### Wavetable synthesis (таблично-волновой синтез).

Таблично-волновой синтез использует схожий принцип воспроизведения звука, что и метод сэмплинга. Однако, данный метод основан на последовательном или параллельном воспроизведении различных волновых

форм, которые были записаны заранее. В данном методе используются специальные таблицы, ячейки которой содержат определённые волновые формы [36].

#### Subtractive synthesis (субтрактивный синтез).

Субтрактивный синтез основан на фильтрации формы, которая содержит большое количество гармоник в более простую и чистую форму. Богатые формы гармонических волн представляют собой достаточно простые формы, например прямоугольные волны. Такие волны достаточно просто воспроизвести, однако они звучат слишком грязно и не музыкально. В данном методе присутствуют три важных компонента – генератор, фильтр и усилитель. Генератор используется для изначального создания богатого звука, при этом он может быть разной формы: синусоидальный, прямоугольный, пилообразный или треугольный. Затем, используется фильтр, с помощью которого происходит «вычитание» волн. Существуют различные частотные фильтры, которые пропускают только определённые частоты, например самим известными является фильтр нижних частот. Последним этапом является усилитель, который просто усиливает сигнал для его корректного воспроизведения во внешних источниках [37].

#### Vector synthesis (векторный синтез)

Векторный синтез (vector synthesis) — тип синтеза, построенный на динамическом морфинге (перетекании) аудиосигналов для получения более богатых и сложных тембров. Впервые векторный синтез был представлен компанией Дэйва Смита (Dave Smith) Sequential Circuits Instruments (SCI) в инструменте Prophet VS в 1986 году [38].

#### Линейно-арифметический синтез

Линейно-арифметический синтез (Linear Arithmetic Synthesis, LA) — тип синтеза, основанный на формировании звука путем совмещения сэмплированной атаки с генерируемой остальной частью сигнала. Безусловно, как и некоторые другие типы, линейно-арифметический синтез является гибридным, поскольку совмещает в себе два метода синтеза для получения сложного тембра [39].

## Сравнение и анализ методов синтеза звуков

Аддитивный синтез (AC) очень сложен для реализации, из-за необходимости отдельного контроля громкости и высоты каждой гармоники, которых даже несложный тембр насчитывает десятки [1].

FM – синтез (ФМС) хорошо применим для синтеза звука ударных инструментов, синтез же остальных музыкальных инструментов звучит слишком искусственно. Главный недостаток FM-синтеза — неспособность при его помощи полноценно имитировать акустические инструменты [1].

Сэмплинг – использование образцов (ИО) применяется в большинстве современных синтезаторов, так как даёт наиболее реалистичный звук и достаточно прост в реализации [1].

Таблично-волновой синтез (ТВС) и линейно-арифметический синтез (ЛАС) похожи на сэмплерный метод, однако данные методы сложны в реализации, поэтому на практике предпочтение отдаётся семплингу, как наиболее простому.

Субтрактивный синтез (СС) обычно используется совместно с аддитивным, обладает хороших качеством синтеза звуков, однако сложен в реализации.

Векторный синтез (ВС) используется для получения более богатых и сложных тембров, однако в рамках рассматриваемой системы это не существенно [40].

Для детального сравнения выделим следующие критерии:

- простота реализации – 1 (реализовать крайне тяжело или невозможно), 2 (сложно реализовать), 3 (можно реализовать с достаточным количеством усилий), 4 (относительно просто реализовать), 5 (очень просто реализовать);

- ограниченность – 1 (очень ограничен в возможностях синтеза и модификации звуков), 2 (скорее ограничен), 3 (средне ограничен); 4 (скорее гибок и позволяет в реальном времени менять параметры и звуки); 5 (гибок позволяет в реальном времени менять параметры и звуки);

- реалистичность звучания – 1 (инструменты звучат искусственно), 2 (инструменты звучат скорее искусственно), 3 (инструменты звучат более-менее

реалистично), 4 (инструменты звучат скорее реалистично), 5 (инструменты звучат реалистично).

Сравнение методов синтеза звуков представлена в таблице 3.

Таблица 3 – Сравнение методов синтеза звуков

	AC	ФМС	ИО	ТВС	СС	ВС	ЛАС
Простота реализации	2	3	5	3	2	3	3
Ограниченност	4	4	3	4	4	4	4
Реалистичность звучания	4	3	5	5	4	4	5

#### 1.4 Выбор нейронной сети для генерации музыкальных композиций

##### Классификация искусственных нейронных сетей

Существуют различные классификации нейронных сетей в зависимости от классификатора. По количеству слоёв выделяют однослойные и многослойные; по типу связи – прямого распространения и с обратными связями; по способу обучения – с учителем и без учителя [41].

В данной работе нам наиболее интересна классификация по типу связи, поскольку именно она позволяет определить необходимые взаимосвязи в музыкальных данных.

Сети прямого распространения содержат входные нейроны, которые получают какие-либо входные сигналы, затем преобразуют их и, наконец, передают нейронам первого скрытого слоя, если скрытые слои вообще есть в архитектуре сети, затем выход первого слоя передаётся на второй скрытый и так далее по очереди до N-го слоя, который уже определяет выходные сигналы. Обычно в такой сети каждый выход i-ого слоя подаётся на вход всех нейронов i+1-го слоя, однако может быть и произвольная архитектура [42].

Обычно при работе с сетями прямого распространения используются так называемые многослойные сети – сети, в которых существует один и более скрытых слоёв. Именно такие сети обеспечивают наибольшую точность и скорость прогнозирования для различных задач [43].

Сети с обратными связями (часто называются рекуррентными сетями) – это сети, у которых нейроны имеют рекуррентные связи с текущего слоя на предыдущий, таким образом в расчёте очередного состояния нейрона участвуют не только выходы предыдущего слоя, но и предыдущие состояния [44].

Для обучения нейронных сетей существуют различные алгоритмы, однако самым распространённым является метод обратного распространения ошибки. Данный метод изменяет веса в зависимости от найденной ошибки, которая представляет собой разницу между ожидаемым и фактическим выходом нейронной сети. Изменение весов происходит в обратном направлении – от выходного слоя, где как раз можно определить наличие отклонения в ожидаемом и фактическом значении, затем продолжается к скрытым слоям и наконец до входного слоя [45].

#### Вывод о выборе нейронной сети

Важной особенностью нейронных сетей прямого распространения является то, что у данной нейросети есть общее ограничение: и входные и выходные данные имеют фиксированный, заранее обозначенный размер, например, картинка  $100 \times 100$  пикселей или последовательность из 256 бит. Нейросеть с математической точки зрения ведет себя как обычная функция, хоть и очень сложно устроенная: у нее есть заранее обозначенное число аргументов, а также обозначенный формат, в котором она выдает ответ [11, 46].

Вышеперечисленные особенности не представляют больших трудностей, если речь идет о тех же картинках или заранее определенных последовательностях символов. Но для обработки любой условно бесконечной последовательности, в которой важно не только содержание, но и порядок, в котором следует информация, например, текст или музыка, необходимо

использовать нейронные сети с обратными связями – рекуррентные нейронные сети (RNN). В рекуррентных нейросетях нейроны обмениваются информацией между собой: например, вдобавок к новому кусочку входящих данных нейрон также получает некоторую информацию о предыдущем состоянии сети. Таким образом в сети реализуется некоторое подобие памяти, что принципиально меняет характер ее работы и позволяет анализировать любые последовательности данных, в которых важно, в каком порядке идут значения, а также взаимосвязи с предыдущими состояниями [11, 47].

Однако большой сложностью сетей RNN является проблема исчезающего (или взрывного) градиента, которая заключается в быстрой потере информации с течением времени. Конечно, это влияет лишь на веса, а не на состояния нейронов, но ведь именно в них накапливается информация. Сети с долгой краткосрочной памятью (long short term memory, LSTM) стараются решить вышеупомянутую проблему потери информации, используя фильтры и явно заданную ячейку памяти. У каждого нейрона есть клетка памяти и три фильтра: входной, выходной и забывающий. Целью этих фильтров является защита информации. Входной фильтр определяет, сколько информации из предыдущего слоя будет храниться в ячейке. Выходной фильтр определяет, сколько информации получат следующие слои. Такие сети способны научиться создавать сложные структуры, например, сочинять тексты в стиле определённого автора или сочинять простую музыку, однако при этом потребляют большое количество ресурсов [11].

Таким образом, для реализации системы автоматизированной генерации музыкальных композиций необходимо использовать именно рекуррентные нейронные сети с долгой краткосрочной памятью – RNN LSTM (долгая краткосрочная память – разновидность архитектуры рекуррентных нейронных сетей). Именно данный вид нейронных сетей используется для генерации музыкальных композиций в программе Magenta – это музыкальный проект с открытым исходным кодом от Google, также RNN LSTM используется в программе сочинения композиций в стиле И.С. Баха – BachBot, а также в DeepJaz

– система позволяет генерировать джазовые композиции на основе анализа midi файлов [11, 48].

## 1.5 Выводы по первой главе

В результате исследования и сравнения аналогов были рассмотрены следующие программные системы: Music in image, UPIC, MetaSynth, Iamascopre, Magenta, BachBot, FlowMachines, DeepJaz. Среди данных систем, единственная система (Music in image) является прямым аналогом создаваемой программы, остальные – косвенными.

Таким образом, из рассмотренных, наибольшую конкуренцию создаваемой системе может составить Music in image, однако она не в достаточной мере учитывает теоретические правила создания музыкальных произведений, также изображение анализируется не как целое, что приводит к потере мелодичности и цельности композиции.

В процессе исследования методов синтеза звука, были рассмотрены семь методов. Каждый из них обладает своими плюсами и минусами, однако для реализации системы был выбран Сэмплинг. Данный метод даёт наиболее реалистичное звучание инструментов, что является важной характеристикой для системы, также данный метод относительно прост в реализации. Недостатком Сэмплинга является ограниченность метода, однако в рамках реализации системы – это не существенно, так как пользователь будет выбирать из ограниченного набора заранее известных инструментов, таким образом не требуется больших возможностей изменения готовых пресетов [1, 40]. Таким образом, лучшим методом синтеза звука для реализации системы является сэмплинг.

В результате анализа и сравнения методов алгоритмической музыкальной композиции были рассмотрены следующие методы: марковские модели; порождающая грамматика; сети Петри; хаотические системы; системы Линденмайера; генетический алгоритм; клеточные автоматы; искусственные

нейронные сети. Было выявлено, что для генерации музыкального материала в данной работе будет использоваться метод, основанный на искусственных нейронных сетях. Именно нейронные сети позволяют выявить наиболее сложные взаимосвязи, характерные музыкальным произведениям. Также большинство современных программ для автоматизированной генерации музыкальных композиций используют именно нейронные сети, например система Magenta от Google.

В результате выбора архитектуры нейронной сети для генерации композиций было решено, что в данной работе будут использоваться рекуррентные нейронные сети с долгой краткосрочной памятью – RNN LSTM.

## 2 Комплексный метод синтезирования цельной музыкальной композиции по цветовому изображению с использованием нейронных сетей

Комплексный метод синтезирования цельной музыкальной композиции по цветовому изображению с использованием нейронных сетей состоит из трёх основных этапов: анализ изображения, построение структурированного представления музыкальной композиции и синтез звуков. Шаги метода представлены на рисунке 10 [49].

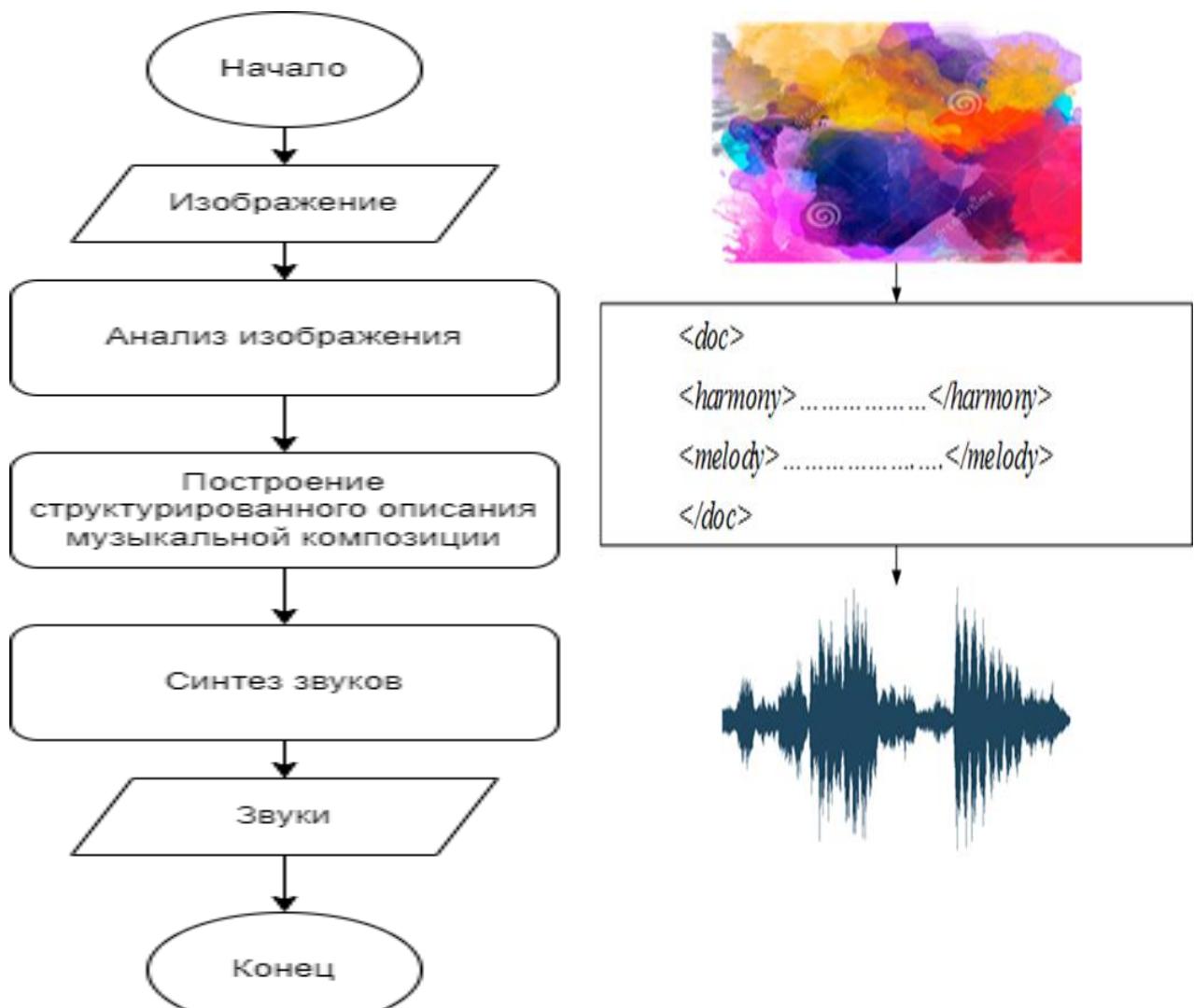


Рисунок 10 – Комплексный метод синтезирования цельной музыкальной композиции по цветовому изображению с использованием нейронных сетей

Первым этапом комплексного метода является анализ изображения и определение характеристик произведения с изображения. На данном этапе анализируется каждый пиксель изображения и получается преимущественный цвет. Кроме того, на данном этапе происходит определение первых 20% отличающихся цветов с изображения, для дальнейшего построения мелодии.

## **2.1 Метод структурированного представления музыкальной композиции**

Вторым этапом комплексного метода синтезирования цельной музыкальной композиции по цветовому изображению с использованием нейронных сетей является анализ изображения и определение характеристик произведения с изображения, для этого было определено соотношение цветовых и музыкальных характеристик (рисунок 11 и таблица 4), а также разработан метод соотнесения цветовых и музыкальных характеристик (рисунок 15) [50].

Определим соотношение цветовых и музыкальных характеристик [51, 52]:

Таблица 4 – Цветовые и музыкальные характеристики

Цветовые характеристики	Музыкальные характеристики
Оттенок (красный, синий, жёлтый...)	Нота (до, до-диез, ре, ре-диез, ми, ми, фа, фа-диез, соль, соль-диез, ля, ля-диез, си)
Цветовая группа (тёплый/холодный)	Музыкальный лад (мажор/минор)
Яркость	Октаава ноты
Насыщенность	Длительность ноты

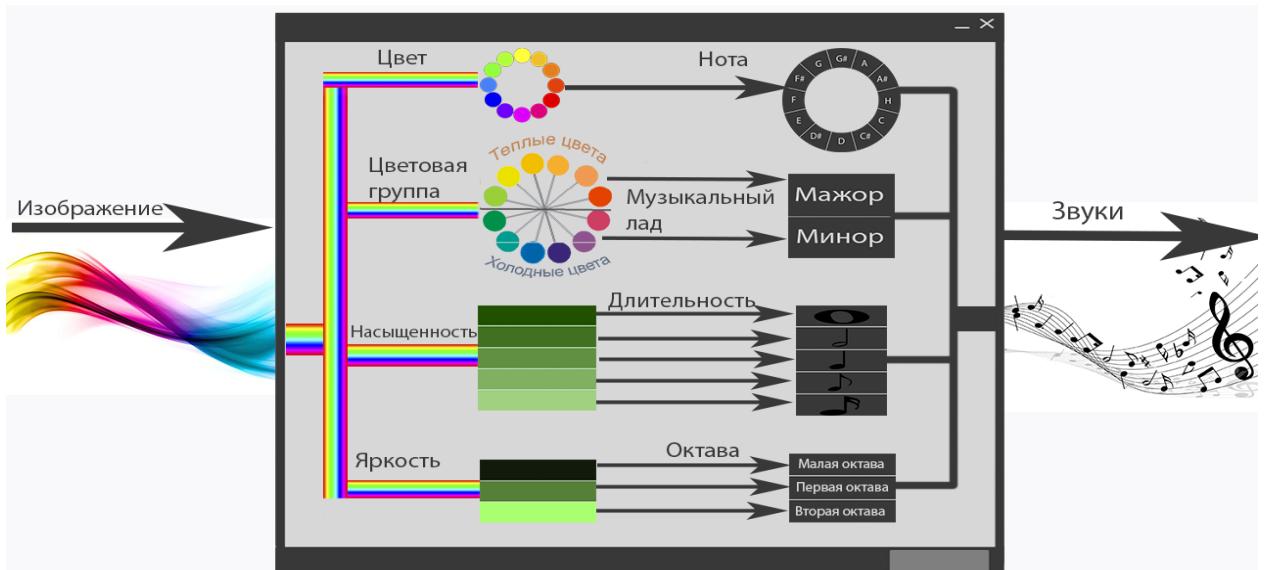


Рисунок 11 – Модель соотношения цветовых и музыкальных характеристик

Опишем модель соотношения цветовых и музыкальных характеристик в виде кортежа:

$$\text{ITM} = \langle I, \text{CN}, \text{CGMD}, \text{SATDU}, \text{BROCT} \rangle, \quad (1)$$

где I – входные данные в виде изображения; CN – соотношение цвета с наименованием ноты; CGMD – соотношение цветовой группы цвета с музыкальным ладом; SATDU – соотношение насыщенности цвета с длительностью ноты; BROCT – соотношение яркости цвета с высотой (октавой) ноты.

Согласно цветомузыкальной теории Афанасьева В.В. невозможно навсегда привязать какую-либо ноту к определенному цвету в силу их различной природы. Поэтому необходимо связать отношения звуков и цветов в зависимости от того, в какой плоскости они представлены: мелодия, гармония или тональность. Иначе говоря, одна и та же нота в произведении может быть окрашена разными цветами [3, 53].

Таким образом, метод структурированного представления музыкальной композиции состоит из следующих шагов:

- 1) Анализируя в целом изображение, определяем преимущественный цвет и последовательность цветов с характеристиками. Далее согласно преимущественному цвету и одной из пяти схем соотнесения цветов и нот,

определяется тональность будущего произведения [3]. Схемы соотнесения цветов и нот представлены в приложении А.

Максимальное родство цветов определяется между двумя соседними хроматическими цветами, а степень тонального родства – между звуками, находящимися на расстоянии семи полутонов друг от друга и отражена в квинтовом круге тональностей [3].

Таким образом, совмешая хроматический цветовой круг и квинтовый круг тональностей, получим необходимую последовательность для модуляций [3].

Например, пусть в качестве преимущественного цвета был выбран красный. Согласно схеме Ньютона, это соответствует ноте До.

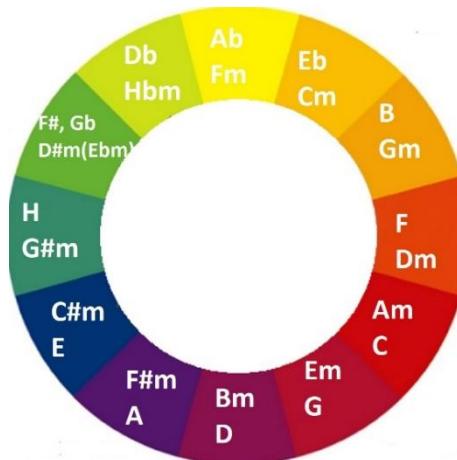


Рисунок 12 – Наложение цветового круга и квинтового круга тональностей

2) Затем, сопоставляя хроматическую гамму тоники с цветовым кругом, получим соответствие нот и цветов для мелодической части будущего произведения [3, 54].

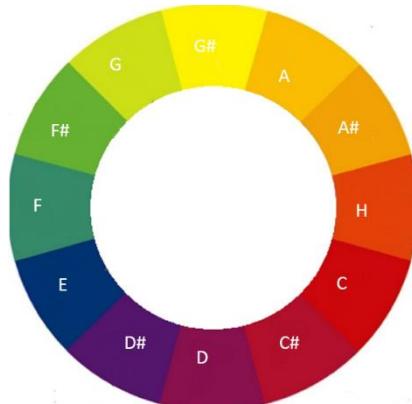


Рисунок 13 – Наложение хроматической гаммы и цветового круга

3) После этого, необходимо определить соответствие цветов и ступеней для построения гармоний произведения [3].

В музыке существуют функциональные отношения (тоника – субдоминанта – доминанта), а в цветоведении этому соответствует принцип дополнительности цветов (красный – синий – желтый). То есть, главными цветами при принятом основном, например, красном, являются дополнительные – желтый и синий, главными трезвучиями в музыке – тоническое, субдоминантовое и доминантовое [3, 54].

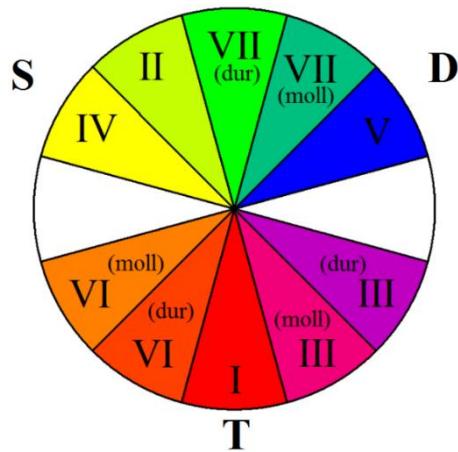


Рисунок 14 – Гармоническое соответствие

На предыдущих пунктах получили тональность будущего произведения, определили необходимую последовательность для модуляций, выявили соответствие цветов и нот применительно для мелодического сопровождения, а также получили гармонию [3].

4) Затем, согласно определённым ранее соответствиям цветов и нот для мелодического сопровождения, необходимо преобразовать полученный набор цветов в набор нот [3].

Метод структурированного представления музыкальной композиции можно представить в виде схемы, описанной на рисунке 15.

Таким образом, в данной работе предполагается следующий подход получения композиции по изображению [3, 55]:

- 1) согласно методу соотнесения цветовых и музыкальных характеристик получаем тональность произведения и последовательность первых 20% нот, считанных с изображения;
- 2) далее по полученной последовательности нот предсказываем продолжение произведения с помощью обученной модели и нейронной сети;
- 3) по итоговой последовательности нот и тональности, согласно методу соотнесения цветовых и музыкальных характеристик, строим гармоническую часть произведения.



Рисунок 15 - Метод структурированного представления  
музыкальной композиции

## 2.2 Моделирование музыкальных композиций и жанров

Описание используемой искусственной нейронной сети

Рекуррентная нейронная сеть (RNN) имеет циклические соединения, которые позволяют сети хранить информацию по входам. Эти связи можно считать похожими на память. RNN особенно полезны для изучения последовательных данных, таких как музыка [11].

В рекуррентных сетях циклические соединения можно развернуть в эквивалентную нейронную сеть прямого распространения с N слоями (рисунок 16), затем эту сеть обучают с использованием метода градиентного спуска [11].

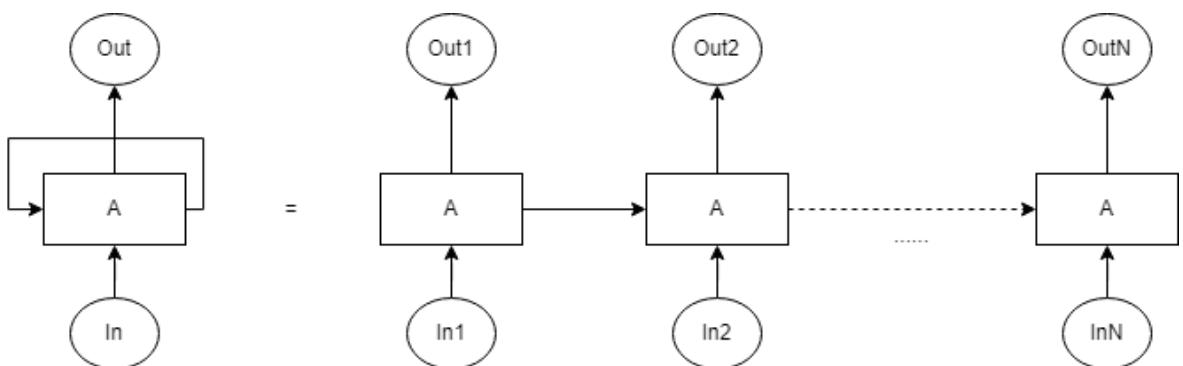


Рисунок 16 – Циклическое соединение RNN

Существуют различные способы, с помощью которых в рекуррентной сети нейроны могут иметь циклические соединения. Наиболее распространенным и эффективным способом являются так называемые сети с долгой краткосрочной памятью (long short term memory, LSTM) и управляемые рекуррентные нейроны (gated recurrent units, GRU). В обоих случаях сети имеют мультипликативные нейроны, которые защищают их внутреннюю память от перезаписи, позволяя нейронным сетям обрабатывать более длинные последовательности. В данной работе предполагается использование LSTM [11, 56].

Все рекуррентные нейронные сети имеют форму цепочки повторяющихся модулей нейронной сети. В стандартных RNN этот повторяющийся модуль имеет очень простую структуру, например, один слой  $\tanh$  (рисунок 17) [11, 57].

