

# Минобрнауки России

## Юго-Западный государственный университет

Кафедра программной инженерии

### ОТЧЕТ

о преддипломной (производственной) практике

наименование вида и типа практики

на (в) ООО "Предприятие ВТИ-сервис"

наименование предприятия, организации, учреждения

Студента 4 курса, группы ПО-116

курса, группы

Матвеева Игоря Олеговича

фамилия, имя, отчество

Руководитель практики от  
предприятия, организации,  
учреждения

Оценка \_\_\_\_\_

Директор

должность, звание, степень

Федосов Денис Валерьевич

фамилия и. о.

подпись, дата

Руководитель практики от  
университета

Оценка \_\_\_\_\_

к.т.н. доцент

должность, звание, степень

Чаплыгин А. А.

фамилия и. о.

подпись, дата

Члены комиссии

подпись, дата

фамилия и. о.

подпись, дата

фамилия и. о.

подпись, дата

фамилия и. о.

Курск 2025 г.

## СОДЕРЖАНИЕ

1	Анализ предметной области	4
1.1	Понятия и основная терминология	4
1.1.1	Эквалайзер	5
1.1.2	Компрессор	6
1.1.3	Реверберация	7
1.2	История создания и развития аудиопроцессоров	8
1.3	Классификация и технические особенности аудиопроцессоров	9
1.4	Технические проблемы и перспективы развития аудиопроцессоров	11
2	Техническое задание	14
2.1	Основание для разработки	14
2.2	Цель и назначение разработки	14
2.3	Требования к программной системе	14
2.3.1	Требования к данным программной системы	14
2.3.2	Функциональные требования к программной системе	15
2.3.2.1	Вариант использования «Оформление заказа»	16
2.3.2.2	Вариант использования «Воспроизведение аудио»	16
2.3.2.3	Вариант использования «Остановка воспроизведения»	17
2.3.2.4	Вариант использования «Изменения позиции воспроизведения»	17
2.3.2.5	Вариант использования «Применения эквалайзера»	18
2.3.2.6	Вариант использования «Применения компрессора»	18
2.3.2.7	Вариант использования «Применения реверберации»	19
2.3.2.8	Вариант использования «Сохранение обработанного файла»	19
2.3.2.9	Вариант использования «Визуализация данных»	19
2.3.3	Требования пользователя к интерфейсу веб-приложения	20
2.3.4	Нефункциональные требования к программной системе	21
2.3.4.1	Требования к надёжности	21
2.3.4.2	Требования к безопасности	22

2.3.4.3	Требования к программному обеспечению	23
2.3.4.4	Требования к аппаратному обеспечению	23
2.3.5	Требования к оформлению документации	23
3	Технический проект	24
3.1	Общая характеристика организации решения задачи	24
3.2	Обоснование выбора технологий проектирования	24
3.2.1	Язык программирования Python	24
3.2.2	Кроссплатформенная библиотека FFmpeg	25
3.3	Эффекты	26
3.3.1	Устройство эквалайзера	26
3.3.2	Устройство компрессора	27
3.3.3	Устройство реверберации	28
3.4	Архитектура программной системы	29
3.5	Компоненты программы	31
3.6	Архитектура приложения	32
3.7	Проект данных программной системы	33
3.7.1	Описание сущностей графического интерфейса	34
3.7.2	Описание сущностей логики процессора	35
3.7.3	Описание сущностей системного уровня	35
3.8	Проектирования пользовательского интерфейса	36
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	39

# **1 Анализ предметной области**

## **1.1 Понятия и основная терминология**

Аудиопроцессор - это электронное устройство или программный комплекс, предназначенный для обработки и управления звуковыми сигналами с целью улучшения качества звучания и адаптации аудиосистемы под конкретные условия. Мощный инструмент для обработки, анализа и улучшения звука.

Аудиопроцессор полезен как звукорежиссёрам, так и обычным пользователям, которые работают со звуком. Он включает в себя:

- модификацию аудио (изменять громкость, регулировать частоты);
- добавление эффектов (реверберация);
- анализирование звука (визуализация частот и формы волны);
- оптимизирование записи (компрессия, нормализация).

Преимущества аудиопроцессоров:

1. Гибкость обработки: возможность применять разные эффекты (эквалайзер, компрессор, реверб) в любом порядке.
2. Режим реального времени: обработка звука без задержек.
3. Визуализация звука: осциллограммы, спектрограммы и другие графики помогают анализировать аудио.
4. Доступность: программные аудиопроцессоры не требуют дорогого оборудования.

Недостатки аудиопроцессоров:

1. Задержка: в режиме реального времени возможны задержки из-за сложных вычислений.
2. Требовательность к ресурсам: некоторые эффекты (например, реверберация с большим буфером) нагружают процессор.
3. Качество зависит от алгоритмов: дешевые или плохо настроенные процессоры могут ухудшать звук (артефакты, искажения).
4. Ограниченная совместимость: некоторые форматы аудио могут не поддерживаться, например, редкие кодеки.

5. Сложность настройки: для тонкой настройки эффектов нужен опыт, например, подбор параметров компрессора.

### 1.1.1 Эквалайзер

Эквалайзер (англ. Equalizer, EQ) – это аудиопроцессор, который регулирует амплитуду (громкость) звука в разных частотных диапазонах. Он используется для: коррекции тонального баланса, компенсации акустики помещения, творческой обработки. Эквалайзер работает на принципе частотной фильтрации звукового сигнала. Звук как физическое явление представляет собой колебания воздуха с определенными частотными характеристиками. Человеческое ухо воспринимает частоты в диапазоне 20 Гц - 20 кГц, и эквалайзер позволяет управлять амплитудой этих частотных составляющих.

Основные физические параметры: частота в Герцах, амплитуда и фаза.

Эквалайзер состоит из: Полос (bands) частотные диапазоны, которые можно регулировать (низкие, средние и высокие); фильтры – электронные или цифровые схемы, изменяющие уровень определённых частот; регуляторы усиления (Gain) – позволяют увеличивать или уменьшать громкость в выбранной полосе; частота среза (Cutoff Frequency) – граница, на которой фильтр начинает действовать; добротность (Q-factor) – ширина полосы воздействия.

Принцип работы эквалайзера:

- ФНЧ (фильтр низких частот) – пропускает только низкие частоты;
- ПФ (полосовой фильтр) – выделяет средние частоты;
- ФВЧ (фильтр высоких частот) – пропускает только высокие частоты.

Каждый диапазон частот обрабатывается с помощью: усиления – увеличение громкости выбранной частоты, ослабление – уменьшение громкости.

После обработки всех полос сигналы суммируются, и на выходе получается модифицированный звук.

Эквалайзер - мощный инструмент, требующий понимания как технических аспектов, так и особенностей слухового восприятия. Грамотное ис-

пользование позволяет значительно улучшить качество звучания, в то время как неправильное применение может ухудшить звук.

### **1.1.2 Компрессор**

Компрессор или аудиокomppressor - это процессор динамического диапазона, который автоматически регулирует громкость аудиосигнала. Его основная функция - уменьшать разницу между самыми тихими и самыми громкими частями сигнала, делая общее звучание более ровным и контролируемым.

Компрессор состоит из:

1. Детектора огибающей – анализирует уровень входного сигнала в реальном времени. Существуют различные типы детектирования: пиковое (реагирует на мгновенные пики), среднеквадратичное (реагирует на воспринимаемую громкость) и комбинированные методы.

2. Блок управления состоит из: сравнивающего устройства (определяет превышение порога), вычислитель коэффициента усиления (рассчитывает необходимое ослабление), генератор управляющего напряжения (преобразует решение в управляющий сигнал).

3. Усилитель/ослабитель: динамическое регулирование уровня сигнала в соответствии с управляющим напряжением от детектора огибающей.

Основные параметры компрессии:

1. Threshold (порог) измеряется в dB - определяет уровень, при котором начинается компрессия.

2. Ratio (коэффициент компрессии) – определяет степень уменьшения динамического диапазона.

3. Attack (атака) – время реакции на превышение порога. Быстрая атака: лучше контролирует пики, медленная атака: сохраняет атаку инструментов.

4. Release (восстановление) – время возврата к нормальному уровню. Короткое восстановление: для быстрых сигналов. Длинное восстановление: для плавных изменений.

5. Knee (характеристика перехода) – определяет плавность включения компрессии.

6. Make-up Gain (компенсационное усиление) – усиление выходного сигнала после компрессии. Компенсирует общее снижение уровня.

