ГБОУ «Школа № 444»

Проектная работа по информатике:

«Конвертация MP3 в нотный лист»

Работу выполнил: Волков Дмитрий Ильич

ученица 10 «В» класса

Работу приняла: Мякова Елена Георгиевна

учитель информатики

Москва

2021 год

# **Введение**

Данная программа поможет всем музыкантам: от новичков до профессионалов. Что же делает эта программа? Представим, что пользователь услышал музыку на радио или, например, нашел MP3 файл в интернете и пользователь хочет сыграть именно эту аранжировку. Но как получить ему ноты именно этого исполнения? В данном случае у пользователя есть три пути:

Первый способ: пользователь купить профессиональную программу и там получить ноты понравившегося произведения. В этом случае пользователю придется мучиться в двумя вещами: расстаться с “кровными” деньгами и разбираться в купленной программе

Второй способ: пользователь может скачать две отдельный программы для того, чтобы получить ноты из MP3 формата. В этом случае пользователь может бесплатно конвертировать MP3 в ноты, но придется помучиться, ведь придется использовать 2 отдельные программы, в которых скорее всего также придется разбираться

Третий способ: скачать мою программу, “закинуть” в программу MP3 файл и получить на выходе ноты мелодии понравившейся аранжировки. Также, в моей программе не будет сложностей с понятием функционала, весь интерфейс интуитивно понятен. И главное, данная программа будет абсолютно бесплатно.

# **Цель и задача**

Целью проекта является создание приложения, которое облегчит жизнь пользователя. К примеру, музыкант услышал понравившуюся аранжировку. С помощью данной программы он сможет получить ноты из MP3 формата.

Для создания данного проекта нужно изучить кодировку MP3 и MIDI, изучить библиотеки, выбрать среду разработки, изучить алгоритмы, разработать интерфейс, создать приятный глазу дизайн и многое другое.

# **Основная часть**

## **1. Описание MP3 формата**

### 1.1 История

MP3 разработан рабочей группой института Фраунгофера под руководством Карлхайнца Бранденбурга и университета Эрланген-Нюрнберг в сотрудничестве с AT&T Bell Labs и Thomson (Джонсон, Штолл, Деери и др.).

Основой разработки MP3 послужил экспериментальный кодек ASPEC (Adaptive Spectral Perceptual Entropy Coding). Первым кодировщиком в формат MP3 стала программа L3Enc, выпущенная летом 1994 года. Спустя один год появился первый программный MP3-плеер — Winplay3.

При разработке алгоритма тесты проводились на вполне конкретных популярных композициях. Основной стала песня Сюзанны Веги «Tom’s Diner». Отсюда возникла шутка, что «MP3 был создан исключительно ради комфортного прослушивания любимой песни Бранденбурга», а Вегу стали называть мамой MP3.

Почти полный стандарт появился в открытом доступе 6 декабря 1991 года.

### 1.2 Описание формата

Как и формат JPEG, MP3 использует спектральные отсечения, согласно психоакустической модели. Звуковой сигнал разбивается на равные по продолжительности отрезки, каждый из которых после обработки упаковывается в свой фрейм (кадр). Разложение в спектр требует непрерывности входного сигнала, в связи с этим для расчётов используется также предыдущий и следующий фрейм. В звуковом сигнале есть гармоники с меньшей амплитудой и гармоники, лежащие вблизи более интенсивных — такие гармоники отсекаются, так как среднестатистическое человеческое ухо не всегда сможет определить присутствие либо отсутствие таких гармоник. Такая особенность слуха называется эффектом маскировки. Также возможна замена двух и более близлежащих пиков одним усреднённым (что, как правило, и приводит к искажению звука). Критерий отсечения определяется требованием к выходному потоку. Поскольку весь спектр актуален, высокочастотные гармоники не отсекаются, как в JPEG, а только выборочно удаляются, чтобы уменьшить поток информации за счёт разрежения спектра. После спектральной «зачистки» применяются математические методы сжатия и упаковка во фреймы. Каждый фрейм может иметь несколько контейнеров, что позволяет хранить информацию о нескольких потоках (левый и правый канал либо центральный канал и разница каналов). Степень сжатия можно варьировать, в том числе в пределах одного фрейма. Интервал возможных значений битрейта составляет 8—320 кбит/c.