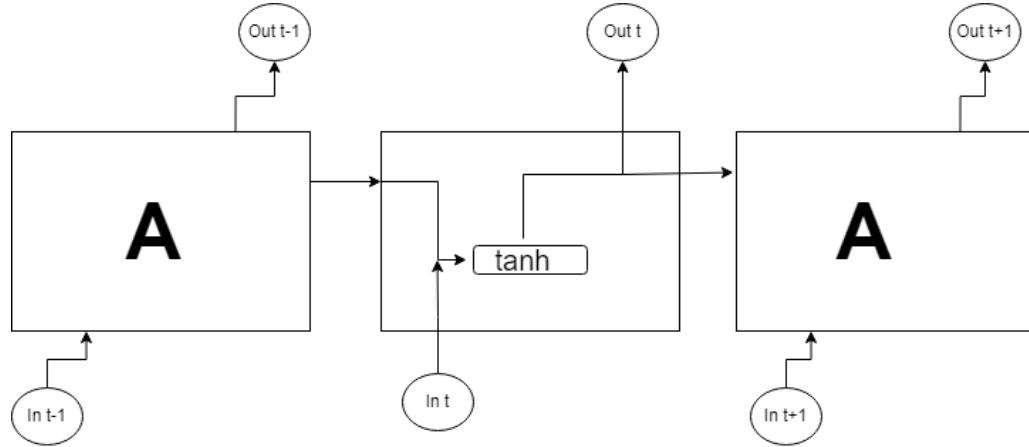


Рисунок 17 – Повторяющийся модуль в стандартной рекуррентной нейронной сети

Сети LSTM также имеют эту цепочку, но повторяющийся модуль имеет более сложную структуру. Вместо одного слоя нейронной сети, существуют четыре, взаимодействующих между собой особым образом (рисунок 18) [3, 57].

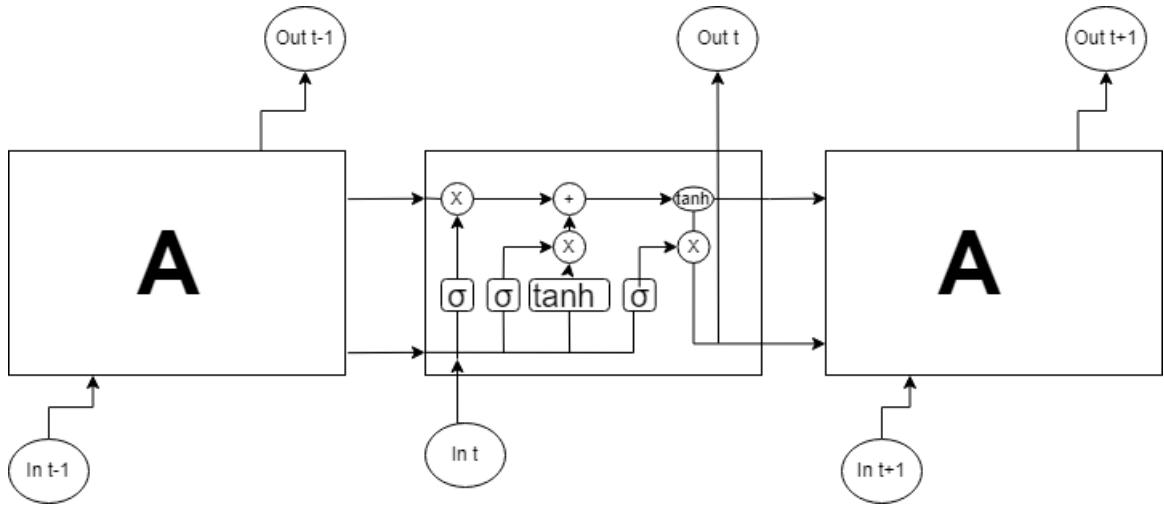


Рисунок 18 – Повторяющийся модуль в стандартной однослойной рекуррентной нейронной сети

На приведенном выше рисунке каждая строка несет целый вектор, начиная с выхода одного узла и входа другого. Круги представляют собой поточечные операции, такие как сложение векторов, в то время как в ячейках представлены

слои нейронной сети. Соединение линий означает объединение, в то время как разделение линии обозначает, что ее содержимое копируется в различные входы [3].

Первым шагом в LSTM является решение, какую информацию мы собираемся выбросить из состояния ячейки, иными словами, забыть. Это решение принимается сигмоидным слоем. Данный слой смотрит на значение выхода и входа, рассчитывает значение в диапазон от 0 до 1 для каждого состояния ячейки. Если слой вернул значений 1, это означает, что данное значение необходимо оставить (запомнить), если 0 – удалить из состояния ячейки (рисунок 19). Например, в состоянии ячейки могут храниться характеристики текущего такта – если такт ещё не закончен, то необходимо оставить характеристики в памяти, если идёт работа уже с новым тактом, то необходимо запомнить новые параметры [3].

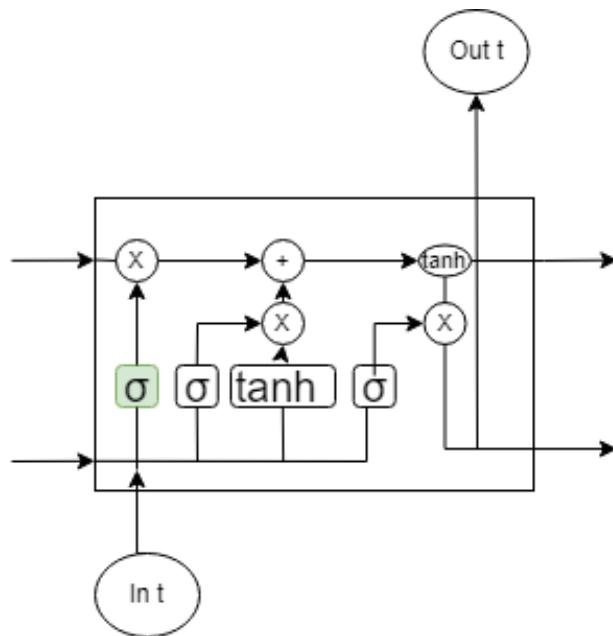


Рисунок 19 – Использование сигмоидного слоя для удаления информации из состояния

Следующим шагом является принятие решения о том, какую новую информацию мы собираемся хранить в состоянии ячейки. Для этого, во-первых, сигмоидный слой принимает решение, какие значения мы будем обновлять.

Далее, слой  $\tanh$  создает вектор новых значений кандидата, которые могут быть добавлены в состояние (рисунок 20) [3].

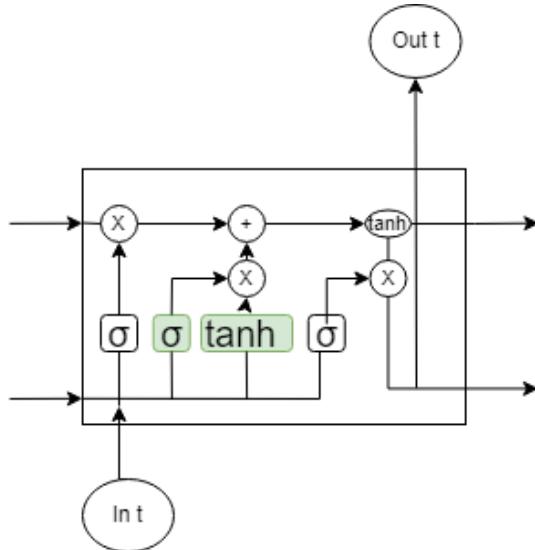


Рисунок 20 – Принятие решения о хранении новой информации в состоянии ячейки

Следующим шагом является обновление старого состояния ячейки. Для этого необходимо снова использовать  $\tanh$  функцию, таким образом произведём удаление информации из состояния и затем получим новые значения кандидатов (рисунок 21) [11].

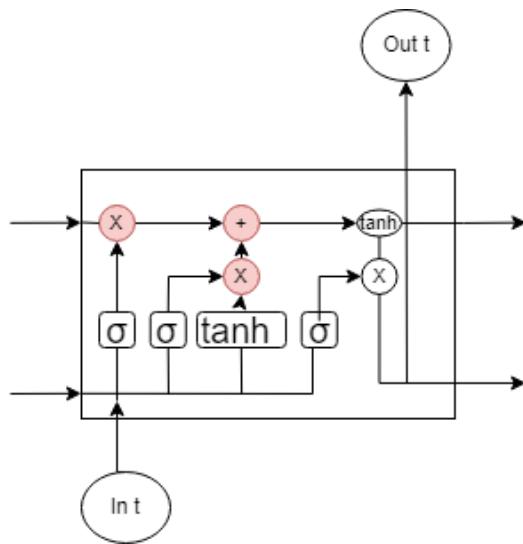


Рисунок 21 – Обновление старого и нового состояния ячейки

На последнем шаге необходимо решить, что будет выводить данный слой. Этот вывод будет основан на состоянии ячейки. Сначала пропускаем входное

значение через сигмоидный слой, который решает, какие части состояния ячейки необходимо вывести. Затем обрабатываем состояние ячейки с использованием  $\tanh$  и умножаем его на выход сигмоидного слоя [3].

Описание способа предсказания композиции.

Поведение нейронной сети определяется набором весов и смещений, которые имеют каждый узел. Поэтому, для корректной работы нейронной сети необходимо настроить их на некоторое правильное значение [3].

Во-первых, необходимо определить, насколько правильны или нет какой-либо выход согласно входному значению. Это значение называется стоимостью. Как только стоимость получена, необходимо использовать метод обратного распространения ошибки. По сути, он сводится к вычислению градиента стоимости по отношению к весам (т. е. производной стоимости по каждому весу для каждого узла в каждом слое), а затем необходимо использовать метод оптимизации для корректировки весов для снижения стоимости. В данной работе будем использовать метод градиентного спуска [3].

Для обучения нейронной сети предполагается подавать на вход вектор, который содержит следующие части [3, 58]:

- название ноты: MIDI обозначение текущей ноты. Используется для представления высоты ноты;
- время начала включения ноты;
- время выключения ноты;
- сила (громкость) воспроизведения ноты (velocity).

Для определения правильного выхода согласно входу, предполагается преобразовать вектор следующим образом: пусть имеется вектор нот  $\{c, d, e, f, g, a, h\}$ , тогда обучающий вектор будет –  $\{\{c, d\}, \{d, e\}, \{e, f\}, \{f, g\}, \{g, a\}, \{a, h\}\}$ . Такой способ обучения нейронной сети используется, например, для прогнозирования временных рядов [3, 59].

Схематично, данный подход представлен на рисунке 22.



Рисунок 22 – Построение мелодической последовательности с помощью обученной модели и нейронной сети

### Сбор и классификация обучающих данных

Для прогнозирования продолжения музыкального произведения с помощью нейронной сети, основной задачей является корректное обучение нейронной сети для композиций с различными музыкальными жанрами. И наиболее важным этапом этого процесса является сбор и классификация данных для обучения нейронной сети [60].

В данной работе мы будем обучать нейронную сеть midi-файлами – именно данный тип файлов содержит описание последовательности проигрываемых нот с их длительностями, силой звука, инструментами и прочее. Таким образом, именно анализируя midi-файлы мы можем наиболее точно получить описание музыкальной композиции, которое приближенно к человеческому описанию (музыкальная нотация).

Для данной цели, был взят набор данных «The Lakh MIDI dataset» [61]. Данный набор данных содержит коллекцию из 176581 уникальных midi-файлов, 45129 из которых соотнесены с метаинформацией из Million Song Dataset [62]. Данный набор данных включает файлы, которые содержат в наименовании

название автора и имя композиции в совершенно произвольной форме без указания на жанр, например F.FENDER.Wasted days n wasted nights.mid, i\_am\_the\_walrus.mid, или IBelieveICanFly.mid. Таким образом, следующим этапом подготовки обучающих данных является классификация данных midi-файлов по жанрам.

Для начала, был использован Rest API Spotify, который обеспечивает возможность для поиска композиции и получения жанра по автору. Для того, чтобы это сделать, необходимо получить наименование midi-файла и отправить запрос в Spotify с этим наименованием, например /v1/search?q=N COLE Almost like being in love&type=track&limit=2. Данный запрос вернёт результат поиска композиции по заданной поисковой строке в формате JSON. Данный ответ содержит некоторые метаданные композиции, например наименование, изображения и блок с артистом (исполнителем), пример такого блока ответа представлен на рисунке 23.

```

"artists": [
  {
    "external_urls": {
      "spotify": "https://open.spotify.com/artist/7v4imS0moSyGdXyLgVTIV7"
    },
    "href": "https://api.spotify.com/v1/artists/7v4imS0moSyGdXyLgVTIV7",
    "id": "7v4imS0moSyGdXyLgVTIV7",
    "name": "Nat King Cole",
    "type": "artist",
    "uri": "spotify:artist:7v4imS0moSyGdXyLgVTIV7"
  }
]

```

Рисунок 23 – Ответ на запрос поиска композиции с использованием Spotify API

Используя данный ответ, можно получить наименование исполнителя используя прямую ссылку из блока href. Данный ответ содержит некоторые метаданные исполнителя, например наименование и блок жанров, пример такого ответа представлен на рисунке 24.

```

    "genres": [
        "adult standards",
        "lounge",
        "soul",
        "vocal jazz"
    ]
  
```

Рисунок 24 – Ответ на запрос о получении информации по жанрам с использованием Spotify API

Таким образом, путём отправки двух запросов мы можем получить список жанров, к которым относится композиция. Мы обработали все композиции из набора, часть композиций не нашлась в сервисе, для некоторых не было жанров – после фильтрации всех неактуальных файлов мы получили около 50 тысяч уникальных композиций, которым соответствует 1213 уникальных жанров. Мы сохранили все результаты в базе данных с довольно простой структурой, показанной на рисунке 25.

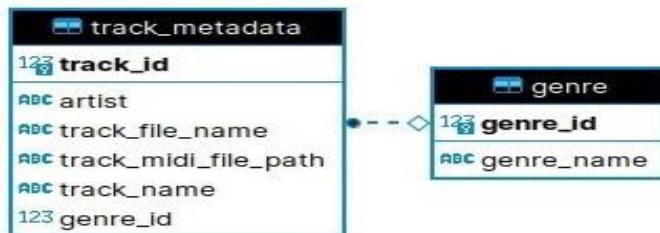


Рисунок 25 – Схема базы метаданных композиций

На предыдущем этапе мы получили около 1 тысячи уникальных жанров, большинство из которых слишком узкие, очевидно, что для предсказания композиций это слишком много. Теперь необходимо расширить все жанры до нескольких простых. Подходящего API для этого не нашлось, поэтому данное распределение было сделано вручную. Мы получили около 20 простых жанров, распределение композиций по простым жанрам показано в таблице 5.

Таблица 5 – Распределение композиций по простым жанрам

Наименование жанра	Кол-во композиций
Blues	354
Children music	219
Relax	109
Choir	58
Classical	1 451
Country	680
Dance	450
Disco	466
Drum	7
Electronic	727
Folk	385
Hip-hop	710
Indie music	312
Jazz	473
Latin music	408
Pop music	10 438
Rap music	280
Reggae	141
Religious	197
Rock	11 464
Soul	407
Soundtrack	882

### Модель жанров музыкальных композиций

В результате сбора и классификация обучающих данных изначально получено около 50 тысяч уникальных композиций, которым соответствует 1213 уникальных жанров. Для предсказания композиций это слишком много, поэтому было произведено распределение композиций по простым 20 жанрам. В данной работе было решено обучить модели нейронной сети по трём основным музыкальным жанрам: классика, рок и блюз. Для этого были взяты ранее распределённые по жанрам наборы midi файлов с общим количеством 13269. Помимо обученной нейронной сети и получены три модели музыкальных композиций по жанрам. Каждая модель используется для генерации композиций с определённым жанром, что позволяет повысить мелодичность жанровых музыкальных композиций.

Таблица 6 – Гиперпараметры нейронной сети и их значения

Гиперпараметр	Значение
Количество слоёв	7
Количество ячеек в слое	512
Размер партии обучения	512
Шаг обучения	0,001
Оптимизатор	Adam
Функция вычисления ошибок	Метод обратного распространения ошибки

Как уже было описано ранее, для генерации композиций используется нейронная сеть с долгой краткосрочной памятью (LSTM). Вся сеть содержит 7 слоёв, каждый из которых содержит 512 ячеек. Также был добавлен исключающий слой, который позволяет предотвратить переобучение сети.

Гиперпараметры нейронной сети описаны в таблице 6.

В результате обучения нейронной сети была получена модель жанров музыкальных композиций:

$$\text{GNRNN} = \langle I, \text{GNR}, \text{LSTM}, \text{DNS}, \text{ACT} \rangle, \quad (2)$$

где  $I$  – слой входной последовательности данных;  $\text{GNR}$  – слой, отвечающий за выбор нужных параметров для генерации мелодии в определённом жанре;  $\text{LSTM}$  – слой рекуррентной нейронной сети;  $\text{DNS}$  – dense слой, отвечающий за полное связывание каждого входного узла с выходным;  $\text{ACT}$  – слой (функция) активации.

Модель структуры музыкальной композиции

Для описания музыкальной композиции была разработана модель структуры музыкальной композиции. Ключевыми тэгами верхнего уровня текста являются два больших блока – теги *melody* и *harmony*, которые содержат описание мелодии и гармонии композиции соответственно. Также есть тэги для описания характеристик композиции – *tonality* и *tempo*, которые указывают на тональность и темп произведения. Данную модель можно описать в виде кортежа:

$$\text{MSC} = \langle I, \text{TN}, \text{TMP}, \text{HRM}, \text{MLD} \rangle, \quad (3)$$

где I – входная последовательность данных в виде композиции, TN – тэг для описания тональности композиции (tonality); TMP – тэг для описания темпа композиции (tempo); HRM – комплексный тэг для описания гармонической части композиции (harmony); MLD – комплексный тэг для описания мелодической части композиции (melody).

В свою очередь компонент HRM состоит из набора компонент CHRD:

$$\text{CHRD} = \langle \text{I}, \text{DUR}, \text{TP}, \text{NAME}, \text{MODE} \rangle, \quad (4)$$

где I – входная последовательность данных; DUR – свойство тега, описывающее длительность аккорда (duration); TP – тэг для типа аккорда (type); NAME – тэг для описания наименования аккорда (chord\_name); MODE – тэг для описания музыкального лада аккорда (mode).

Компонент MLD состоит из набора компонент NOTE, который можно представить в виде:

$$\text{NOTE} = \langle \text{I}, \text{DUR}, \text{NAME}, \text{OCTVE} \rangle, \quad (5)$$

где I – входная последовательность данных; TP – тэг для типа аккорда (type); NAME – тэг для описания наименования ноты (note\_name); OCTVE – тэг для описания октавы ноты (octave).

Разработанную модель можно представить в виде XML-текста, содержащего теги:

- tonality - тэг, обозначающий тональность композиции.

Пример: `<tonality>d_minor</tonality>`. Вместо d\_minor может быть любая другая тональность (c\_minor, g\_sharp\_major и др) [1].

- tempo - тэг, обозначающий темп композиции.

Пример: `<tempo value="60"></tempo>` [40]

• harmony – тэг, содержащий описание гармонии. Внутри этого тэга находится тэг chord

- chord - тэг, внутри которого находятся тэги type, chord\_name и mode.

Свойством тэга является duration, обозначающее длительность аккорда.

• type – тэг, важной частью которого является свойство value, обозначающее тип аккорда – стандартный (standard) или собственный (own)

- chord\_name – тэг, обозначающий для стандартных аккордов, название ноты тональности (c, d, e, f, g, a, h и др.)
- mode – тэг, обозначающий музыкальный лад аккорда (major или minor).
- в случае, если тип аккорда own, то вместо тэгов type, chord\_name и mode, внутри тэга chord пишется тэг notes, в свойстве value которого, пишутся ноты и длительности создаваемого аккорда, идущие через запятую, пары разделены точкой с запятой.

Пример: <notes value="f,2;c,3;f,3;g,3;c,4"/>

- После закрытия тэга chord следует тэг melody, обозначающий мелодическую часть композиции. Этот тэг, также как и тэг chord, требует явного закрытия тэгом </chord> и </melody> соответственно.
- Внутри тэга melody расположен тэг note, note\_name и octave. Первый содержит свойство duration, обозначающее длительность ноты, второй содержит свойство value, которое обозначает имя аккорда (c, d, e, f, g, a, h и др.).
- Наконец третий тэг octave имеет свойство value, обозначающее октаву ноты [1].

Пример xml текста [40]:

```
<doc>
<tonality>d_minor</tonality>
<tempo value="60"></tempo>
<harmony>
<chord      duration="4"><type      value="standard"/><chord_name
value="d"/><mode value="minor"/></chord>
<chord      duration="4"><type      value="standard"/><chord_name
value="d"/><mode value="minor"/></chord>
<chord      duration="4"><type      value="standard"/><chord_name
value="d"/><mode value="minor"/></chord>
...
.....
```

```

<chord           duration="4"><type           value="own"/><notes
value="f,2;c,3;f,3;g,3;c,4"/></chord>
</harmony>
<melody>
<note duration="2"><note_name value="silence"/></note>
<note duration="8"><note_name value="f"/><octave value="4"/></note>
<note duration="8"><note_name value="g"/><octave value="4"/></note>
<note duration="8"><note_name value="f"/><octave value="4"/></note>
<note duration="4"><note_name value="c"/><octave value="5"/></note>
<note duration="4"><note_name value="a"/><octave value="4"/></note>
.....
<note duration="8"><note_name value="f"/><octave value="4"/></note>
</melody>
</doc>

```

Таким образом, рассмотренная выше структура, позволяет описать мелодию и гармонию произведения, однозначно определить каждый аккорд и ноту в произведении, а также произвести удобный обмен информацией между двумя модулями системы [1].

Разработанную модель можно представить в виде XSD диаграммы, представленной на рисунке 26.

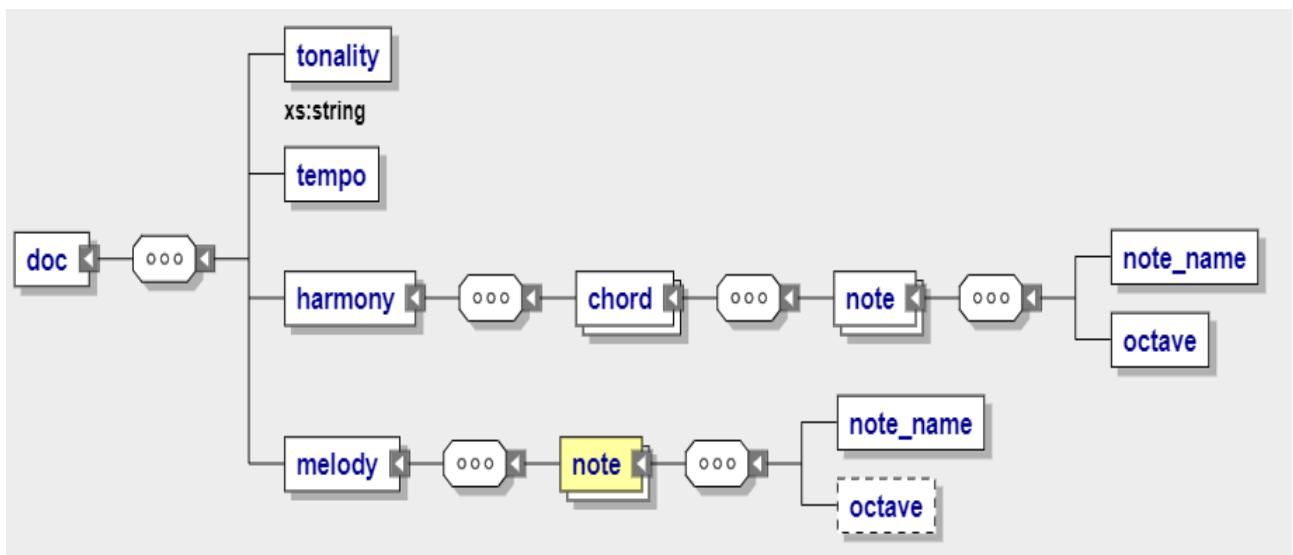


Рисунок 26 – Модель структуры музыкальной композиции в виде XSD

## 2.3 Метод воспроизведения звуков музыкальных инструментов по структурированному представлению музыкальной композиции

Последним этапом синтезирования цельной музыкальной композиции по цветовому изображению с использованием нейронных сетей является непосредственный синтез звуков. Для этого был разработан метод воспроизведения звуков музыкальных инструментов по структурированному представлению музыкальной композиции, который схематично представлен на рисунке 27.

Метод состоит из пяти основных этапов – предобработка xml-текста, получение характеристик композиции из текста, синтез мелодии, синтез гармонии и объединение частей в одну композицию. В данном методе используется оригинальная собранная база данных сэмплов – музыкальных файлов в формате wav, каждый из которых содержит звук отдельной ноты в определённой октаве для определённого музыкального инструмента. Данная база данных содержит четыре музыкальных инструмента – акустическая гитара, электрогитара, электрогитара с эффектом перегруза и фортепиано. Каждый инструмент содержит четыре октавы.

Таким образом, в данном методе был использован ранее выбранный метод сэмплинга – синтез звуков путём последовательного или параллельного воспроизведения различных заранее записанных звуков [63].

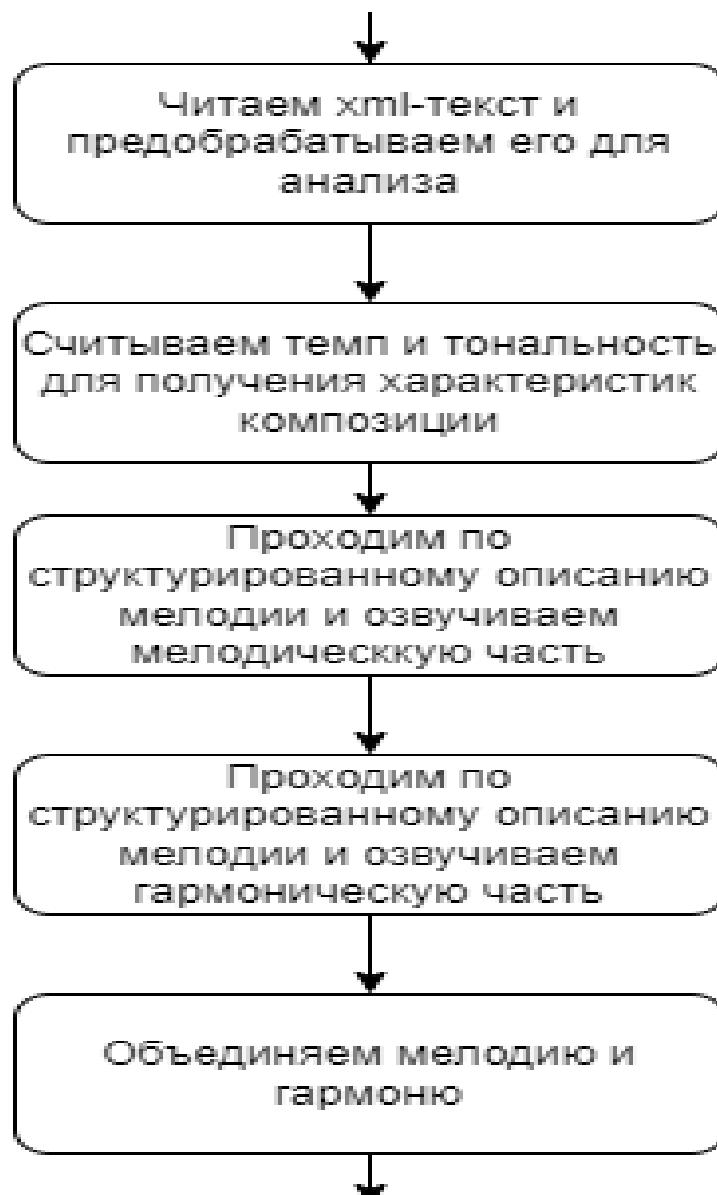


Рисунок 27 – Метод воспроизведения звуков по структурированному описанию музыкальной композиции

## 2.4 Выводы по второй главе

Во второй главе были описаны основные разработанные методы для синтеза музыкальной композиции по цветовой гамме изображений. Для этого были описаны следующие методы:

- комплексный метод синтезирования цельной музыкальной композиции по цветовому изображению с использованием нейронных сетей;

- метод структурированного представления музыкальной композиции;
- метод воспроизведения звуков музыкальных инструментов по структурированному представлению музыкальной композиции;

Также была описана модель структуры музыкальной композиции, представленная в виде xml-текста.

