Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«Московский государственный технический университет  имени Н.Э. Баумана»***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ \_\_\_\_\_\_\_Информатика и системы управления\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_Информационные системы и телекоммуникации\_\_\_\_\_\_\_\_\_\_

**РАСЧЁТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовой работе на тему:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Программное устройство обработки аудио сигнала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент Группа ИУ3-63  Н.Е. Данилов

(Подпись, дата) (И.О.Фамилия)

Руководитель курсовой работы  В.М. Недашковский

(Подпись, дата) (И.О.Фамилия)

Москва, 2016

**Содержание**

[Задание 3](#_Toc450729555)

[1. Теоретическая часть 4](#_Toc450729556)

[1.1. Основные понятия о цифровом звуке 4](#_Toc450729557)

[1.2. Формат WAV 11](#_Toc450729558)

[1.3. Работа с WAV-файлами 14](#_Toc450729559)

[2. Звуковые эффекты 17](#_Toc450729560)

[2.1. Обработка сигнала 17](#_Toc450729561)

[2.2. Эффект Echo 19](#_Toc450729562)

[2.3. Эффект Overdrive 20](#_Toc450729563)

[3. Эквалайзер 22](#_Toc450729564)

[4. Интерфейс эквалайзера в среде Java 28](#_Toc450729565)

[5. Схемы работы программы 29](#_Toc450729566)

[6. Листинг программы 31](#_Toc450729567)

[7. Выводы 47](#_Toc450729568)

[8. Литература 48](#_Toc450729569)

# Задание

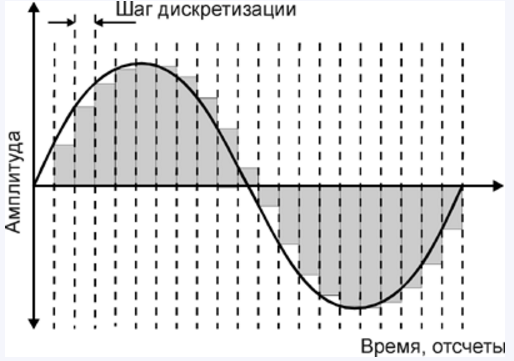
Разработать программу проигрывателя \*.wav файлов на любом выбранном языке с поддержкой заданной цепи из двух эффектов (эффект 1 и эффект 2) и графическим N-полосовым эквалайзером. Эффекты (эффект 1 – «Echo», эффект 2 – «Overdrive»), число полос эквалайзера (N = 6):

* Количество полос, N = 6 шт;
* Диапазон частот D = 1 Гц – 44,1КГц;
* Частота дискретизации – 44,1КГц
* Тип используемого фильтра – КИХ, Оконный, Окно Хэмминга;
* Метод проектирования, см. таблицу;
* Ослабление сигнала в полосе заграждения, As = -40дБ;
* Apass = 3 ±Δ дБ;
* Графический интерфейс;
* Открытие и воспроизведение \*.wav файлов;
* Регулирование громкости воспроизводимого звука;
* Включение и отключение эффектов эквалайзера;
* Регулирование параметров эффектов;
* Возможность одновременной работы эффектов и эквалайзера.

# Теоретическая часть

## Основные понятия о цифровом звуке

В соответствии с теорией математика Фурье, звуковую волну можно представить в виде спектра входящих в нее частот (рис. 1).



*Рис.1. Спектр частот звуковой волны.*

Частотные составляющие спектра - это синусоидальные колебания (так называемые чистые тона), каждое из которых имеет свою собственную амплитуду и частоту. Таким образом, любое, даже самое сложное по форме колебание (например, человеческий голос), можно представить суммой простейших синусоидальных колебании определенных частот и амплитуд. И наоборот, сгенерировав различные колебания и наложив их друг на друга (смикшировав, смешав), можно получить различные звуки.

**Справка:** человеческий слуховой аппарат/мозг способен различать частотные составляющие звука в пределах от 20 Гц до ~20 КГц (верхняя граница может колебаться в зависимости от возраста и других факторов). Кроме того, нижняя граница сильно колеблется в зависимости от интенсивности звучания.

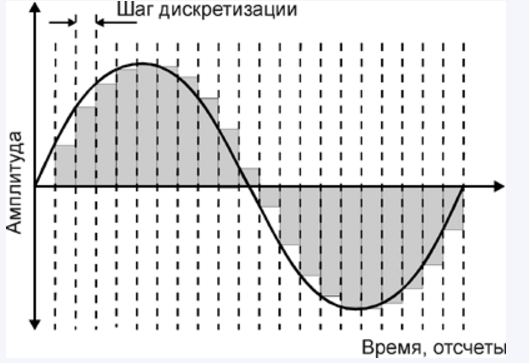
«Обычный» аналоговый звук представляется в аналоговой аппаратуре непрерывным электрическим сигналом. Компьютер оперирует с данными в цифровом виде. Это означает, что и звук в компьютере представляется в цифровом виде.

Дадим несколько определений цифровому звуку:

1. Цифровой звук – это способ представления электрического сигнала посредством дискретных численных значений его амплитуды.

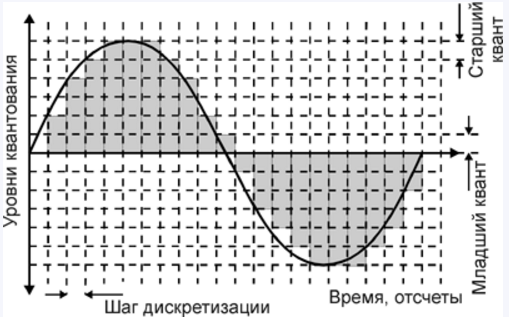
2. Цифровой звук – результат преобразования аналогового сигнала звукового диапазона в цифровой аудиоформат.

Допустим, мы имеем аналоговую звуковую дорожку хорошего качества (нешумная запись, содержащая спектральные составляющие из всего слышимого диапазона частот – приблизительно от 20 Гц до 20 КГц) и хотим оцифровать ее без потери качества. Как этого добиться и как происходит оцифровка? Звуковая волна – это некая сложная функция, зависимость амплитуды звуковой волны от времени. Казалось бы, что раз это функция, то можно записать ее в компьютер «как есть», то есть описать математический вид функции и сохранить в памяти компьютера. Однако практически это невозможно, поскольку звуковые колебания нельзя представить аналитической формулой (как y=x2, например). Остается один путь – описать функцию путем хранения ее дискретных значений в определенных точках. Иными словами, в каждой точке времени можно измерить значение амплитуды сигнала и записать в виде чисел. Однако и в этом методе есть свои недостатки, так как значения амплитуды сигнала мы не можем записывать с бесконечной точностью, и вынуждены их округлять. Говоря иначе, мы будем приближать эту функцию по двум координатным осям – амплитудной и временной. Таким образом, оцифровка сигнала включает в себя два процесса - процесс дискретизации (осуществление выборки) и процесс квантования. Процесс дискретизации - это процесс получения значений величин преобразуемого сигнала в определенные промежутки времени (рис. 2).



*Рис.2. Процесс дискретизации.*

Квантование - процесс замены реальных значений сигнала приближенными с определенной точностью (рис. 3). Таким образом, оцифровка – это фиксация амплитуды сигнала через определенные промежутки времени и регистрация полученных значений амплитуды в виде округленных цифровых значений. Записанные значения амплитуды сигнала называются отсчетами. Очевидно, что чем чаще мы будем делать замеры амплитуды (чем выше частота дискретизации) и чем меньше мы будем округлять полученные значения (чем больше уровней квантования), тем более точное представление сигнала в цифровой форме мы получим. Оцифрованный сигнал в виде набора последовательных значений амплитуды можно сохранить.

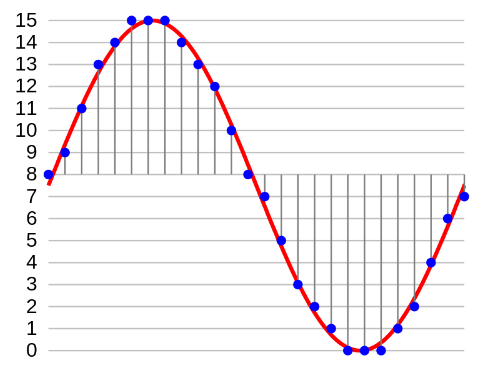


*Рис.3. Процесс квантования.*

Цифровая звукозапись — технология преобразования аналогового звука в цифровой с целью сохранения его на физическом носителе для возможности последующего воспроизведения записанного сигнала.

Представление аудиоданных в цифровом виде, позволяет очень эффективно изменять исходный материал при помощи специальных устройств или компьютерных программ - звуковых редакторов, что нашло широкое применение в промышленности, медиа-индустрии и быту.

Простейший метод преобразования, Импульсно-кодовая модуляция (ИКМ), состоит в представлении последовательности мгновенных значений уровня сигнала, измеряемого аналого-цифровым преобразователем (АЦП) через равные промежутки времени. График, изображенный на рис. 4, изображает кодово-импульсную модуляцию – аналого-цифровое преобразование, используемое в аудиоформатах AIFF и WAV. Красная линия – это оригинальный аналоговый сигнал, который постоянно меняет свою амплитуду и поступает по кабелю от микрофона.