## **2. Описание MIDI формата**

### 2.1 История

К концу 70-х годов XX века получили распространение музыкальные синтезаторы. Они представляли собой наборы генераторов звуковых частот, управляемых напряжением. Конкретные модели синтезаторов могли иметь особенности звучания и характерные специальные эффекты. Рабочее место музыканта того времени могло состоять из нескольких разнородных синтезаторов; работать с таким "зоопарком" было сложно. В начале 80-х годов компании-производители синтезаторов удачно смогли договориться о разработке и поддержке единого стандарта на интерфейс управления синтезаторами, который и появился в 1982 году. Интерфейс MIDI фактически стал средством передачи положения клавиш и регуляторов от музыкальной клавиатуры к аппаратуре синтеза звука. Стандарт быстро завоевал популярность. Постепенно дополняясь новыми возможностями, он не потерял актуальности и в наше время.

### 2.2 Стандарт на аппаратуру и программное обеспечение

Физический уровень интерфейса представляет собой токовую петлю (что обеспечивает гальваническую развязку и безопасность при соединении устройств между собой). Передатчик активный, 5 мА, наличие тока обозначает 0, нет тока — 1. Разъём 5-штырьковый DIN 41524. Приёмопередатчик асинхронный, скорость 31,25 кбит/с, формат 8-N-1.

Стандарт описывает аппаратный интерфейс, который позволяет соединять электронные музыкальные инструменты и компьютеры различных производителей, описывает протоколы связи для передачи данных от одного устройства к другому. MIDI-устройства могут взаимодействовать с программными приложениями, используя коммуникационный протокол MIDI. Используя соответствующий программный MIDI-секвенсор, внешние MIDI-устройства могут посылать информацию на синтезатор звуковой карты. MIDI базируется на пакетах данных, каждый из которых соответствует MIDI-событию (англ. MIDI-events), от нажатия клавиши до простой паузы, эти события разделяются по каналам. Сложная среда MIDI может включать различную аппаратуру, причём каждая часть системы будет отвечать за события на соответствующем канале. Альтернативным вариантом может быть одиночный синтезатор, который сам может управлять всеми каналами.

### 2.3 Формат музыкальных файлов

В отличие от других форматов это не оцифрованный звук, а наборы команд, которые могут воспроизводиться по-разному в зависимости от устройства воспроизведения. Удобство формата MIDI как формата представления данных позволяет реализовывать устройства, производящие автоматическую аранжировку по заданным аккордам, а также приложения 3D-визуализации звука. Кроме того, такие файлы, как правило, имеют на несколько порядков меньший размер, чем оцифрованный звук сравнимого качества.

Стандартный MIDI-файл (SMF — Standard MIDI File) — это специально разработанный формат файлов, предназначенный для хранения данных, записываемых и/или исполняемых секвенсором, секвенсор может быть как программой для компьютера, так и аппаратно выполненным модулем.

В этом формате хранятся стандартные MIDI-сообщения, а также временные метки или маркеры для каждого сообщения. Этот формат позволяет сохранять информацию о темпе, временном разрешении, выраженном в количестве тиков на одну четвертную длительность, обозначения размера, информацию о музыкальных ключах, а также хранить названия треков и паттернов. Формат предусматривает возможность сохранения в одном файле нескольких паттернов и треков таким образом, что программы-приложения могут выбирать из всего набора хранимой информации ту, которая будет понятна данному приложению.

## **3. Структура приложения**

Как будет выглядеть программа:

1. Открываем файл (MP3, WAV или MID)
2. Файлы добавляются в ListBox, там же мы можем выбрать файл, который хотим прослушать
3. Кнопка Play/Pause включает/останавливает произведение
4. MP3 файл можно перевести в WAV при нажатии на специальную кнопку
5. Из WAV файла можно выделить мелодию и перевести ее в MID при нажатии на специальную кнопку
6. Сохранение MID файла как нотный лист в PDF формате при нажатии на специальную кнопку
7. Печать MID файла как нотный лист при нажатии на специальную кнопку

Плагин “Melodia” выделяет мелодию в несколько шагов:

1. Извлечение синусоиды

Для начала, нужно выяснить, какие частоты присутствуют в звуковом сигнале в каждый момент времени. Они нужны для того, чтобы определить, какие высоты присутствуют в сигнале в каждый момент времени. Чтобы это узнать, плагин усиливает частоты, к которым человеческое ухо более чувствительно, а остальные ослабляет. Затем алгоритм разбивает сигнал на небольшие блоки для дальнейшей обработки, каждый из которых представляет определенный момент времени. К каждому блоку применяется дискретное преобразование Фурье, которое представляет собой преобразование, дающее интенсивность каждой частоты в аудиоблоке. Спектральные пики (то есть синусоиды) являются пиками этого преобразования, то есть они являются наиболее «энергичными» частотами, присутствующими в аудиосигнале в данном блоке. Для каждого блока сохраняется только эта пиковая частота и отбрасываются остальные частоты. Однако, у ДПФ есть недостаток, а именно ограниченное разрешение по частоте, что означает, что значение этих частот может быть немного неверным. Для исправления этого недочета алгоритм уточняет оценку частоты, данную ДПФ, вычисляя мгновенную частоту каждого спектрального пика с использованием разницы между последовательными фазовыми спектрами. Теперь для каждого момента времени имеется набор точных «энергетических» частот, которые присутствуют в сигнале.

1. Функция заметности

Теперь частоты, которые присутствуют в звуковом сигнале в каждый момент времени используются для оценивания, какие высоты присутствуют в каждый момент времени и насколько они заметны. Чтобы получить оценку того, насколько важен каждый шаг, используется гармоническое суммирование – то есть для каждого возможного шага (в разумном диапазоне) ищется гармонический ряд частот, которые будут способствовать восприятию этого тона. Сумма энергии этих гармонических частот считается «заметностью» этого тона. Это повторяется для каждого момента времени, что приводит к представлению значимости основного тона с течением времени, которая называется функцией заметности.

1. Создание контура

С помощью функции заметности отслеживаются контуры высоты тона. Контур основного тона представляет собой серию последовательных значений основного тона, которые являются непрерывными как по времени, так и по частоте. Продолжительность контура высоты тона может быть любой, от отдельной ноты до короткой фазы. Чтобы отследить эти контуры, берутся пики функции заметности в каждый момент времени, поскольку они представляют собой наиболее заметные высоты тона. Затем используется набор сигналов, основанный на потоковой передаче звука, чтобы сгруппировать эти пики в контуры.

1. Выбор мелодии

Теперь имеются все контуры высот тонов, осталось понять, какие из них принадлежат мелодии, а какие нет. Данный подход основан на расчете контурных характеристик. Для каждого контура вычисляется набор характеристик на основе его заметности и изменении высоты тона. Вычисляются такие характеристики, как средняя высота шага контура и его заметность, величина отклонения траектории шага контура и проверка, содержит ли контур вибрато или нет. Изучая распределение этих характеристик для контуров, принадлежащих мелодиям, и контуров, принадлежащих аккомпанементу, получается разработать набор правил для фильтрации немелодичных контуров. После применения этих правил фильтрации получается мелодия.

Конвертация в MIDI происходит так: для начала извлекается мелодия (описано выше), следующим шагом будет сегментирование мелодии н ноты и квантование высоты тона каждой ноты, создавая дискретную серию нот, которая затем может быть экспортирована в любой символический формат (MIDI, JAMS). Квантование непрерывной последовательности высоты тона в серию нот происходит следующим образом:

1. Преобразование последовательности высоты тона из герц в (дробные) номера MIDI-нот
2. Округление каждого значения до ближайшего целого числа MIDI-ноты
3. Использование медианного фильтра для сглаживания коротких скачков высоты звука (прим. из-за вибрато)
4. Перебирание последовательности и всякий раз, когда изменяется высота звука, начинается все с новой ноты

## **4. Выбор среды программирования и языка программирования**

Проект будет написан в Visual Studio.

Visual Studio – это интегрированная среда разработки программного обеспечения и ряд других инструментальных средств, разработанная компанией Microsoft. Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как, например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server).

Языком программирования был выбран C# и Python.

C# - объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров компании Microsoft как язык разработки приложений для платформы Microsoft .NET Framework. C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Переняв многое от своих предшественников — языков C++, Delphi, Модула, Smalltalk и, в особенности, Java — С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например, C# в отличие от C++ не поддерживает множественное наследование классов (между тем допускается множественная реализация интерфейсов).

Python – высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ. Python - это активно развивающийся скриптовый язык, который используют для решения большого объема самых разноплановых проблем и задач.

Причиной, почему я выбрал данные язык, является то, что для этих языков написано много музыкальных библиотек, которые могут быть полезны для написания данного проекта (NAudio, SoundFile, MIDIUtil и т.д.).