Описан сбор обучающих данных и обучение нейронной сети с долгой краткосрочной памятью (LSTM) для генерации композиций. Вся сеть содержит 7 слоёв, каждый из которых содержит 512 ячеек. Также был добавлен исключающий слой, который позволяет предотвратить переобучение сети.

### **3 Алгоритмическое обеспечение системы синтезирования музыкальной композиции по цветовому изображению**

#### **3.1 Алгоритмы анализа изображений**

Общий вид алгоритма анализа изображений представлен на рисунке 28.

Данный алгоритм позволяет, по загруженному пользователем изображению, получить лист (массив) простых цветов, каждым элементом которого является набор – название, насыщенность и яркость.

Входом данного алгоритма является путь к загруженному пользователем изображению.

Выходом алгоритма является лист (массив) названий, насыщенностей и яркостей каждого пикселя изображения.

Считывание изображения происходит с помощью библиотеки OpenCV:

```
input_img_bgr = cv2.imread(self.path_to_image)
```

После считывания изображение необходимо преобразовать в цветовое пространство HSV, для этого также используется библиотека OpenCV:

```
input_img_hsv = cv2.cvtColor(input_img_bgr, cv2.COLOR_BGR2HSV)
```

После этого необходимо преобразовать каждый пиксель изображения к удобному для дальнейшего анализа виду. Алгоритм преобразования представлен на рисунке 29.

Для преобразования, алгоритм проходит по каждому пикслю изображения и определяет название цвета по параметру hue, насыщенность по параметру saturation и яркость по параметру brightness.

Название определяется по принадлежности значения hue к диапазону цвета (красный – [0; 5] или [176; 180], красно-оранжевый – [6; 10], оранжевый – [11; 20], жёлто-оранжевый – [21; 25], жёлтый – [26; 30], жёлто-зелёный – [31; 40], зелёный – [41; 75], сине-зелёный – [75; 100], синий – [101; 120], сине-фиолетовый – [121; 140], фиолетовый – [141; 160], розовый – [161; 175]) [64, 65]



Рисунок 28 – Алгоритм анализа изображений. Общий вид

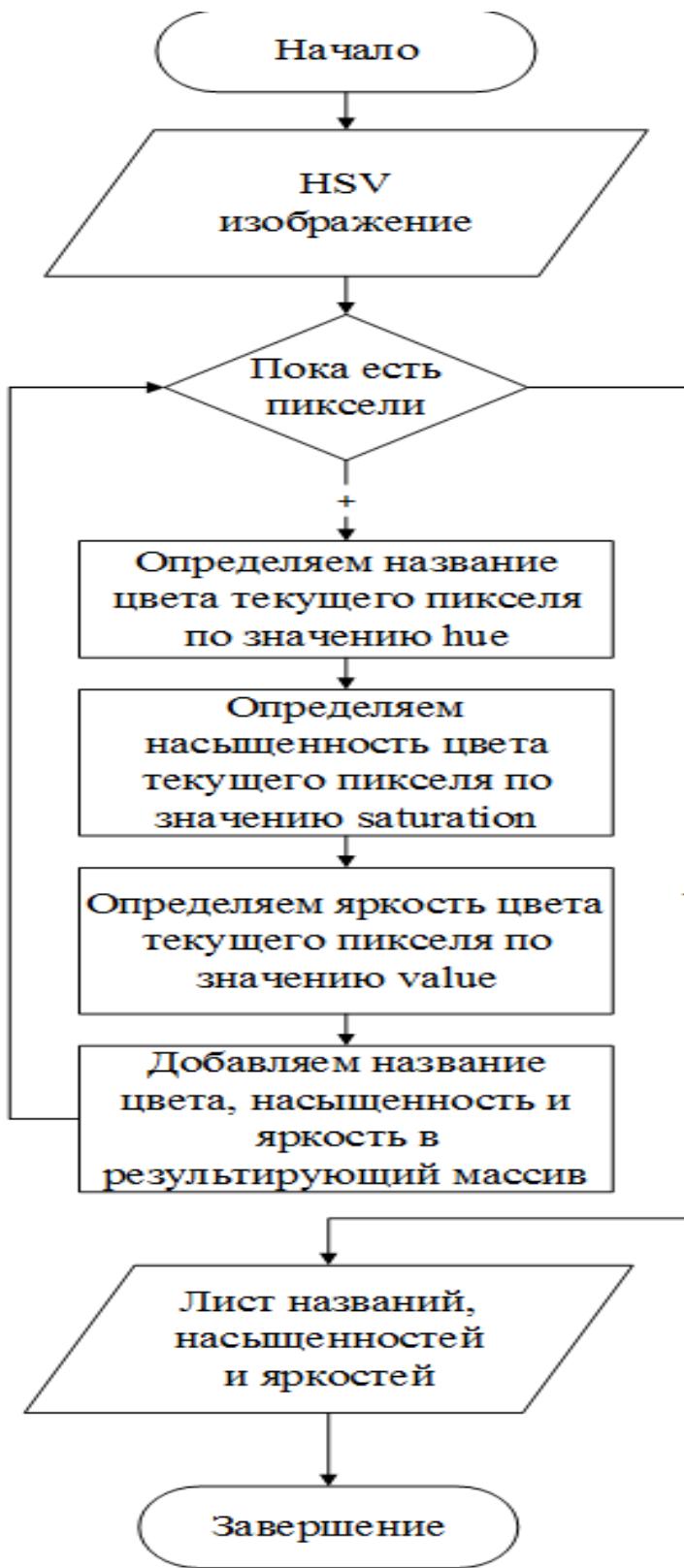


Рисунок 29 – Алгоритм анализа изображений. Получение массива простых цветов

Насыщенность и яркость пересчитываются по параметрам saturation и brightness, чтобы получались числа [0; 4] и [0; 2] соответственно.

Далее происходит объединение одинаковых подряд идущих цветов по алгоритму, представленному на рисунке 30 [66].

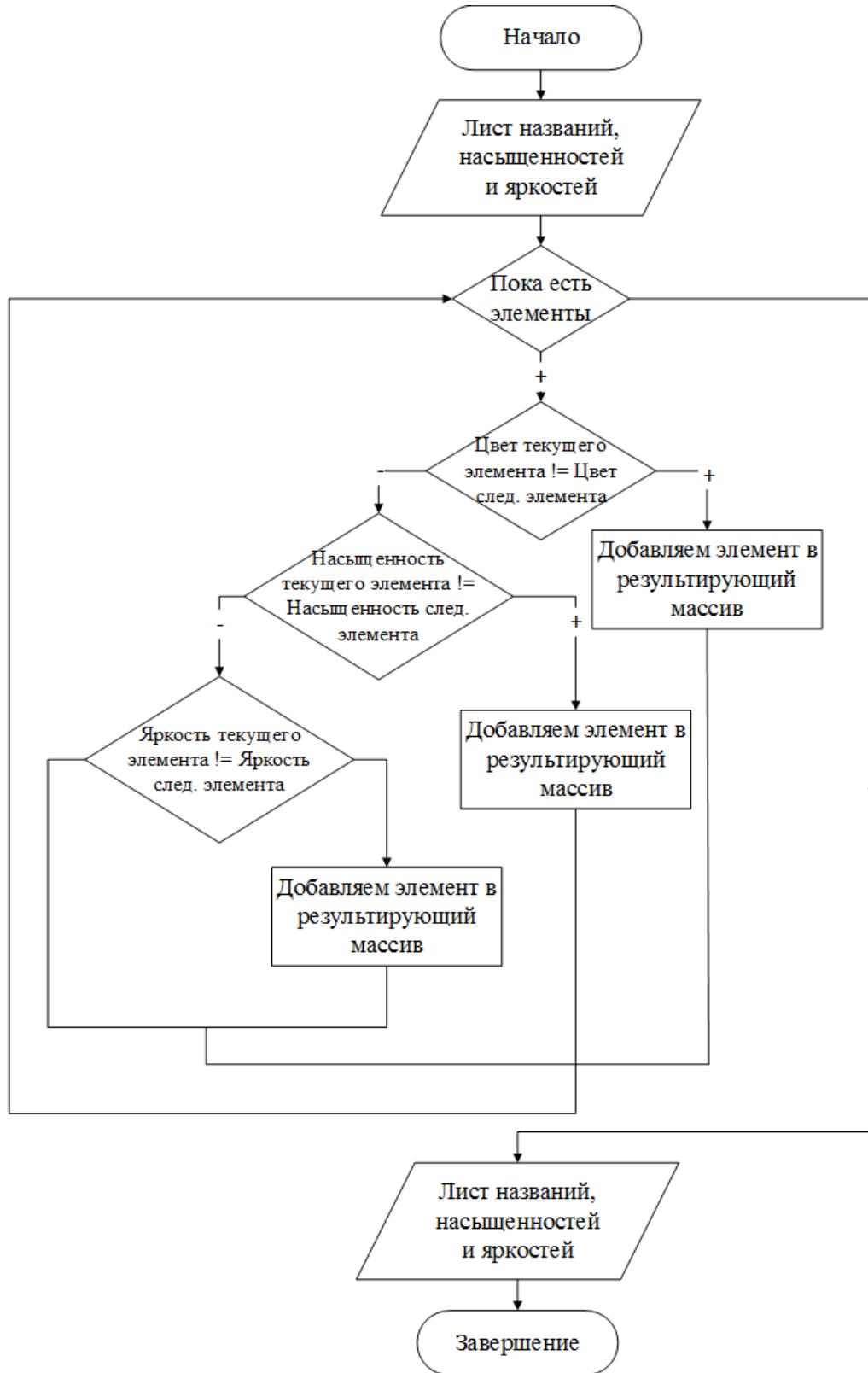


Рисунок 30 – Алгоритм анализа изображений. Объединение одинаковых подряд идущих простых цветов

### 3.2 Алгоритмы структурированного представления музыкальной композиции

После получения и преобразования названий, насыщенностей и яркостей каждого пикселя загруженного изображения, необходимо соотнести художественные характеристики с музыкальными. Общий вид алгоритма соотнесения представлен на рисунке 31.

Входом алгоритма является лист (массив) названий, насыщенностей и яркостей каждого пикселя изображения (массив простых цветов).

Выходом алгоритма является xml – текст, содержащий описание музыкальной композиции.



Рисунок 31 – Алгоритмы структурированного представления музыкальной композиции. Общий вид

Первым шагом алгоритма является получение преимущественного цвета на изображении, алгоритм его нахождения представлен на рисунке 32.

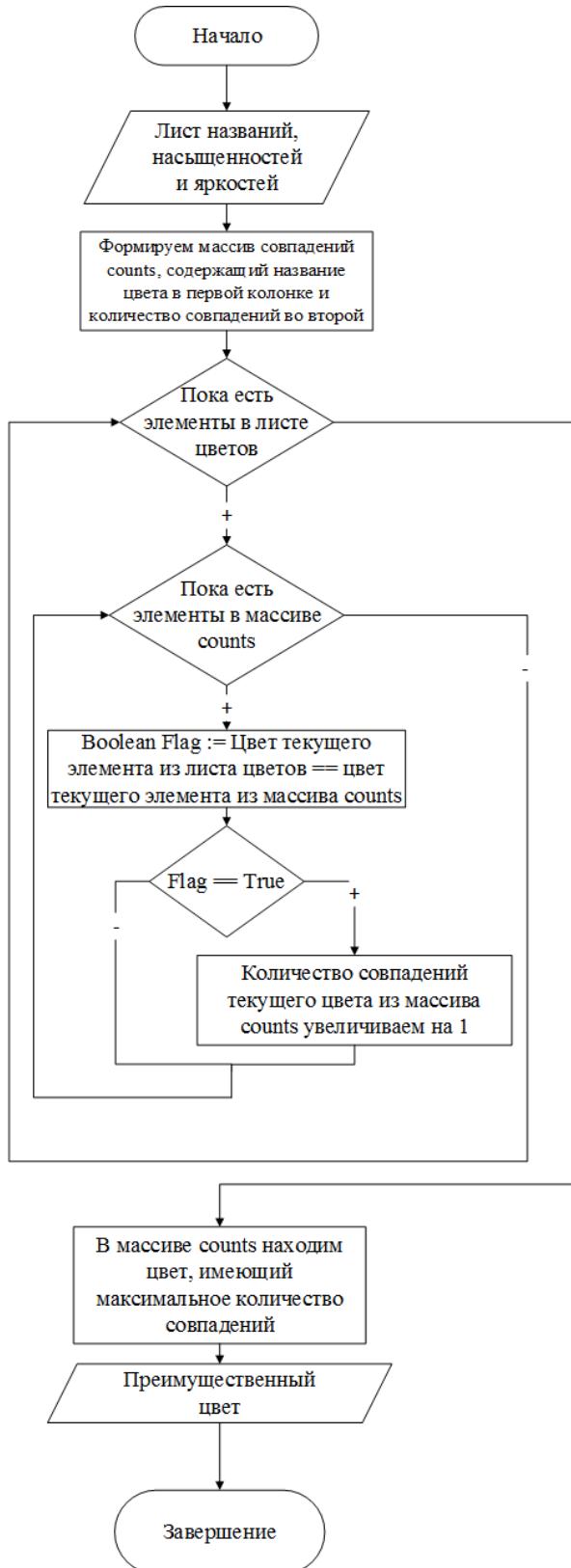


Рисунок 32 – Алгоритмы структурированного представления музыкальной композиции. Определение преимущественного цвета

Нахождение преимущественного цвета осуществляется путём подсчёта в листе названий, насыщенностей и яркостей, количества вхождений каждого из 12 цветов. Цвет, имеющий наибольшее количество совпадений, является преимущественным.

Следующим шагом алгоритма соотнесения цветовых и музыкальных характеристик является нахождение тональности будущего произведения. Алгоритм поиска тональности представлен на рисунке 33.

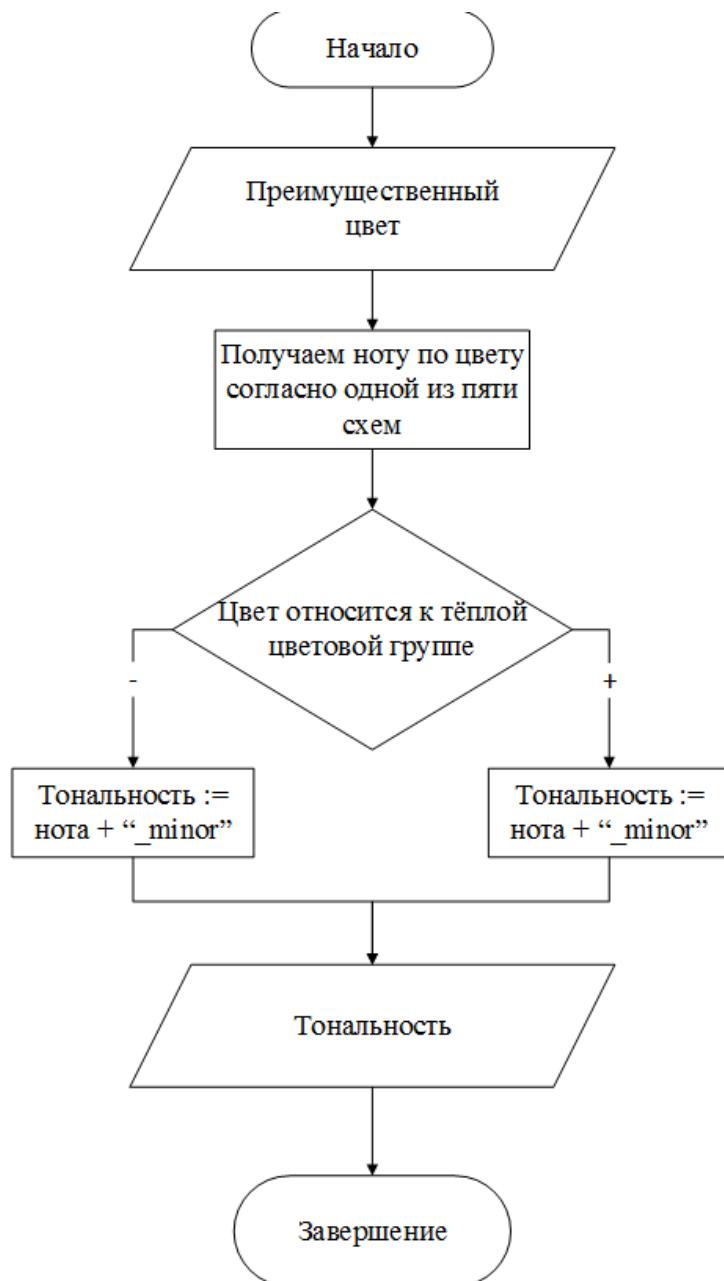


Рисунок 33 – Алгоритмы структурированного представления музыкальной композиции. Получение тональности

Для получения тональности необходимо соотнести полученный преимущественный цвет с одной из схем соотнесения цветов и нот (схемы Ньютона, Кастеля, Риминтона, Эппли и Бельмонта).

После получения тональности, необходимо определить мажорное ли будет произведение или минорное. Для этого определяется вхождение цвета в тёплую (жёлтый, жёлто-оранжевый, оранжевый, оранжево-красный, красный и розовый) или холодную (синий, сине-зелёный, зелёный, жёлто-зелёный, фиолетовый и фиолетово-синий) цветовую группу, если в первую – то тональность мажорная, иначе минорная [67].

Следующим этапом соотнесения цветовых и музыкальных характеристик является нахождение темпа произведения. Темп произведения находится по тональности произведения, если тональность мажорная, то темп 120, иначе 85 [68].

После получения темпа, можно приступить к определению мелодии произведения. Алгоритм получения мелодической последовательности представлен на рисунке 34.

Для получения мелодической последовательности, для начала необходимо соотнести хроматическую гамму тоники с цветовым кругом, то есть получить соответствие цветов и нот для мелодической части произведения [40]. Получение такого соответствия представлено на рисунке 35.

Затем необходимо пройти по каждому элементу из листа простых цветов и определить, является ли цвет текущего элемента белым(чёрным), если да, то добавить в результирующий массив паузу, имеющую длительность согласно насыщенности (если насыщенность 0 или 1, то длительность половинная, если 2, то четвертная, если 3, то восьмая, если 4, то шестнадцатая). Если же цвет не белый, то необходимо определить ноту по названию текущего цвета, согласно полученному ранее соответству цветов и нот. Затем необходимо добавить полученную ноту в результирующий массив, с длительностью согласно насыщенности, и октавой, согласно яркости. И заключительным этапом

получения мелодической последовательности является преобразование одинаковых подряд идущих нот в одну.

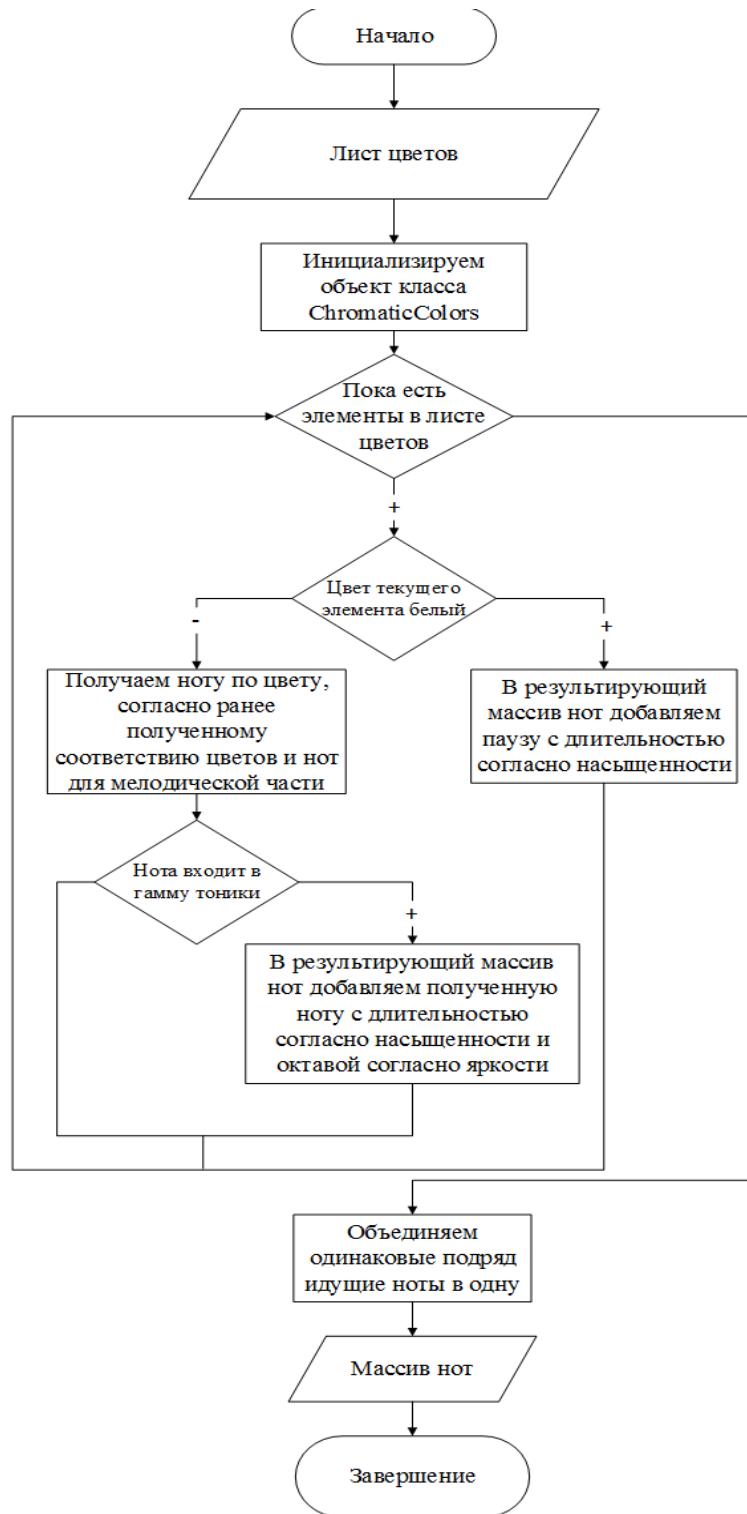


Рисунок 34 – Алгоритмы структурированного представления музыкальной композиции. Получение мелодической последовательности

Следующим этапом получения мелодической части произведения является «предсказание» дальнейшего произведения по полученному ранее массиву нот. Алгоритм «предсказания» дальнейшего произведения представлен на рисунке 36.

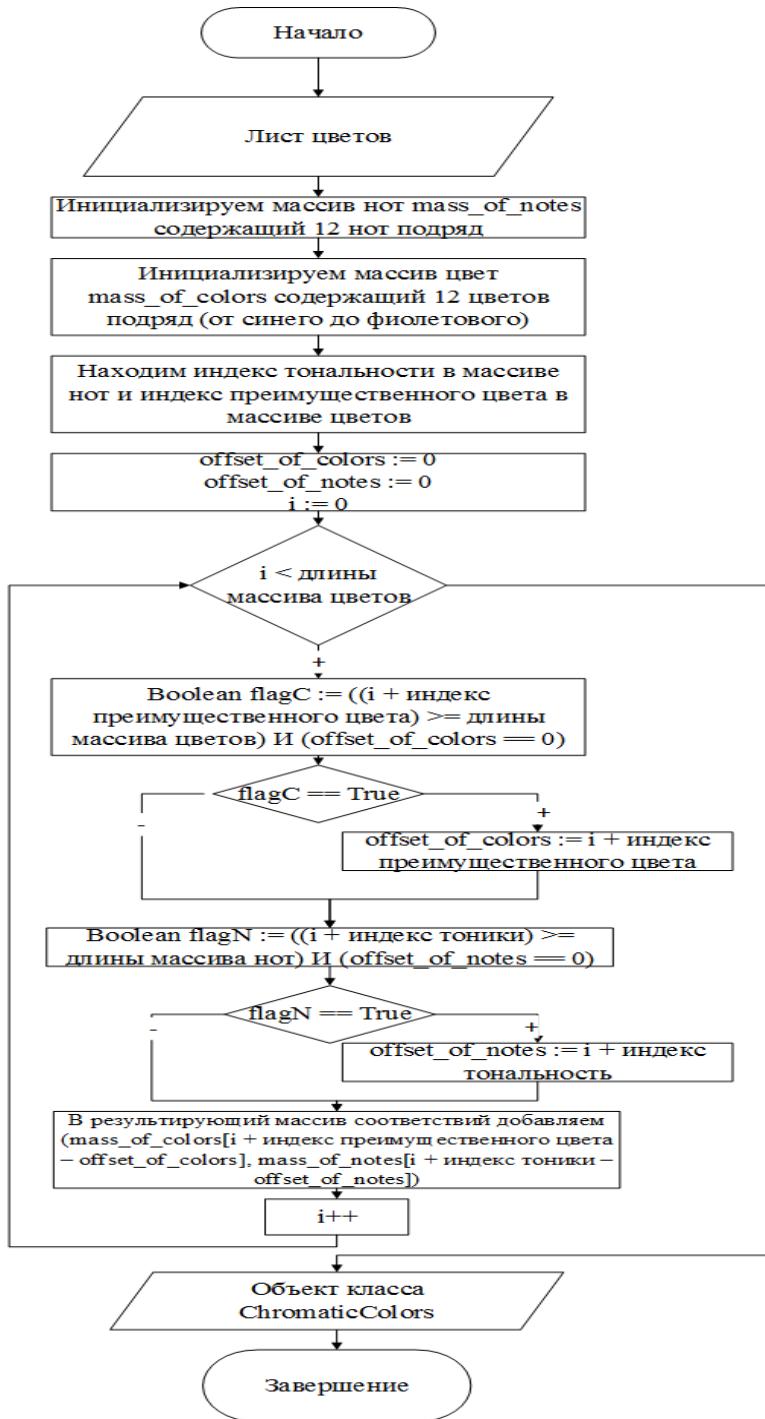


Рисунок 35 – Алгоритмы структурированного представления музыкальной композиции. Инициализация объекта класса ChromaticColors



Рисунок 36 – Алгоритм предсказания композиции

Следующим шагом алгоритма соотнесения цветовых и музыкальных характеристик является нахождение количества тактов в произведении (рис. 37) [69].

Следующим этапом соотнесения цветовых и музыкальных характеристик является нахождение гармонии произведения (рис. 38). Для нахождения гармонии необходимо найти гамму тональности, и получить музыкальный лад тональности (мажор или минор). После этого необходимо найти в гамме тонику, доминанту и субдоминанту, построить на данных ступенях аккорды [70]. Затем необходимо

добавить их в результирующий массив аккордов по следующему правилу: пока количество тактов не превышено добавить 4 тонических аккорда, 4 доминантных и 8 субдоминантных аккордов [52, 71].