**

*Рис.4. График ИКМ.*

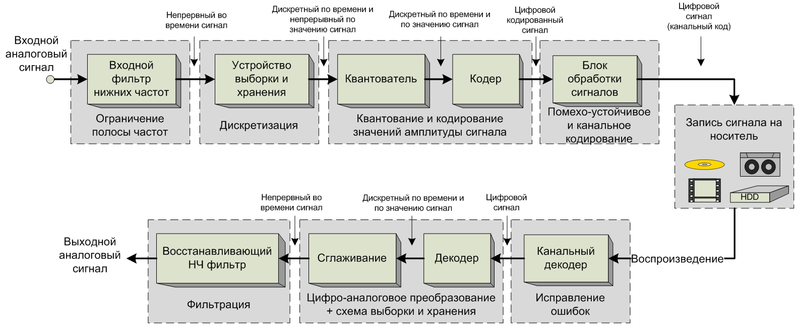
Компьютер считывает уровень напряжения через постоянные временные интервалы, которые изображены на графике как вертикальные линии. Синие точки показывают считанный компьютером уровень напряжения в данный момент. Горизонтальные линии отображают другие возможные значения, которые компьютер может сохранить и считать; из всех этих возможных значений он всегда выбирает наиболее близкое к действительному. Аудиофайлы форматов AIFF и WAV представляют собой длинный список чисел, которые являются значениями уровня напряжения. Чем чаще компьютер считывает показания, и чем точнее оказывается каждое из них, тем качественнее будет звучать цифровая запись. Частота, с которой компьютер считывает показания, называется частотой дискретизации, а точность, с которой он это производит – квантованием. **[3]**

Разновидностью ИКМ является дельта-модуляция, где в каждый момент отсчёта сигнал сравнивается с пилообразным напряжением на каждом шаге дискретизации. Сигма-дельта модуляция — способ представления сигнала на основе принципа избыточной дискретизации и формирования шума квантования позволяет снизить уровень шума.

Современные методы используют более сложные алгоритмы преобразования. Помимо представления звуковых колебаний в цифровом виде, применяется также создание специальных команд для автоматического воспроизведения на различных электронных музыкальных инструментах, ярчайшим примером такой технологии является MIDI.

Для воспроизведения цифрового звука применяют специальное оборудование, например, музыкальные центры, цифровые плееры, компьютеры с звуковой картой и установленным программным обеспечением аудиоплеером или медиаплеером.

На Рис. 5 представлена схема записи и воспроизведения цифровых файлов.

[](http://commons.wikimedia.org/wiki/File:Digital_sound_recording_scheme.png?uselang=ru)

*Рис.5. Структурная схема цифровой звукозаписи и воспроизведения*

Принцип цифрового представления колебаний звукозаписи достаточно прост:

* вначале нужно преобразовать аналоговый сигнал в цифровой, это осуществляет устройство — аналого-цифровой преобразователь (АЦП);
* произвести сохранение полученных цифровых данных на носитель: магнитную ленту (DAT), жёсткий диск, оптический диск или флеш-память;
* для того чтобы прослушать сделанную запись, необходимо воспроизведение сделанной записи с носителя и обратное преобразование из цифрового сигнала в аналоговый, с помощью цифро-аналогового преобразователя (ЦАП);

Принцип действия АЦП тоже достаточно прост: аналоговый сигнал, полученный от микрофонов и электромузыкальных инструментов, преобразовывается в цифровой. Это преобразование включает в себя следующие операции:

1. Ограничение полосы частот производится при помощи фильтра нижних частот для подавления спектральных компонент, частота которых превышает половину частоты дискретизации.
2. Дискретизацию во времени, то есть замену непрерывного аналогового сигнала последовательностью его значений в дискретные моменты времени — отсчетов. Эта задача решается путём использования специальной схемы на входе АЦП — устройства выборки-хранения.
3. Квантование по уровню представляет собой замену величины отсчета сигнала ближайшим значением из набора фиксированных величин — уровней квантования.
4. Кодирование или оцифровку, в результате которого значение каждого квантованного отсчета представляется в виде числа, соответствующего порядковому номеру уровня квантования.

Делается это следующим образом: непрерывный аналоговый сигнал «режется» на участки, с частотой дискретизации, получается цифровой дискретный сигнал, который проходит процесс квантования с определенной разрядностью, а затем кодируется, то есть заменяется последовательностью кодовых символов. Для качественной записи звука в полосе частот 20-20 000 Гц применяется минимальная стандартная частота дискретизации от 44,1 кГц и выше (в настоящее время появились АЦП и ЦАП c частотой дискретизации 192,3 и даже 384,6 кГц). Для получения довольно качественной записи достаточно разрядности 16 бит, однако для расширения динамического диапазона и повышения качества звукозаписи используется разрядность 24 (реже 32) бита.

Цифровой сигнал, полученный с декодера, преобразовывается в аналоговый. Это преобразование происходит следующим образом:

1. Декодер ЦАП преобразует последовательность чисел в дискретный квантованный сигнал
2. Путем сглаживания во временной области из дискретных отсчетов вырабатывается непрерывный во времени сигнал
3. Окончательное восстановление сигнала производится путем подавления побочных спектров в аналоговом фильтре нижних частот

На цифровых носителях и в персональных компьютерах для хранения звука (музыки, голоса и т. п.) применяются различные форматы, позволяющие выбрать приемлемое соотношение сжатия, качества звука и объёма данных. Популярные форматы файлов для персональных компьютеров и соответствующих устройств: **[4]**

* OGG
* MP3
* WAV
* WMA

В рамках данной курсовой работы разрабатывается эквалайзер для WAV-файлов, поэтому рассмотрим структуру этого формата.

## Формат WAV

**WAVE** или **WAV** является короткой формой **Wave Audio File Format** (реже именуемой как Аудио для Windows). Этот формат является стандартом для хранения аудио потока на ПК. Он является сферой приложения формата RIFF для хранения аудио в «цепочках». Он напоминает форматы 8SVX и AIFF, используемые компьютерами Amiga и Macintosh соответственно. Это также основной формат на системах Windows для хранения обычного несжатого звука. Как правило, для этого применяется кодирование методом линейной импульсно-кодовой модуляции. Однако контейнер не налагает каких-либо ограничений на используемый алгоритм кодирования.

WAV файл представляет собой две, четко делящиеся, области. Одна из них – заголовок файла, другая – область данных. В заголовке файла хранится информация о:

* Размере файла.
* Количестве каналов.
* Частоте дискретизации.
* Количестве бит в сэмпле (глубина звучания).

Для большего понимания смысла величин в заголовке следует еще рассказать об области данных и оцифровке звука. Звук состоит из колебаний, которые при оцифровке приобретают ступенчатый вид. Этот вид обусловлен тем, что компьютер может воспроизводить в любой короткий промежуток времени звук определенной амплитуды (громкости) и этот короткий момент далеко не бесконечно короткий. Продолжительность этого промежутка и определяет частота дискретизации. Например, у нас файл с частотой дискретизации 44.1 kHz, это значит, что тот короткий промежуток времени равен 1/44100 секунды (следует из размерности величины Гц = 1/с). Современные звуковые карты поддерживают частоту дискретизации до 192 kHz.

От амплитуды зависит точность звука. Амплитуда выражается числом, занимаемым в памяти (файле) 8, 16, 24, 32 бит. Как известно, 8 бит = 1 байту, следовательно, какая-то одна амплитуда в какой-то короткий промежуток времени в памяти (файле) может занимать 1, 2, 3, 4 байта соответственно. Таким образом, чем больше число занимает места в памяти (файле), тем больше диапазон значений для этого числа, а значит и для амплитуды.

* 1 байт – 0..255
* 2 байта – 0..65 535
* 3 байта – 0..16 777 216
* 4 байта – 0..4 294 967 296

В моно варианте значения амплитуды расположены последовательно. В стерео сначала идет значение амплитуды для левого канала, затем для правого, затем снова для левого и так далее. Совокупность амплитуды и короткого промежутка времени носит название сэмпл.

