

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Забайкальский государственный университет»  
(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет

Кафедра информатики, вычислительной техники и прикладной математики

## **ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

Тема: «Система для распознавания мелодии и генерация её нотной записи»

Выполнил:

студент гр. ВМК–16

Молодогоженов Никита Алексеевич

Руководитель ВКР:

к.т.н., доцент кафедры ИВТ и ПМ

Макаров Дмитрий Андреевич.

Чита  
2020

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Забайкальский государственный университет»  
(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет  
Кафедра информатики, вычислительной техники и прикладной математики

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
к выпускной квалификационной работе

по направлению подготовки 09.03.01 Информатика и вычислительная техника  
профиль Вычислительные машины, комплексы, системы и сети

на тему «Система для распознавания мелодии и генерация её нотной записи»

Выполнил студент группы ВМК-16 Молодого Никита Алексеевич

Консультанты:

- а) Экономическая часть: Ткач доцент кафедры ЭТиМЭ,  
Т.И. Кашурникова
- б) Безопасность и экологичность проекта: Бондарь доцент кафедры ВХЭиПБ, к.т.н.,  
И.А. Бондарь
- в) Специальная часть: Макаров доцент кафедры ИВТ и ПМ, к.т.н.,  
Д.А. Макаров

Нормоконтроль: Шевелёва старший преподаватель кафедры ИВТ и ПМ, Е.Б. Шевелёва

Руководитель работы: Макаров доцент кафедры ИВТ и ПМ, к.т.н.,  
Макаров Дмитрий Андреевич

Допускаю к защите:

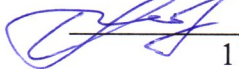
Зав. кафедрой ИВТ и ПМ Валова О.В. Валова

15 июня 2020 г.

Чита  
2020

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Забайкальский государственный университет»  
(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет  
Кафедра информатики, вычислительной техники и прикладной математики

УТВЕРЖДАЮ  
Зав.кафедрой ИВТ и ПМ  
 О.В. Валова  
11 мая 2020 г.

ЗАДАНИЕ  
на выпускную квалификационную работу

Студенту Молодогожену Никите Алексеевичу  
направления подготовки 09.03.01 Информатика и вычислительная техника  
профиль Вычислительные машины, комплексы, системы и сети

1 Тема выпускной квалификационной работы «Система для распознавания мелодии и генерация её нотной записи»

Утверждена приказом (распоряжением) ректора по университету от 20.12.2019 № 1003-с

2 Срок подачи студентом законченной работы 15 июня 2020 г.

3 Исходные данные к работе:

- документация C#;
- документация Accord.NET;
- информация о работе распознавательных систем.

4 Перечень подлежащих разработке в выпускной квалификационной работе вопросов:

- а) специальная часть;
- б) экономическая часть;
- в) Охрана труда при работе с ПЭВМ.

5 Перечень графического материала (если имеется): -

6 Консультанты по выпускной квалификационной работе (с указанием относящихся к ним разделов):

а) экономическая часть: Кашурникова Тина Иннокентьевна Тина

б) охрана труда при работе с ПЭВМ: Бондарь Ирина Алексеевна Ирина

в) нормоконтроль: Шевелёва Екатерина Борисовна Екатерина

г) специальная часть: Макаров Дмитрий Андреевич Дмитрий

Дата выдачи задания: 11 мая 2020 г.

Руководитель ВКР Д. А. Макаров /Д. А Макаров/

Задание принял к исполнению

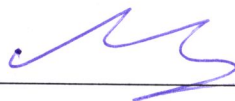
11 мая 2020 г.

Подпись студента Никита Алексеевич Молодоженов / Никита Алексеевич Молодоженов /

# Календарный план

№	Наименование раздела выпускной квалификационной работы	Неделя							
		1	2	3	4	5	6	7	8
1	Теоретическая часть								
2	Специальная часть								
3	Экономическая часть								
4	Безопасность и экологичность проекта								
5	Защита выпускной квалификационной работы								

(подпись, расшифровка подписи)



/ Д.А. Макаров /

## РЕФЕРАТ

Пояснительная записка – 60 с, 21 рис., 7 источников, 1 приложение

СПЕКТР, ОСНОВНАЯ ЧАСТОТА, АНАЛИЗ ЧАСТОТ, НОТА, C#,  
VISUAL STUDIO, ACCORD.NET FRAMEWORK, ДИСКРЕТНОЕ  
ПРЕОБРАЗОВАНИЕ ФУРЬЕ, БЫСТРОЕ ПРЕОБРАЗОВАНИЕ ФУРЬЕ

В данной работе представлена реализация метода анализа частоты  
помощью средств языка C#, в среде разработки Visual Studio 2017.



# СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	9
1 Постановка и анализ задачи .....	10
1.1 Теоретическая часть .....	10
1.1.1 Математическая теория .....	100
1.1.2 Структура звукового файла.....	<b>Ошибка! Закладка не определена.</b> 6
1.2 Анализ существующих систем .....	19
1.3 Постановка задачи.....	20
2 Выполнение работы .....	22
2.1 Анализ данных.....	22
2.2 Программная реализация .....	22
2.3 Обоснование выбора средств реализации .....	27
2.4 Тестирование .....	28
3 Документирование .....	32
3.1 Техническое задание.....	32
3.1.1 Введение.....	32
3.1.2 Назначение разработки.....	32
3.1.3 Требования к функциональным характеристикам .....	32
3.1.4 Требования к надежности .....	32
3.1.5 Требования к составу и параметрам технических средств.....	33
3.1.6 Информационная и программная совместимость .....	33
3.1.7 Требования к программной документации .....	33
3.2 Руководство пользователя.....	34
3.2.1 Минимальные требования.....	34
3.2.2 Управление приложением.....	34
4 Экономическая часть .....	36
4.1 Основные положения экономической части .....	36
4.2 Трехуровневый анализ продукта.....	37

4.3	Определение трудоемкости разработки программного продукта	Ошибка! Закладка не определена.
4.4	Определение стоимости программного обеспечения .....	42
4.5	Оценка эффективности программы	Ошибка! Закладка не определена.
5	Охрана труда.....	47
5.1	Общие положения охраны труда.....	Ошибка! Закладка не определена.
5.2	Анализ достоинств интерфейса .....	47
5.3	Рекомендации по организации рабочего места .....	48
5.4	Требования к ПЭВМ и работе с ними.....	49
5.5	Требования к освещению рабочих мест .....	51
5.6	Рекомендации по организации труда и отдыха .....	53
5.7	Требования электробезопасности .....	54
5.8	Требования к пожарной безопасности.....	54
5.9	Меры оказания первой медицинской помощи при поражении электрическим током .....	56
ЗАКЛЮЧЕНИЕ .....		57
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....		58
ПРИЛОЖЕНИЯ.....		59



## ВВЕДЕНИЕ

Традиционно в цифровой звукозаписи аудиодорожка представляется в виде осциллограммы, отображающей форму звуковой волны, то есть зависимость амплитуды звука от времени. Такое представление достаточно наглядно для опытного звукорежиссёра: осциллограмма позволяет увидеть основные события в звуке, такие как изменения громкости, паузы между частями произведения и зачастую даже отдельные ноты в сольной записи инструмента. Но одновременное звучание нескольких инструментов на осциллограмме смешивается и визуальный анализ сигнала становится затруднительным, тоже самое происходит, когда звучание инструмента перенасыщено тембральными деталями. Здесь в работу вступают средства спектрального анализа аудио.

Подобный вид программных средств по ряду причин имеет затруднительную реализацию, а задачи, решаемые посредством их использования, требуют применения высокоэффективных алгоритмов. Несмотря на сложность разработки таких систем, данная область представляет высокий интерес с исследовательской точки зрения, так как позволяет делать анализ звукового файла отталкиваясь от его физических свойств.

Результатом данной работы является создание приложения, реализующего анализ звукового сигнала и выявления его нот.

# 1. Постановка и анализ задачи

## 1.1 Теоретическая часть

В настоящий момент во многих студиях звукозаписи для облегчения работы звукооператора используются программы для анализа звука. Поскольку человеческий фактор напрямую влияет на качество работы, сведения и мастеринга композиции, зачастую необходимо использовать программный аппарат для правильного представления амплитудно-частотной характеристики композиции, чтобы избежать конфликтов частот и пиковых значений громкости отдельных звуков.

Проанализировав способы работы со звуковыми файлами, было принято решение реализовать систему распознавания нот аудиофайла.

### 1.1.1 Математическая теория

В данной работе производятся операции над комплексными числами.

Комплексное число – это выражение вида  $a + b * i$ , где  $a$ ,  $b$  – действительные числа,  $a * i$  – так называемая мнимая единица, символ, квадрат которого равен  $-1$ , то есть  $i^2 = -1$ .

Число  $a$  называется действительной частью, а число  $b$  – мнимой частью комплексного числа  $z = a + b * i$ .

Видно, что действительные числа – это частный случай комплексных чисел.

Арифметические действия:

1. Сложение и вычитание происходят по правилу:

$$(a + b * i) \pm (c + d * i) = (a \pm c) + (b \pm d) * i.$$

2. Умножение по правилу:

$$(a + b * i) * (c + d * i) = (a - b * d) + (a * d + b * c) * i.$$

3. Число  $\bar{z} = a - b * i$  называется комплексно-сопряженным

$$k * z = a + b * i.$$

Равенство  $z \cdot \bar{z} = a^2 + b^2$  позволяет понять, как делить одно комплексное число на другое (ненулевое) комплексное число:

$$\frac{a+bi}{c+di} = \frac{(a+bi) \cdot (c-di)}{(c+di) \cdot (c-di)} = \frac{(ac+bd) + (bc-ad)i}{c^2+d^2} = \frac{ac+bd}{c^2+d^2} + \frac{bc-ad}{c^2+d^2}i$$

У комплексных чисел есть удобное и наглядное геометрическое представление: число  $z = a + b \cdot i$  можно изображать вектором с координатами  $(a; b)$  на декартовой плоскости на рисунке 1.