И заключительным этапом соотнесения цветовых и музыкальных характеристик является формирование xml – текста, по полученным массивам нот и аккордов. Алгоритм формирования текста представлен на рисунке 39 [72].

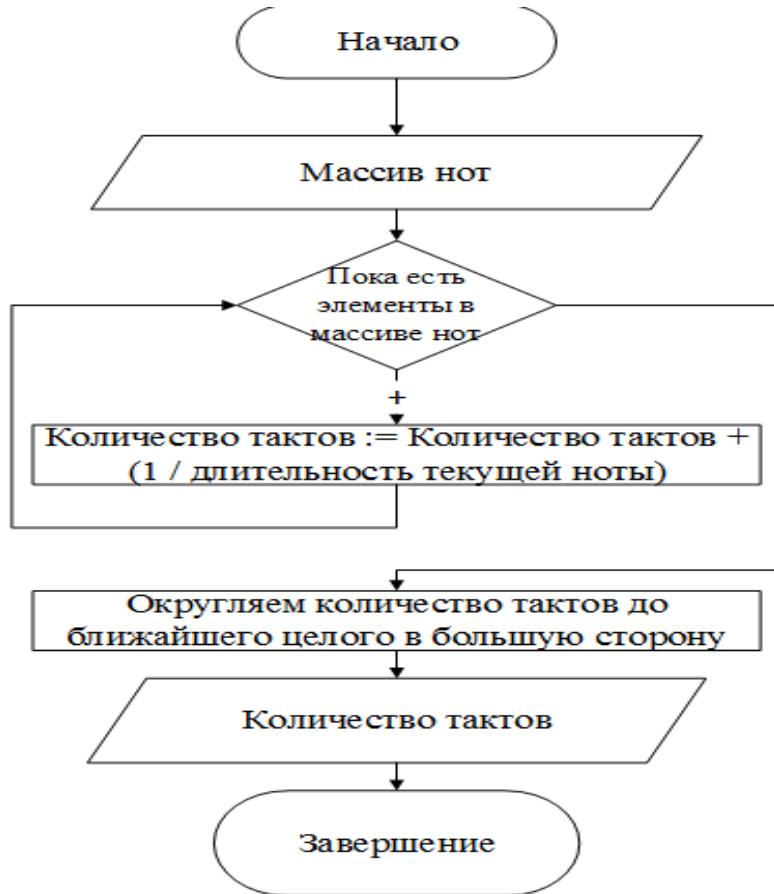


Рисунок 37 – Алгоритмы структурированного представления музыкальной композиции. Определение количества тактов

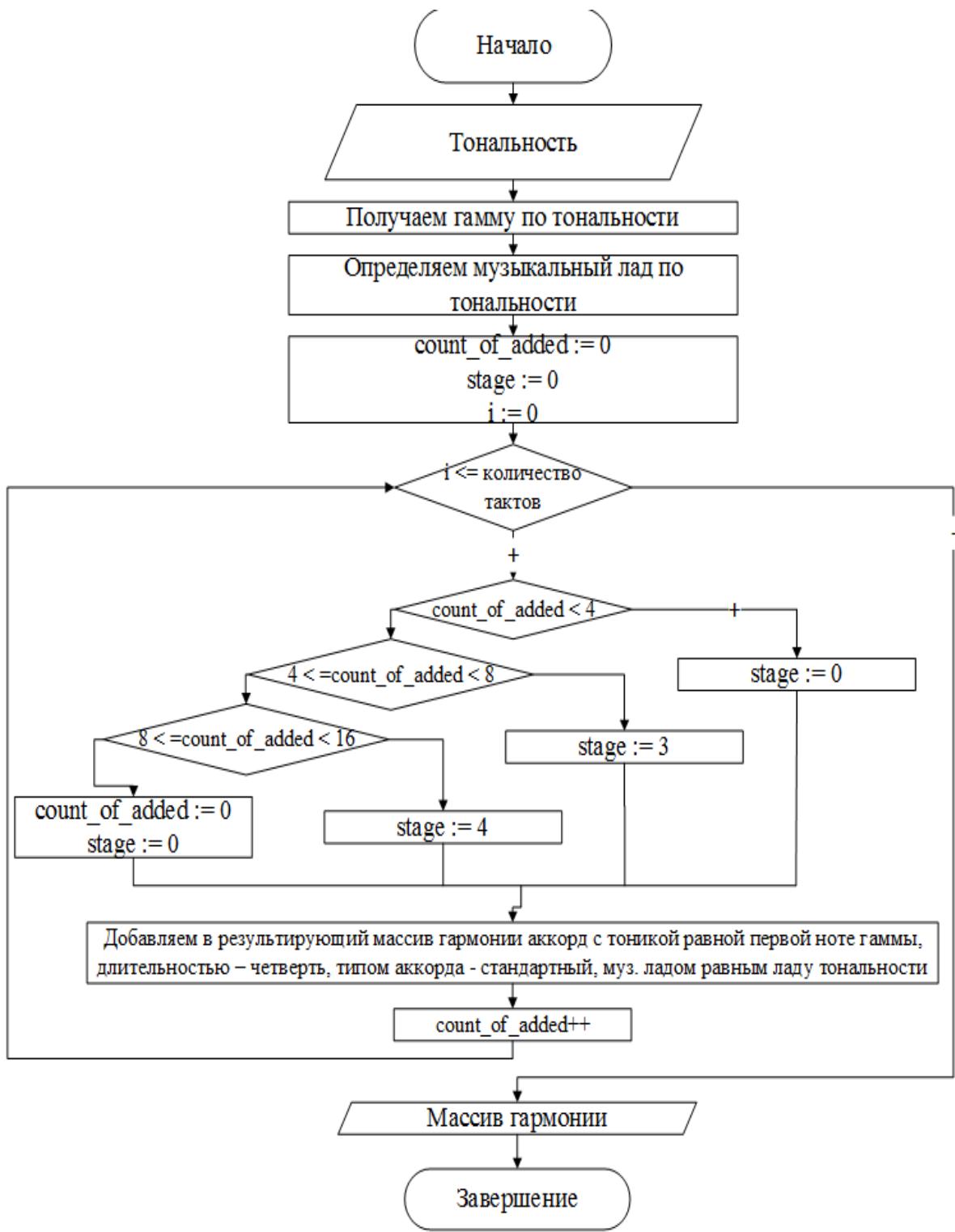


Рисунок 38 – Алгоритмы структурированного представления музыкальной композиции. Получение гармонической последовательности

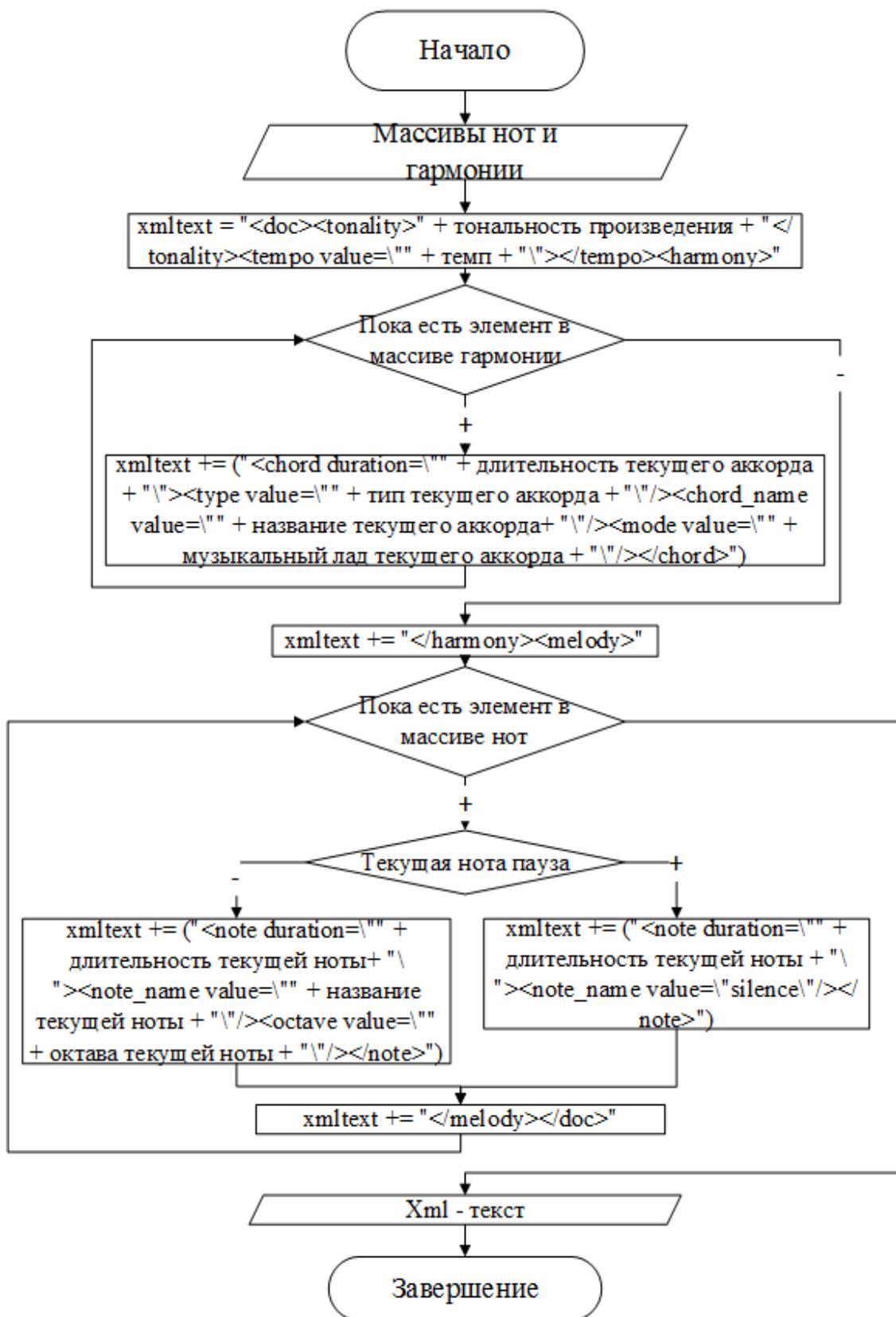


Рисунок 39 – Алгоритмы структурированного представления музыкальной композиции. Создание xml текста

### 3.3 Алгоритм синтеза звуков по структурированному описанию музыкальной композиции

#### Алгоритмы синтеза мелодии

После формирования xml – текста, содержащего формализованное описание музыкальной композиции, необходимо провести синтез звука по сформированному тексту. Общий вид алгоритма синтеза мелодии представлен на рисунке 40.

Входом алгоритма является xml – текст.

Выходом – Объект класса AudioSegment, содержащий мелодию.

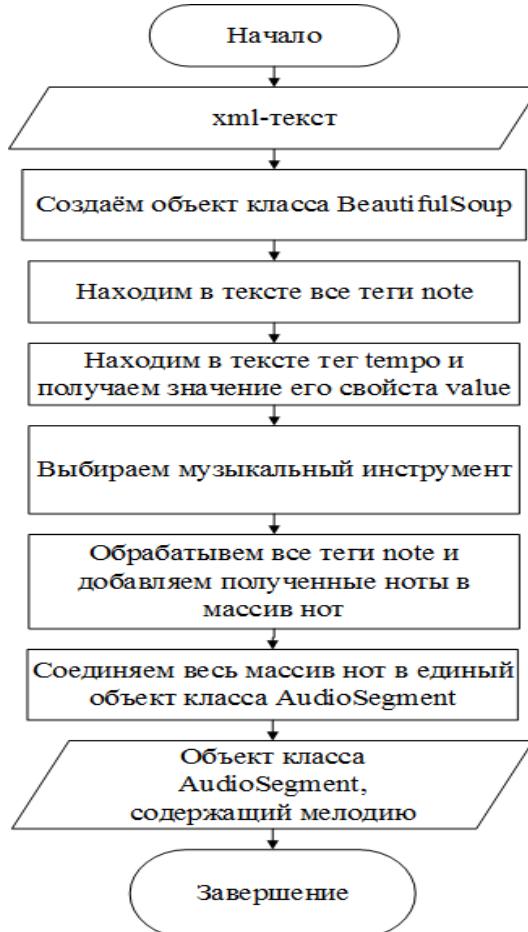


Рисунок 40 – Алгоритм синтеза звуков по структурированному описанию музыкальной композиции. Алгоритм синтеза мелодии – общий вид

Первым этапом синтеза мелодии является создание объекта класса BeautifulSoup, для парсинга xml – текста. Для этого необходимо либо передать в конструктор сформированный xml – текст, либо, если это необходимо, прочитать xml файл по переданному пути, и затем передать в конструктор прочитанный xml – текст.

Затем в xml – тексте необходимо найти все теги note, отвечающие за мелодическую часть произведения, делается это с помощью библиотеки BeautifulSoup следующим образом:

```
notes = xml_music_notation.doc.melody.findAll("note")
```

После этого необходимо найти в тексте тег tempo и получить значение его свойства value:

```
tempo = xml_music_notation.doc.tempo["value"]
```

Следующим шагом синтеза мелодии является выбор музыкального инструмента. Затем, для синтеза мелодии, необходимо обработать все теги note (рис. 41). В каждом теге note, необходимо найти тег note\_name и получить значение его свойства value, для нахождения названия ноты. Затем необходимо получить значение свойства value тега duration, для определения длительности ноты. После этого, если нота является паузой, то необходимо добавить в результирующую композицию паузу, согласно значению свойства value тега duration. Если же нота не является паузой, то необходимо получить значение тега octave и добавить в результирующую композицию ноту с длительностью duration и октавой octave.

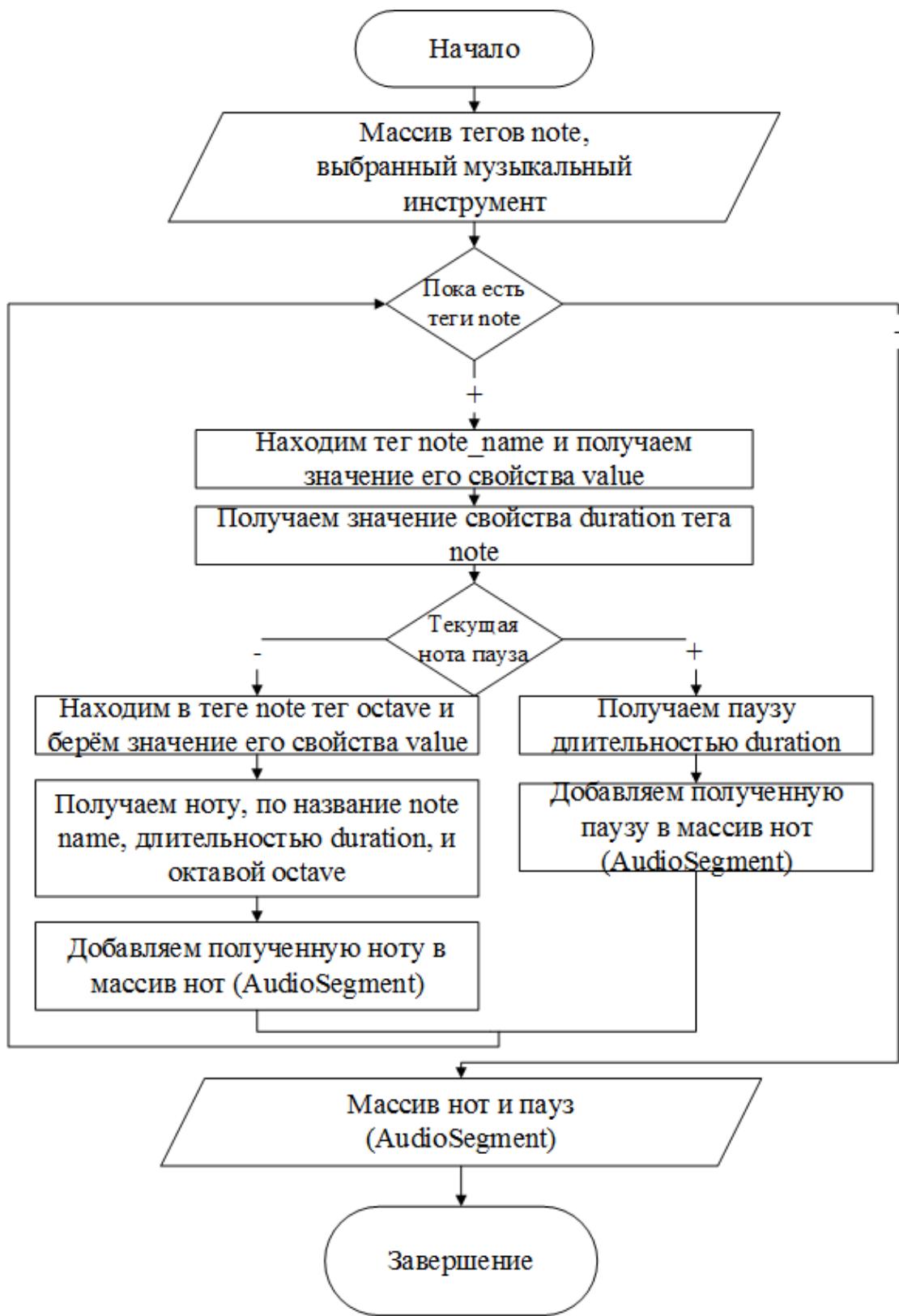


Рисунок 41 – Алгоритм синтеза звуков по структурированному описанию музыкальной композиции. Алгоритм синтеза мелодии – обработка всех тегов note

Для получения ноты по названию, длительности и октаве, необходимо получить путь к текущему расположению, затем находим папку /samples/, в ней находим поддиректорию с названием выбранного музыкального инструмента, если необходимо добавить паузу, то читаем файл silence.wav, если нет, то находим поддиректорию с выбранной октавой и в ней файл с названием ноты (например, c\_sharp.wav) (рисунок 42).

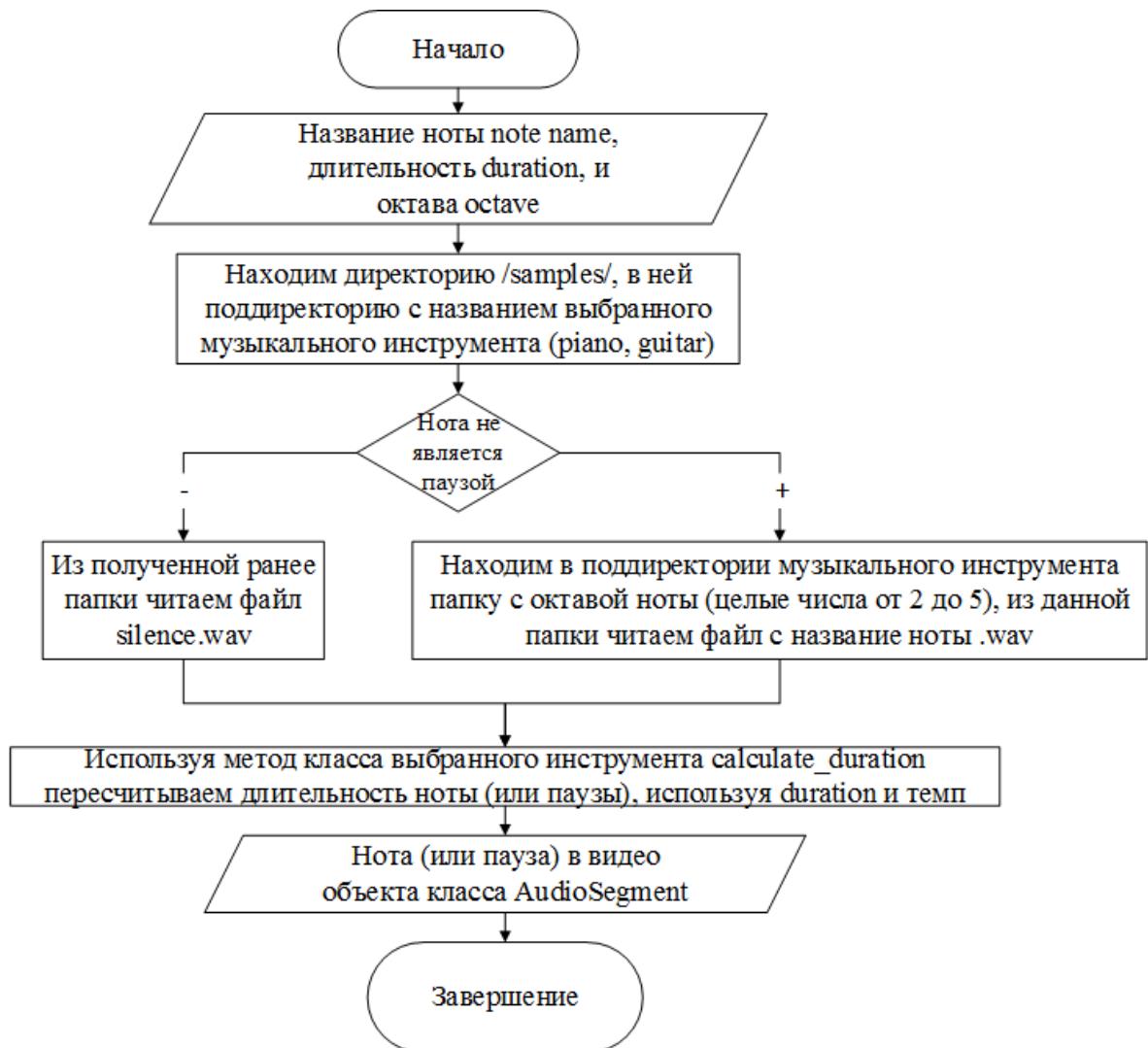


Рисунок 42 – Алгоритм синтеза звуков по структурированному описанию музыкальной композиции. Алгоритм синтеза мелодии – получение ноты

После получения массива нот, представляющих собой считанные wav файлы (а точнее объекты класса AudioSegment), необходимо объединить все ноты

в единый музыкальный отрывок. Алгоритм соединения представлен на рисунке 43.

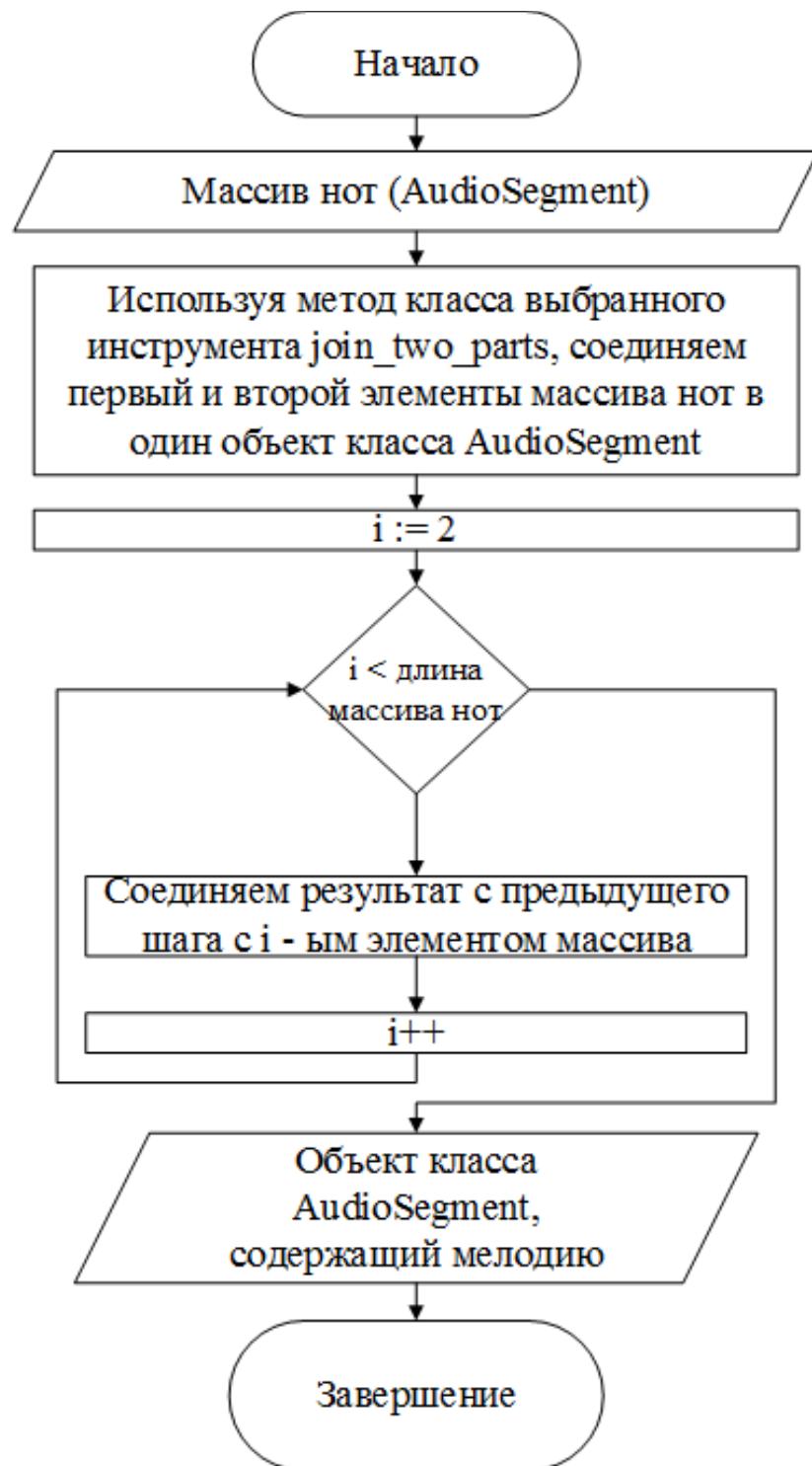


Рисунок 43 – Алгоритм синтеза звуков по структурированному описанию музыкальной композиции. Алгоритм синтеза мелодии – соединение массива нот в единый музыкальный отрывок

## Алгоритмы синтеза гармонии

После синтеза мелодии, необходимо осуществить синтез гармонии, общий вид алгоритма представлен на рисунке 44.

Входом алгоритма является xml – текст.

Выходом – Объект класса `AudioSegment`, содержащий гармонию.