Формула вычисления выходного сигнала:

$$\text{Output} = \text{Threshold} + (\text{Input} - \text{Threshold})/\text{Ratio}$$

Компрессор - это мощный инструмент, требующий глубокого понимания его параметров и их влияния на звук. Грамотное использование компрессии может значительно улучшить качество звучания, в то время как неправильное применение часто приводит к неестественному и ухудшенному звуку.

### 1.1.3 Реверберация

Реверберация — это акустическое явление, возникающее при многократных отражениях звуковых волн от поверхностей помещения. В отличие от эха, отдельные отражения при реверберации сливаются в непрерывный затухающий звуковой "хвост".

Принципы работы реверберации:

1. Временная иерархия: прямой звук -> ранние отражения -> поздние отражения -> диффузный хвост. Каждая фаза формирует определённые аспекты пространственного восприятия.

2. Частотная зависимость: высокочастотные компоненты затухают быстрее низкочастотных. Материалы поверхностей влияют на спектральный баланс отражений.

3. Плотности отражений: количество отражений растёт экспоненциально со временем. Достижение критической плотности формирует диффузное поле.

4. Пространственной дисперсии: отражения приходят со всех направлений. Корреляция между каналами уменьшается со временем.

5. Decay Time (RT60) - время затухания энергии на 60 дБ.

6. Dry/Wet Mix - баланс прямого и обработанного сигнала.

## 1.2 История создания и развития аудиопроцессоров

История аудиопроцессоров началась с первых попыток управления звуковыми сигналами в начале XX века, когда инженеры искали способы улучшения качества записей и передачи звука. Первые устройства обработки звука были чисто аналоговыми и основывались на пассивных и активных электронных компонентах – резисторах, конденсаторах, трансформаторах и лампах. Одним из первых аудиопроцессоров можно считать компрессор, разработанный в 1930-х годах для радиовещания, чтобы предотвратить перегрузку передатчиков. Эти ранние устройства использовали лампы и имели ограниченную функциональность, но заложили основы динамической обработки звука.

В 1950-х годах появились первые специализированные аналоговые процессоры, такие как эквалайзеры и ревербераторы. Например, EMT 140 – пластинчатый ревербератор, созданный в 1957 году, стал стандартом в студиях звукозаписи благодаря своему характерному теплему звучанию. В тот же период разрабатывались транзисторные компрессоры, такие как UREI 1176 (1967), который использовал полевые транзисторы (FET) для быстрого и агрессивного сжатия. Эти устройства были аналоговыми, но уже обладали регулируемыми параметрами (атака, релиз, порог), что делало их универсальными инструментами в студии.

1970-е годы стали временем расцвета аналоговых процессоров. Появились параметрические эквалайзеры, позволяющие точно настраивать частоту, добротность и усиление (например, API 550). В этот же период были созданы VCA-компрессоры (Voltage Controlled Amplifier), такие как dbx 160, которые предлагали более точное управление динамикой. Также развивались аналоговые задержки (например, Roland RE-201 Space Echo), использующие магнитную ленту для создания эффектов эха и повторов.

Переломным моментом стало появление цифровых аудиопроцессоров в конце 1970-х – начале 1980-х. Первые цифровые ревербераторы, такие как Lexicon 224 (1978), использовали алгоритмы на основе линий задержки с об-



ратной связью (алгоритм Шрёдера) и позволяли имитировать акустику разных помещений. В 1980-х цифровая обработка звука стала массовой благодаря развитию микропроцессоров. Появились мультиэффект-процессоры (например, Eventide H3000), которые объединяли реверберацию, задержку, хорус и другие эффекты в одном устройстве.

1990-е годы ознаменовались переходом к программной обработке звука (DSP). С появлением мощных компьютеров и плагинов (таких как Waves, TC Electronic) аудиопроцессоры стали виртуальными. Это позволило использовать сверточную реверберацию (Convolution Reverb), которая воспроизводит импульсные характеристики реальных помещений с высокой точностью. Также развивались алгоритмические ревербераторы, такие как Altiverb и ValhallaDSP, которые сочетали физическое моделирование с гибкостью цифровых методов.

В 2000-х и 2010-х годах аудиопроцессоры стали еще более сложными благодаря искусственному интеллекту и машинному обучению. Например, iZotope RX использует спектральный анализ для восстановления поврежденных записей, а Neural DSP применяет нейросети для эмуляции гитарных усилителей и эффектов. Современные процессоры, такие как Universal Audio UAD и Plugin Alliance, сочетают аппаратное ускорение с продвинутыми алгоритмами, обеспечивая минимальные задержки и студийное качество в реальном времени.

Сегодня аудиопроцессоры продолжают развиваться в сторону иммерсивного звука (Dolby Atmos, Ambisonics) и облачных технологий, позволяющих обрабатывать звук удаленно. История аудиопроцессоров – это эволюция от простых аналоговых схем к сложным цифровым системам, которые могут точно моделировать физические процессы и создавать принципиально новые звуковые эффекты.

### **1.3 Классификация и технические особенности аудиопроцессоров**

Классификация аудиопроцессоров по сфере применения охватывает несколько основных категорий устройств. Студийные процессоры предна-

значены для профессиональной звукозаписи и сведения, они отличаются высокой точностью обработки с разрядностью 24-32 бита и частотой дискретизации до 192 кГц, минимальным уровнем шумов и искажений, расширенным набором параметров включая многополосную обработку и side-chain, а также поддержкой профессиональных интерфейсов таких как Dante и MADI. Концертные и лайв-процессоры оптимизированы для работы в реальном времени и характеризуются сверхнизкой задержкой менее 2 мс, упрощенным управлением с предустановками, повышенной надежностью конструкции и специализированными функциями вроде подавления обратной связи и автоматического микширования. Потребительские решения ориентированы на массовый рынок и предлагают компактные размеры, портативность, упрощенные алгоритмы обработки и интеграцию с мобильными устройствами через Bluetooth и USB при доступной цене. Встраиваемые системы представляют собой специализированные решения для бытовой техники с минимальным энергопотреблением, аппаратной оптимизацией под конкретные задачи и автоматическими режимами работы, применяемые в телевизорах, автомобильных аудиосистемах и умных колонках.

С технической точки зрения цифровые процессоры реализуются на различных платформах. DSP (цифровые сигнальные процессоры) используют специализированные чипы вроде Analog Devices SHARC или Texas Instruments C6000, оптимизированные для потоковой обработки с параллельным выполнением операций. FPGA (программируемые логические матрицы) обеспечивают сверхнизкую задержку на уровне тактов процессора и гибкость в реализации нестандартных алгоритмов. Ключевыми характеристиками цифровых процессоров являются производительность в GMAC/s, разрядность обработки 32 или 64 бита, поддержка плавающей точки и энергоэффективность.

Методы обработки сигналов в аудиопроцессорах включают линейные и нелинейные преобразования. Линейные методы охватывают частотную коррекцию с использованием БИХ и КИХ-фильтров, линейную свертку для реверберации и пространственной обработки, а также корреляционный анализ.

Нелинейные методы включают динамическую обработку типа компрессии и лимитирования, тональные преобразования вроде дисторшна и сатурации, а также амплитудную модуляцию. Адаптивные алгоритмы позволяют автоматически подстраивать параметры для шумоподавления и устранения обратной связи, используя статистический анализ сигнала и системы с обратной связью. Современные подходы включают машинное обучение с нейросетевыми моделями для восстановления аудио, разделения источников и интеллектуального сведения, а также GAN-архитектуры для синтеза эффектов. Биоинспирированные методы моделируют слуховую систему человека с использованием кохлеарных фильтров и психоакустической оптимизации.

Каждый метод обработки имеет свои преимущества и ограничения. Линеинные методы обеспечивают предсказуемость и стабильность, но имеют ограниченный диапазон задач. Нелинейные методы предоставляют широкие творческие возможности, но могут вызывать артефакты обработки. Адаптивные алгоритмы автоматизируют рутинные операции, но требуют времени на адаптацию. Машинное обучение открывает качественно новые возможности, но крайне требовательно к вычислительным ресурсам. Выбор конкретного метода и платформы зависит от требований к качеству обработки, работе в реальном времени и экономической целесообразности, что в совокупности определяет современное состояние и перспективы развития технологий аудиообработки.