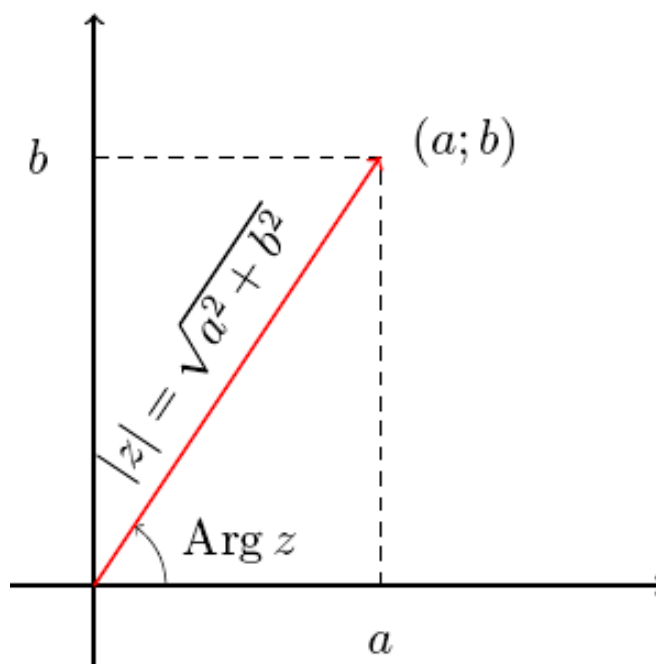


Рисунок 1 – Геометрическое представление

При этом сумма двух комплексных чисел изображается как сумма соответствующих векторов (которую можно найти по правилу параллелограмма).

По теореме Пифагора длина вектора с координатами  $(a; b)$  равна  $\sqrt{a^2 + b^2}$ .

Эта величина называется модулем комплексного числа  $z = a + b \cdot i$  и обозначается  $|z|$ .

Угол, который этот вектор образует с положительным направлением оси абсцисс, отсчитанный против часовой стрелки, называется аргументом комплексного числа  $z$  и обозначается  $\text{Arg}(z)$ . Аргумент определён не однозначно, а лишь с точностью до прибавления величины, кратной  $2\pi$

радиан, поскольку поворот на 360 градусов (вокруг начала координат) не изменяет угол.

Если вектор длины  $r$  образует угол  $\varphi$  с положительным направлением оси абсцисс, то его координаты равны  $(r \cdot \cos \varphi; r \cdot \sin \varphi)$ .

Отсюда получается тригонометрическая форма записи комплексного числа:

$$z = |z| \cdot (\cos (\operatorname{Arg}(z)) + i \cdot \sin(\operatorname{Arg}(z))).$$

Впоследствии будем использовать это представление.

Для подготовки звукового файла к определению ноты нужно получить его амплитудно–частотную характеристику поэтому в этой работе важнейшим алгоритмическим решением является дискретное преобразование Фурье.

В этой работе будут рассмотрены алгоритмы классического и быстрого преобразования Фурье.

#### 1. Дискретное преобразование Фурье (ДПФ).

Пусть звуковой сигнал представлен функцией  $x(t)$  с периодом  $T$ .

Тогда для существования преобразования Фурье достаточно выполнения следующих условий:

1.  $x(t)$  - должна быть ограничена при  $t \in (-\infty; +\infty)$ ;
2.  $x(t)$  - абсолютно интегрируема на  $(-\infty; +\infty)$ ;
3. Число точек разрыва, максимума и минимума функции  $x(t)$  – конечно.

При соблюдении этих условий функцию  $x(t)$  можно представить [4]:

$$x(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left( a_n \cos \frac{2\pi n t}{T} + b_n \sin \frac{2\pi n t}{T} \right) \quad (1)$$

где  $a_n$  и  $b_n$  рассчитываются по формулам (2) и (3):

$$a_n = \frac{2}{T} \int_{-T/2}^{T/2} x(t) \cos 2\pi \nu_0 n t dt, \quad (2)$$

$$b_n = \frac{2}{T} \int_{-T/2}^{T/2} x(t) \sin 2\pi \nu_0 n t dt, \quad (3)$$

При составлении разложения функции  $x(t)$ , будет некорректным характеризовать каждую из полученных гармоник значениями неизвестных коэффициентов  $a_n$  и  $b_n$ . Задача состоит в том, чтобы представить их в виде амплитуды и фазы. Для этого каждую пару слагаемых вида

$$a_n \cos \frac{2\pi n t}{T} + b_n \sin \frac{2\pi n t}{T}$$

Нужно записать в виде:

$$A_n \cos\left(\frac{2\pi n t}{T} - \varphi_n\right) \quad (4)$$

где  $A_n$  - Амплитуда;

$\varphi_n$  - фаза.

Тогда их можно найти по формулам (5) и (6).

$$A_n = \sqrt{a_n^2 + b_n^2}, \quad (5)$$

$$\varphi_n = \arctan\left(\frac{a_n}{b_n}\right) \quad (6)$$

Множество величин  $A_n$  и  $\varphi_n$  будут являться спектром исходного сигнала.

Для периодической функции с дискретными отсчётами  $X_n$  формула будет выглядеть так (7).

$$X_k = \sum_{n=0}^{N-1} x_n \left[ \cos\left(\frac{2\pi k n}{N}\right) - i \sin\left(\frac{2\pi k n}{N}\right) \right], \quad (7)$$

$(k = 0, \dots, N-1)$

где  $N$  - количество значений сигнала, измеренных за период, а также количество компонент разложения.

$X_k$ ,  $k=0, \dots, N-1$  -  $N$  комплексных амплитуд синусоидальных сигналов, слагающих исходный сигнал; являются выходными данными для прямого преобразования. Поскольку амплитуды комплексные, то по ним можно

вычислить одновременно и амплитуду, и фазу:

Обычная (вещественная) амплитуда  $k$ -го синусоидального сигнала представлена на формуле (8).

$$\frac{|X_k|}{N} \quad (8)$$

Фаза  $k$ -го синусоидального сигнала (аргумент комплексного числа) выглядит как показано на формуле (9).

$$\arg(X_k) \quad (9)$$

где  $k$  -индекс частоты.

Частота  $k$  - го сигнала равна (10):

$$\frac{k}{T}, \quad (10)$$

где  $T$  – период времени, в течение которого брались входные данные.

Для выполнения обратной задачи, преобразования амплитудно–частотной характеристики в исходный сигнал необходимо воспользоваться обратным преобразованием Фурье формула (11).

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k [\cos(\frac{2\pi kn}{N}) + i \sin(\frac{2\pi kn}{N})], \quad (11)$$

$(n = 0, \dots, N-1);$

## 2. Быстрое преобразование Фурье (БПФ)

БПФ это алгоритм ускоренного вычисления дискретного преобразования Фурье, позволяющий получить результат за время, меньшее чем  $O(N * \log(N))$ .

Быстрое преобразование Фурье является рекурсивным алгоритмом с древовидной структурой спуска, на каждом шаге спуска рекурсии набор отсчётов делится на две группы с чётными и не чётными номерами отсчётов. Спуск происходит до тех пор, пока каждая из групп не станет содержать в себе один единственный отсчёт. Поэтому число отсчётов должно быть равно степени двойки [5].

Пусть имеется многочлен  $A(x)$  порядка  $n$  как показано на формуле (11), где  $n > 1$ ,  $n=2^t$ . Если  $n$  не является степенью двойки, то необходимо добавить

недостающие члены и положить коэффициенты равными нулю.

$$A(x) = a_0x^0 + a_1x^1 + \dots + a_{n-1}x^{n-1} \quad (12)$$

Разделим  $A(x)$  на два многочлена, где один будет с чётными, а другой с нечётными коэффициентами, как показано в формуле (13) и (14):

$$A_0(x) = a_0x^0 + a_2x^1 + \dots + a_{n-2}x^{\frac{n}{2}-1}, \quad (13)$$

$$A_1(x) = a_1x^0 + a_3x^1 + \dots + a_{n-1}x^{\frac{n}{2}-1} \quad (14)$$

Многочлен  $A(x)$  получается из  $A_0(x)$  и  $A_1(x)$  следующим образом (15):

$$A(x) = A_0(x^2) + xA_1(x^2) \quad (15)$$

Следующим шагом нужно найти БПФ( $A_0$ ) и БПФ( $A_1$ ). Пусть

$$\mathcal{DПФ}(A_0) = \{y_k^0\}_{k=0}^{\frac{n}{2}-1}, \quad \mathcal{DПФ}(A_1) = \{y_k^1\}_{k=0}^{\frac{n}{2}-1}$$

Найдём вектор значений  $\mathcal{ДПФ}(A) = \{y_k\}_{k=0}^{n-1}$

Из формулы (14) получаем значения для первой половины коэффициентов:

$$\begin{aligned} y_k &= y_k^0 + \omega_n^k y_k^1 \\ k &= 0 \dots \frac{n}{2} - 1 \end{aligned} \quad (16)$$

Для второй половины получаем формулу (16):

$$\begin{aligned} y_{k+\frac{n}{2}} &= A(\omega_n^{k+\frac{n}{2}}) = A_0(\omega_n^{2k+n}) + \omega_n^{k+\frac{n}{2}} A_1(\omega_n^{2k+n}) = \\ &= A_0(\omega_n^{2k} \omega_n^n) + \omega_n^k \omega_n^{\frac{n}{2}} A_1(\omega_n^{2k} \omega_n^n) \end{aligned} \quad (17)$$

где  $\omega_n^n = 1$ ,

$$\omega_n^{\frac{n}{2}} = -1$$

тогда  $y_{k+\frac{n}{2}}$  принимает вид (17):

$$y_{k+\frac{n}{2}} = A_0(\omega_n^{2k}) - \omega_n^k A_1(\omega_n^{2k}) = y_0 - \omega_n^k y_k^1 \quad (18)$$



Таким образом получаем формулы (19) и (20):

$$y_k = y_k^0 + \omega_n^k y_k^1, k = 0 \dots \frac{n}{2} - 1 \quad (19)$$

$$y_{k+\frac{n}{2}} = y_k^0 - \omega_n^k y_k^1, k = 0 \dots \frac{n}{2} - 1 \quad (20)$$

### 1.1.2 Структура звукового файла

Звук, как и любые данные в памяти компьютера хранится в дискретной форме, то есть виде последовательности битов.

Для примера рассмотрим аудиофайл формата WAV. WAV-файл использует стандартную RIFF-структуру, которая группирует содержимое файла из отдельных секций (chunks) – формат выборок аудиоданных. Каждая секция имеет свой отдельный заголовок секции и отдельные данные секции. Заголовок секции указывает на тип секции и количество содержащихся в ней байт.

Такой принцип организации позволяет программам анализировать только необходимые секции, пропуская остальные секции, которые не известны или которые не требуют обработки. Некоторые определённые секции могут иметь в своем составе подсекции (sub-chunks). Рисунок 2, описывает основной формат WAV-файла, секции «fmt» и «data» являются подсекциями секции «RIFF».