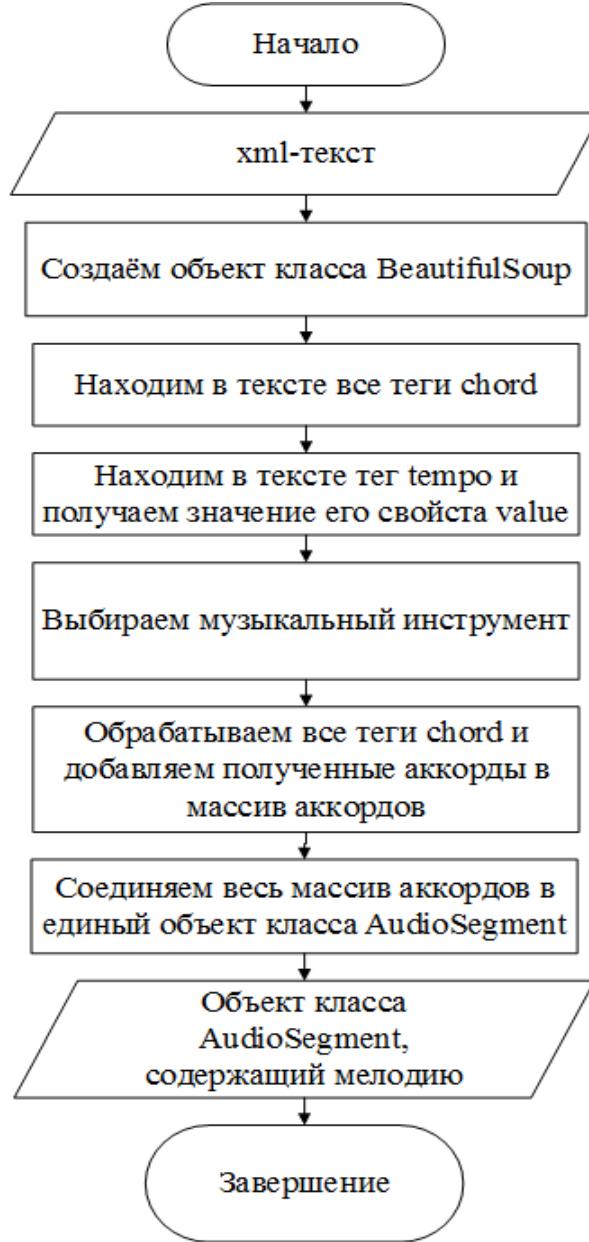


Рисунок 44 – Алгоритм синтеза звуков по структурированному описанию музыкальной композиции. Алгоритм синтеза гармонии – общий вид

Алгоритм имеет схожее с алгоритмом синтеза мелодии начало, а именно создание объекта класса `BeautifulSoup` происходит точно таким же образом.

Затем также с помощью библиотеки BeautifulSoup находятся все теги chord в тексте, определяется темп произведения и происходит выбор музыкального инструмента, как и в предыдущем алгоритме.

Следующим шагом является обработка тегов chord (рис. 45). Для каждого тега chord в тексте, находится значение свойства value тега type, для определения типа аккорда (стандартный, не стандартный).

Если аккорд является стандартным, то находятся теги chord\_name и mode, для получения названия аккорда и его музыкального лада. Затем необходимо получить аккорд по его названию и музыкальному ладу (рис. 46). Для этого по названию аккорда определяется последовательность нот, входящих в каждый из стандартных аккордов [73].

Затем происходит чтение каждой ноты из директории и объединение с предыдущими.

Пример формирования аккорда До Мажор представлен на рисунке 47.

Если же аккорд является не стандартным, то необходимо получить значение тега notes аккорда, разделить полученное значение по символу ‘ ; ’, считать и объединить каждую ноту в единый аккорд (рис. 48).

И, наконец, завершающим этапом синтеза гармонии является соединение полученного массива аккордов в единый музыкальный отрывок.

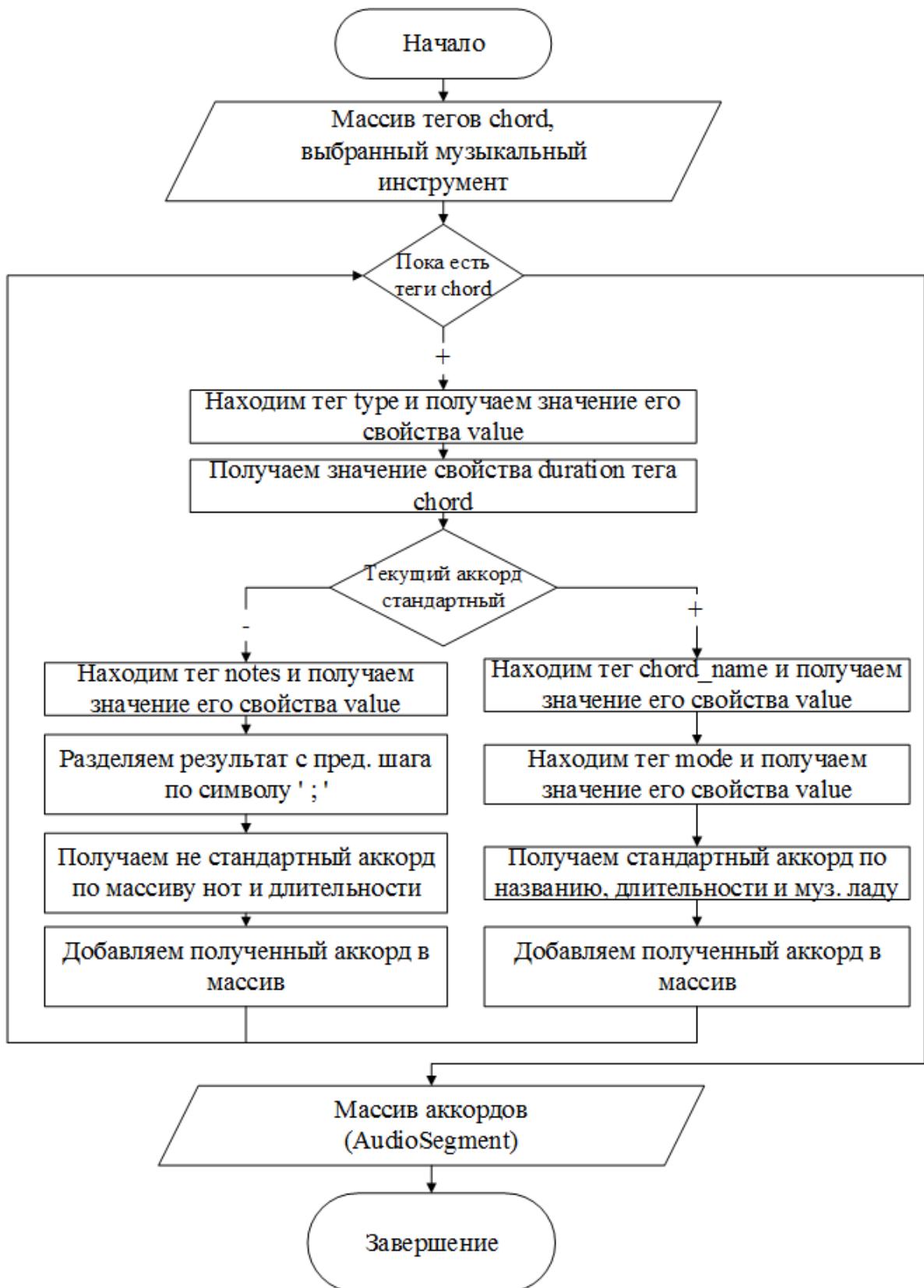


Рисунок 45 – Алгоритм синтеза звуков по структурированному описанию музыкальной композиции. Алгоритм синтеза гармонии – обработка тегов chord

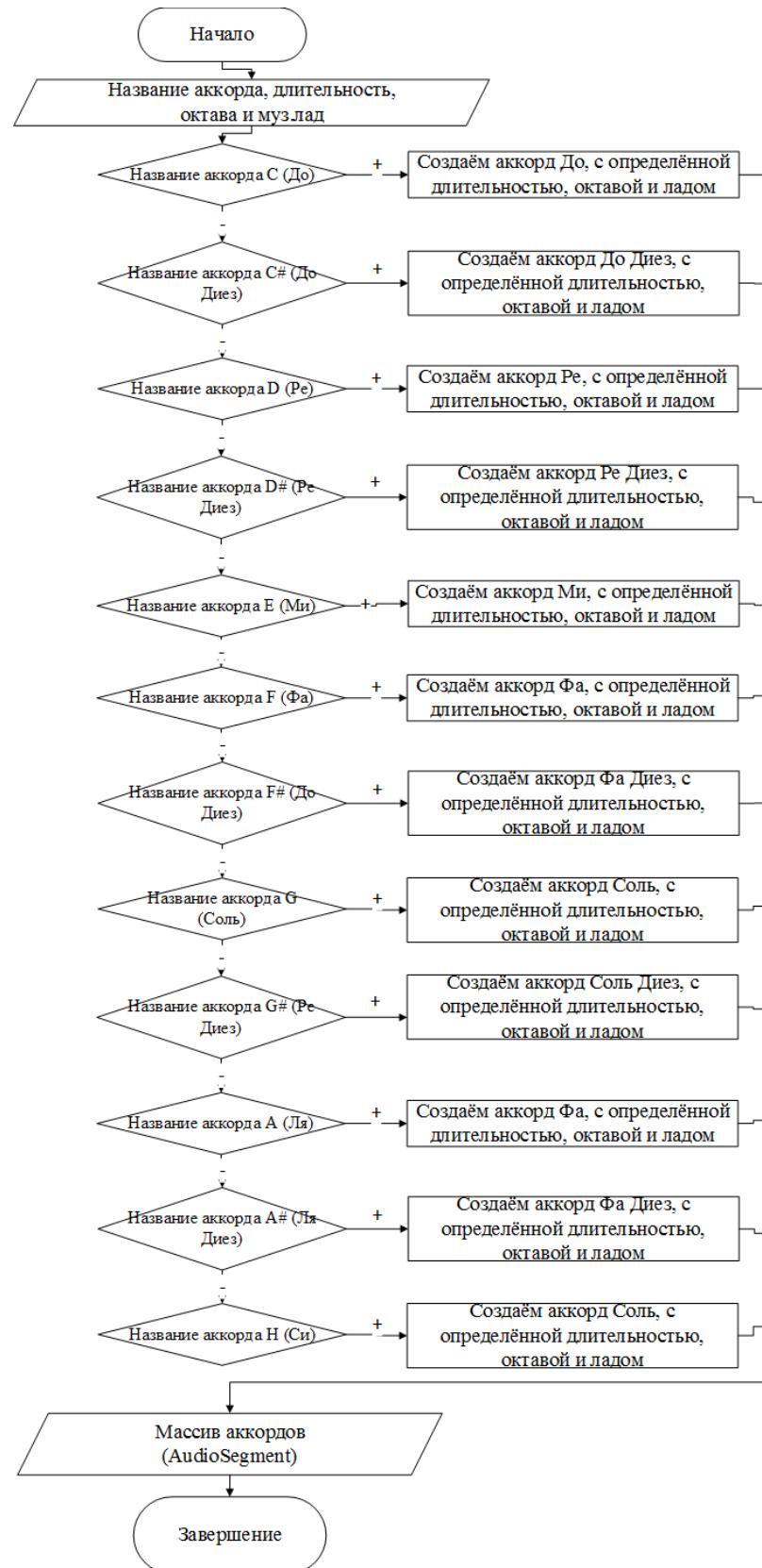


Рисунок 46 – Алгоритм синтеза звуков по структурированному описанию музыкальной композиции. Алгоритм синтеза гармонии – получение стандартного аккорда

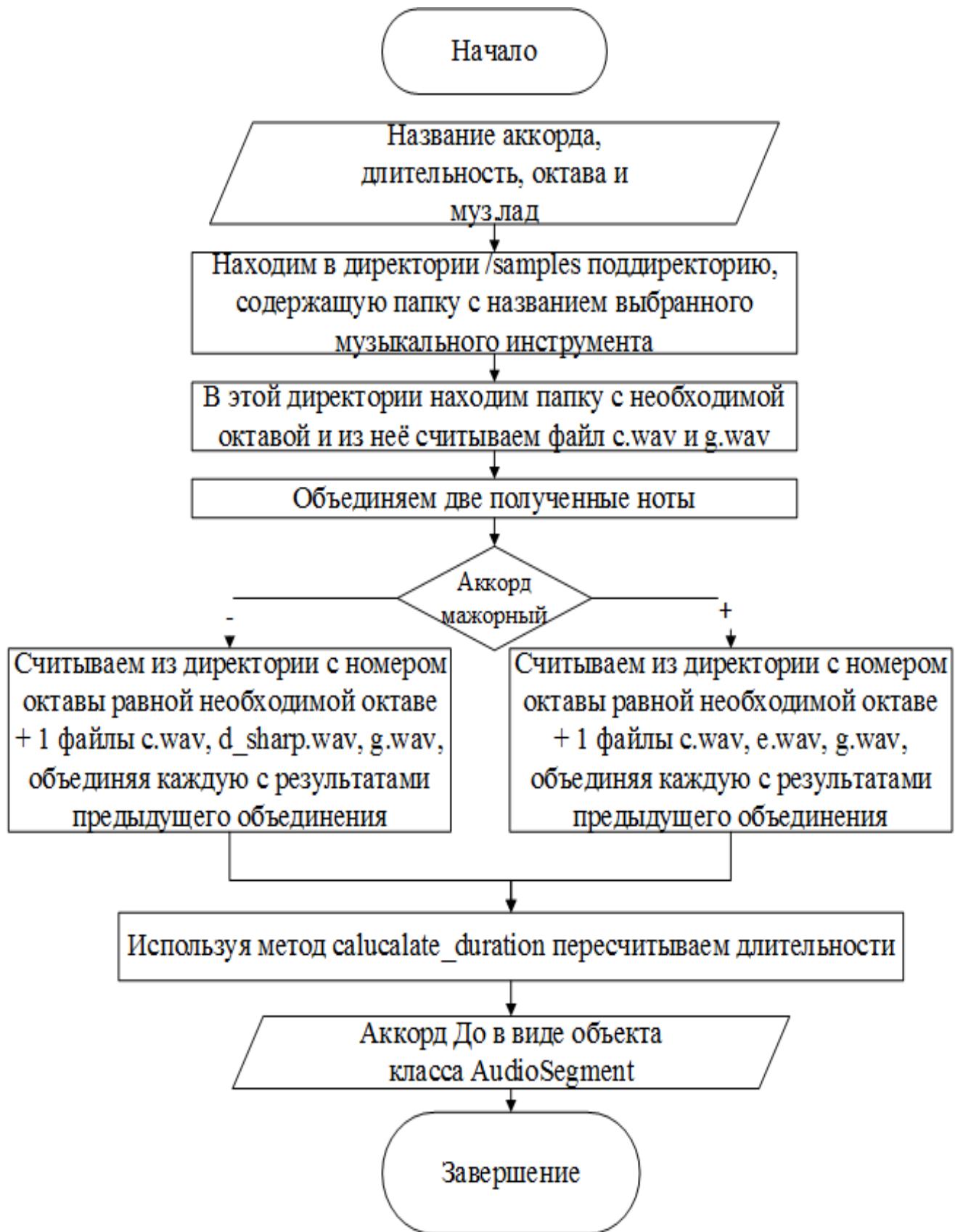


Рисунок 47 – Алгоритм синтеза звуков по структурированному описанию музыкальной композиции. Алгоритм синтеза гармонии – создание аккорда До

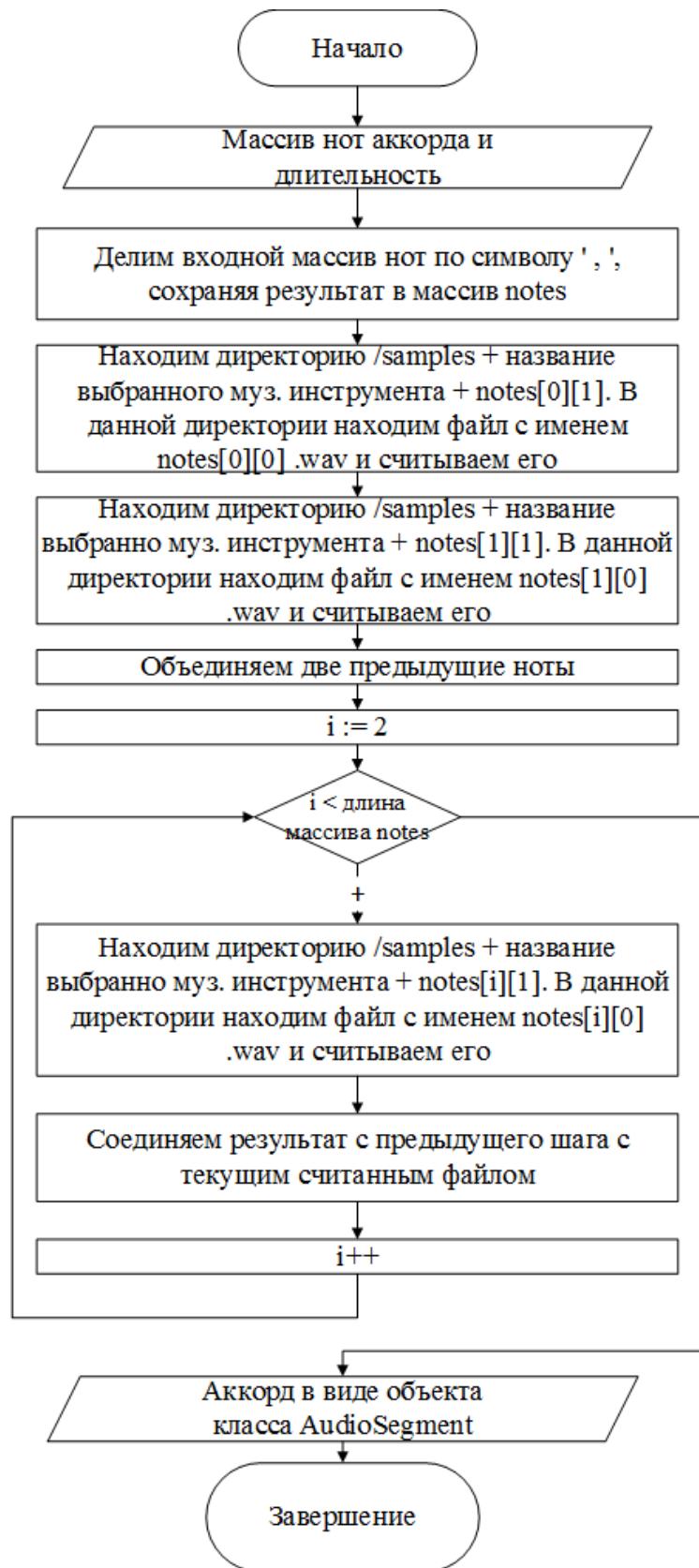


Рисунок 48 – Алгоритм синтеза звуков по структурированному описанию музыкальной композиции. Алгоритм синтеза гармонии – получение не стандартного аккорда

### 3.4 Выводы по третьей главе

В результате разработки алгоритмического обеспечения системы синтезирования музыкальной композиции по цветовому изображению были разработаны алгоритмы, основанные на описанных ранее методах синтеза композиции по изображению. Среди разработанных можно выделить три группы алгоритмов, которые повторяют описанные методы:

- алгоритмы анализа изображений. Общая асимптотическая сложность алгоритмов –  $O(n)$ .
- алгоритмы структурированного представления музыкальной композиции. Общая асимптотическая сложность алгоритмов –  $O(n)$ .
- алгоритм синтеза звуков по структурированному описанию музыкальной композиции. Общая асимптотическая сложность алгоритмов –  $O(n)$ .

## 4 Реализация системы синтезирования музыкальной композиции по цветовому изображению

### 4.1 Архитектура системы

Архитектура системы представлена на рисунке 79 [74, 75].

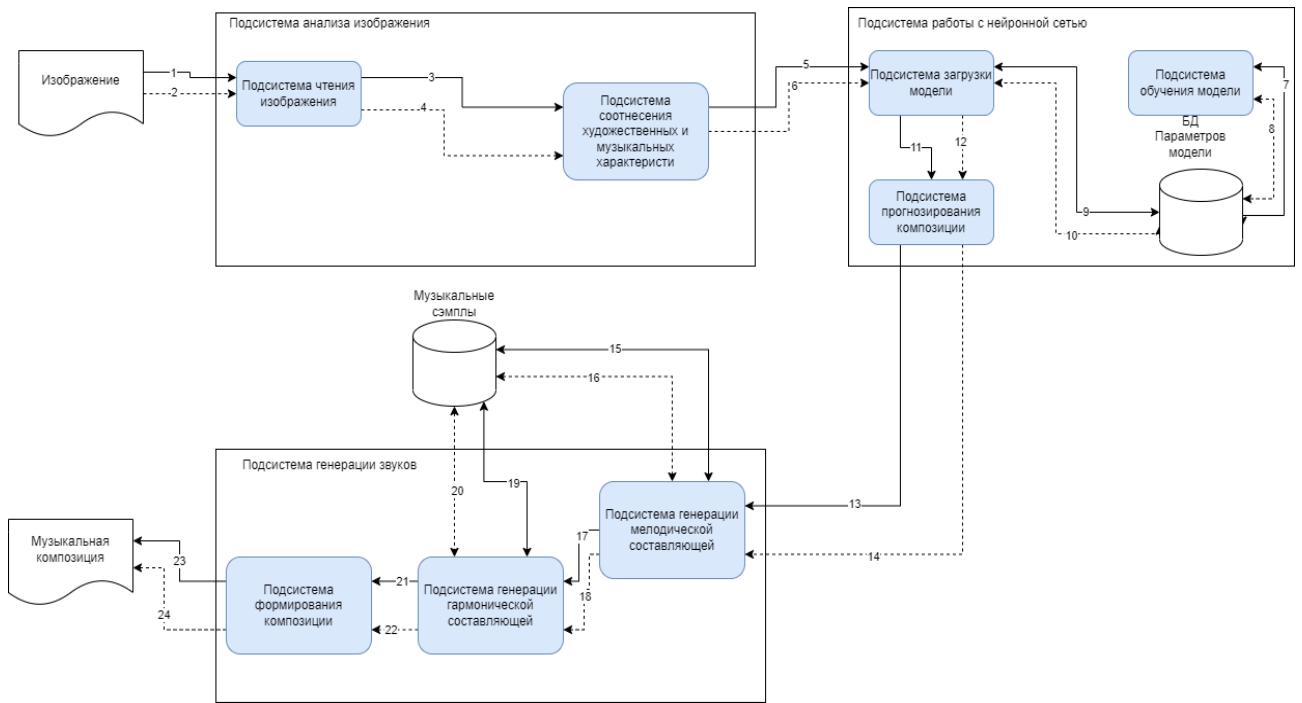


Рисунок 49 – Архитектура системы

- 1 – вызов подсистемы чтения изображения;
- 2 – загруженное пользователем изображение;
- 3 – вызов подсистемы соотнесения художественных и музыкальных характеристик;
- 4 – данные о цветах и их характеристиках;
- 5 – вызов подсистемы выбора модели;
- 6 – данные о тональности и начало мелодической части произведения;
- 7 – вызов подсистемы обучения модели/обращение к базе данных моделей;
- 8 – обученная модель;

- 9 – вызов подсистемы выбора модели/обращение к базе данных моделей;
- 10 – обученная модель;
- 11 – вызов подсистемы прогнозирования композиции;
- 12 – обученная модель;
- 13 – вызов подсистемы генерации мелодической составляющей;
- 14 – xml-текст;
- 15 – обращение к хранилищу с музыкальными сэмплами;
- 16 – данные об инструменте, октаве и названии ноты;
- 17 – вызов подсистемы генерации гармонической составляющей;
- 18 – мелодическая составляющая в виде объекта класса Pydub;
- 19 – обращение к хранилищу с музыкальными сэмплами;
- 20 – данные об инструменте, октаве и названии нот;
- 21 – вызов подсистемы формирования композиции;
- 22 – композиция в виде объекта класса Pydub;
- 23 – обращение к директории сохранения музыкальной композиции;
- 24 – музыкальная композиция в виде файла в формате .mp3.

## 4.2 Интерфейс и функции системы

Интерфейс и функции системы представлены на рисунках 50-54.