#### **1.4 Технические проблемы и перспективы развития аудиопроцессоров**

При разработке и эксплуатации аудиопроцессоров специалисты сталкиваются с рядом технических сложностей и ограничений. Одной из ключевых проблем являются фазовые искажения, возникающие при обработке сигнала различными фильтрами и эффектами, что может приводить к ухудшению стереокартины и неестественному звучанию. Не менее важной проблемой выступают артефакты обработки - нежелательные звуковые искажения, проявляющиеся в виде цифровых щелчков, металлического призвука

или неестественного окрашивания тембра. Эти артефакты особенно заметны при агрессивной обработке или каскадном включении нескольких эффектов.

Вычислительная сложность современных алгоритмов обработки звука создает серьезные требования к аппаратным ресурсам. Сложные эффекты вроде сверточной реверберации или нейросетевой обработки требуют значительной процессорной мощности, что ограничивает их применение в реальном времени. Проблема задержки (latency) особенно критична для лайв-обработки и мониторинга, где даже небольшие задержки в 10-20 мс могут нарушить восприятие музыкантами своего исполнения.

Вопросы совместимости форматов остаются актуальными в условиях многообразия аудиостандартов. Проблемы возникают при взаимодействии оборудования разных производителей, использовании устаревших протоколов или при попытках интеграции профессиональных и потребительских решений. Особенно остро это проявляется при работе с многоканальными форматами и метаданными.

Перспективные направления развития аудиопроцессоров включают несколько многообещающих технологий. Квантовые методы обработки теоретически могут решить проблему вычислительной сложности для определенных классов алгоритмов, хотя практические реализации пока находятся в стадии исследований. Иммерсивный звук (3D-аудио) становится новым стандартом для кинопроизводства и игровой индустрии, требуя разработки специализированных процессоров для работы с объектно-ориентированным звуком в форматах Dolby Atmos и Ambisonics.

Адаптивные системы на основе искусственного интеллекта позволяют автоматически подстраивать параметры обработки под конкретный материал и акустические условия. Интеграция с VR/AR открывает новые возможности для создания полностью интерактивных звуковых ландшафтов, где обработка происходит в реальном времени с учетом действий пользователя. Экологичные решения направлены на снижение энергопотребления аудиооборудования без потери качества обработки.

Аспекты безопасности и надежности включают несколько важных направлений. Защита от перегрузок предотвращает повреждение оборудования при работе с мощными сигналами. Системы диагностики позволяют оперативно выявлять неисправности и отклонения в работе процессоров. Резервирование критически важных каналов обеспечивает бесперебойную работу в профессиональных приложениях. Защита от электромагнитных помех остается актуальной проблемой, особенно для аналоговых трактов и высокочувствительных микрофонных входов.

Современные разработки в области аудиопроцессоров направлены на преодоление существующих ограничений через внедрение новых алгоритмов и аппаратных архитектур. Особое внимание уделяется созданию интеллектуальных систем, способных адаптироваться к изменяющимся условиям работы, сохраняя при этом высокое качество звучания и надежность. Параллельно ведется работа над упрощением интерфейсов и снижением энергопотребления, что делает профессиональные технологии обработки звука доступными для более широкого круга пользователей.

## **2 Техническое задание**

### **2.1 Основание для разработки**

Полное наименование системы: «Программно-информационная система для создания и работы с аудиозаписями. Подсистема анализа и улучшения качества». Основанием для разработки программы является приказ ректора ЮЗГУ от «17» апреля 2025 г. №1828-с «Об утверждении тем выпускных квалификационных работ».

### **2.2 Цель и назначение разработки**

Создание удобного и функционального инструмента для обработки аудиофайлов с возможностью применения различных звуковых эффектов в реальном времени, визуализации аудиоданных и сохранения результатов обработки. Назначение программы - предоставить пользователям простой в использовании, но мощный инструмент для базовой звуковой обработки, включающий эквалайзер, компрессор и ревербератор, с интуитивно понятным графическим интерфейсом и визуальной обратной связью. Код реализует многопоточную обработку аудио для минимизации задержек, поддерживает загрузку различных аудиоформатов, обеспечивает плавную визуализацию формы волны и частотного спектра, а также предлагает оптимизированные алгоритмы обработки звука. Приложение предназначено для музыкантов, звукорежиссеров и людей, которым требуется быстрый доступ к основным инструментам звуковой обработки без необходимости использования сложных профессиональных DAW.

### **2.3 Требования к программной системе**

#### **2.3.1 Требования к данным программной системы**

Входные данные:

- аудиофайлы форматов WAV, MP3, OGG, FLAC;

- параметры эффектов (эквалайзер, компрессор, реверберация), задаваемые пользователем.

Обрабатываемые данные:

- аудиосигнал в виде массива `numpy.ndarray` (форма (N, 1) для моно, (N, 2) для стерео);
- частота дискретизации (стандартно 44100 Гц, но поддерживаются другие значения);
- временные и спектральные данные для визуализации (форма волны, частотный спектр).

Выходные данные:

- обработанный аудиофайл (экспорт в WAV, MP3);
- графики формы сигнала и АЧХ в реальном времени.

### **2.3.2 Функциональные требования к программной системе**

В разрабатываемой программе пользователь может:

- загружать аудиофайлы разных форматов (wav, mp3, ogg, flac);
- экспортировать результат, с применением выбранных им эффектов, в двух форматах: wav и mp3;
- воспроизводить, останавливать аудио, изменить позицию воспроизведения, узнать текущее время проигрывания;
- применять эффекты в режиме реального времени, включать и отключать их;
- применить эквалайзер, настроить низкие/средние/высокие частоты;
- применить компрессор, настроить порог, коэффициент сжатия, время атаки и восстановления сигнала, усиление;
- применить реверберацию, настроить уровень эффекта, размер комнаты, затухание и чистый звук;
- видеть форму сигнала в режиме реального времени, посмотреть АЧХ аудио.

На рисунке 2.1 изображена диаграмма прецедентов

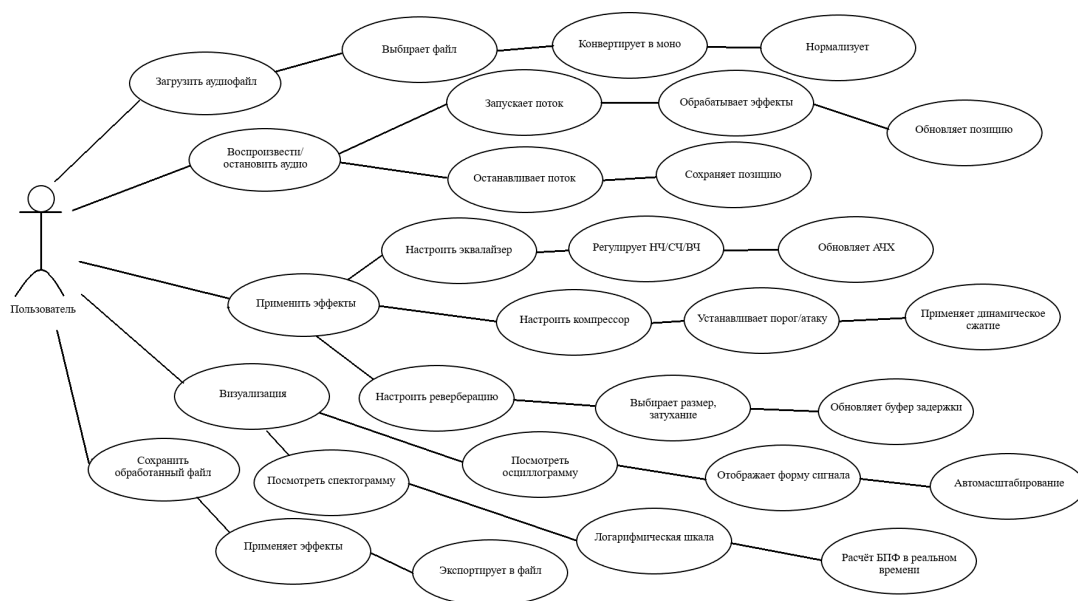


Рисунок 2.1 – Диаграмма прецедентов

### 2.3.2.1 Вариант использования «Оформление заказа»

Заинтересованные лица и их требования: пользователь хочет загрузить аудиофайл для последующей обработки.

Требования: поддержка форматов (WAV, MP3, OGG, FLAC), автоматическая конвертация в моно, нормализация амплитуды.