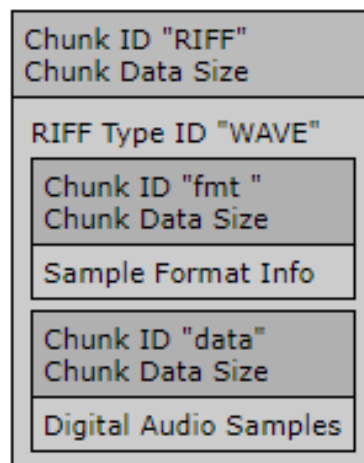


Рисунок 2 – Основной формат WAV-файла

Заголовки WAV - файла используют стандартный формат RIFF. Первые 8 байт файла - стандартный заголовок секции RIFF, который имеет ID секции «RIFF» и размер секции, равный размеру файла минус 8 байт, используемых для RIFF – заголовка, представлены на рисунке 3. Первые 4 байта данных в секции «RIFF» определяют тип ресурса, который можно найти в секции. WAV - файлы всегда используют тип ресурса «WAVE». После типа ресурса (ID «WAVE») идут все секции звукового файла, которые определяют аудиосигнал.

Смещение	Размер	Описание	Значение
0x00	4	Chunk ID	"RIFF" (0x52494646)
0x04	4	Chunk Data Size	(file size) - 8
0x08	4	RIFF Type	"WAVE" (0x57415645)
0x10	Wave chunks (секции WAV-файла)		

Рисунок 3 – Значения полей секции RIFF

Существует довольно много типов секций, заданных для файлов WAV, но большинство WAV-файлов содержат только две из них - секцию формата («fmt») и секцию данных («data»). Это именно те секции, которые необходимы для описания формата выборок аудиоданных, и для хранения самих аудиоданных. Хотя официальная спецификация не задаёт чёткий порядок следования секций, наилучшей практикой будет размещение секции формата перед секцией данных. Многие программы ожидают именно такой порядок секций, и он наиболее разумен для передачи аудиоданных через медленные, последовательные источники. Иначе если формат придет после данных, то перед стартом анализа или воспроизведения необходимо считать и запомнить все аудиоданные и только после получения формата начнётся обработка.

Все секции формата RIFF и соответственно секции Wave хранятся в следующем формате, представленном на рисунке 4. Заметим, что даже вышеупомянутая секция RIFF соответствует этому формату.

Смещение	Размер	Описание
0x00	4	Chunk ID
0x04	4	Chunk Data Size
0x08	Chunk Data Bytes	

Рисунок 4 – секции RIFF и Wave

Секция данных Wave (Wave Data Chunk) содержит данные цифровых выборок аудиосигнала, которые можно декодировать с использованием формата и метода компрессии, указанных в секции формата Wave (Wave Format Chunk). Если код компрессии 1 (несжатый PCM, Pulse Code Modulation), то данные представлены в виде сырых, непреобразованных (raw) величин выборок. WAV - файлы обычно содержат только одну секцию данных, но секций может быть несколько, если они содержатся в секции списка Wave (Wave List Chunk «wavl»), представлены на рисунке 5.

Смещение	Длина	Тип	Описание	Значение
0x00	4	char[4]	chunk ID	"data" (0x64617461)
0x04	4	dword	chunk size	зависит от количества выборок и компрессии
0x08	данные выборки (sample data)			

Рисунок 5 – Формат секции данных «data»

Звук состоит из колебаний, которые при оцифровке приобретают ступенчатый вид. Этот вид обусловлен тем, что компьютер может воспроизводить в любой короткий промежуток времени звук определённой амплитуды (громкости) и этот короткий момент конечен. Продолжительность этого промежутка определяет частота дискретизации.

У файла с частотой дискретизации 44.1 кГц, этот короткий промежуток времени равен  $\frac{1}{44100}$  секунды, это следует из размерности величины 1 Гц равен 1/с.

Амплитуда выражается числом, которое занимает в файле 8, 16, 24, 32 бита (теоретически можно и больше), в зависимости от этого увеличивается точность звука. Так как 8 бит равен 1 байту, то одно значение амплитуды в какой-то короткий промежуток времени в файле занимает 1, 2, 3, 4 байта соответственно.

Таким образом, чем больше число занимает места в файле, тем шире возможный диапазон значений для этого числа, а значит и больше точность амплитуды.

Совокупность амплитуды и короткого промежутка времени носит название сэмпл (Sample). В данной работе сэмпл представляет собой элементарную структурную единицу звукового файла, именно над ней производятся необходимые операции для определения его частоты и ноты впоследствии.

## **1.2 Анализ существующих систем**

Тема программного обеспечения очень широка, поэтому здесь мы только вкратце обсудим основных представителей программ для обработки звука.

Важный класс программ – редакторы аудио файлов. Основные возможности таких программ это, как минимум, обеспечение возможности записи (оцифровки) аудио и сохранение на диск. Развитые представители такого класса программ позволяют намного больше: многоканальное сведение аудио на нескольких виртуальных дорожках, обработка специальными эффектами (как встроенными, так и подключаемыми извне), очистка от шумов, имеют развитую навигацию и инструментарий в виде спектро스코па и прочих виртуальных приборов, управление/управляемость внешними устройствами, преобразование аудио из формата в формат, генерация сигналов, запись на компакт диски и многое другое. Некоторые из таких программ: Cool Edit Pro - Syntrillium, Sound Forge - Sonic

Foundry, Nuendo - Steinberg, Samplitude Producer - Magix, Wavelab - Steinberg.

Также выделяют отдельный класс программ - секвенсоров. При наличии всех функций музыкального редактора, такие программы обладают возможностью работы с MIDI сигналами, что делает их более функциональными и в свою очередь более популярными у пользователей. Некоторые из таких программ: FL Studio - Image-Line Software, Adobe Audition - Adobe Systems, Ableton Live - Ableton.

### **1.3 Постановка задачи**

Данное приложение разрабатывается для музыкантов разных уровней и студий звукозаписи. Программы этого класса крайне востребованы на рынке и с развитием музыкальной индустрии, растёт и количество пользователей, заинтересованных этой сферой.

Целью данной работы является реализация алгоритма, суть которого заключается в анализе звуковой дорожки. Необходимо распознать ноту инструмента или голоса звучащую в данный момент времени. Программа должна позволять пользователю проводить анализ звука, не зависимо от модели микрофона и аудиокарты.

Простота интерфейса - также является целью данной работы.

Работа с программой предполагает единственный вид действующего лица, которым может выступать любой пользователь. Поскольку система не является общедоступным сервисом и запускается индивидуально на устройстве каждого из пользователей, не возникает необходимости в разграничении прав. Диаграмма вариантов использования представлена на рисунке 6.



Рисунок 6 – Диаграмма вариантов использования

Взаимодействуя с программой, пользователь может осуществлять следующие действия:

- анализировать аудио в реальном времени;
- вывести полученный результат.

## **2 Выполнение работы**

### **2.1 Анализ данных**

Входными данными для системы распознавания являются следующие типы аудио файлов: MP3, WAV, FLAC, Ogg. Данные форматы являются наиболее распространёнными и чаще всего используются на практике.

Во время работы системы промежуточными данными являются фрейм – минимальные структурные единицы аудиофайла, над которыми возможно производить какие-либо операции, для извлечения полезной информации.

При динамическом анализе звука фреймы являются входными данными, поскольку во время записи, на вход алгоритма они подаются минуя алгоритм сэмплирования. Фрейм –  $N$  - байтовый элемент массива, в текущем представлении данных. Аналоговый сигнал, предварительно дискретизированный в цифровой формат.

Результат работы программы по анализу звуковых файлов, представляющий совокупность найденных нот может быть выведен на экран для последующего использования.

### **2.2 Программная реализация**

Основным механизмом работы с данными является дискретное преобразование Фурье. Для удобства обработки был создан статический класс `FourierTransform`. Класс включает в себя реализацию нескольких методов необходимых преобразований для комплексного сигнала, чтобы преобразования Фурье могли корректно работать. Так же класс включает в себя реализацию дискретного преобразования Фурье и быстрого преобразования Фурье. Методы класса образуют базовый набор возможностей и позволяют сделать большинство необходимых преобразований.



```

public static void DFT (Complex[] data, Direction direction)
public static void FFT (Complex[] data, Direction direction)
private static int[] GetReversedBits(int numberOfBits)
private static Complex[] GetComplexRotation(int numberOfBits, Direction direction)
private static void ReorderData(Complex[] data)

```

Первая задача, которую нужно решить это сэмплирование аудиофайла. Дело в том, что быстрое преобразование Фурье в силу своей рекурсивной природы не может оперировать сигналом, у которого число дискретных отсчётов не является степенью двойки. Чтобы это было возможно, звуковой файл разделяют на фреймы. Последние фреймы в звуковом файле чаще всего не подходят для обработки алгоритмом. Количество отсчётов последнего фрейма не обязано быть степенью двойки, поэтому недостающие отсчёты заполняются нулями, чтобы программа могла его проанализировать.

Для начала анализа необходимо выбрать аудиоинтерфейс с которым программа будет работать, при помощи кода с рисунка 7.

```

var info = comboBox1.SelectedItem as AudioDeviceInfo;
if (info == null)
{
    MessageBox.Show("Нет доступных аудиоустройств.");
    return;
}

```

Рисунок 7 – Аудиоинтерфейс не выбран

Получаем все аудиоинтерфейсы и добавляем их в выпадающий список, если к компьютеру не подключены аудиоустройства, то добавляется элемент «Нет локальных устройств захвата». Здесь же происходит конфигурация графика сигнала для отображения спектра.

```
protected override void OnLoad(EventArgs e)
{
    base.OnLoad(e);

    if (comboBox1.Items.Count == 0)
    {
        comboBox1.Items.Add("Нет локальных устройств захвата");
        comboBox1.Enabled = false;
    }
    comboBox1.SelectedIndex = 0;

    chart1.AddWaveform("fft", Color.Black, 1, false);
    var audioDevices = new AudioDeviceCollection(AudioDeviceCategory.Capture);
    foreach (AudioDeviceInfo device in audioDevices)
    {
        comboBox1.Items.Add(device);
    }
}
```

Рисунок 8 – Выбор аудиоинтерфейса и конфигурация

Если не выбран интерфейс, то выводится сообщение «Нет доступных аудиоустройств», как представлено на рисунке 8.

Далее необходимо создать устройство захвата звука, как представлено на рисунке 9, где поле `DesiredFrameSize` - задаёт размер окна, что будет гарантировать выборку временных отрезков с числом отсчётов  $2^n$ .

```
source = new AudioCaptureDevice(info)
{
    DesiredFrameSize = 2048,
    SampleRate = 22050
};
```

Рисунок 9 – Создание устройства захвата

На рисунке 10 показан один из ключевых методов `source_NewFrame`. Сначала программа конвертирует аудиофрейм в комплексный сигнал. Затем если необходимо использовать оконную функцию, то применяет её. Далее фрейм преобразуется при помощи `ForwardFourierTransform`.