| **Местоположение** | **Поле** | **Описание** |
| --- | --- | --- |
| 0..3 (4 байта) | chunkId | Содержит символы “RIFF” в ASCII кодировке (0x52494646 в big-endian представлении). Является началом RIFF-цепочки. |
| 4..7 (4 байта) | chunkSize | Это оставшийся размер цепочки, начиная с этой позиции. Иначе говоря, это размер файла – 8, то есть, исключены поля chunkId и chunkSize. |
| 8..11 (4 байта) | format | Содержит символы “WAVE” (0x57415645 в big-endian представлении) |
| 12..15 (4 байта) | subchunk1Id | Содержит символы “fmt “ (0x666d7420 в big-endian представлении) |
| 16..19 (4 байта) | subchunk1Size | 16 для формата [**PCM**](http://audiocoding.ru/formats/wav/). Это оставшийся размер подцепочки, начиная с этой позиции. |
| 20..21 (2 байта) | audioFormat | Аудио формат, полный список можно получить [здесь](http://audiocoding.ru/assets/meta/2008-05-22-wav-file-structure/wav_formats.txt). Для [**PCM**](http://audiocoding.ru/formats/wav/) = 1 (то есть, Линейное квантование). Значения, отличающиеся от 1, обозначают некоторый формат сжатия. |
| 22..23 (2 байта) | numChannels | Количество каналов. Моно = 1, Стерео = 2 и т.д. |
| 24..27 (4 байта) | sampleRate | Частота дискретизации. 8000 Гц, 44100 Гц и т.д. |
| 28..31 (4 байта) | byteRate | Количество байт, переданных за секунду воспроизведения. |
| 32..33 (2 байта) | blockAlign | Количество байт для одного сэмпла, включая все каналы. |
| 34..35 (2 байта) | bitsPerSample | Количество бит в сэмпле. Так называемая “глубина” или точность звучания. 8 бит, 16 бит и т.д. |
| 36..39 (4 байта) | subchunk2Id | Содержит символы “data” (0x64617461 в big-endian представлении) |
| 40..43 (4 байта) | subchunk2Size | Количество байт в области данных. |
| 44.. | data | Непосредственно WAV-данные. |

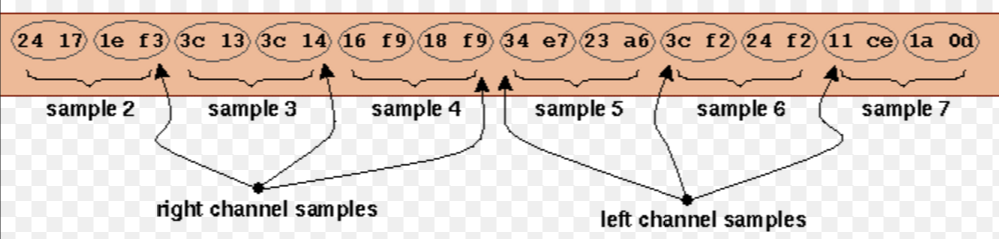
*Таблица 1. Структура WAV-файла.* ***[5]***

**WAV** файлы достаточно большие, что делает этот формат неудобным для обмена по сети Интернет, и это сильно подрывает его популярность. Однако, этот формат – как правило, чаще всего используется для сохранения первозданного вида для файлов высокого качества в таких случаях, где размер свободного дискового пространства не является ограничением. Он также используется в программах для редактирования аудио, где экономят время на сжатии и распаковке данных.

## Работа с WAV-файлами

Метод работы с файлом зависит от того, сколько каналов имеет аудио. В секции «data» хранятся непосредственно сами аудио данные. Если аудио запись имеет только 1 канал (моно), то аудио семплы записаны последовательно. Если же аудио запись имеет 2 канала (стерео), то семплы каналов записываются поочередно. Это позволяет проигрывать файл, не загружая его до конца в оперативную память, либо позволяет прослушивать поврежденные wav файлы. Кроме того, это удобно тогда, когда проигрывается большой файл, или, когда идет передача в последовательном потоке, например, через сетевое соединение.

Аудиовыборки многоканального цифрового аудио сохраняются как чередуемые данные, которые просто означают последовательные аудиовыборки нескольких каналов. Выборки каналов сохранены последовательно друг за другом, перед тем как произойдет переход к следующему времени выборки. Это сделано с целью возможности последовательного проигрывания файла даже тогда, когда еще не весь файл прочитан целиком. Это удобно, когда проигрывается большой файл с диска (который не может быть размещен целиком в памяти) или файл передается в последовательном потоке данных через сетевое соединение. На рис. 6 показано как хранятся данные двухканального звука.

**

*Рис.6. Хранение данных двухканального звука.*

При чтении файла, реализация, возможно, будет обеспечивать побайтное чтение в буфер. Нужно помнить, что элементы такого буфера не будут являться отсчетами, т.к. размер отсчета («битность») у разных файлов разная. В 16-битных wav-файлах на один отсчет одного канала приходится два расположенных друг за другом байта, в 24-битных – 3 байта.

Рассмотрим вариант 16-битного файла. На один отсчет приходится 2 байта. Для обеспечения наиболее качественного звука было бы правильно реализовать структуру, содержащую все 16 бит и основные арифметические операции для нее. Поэтому в данной курсовой работе для облегчения и уменьшения объема работы будем использовать тип short int, размер которого как раз и равен 16 разрядам.

При преобразовании всего буфера, содержащего последовательность байтов, в буфер, содержащий отсчеты типа short int, а затем обратного преобразования никаких изменений в качестве звука при проигрывании не произойдет. Это значит, что данный тип можно использовать для обработки, но следует внимательно следить за значениями буфера. Если вы попытаетесь умножить значение выборки на слишком большой коэффициент, результат может превысить 2 байта. На практике вы будете слышать помехи, щелчки и т.д. Эти явления допускаются, но для их минимизации следует проверять результат и при превышении 2-х байт уменьшать либо вообще не умножать отсчет. **[5]**

# Звуковые эффекты

## Обработка сигнала

Звук обладает следующими характеристиками:

• Амплитуда, громкость, динамический диапазон

• Частота, частотный диапазон

• Фаза

К преобразованию звука прибегают в основном с целью изменения каких-то характеристик звука. Кроме того, на основе описанных ниже преобразований базируются механизмы создания различных звуковых эффектов, а также способы очистки звука от нежелательных шумов, изменения тембра и т.п. Все эти преобразования сводятся, в конечном счете, к нижеследующим:

**Амплитудные преобразования**. Выполняются над амплитудой сигнала. Такую процедуру можно проделать двумя способами: либо умножая амплитуду сигнала на некоторое фиксированное число, в результате чего получится одинаковое изменение интенсивности сигнала на всей его протяженности, то есть усиление или ослабление, либо изменяя амплитуду сигнала по какому-то закону, то есть умножая амплитуду сигнала на модулирующую функцию. Последний процесс называется амплитудной модуляцией.

**Спектральные (частотные) преобразования**. Такие преобразования выполняются над частотными составляющими звука. Фактически сигнал представляется рядом Фурье, то есть раскладывается на простейшие синусоидальные колебания различных частот и амплитуд. Затем производится обработка необходимых частотных составляющих (например, фильтрация) и обратная свертка. В отличие от амплитудных преобразований, эта процедура значительно более сложная в исполнении, так как сам процесс разложения звука на простейшие синусоидальные колебания очень трудоемок.

**Фазовые преобразования**. Выполняются либо путем постоянного сдвига фазы сигнала, либо путем наложения некоторой фазомодулирующей функции. Такие преобразования, например, стереосигнала, позволяют реализовать эффект вращения или "объёмности" звука.

**Временные преобразования**. Реализуются путем наложения на сигнал одной или нескольких его копий, сдвинутых во времени. Позволяют создать эффекты эха или хора. Кроме того, временные преобразования могут влиять на пространственные характеристики звука.

**Формантные преобразования**. Выполняются над формантами - усиленными участками спектра звука. Применительно к звуку, сформированному речевым аппаратом человека, изменяя параметры формант, фактически можно изменять восприятие тембра и высоты голоса.

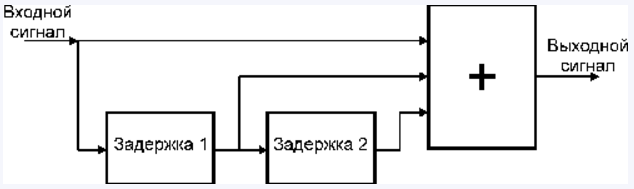
Отдельно необходимо обсудить **фильтрацию звука**, так как она тоже является одним из способов преобразования звука. К фильтрации прибегают в случаях, когда необходимо ограничить или изменить спектр звукового сигнала в каком-то определенном частотном диапазоне. Путем фильтрации звука, можно избавиться, например, от нежелательных шумов или помех, подавить определенные частотные полосы. Существует и еще один немаловажный аспект применения фильтрации. Часто устройства, с помощью которых производится запись и преобразования звуковых сигналов, имеют нелинейную зависимость амплитуды от частоты сигнала. Это означает, что при записи одни частотные составляющие звука могут быть завышены, а другие занижены. Фильтрация позволяет нормализовать частотные составляющие в необходимом диапазоне.

**Фильтрация** – процесс, в результате которого происходит усиление или ослабление отдельных частотных составляющих спектра, полное подавление частотных составляющих в определенной полосе частот. **[3]**

## Эффект Echo

**Echo -** эффект задержки, применяется чаще в случаях, когда моносигнал требуется преобразовать в нечто вроде псевдостерео. Если моносигнал подать в оба канала стереофонической акустической системы, то путем некоторой задержки сигнала в одном из каналов можно добиться получения стереоэффекта. Если же в оба канала сигнал приходит одновременно, то слушателю будет казаться, что источник звука расположен посредине. Меняя задержку сигнала в одном из каналов в пределах 8 мс можно получить эффект перемещения источника звука по стерео панораме.

Эффект реализуется добавлением к исходному сигналу его копии или нескольких копий, задержанных по времени.