Предусловие: приложение запущено, файл существует и доступен для чтения.

Постусловие: аудиоданные загружены в память, форма волны отображена на графике, кнопки "Воспроизвести" и "Сохранить" активированы.

Основной успешный сценарий:

1. Пользователь нажимает кнопку "Загрузить".
2. Открывается диалоговое окно выбора файла.
3. Пользователь выбирает файл и подтверждает выбор.

### 2.3.2.2 Вариант использования «Воспроизведение аудио»

Заинтересованные лица и их требования: пользователь хочет прослушать загруженный файл с возможностью применения эффектов в реальном времени.



Предусловие: аудиофайл успешно загружен, аудиоустройство вывода доступно.

Постусловие: аудиопоток запущен, графики обновляются в реальном времени.

Основной успешный сценарий:

1. Пользователь нажимает кнопку «Воспроизвести».
2. Система инициализирует аудиопоток.
3. Callback-функция начинает передавать данные в устройство.
4. Позиция воспроизведения обновляется каждые 100 мс.
5. Сигнал и АЧХ отображаются в реальном времени.

### **2.3.2.3 Вариант использования «Остановка воспроизведения»**

Заинтересованные лица и их требования: пользователь хочет немедленно прекратить воспроизведение аудио.

Предусловие: идёт воспроизведение аудиофайла.

Постусловие: аудиопоток полностью остановлен, интерфейс обновлен.

Основной успешный сценарий:

1. Пользователь нажимает кнопку "Стоп".
2. Система останавливает воспроизведение аудио.
3. Текущая позиция фиксируется для возможности дальнейшего продолжения.

### **2.3.2.4 Вариант использования «Изменения позиции воспроизведения»**

Заинтересованные лица и их требования: пользователь хочет перемотать аудио на произвольную позицию.

Предусловие: аудиофайл загружен.

Постусловие: текущая позиция воспроизведения изменена, при активном воспроизведении звук продолжается с новой позиции, интерфейс синхронизирован.

Основной успешный сценарий:

1. Пользователь перемещает ползунок позиции мышью.
2. Система пересчитывает позицию.
3. При воспроизведении поток начинает чтение с новой позиции.

#### **2.3.2.5 Вариант использования «Применения эквалайзера»**

Заинтересованные лица и их требования: пользователь хочет настроить частотный баланс (НЧ/СЧ/ВЧ).

Предусловие: файл загружен, вкладка "Эквалайзер" открыта.

Постусловие: параметры эквалайзера применены к аудиопотоку, график АЧХ обновлен.

Основной успешный сценарий:

1. Пользователь включает чекбокс «Эквалайзер».
2. Регулирует ползунки НЧ/СЧ/ВЧ.
3. Система пересчитывает коэффициент БИХ-фильтров.
4. Аудиозапись изменяется в соответствии с настройками пользователя.
5. АЧХ отображается на графике.

#### **2.3.2.6 Вариант использования «Применения компрессора»**

Заинтересованные лица и их требования: пользователь хочет изменить динамический диапазон аудиосигнала для более равномерного звучания.

Предусловие: аудиофайл успешно загружен, вкладка "Компрессор" открыта.

Постусловие: параметры компрессора применены к аудиопотоку, изменения слышны при воспроизведении.

Основной успешный сценарий:

1. Пользователь активирует чекбокс «Компрессор».
2. Устанавливает порог, коэффициент компрессии, время атаки и восстановления, усиление выходного сигнала.
3. Аудио обрабатывается в режиме реального времени.

### **2.3.2.7 Вариант использования «Применения реверберации»**

Заинтересованные лица и их требования: пользователь хочет добавить эффект реверберации к аудиосигналу.

Предусловие: аудиофайл успешно загружен, вкладка "Реверберация" открыта.

Постусловие: эффект реверберации применен к аудиопотоку, буфер реверберации инициализирован с новыми параметрами.

Основной успешный сценарий:

1. Пользователь активирует чекбокс "Реверберация".
2. Пользователь регулирует параметры: уровень эффекта, исходный звук, размер комнаты, затухание.
3. Аудио обрабатывается в режиме реального времени.

### **2.3.2.8 Вариант использования «Сохранение обработанного файла»**

Заинтересованные лица и их требования: пользователь хочет экспортировать результат с примененными эффектами.

Предусловие: файл загружен, эффекты настроены.

Постусловие: файл сохранен на диск в выбранном формате.

Основной успешный сценарий:

1. Пользователь нажимает "Сохранить".
2. Открывается диалог выбора формата (WAV/MP3) и пути.
3. Пользователь выбирает формат, путь и название файла.
4. Система применяет выбранные эффекты, экспортирует файл.
5. Файл успешно сохранён на диске.

### **2.3.2.9 Вариант использования «Визуализация данных»**

Заинтересованные лица и их требования: пользователь хочет видеть графическое представление аудиосигнала.

Предусловие: аудиофайл загружен и идет воспроизведение.

Постусловие: графики формы волны и АЧХ отображаются и обновляются в режиме реального времени.

Основной успешный сценарий:

1. Пользователь нажимает «Воспроизвести».
2. Система отображает графики формы волны и АЧХ.

### **2.3.3 Требования пользователя к интерфейсу веб-приложения**

Приложение должно иметь следующие основные функции:

1. Основные элементы управления - кнопки управления воспроизведением: "Загрузить" "Воспроизвести" "Стоп" "Сохранить". Ползунок позиции воспроизведения, точное позиционирование, индикатор текущей позиции в формате MM:SS / MM:SS.

2. Визуализация аудио - два синхронизированных графика: верхний график формы волны и нижний график амплитудно-частотной характеристики. Подписи осей с единицами измерения (амплитуда, частота).

3. Панель эффектов (вкладки):

(а) Эквалайзер - Три ползунка с подписями: низкие частоты (150Hz):  $\pm 24$  дБ, средние частоты (1kHz):  $\pm 24$  дБ, высокие частоты (5kHz):  $\pm 24$  дБ. Чекбокс активации эффекта.

(b) Компрессор - ползунки параметров: порог: -60..0 дБ, соотношение: 1..20, атака: 1..500 мс, восстановление: 10..2000 мс, усиление: 0..24 дБ. Чекбокс активации эффекта.

(c) Реверберация: ползунки параметров: уровень эффекта: 0..1, сухой сигнал: 0..1, размер помещения: 0.1..2.0, затухание: 0.1..0.9. Чекбокс активации эффекта.

4. Обратная связь и состояние: индикатор загрузки при обработке файлов. Всплывающие уведомления об ошибках: при неудачной загрузке файла, при проблемах с аудиоустройством, при ошибках сохранения. Изменение состояния кнопок: "Воспроизвести" активна только при загруженном файле, "Стоп" активна только во время воспроизведения, "Сохранить" активна только после загрузки файла.

5. Производительность: задержка отклика интерфейса не более 100 мс, плавная анимация графиков без рывков, минимальное потребление ресурсов в фоновом режиме.

## **2.3.4 Нефункциональные требования к программной системе**

### **2.3.4.1 Требования к надёжности**

В процессе работы аудио-процессора могут возникнуть следующие аварийные ситуации:

- потеря доступа к аудиоустройству в связи с его отключением, изменением системных настроек звука или конфликтом с другим приложением;
- попытка загрузки повреждённого или неподдерживаемого аудиофайла с несовместимым форматом, битрейтом или частотой дискретизации;
- неожиданное прекращение работы из-за нехватки системных ресурсов при обработке объёмных файлов или одновременном применении нескольких ресурсоёмких эффектов;
- ошибки в работе эффектов обработки звука, приводящие к искажению аудиосигнала или сбоям в воспроизведении.

Приложение должно автоматически определять доступные аудиоустройства и переключаться между ними при потере соединения с текущим устройством. В случае невозможности восстановления подключения система должна сохранять возможность обработки звука без функции воспроизведения с уведомлением пользователя о возникшей проблеме.

Для работы с аудиофайлами приложение должно проверять их целостность и соответствие поддерживаемым форматам перед загрузкой, а также автоматически выполнять необходимые преобразования (нормализацию, конвертацию в моно, приведение к стандартной частоте дискретизации). При обнаружении критических ошибок в файле пользователь должен получить понятное сообщение о проблеме с указанием конкретной причины.