На этом этапе фрейм представляет собой два набора частот и амплитуд в сыром виде (raw data). Далее сырые данные преобразуются в нормальные значения амплитуд и частот фрейма чтобы отобразить его на спектрограмме.

```

void source_NewFrame(object sender, NewFrameEventArgs eventArgs)
{
    ComplexSignal signal = ComplexSignal.FromSignal(eventArgs.Signal);
    if (window != null)
    {
        signal = window.Apply(signal, 0);
    }
    signal.ForwardFourierTransform();
    Complex[] channel = signal.GetChannel(0);
    double[] power = Accord.Math.Transforms.FourierTransform2.
        GetPowerSpectrum(channel);
    double[] freqv = Accord.Math.Transforms.FourierTransform2.
        GetFrequencyVector(signal.Length, signal.SampleRate);
    power[0] = 0;
    float[] g = new float[power.Length];
    for (int i = 0; i < power.Length; i++)
    {
        g[i] = (float)power[i];
    }
    chart1.RangeX = new DoubleRange(freqv[0], freqv[freqv.Length - 1]
        / hScrollBar1.Value);
    chart1.RangeY = new DoubleRange(0f, Math.Pow(10,
        -vScrollBar1.Value));
    chart1.UpdateWaveform("fft", g);
    mainFreq = freqMax(power, freqv);
}

```

Рисунок 10 – Метод source\_NewFrame

Метод source\_NewFrame вызывается, когда устройство захвата получает новый фрейм. Для обработки захвата нового фрейма используем событийную систему с методом source\_NewFrame. Здесь же в случае неисправности аудио источника, вызывается метод необходимый для обработки исключения, чтобы не прерывать процесс обработки аудио.

```

source.NewFrame += source_NewFrame;
source.AudioSourceError += source_AudioSourceError;

```

Рисунок 11 – Событийная система

Теперь для распознавания ноты получаем лидирующую частоту внутри метода source\_NewFrame при помощи функции freqMax, которая на вход принимает массивы амплитуд и частот, как представлено на рисунке 11,

затем используя подготовленную заранее структуру `note` в функции `fraqs`, лидирующая частота сопоставляется со реальными значениями нот.

Структура `note` заполняется серединами отрезков между соответствующими нотами и их именами в функции `inNote`, представлено на рисунках 12, 13, 14.

```
double·freqMax(double[]·power,·double[]·freq)
{
    ...double·maxPower=0;
    ...double·currFreq=0;
    ...for(int·i=0;·i<·freq.Length;·i++)
    ...{
        ...if(maxPower<·power[i])
        ...{
            ...maxPower=power[i];
            ...currFreq=freq[i];
        }
    }
    ...return·currFreq;
}
```

Рисунок 12 – Лидирующая текущая частота

```
public·struct·note
{
    ...public·double·freq1;
    ...public·double·freq2;
    ...public·string·name;
    ...public·note(double·freq1,·double·freq2,·string·name)
    ...{
        ...this.freq1=·freq1;
        ...this.freq2=·freq2;
        ...this.name=·name;
    }
}

void·inNote()
{
    ...for(int·i=0;·i<·n;·i++)
    ...{
        ...notes[i]=·new·note(
        ...    (GetNoteFreq(i)+GetNoteFreq(i+1))/2,
        ...    (GetNoteFreq(i+1)+GetNoteFreq(i+2))/2,·names[i]);
    }
}
```

Рисунок 13 – Структура `note` и функция `inNote`

```

public string fraqs(double f)
{
    string N = "0";
    for (int i = 0; i < n; i++)
    {
        if ((notes[i].freq1 < f) && (f < notes[i].freq2))
            N = notes[i].name;
    };
    return N;
}

```

Рисунок 14 – Определение ноты

В случае закрытия формы останавливается аудиозахват и освобождаются ресурсы, представленные на рисунке 15.

```

protected override void OnClosed(EventArgs e)
{
    base.OnClosed(e);
    if (source != null)
        source.SignalToStop();
}

```

Рисунок 15 –Закрытие формы

## 2.3 Обоснование выбора средств реализации

В качестве основного средства реализации для написания работы был применён язык программирования C#. Обоснованность выбора заключается в том, что данный язык является высокоуровневым, что позволяет использовать его для быстрого создания прикладных приложений с возможностью удобного подключения сторонних программных решений.

Для того чтобы реализовать необходимый функционал было принято решение использовать фреймворк Accord.NET с открытым кодом служащий для произведения научных вычислений.

Библиотека предоставляет функции для работы с численной линейной алгеброй, числовой оптимизацией и базовым набором возможностей для обработки сигналов.

В качестве редактора исходного кода был задействован продукт Microsoft под названием Visual Studio. Данная среда разработки представляет возможность осуществлять простейший рефакторинг кода, производить отладку приложения и его тестирование. Поддерживает ряд технологий и позволяет создавать программы с графическим интерфейсом.

## **2.4 Тестирование**

Перед представлением программы пользователям, были проведены многочисленные тесты для выявления слабых и опасных для системы мест.

В ходе тестирования были выявлены несколько проблем которые невозможно решить при помощи данного алгоритма.

Суть проблемы кроется в природе звуковых колебаний, некоторые гармоники в звуковой волне невозможно проанализировать поскольку они имеют слишком короткое время распространения. Такие колебания не имеют периода и по сути являются щелчками - единичными колебаниями с большой амплитудой, они в свою очередь напоминают ударную волну небольшой громкости. Как уже было сказано их невозможно проанализировать, но они влияют на конечный результат работы программы выступая в качестве непериодического шума.

Такие щелчки зачастую образуются в начале каждой новой гармоники и являются своего рода началом звуковой волны. Если инструмент не искусственно синтезирован, то такого щелчка избежать невозможно.

Следующая проблема кроется в структуре резонатора музыкального инструмента. В общем случае все звуковые волны, независимо от инструмента – схожи. Но материалы для изготовления инструментов могут обогащать звуковую волну различными тембральными окрасами, что может повлечь за собой частотные артефакты. Они являются самыми большими сложностями для определения ноты.

Примеры волн некоторых инструментов:

1. Звуковая волна камертона. Обычно камертон издаёт эталонную частоту 440 Гц, что соответствует точному значению частоты ноты «Ля» (А) первой октавы, представленной на рисунке 16.

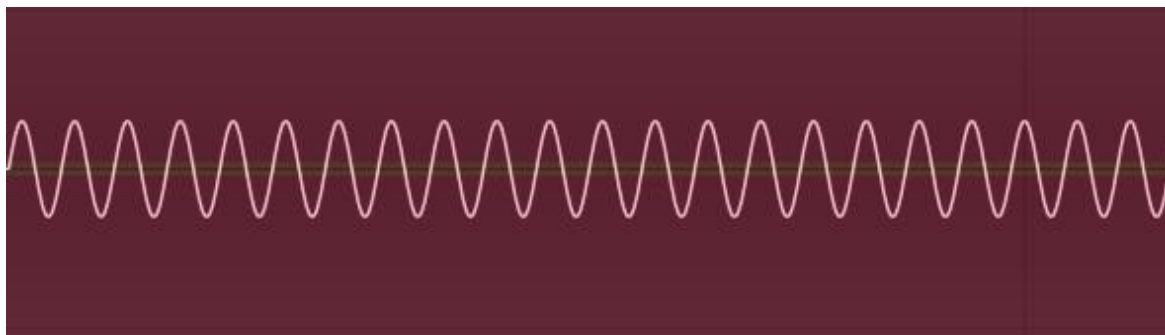


Рисунок 16 – Звуковая волна камертона на ноте «Ля»

2. Звуковая волна рояля. Как видно на рисунке 17, она обогащена множеством различных гармоник, это связано со сложностью резонатора рояля. А также природой колебания струны внутри него. Ко всему прочему внутри рояля находятся 88 струн, которые иногда входят в резонанс со звучащей струной и тоже начинают колебаться.



Рисунок 17 – Звуковая волна рояля для ноты «Ля»

3. Звуковая волна гитары. На рисунке 18 представлена волна не так обогащена гармониками, как рояль и рисунок волны более ясный. Это связано с тем, что корпус гитары мал и лёгкий, по сравнению с роялем и обладает гораздо меньшим количеством струн.



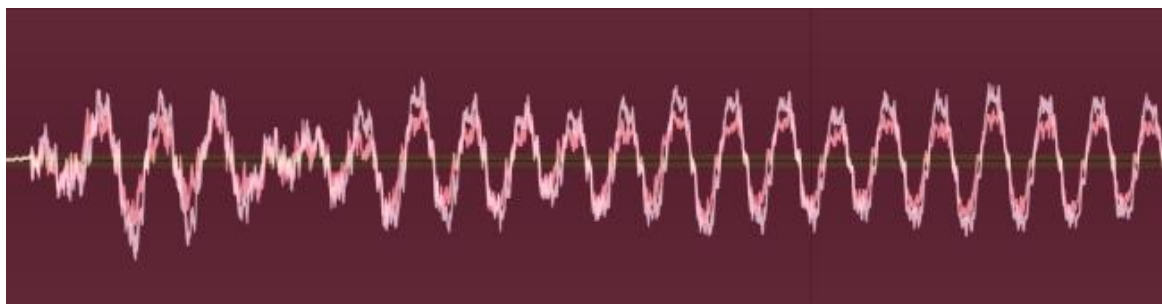


Рисунок 18 – Звуковая волна гитары для ноты «Ля»

С голосом человека всё намного сложнее.

Человеческий речевой аппарат обладает двумя сложными резонаторами:

1. Грудной – трахея и бронхи.
2. Головной – полость носа, полость рта и полость глотки.

Источником звука же выступают связки. Резонаторы и связки имеют своё строго индивидуальное строение у каждого человека.

Проблема состоит в том, что издаваемые человеком звуки украшаются не только тембрально, но и артикуляционно, что в свою очередь создаёт особые гармонические артефакты, которые не являются полезным сигналом для распознавания ноты.

В случае распознавания человеческого голоса, звуковая волна находится в определённом тональном интервале, но тембральная составляющая может звучать совершенно в другом. Это влечёт за собой проблему ложного определения ноты, поскольку тембральный окрас иногда звучит с большей амплитудой, нежели неокрашенный голос.

Тесты проводились на двух инструментах и голосе человека. Распознавание звука давало не корректный результат в случае:

1. Звучания с малой амплитудой.

Звук теряется на фоне шума. Вдобавок ко всему, струны входят в состояние резонанса и издают лишние гармоники, амплитуда которых соразмерна амплитуде звучащей струны.

2. Звучание с большой амплитудой.

В этом случае амплитуды резонирующих струн настолько велики, что в силу аппаратного среза пиковых значений амплитуд, они находятся на одной громкости, и среди них невозможно выделить лидирующие.

Голос разных людей распознаётся по-разному:

1. Женские высокие (от 400 до 1000 Гц) голоса распознаются без проблем и артефактов.

2. Мужские высокие (от 300 до 900 Гц) так же без проблем.

3. Мужские низкие (от 150 до 500 Гц) могут давать ложный результат.