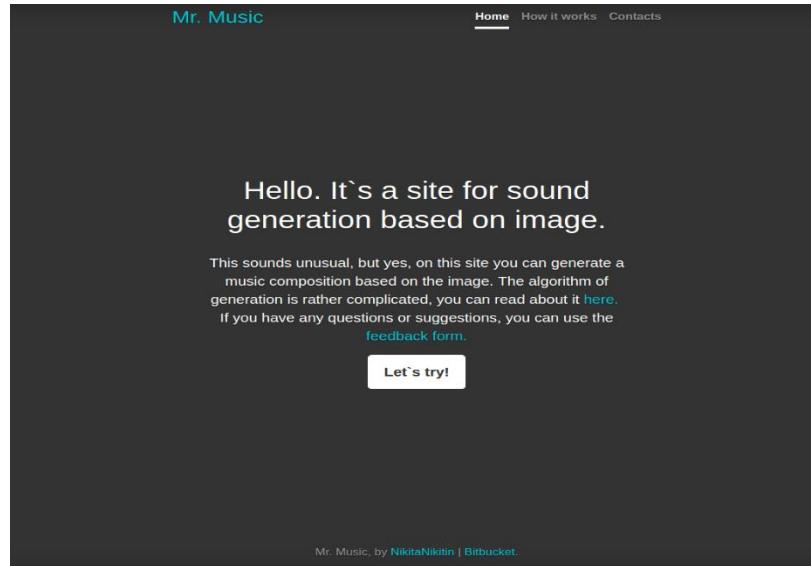


Рисунок 50 – Главная страница веб-сайта

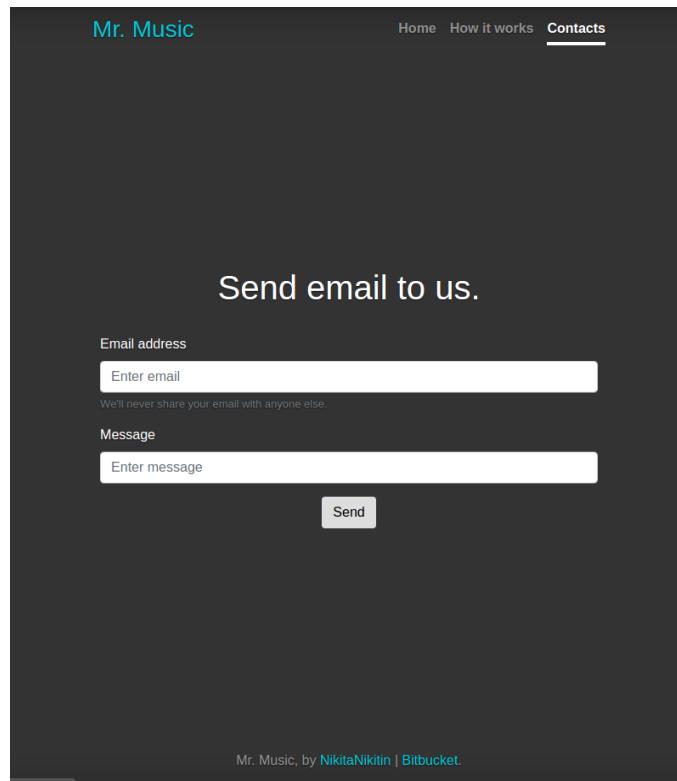


Рисунок 51 – Страница обратной связи

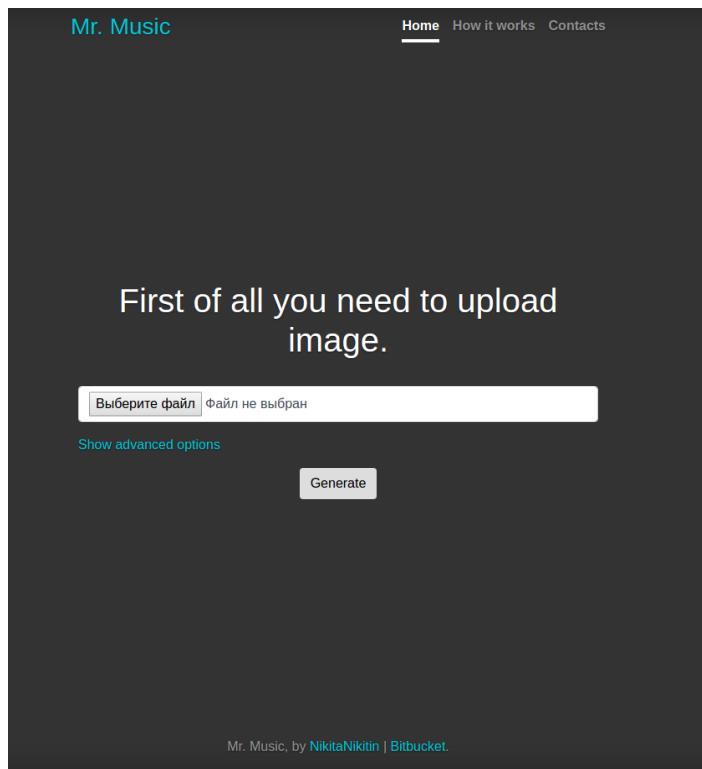


Рисунок 52 – Страница генерации звуков по изображению

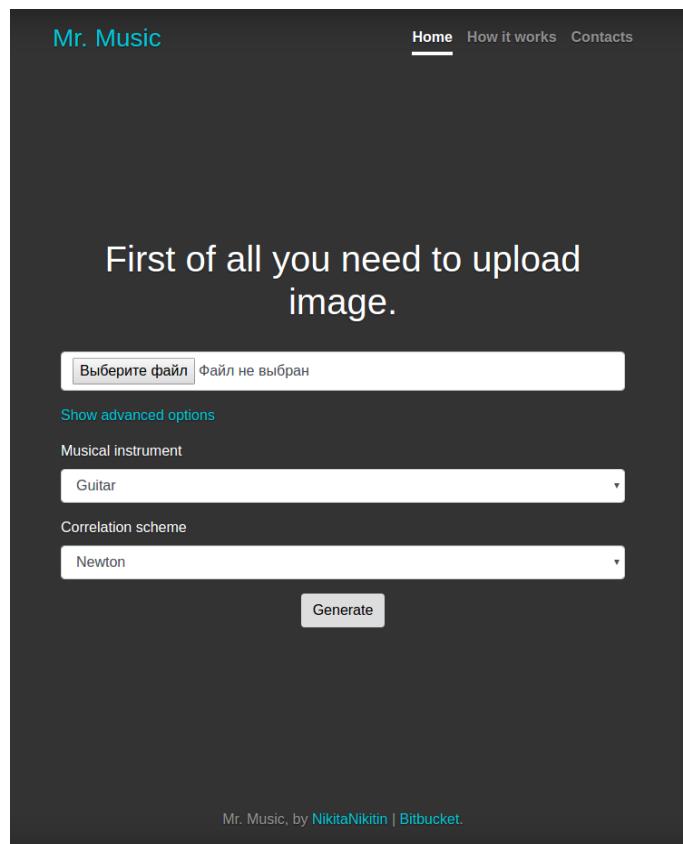


Рисунок 53 – Отображение расширенных параметров на странице генерации звуков по изображению

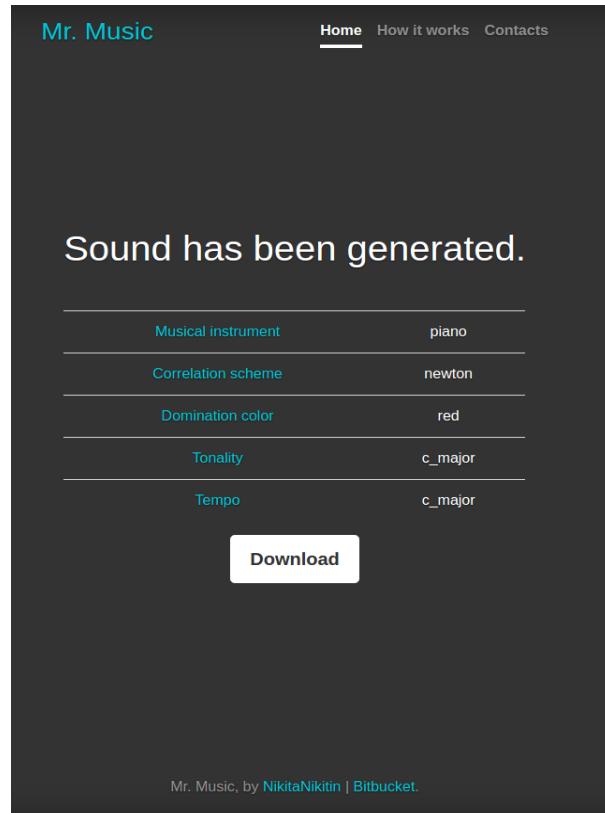


Рисунок 54 – Страница скачивания сгенерированных звуков

#### 4.3 Язык программирования и используемые библиотеки

##### Язык программирования

В качестве языка программирования был выбран язык Python. Важной особенностью данного языка является кроссплатформенность, а также ориентированность на повышение производительности разработчика и читаемости кода [52].

## Описание используемых библиотек

### Библиотека pydub

Библиотека для языка Python, предоставляющая большие возможности для работы с аудио.

Пример использования библиотеки в работе:

Загрузка аудио файла ноты до.

```
from pydub import AudioSegment
AudioSegment.from_file(self.path_to_directory +
"/samples/piano/" + str(octave) + "/c.wav")
```

Соединение два аудио файла в один

```
result = first_part.overlay(second_part)
```

### Библиотека OpenCV

OpenCV – это библиотека компьютерного зрения, с открытым исходным кодом (Open Source). Библиотека написана на С и С ++ и работает под управлением Linux, Windows и Mac OS X. Существует активное развитие интерфейсов для Python, Ruby, Matlab и других языков. Может свободно использоваться в академических и коммерческих целях [76].

### Библиотека BeautifulSoup

BeautifulSoup – это библиотека для языка Python, предоставляющая возможности парсинга HTML/XML текста.

Пример использования библиотеки в работе:

Получение списка всех тегов *chord* в тексте

```
chords = xml_music_notation.doc.harmony.findAll("chord")
```

## Библиотека Keras

Keras – это высокоуровневый API для разработки нейронных сетей, написанный на Python и способный работать поверх TensorFlow, CNTK или Theano. Библиотека была разработана с упором на возможность быстрого экспериментирования. В данной работе используется поверх TensorFlow [11].

## Библиотека Flask

Flask — это микрофреймворк для разработки веб-приложений на языке Python. Использует набор инструментов Werkzeug и шаблонизатор Jinja2.

## Библиотека Python Midi

Python Midi — это библиотека для высокоуровневой работы с midi файлами. Данная библиотека содержит функции чтения и записи файлов, классы для управления midi событиями, карту темпа, которая активно отслеживает изменения темпа в дорожке.

## 4.4 Процесс проектирования в нотации UML

Диаграмма вариантов использования представлена на рисунке 55.

Диаграмма классов представлена на рисунке 56.

Диаграмма последовательности представлена на рисунке 57.

Диаграмма активности представлена на рисунке 58.

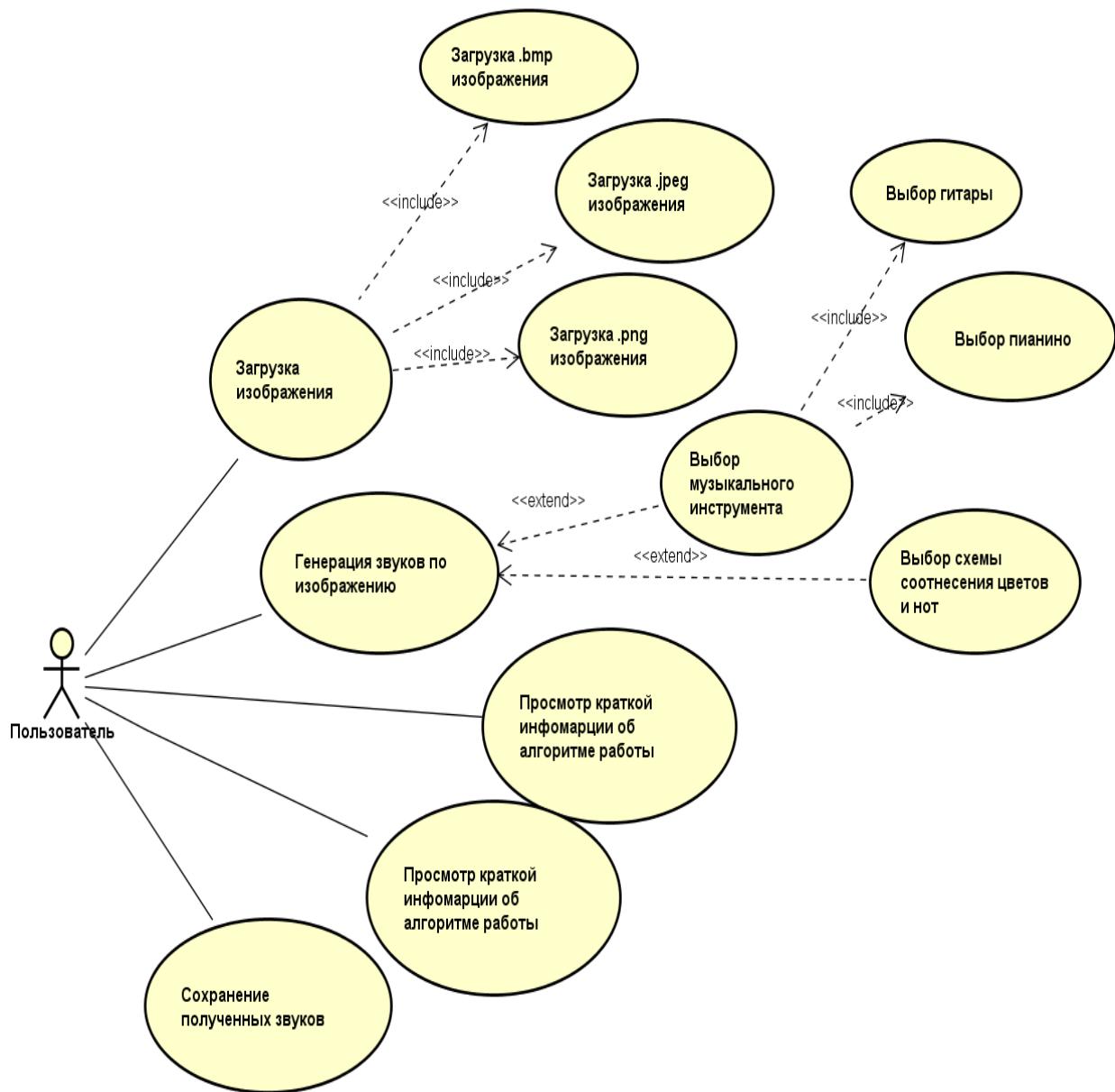


Рисунок 55 – Диаграмма прецедентов



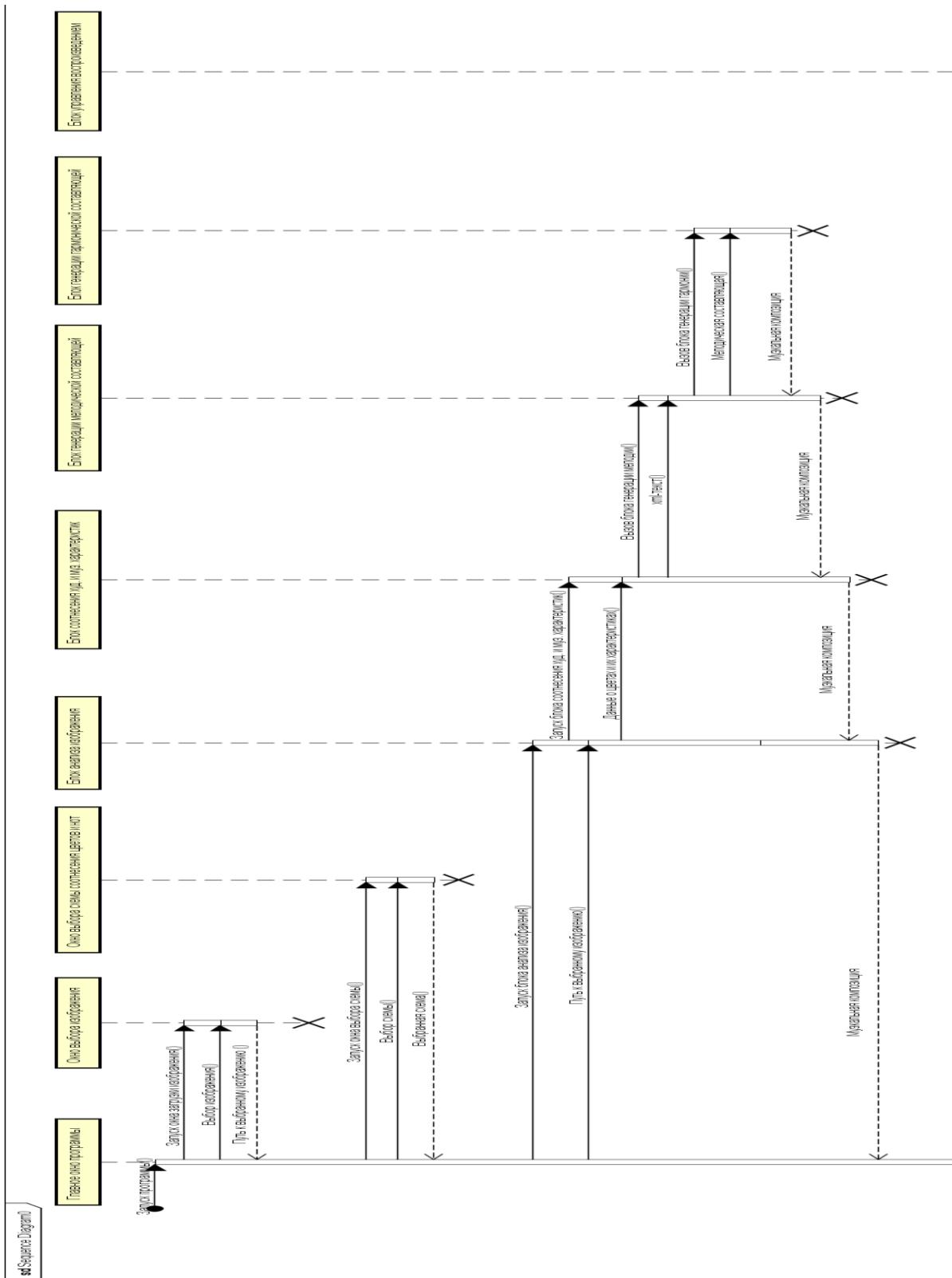


Рисунок 57 – Диаграмма последовательности

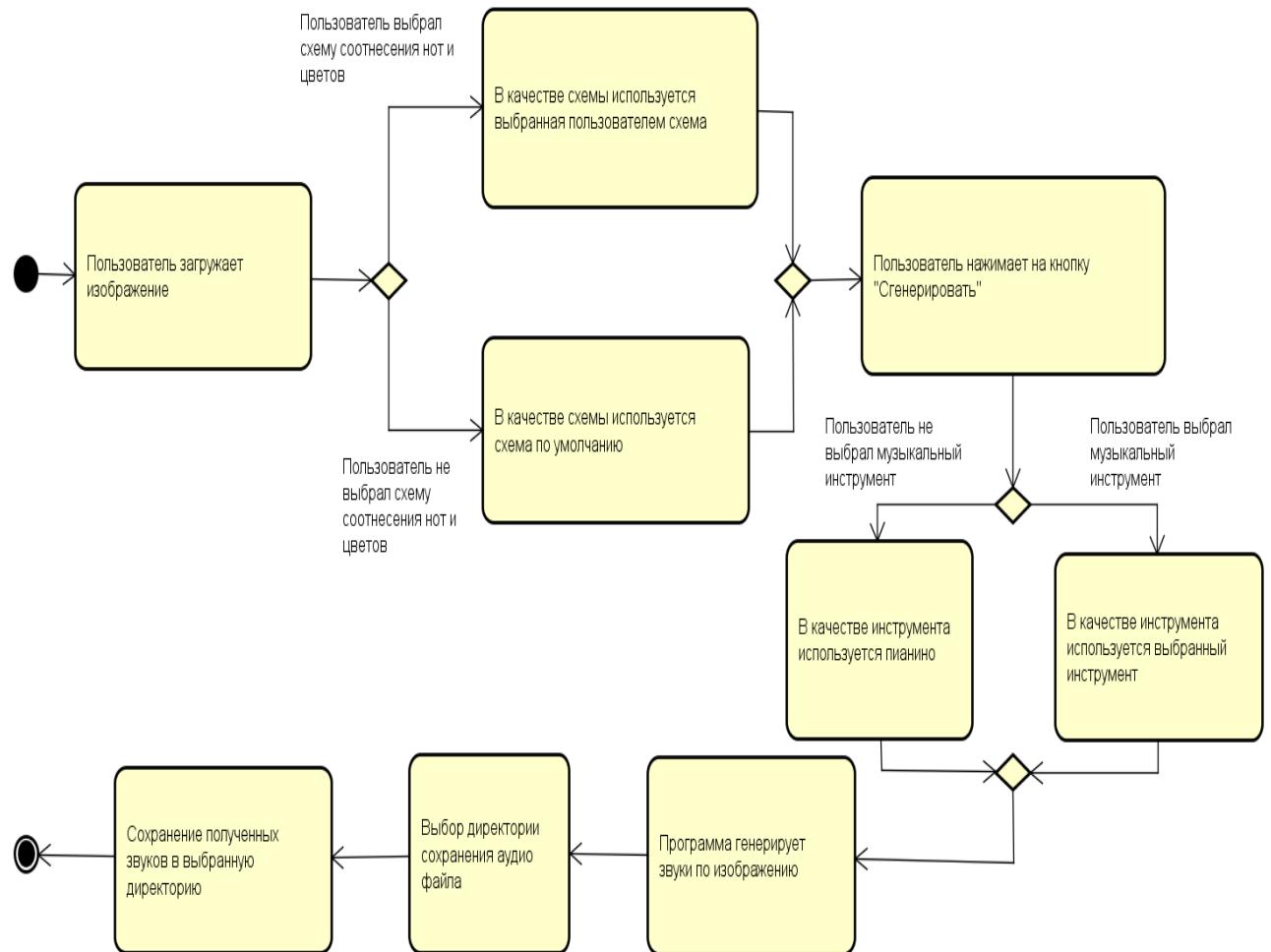


Рисунок 58 – Диаграмма активности

#### 4.5 Выводы по четвёртой главе

В результате проектирования и разработки системы синтезирования музыкальных композиций по цветовому изображению были разработаны алгоритмы, которые детально описывают разработанные ранее методы. Для разработки системы было решено использовать язык программирования Python совместно с библиотекой Keras, поскольку этот язык является кроссплатформенным, а также ориентированным на повышение производительности разработчика и читаемости кода. Кроме того, он позволяет проводить быстрое прототипирование.

## **5 Вычислительные эксперименты по оцениванию музыкальных композиций и компьютерной системы**

### **5.1 Тестирование системы**

Описание тестов и тестовых примеров

Функциональное тестирование

Тест № 1

Название: главная страница.

Шаги: зайти на веб-сайт автоматизированной системы генерации звуков по изображению.

Ожидаемый результат: отображается главная страница системы генерации звуков по изображению.

Тест № 2

Название: страница обратной связи.

Предусловия: открыта главная страница.

Шаги: на главной странице нажать на ссылку «Contacts».

Ожидаемый результат: отображается страница обратной связи, содержащая два поля ввода – Email и Message, кнопку отправки сообщения.

Тест № 3

Название: отправка сообщения на странице обратной связи.

Предусловия: открыта страница обратной связи.

Шаги: заполнить поля ввода и нажать на кнопку «Send»

Ожидаемый результат:

- отправится сообщение с введённым текстом на email:

info.mrmusic@gmail.com;

- отобразится главная страница системы.

Тест № 4

Название: отправка сообщения на странице обратной связи при пустом поле email.

Предусловия: открыта страница обратной связи.

Шаги: заполнить поле ввода «Message» и нажать на кнопку «Send»

Ожидаемый результат: отобразится страница обратной связи с ошибкой: «Field Email is required».

#### Тест № 5

Название: отправка сообщения на странице обратной связи при пустом поле message.

Предусловия: открыта страница обратной связи.

Шаги: заполнить поле ввода «Email» и нажать на кнопку «Send»

Ожидаемый результат: отобразится страница обратной связи с ошибкой: «Field Message is required».

#### Тест № 6

Название: отправка сообщения на странице обратной связи при всех пустых полях.

Предусловия: открыта страница обратной связи.

Шаги: нажать на кнопку «Send»

Ожидаемый результат: отобразится страница обратной связи с ошибкой: «Field Email is required. Filed Message is required».

#### Тест № 7

Название: отправка сообщения на странице обратной связи при неправильном значении поля Email.

Предусловия: открыта страница обратной связи.

Шаги: заполнить поле ввода «Email» не правильным значением (т.е. текстом, без символа @, или указания домена, например myemail) и нажать на кнопку «Send»

Ожидаемый результат: отобразится страница обратной связи с ошибкой: «Field Email is wrong».

## Тест № 8

Название: отображение страницы «How it works».

Предусловия: открыта любая страница веб-сайта.

Шаги: нажать на ссылку «How it works».

Ожидаемый результат: в новой вкладке откроется страница

<https://bitbucket.org/NikitaNiktin/mrmusic-public>

## Тест № 9

Название: отображение страницы генерации звуков.

Предусловия: открыта главная страница веб-сайта.

Шаги: нажать на кнопку «Let's try».

Ожидаемый результат: откроется страница генерации звуков.

## Тест № 10

Название: загрузка изображения на странице генерации звуков.

Предусловия: открыта страница генерации звуков.

Шаги:

- нажать на кнопку «Выбрать файл»;

- в открывшемся диалоговом окне выбрать изображение (файл с расширением .jpeg, .jpg, .png);

Ожидаемый результат: загрузится выбранное изображение (рядом с кнопкой отобразится выбранный файл).

## Тест № 11

Название: генерации звуков только при загруженном изображении.

Предусловия:

- открыта страница генерации звуков;

- загружено изображение (файл с расширением .jpeg, .jpg, .png)

Шаги:

- нажать на кнопку «Generate».

Ожидаемый результат: отобразится страница сгенерированных звуков, содержащая таблицу с параметрами композиции (темп, тональность,

преимущественный цвет), схемой – Ньютона, инструмент – фортепьяно, также страница содержит кнопку «Download».

### Тест № 12

Название: отображение расширенных параметров на странице генерации звуков.

Предусловия:

- открыта страница генерации звуков.

Шаги:

- нажать на ссылку «Show advanced options»;

Ожидаемый результат: на странице отобразятся два поля выбора – «Instrument», «Correlation scheme».

### Тест № 13

Название: генерации звуков с выбором музыкального инструмента.

Предусловия:

- открыта страница генерации звуков;
- загружено изображение (файл с расширением .jpeg, .jpg, .png)

Шаги:

- нажать на ссылку «Show advanced options»;
- выбрать музыкальный инструмент «Guitar»
- нажать на кнопку «Generate».

Ожидаемый результат:

- отобразится страница сгенерированных звуков, содержащая таблицу с параметрами композиции (темп, тональность, преимущественный цвет), схемой – Ньютона, инструмент – гитара, также страница содержит кнопку «Download»;
- сгенерированная композиция будет озвучена инструментов гитара.

### Тест № 14

Название: генерации звуков с выбором музыкального инструмента и схемы соотнесения цветов и нот.

Предусловия:

- открыта страница генерации звуков;
- загружено изображение (файл с расширением .jpeg, .jpg, .png)

Шаги:

- нажать на ссылку «Show advanced options»;
- выбрать музыкальный инструмент «Guitar»;
- выбрать схему соотнесения цветов и нот - Remington
- нажать на кнопку «Generate».

Ожидаемый результат:

- отобразится страница сгенерированных звуков, содержащая таблицу с параметрами композиции (темп, тональность, преимущественный цвет), схемой – Римингтона, инструмент – гитара, также страница содержит кнопку «Download»;
- сгенерированная композиция будет озвучена инструментов гитара и сгенерирована с использованием схемы Римингтона.

### Тест № 15

Название: генерации звуков с выбором конкретного изображения – абстрактное изображение.

Предусловия:

- открыта страница генерации звуков;
- загружено изображение с рисунка 43.

Шаги:

- нажать на ссылку «Show advanced options»;
- выбрать музыкальный инструмент «Piano»;
- выбрать схему соотнесения цветов и нот - Castel
- нажать на кнопку «Generate».

Ожидаемый результат: отобразится страница сгенерированных звуков, содержащая таблицу с параметрами композиции:

- схема: Castel;
- инструмент: Piano;
- преимущественный цвет: red;

- тональность: G major;
- темп: 120.

#### Тест № 16

Название: генерации звуков с выбором конкретного изображения – город.

Предусловия:

- открыта страница генерации звуков;
- загружено изображение с рисунка 50.

Шаги:

- нажать на ссылку «Show advanced options»;
- выбрать музыкальный инструмент «Piano»;
- выбрать схему соотнесения цветов и нот - Remington
- нажать на кнопку «Generate».

Ожидаемый результат: отобразится страница сгенерированных звуков, содержащая таблицу с параметрами композиции:

- схема: Remington;
- инструмент: Piano;
- преимущественный цвет: Violet-blue;
- тональность: G sharp major;
- темп: 85.

#### Тест № 17

Название: генерации звуков с неправильным изображением.

Предусловия:

- открыта страница генерации звуков;

Шаги:

- нажать на кнопку «Выбрать файл»;
- в открывшемся диалоговом окне выбрать файл с расширением, отличающимся от .jpg, .jpeg, .png (например, .txt);
- нажать на кнопку «Generate».

Ожидаемый результат: отобразится страница генерации звуков с сообщением об ошибке «Please, select JPEG or PNG image».

#### Тест № 18

Название: скачивание/прослушивание сгенерированных звуков.

Предусловия:

- сгенерированы звуки по изображению;
- открыта страница сгенерированных звуков с отображением

параметров.

Шаги:

- нажать на кнопку «Download».

Ожидаемый результат: в новой вкладке откроется страница прослушивания/скачивания сгенерированных звуков в .mp3 формате.

Юзабилити-тестирование (оценка качества интерфейса)

#### Тесты № 19-21

Название: тестирование корректности отображения сайта в GoogleChrome/MozillaFirefox/Opera.

Предусловия:

- установлен браузер Google Chrome/Mozilla Firefox/Opera.

Шаги:

- зайти на веб-сайт автоматизированной системы генерации звуков по изображению через браузер GoogleChrome/MozillaFirefox/Opera.

Ожидаемый результат: все страницы сайта отображаются корректно.

#### Тесты № 22-24

Название: тестирование корректности отображения сайта в мобильном браузере на ОС Android.

ОС: Android 6.0

Размер экрана: 5.5 дюймов.

Предусловия:

- установлен браузер Google Chrome/Mozilla Firefox/Opera.

Шаги:

- зайти на веб-сайт автоматизированной системы генерации звуков по изображению через браузер GoogleChrome/MozillaFirefox/Opera.

Ожидаемый результат: все страницы сайта отображаются корректно.

Тесты № 25-27

Название: тестирование корректности отображения сайта в мобильном браузере на ОС Android.