Для предотвращения аварийного завершения из-за нехватки ресурсов система должна контролировать использование оперативной памяти и про-

цессорного времени, ограничивать размеры буферов обработки и корректно завершать работу при приближении к предельным значениям. Все параметры эффектов и текущее состояние обработки должны сохраняться даже при аварийном завершении работы.

При возникновении ошибок в работе эффектов обработки звука система должна автоматически сбрасывать параметры эффектов к безопасным значениям, сохраняя при этом возможность дальнейшей работы. Все критические ошибки должны фиксироваться в системном логе с указанием времени возникновения, типа ошибки и состояния системы на момент сбоя.

#### **2.3.4.2 Требования к безопасности**

Требования к приложению:

1. Перед загрузкой файла, система должна проверять формат и корректность заголовков. При обнаружении повреждённых данных программа будет выводить ошибку без попытки обработки.

2. Для безопасности управления памятью и ресурсами необходима изоляция обработки аудио, то есть каждый эффект будет применяться в отдельном потоке с контролем потребления ресурсов и будет ограничен буфе. Так же потребуются очистка временных данных, чтобы после завершения обработки или при ошибке все промежуточные буферы будут очищаться.

3. Для защиты от вирусов применится проверка структуры файлов, анализ сигнатуры файла перед декодированием, ограничение максимальной длительности обрабатываемого аудио.

4. Пользовательские настройки будут проверяться перед применением, а недопустимые значения автоматически скорректируются до ближайших безопасных.

5. При завершении работы, приложение будет останавливать все потоки, освобождать аудиоустройства.

#### **2.3.4.3 Требования к программному обеспечению**

Для реализации программы будет использоваться язык программирования Python и библиотеки: NumPy, SciPy, PyDub, SoundDevice, Matplotlib, Numba.

Для работы приложения требуется Windows 10/11 (64-bit)

#### **2.3.4.4 Требования к аппаратному обеспечению**

Минимальная конфигурация:

Центральный процессор с количеством ядер от 2 и выше с тактовой частотой от 2.0 ГГц. Поддержка инструкций SSE4.2. Оперативная память - 4 ГБ. 100 МБ свободного места для установки. Поддержка ASIO/WASAPI для профессионального использования.

#### **2.3.5 Требования к оформлению документации**

Требования к стадиям разработки программ и программной документации для вычислительных машин, комплексов и систем независимо от их назначения и области применения, этапам и содержанию работ устанавливаются ГОСТ 19.102-77 Программная документация должна включать в себя:

- анализ предметной области;
- техническое задание;
- технический проект.

### **3 Технический проект**

#### **3.1 Общая характеристика организации решения задачи**

Необходимо спроектировать и разработать программный комплекс для обработки аудиосигналов, сочетающий функциональность профессиональных аудиоредакторов с простотой и удобством интерфейса.

В приложении должно быть:

- модульная структура с чётким разделением на: ядро обработки, пользовательский интерфейс, системную обвязку;
- гибкая система эффектов: трёхполосный параметрический эквалайзер с настраиваемыми частотами среза, компрессор с регулируемыми параметрами, алгоритм реверберации;
- оптимизированная обработка сигнала: поддержка многопоточной обработки для работы в реальном времени, аппаратно-независимая реализация звукового движка, автоматическая нормализация входного и выходного сигнала;
- интерактивная визуализация; формы волны во временной области, АЧХ применяемых фильтров.

#### **3.2 Обоснование выбора технологий проектирования**

Используемые для создания программно-информационной системы языки и технологии отвечают современным практикам разработки, позволяют достичь высокой производительности и отказоустойчивости программы.

##### **3.2.1 Язык программирования Python**

Python - основной язык разработки проекта. Python выбран благодаря своей гибкости, богатой экосистеме библиотек и кроссплатформенной поддержке. Его синтаксис позволяет быстро разрабатывать сложные алгоритмы обработки сигналов, а динамическая типизация упрощает интеграцию с различными аудиоформатами и устройствами.

Используемые библиотеки:



- NumPy - обеспечивает высокопроизводительные вычисления с многомерными массивами, что критично для обработки аудиоданных, представленных в виде временных рядов;
- SciPy - используется для реализации цифровых фильтров (эквалайзер), БПФ (анализ спектра) и других математических операций;
- PyDub - библиотека для работы с аудиофайлами, предоставляющая: простые методы загрузки и сохранения в форматах WAV, MP3, OGG, FLAC, базовые операции, такие как обрезка, наложение, изменение громкости, конвертация частоты дискретизации, интеграцию с FFmpeg для поддержки дополнительных кодеков;
- SoundDevice - обеспечивает низкоуровневый доступ к аудиоустройствам через PortAudio. Ключевые функции: воспроизведение и запись в реальном времени с минимальной задержкой, поддержка ASIO, WASAPI, Core Audio для профессиональных аудиоинтерфейсов, гибкая настройка параметров потока: частота дискретизации, размер буфера, количество каналов;
- Matplotlib - используется для визуализации аудиоданных: построение осциллограмм (форма сигнала во временной области), отображение АЧХ (амплитудно-частотных характеристик) с логарифмической шкалой, интерактивное обновление графиков в реальном времени;
- Tkinter - стандартная библиотека Python для создания графического интерфейса. В проекте применяется для: построения основного окна с вкладками (эквалайзер, компрессор, реверберация), реализации интерактивных элементов: ползунки, кнопки, метки, интеграции графиков Matplotlib через FigureCanvasTkAgg;
- Numba - JIT-компилятор для оптимизации вычислительно сложных участков кода: ускорение алгоритмов компрессии и реверберации в 5–10 раз, поддержка многопоточности.

### **3.2.2 Кроссплатформенная библиотека FFmpeg**

FFmpeg — это мощная кроссплатформенная библиотека для обработки мультимедиа, используемая в проекте для работы с аудиофайлами различных

форматов. Взаимодействие с FFmpeg осуществляется через обёртку PyDub, что значительно упрощает операции чтения, записи и конвертации аудио.

Основные функции FFmpeg в проекте:

1. Поддержка множества аудиоформатов. Позволяет загружать и сохранять файлы в форматах: без сжатия: WAV, AIFF, с потерями: MP3, AAC, OGG, без потерь: FLAC, ALAC. Обеспечивает автоматическое определение кодека при загрузке.

2. Конвертация аудио: изменение частоты дискретизации, преобразование между форматами, конвертация числа каналов.

3. Нормализация и обработка. Автоматическая регулировка громкости.

FFmpeg был выбран по ряду преимуществ:

- универсальность: поддержка 100+ кодеков и контейнеров;
- стабильность: отлаженные алгоритмы декодирования/кодирования;
- производительность: оптимизированные нативные библиотеки;
- гибкость: Возможность тонкой настройки параметров через командные опции.

### **3.3 Эффекты**

#### **3.3.1 Устройство эквалайзера**

Эквалайзер — это устройство или программный алгоритм, предназначенный для корректировки амплитудно-частотной характеристики (АЧХ) звукового сигнала. Он позволяет усиливать или ослаблять определённые частотные диапазоны, изменяя тембр звука. Эквалайзер работает, разделяя входной аудиосигнал на несколько частотных полос (диапазонов), каждая из которых обрабатывается отдельно. После обработки всех полос сигналы складываются, и на выходе получается звук с изменённой частотной характеристикой.

В программе реализован цифровой параметрический эквалайзер, у которого есть три полосы частот, а так же коррекция их по громкости в децибе-

лах. Такие эквалайзеры часто используются в профессиональной звукозаписи и сведении аудио.

Цифровой IIR-фильтр (Биквадратный, на основе Баттерворта)

IIR-фильтр (Infinite Impulse Response — бесконечная импульсная характеристика) — это тип цифрового фильтра, который использует обратную связь, благодаря чему может иметь очень крутые склоны АЧХ при малом порядке.

Биквадратный фильтр (Biquad) — это частный случай IIR-фильтра 2-го порядка, который реализуется с помощью разностного уравнения и часто используется в эквалайзерах из-за своей эффективности.

Фильтр Баттерворта — один из самых популярных типов фильтров, обеспечивающий максимально гладкую АЧХ в полосе пропускания без пульсаций.

Этот эквалайзер обеспечивает:

- низкую задержку (важно для реального времени);
- минимальные фазовые искажения;
- точную обработку частот;
- защиту от перегрузки.