В программе было использовано два метода преобразования Фурье.

Классическое и быстрое, оба метода дают идентичный результат. Но несмотря на то, что сигнал не имеет никаких ограничений для дискретного преобразования Фурье, быстрое преобразование - эффективнее.

Сложность алгоритма ДПФ:  $O(N^2)$ .

Сложность алгоритма БПФ:  $O(N \log(N))$ .

Результатом тестирования стало подтверждение того, что тембральный окрас инструмента и голоса могут негативно влиять на результат работы программы.

На текущий момент система находится в разработке.

Следовательно, при каждом добавлении новых функций и возможностей происходит тестирование и отладка рабочих релизов.

### **3 Документирование**

#### **3.1 Техническое задание**

##### **3.1.1 Введение**

Приложение «Spectrum» предназначено для анализа звукового файла с дальнейшей математической обработкой и определения его нот.

##### **3.1.2 Назначение разработки**

Данное приложение разрабатывается для удобства работы со звуком.

Приложение нацелено на пользователей музыкальной индустрии. Программа позволяет ускорить получение и обработку информации, а также исключает человеческий фактор при качественном анализе композиции.

##### **3.1.3 Требования к функциональным характеристикам**

Работа с программой предполагает одно действующее лицо: Пользователь. Пользователю программный продукт должен предоставлять следующие возможности:

1. Динамический анализ.
2. Вывод полученных данных.

##### **3.1.4 Требование к надёжности**

В случае возникновения ошибок программный продукт должен выводить сообщение пользователю о своей неработоспособности, чтобы пользователь мог обратиться в службу поддержки программного продукта для устранения проблем.

В случае несерьёзных ошибок в работе программного продукта или некорректного ввода данных пользователю должно выводиться оповещение, о том, что необходимо сделать, чтобы исправить ошибку.

### **3.1.5 Требования к составу и параметрам технических средств**

Минимальные требования для корректной работы программы: необходимо иметь персональный компьютер под управлением ОС семейства Windows 7 и выше, с оперативной памятью 512 Мб, также свободного места на диске должно быть 350 Мб, процессор с частотой 1 ГГц.

Звуковая карта (встроенная или внешняя) и динамический микрофон.

Оптимальные требования для корректной работы программы: необходимо иметь персональный компьютер под управлением ОС семейства Windows 7, с оперативной памятью не менее 512 Мб, также свободного места на диске должно быть не менее 700 Мб, процессор с частотой 2 ГГц. Звуковая карта (встроенная или внешняя) и конденсаторный микрофон.

### **3.1.6 Информационная и программная совместимость**

Программа должна функционировать под управлением ОС семейства Windows 7 и выше.

### **3.1.7 Требования к программной документации**

Программная документация должна быть представлена руководством пользователя и встроенной справочной системой.

## 3.2 Руководство пользователя

### 3.2.1 Установка и минимальные требования

Для нормальной работы с программой необходим компьютер с оперативной памятью не менее 256 Мб под управлением ОС Windows 7 или выше, свободным пространством на жёстком диске не менее 2Мб, монитором SVGA 640x480.

### 3.2.2 Управление приложением

Чтобы начать работать в приложении, нажмите на всплывающий список и выберите ваш аудиоинтерфейс, если не знаете, что выбрать нажмите «первичный драйвер записи звука». Снимок рабочего процесса представлен на рисунке 19.

Нажмите кнопку «Начать», чтобы программа начала прослушивать ваш аудиоинтерфейс, если к аудиокарте подключен микрофон, то программа сразу начнёт анализ звука как представлено на рисунке 20.

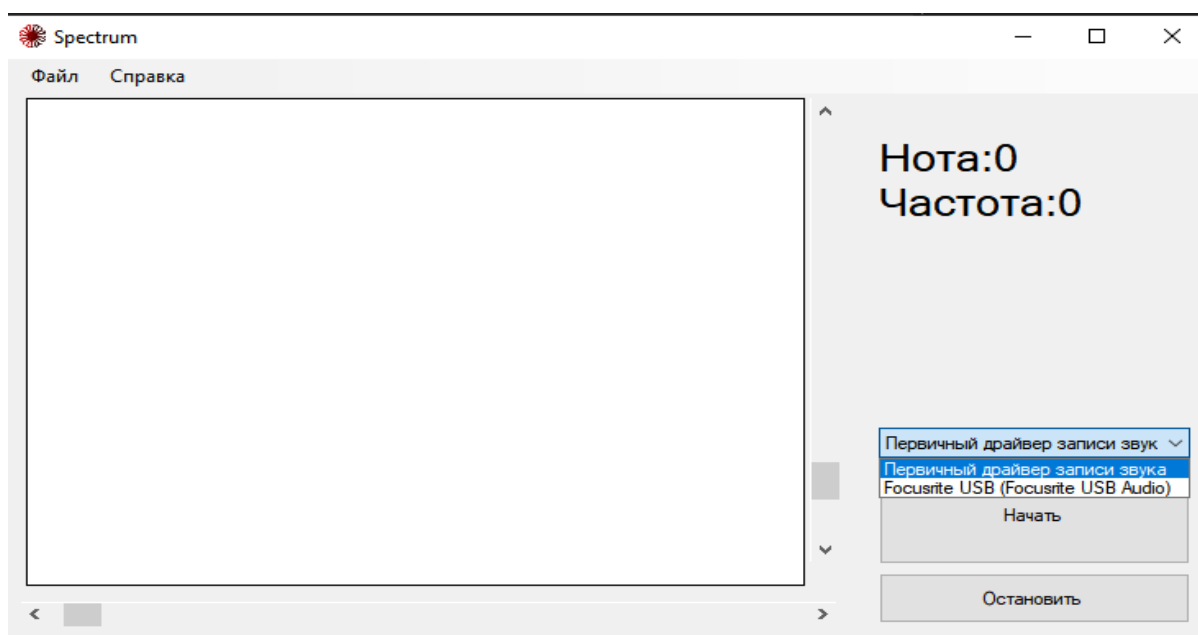


Рисунок 19 – Выбор аудиоинтерфеса

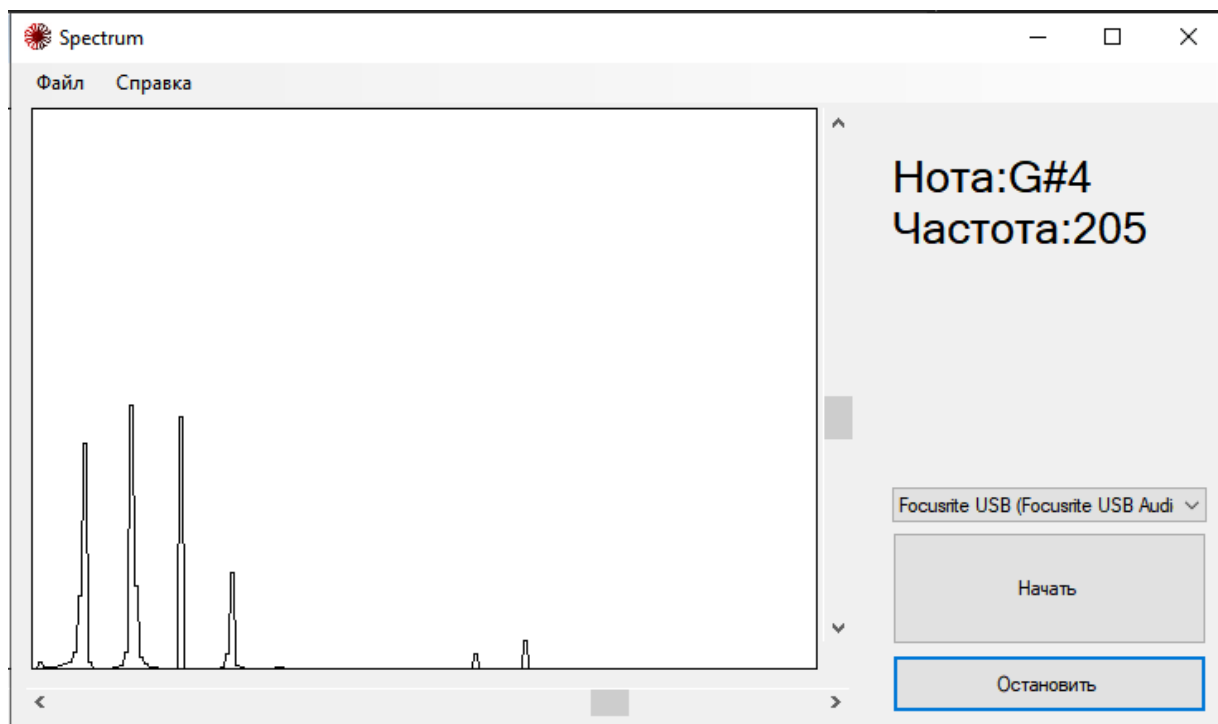


Рисунок 20 – Анализ звука

## **4 Экономическая часть**

### **4.1 Основные положения экономической части**

На сегодняшний день быстрые темпы развития музыкальной индустрии внесли множество новшеств в экономическую сферу продюсерских центров и студий звукозаписи.

В настоящий момент для анализа звука во многих студиях звукозаписи для облегчения работы звукооператора используются специализированные программы. Зачастую человеческий фактор влияет на качество работы, сведения и мастеринга композиции, поэтому необходимо использовать программный аппарат для правильного представления частотной и амплитудной картины композиции.

Изучив рынок музыкального программного обеспечения, было принято решение создать экономически менее затратное приложение сохранив основной функционал аналогов.

Любой программный продукт, разрабатываемый для музыкальных компаний, внедряется чтобы ускорить или упростить выполнение каких-либо задач. Программы сокращают работу над регулярными задачами и ускоряют выполнение несложных операций, занимающих достаточно много рабочего времени у персонала.

Разработанное приложение «Spectrum» предназначено для упрощения процесса создания музыки, её сведения и мастеринга. Программа разработана по современным стандартам и реализует необходимый функционал для комфортной работы. Имеет удобный интерфейс на русском языке и обучающие материалы.

Показатели экономической эффективности от внедрения программы определяются всеми полученными позитивными экономическими результатами и как результат: уменьшение расходов компании.

В данном разделе будут рассмотрены расчёты трудоёмкости, расчёты экономической эффективности и сценарии ценообразования.



## 4.2 Трёхуровневый анализ

Существует различные способы анализа товаров/услуг, одним из которых является трёхуровневый анализ. Данный способ предполагает выделение трёх основных компонентов, а именно: сущность продукта, фактический продукт и дополнительный продукт.

Понятие сущности продукта определяет базовую выгоду, которая интересна потребителю при покупке продукта. К этой категории относятся те потребности, ради удовлетворения которых потребитель покупает данный продукт.