*Рис.7. Принцип работы эффекта echo.*

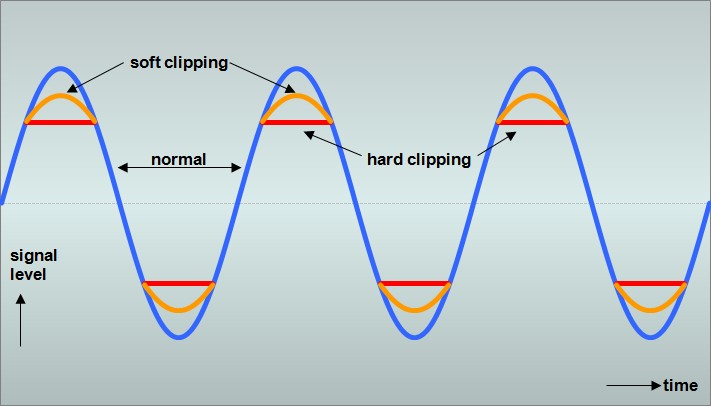
## Эффект Overdrive

Overdriveэффект продуцирует искаженные звуки, может имитировать голос роботов или радиотелефонные сигналы.

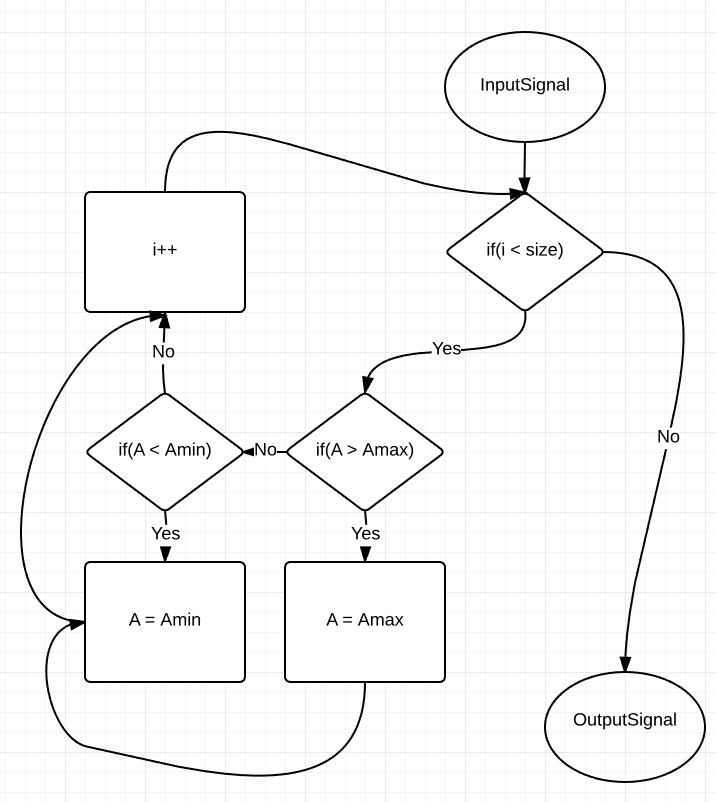
Если говорить о природе происхождения, то овердрайв и [дисторшн](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D1%81%D1%82%D0%BE%D1%80%D1%88%D0%BD" \o "Дисторшн) работают по одному физическому принципу — ограничение сигнала по амплитуде. В овердрайве это ограничение «мягкое», то есть верхушки [синусоиды](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BD%D1%83%D1%81%D0%BE%D0%B8%D0%B4%D0%B0) обрезаются не ровной линией, а плавными скруглениями. В дисторшне ограничение «жёсткое», то есть верхушки синусоиды просто ровно обрезаются.

Из-за «мягкого» ограничения выходной сигнал начинает искажаться пропорционально уровню входного сигнала. Таким образом, при использовании овердрайва для обработки гитарного сигнала можно подчеркнуть динамику звучания. В зависимости от силы удара по струнам будет меняться искажение гитарного сигнала, что кардинально отличается от дисторшна. Дисторшн искажает входной сигнал независимо от его уровня (амплитуды).

Если амплитуда входного сигнала больше максимальной или минимальной заданной амплитуды, то входной сигнал обрезается до уровня максимальной и минимальной из заданных амплитуд соответственно. **[3]**



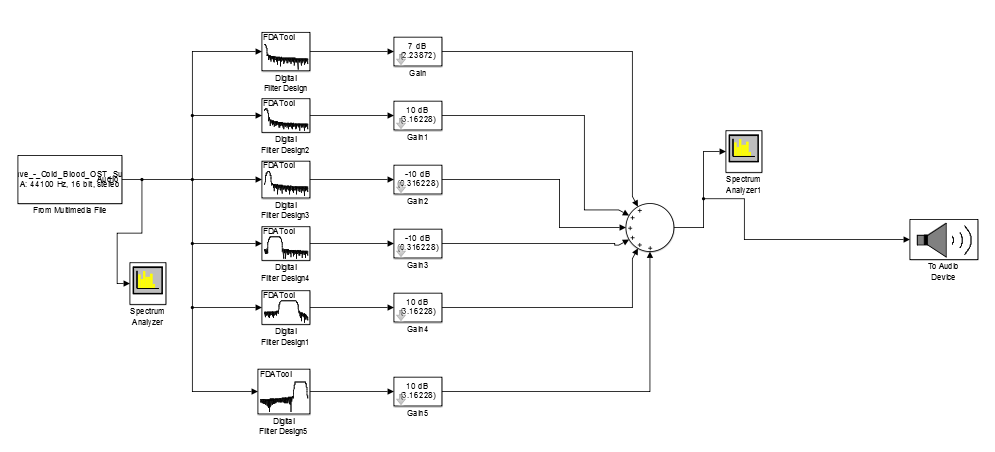
*Рис.8. Способы обрезания амплитуды.*



*Рис.9. Блок-схема вычисления эффекта Overdrive.*

# Эквалайзер

Проектирование цифровых фильтров, параметры которых необходимы для реализации эквалайзера, осуществляется в пакете MatLab2014.



* Типы фильтров и их параметров выбирается в соответствии с требованиями ТЗ: КИХ-фильтр, метод окон, окно Хэмминга;
* Ослабление сигнала в полосе задерживания не менее 40 дБ;
* Порядок фильтров 120.

В соответствии с количеством полос N = 6 разделим весь диапазон частот на полосы:

|  |  |  |
| --- | --- | --- |
| **№ полосы** | **Частоты (Hz)** | **Тип фильтра** |
| 1 | < 450 | ФНЧ |
| 2 | 400 – 1200 | ПФ |
| 3 | 1100 – 2600 | ПФ |
| 4 | 2400 – 8200 | ПФ |
| 5 | 8000 – 16400 | ПФ |
| 6 | 16000 - 21000 | ПФ |

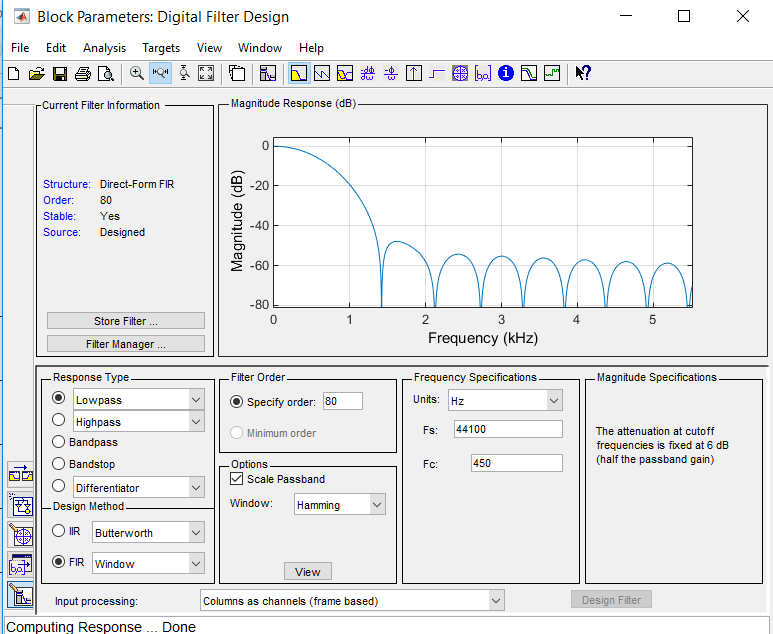
Для определения конечных параметров проектируемых фильтров была проанализирована работа эквалайзера на фильтрах с различными значениями порядков. По результатам опыта было определено, что, начиная с фильтров порядка более 80, эквалайзер обрабатывает полосы частот нужным образом.