### 3.3.2 Устройство компрессора

Feed-forward компрессор — это тип динамического процессора, в котором уровень входного сигнала анализируется до применения gain reduction, что позволяет более точно контролировать амплитудные изменения. В сочетании с RMS-детектированием огибающей такой компрессор обеспечивает плавное и музыкальное сжатие, идеально подходящее для обработки вокала, гитар и других сложных сигналов с выраженными динамическими перепадами.

RMS (Root Mean Square) — среднеквадратичное значение сигнала, которое лучше отражает воспринимаемую громкость, чем пиковый уровень (Peak). На практике RMS вычисляется с использованием скользящего среднего или фильтра низких частот (ФНЧ) для сглаживания.

Основные этапы обработки:

- входной сигнал поступает на детектор огибающей;
- RMS-детектор вычисляет усреднённый уровень сигнала;
- порог (Threshold) определяет момент начала компрессии;
- коэффициент сжатия (Ratio) задаёт силу уменьшения уровня;
- временные параметры (Attack/Release) регулируют скорость реакции;
- Make-up Gain компенсирует потерянную громкость.

Ключевые особенности:

- плавность: RMS-детектирование снижает резкость обработки по сравнению с Peak;
- гибкость: Attack/Release настраиваются отдельно для атаки и восстановления;
- применение: вокал, живые инструменты, мастер-каналы.

### 3.3.3 Устройство реверберации

FDN (Feedback Delay Network) — это один из самых мощных алгоритмов цифровой реверберации, используемый для создания реалистичных пространственных эффектов. Он основан на множестве линий задержки с обратной связью, образующих сложную сеть, имитирующую отражения в помещении.

Принцип работы:

- входной сигнал разделяется на несколько параллельных линий задержки, каждая из которой задерживает сигнал на разное время;
- после задержки сигнал проходит через фильтр демпфирования (имитация потери высоких частот);
- затем он умножается на коэффициенты матрицы обратной связи, определяющей, какая часть сигнала возвращается в каждую линию;
- обновлённый сигнал снова поступает на вход линий задержки;
- результирующий реверберационный сигнал формируется как сумма выходов всех линий.

Ключевые особенности FDN:

- реалистичность — лучше имитирует поздние отражения, чем алгоритмы на основе простых задержек;
- гибкость — можно настраивать время реверберации, демпфирование и пространственность;
- стабильность — ортогональная матрица гарантирует отсутствие бесконечного нарастания.

### **3.4 Архитектура программной системы**

На рисунке 3.1 представлена многослойная архитектура системы с чётким разделением ответственности между компонентами.

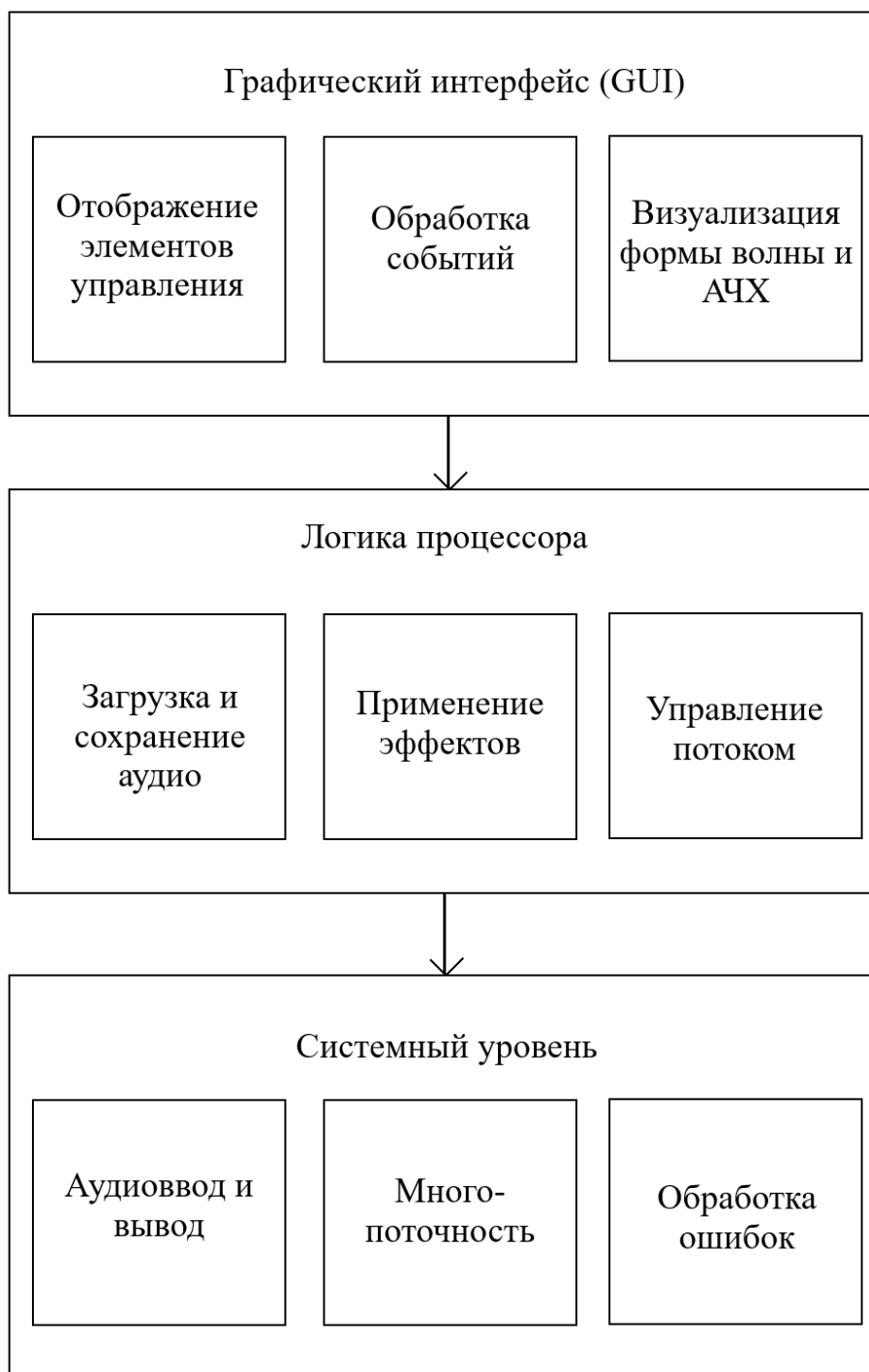


Рисунок 3.1 – Архитектура программной системы

1. Графический интерфейс отображает элементы управления (кнопки, ползунки, вкладки), визуализирует аудиоданные в реальном времени (Осциллограмма, АЧХ), обрабатывает пользовательские события (нажатия кнопок, изменение параметров).

2. Логика процессора отвечает за загрузку/сохранение аудио, чтение файлов через PyDub/FFmpeg, конвертирует в формат float32 для обработки. Применяет эффекты: эквалайзер (БИХ-фильтры для НЧ/СЧ/ВЧ), компрессор (динамическое сжатие с параметрами), реверберация. Буферизирует данные и синхронизирует позицию воспроизведения.

3. На уровне системы происходит ввод и вывод аудио, создаётся отдельный поток для обработки аудио и очереди для передачи данных между потоками. Обрабатываются ошибки, перехватываются исключения и логируются в файл.

### **3.5 Компоненты программы**

Приложение Audio Processor состоит из следующих ключевых компонентов, организованных в модульную структуру:

- графический интерфейс включает главное окно, состоящее из кнопок управление воспроизведением и загрузки/сохранения, вкладок эффектов (эквалайзер, компрессор и ревербератор), окна отображение визуализации формы волны и АЧХ;
- логика процессора читает и обрабатывает аудиофайлы, применяется выбранные эффекты по очереди, управляет всей работой программы;
- системное устройство воспроизводит звук, работает с файлами, открывает и сохраняет их, управляет несколькими задачами одновременно;
- вспомогательные модули включают частотный анализ, автоматически регулирует громкость и записывает ошибки и события.

Все части программы взаимодействуют между собой по чётким правилам. Когда пользователь меняет настройки, интерфейс передаёт их в модуль обработки, который применяет эффекты и отправляет результат на воспроизведение и отображение. Для плавной работы использовались очереди за-

дач для работы в многопоточном режиме, оптимизирует сложные вычисления для быстрой работы.