Фактический продукт представляет собой ключевые характеристики, описывающие товар/услугу. Данная составляющая является базовой в определении конкурентоспособности. Дополнительный продукт включает в себя дополнительные выгоды, которые производитель может предложить покупателю сверх фактического продукта. Как правило, эти выгоды влияют на выбор товара того или иного производителя.



Рисунок 21 – Схема трёхуровневого анализа

Сущностью товара выступает частотный анализ композиции, где результатом является информация о нотной структуре мелодий.

Разработанное программное обеспечение полезно музыканту любого уровня, а для музыкально деятеля начального уровня - может оказаться незаменимо.

Данная программа для своей работы использует микрофон и позволяет решить несколько важных задач:

1. Определить тональность композиции.
2. Распознать ноты выходящие за границы тональности.
3. Найти отклонения от эталонных нот в необработанном вокале или в партии инструмента.

Такие интервалы (отклонения) не слышны человеческому уху, поэтому это полезно в звукорежиссуре при сведении нескольких инструментов в единую звуковую дорожку. Дело в том, что такие отклонения от эталонного тона приводят к грязному звучанию всей композиции. Особенно это можно проследить в басовой партии.

Фактическим продуктом является архив с исполняемым приложением. Приложение, в зависимости от модели аудиокарты и микрофона, способно обрабатывать звуковые колебания в интервале частот, который предоставляет их производитель в общем случае это от 20 до 20000 Гц, диапазон, который способен слышать человек.

Программа обладает простым интерфейсом на русском языке.

В качестве дополнительного продукта можно выделить документацию и обучающие материалы.

Схема трёхуровневого анализа программного продукта представлена на рисунке 21.

#### **4.3 Определение трудоёмкости разработки программного продукта**

Задача написания экономической части – определение трудоёмкости разработки, стоимости и эффективности использования программного продукта определенной сфере деятельности. Основой определения трудоемкости в области разработки программного обеспечения являются

нормы, принятые в области компьютерных информационных технологий. Такими нормами являются нормы времени, служащие для определения нормы труда специалистов, разрабатывающих программное обеспечение для электронно–вычислительных машин, обоснования трудности разработки программного обеспечения, а также численности разработчиков. Эти нормы включают в себя статические задачи, комплексы задач всех систем и подсистем управления, а также различные задачи, связанные с расчетами трудоемкости.

В ходе расчета трудоемкости выделяется 5 стадий выполнения работ.

Техническое задание. Это стадия, на которой заказчиком формируются основные требования к автоматизированной системе. После формирования требований производится обоснование возможности решения задачи, производится разработка ведущей концепции и согласование сроков разработки.

Эскизный проект. На данной стадии происходит дальнейшая проработка технического задания, по итогам которой разрабатывается математическая модель и создается алгоритм разработки автоматизированной системы.

Технический проект. На этой стадии происходит разработка программной документации и выбор наиболее эффективных средств реализации

Рабочий проект. Наиболее трудоемкая стадия, во время которой выполняется программная реализация, ее тестирование и отладка. Результатом по завершении этой стадии служит готовая автоматизированная система, сопровождаемая рабочей документацией и руководством пользователя.

Стадия внедрения включает себя проверку правильности работы автоматизированной системы на практике в процессе подготовки требуемой документации, а после автоматизированная система может эксплуатироваться пользователями.

Для расчета трудозатрат программиста при разработке данного программного продукта следует определить степень новизны.

У разрабатываемых комплексов задач существует четыре степени новизны:

1. Степень А – это разработка комплекса задач, включающих применение совершенно новых методов разработки, а также осуществление научно-исследовательских работ.

2. Степень Б – это разработка принципиально новых проектных решений, задач, методов и систем, не имеющих аналогов.

3. Степень В – это разработка проекта с использованием уже существующих аналогичных проектных решений с условием их изменения.

4. Степень Г – привязка готовых проектных решений к собственному проекту.

Разрабатываемое ПО соответствует степени новизны В.

На стадии «Техническое задание» расход времени на выполнение задачи обозначается за  $T_1$ . В свою очередь  $T_1$  имеет значение равное 42 (чел./дн.) [1].

На стадии «Эскизный проект» расход времени на выполнение задачи обозначается за  $T_2$ . В свою очередь  $T_2$  имеет значение равное 53 (чел./дн.) [1].

Для остальных этапов («Технический проект», «Рабочий проект» и «Внедрение») предварительные нормы затрат времени определяются по формуле (22):

$$T_{3,4,5} = r_1 \Phi_1^{r_2} \Phi_2^{r_3} \text{ (чел./дн.)}, \quad (22)$$

где  $r_1, r_2, r_3$  – коэффициенты, приведенные в таблицах 3 – 6 [1];

$\Phi_1$  – количество макетов входной информации;

$\Phi_2$  – количество разновидностей форм выходной информации.

Коэффициенты  $r_1, r_2, r_3$  принимают следующие значения в соответствии с таблицей 1.

Так как разработчик комплекса участвовал в подготовке

информационного обеспечения, на стадиях «Технический проект» и «Рабочий проект» коэффициенты  $r_1$ ,  $r_2$ ,  $r_3$  были увеличены в 1,1 раз (повышающий коэффициент).

Таблица 1 – Значения коэффициентов  $r_1$ ,  $r_2$ ,  $r_3$

Стадия разработки	$r_1$	$r_2$	$r_3$
Технический проект ( $T_3$ )	10,26	0,51	0,16
Рабочий проект ( $T_4$ )	37,17	0,49	0,47
Внедрение ( $T_5$ )	8,1	0,45	0,52

Для разрабатываемого проекта  $\Phi_1 = 3$ , а  $\Phi_2 = 1$ .

Ниже представлен расчет норм временных затрат  $T_3$ ,  $T_4$ ,  $T_5$ .

$$T_3 = 10,26 * 3^{0,51} * 1^{0,16} = 17,96 \text{ (чел./дн.)};$$

$$T_4 = 37,17 * 3^{0,49} * 1^{0,47} = 63,68 \text{ (чел./дн.)};$$

$$T_5 = 8,1 * 3^{0,45} * 1^{0,52} = 13,28 \text{ (чел./дн.)}.$$

Тогда общие трудозатраты будут равны:

$$T_{\Sigma} = T_1 + T_2 + T_3 + T_4 + T_5 = 42 + 53 + 17,96 + 63,67 + 13,28 = 189,91 \text{ (чел./дн.)}.$$

Численность исполнителей, необходимая для выполнения работ по созданию сайта вычисляется по формуле (23):

$$Ч = \frac{T_{вр}}{\Phi_{пл}}, \quad (23)$$

где  $\Phi_{пл}$  – фонд рабочего времени, планируемый на одного разработчика;

$T_{вр}$  – временные трудозатраты на реализацию программы.

Программное обеспечение разрабатывается в течении девяти месяцев с сентября по май, что составляет 182 рабочих дня, соответственно рекомендуемая численность разработчиков проекта составляет:

$$Ч = 189,91 / 182 = 1,043 \approx 1 \text{ (чел.)}.$$

#### 4.4 Определение стоимости программного обеспечения

Себестоимость – это стоимостная оценка используемых в процессе производства материальных, трудовых и других затрат, а также затрат на реализацию продукции. Стоимость сайта определена на основе укрупненного метода учета затрат по материальным затратам, расходам на оплату труда, отчислений на социальное страхование, амортизации основных фондов и прочих расходов. Стоимость рассчитывается по формуле (24):

$$C_{\text{по}} = \text{МЗ} + \text{ФЗП} + A_0 + \text{П}, \quad (24)$$

где МЗ– материальные затраты;

ФЗП – фонд заработной платы разработчика программного продукта;

$A_0$  – амортизационные отчисления;

П – прочие расходы.

Количество рабочих часов, затраченных на разработку данного Программного обеспечения, при учете того, что проект разрабатывался в течение 182 рабочих дней, а длительность рабочего дня составляет 8 (час.), получим общее количество рабочих часов:

$$\text{КЧ} = T_{\Sigma \text{ок}} * 8 = 182 * 8 = 1456 \text{ (час.)}.$$

Материальные затраты включают в себя расходы на электроэнергию.

Определение затрат на электроэнергию ведется из расчета того, что потребление энергии компьютером составляет 0,5 кВт/ч, а стоимость 1 кВт/ч равна 4,20 (р.), тогда затраты на электроэнергию составят:

$$1456 * 0,5 * 4,20 = 3057,6 \text{ (р.)}.$$

Заработная плата определяется на основе трудового договора между работником и работодателем. Должностной оклад инженера-программиста согласно штатному расписанию ФГБОУВО «Забайкальский государственный университет» составляет 12130 (р.)

По закону Забайкальского края от 26.09.2008 № 39-33К «О районном коэффициенте и процентной надбавке к заработной плате работников бюджетных организаций» установлен приблизительный районный

коэффициент в размере 20 %, а также предусмотрена надбавка 30 % за стаж работы в районах Крайнего Севера и приравненных к ним местностях, а также в остальных районах Севера.

Таким образом, ежемесячная заработная плата составляет:

$$12130 * 1,5 = 18195 \text{ (р.)}.$$

Учитывая, что разработка программно-аппаратного комплекса ведется в течение 9 месяцев, фонд заработной платы будет равен:

$$\text{ФЗП} = 18195 * 9 = 163755 \text{ (р.)}.$$

Начиная с 01.01.2018 года, взносы на социальное страхование составляют 30 % от общей заработной платы. Законом установлены тарифы страховых взносов, описанных ниже:

- Пенсионный фонд Российской Федерации – 22,00 %;
- Фонд социального страхования Российской Федерации – 2,9 %;
- Федеральный фонд обязательного медицинского страхования – 5,1 %.

Таким образом, отчисления составили:

$$163755 * 0,3 = 49126,5 \text{ (р.)}$$

Основное средство, подлежащее амортизации при разработке программной системы – это компьютер. Первоначальная стоимость одного компьютера 25000 (р.).

Вычислительная техника входит в амортизационную группу 2. По установленным нормам, постановление Правительства Российской Федерации от 18.11.2006 № 697, время службы компьютера берется от 2 до 3 лет включительно. Установим норму службы компьютера равную 2,5 года (30 мес.).

Следовательно, амортизация рассчитывается следующим образом.

$$A = \frac{\text{Первонач. ст} - \text{ст} \times H_a}{100\%} \quad (25)$$

$$H_a = 1/30 * 100 \% = 3,33 \%$$

$$A \text{ (месяц)} = 25000 * 3,33 \% / 100 \% = 832,5 \text{ (р.)}$$

$$A \text{ (год)} = 832,5 * 12 = 9990 \text{ (р.)}$$

$$A \text{ (9 мес.)} = 832,5 * 9 = 7492,5 \text{ (р.)}$$

Величина прочих расходов составляет 15 % от основной заработной платы. В прочие расходы входит амортизация нематериальных активов, а также накладные расходы:

$$П = 18195 * 0,15 * 9 = 24563,25 \text{ (р.)}$$

Таким образом, общая стоимость затрат на создание программы составляет:

$$С_{\text{ПО}} = 3057,6 + 163755 + 49126,5 + 7492,5 + 24\,563,25 = 223431,6 \text{ (р.)}$$

Программный продукт является объектом нематериальных активов, и его первоначальная стоимость оценивается по совокупным затратам на создание, то есть равна 223431,6 (р.)