***Исследование точности коэффициентов фильтров***

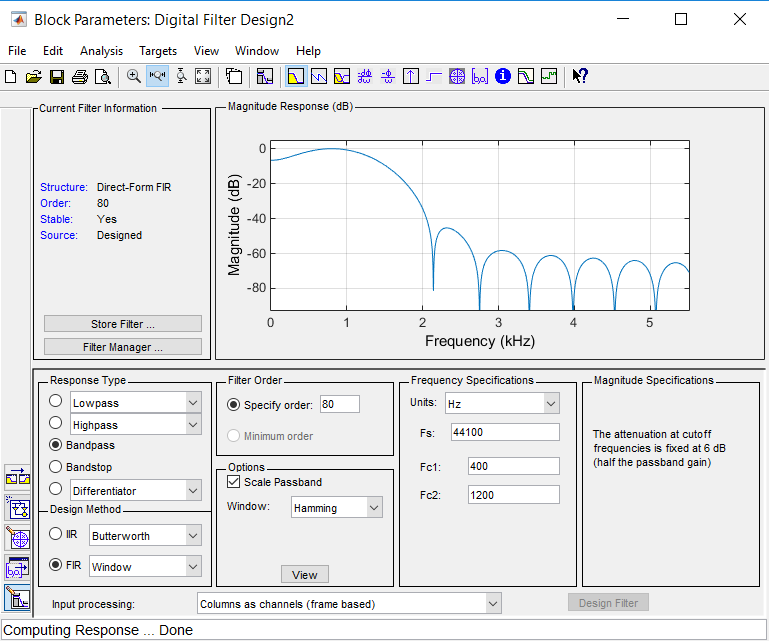
В рамках выполнения лабораторных работ по дисциплине микропроцессорная техника и теория цифровых автоматов была изучена возможность выбора точности представления проектируемых фильтров в пакете MatLab2014. Эта функция осуществляется через квантование коэффициентов и представление их как чисел с плавающей запятой. Данное преобразование позволяет добиться необходимой точности, в отличие от стандартного типа представления в double.

В рамках исследования было проанализировано необходимое и достаточное количество значащих символов для представления коэффициентов фильтров. При проектировании фильтров были практически опробованы значения от 13 до 25 значащих символов в представлении коэффициентов и выбрано то значение, которое давало достаточное качество изменения звука, заполнение памяти и компиляцию программного приложения. Данное значение по результатам исследований для заданного варианта равно 20 значащих знаков.

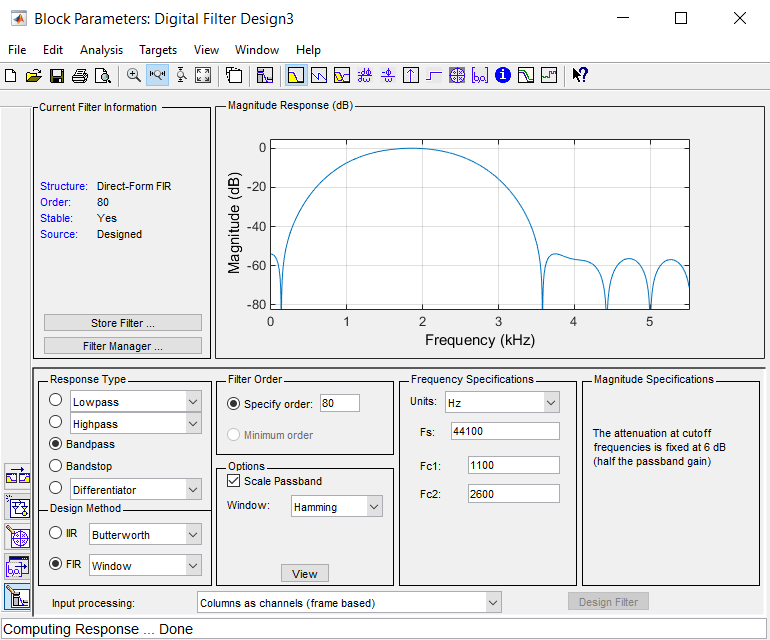
Параметры используемых фильтров представлены следующим образом:



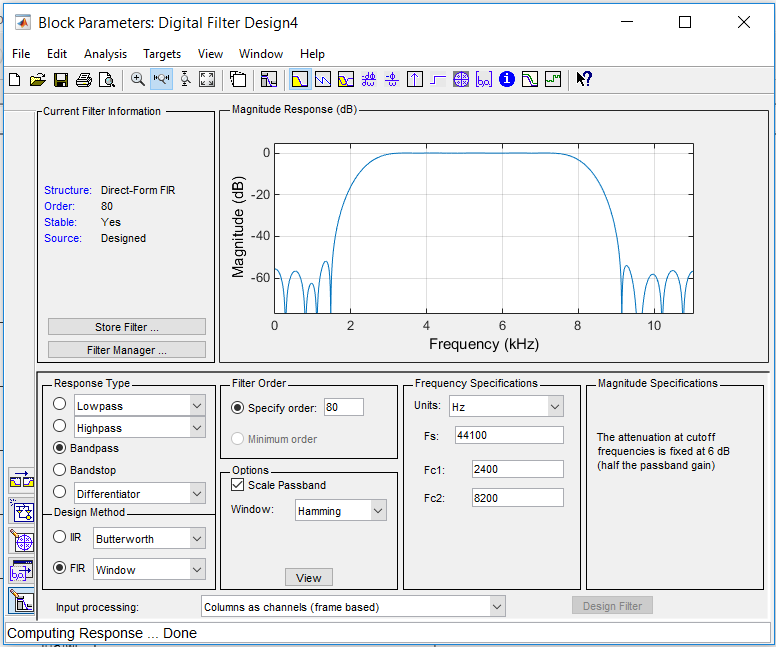
*Рис.10. Проектирование ФНЧ.*



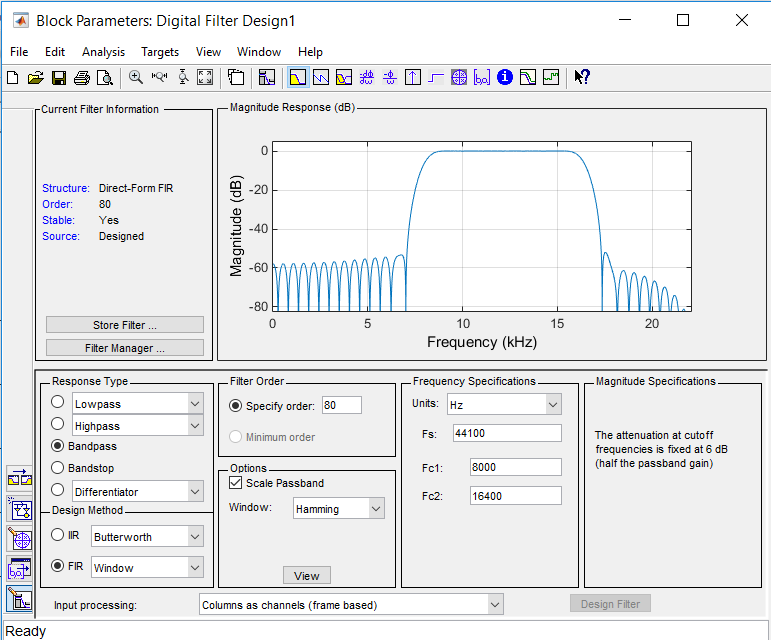
*Рис.11. Проектирование ПФ №1.*



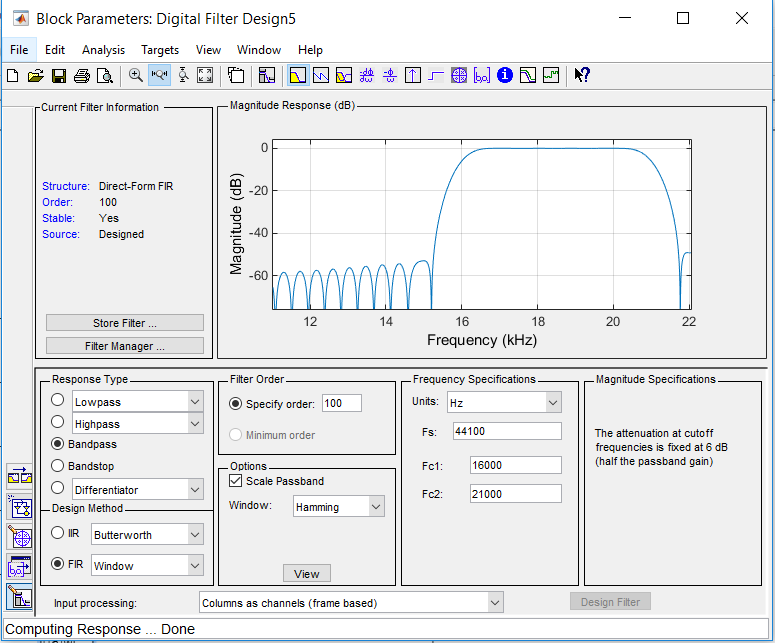
*Рис.12. Проектирование ПФ №2.*



*Рис.13. Проектирование ПФ №3.*

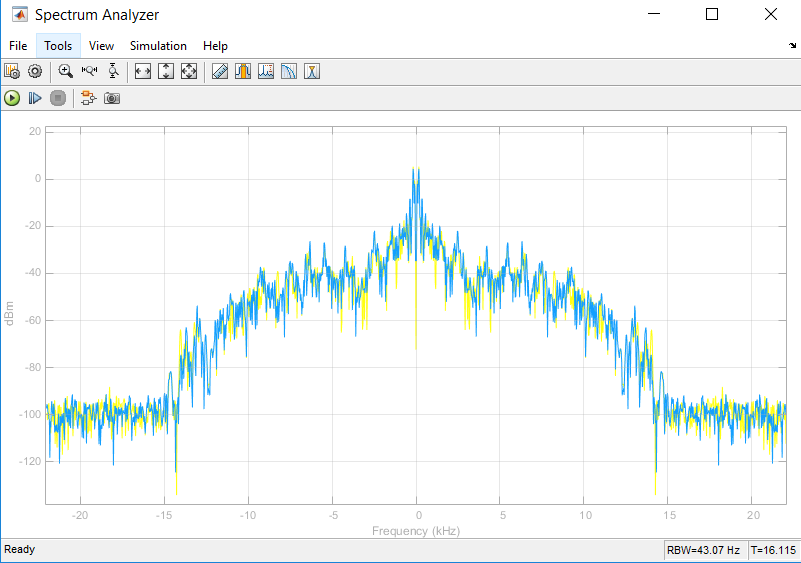


*Рис.14. Проектирование ПФ №4.*

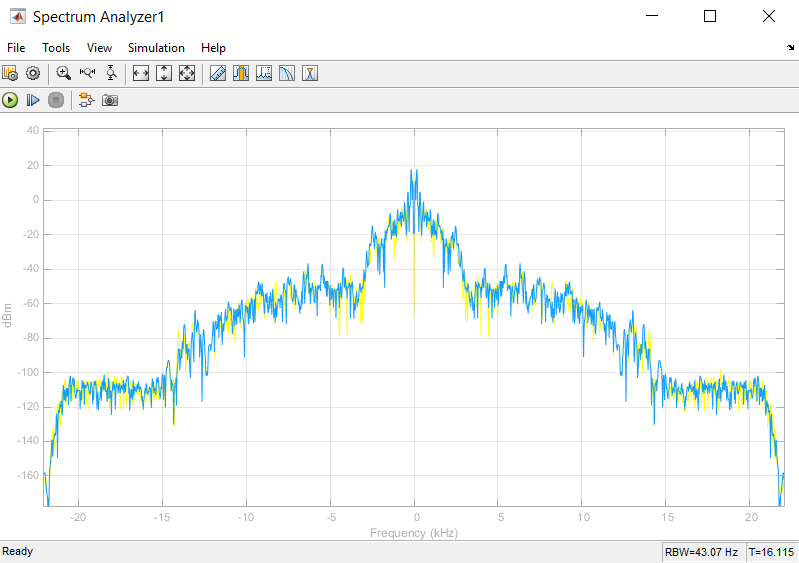


*Рис.15. Проектирование ПФ №5.*

Результат работы разработанного приложения представлен на рис.16 а,б при условии повышения частот для трех нижних полос и уменьшения для трех верхних. В следующем разделе будет представлен график работы приложения при таких же условиях, но реализованный на языке Java.



*Рис.16а. График исходного сигнала.*



*Рис.16б. График измененного сигнала.*

# Интерфейс эквалайзера в среде Java

*Рис.17. Графический интерфейс эквалайзера.*

Сравнение графика на рис.17 с соответствующими графиками на рис.16а, б позволяет сделать вывод о корректности полученных результатов и успешной разработке приложений аудио-плееров в двух средах программирования.

# Схемы работы программы

C:\Users\Vilkova\Downloads\Untitled Diagram (2).png

ECHO

*Рис.18а. Функциональная схема работы программы.*

Для того, чтобы выполнить все вычисления, производимые в данной программе в одном потоке, необходима достаточная высокая мощность процессора из-за большой нагрузки. Таким образом, было решено разбить вычисления, производимые в классе эквалайзера, на потоки. Распределение данных и задач по потокам представлено на рис. 18б.

C:\Users\Vilkova\Downloads\Untitled Diagram (1).png

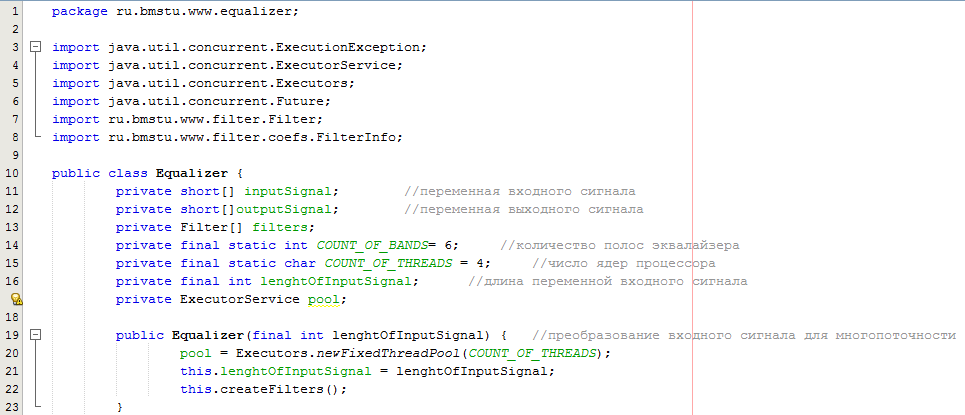
*Рис.18б. Принципиальная схема работы программы.*

# Листинг программы

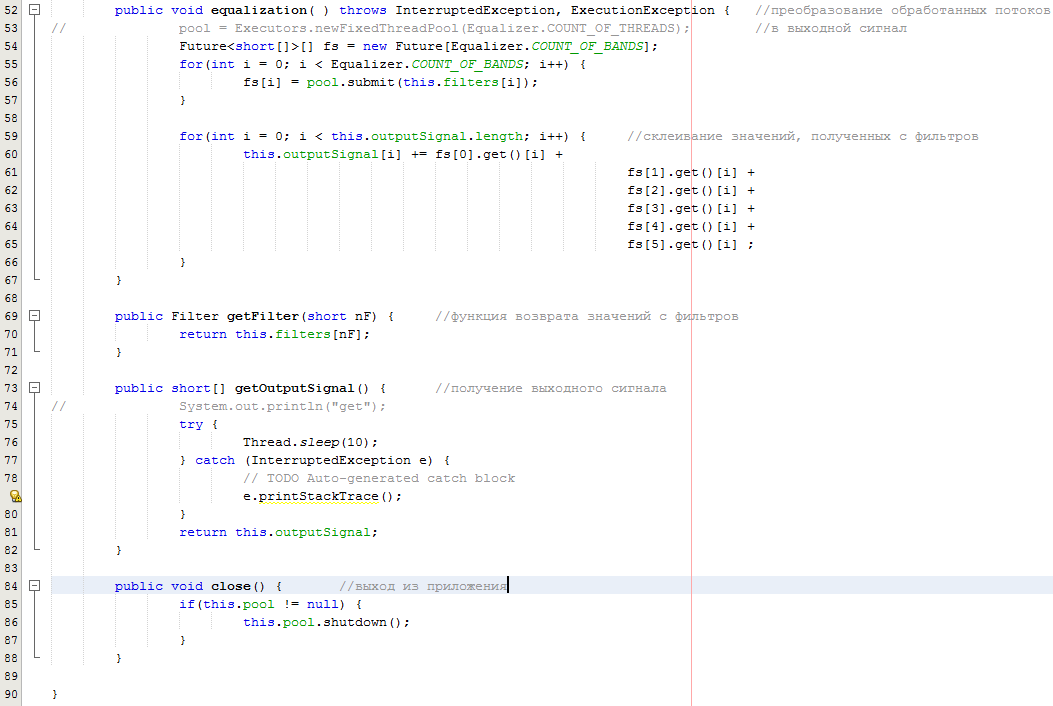
Для написания программы был выбран язык Java. Удобство использования данного языка заключается в возможности полного переноса программного продукта на все машины, где установлена JRE не ниже версии 1.8.0. При написании программы использовались следующие библиотеки: JavaFX 2.0, java.sound.\*. При написании программы использовался паттерн проектирования MVC, что делает программу легко модифицируемой при ее дальнейшем развитии.

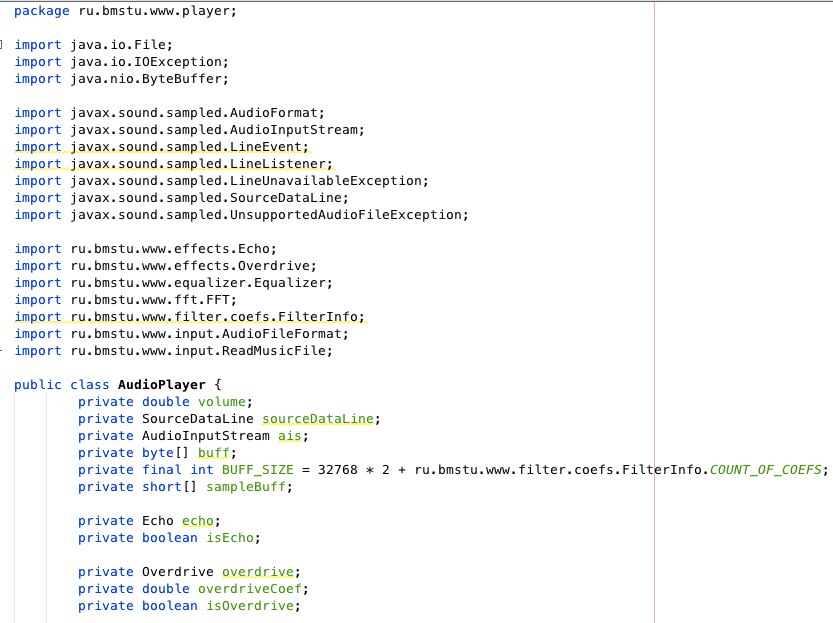
JavaFX – современная графическая библиотека, поддерживающая OpenGL и DirectX, что позволяет избавиться от присущих Java тормозов при отрисовке интерфейса(UI) на машинах, которые имеют в своем составе GPU, поддерживающие OpenGL или DirectX. Данная библиотека также позволяет снять нагрузку с CPU по отрисовке UI, если выполнено условие, обговоренное выше. JavaFX на сегодняшний день является одной из самых удобных сред для создания приложений.