### 3.6 Архитектура приложения

Приложение построено по многослойной модульной архитектуре с разделением на логические компоненты, взаимодействующие через четко определенные интерфейсы. В основе лежит ядро обработки аудиосигналов, окруженное графическим интерфейсом, системными сервисами и вспомогательными модулями. Графический интерфейс реализован на Tkinter и включает:

- главное окно (MainWindow) с вкладками для управления эффектами, кнопками воспроизведения/паузы и областью визуализации;
- панель эквалайзера (EQPanel) с тремя ползунками (НЧ/СЧ/ВЧ), отображающая АЧХ через Matplotlib;
- панель компрессора (CompressorPanel) для настройки порога, сжатия, атаки, восстановления, усиления;
- панель реверберации (ReverbPanel) для настройки эффект, сухой сигнал, размер комнаты, затухание;
- визуализатор спектра (SpectrumAnalyzer), отображающий осциллограмму (форма сигнала) и частотный спектр (БПФ) в реальном времени;
- прогресс-бар с управлением позицией воспроизведения и отображением длительности трека.

Логический процессор — центральный модуль, отвечающий за:

- загрузку и конвертацию аудио через PyDub (с использованием FFmpeg как бэкенда), включая поддержку WAV, MP3, FLAC;
- эквалайзер на БИХ-фильтрах с частотами 150 Гц (НЧ), 1 кГц (СЧ), 5 кГц (ВЧ);
- компрессор с параметрами knee, makeup gain;
- реверберация на основе алгоритма Schroeder (комбинация линий задержки и FIR-фильтров);
- буферизацию данных для плавного воспроизведения и нормализацию уровня сигнала.



Системный слой обеспечивает интеграцию с ОС и оборудованием:

- аудиопоток (AudioStream) на базе SoundDevice, обрабатывающий ввод/вывод в реальном времени с настройкой latency и sample rate (44.1–192 кГц);
- менеджер потоков (ThreadManager) для параллельной обработки (отдельные потоки для: GUI, аудиообработки, визуализации);
- файловый менеджер (FileManager) с кэшированием загруженных треков и поддержкой метаданных (ID3-теги).

Архитектура обеспечивает масштабируемость (добавление новых эффектов через модули), производительность (JIT-компиляция, многопоточность) и отказоустойчивость (изоляция сбоев в отдельных потоках).

### **3.7 Проект данных программной системы**

Система оперирует тремя основными категориями данных: входными, промежуточными и выходными. Входные данные включают аудиофайлы в форматах WAV, MP3, FLAC и OGG с поддержкой различных характеристик (битность 16-24 бит, частота дискретизации 44.1-192 кГц), а также параметры эффектов, настраиваемые пользователем через графический интерфейс. Для эквалайзера это уровни усиления по трём полосам (низкие на 150 Гц, средние на 1 кГц, высокие на 5 кГц с диапазоном  $\pm 24$  дБ), для компрессора — порог срабатывания от -60 до 0 дБ, коэффициент компрессии от 1:1 до 20:1, время атаки и восстановления от 0.1 до 500 мс, для реверберации — соотношение обработанного и исходного сигнала, коэффициент затухания и виртуальный размер помещения.

Промежуточные данные представлены в виде нормализованных аудио-буферов в формате 32-битных чисел с плавающей запятой, организованных как моно- или стереоканальные массивы. Система использует кольцевые буферы для обработки в реальном времени и промежуточные спектральные данные, полученные через быстрое преобразование Фурье с применением оконной функции. Для хранения состояния эффектов используются специа-

лизированные структуры: коэффициенты БИХ-фильтров эквалайзера, параметры огибающей компрессора и линии задержки реверберации.

Выходные данные включают обработанные аудиофайлы в выбранных пользователем форматах (WAV с PCM-кодированием или MP3 с переменным битрейтом), сохраняющие исходные параметры частоты дискретизации или конвертируемые к стандартным значениям. Визуализационные данные содержат три основных компонента: осциллограмму последних обработанных сэмплов, частотный спектр с разрешением 8192 точек и амплитудно-частотную характеристику активных фильтров, отображаемую в логарифмическом масштабе от 20 Гц до 20 кГц.

Метаданные системы включают пользовательские пресеты эффектов, содержащие полный набор параметров обработки, историю операций для возможности отмены действий, а также технические лог-файлы с информацией о производительности и ошибках. Все данные передаются между компонентами системы через единый формат аудиоблоков, сопровождаемых метаинформацией о текущих настройках обработки, что обеспечивает согласованность работы многопоточной архитектуры.

### **3.7.1 Описание сущностей графического интерфейса**

Графический интерфейс Audio Processor представляет собой комплекс взаимосвязанных визуальных компонентов, объединенных в единое интуитивное пространство.

Интерфейс поддерживает несколько режимов отображения - компактный для мониторов с малым разрешением, расширенный с дополнительными инструментами анализа для профессиональной работы, и режим презентации с увеличенными элементами управления. Все визуальные компоненты реализованы с учетом эргономики - важные элементы выделены акцентным цветом, соблюдены принципы визуальной иерархии, обеспечена последовательная реакция на пользовательские действия. Особенностью интерфейса является синхронизация всех элементов - изменения параметров эффектов

мгновенно отражаются на графиках, а действия пользователя сопровождаются тактильной обратной связью в виде тонких анимационных эффектов.

### **3.7.2 Описание сущностей логики процессора**

Основу обработки составляет низкоуровневый модуль, работающий с потоком аудиосэмплов в формате 32-битных чисел с плавающей точкой, обеспечивающий базовые операции нормализации и передискретизации. Система эффектов построена вокруг трех ключевых процессоров: эквалайзер реализует трехполосную фильтрацию через каскад БИХ-фильтров с настраиваемыми частотами среза и коэффициентами усиления, компрессор отвечает за компрессию сигнала с алгоритмами расчета огибающей и адаптивными параметрами атаки/восстановления, а ревербератор генерирует реверберацию через комбинацию линий задержки с регулируемыми параметрами затухания.

Поток данных управляется специализированным менеджером маршрутизации, который обеспечивает передачу аудиоблоков между модулями с минимальной задержкой, используя кольцевые буферы и механизм синхронизации для многопоточной работы. Состояние системы отслеживает активные эффекты, параметры обработки и текущий режим работы (реальное время/оффлайн обработка), предоставляя единый интерфейс для управления конвейером обработки.

### **3.7.3 Описание сущностей системного уровня**

Системная архитектура приложения построена на нескольких ключевых компонентах, обеспечивающих интеграцию с операционной средой и аппаратными ресурсами. Центральным элементом выступает абстрактный слой взаимодействия с аудиодрайверами, реализующий поддержку ASIO, WASAPI и Core Audio через единый кроссплатформенный API. Для управления устройствами ввода-вывода используется DeviceManager, который автоматически обнаруживает доступные аудиоинтерфейсы, анализирует их ха-

рактеристики и предоставляет унифицированный интерфейс для работы с ними.

Файловая подсистема основана на компоненте, объединяющем возможности стандартных Python-библиотек для работы с файлами и мощь FFmpeg для обработки мультимедиа. Этот модуль включает кэширующий механизм для ускорения повторного доступа к аудиофайлам и систему контроля целостности данных.

Многопоточная архитектура координируется центральным диспетчером задач, который создает и управляет тремя основными типами потоков: высокоприоритетным аудиопотоком реального времени, фоновыми рабочими потоками для обработки эффектов и служебными потоками для визуализации и логирования.

### **3.8 Проектирования пользовательского интерфейса**

Центральное место занимает динамическая визуализация аудиопотока - двойной дисплей с синхронизированными осциллограммой и АЧХ, выполненный в темной цветовой гамме с акцентными элементами салатового цвета для выделения ключевых параметров сигнала. Основная рабочая область: верхняя треть экрана отведена под графики, центральная часть содержит компактные панели эффектов с интуитивными регуляторами, а нижний сектор занимает расширенная панель транспорта с профессиональными элементами управления.

Навигационная система реализована через адаптивную панель вкладок с контекстно-зависимыми элементами управления - при выборе конкретного эффекта (эквалайзер, компрессор, реверберация) нижняя панель автоматически наполняется соответствующими регуляторами, сохраняя при этом быстрый доступ к основным функциям. Все элементы управления спроектированы с учетом тактильного взаимодействия - ползунки имеют выраженные рифленые ручки с магнитными точками для часто используемых значений, кнопки обладают трехступенчатой визуальной обратной связью (покой, на-

ведение, нажатие), а переключатели сопровождаются мягкими анимационными переходами.