ОС: Android 7.1

Размер экрана: 5.2 дюйма.

Предусловия:

- установлен браузер Google Chrome/Mozilla Firefox/Opera.

Шаги:

- зайти на веб-сайт автоматизированной системы генерации звуков по изображению через браузер GoogleChrome/MozillaFirefox/Opera.

Ожидаемый результат: все страницы сайта отображаются корректно.

Тесты № 28-30

Название: тестирование корректности отображения сайта в мобильном браузере на ОС Android.

ОС: Android 4.2

Размер экрана: 5.0 дюйма.

Предусловия:

- установлен браузер Google Chrome/Mozilla Firefox/Opera.

Шаги:

- зайти на веб-сайт автоматизированной системы генерации звуков по изображению через браузер GoogleChrome/MozillaFirefox/Opera.

Ожидаемый результат: все страницы сайта отображаются корректно.

## Протоколы тестирования

Таблица 7 – Протокол тестирования системы [77-82].

Номер теста	Название операции	Ожидаемый результат	Тест пройден
1	Отображение главной страницы	Отображается главная страница системы генерации звуков по изображению	+
2	Отображение страницы обратной связи	Отображается страница обратной связи, содержащая два поля ввода – Email и Message, кнопку отправки сообщения	+
3	Отправка сообщения на странице обратной связи	Отправится сообщение с введённым текстом на email: info.mrmusic@gmail.com; Отобразится главная страница системы	+
4	Отправка сообщения на странице обратной связи при пустом поле email	Отобразится страница обратной связи с ошибкой: «Field Email is required»	+
5	Отправка сообщения на странице обратной связи при пустом поле message	Отобразится страница обратной связи с ошибкой: «Field Message is required»	+
6	Отправка сообщения на странице обратной связи при всех пустых полях	Отобразится страница обратной связи с ошибкой: «Field Email is required. Filed Message is required»	+
7	Отправка сообщения на странице обратной связи при неправильном значении поля Email	Отобразится страница обратной связи с ошибкой: «Field Email is wrong»	+/-
8	Отображение страницы «How it works»	В новой вкладке откроется страница <a href="https://bitbucket.org/NikitaNiktin/mrmusic-public">https://bitbucket.org/NikitaNiktin/mrmusic-public</a>	+
9	Отображение страницы генерации звуков	Откроется страница генерации звуков	+
10	Загрузка изображения на странице генерации звуков	Загрузится выбранное изображение (рядом с кнопкой отобразится выбранный файл)	+

Продолжение таблицы 7.

11	Генерации звуков только при загруженном изображении	Отобразится страница сгенерированных звуков, содержащая таблицу с параметрами композиции (темп, тональность, преимущественный цвет), схемой – Ньютона, инструмент – фортепьяно, также страница содержит кнопку «Download»	+
12	Отображение расширенных параметров на странице генерации звуков	на странице отобразятся два поля выбора – «Instrument», «Correlation scheme»	+
13	Генерации звуков с выбором музыкального инструмента	Отобразится страница сгенерированных звуков, содержащая таблицу с параметрами композиции (темп, тональность, преимущественный цвет), схемой – Ньютона, инструмент – гитара, также страница содержит кнопку «Download»; сгенерированная композиция будет озвучена инструментов гитара	+
14	Генерации звуков с выбором музыкального инструмента и схемы соотнесения цветов и нот	Отобразится страница сгенерированных звуков, содержащая таблицу с параметрами композиции (темп, тональность, преимущественный цвет), схемой – Римингтона, инструмент – гитара, также страница содержит кнопку «Download»; Сгенерированная композиция будет озвучена инструментов гитара и сгенерирована с использованием схемы Римингтона	+

## Продолжение таблицы 7.

15	Генерации звуков с выбором конкретного изображения абстрактное изображение	-	Отобразится страница сгенерированных звуков, содержащая таблицу с параметрами композиции: схема - Castel; инструмент - Piano; преимущественный цвет - red; тональность – G major; темп - 120.	+     
16	Генерации звуков с выбором конкретного изображения – город		Отобразится страница сгенерированных звуков, содержащая таблицу с параметрами композиции: схема - Remington; инструмент - Piano; преимущественный цвет - Violet-blue; тональность – G sharp major; темп - 85.	+     
17	Генерации звуков с неправильным изображением		Отобразится страница генерации звуков с сообщением об ошибке «Image has wrong format»	+/-     
18	Скачивание/прослушивание сгенерированных звуков		В новой вкладке откроется страница прослушивания/скачивания сгенерированных звуков в .mp3 формате	+     
19	Просмотр сайта на ПК (desktop)	Google Chrome	все страницы сайта отображаются корректно	+     
20		Mozilla Firefox		
21		Opera		
22	Просмотр сайта в мобильном браузере на ОС Android 6.0	Google Chrome	все страницы сайта отображаются корректно	+/-     
23		Mozilla Firefox		
24		Opera		

Продолжение таблицы 7.

25	Просмотр сайта в мобильном браузере на ОС Android 7.1	Google Chrome		+/-
26		Mozilla Firefox		
27		Opera		
28	Просмотр сайта в мобильном браузере на ОС Android 4.2	Google Chrome	все страницы сайта отображаются корректно	+
29		Mozilla Firefox		
30		Opera		

## 5.2 Модельный массив тестовых изображений

Тестовое изображение №1

Изображение, загруженное в систему представлено на рисунке 59.



Рисунок 59 – Тестовое изображение №1

Параметры изображения и сгенерированной композиции представлены в таблице 8.

Таблица 8 - Параметры изображения и сгенерированной композиции для теста №1

Тип изображения	Абстрактный
Размер изображения	300 x 228
Инструмент	Фортепьяно
Схема	Ньютона
Преимущественный цвет	Фиолетово-синий
Тональность	Соль диез минор

### Тестовое изображение №2

Изображение, загруженное в систему представлено на рисунке 60:



Рисунок 60 – Тестовое изображение №2

Параметры изображения и сгенерированной композиции представлены в таблице 9.

Таблица 9 - Параметры изображения и сгенерированной композиции для теста №2

Тип изображения	Абстрактный
Размер изображения	346 x 180
Инструмент	Фортепьяно
Схема	Римингтона
Преимущественный цвет	Синий
Тональность	Ля диез минор

### Тестовое изображение №3

Изображение, загруженное в систему представлено на рисунке 61 [83]:

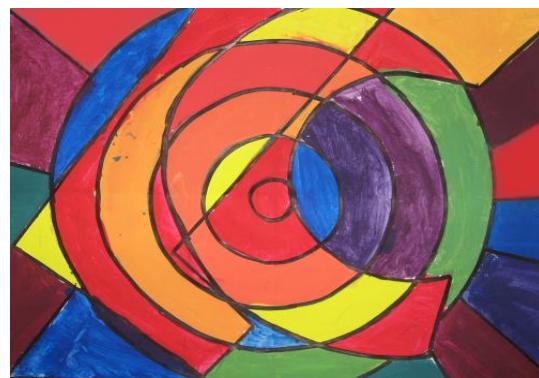


Рисунок 61 – Тестовое изображение №3

Параметры изображения и сгенерированной композиции представлены в таблице 10.

Таблица 10 - Параметры изображения и сгенерированной композиции для теста №3

Тип изображения	Абстрактный
Размер изображения	500 x 383
Инструмент	Фортепьяно
Схема	Кастеля
Преимущественный цвет	Красный
Тональность	Соль мажор

Тестовое изображение №4

Изображение, загруженное в систему представлено на рисунке 62:



Рисунок 62 – Тестовое изображение №4

Параметры изображения и сгенерированной композиции представлены в таблице 11.

Таблица 11 - Параметры изображения и сгенерированной композиции для теста №4

Тип изображения	Абстрактный
Размер изображения	792 x 650
Инструмент	Фортепьяно
Схема	Эпли
Преимущественный цвет	Фиолетовый
Тональность	Ля минор

Тестовое изображение №5

Изображение, загруженное в систему представлено на рисунке 63:



Рисунок 63 – Тестовое изображение №5

Параметры изображения и сгенерированной композиции представлены в таблице 12.

Таблица 12 - Параметры изображения и сгенерированной композиции для теста №5

Тип изображения	Пейзаж
Размер изображения	500 x 281
Инструмент	Фортепьяно
Схема	Бельмонта
Преимущественный цвет	Оранжевый
Тональность	Ре минор

## Тестовое изображение №6

Изображение, загруженное в систему представлено на рисунке 64 [83]:



Рисунок 64 – Тестовое изображение №6

Параметры изображения и сгенерированной композиции представлены в таблице 13.

Таблица 13 - Параметры изображения и сгенерированной композиции для теста №6

Тип изображения	Пейзаж
Размер изображения	500 x 277
Инструмент	Гитара
Схема	Римингтона
Преимущественный цвет	Сине - зелёный
Тональность	Соль минор

## Тестовое изображение №7

Изображение, загруженное в систему представлено на рисунке 65:



Рисунок 65 – Тестовое изображение №7

Параметры изображения и сгенерированной композиции представлены в таблице 14.

Таблица 14 - Параметры изображения и сгенерированной композиции для теста №7

Тип изображения	Пейзаж
Размер изображения	500 x 333
Инструмент	Фортепьяно
Схема	Ньютона
Преимущественный цвет	Зелёный
Тональность	Ля минор

#### Тестовое изображение №8

Изображение, загруженное в систему представлено на рисунке 66:



Рисунок 66 – Тестовое изображение №9

Параметры изображения и сгенерированной композиции представлены в таблице 15.

Таблица 15 - Параметры изображения и сгенерированной композиции для теста №9

Тип изображения	Город
Размер изображения	500 x 259
Инструмент	Фортепьяно
Схема	Ньютона
Преимущественный цвет	Зелёный
Тональность	Фа минор

## Тестовое изображение №9

Изображение, загруженное в систему представлено на рисунке 67:



Рисунок 67 – Тестовое изображение №9

Параметры изображения и сгенерированной композиции представлены в таблице 16.

Таблица 16 - Параметры изображения и сгенерированной композиции для теста №9

Тип изображения	Люди
Размер изображения	500 x 281
Инструмент	Фортепьяно
Схема	Риминтона
Преимущественный цвет	Красный
Тональность	До мажор

## Тестовое изображение №10

Изображение, загруженное в систему представлено на рисунке 68 [83]:



Рисунок 68 – Тестовое изображение №10

Параметры изображения и сгенерированной композиции представлены в таблице 17.

Таблица 17 - Параметры изображения и сгенерированной композиции для теста №10

Тип изображения	Город
Размер изображения	500 x 281
Инструмент	Фортепьяно
Схема	Римингтона
Преимущественный цвет	Фиолетово-синий
Тональность	Соль диез минор

### 5.3 Экспертное оценивание синтезированных музыкальных композиций

#### Описание методики проведения экспериментов

Поскольку автоматизированных способов оценки качества музыкальных произведений не существует, для оценки качества будем использовать экспертов. Для оценки качества синтеза композиций были привлечены 10 экспертов: преподаватели музыкальных школ или музыканты с образованием. Были составлены наборы из изображений и сгенерированных композиций, которые были отправлены экспертам для анализа и оценки по следующим критериям [12, 84]:

- соответствие характеру изображения;
- реалистичность звучания инструмента (фортепьяно или гитара);
- мелодичность композиции;
- качество гармонии (аккомпанемента);
- благозвучность мелодии;
- цельность композиции;
- реалистичность/искусственность композиции.

Во всех экспериментах использовались следующие критерии и шкалы:

- соответствие характеру изображения – 1/полностью не соответствует характеру изображения, 2/скорее не соответствует характеру изображения,

3/более-менее соответствует характеру изображения, 4/скорее соответствует характеру изображения, 5/соответствует характеру изображения;

- реалистичность звучания инструментов – 1/все инструменты звучат искусственно, 2/инструменты скорее звучат искусственно, 3/часть инструментов скорее звучит искусственно, часть реалистично, 4/инструменты скорее звучат реалистично, 5/инструменты звучат реалистично;

- мелодичность композиции – 1/композиция звучит немелодично, ноты выбиваются, гармония не соответствует мелодии, 2/композиция звучит скорее немелодично, в основном ноты выбиваются, гармония не соответствует, 3/композиция звучит неплохо, мелодия построена хорошо, гармония скорее не соответствует мелодии, 4/в целом композиция звучит мелодично, однако некоторые ноты выбиваются из общей структуры или не попадают в такт, 5/композиция звучит так, как будто её написал композитор;

- качество гармонии – 1/гармония полностью не соответствует мелодии, 2/гармония скорее не соответствует мелодии, 3 /гармония частично соответствует мелодии, 4/гармония скорее соответствует мелодии, 5/гармония полностью соответствует мелодии;

- благозвучность мелодии – 1/звукит не связано, просто набор отдельных нот, 2/звукит скорее не связано, некоторые сочетания нот подходят, но в основном просто набор отдельных нот, 3/звукит более-менее связано, некоторые отрывки композиции подходят друг к другу, но много несвязанных участков, 4/звукит скорее благозвучно, прослеживает паттерн, некоторые ноты не попадают в такт или выбиваются из гаммы, 5/звукит полностью благозвучно, ноты сочетаются и попадают в гамму;

- цельность композиции – 1/композиция звучит абсолютно не цельно, 2/композиция звучит скорее не цельно, однако прослеживаются некоторые связанные части, 3/композиция звучит более-менее цельно, частично попадаются связанные части, 4/композиция звучит скорее цельно, однако попадаются несвязанные ноты, 5/композиция звучит полностью цельно;



Тест №8	4	4	4	5	4	4	4
Тест №9	5	4	3	5	3	3	3
Тест №10	5	4	5	5	5	5	5

1 - соответствие характеру изображения (по пяти бальной шкале); 2 -реалистичность звучания инструмента (фортепьяно или гитара); 3 - мелодичность композиции; 4 - качество гармонии (аккомпанемента); 5 - благозвучность мелодии; 6 - цельность композиции; 7 - реалистичность/искусственность композиции.

Эксперимент по анализу зависимости качества генерируемых звуков с использованием нейронных сетей от подхода, без использования нейронных сетей

Входные данные: модель, обученная на классических произведениях.

Выходные данные: зависимости качества генерируемых звуков с использованием нейронных сетей от подхода, без использования нейронных сетей.

Описание эксперимента: также, как в эксперименте №1 для оценки произведений будем использовать экспертов. Для данного эксперимента потребуется сгенерировать 5 произведений по изображениям с использованием нейронных сетей (можно взять из предыдущего эксперимента), а также 5 произведений, сгенерированных по этим же изображениям, но без использования нейронных сетей. Затем данные композиции необходимо отправить экспертам, которые будут сравнивать два произведения. В результате эксперт будет ставить каждой паре одну из трёх оценок [3]:

- -1 – произведение без использования нейронных сетей лучше, чем с их использованием;
- 0 – произведение без использования нейронных сетей похоже на произведение с их использованием;
- 1 – произведение без использования нейронных сетей хуже, чем с их использованием.

В данном эксперименте были сгенерированы 5 музыкальных композиций с использованием искусственных нейронных сетей и 5 без их использования.

Результаты оценки экспертами пар произведений представлены в таблице 19 [3, 72].

Таблица 19 – Результаты проведения эксперимента по анализу зависимости качества генерируемых звуков с использованием нейронных сетей от подхода, без использования нейронных сетей

	1	2	3	4	5	6	7
Тест №1	0	0	1	1	1	1	1
Тест №2	0	0	1	1	1	1	1
Тест №3	0	0	1	1	1	1	1
Тест №4	0	0	1	1	1	1	0
Тест №5	0	0	0	1	1	0	1

1 - соответствие характеру изображения (по пяти бальной шкале); 2 -реалистичность звучания инструмента (фортепиано или гитара); 3 - мелодичность композиции; 4 - качество гармонии (аккомпанемента); 5 - благозвучность мелодии; 6 - цельность композиции; 7 - реалистичность/искусственность композиции.

## 5.4 Анализ результатов тестирования и валидации

### 5.4.1 Анализ результатов тестирования

Для проведения тестирования было составлено полное тестовое покрытие функций системы, включая обработку ошибок. В ходе проведения тестирования все тесты были успешно пройдены системой. Однако, для тестов №7, №17, №25-27 потребовалась доработка системы, для прохождения данных тестов.

В ходе проведения юзабилити тестирования, также было оценено корректное отображение сайта в различных операционных системах, на различных устройствах. Для тестирования были взяты наиболее распространённые браузеры – GoogleChrome, MozillaFirefox и Opera [85, 86]. В данных браузерах, как в desktop версии, так и в мобильных версиях все страницы сайта отображаются корректно и позволяют осуществить весь функционал системы.

### 5.4.2 Анализ результатов валидации

#### Анализ результатов эксперимента №1

Проанализировав оценки всех экспертов и высчитав средние по каждому критерию, можно сделать вывод о том, что фортепьяно на слух экспертов звучит реалистичнее, чем гитара. Также можно сделать вывод о том, что зачастую композиция, сгенерированная по абстрактным изображениям, более приятна на слух, чем генерация по пейзажам, однако тест №10 является городским пейзажем, но при этом получил довольно неплохие экспертные оценки [3].

В целом общее впечатление от сгенерированных звуков у экспертов положительное. Среди минусов некоторые эксперты выделяют однотипность гармонии, иногда рваность и не достаточную реалистичность произведения, и не достаточную реалистичность гитары [3, 72].

Делая вывод по каждому критерию, можно сказать, что все эксперты оценили на высокий бал соответствие произведения характеру изображения (кроме 8 теста), по второму критерию – инструмент фортепьяно звучит довольно реалистично. Мелодичность композиций разделилась пополам, то есть половина композиций эксперты оценили на высший бал, другую половину на 4. Качество гармонии также было оценено экспертами на высший бал (кроме 3 теста). Благозвучность мелодии получил 60% высших балов и 40% четвёрок, что говорит о том, что некоторые произведения (тесты 7, 8, 9) звучат не вполне реалистично. Реалистичность и цельность композиций в среднем оценено на 4, что является естественным результатом для компьютерной генерации звуков [3].

Для окончательного подведения итогов тестирования, можно высчитать средние значения для каждого критерия по всем тестам, результаты такого расчёта представлены в таблице 20 [3].

Таблица 20 – Средние значения критериев по всем тестам для эксперимента №2.

Критерий	Среднее значение для всех тестов
Соответствие характеру изображения	4.9

Реалистичность звучания инструмента	3.9
Мелодичность композиции	4.4
Качество гармонии	4.9
Благозвучность мелодии	4.5
Цельность композиции	4.4
Реалистичность композиции	4.3

Таким образом, были получены все оценки выше 4, кроме реалистичности звучания инструмента, что связано с некоторой искусственностью звучания гитары [3].

### Анализ результатов эксперимента №2

Для исследования результатов эксперимента по анализу зависимости качества генерируемых звуков с использованием нейронных сетей от подхода, без использования нейронных сетей, были посчитаны средние оценки для каждого критерия по всем тестам (таблица 21) [3].

Таблица 21 – Средние оценки для каждого критерия по всем тестам для эксперимента №2

Критерий	Среднее значение для всех тестов
Соответствие характеру изображения	0
Реалистичность звучания инструмента	0
Мелодичность композиции	0.8
Качество гармонии	1
Приятность для восприятия	1
Цельность композиции	0.8
Реалистичность композиции	0.8

Таким образом, из таблицы видно, что критерии «Соответствие характеру изображения» и «Реалистичность звучания инструмента» не изменились, при изменении способа генерации звуков (с использование искусственных нейронных сетей и без). Это связано с тем, что параметр «Соответствие характеру изображения» определяется выбранной тональностью, что не зависит от использования нейронных сетей. А параметр «Реалистичность звучания

инструмента» зависит от модуля синтеза звуков, который тоже не был затронут при изменении подхода генерации композиций [3].

Наиболее важные параметры для оценки достижения цели исследования – «Мелодичность композиции», «Качество гармонии» и «Благозвучность мелодии» были отмечены экспертами, как улучшившиеся. Это связано с тем, что без использования нейронных сетей был использован «наивный» подход к генерации композиций, в то время как нейронные сети, позволяют выявить такие связи, которые не видны человеку, что сказывается на качестве генерируемых музыкальных композиций [3].

Параметры «Цельность композиции» и «Реалистичность композиции» немного улучшились с использованием нейронных сетей. Это связано с тем, что данные критерии зависят от предыдущих трёх критериев, так как они улучшились, значит и на данные критерии положительно повлияло использование нейронных сетей [3].

#### Оценка достижения цели исследования

Для оценки достижения цели исследования были выдвинуты следующие критерии [3]: средняя оценка качества генерируемых звуков по изображению выше 4; средняя оценка для эксперимента №3 больше 0.5.

Для проверки достижения первого критерия рассчитаем среднюю оценку по всем критериям для всех тестов. Для этого необходимо сложить все средние оценки из таблицы 21 и поделить на общее количество критериев. Таким образом, получим среднюю оценку 4.5. Так как 4.5 больше 4, значит цель согласно первому критерию достигнута [3].

Для проверки достижения второго критерия необходимо рассчитать среднюю оценку по всем критериям в таблице 21. Средняя оценка – 0.6, что больше 0.5, а значит цель достигнута и по второму критерию [3].

## 5.5 Выводы по пятой главе

В ходе проведения валидации моделей было проведено два эксперимента: эксперимент по оценке качества генерируемых звуков и эксперимент по анализу зависимости качества генерируемых звуков с использованием нейронных сетей от подхода, без использования нейронных сетей.

По результатам первого эксперимента можно сделать вывод о том, что в целом общее впечатление от сгенерированных звуков у экспертов положительное. Среди минусов некоторые эксперты выделяют однотипность гармонии, иногда рваность и не достаточную реалистичность произведения, а также не достаточную реалистичность гитары.

В ходе проведения второго эксперимента, можно сделать вывод о том, что использование нейронных сетей оправдано для задачи генерации музыкальных композиций по цветовой гамме изображений, поскольку для наиболее важных критериев, отвечающих за гармоничность и мелодичность звуков было отмечено улучшение качества выходной музыкальной композиции.

Для оценки достижения цели исследования были выдвинуты следующие критерии: средняя оценка качества генерируемых звуков по изображению выше 4; средняя оценка для эксперимента №3 больше 0.5 (то есть в процессе округления даст 1). В результаты оценки было выявлено, что по всем критериям разработанная система удовлетворяет цели исследования.

## Заключение

В результате выполнения данной работы были решены следующие поставленные задачи:

1. Проведен анализ современных подходов и методов автоматизированного синтеза звуковых последовательностей по цветовому изображению.
2. Предложены критерии оценки и сравнения компьютерных средств генерирования музыкальных композиций по цветовому изображению, алгоритмических методов построения звуковых последовательностей, методов воспроизведения звучания музыкальных инструментов.
3. Разработаны метод структурированного представления музыкальной композиции и модель соотнесения цветовых и музыкальных характеристик для получения параметров музыкальной композиции.
4. С использованием машинного обучения построена модельная база композиций с различными музыкальными жанрами для выбора жанра музыкальной композиции.
5. Разработан метод воспроизведения звуков музыкальных инструментов по структурированному представлению музыкальной композиции.
6. Разработан комплексный метод синтезирования цельной музыкальной композиции по цветовому изображению с использованием нейронных сетей.
7. Разработанные модели, методы и алгоритмы реализованы в виде компьютерной системы, автоматизирующей процесс создания музыкальной композиции.
8. Проведены тестирование и экспертное оценивание мелодичности и гармоничности синтезированных музыкальных композиций.
8. Проведено исследование эффективности разработанного алгоритмического и программного обеспечения на предмет мелодичности и гармоничности.

В процессе обзора аналогов создаваемой системы было выявлено, что наибольшую конкуренцию создаваемой системе может составить Music in image, однако она не в достаточной мере учитывает теоретические правила создания музыкальных произведений, также изображение анализируется не как целое, что приводит к потере мелодичности и цельности композиции.

В ходе анализа и сравнения алгоритмических методов построения звуковых последовательностей было выявлено, что для генерации музыкального материала в данной работе будет использоваться метод, основанный на искусственных нейронных сетях. Именно нейронные сети позволяют выявить наиболее сложные взаимосвязи, характерные музыкальным произведениям. Также большинство современных программ для автоматизированной генерации музыкальных композиций используют именно нейронные сети, например система Magenta от Google.

В процессе исследования методов синтеза звука, были рассмотрены восемь методов. Каждый из них обладает своими плюсами и минусами, однако для реализации системы был выбран Сэмплинг. Данный метод даёт наиболее реалистичное звучание инструментов, что является важной характеристикой для системы, также данный метод относительно прост в реализации. Недостатком Сэмплинга является ограниченность метода, однако в рамках реализации системы – это не существенно, так как пользователь будет выбирать из ограниченного набора заранее известных инструментов, таким образом не требуется больших возможностей изменения готовых пресетов [1, 40].

В ходе проведения валидации моделей было проведено два эксперимента:

- эксперимент по оценке качества генерируемых звуков;
- эксперимент по анализу зависимости качества генерируемых звуков с использованием нейронных сетей от подхода, без использования нейронных сетей.

По результатам первого эксперимента можно сделать вывод о том, что в целом общее впечатление от сгенерированных звуков у экспертов положительное.

Среди минусов - некоторые эксперты выделяют однотипность гармонии, иногда рваность и не достаточную реалистичность произведения, а также не достаточную реалистичность гитары.

В ходе проведения второго эксперимента, можно сделать вывод о том, что использование нейронных сетей оправдано для задачи генерации музыкальных композиций по цветовой гамме изображений, поскольку для наиболее важных критериев, отвечающих за гармоничность и мелодичность звуков было отмечено улучшение качества выходной музыкальной композиции [87].

Для оценки достижения цели исследования были выдвинуты следующие критерии: средняя оценка качества генерируемых звуков по изображению выше 4; средняя оценка для эксперимента №3 больше 0.5 (то есть в процессе округления даст 1). В результаты оценки было выявлено, что по всем критериям разработанная система удовлетворяет цели исследования.

По результатам работы создан программный продукт, который получил Свидетельство о государственной регистрации (приложении 3).

**Перспективы дальнейшей разработки темы.** Сейчас модуль работы с нейронной сетью просто генерирует целую композицию, при этом полученные ноты не всегда могут находиться в соответствии и могут получаться диссонирующие звуки. В связи с этим, одно из дальнейших развитий разработанных моделей и методов может быть направлено на улучшение способа генерации музыкальной информации: генерировать некие шаблоны в зависимости от гаммы, а в композиции эти шаблоны повторять. Кроме того, нужно разделять композицию на смысловые части (например, куплет и припев) и генерировать гармоническую часть с использованием методов машинного обучения.