**Класс** Equalizer.java отвечает за многопоточность, которая задается мощностью процессора.







**Пакет** ru.bmstu.www.player содержит в своем составе класс AudioPlayer.java, который непосредственно занимается обработкой данных, запуском потока воспроизведения.

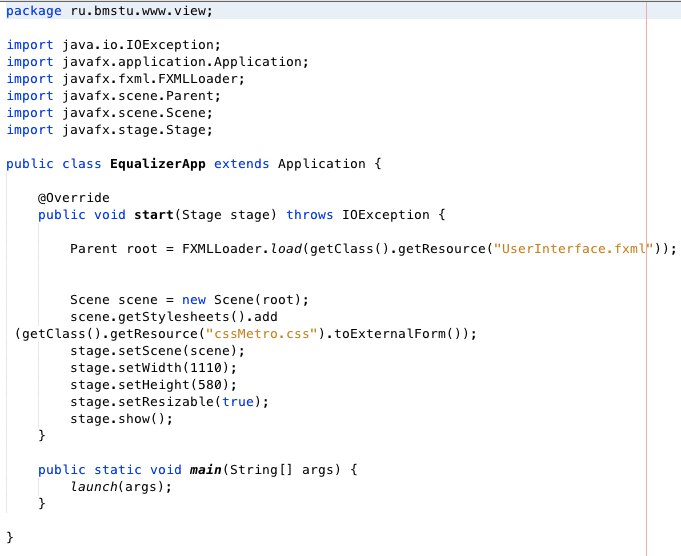


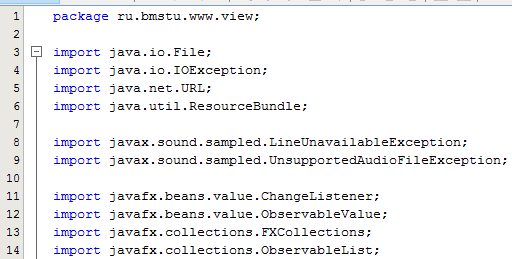


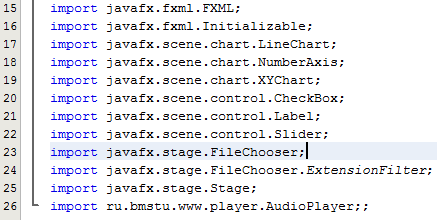


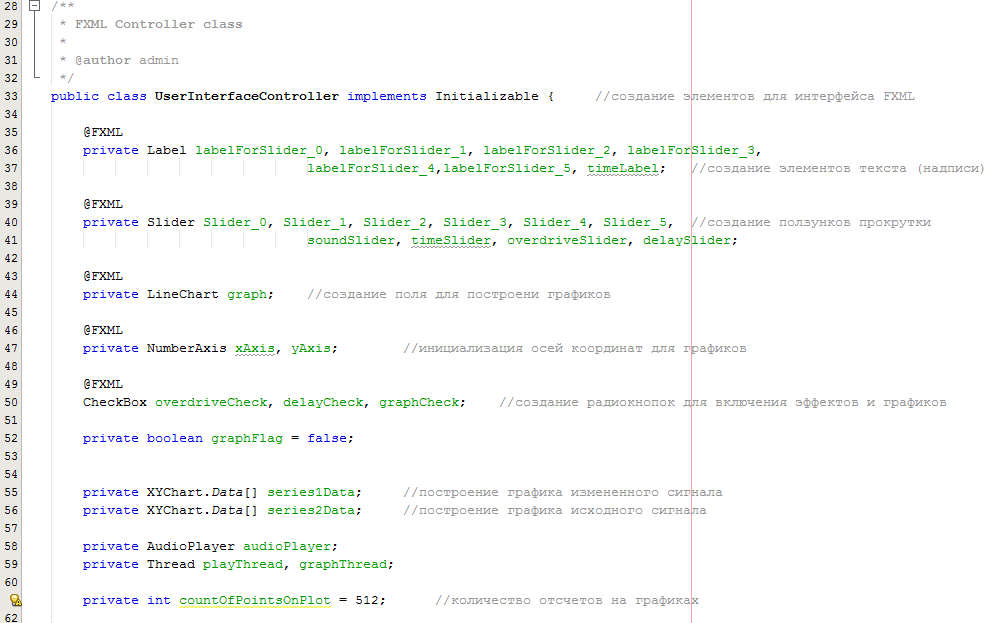


**Пакет** ru.bmstu.www.view содержит точку входа – класс EqualizerAPP.java, контроллер UI – UserUnterfaceController.java и саму верстку UI – UserInterface.fxml