#### **4.5 Оценка эффективности программы**

Для оценки экономической эффективности рассмотрим работу музыканта в студии. Учитывая, что работа в студии крайне дорогая услуга, то в этой ситуации разрабатываемое программное обеспечение, поможет существенно снизить время пребывания в студии, что в свою очередь снизит стоимость аренды и прочих услуг.

Программа решает узкоспециализированную, но популярную задачу, поэтому пользователю не придётся приобретать дорогостоящие ПО для выполнения задачи;

Сократится время тонального анализа композиции, для не опытного пользователя (музыканта).

Рассмотрим две студийные услуги:

1. Тюнинг вокала. Средняя цена услуги 1500 (р.).

В общем случае услуга платная и не входит в стоимость пребывания в студии. На практике при использовании разрабатываемого продукта по двум причинам этой траты можно избежать вовсе:

– пользование программой позволит пользователю самостоятельно отстроить вокал (при условии, что он уже записан) зная тональность



композиции, тюнинг производится крайне просто, в домашних условиях;

– пользователь может использовать программу для тренировки своего музыкально слуха.

2. Аранжировка композиции. Цена услуги от 10000 до 30000 (р.).

Разрабатываемая программа, способна распознавать мелодию звукового файла, поэтому пользователь может подбирать мелодические рисунки на любом инструменте или при помощи голоса. Это позволит заранее продумать структуру и некоторые детали аранжировки, что позволит пользователю показать аранжировщику предполагаемую структуру композиции, что снижает цену от 20 до 30 %.

Расчет (26):

$$E = C_6 - C_H \quad (26)$$

где  $E$  – экономический эффект от внедрения новой разработки;

$C_6$  – стоимость обработки экономической информации при использовании базового варианта;

$C_H$  – стоимость обработки экономической информации при предлагаемом варианте.

Пусть  $C_6 = 15000$  (р.) (средняя цена аранжировки), тогда

$$C_H = 15000 * (1 - 0,25) = 11250 \text{ (р.)}$$

$$E = 15000 - 11250 = 3750 \text{ (р.)}$$

Расчет экономической эффективности разрабатываемого программного обеспечения в сравнении со схожими ПО (по цене), по формуле (27).

$$E = \text{СЦ} - \text{ЦПО} \quad (27)$$

где  $\text{СЦ}$  - средняя цена схожих ПО;

$\text{ЦПО}$  -цена разрабатываемого программного обеспечения.

Цена схожего ПО среди самых популярных разработчиков варьируется от 6000 (р.) (Ableton) до 73000 (р.) (FL Studio).

$$\text{СЦ} = \frac{(4000 + 63000)}{2} = 33500 \text{ (р.)}$$

В зависимости от сценария ценообразования таблицы 2. Цена разрабатываемого программного обеспечения может быть разной.

Основываясь на данных, представленных в таблице, цена за одну копию программы, в пессимистическом сценарии, слишком высока.

Оптимистический же и рационалистический сценарии, предполагают вполне подходящую стоимость для целевой аудитории.

$$\text{ЦПО1} = 223431,6 * 1,25 / 500 = 558,5 \text{ (р.)}$$

$$\text{ЦПО2} = 223431,6 * 1,05 / 200 = 1173 \text{ (р.)}$$

$$E1 = 33500 - 558,5 = 32941 \text{ (р.)}$$

$$E2 = 33500 - 1173 = 32327 \text{ (р.)}$$

Таблица 2 – Сценарий ценообразования

Сценарий ценообразования	Описание	Цена за одну копию
Пессимистический	Спрос – низкий. Ожидаемое количество реализованных штук равно 1. Рентабельность равна 0 %	223431,6 (р.)
Оптимистический	Спрос – высокий. Ожидаемое количество реализованных штук равно 500. Рентабельность равна 25 %	$223431,6 * 1,25 / 500 = 558,5 \text{ (р.)}$
Рационалистический	Спрос – средний. Ожидаемое количество реализованных штук равно 200. Рентабельность равна 5 %	$223431,6 * 1,05 / 200 = 1173 \text{ (р.)}$

Разработанное приложение обладает всеми необходимыми экономическими преимуществами для конкуренции с аналогами: доступная цена и необходимый для работы функционал.

## **5 Охрана труда**

### **5.1 Общие положения охраны труда**

Гигиенические требования к персональным электронно-вычислительным машинам (ПЭВМ) и организации работы с ними контролируются санитарно-эпидемиологическими правилами и нормами СанПиН 2.2.2/2.4.1340-03 (далее – санитарные правила) [2]. Настоящие санитарные правила разработаны в соответствии с Федеральным законом «О санитарно-эпидемиологическом благополучии населения» от 30.03.1999 г. № 52-ФЗ и Положении о государственном санитарно-эпидемиологическом нормировании, утвержденным постановлением Правительства Российской Федерации от 24.07.2000 г. №554.

Санитарные правила действуют на всей территории Российской Федерации и устанавливают требования к ПЭВМ и условиям труда. Требования санитарных правил направлены на предотвращение неблагоприятного влияния вредных факторов производства и трудового процесса при работе с ПЭВМ на здоровье человека.

Требования санитарных правил распространяются на: организацию и условия работы с ПЭВМ, персональные и портативные ЭВМ, устройства отображения информации, периферийные устройства и игровые комплексы на базе ПЭВМ.

### **5.2 Анализ достоинств интерфейса**

Разрабатываемый программный продукт для взаимодействия с пользователем оснащен графическим интерфейсом, что упрощает работу пользователя с программой. Важно отметить, что в программе сведена к минимуму работа по вводу информации с клавиатуры, что существенно уменьшает время, потраченное на выполнение однотипных операций с данными, таких как загрузка файлов, а также их анализ.

В случае, если пользователь допустил ошибку в действиях, например, выбрал неправильный формат файла, то система автоматически оповестит пользователя об ошибке в окне информации. От пользователя будет требоваться только исправить введенные данные.

### **5.3 Рекомендации по организации рабочего места**

При размещении рабочих мест с ПЭВМ расстояние между рабочими столами с мониторами администраторов, должно быть не менее 2,0 м, а расстояние между боковыми поверхностями мониторов – не менее 1,2 м.

Рабочие места с ПЭВМ в помещениях с источниками вредных факторов производства должны находиться в закрытых кабинах с установленным воздухообменом.

Экран монитора должен находиться на расстоянии от 600 мм до 700 мм от глаз пользователя.

Рабочая поверхность стола, используемого для работы, должна обеспечивать оптимальное размещение используемого оборудования, поддержания правильной позы для работы за столом при работе на ПЭВМ, а также позволять менять позу с целью снижения статического напряжения мышц шейно-плечевого отдела и спины для предупреждения усталости. Поверхность рабочего стола должна иметь приблизительный коэффициент отражения от 0,5 до 0,7.

Тип рабочего кресла должен выбираться с учётом показателей роста пользователя, продолжительности работы с ПЭВМ и характера работы. Рабочее кресло должно быть оборудовано подъёмно-поворотным механизмом, позволяющего регулировать высоты и угол наклона сиденья и спинки, а также изменять расстояние спинки от переднего края сиденья. При этом настройка параметров кресла должна быть независимой друг от друга и легко осуществляемой, а также иметь надёжную фиксацию.

Поверхность рабочего места должна быть полумягкой, с нескользящей, слабо электризующейся и воздухопроницаемым покрытием, обеспечивающим легкую чистку материала от загрязнений.

Поверхности рабочих столов, также должны регулироваться по высоте с надёжной фиксацией в заданном положении. Высота рабочего стола для взрослого пользователя должна быть в пределах от 680 до 880 мм; при отсутствии возможности регулирования высота рабочей поверхности должна составлять 725 мм.

Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной не менее 500 мм, глубиной на уровне колен – не менее 450 мм и на уровне вытянутых ног – не менее 650 мм.

Рабочее место пользователя ПЭВМ следует оборудовать подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до 20 градусов. Поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм.

Обязательно должна проводиться ежедневно влажная уборка и проветривание после каждого часа работы ПЭВМ.

#### **5.4 Требования к ПЭВМ и работе с ними**

ПЭВМ должны соответствовать требованиям настоящих санитарных правил. Каждый их тип подлежит обязательной санитарно-эпидемиологической экспертизе. Данная экспертиза должна проводиться в специальных аккредитованных испытательных лабораториях.

Список контролируемых гигиенических параметров опасных и вредных факторов представлен в таблице 3.

Допустимые уровни звукового давления и звука, создаваемые ПЭВМ, не должны превышать значений, указанных в таблице 4. Все измерения проводятся на расстоянии 50 см от поверхности оборудования.

Таблица 3 – Контролируемые гигиенические параметры

№	Вид продукции	Код ОКП	Контролируемые гигиенические параметры
1	Персональные и портативные ЭВМ	40 1300, 40 1350, 40 1370	Уровни электромагнитных полей (ЭМП), акустического шума, концентрации вредных веществ в воздухе, визуальные показатели ВДТ, мягкое рентгеновское излучение
2	Периферийные устройства	40 3000	Уровни ЭМП, акустического шума, концентрация вредных веществ в воздухе
3	Устройства отображения информации	40 3200	Уровни ЭМП, визуальные показатели, концентрация вредных веществ в воздухе, мягкое рентгеновское излучение
4	Игровые автоматы на базе ПЭВМ	96 8575	Уровни ЭМП, акустического шума, концентрация вредных веществ в воздухе, визуальные показатели ВДТ, мягкое рентгеновское излучение

Таблица 4 – Допустимые значения уровней звукового давления

Уровни звукового давления в октановых полосах со среднегеометрическими частотами									Уровни звука в дБА
31,5Гц	63Гц	125Гц	250Гц	500Гц	1000Гц	2000Гц	4000Гц	8000Гц	
86 дБ	71 дБ	61 дБ	54 дБ	49 дБ	45 дБ	42 дБ	40 дБ	38 дБ	50

Временные допустимые уровни (ВДУ) электромагнитных полей (ЭМП), создаваемые ПЭВМ, не должны превышать значений, указанных в таблице 5.