# **Заключение**

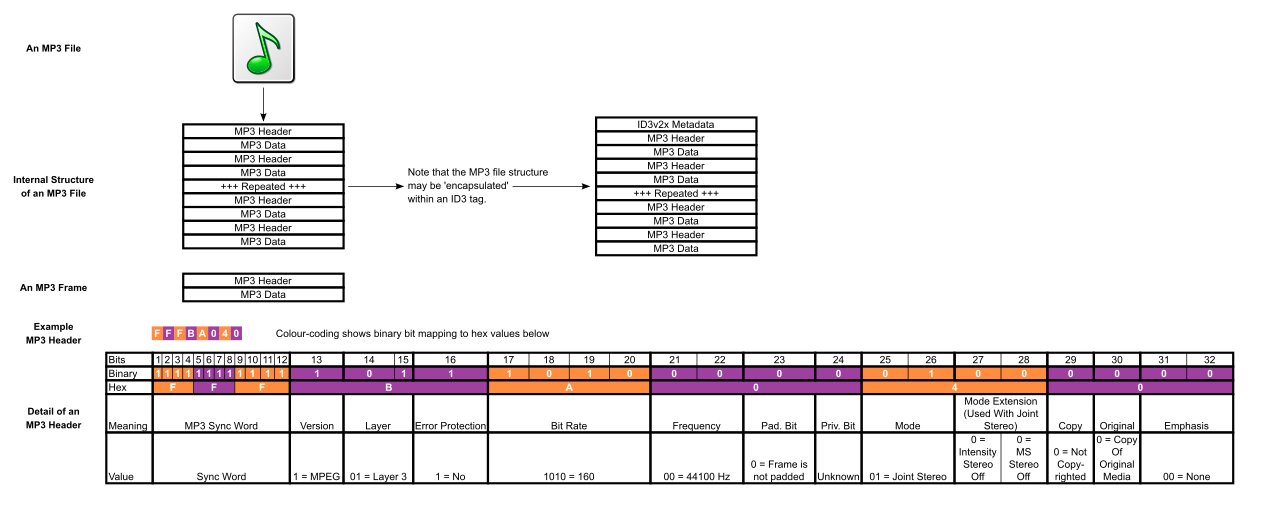
Подведу итоги. Во-первых, данная программа является абсолютно бесплатной. Во-вторых, у пользователя не будет заморочек с интерфейсом, когда пользователю приходится искать в интернете вопросы по интерфейсу. В то же время у программы такая же функция, что и у профессиональных программ, за которые приходится платить. Мне удалось сделать весь обязательный функционал (выделение мелодии, конвертация мелодии в MIDI, нотный лист), а также п

# **Список литературы**

1. <https://stackoverflow.com>
2. <https://habr.com>
3. <https://ru.wikipedia.org>
4. <https://web.archive.org/web/20160522222732/http://www.justinsalamon.com/melody-extraction.html>
5. <https://web.archive.org/web/20170921235215/http://justinsalamon.weebly.com/uploads/4/3/9/4/4394963/salamongomezmelodytaslp2012.pdf>
6. <https://en.wikipedia.org/wiki/Discrete_Fourier_transform>
7. <https://ourcodeworld.com/articles/read/983/how-to-extract-the-melody-from-an-audio-file-and-export-it-to-midi-generate-quantized-midi-using-python-in-ubuntu-18-04>
8. <https://progi.pro/naudio-t1909>
9. <https://web.archive.org/web/20210118180653/http://www.justinsalamon.com/phd-thesis.html>

# **Вложения**

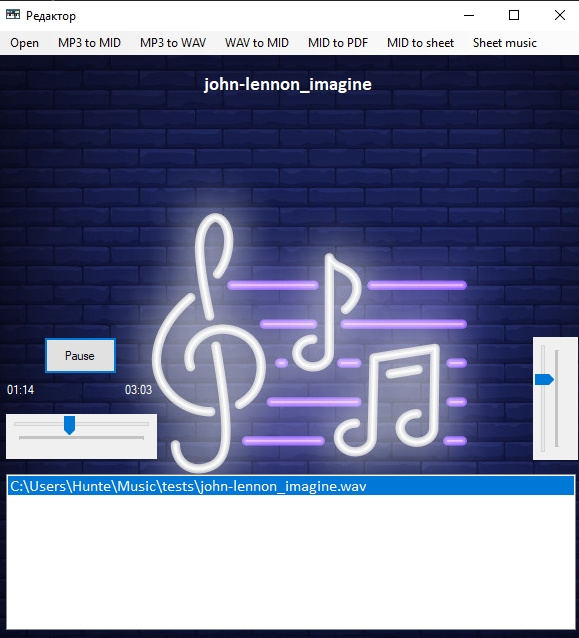
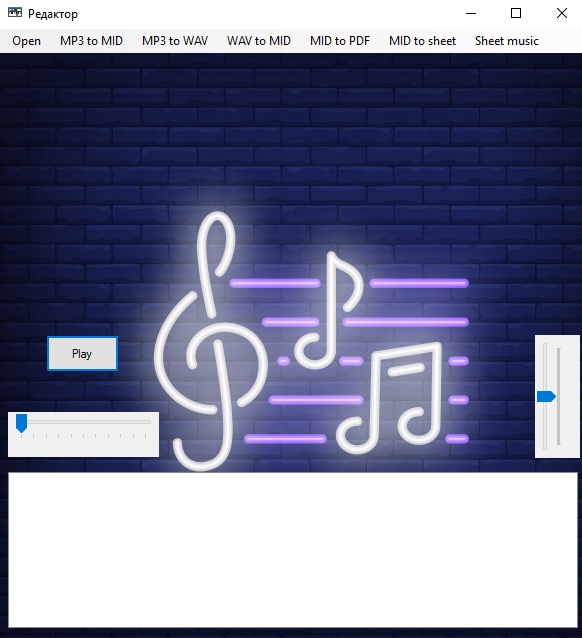
* Структура MP3 файла