На рисунке 3.2 представлен интерфейс программы при запуске. Макет содержит следующие элементы:

1. Окно вывода графика формы волны.
2. Окно вывода АЧХ.
3. Активная кнопка для загрузки аудио.
4. Активная кнопка для воспроизведения аудио.
5. Не активная кнопка для остановки аудио.
6. Не активная кнопка для сохранения аудио.
7. Вкладка выбора эффекта эквалайзер (выбран).
8. Вкладка выбора эффекта компрессор (не выбран).
9. Вкладка выбора эффекта реверберации (не выбран).
10. Бегунок регулировки низких частот.
11. Бегунок регулировки средних частот.
12. Бегунок регулировки высоких частот.
13. Чекбокс включения эквалайзера.
14. Бегунок управления временем начала воспроизведения.
15. Время, в которое воспроизводится аудио и полная длина файла.

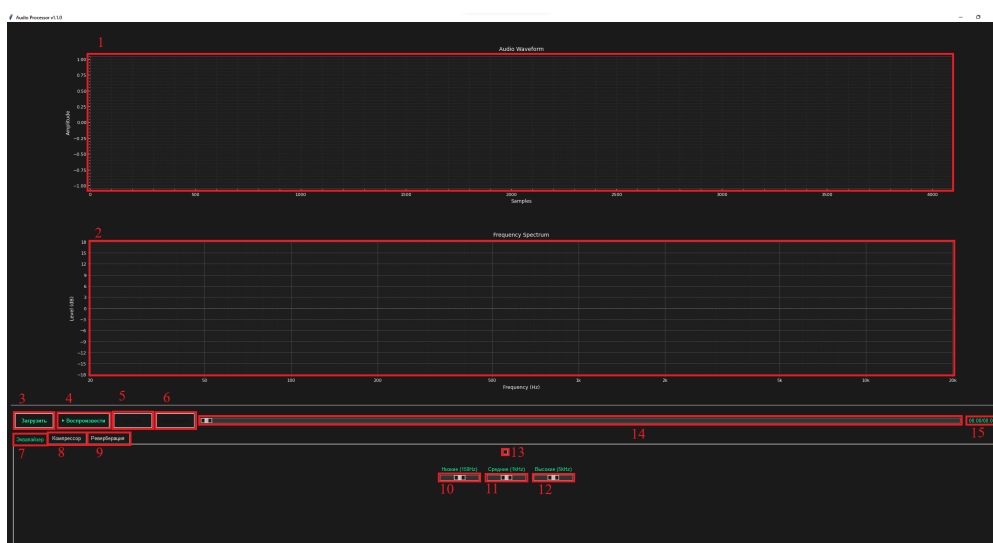


Рисунок 3.2 – Макет интерфейса программы

На рисунке 3.3 изображён макет вкладки компрессора. Макет содержит следующие элементы:

1. Бегунок настройки порогового значения.
2. Бегунок настройки параметров сжатия.
3. Бегунок настройки времени атаки в миллисекундах.
4. Бегунок настройки времени восстановления сигнала в миллисекундах.
5. Бегунок настройки усиления выходного сигнала.
6. Чекбокс включения эффекта.

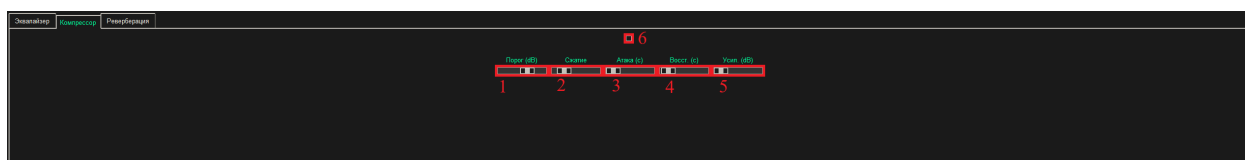


Рисунок 3.3 – Макет окна компрессора

На рисунке 3.4 изображён макет вкладки реверберации. Макет содержит следующие элементы:

1. Бегунок настройки уровня эффекта.
2. Бегунок настройки исходного звука.
3. Бегунок настройки размера пространства.
4. Бегунок настройки затухания.
5. Чекбокс включения эффекта.



Рисунок 3.4 – Макет окна реверберации

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Стоян Стефанов, JavaScript. Шаблоны / Стоян Стефанов – Москва: Символ-Плюс, 2011. – 262 с. – ISBN 978-5-93286-208-7 – Текст : непосредственный.
2. Васильев, В. И. Информационные системы / В. И. Васильев. – Москва: Финансы и статистика, 2009. – 416 с. – ISBN 978-5-279-03153-4. – Текст : непосредственный.
3. Гамзатов, М. Г. Информационные технологии / М. Г. Гамзатов, И. Г. Ахмедов. – Москва: Юрайт, 2018. – 392 с. – ISBN 978-5-534-05640-2. – Текст : непосредственный.
4. Голицына, О. Л. Информационные системы / О. Л. Голицына, Н. В. Максимов, И. И. Попов. – Москва: Форум, 2008. – 496 с. – ISBN 978-5-91134-245-4. – Текст : непосредственный.
5. Дейт, К. Введение в системы баз данных / К. Дейт. – Москва: Вильямс, 2006. – 1328 с. – ISBN 5-8459-0788-8. – Текст : непосредственный.
6. Информационные технологии: учебник для вузов / под ред. Н. В. Макаровой. – Москва: Финансы и статистика, 2005. – 768 с. – ISBN 5-279-02207-8. – Текст : непосредственный.
7. Кнут, Д. Искусство программирования / Д. Кнут. – Москва: Вильямс, 2000. – Т. 1. – 720 с. – ISBN 5-8459-0080-8. – Текст : непосредственный.
8. Кузнецов, С. В. Базы данных: учебник / С. В. Кузнецов. – Москва: Академия, 2007. – 496 с. – ISBN 978-5-7695-2477-0. – Текст : непосредственный.
9. Литвиненко, В. В. Разработка информационных систем / В. В. Литвиненко. – Москва: Юрайт, 2018. – 344 с. – ISBN 978-5-534-07421-5. – Текст : непосредственный.
10. Маклаков, С. В. BPWin и ERWin. CASE-средства разработки информационных систем / С. В. Маклаков. – Москва: ДИАЛОГ-МИФИ, 2000. – 304 с. – ISBN 5-86404-090-3. – Текст : непосредственный.

11. Титтел, Э. HTML5 и CSS3 для чайников / Э. Титтел, К. Минник. – Москва: Вильямс, 2016. – 400 с. – ISBN 978-1-118-65720-1. – Текст : непосредственный.
12. Новиков, Ф. А. Дискретная математика для программистов / Ф. А. Новиков. – Санкт-Петербург: Питер, 2000. – 304 с. – ISBN 5-272-00176-4. – Текст : непосредственный.
13. Робертсон, С. Осваиваем SQL за 24 часа / С. Робертсон. – Москва: Вильямс, 2002. – 272 с. – ISBN 5-8459-0315-7. – Текст : непосредственный.
14. Таненбаум, Э. Современные операционные системы / Э. Таненбаум. – Санкт-Петербург: Питер, 2002. – 1040 с. – ISBN 5-318-00299-4. – Текст : непосредственный.
15. Мартин, Д. Организация баз данных в вычислительных системах / Д. Мартин. – Москва: Мир, 1980. – 662 с. – Текст : непосредственный.
16. Электронный ресурс]. – URL: <http://www.website.com> (дата обращения: 15.05.2023). – Текст : электронный.
17. Иванов, И. И. Анализ предметной области [Электронный ресурс] / И. И. Иванов. – Режим доступа: URL: <http://www.example.com/ivanov> (дата обращения: 20.05.2023). – Текст : электронный.
18. Петров, П. П. Разработка информационной системы [Электронный ресурс] / П. П. Петров. – Режим доступа: URL: <http://www.example.com/petrov> (дата обращения: 25.05.2023). – Текст : электронный.
19. Сидоров, С. С. Тестирование программного обеспечения [Электронный ресурс] / С. С. Сидоров. – Режим доступа: URL: <http://www.example.com/sidorov> (дата обращения: 30.05.2023). – Текст : электронный.
20. Смирнов, А. А. Введение в базы данных [Электронный ресурс] / А. А. Смирнов. – Режим доступа: URL: <http://www.example.com/smirnov> (дата обращения: 05.06.2023). – Текст : электронный.