## Литература

1. Генерация звуков по цветовой гамме изображения / Н.А. Никитин, В.Л. Розалиев, Ю.А. Орлова, А.В. Заболеева-Зотова // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2016) : матер. VI междунар. науч.-техн. конф. (г. Минск, 18-20 февр. 2016 г.) / редкол.: В.В. Голенков (отв. ред.) [и др.] ; УО «Белорусский гос. ун-т информатики и радиоэлектроники», ГУ «Администрация Парка высоких технологий». – Минск, 2016. – С. 561-564.
2. Розалиев, В.Л. Methods and Models for Identifying Human Emotions by Recognition Gestures and Motion / Розалиев В.Л., Заболеева-Зотова А.В. // The 2013 2nd International Symposium on Computer, Communication, Control and Automation 3CA 2013, December 1-2, 2013, Singapore : Papers. – [Amsterdam – Beijing – Paris] : Atlantis Press, 2013. – P. 67-71.
3. Применение методов машинного обучения для задачи генерации музыкальных композиций / Н.А. Никитин, В.Л. Розалиев, Ю.А. Орлова, А.В. Заболеева-Зотова // Прикаспийский журнал: управление и высокие технологии. - 2018. - № 2 (42). - С. 84-95.
4. D. Cope, Computer Models of Musical Creativity, MIT Press, Cambridge, Mass., 2005.
5. Xiaoying Wu. A study on image-based music generation. Master's thesis. Simon Fraser University, Burnaby, 2008.
6. Чернышев, Д. Цвета и ноты [Электронный ресурс]. – М., 2014. - Режим доступа: <http://mi3ch.livejournal.com/2506477.html>
7. Автоматическая генерация звуков по цветовой гамме изображений / Н. А. Никитин, В. Л. Розалиев, Ю. А. Орлова, А. В. Заболеева-Зотова // Известия ВолГГТУ. – 2016. – № 3. – С. 58-62.
8. Liang, F. BachBot: Automatic composition in the style of Bach chorales: master's thesis / F. Liang. – Cambridge. – 93 p. – 2016

9. Jaques, N., Gu, S., Turner, R., Eck, D. Tuning recurrent neural networks with reinforcement learning / N. Jaques, S. Gu, R. Turner, D. Eck // arXiv preprint arXiv:1611.02796v3. – 2017.
10. Roche, F. Music sound synthesis using machine learning: Towards a perceptually relevant control space: doctoral thesis / F. Roche. – Grenoble, French. – 2020
11. Никитин, Н.А. Программная генерация звуков с использованием рекуррентной нейронной сети / Н.А. Никитин, В.Л. Розалиев, Ю.А. Орлова // Инновационные, информационные и коммуникационные технологии (ИНФО–2017) : сб. тр. XIV междунар. науч.-практ. конф. (г. Сочи, 1-10 октября 2017 г.) / под ред. С.У. Увайсова ; Московский технологический ун-т (МИРЭА), Ассоциация выпускников и сотрудников ВВИА им. проф. Жуковского, РФФИ (Проект № 17-07-20539). - Москва, 2017. - С. 427-432.
12. Никитин, Н.А. Разработка программы автоматической генерации музыкальных композиций по изображению / Н.А. Никитин // XXIII Региональная конференция молодых исследователей Волгоградской области (г. Волгоград, 11-14 декабря 2018 г.) : тез. докл. / редкол.: А. В. Навроцкий (отв. ред.) [и др.] ; Комитет образования, науки и молодёжной политики Волгоградской обл., Совет ректоров вузов Волгоградской обл., Волгоградский гос. техн. ун-т. - Волгоград, 2019. - С. 157-159.
13. Algorithmic music production from digital images [Электронный ресурс]. – 2015. – Режим доступа: <http://www.musicinimages.com/>
14. Upic [Электронный ресурс]. – 2014. – Режим доступа: <http://www.musicainformatica.org/topics/upic.php>
15. MetaSynth. Electronic music studio and sound designer's dream tool for Mac OS X [Электронный ресурс]. – 2015. – Режим доступа: <http://www.uisoftware.com/MetaSynth/index.php>
16. Fels, S. and Mase, K. Iamascope: A Graphical Musical Instrument / S. Fels, K. Mase // Computers and Graphics. – 1999. – Volume 23. – Issue 2. – pp. 277-286

17. Никитин, Н.А. Программная генерация звуков по цветовой гамме изображений с использованием рекуррентной нейронной сети / Н.А. Никитин, В.Л. Розалиев, Ю.А. Орлова // Информационные технологии в науке, управлении, социальной сфере и медицине : сб. науч. тр. IV междунар. конф. (5-8 декабря 2017 г.) / под ред. О.Г. Берестневой, А.А. Мицеля, Т.А. Гладковой ; ФГБОУ ВО «Национальный исследовательский Томский политехнический ун-т», ФГБОУ ВО «Томский гос. ун-т систем управления и радиоэлектроники». - Томск, 2017. - С. 22-26.
18. Никитин, Н.А. Обзор математических методов для генерации музыкальных композиций / Н.А. Никитин, В.Л. Розалиев, Ю.А. Орлова // Молодой учёный. - 2020. - № 52 (342), ч. 1. - С. 36-39.
19. Doornbusch, P. Gerhard Nierhaus: Algorithmic Composition: Paradigms of Automated Music Generation / P. Doornbusch // Computer Music Journal. - Volume: 34, Issue: 3. – 2014.
20. Briot, J. From Artificial Neural Networks to Deep Learning for Music Generation -- History, Concepts and Trends / J. Briot // arXiv preprint arXiv: 2004.03586. – 2020.
21. Frankel-Goldwater, Lee. 'Computers Composing Music: An Artistic Utilization of Hidden Markov Models for Music Composition.' jur: Journal of Undergraduate Research, v. 5, no. 1-2 (2012), pp. 17-20.
22. Donya Quick and Paul Hudak. Grammar-based automated music composition in Haskell. In Proceedings of the first ACM SIGPLAN workshop on Functional art, music, modeling & design, FARM '13, pages 59–70. ACM, 2013.
23. Barate, A., Haus, G., Ludovico L. Formalisms and Interfaces to Manipulate Music Information: The Case of Music Petri Nets / A. Barate, G. Haus, L. Ludovico // In Proceedings of the 2nd International Conference on Computer-Human Interaction Research and Applications (CHIRA 2018). – 2018. – P. 81-90
24. Gengxian, C. Chaos and Fractal in Chinese Traditional Music Texts and Its Inheritance / C. Gengxian // In Proceedings of the 2021 Summit of the International Society for the Study of Information. – Vol. 81. – No 1. – 2022

25. Gerhard Nierhaus, Algorithmic Composition: Paradigms of Automated Music Generation, Springer, 2009, pp.131–155 (Chapter 6) Springer, 2009. Перевод с английского под редакцией А.Ю.Лоскутова. – Нелинейная динамика. 2012. Т. 7. № 1. С. 153–175.
26. Hendrik Vincent Koops, José Pedro Magalhães, and W. Bas de Haas. A functional approach to automatic melody harmonisation. In Proceedings of the First ACM SIGPLAN Workshop on Functional Art, Music, Modeling & Design, FARM ’13, pages 47–58. ACM, 2013.
27. Sopov, E. A. Self-configuring multi-strategy genetic algorithm for non-stationary environments // Вестник СибГАУ. 2015. №1 С.124-130.
28. Majumder, S., Smith, B. Music Recombination using a Genetic Algorithm / S. Majumder; B. Smith // Proc. of the International Computer Music Conference (ICMC). International Computer Music Association, Michigan. –2018
29. Tavares, T., Pessanha, T., Nishihara, G., Avila, G. Alloy sounds: non-repeating sound textures with probabilistic cellular automata / T. Tavares, T. Pessanha, G. Nishihara, G. Avila, G. // Proceedings of the 24rd International Conference on Digital Audio Effects (DAFx-20in21), Vienna, Austria, September 8–10. – 2021. – P. 245-252
30. Abburu, S. Knowledge based Semantic Annotation Generation of Music S. Abburu // International Journal of Computer Applications (0975–888). – Volume 47, No.8, June 2012. – pp. 8-12
31. Kiefer, C. Sample-level sound synthesis with recurrent neural networks and conceptors / C. Kiefer // PeerJ Comput. Sci. – 2019
32. Завгородний, В. Основные типы синтеза звука [Электронный ресурс]. - 2013. - Режим доступа: <http://libdocs.ru/docs/165300/index-2041.html> (Дата обращения: 26.03.2017).
33. Moreno, M. Analysis and resynthesis of real instruments using Pure Data and the MKMR library / M. Moreno // Revista Vórtex. – 2021. – Vol. 9. – No. 2
34. Nielsen, K. Practical linear and exponential frequency modulation for digital music synthesis / K. Nielsen // Proceedings of the 23rd International Conference

on Digital Audio Effects (DAFx-20), Vienna, Austria, September 8–12. – 2022. – P. 132-139

35. Bitton, A., Esling, P., Harada, T. Neural Granular Sound Synthesis / A. Bitton, P. Esling, T. Harada // arXiv preprint arXiv:2008.01393v3. – 2021.
36. Shan, S., Hantrakul, L., Chen, J., Avent, M., Trevelyan, D. Differentiable Wavetable Synthesis / S. Shan, L. Hantrakul, J. Chen, M. Avent, D. Trevelyan // arXiv preprint arXiv: 2111.10003v4. – 2022.
37. Bitton, A., Esling, P., Harada T. Timbre Space Representation of a Subtractive Synthesizer / A. Bitton, P. Esling, T. Harada // arXiv preprint arXiv: 2007.06349. – 2020.
38. Vector Phase Shaping Synthesis / Kleimola, J. and Lazzarini, V and Timoney, J and Valimaki, V // In: Proc. of the 14th Int. Conference on Digital Audio Effects (DAFx-11). – September 19-23, Paris, France. – 2013.
39. Roche, F., Hueber, T., Limier, S., Girin,. L. Autoencoders for music sound synthesis: a comparison of linear, shallow, deep and variational models / F. Roche, T. Hueber, S. Limier, L. Girin // arXiv preprint arXiv: 1806.04096v1. – 2018.
40. Автоматическая генерация звуков по цветовой гамме изображений / А.В. Заболеева-Зотова, Н.А. Никитин, Ю.А. Орлова, В.Л. Розалиев // Пятнадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2016 (г. Смоленск, 3-7 октября 2016 г.) : тр. конф. В 3 т. Т 2. / Российская ассоциация искусственного интеллекта, Федеральный исследовательский центр «Информатика и управление» РАН. - Смоленск, 2016. - С. 174-182.
41. Schmidhuber, J. Deep learning in neural networks: An overview / J. Schmidhuber. – The Swiss AI Lab IDSIA, 2014. – 250 p.
42. Priyadarshini, R. Functional Analysis of Artificial Neural Network for Dataset Classification / R. Priyadarshini // Special Issue of IJCCT. – Volume: 1, Issue 2 for International Conference [ACCTA-2010]. – 2010.

43. Tang, Yi., Salakhutdinov, R. Learning Stochastic Feedforward Neural Networks / Yi. Tang, R. Salakhutdinov // Advances in Neural Information Processing Systems 26 (NIPS). – 2013
44. Graves, A., Jaitly, N., Mohamed, A. Hybrid speech recognition with deep bidirectional LSTM / A. Graves, N. Jaitly, A. Mohamed // Automatic Speech Recognition and Understanding (ASRU). – IEEE Workshop on. IEEE. – pp. 273–278. – 2013
45. Saravanan, K., Sasithra, S. Review on classification based on Artificial Neural Networks / K. Saravanan, S. Sasithra // International Journal of Ambient Systems and Applications (IJASA). – Volume 2, No. 4. – December 2014
46. Никитин, Н.А. Program for sound generation based on image color spectrum with using the recurrent neural network [Электронный ресурс] / Н.А. Никитин, В.Л. Розалиев, Ю.А. Орлова // Proceedings of the IV International research conference «Information technologies in Science, Management, Social sphere and Medicine» (ITSMSSM 2017) / ed. by O.G. Berestneva [et al.]. – [Published by Atlantis Press], 2017. – P. 227-232. – (Ser. Advances in Computer Science Research (ACSR) ; Vol. 72). – URL : <https://www.atlantis-press.com/proceedings/itsmssm-17>.
47. Sak, H., Senior, A., Beaufays, F. Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition / H. Sak, A. Senior, F. Beaufays // ArXiv e-prints. – 2014
48. Brinkkemper, F. Analyzing Six Deep Learning Tools for Music Generation [Электронный ресурс]. – 2015. - Режим доступа: <http://www.asimovinstitute.org/analyzing-deep-learning-tools-music/> (Дата обращения: 03.07.2017).
49. Никитин, Н.А. Разработка метода автоматической генерации звуков по изображению / Н.А. Никитин, В.Л. Розалиев, Ю.А. Орлова // Системы проектирования, технологической подготовки производства и управления этапами жизненного цикла промышленного продукта (CAD/CAM/PDM – 2018) : тр. XVIII междунар. молодёжной конф. (г. Москва, 16-18 октября 2018 г.) / под

ред. А. В. Толока ; Ин-т проблем управления им. В. А. Трапезникова Российской академии наук. - Москва, 2018. - С. 108-112.

50. Automated sound generation based on image colour spectrum with using the recurrent neural network [Электронный ресурс] / Н.А. Никитин, В.Л. Розалиев, Ю.А. Орлова, А.В. Алексеев // CEUR Workshop Proceedings. Vol. 2212 : DS-ITNT 2018. Proceedings of the International Conference «Information Technology and Nanotechnology 2018». Session Data Science (Samara, Russia, 24-27 April, 2018) / ed. by Vladimir Fursov [et al.]. – [Publisher : CEUR-WS.org], 2018. – P. 399-408. – URL : <http://ceur-ws.org/Vol-2212/paper53.pdf>.

51. Caivano, J. L., Colour and sound: Physical and Psychophysical Relations, Colour Research and Application, 12(2), pp. 126-132, 1994

52. Никитин, Н.А. Разработка методов для синтеза музыкальных композиций на основе интуитивного и эмоционального подходов / Н.А. Никитин, В.Л. Розалиев, Ю.А. Орлова // Программная инженерия: современные тенденции развития и применения (ПИ–2020) : сб. материалов IV всерос. науч.-практ. конф., посвящённой 30-летию создания кафедры программной инженерии (12-13 марта 2020 г.) / отв. ред.: Р. А. Томакова ; ФГБОУ ВО «Юго-Западный гос. ун-т». - Курск, 2020. - С. 54-61.

53. Афанасьев, В. В. Светозвуковой музыкальный строй. Элементарная теория аудиовизуальных стимулов / В. В. Афанасьев. – М. : Музыка, 2002. – 70 с.

54. Никитин, Н.А. Программа генерации звуков по цветовой гамме изображений / Н.А. Никитин // Смотр-конкурс научных, конструкторских и технологических работ студентов Волгоградского государственного технического университета (г. Волгоград, 10-13 мая 2016 г.) : тез. докл. / редкол.: А.В. Навроцкий (отв. ред.) [и др.] ; ВолгГТУ, Совет СНТО. - Волгоград, 2016. - С. 152-153.

55. Automation of Musical Compositions Synthesis Process Based on Neural Networks / Н.А. Никитин, В.Л. Розалиев, Ю.А. Орлова, А.В. Заболеева-Зотова // Proceedings of the Fourth International Scientific Conference «Intelligent Information Technologies for Industry» (IITI'19) (Ostrava–Prague (Czech Republic), December 2-7,

2019) / ed. by S. Kovalev [et al.] ; Rostov State Transport University, VŠB – Technical University of Ostrava (Czech Republic), Russian Association for Artificial Intelligence (RAAI) . – Cham (Switzerland) : Springer Nature Switzerland AG, 2020. – P. 51-59. – URL : <https://link.springer.com/chapter/10.1007/978-3-030-50097-9>. – (Book Ser.: Advances in Intelligent Systems and Computing (AISC) ; vol. 1156).

56. Mazurowski, L. Computer models for algorithmic music composition / L. Mazurowski // Proceedings of the Federated Conference on Computer Science and Information Systems. – Szczecin, Poland, 2012. – pp. 733–737

57. Mikolov, T. Recurrent neural network based language model / T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, S. Khudanpur // Proceedings of INTERSPEECH International Speech Communication Association. - vol. 2010, No. 9. – pp. 1045–1048. – 2010

58. Transactions on Engineering Technologies: Special Issue of the World Congress on Engineering and Computer Science / Haeng Kon Kim, Sio-Iong Ao, A. Mahyar. – Springer Publishing Company, New York, 2013. – pp. 796

59. Fernández, J. D., Vico, F. AI Methods in Algorithmic Composition: A Comprehensive Survey / J. D. Fernández, F. Vico // Journal of Artificial Intelligence Research. – № 48. – Málaga, Spain, 2013. – pp. 513-582

60. Proposal of method for generating musical compositions of different genres / Н.А. Никитин, Ю.А. Орлова, В.Л. Розалиев, Аг.С. Кузнецова, В.В. Гилка // Journal of Physics: Conference Series. Vol. 2060 : International Scientific Conference on Artificial Intelligence and Digital Technologies in Technical Systems II-2021 (AIDTTS II-2021) (Volgograd, Russia, 6-7 May 2021) / ed. by K. E. Tokarev. – IOP Publishing, 2021. – 6 p. – DOI: 10.1088/1742-6596/2060/1/012023. – URL: <https://iopscience.iop.org/issue/1742-6596/2060/1>.

61. Raffel, C. Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching / C. Raffel // Doctoral thesis, Columbia University. – 2016

62. Bertin-Mahieux, Th., Ellis, D., Whitman, B. and Lamere, P. The Million Song Dataset / Th. Bertin-Mahieux, D. Ellis, B. Whitman and P. Lamere // Proceedings

of the 12th International Conference on Music Information Retrieval (ISMIR). – 2011. – pp. 591-596

63. Musical instrument recognition in polyphonic audio signals by using a source-filter model / В.Л. Розалиев, Дау Хань Нгуен, Н.А. Никитин, Ю.А. Орлова // Interactive systems: problems of human-computer interaction (Интерактивные системы: проблемы человека-компьютерного взаимодействия) : collection of scientific papers of the 12th international conference (Ulyanovsk, Russia, 25-27 September, 2017) : (сб. науч. тр.) / ed. by P. Sosnin ; Ulyanovsk State Technical University [et al.]. - Ulyanovsk, 2017. - C. 97-104.

64. Szeliski R. Computer Vision: Algorithms and Applications / R. Szeliski. – Springer, 2010. – 979 p.

65. Automated Sound Generation by Image Color Spectrum with Harmony Creation Based on User Ratings [Электронный ресурс] / Н.А. Никитин, В.Л. Розалиев, Ю.А. Орлова, А.В. Заболеева-Зотова // Proceedings of the Second International Scientific Conference «Intelligent Information Technologies for Industry» (IITI'17). Vol. 1 / ed. by A. Abraham [et al.]. – [Springer International Publishing AG], 2018. – P. 520-525. – (Book ser. Advances in Intelligent Systems and Computing (AISC) ; Vol. 679). – URL : [https://link.springer.com/chapter/10.1007/978-3-319-68321-8\\_54](https://link.springer.com/chapter/10.1007/978-3-319-68321-8_54).

66. Никитин, Н.А. Automated sound generation by image color spectrum based on user ratings / Н.А. Никитин, В.Л. Розалиев, Ю.А. Орлова // Information Innovative Technologies (I2T) : materials of the International Scientific-Practical Conference (Prague, Czech Republic, April 24-28, 2017) / ed. by S.U. Uvaysov, I.A. Ivanov ; Moscow Technological Univ. (MIREA), Russian Centre of Science and Culture in Prague, Association of graduates and employees of AFEA named after prof. Zhukovsky [et al.]. - Moscow, 2017. - C. 462-465.

67. Иоханнес, И. Искусство цвета / И. Иоханнес. – М. : Д. Аронов, 2013. – 96 с.

68. Вахромеев, В. Элементарная теория музыки / В. Вахромеев. – М. : Государственное музыкальное издательство, 1962. – 248 с.

69. Russo, W. Composing music: a new approach / W. Russo. – Chicago : The University Of Chicago Press, 1983. – 193 p.
70. Гаранян, Г. Аранжировка для эстрадных инструментальных и вокально-инструментальных ансамблей / Г. Гаранян. – М. : Музыка, 1986. – 224 с.
71. Мясоедов, А. Учебник гармонии / А. Мясоедов. - М. : Музыка, 198. – 319 с.
72. Никитин, Н.А. Автоматизированный синтез музыкальных композиций на основе рекуррентных нейронных сетей / Н.А. Никитин, В.Л. Розалиев, Ю.А. Орлова // Искусственный интеллект в решении актуальных социальных и экономических проблем XXI века : сб. ст. по материалам Четвёртой всерос. науч.-практ. конф., проводимой в рамках Пермского естественнонаучного форума «Математика и глобальные вызовы XXI века» (г. Пермь, 21-24 мая 2019 г.). Ч. II / отв. ред.: Л. Н. Ясницкий ; Пермский гос. национальный исследовательский ун-т [и др.]. - Пермь, 2019. - С. 80-85.
73. Способин, В. Элементарная теория музыки / В. Способин. – М. : Государственное музыкальное издательство, 1963. – 202 с.
74. Никитин, Н.А. Development of web service for generating musical composition from an image / Н.А. Никитин, В.Л. Розалиев, Ю.А. Орлова // Open Semantic Technologies for Intelligent Systems (OSTIS) = Открытые семантические технологии проектирования интеллектуальных систем: сб. науч. тр. / редкол.: В. В. Голенков (гл. ред.) [и др.]; УО «Белорусский гос. ун-т информатики и радиоэлектроники». – Минск (Беларусь), 2020. – Вып. 4. – С. 345-348.
75. Никитин, Н.А. Разработка веб-сервиса для генерации музыкальной последовательности по изображению / Н.А. Никитин, В.Л. Розалиев, Ю.А. Орлова // Молодой учёный. - 2019. - № 51 (289), ч. 1. - С. 27-30.
76. Bradski, G., Kaehler, A. Learning OpenCV / G. Bradski, A. Kaehler. – Sebastopol : O'Reilly Media, Inc, 2008. – 571 p.
77. Сергеев С. Ф. Методы тестирования и оптимизации интерфейсов информационных систем: учебное пособие. – СПб: НИУ ИТМО, 2013. –117 с.

78. Whittaker, J.A., Arbon,J., Carollo,J. How Google Tests Software / James A. Whittaker, Jason Arbon, Jeff Carollo. –Westford: Addison-Wesley Professional, 2012. – 344 p.
79. Black, R. Pragmatic Software Testing: Becoming an Effective and Efficient Test Professional / Rex Black. – Indianapolis: John Wiley & Sons, 2007. – 366 p.
80. Блэк, Р. Ключевые процессы тестирования: Планирование, подготовка, проведение, совершенствование / Р. Блэк. – М.: Издательство «Лори», 2006. - 566 с.
81. Куликов, С. Тестирование программного обеспечения / С. Куликов. – М.: EPAM Systems, 2015. – 298с.
82. Клейн, Т. Дневник охотника за ошибками. Путешествие через джунгли проблем безопасности программного обеспечения / Т. Клейн. – М.: ДМК Пресс, 2015. – 240
83. Results of Using Neural Networks to Automatically Creation Musical Compositions Based on Color Image / В.Л. Розалиев, Н.А. Никитин, Ю.А. Орлова, А.В. Заболеева-Зотова // Proceedings of the Third International Scientific Conference «Intelligent Information Technologies for Industry» (IITI'18) (Sochi, Russia, September 17–21, 2018) / ed. by A. Abraham [et al.] ; Rostov State Transport University (Russia), VŠB – Technical University of Ostrava (Czech Republic), Russian Association for Artificial Intelligence (RAAI). – Cham: Springer Nature Switzerland AG, 2019. – Vol. 2. – P. 148-160. – (Book ser. Advances in Intelligent Systems and Computing (AISC); vol. 875).
84. Sound generation based on image color spectrum with using the recurrent neural network [Электронный ресурс] / Н.А. Никитин, В.Л. Розалиев, Ю.А. Орлова, А.В. Алексеев // Сборник трудов ИТНТ-2018 : IV международная конференция и молодёжная школа «Информационные технологии и нанотехнологии» (г. Самара, 24-27 апреля 2018 г.) : к 25-летию ИСОИ РАН / Правительство Самарской обл., Самарский национальный исследовательский ун-т им. акад. С. П. Королева, Ин-т систем обработки изображений РАН – филиал ФНИЦ «Кристаллография и фотоника» РАН, РФФИ, ООО «Специальные системы. Фотоника». - Самара, 2018.

- С. 2747-2755. - Режим доступа:  
[https://elibrary.ru/download/elibrary\\_34894667\\_94330727.pdf](https://elibrary.ru/download/elibrary_34894667_94330727.pdf).

85. Криспин, Л., Грегори, Дж. Гибкое тестирование. Практическое руководство для тестировщиков ПО и гибких команд / Л.Криспин, Дж. Грегори. - К.: Издательский дом «Вильямс», 2016. – 464

86. Майерс, Г., Баджетт, Т., Сандлер К. Искусство тестирования программ / Г. Майерс, Т. Баджетт, К. Сандлер. – К.: Издательский дом «Вильямс», 2016. – 272

87. Никитин, Н.А., Орлова, Ю.А., Розалиев, В.Л. Генерация жанровых музыкальных композиций по эмоциональному состоянию человека / Н.А. Никитин, Ю.А. Орлова, В.Л. Розалиев // Моделирование, оптимизация и информационные технологии. – 10(2). – 2022

## ПРИЛОЖЕНИЕ А. СХЕМЫ СООТНЕСЕНИЯ ЦВЕТОВ И НОТ

Таблица А1 - Схемы соотнесения цветов и нот

	Исаак Ньютон (1704)	Луи-Берtran Кастель (1734)	А. Уоллеса Римингтон (1893)	А. Эппли (1940)	Л. Дж. Бельмонт (1944)
C	Красный	Синий	Красный	Красный	Красный
C#	Красно - оранжевый	Зелёно-синий	Красно - оранжевый	Красно - оранжевый	Красно - оранжевый
D	Оранжевый	Зелёный	Жёлто - оранжевый	Оранжевый	Оранжевый
D#	Жёлто - оранжевый	Жёлто- зелёный	Оранжевый	Жёлто - оранжевый	Жёлто - оранжевый
E	Жёлтый	Жёлтый	Жёлтый	Жёлтый	Жёлтый
F	Зелёный	Жёлто - оранжевый	Жёлто- зелёный	Жёлто- зелёный	Жёлто- зелёный
F#	Зелёно-синий	Оранжевый	Зелёный	Зелёный	Зелёный
G	Синий	Красный	Зелёно-синий	Зелёно-синий	Зелёно-синий
G#	Сине- фиолетовый	Красно - оранжевый	Сине- фиолетовый	Розовый	Синий
A	Фиолетовый	Розовый	Фиолетовый	Синий	Фиолетовый
A#	Жёлто- зелёный	Сине- фиолетовый	Синий	Сине- фиолетовый	Розовый
H	Розовый	Фиолетовый	Розовый	Фиолетовый	Сине- фиолетовый

**ПРИЛОЖЕНИЕ Б. СВИДЕТЕЛЬСТВО О РЕГИСТРАЦИИ ПО**

**РОССИЙСКАЯ ФЕДЕРАЦИЯ**



**СВИДЕТЕЛЬСТВО**

о государственной регистрации программы для ЭВМ

**№ 2017616834**

**Программа генерации звуков по цветовой гамме  
изображений**

Правообладатель: **Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Волгоградский государственный технический университет»  
(ВолгГТУ) (RU)**

Авторы: **Никитин Никита Андреевич (RU), Розалиев Владимир  
Леонидович (RU), Орлова Юлия Александровна (RU)**

Заявка № **2017613654**

Дата поступления **24 апреля 2017 г.**

Дата государственной регистрации

в Реестре программ для ЭВМ **15 июня 2017 г.**

Руководитель Федеральной службы  
по интеллектуальной собственности

**Г.П. Ивлиев**



Рисунок Б1 – Свидетельство о государственной регистрации программы для ЭВМ