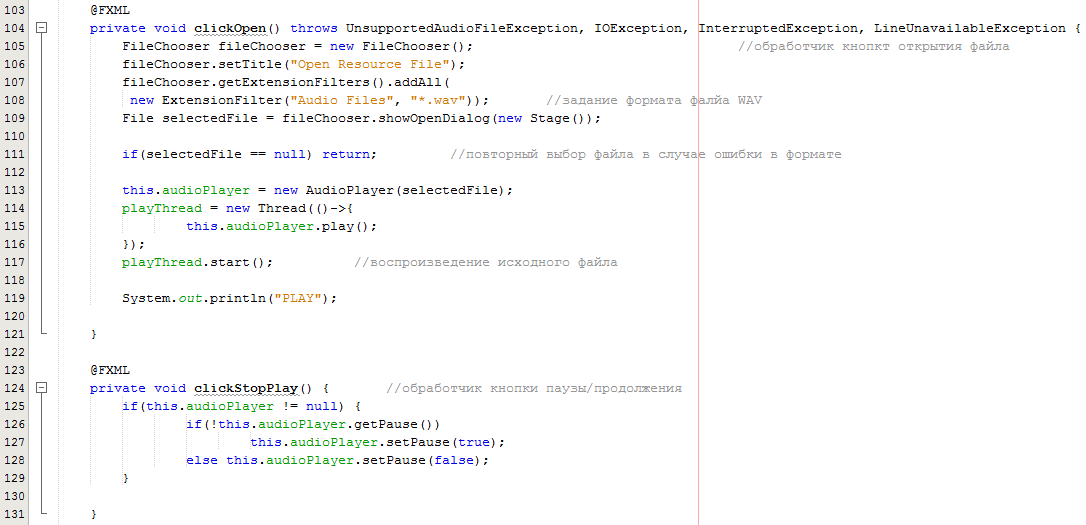
Класс EqualizerAPP.java

Класс UserInterface.java





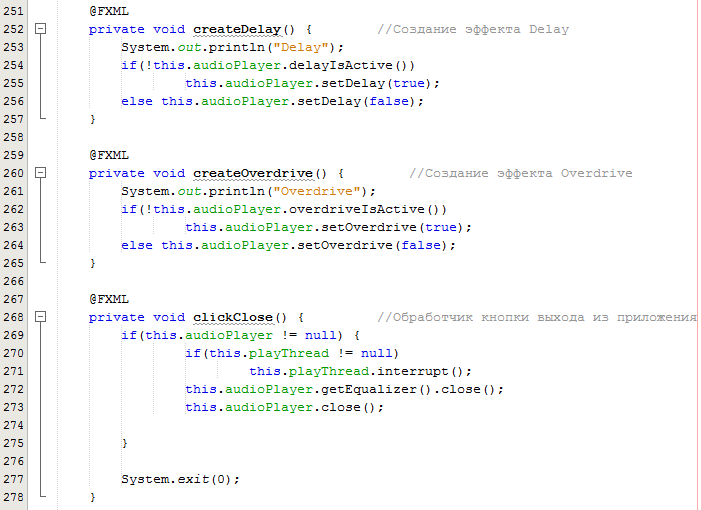






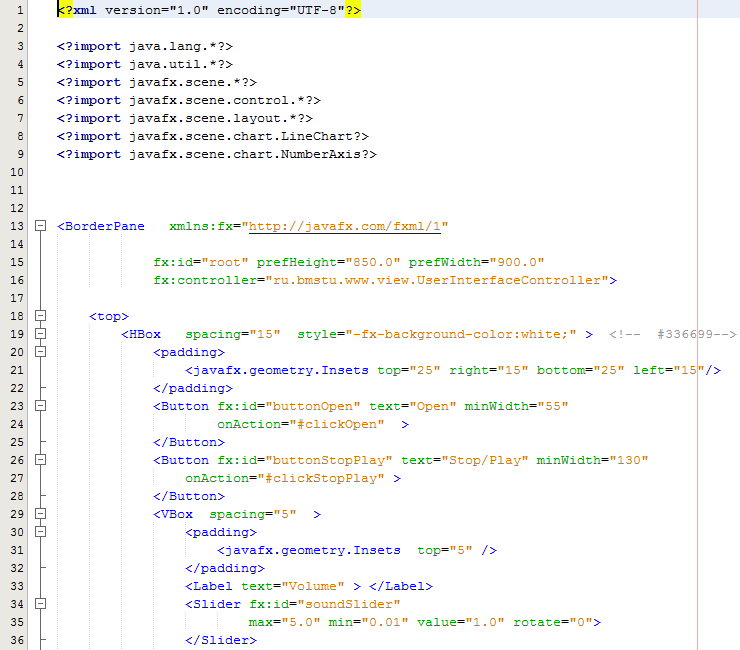






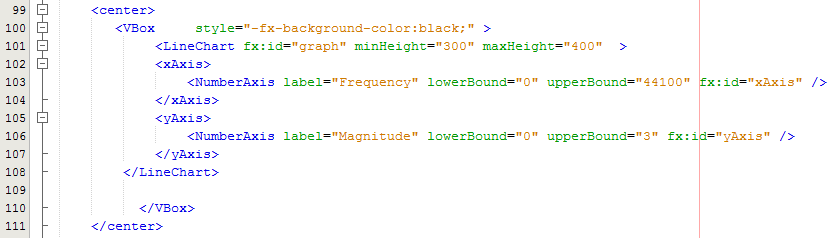


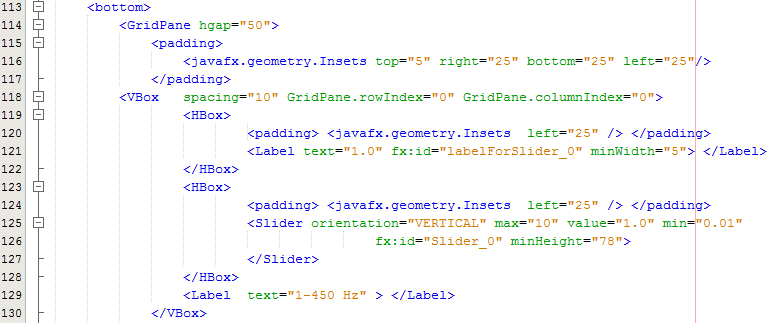
Верстка UserInterface.fxml













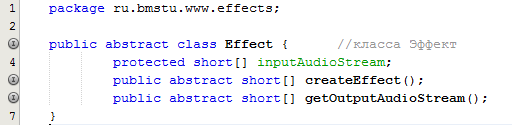


**Пакет** ru.bmstu.www.input содержит в себе класс для чтения файла – ReadMusicFile.java и класс, опиcывающий структуру \*.wav файла – AudioFileFormat.java

**Пакет** ru.bmstu.www.filter содержит в себе класс Filter.java, описывающий работу фильтра посредством свертки, пакет ru.bmstu.www.filter.coefs.Filter.Info.java содержит данные для кэширования, описывающие каждый КИХ фильтр.

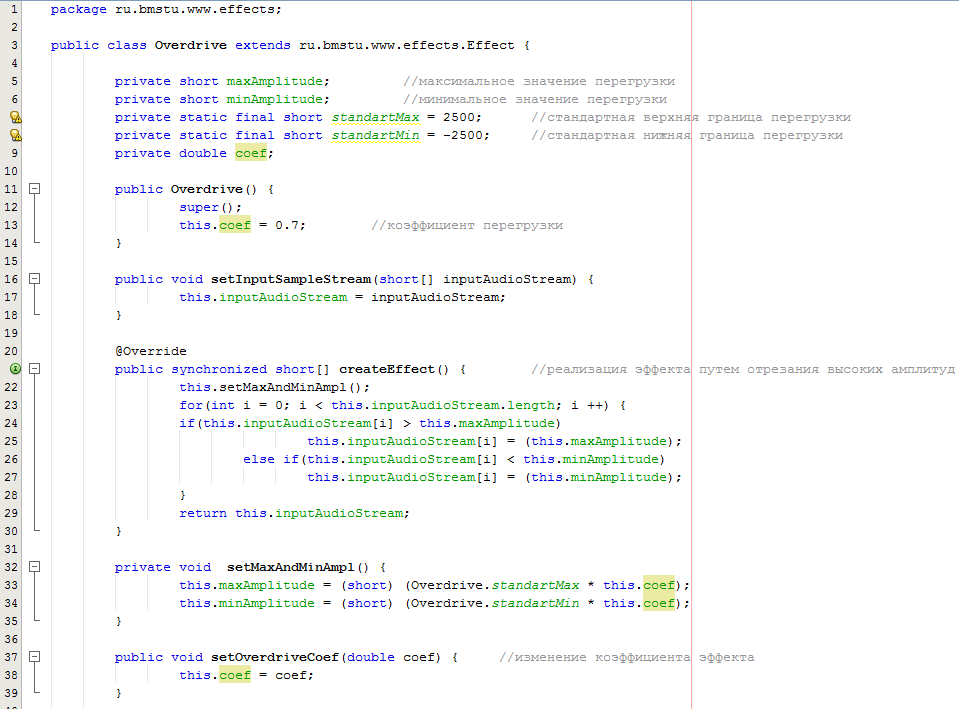
**Пакет** ru.bmstu.ru.www.effects содежит в себе абстрактный класс Effect.java, диктующий правила, по которым должны быть написаны конкретные классы, реализующие конкретные эффекты.

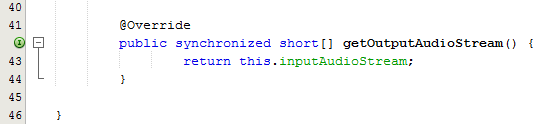
Класс Effect.java



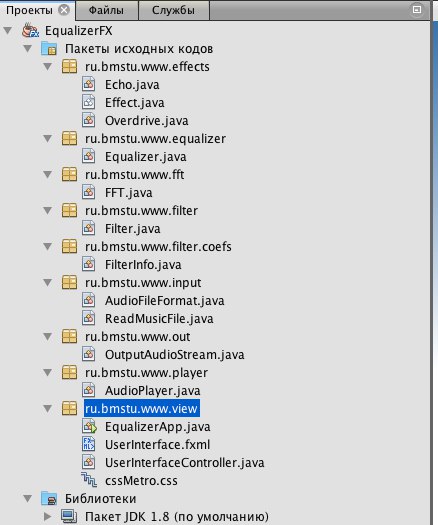
Echo.java – наследник Effect.java, реализующий эффект «задержка».



Overdrive – наследник Effect.java, реализующий эффект «перегрузка».



Иерархия файлов программы:



# Выводы

В результате выполнения курсового проекта был разработан аудио-плеер WAV-файлов с эквалайзером, реализованный в пакете MatLab2014. Затем данная задача была решена другим программным методом с использованием языка программирования Java и библиотек JavaFX 2.0, java.sound.\* . Проект с аудио-плеером был корректно отлажен и скомпилирован. В дополнение в среде NetBeans был создан JAR-файл, являющийся отдельным приложением. Данное приложение может быть использовано на других компьютерах без предварительного построения проекта и компиляции, что повышает удобство использования созданной программы.

Аудио-плеер был оснащен следующими возможностями:

* Работа с WAV-файлами
* Шестиполосный эквалайзер на основе КИХ-фильтров, спроектированных методом Хэмминга
* Изменение частоты сигнала
* Реализация эффекта «echo» или «overdrive», а также возможность их совместного наложения
* Использование приложения с проигрывателем на любом компьютере
* Управление воспроизведением через кнопки PLAY, PAUSE, CLOSE (STOP)
* Построение графиков первоначального и измененного сигналов

В заключении было проведено сравнение работы аудио-плееров, спроектированных в MatLab2014 и в JavaFX, по графикам и получены корректные результаты.

# Литература

1. Недашковский, В.М. Методические указания к выполнению домашнего задания «Проектирование графического цифрового эквалайзера»: учеб. пособие / В.М. Недашковский, В.С. Белолапотков. – М., 2014.

2. Конспекты лекций по курсу «Цифровая обработка сигналов».

3. URL: <http://websound.ru/articles/theory/digitalsound.htm> - Электронный ресурс на статью Александра Радзишевского. Дата обращения – 10.04.2016.

4. URL: <http://audiocoding.ru/formats/wav/> - Электронный ресурс. Дата обращения – 10.04.2016.

5. URL: <http://audiocoding.ru/article/2008/05/22/wav-file-structure.html> - Электронный ресурс. Дата обращения – 10.04.2016.

6. Нейл Бартлетт, Алекс Лесли, Стив Симкин Программирование на Java. Путеводитель .- The Coriolis Group,Inc.,1996, Издательство НИПФ "ДиаСофт Лтд.",1996