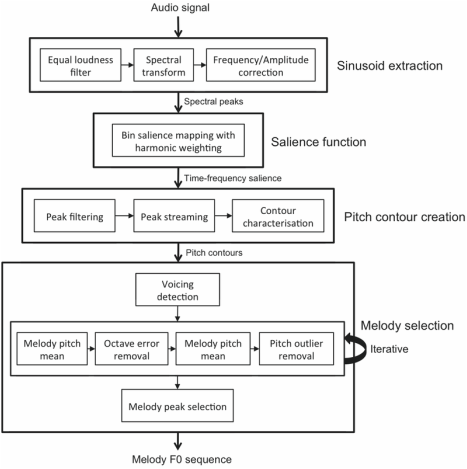
* Структура MIDI файла

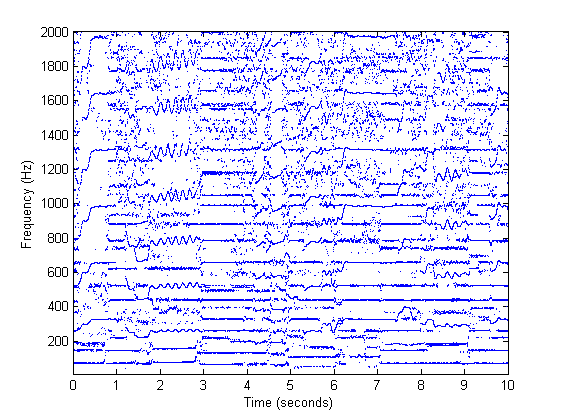


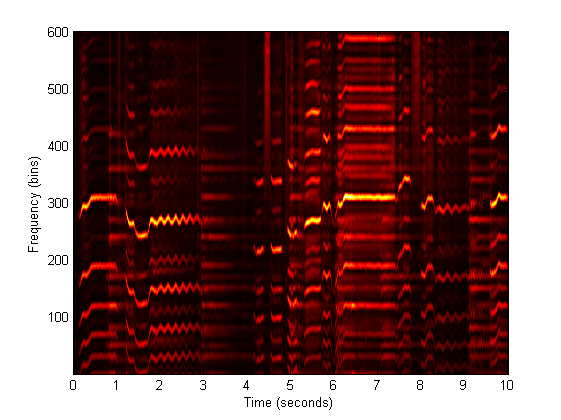
* Как выглядит программа



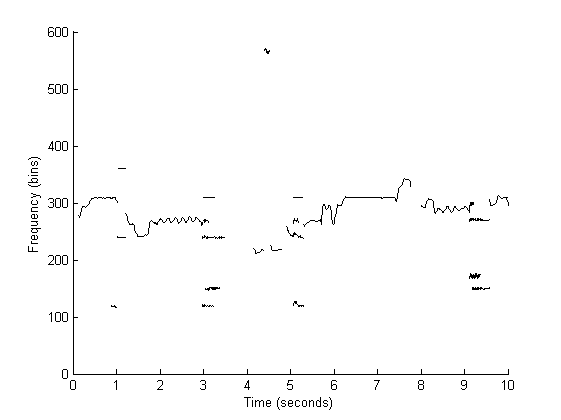
* Алгоритм “Melodia”



* Извлечение синусоиды
* Функция заметности



* Создание контура



* Выбор мелодии