Таблица 5 – Допустимые уровни ЭМП

Наименование параметров		ВДУ ЭМП
Напряженность электрического поля	в диапазоне частот с 5 Гц до 2 кГц	25 В/м
	в диапазоне частот с 2 кГц до 400 кГц	2,5 В/м
Плотность магнитного потока	в диапазоне частот с 5 Гц до 2 кГц	250 нТл
	в диапазоне частот с 2 кГц до 400 кГц	25 нТл
Электростатический потенциал экрана видеомонитора		500 В

Допустимые параметры устройств отображения информации указаны в таблице 6. Для дисплеев на электронно-лучевых трубках частота обновления экрана не должна быть ниже 75 Гц при любом разрешении.

Для остальных типов дисплеев частота обновления экрана должна быть не менее 60 Гц.

Таблица 6 – Допустимые параметры устройств отображения информации

№	Параметр	Допустимые значения
1	Яркость белого поля	Не менее 35 кд/м <sup>2</sup>
2	Неравномерность яркости рабочего поля	Не более $\pm 20 \%$
3	Контрастность (для монохромного режима)	Не менее 3:1
4	Временная нестабильность изображения	Не должна фиксироваться
5	Пространственная нестабильность изображения	Не более $2 \cdot 10^{-4} L$ , где L – проектное расстояние наблюдения, мм

Концентрация вредных веществ, выделяемых ПЭВМ, не должна превышать предельно допустимую концентрацию, установленную для атмосферного воздуха. Мощность мягкого рентгеновского излучения в любой точке на расстоянии 0,05 м от экрана ПЭВМ и его корпуса не должна превышать 1 мкЗв / час.

Корпус ПЭВМ должен быть окрашен в мягких и спокойных тонах с диффузным рассеиванием света. ПЭВМ и периферийные устройства должны иметь матовую поверхность с отсутствием деталей способных создавать блики. При этом коэффициент отражения поверхности должен варьироваться от 0,4 до 0,6 условных единиц.

### 5.5 Требования к освещению рабочих мест

На рабочих местах столы следует располагать таким образом, чтобы дисплеи ПЭВМ были ориентированы боковой стороной к световым проёмам, чтобы естественные свет преимущественно падал с левой стороны от рабочего места.

В помещениях, в которых используется ПЭВМ, с помощью системы общего равномерного освещения обеспечивается искусственное освещение.

В производственных и административно-общественных помещениях, при преимущественной работе с документами, следует применять системы комбинированного освещения. В этом устанавливаются дополнительно светильники местного освещения для зоны работы с документами.

Искусственное освещение должно обеспечивать требуемую освещённость на рабочих местах не ниже нормируемых значений: экран – не более 300 лк; клавиатура и рабочий стол от 300 до 500 лк [2].

Следует свести к минимуму прямую блёскость от источников освещения и отражённую блёскость на поверхностях таких как экран, стол или клавиатура. Уменьшение отражённой блёскости достигается за счет верного выбора типов электроосветительных приборов и схемы расположения рабочих мест по отношению к источникам освещения. При этом яркость потолка, а также светящихся поверхностей, таких как окна или светильники, находящиеся в поле зрения пользователя при работе, не должны быть выше  $200 \text{ кд/м}^2$ , а яркость бликов на дисплее не должно превышать  $40 \text{ кд/м}^2$ .

Неравномерное распределение яркости в поле зрения пользователя ПЭВМ следует свести к минимуму. В областях между рабочими поверхностями соотношение яркости должно быть меньше, чем три к одному, а между рабочими поверхностями, стенами и оборудованием ниже, чем десять к одному.

Для искусственного освещения следует применять преимущественно люминесцентные лампы белого света и компактные люминесцентные лампы. При устройстве отражённого освещения в производственных и административно-общественных помещениях допускается применение металлогалогенных ламп. В светильниках местного освещения допускается применение ламп накаливания, в том числе галогенные.

Для освещения помещений с ПЭВМ следует применять светильники с зеркальными параболическими решетками, укомплектованными электронными пускорегулирующими аппаратами (ЭПРА). Допускается



использование многоламповых светильников с ЭПРА, состоящих из равного числа опережающих и отстающих ветвей. Применение светильников без рассеивателей и экранирующих решеток не допускается. При отсутствии светильников с ЭПРА лампы многоламповых светильников или рядом расположенные светильники общего освещения следует включать на разные фазы трёхфазной сети.

Общее освещение при использовании люминесцентных светильников следует выполнять в виде сплошных или прерывистых линий светильников, расположенных сбоку от рабочих мест, параллельно линии зрения пользователя при рядном расположении дисплеев.

При периметральном расположении компьютеров линии светильников должны располагаться локализовано над рабочим столом ближе к его переднему краю, обращённому к оператору.

Коэффициент запаса для осветительных установок общего освещения должен приниматься равным 1,4 условных единиц. Коэффициент пульсации не должен превышать 5 %.

Для обеспечения нормируемых значений освещённости в помещениях для использования ПЭВМ следует проводить чистку стёкол, оконных рам и светильников не реже двух раз в год и проводить своевременную замену перегоревших ламп.

## **5.6 Рекомендации по организации труда и отдыха**

В зависимости от сложности работы установление числа и длительности перерывов происходит следующим образом:

- для группы А не более 60000 считываемых знаков во время смены перерыв составляет 15 минут, предоставляется 2 раза через два часа после начала работы и перерыва на обед;

- для группы Б не более 40000 вводимых знаков во время смены перерыв составляет 10 минут через каждый трудовой час;

– для группы В не более шести рабочих часов за смену перерыв составляет 15 минут через каждый трудовой час.

Если рабочая смена длится двенадцать часов, время регламентированных перерывов при работе на компьютере за восемь часов работы предоставляется в вышеуказанном порядке за оставшиеся четыре часа – 15 минут за каждый рабочий час вне зависимости от категории.

Во время перерывов следует выполнять специальную гимнастику для снятия напряжения с глаз. Рекомендуемый комплекс упражнений находится в санитарных правилах и нормах.

### **5.7 Требования к электробезопасности**

Во избежание поражения электрическим током необходимо оборудовать помещение, в котором происходит рабочий процесс с установленными ПЭВМ, розетками с заземлением, либо отдельным контуром заземления, на который должны быть заземлены ПЭВМ и все устройства, подключаемые к ним.

### **5.8 Требования к пожарной безопасности**

Наиболее частыми причинами пожаров являются нарушения правил пожарной безопасности и технологических процессов, неправильная эксплуатация электросети и оборудования, грозовые разряды.

Анализ пожаров, возникших из-за нарушения правил пользования электрическими приборами, показывает, что они происходят в основном по двум причинам: из-за нарушения правил при пользовании электробытовыми приборами и скрытой неисправности этих приборов или электрических сетей.

При прохождении тока по проводнику выделяется тепло. В обычных, условиях оно рассеивается в окружающую среду быстрее, чем успевает нагреться проводник.

Поэтому для каждой электрической нагрузки соответственно подбирается проводник определенного сечения. Пожар, возникший из-за неисправности электропроводки или неправильная эксплуатация электросети, обуславливается тем, что если сечение проводника меньше, чем положено по расчету, то выделяющееся тепло не успевает рассеяться и проводник перегревается. Также при включении в одну розетку одновременно несколько бытовых приборов возникает перегрузка, нагрев проводов.

Все средства оповещения о пожаре делятся на речевые, световые и звуковые.

Речевое оповещение о пожаре предполагает трансляцию коротких сообщений, объясняющих ситуацию и дающих инструкции по эвакуации. Это самый действенный способ управления людьми в случае ЧС. Записанные заранее сообщения должны быть краткими, предельно ясными и понятными всем.

Звуковое оповещение о пожаре – это тревожная сирена или особый сигнал, услышав который, люди должны немедленно покинуть здание.

Световое оповещение о пожаре включает в себя светящиеся указатели выхода и иногда – указатели направления эвакуации.

В случае возникновения возгорания или пожара, руководитель обязан:

1. Немедленно сообщить об этом по телефону 01 или 112.
2. Организовать оповещение персонала.
3. Принять (по возможности) меры по эвакуации людей.
4. Организовать встречу подразделений пожарной охраны.
5. При необходимости отдать указание об отключении электроэнергии.
6. Организовать проверку включения в работу автоматических систем противопожарной защиты.

7. По прибытии пожарного подразделения проинформировать руководителя тушения пожара.

### **5.9 Меры оказания первой медицинской помощи при поражении электрическим током**

Если пострадавший соприкасается с токоведущими частями, необходимо быстро освободить его от действия электрического тока. Прикасаться к человеку, находящемуся под напряжением, опасно для жизни. Поэтому нужно быстро отключить ту часть установки, которой касается пострадавший. Для освобождения пострадавшего от провода следует воспользоваться сухой одеждой, доской или каким-либо другим предметом, не проводящему электрический ток или взяться за его одежду, если она сухая, избегая при этом прикосновения к металлическим предметам и открытым частям тела.

Далее необходимо:

1. Уложить пострадавшего на спину на твердую поверхность.
2. Проверить наличие у пострадавшего дыхания, далее определить по подъему грудной клетки, запотеванию зеркала и пр.
3. Проверить наличие пульса на лучевой стороне у запястья или на сонной артерии на переднебоковой поверхности шеи.
4. Выяснить состояние зрачка, широкий зрачок указывает на резкое ухудшение кровоснабжения мозга.
5. Вызов врача по телефону 03 во всех случаях обязателен.

## ЗАКЛЮЧЕНИЕ

Разработанное программное обеспечение позволяет получить тональную информацию записываемого аудио в реальном времени, что в свою очередь открывает большой спектр возможностей для музыканта таких как: определение тональности музыкального отрывка, аранжировка и последующий анализ композиции.

Данное программное обеспечение будет крайне полезно начинающим музыкантам для практики и анализа собственных работ на выявление недостатков. Для опытных музыкантов программа тоже может быть полезна.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Андреев Г.И. Практикум по оценке интеллектуальной собственности: учебное пособие / Г.И. Андреев, В.В. Витчинка. – Москва: Финансы и статистика, 2003. – 176 с.
2. Гигиенические требования к персональным электронно - вычислительным машинам и организации работы. Санитарно-эпидемиологические правила и нормативы СанПиН 2.2.2/2.4.1340-03.
3. Accord.NET (Фреймворк) [Электронный ресурс] / Режим доступа: <http://accord-framework.net/>\_(игровой\_движок) - Загл. с экрана.
4. Дискретное преобразование Фурье [Электронный ресурс] / Режим доступа: [https://en.wikipedia.org/wiki/Discrete\\_Fourier\\_transform](https://en.wikipedia.org/wiki/Discrete_Fourier_transform) - Загл. с экрана.
5. Быстрое преобразование Фурье [Электронный ресурс] / Режим доступа: [https://en.wikipedia.org/wiki/Fast\\_Fourier\\_transform](https://en.wikipedia.org/wiki/Fast_Fourier_transform) - Загл. с экрана.
6. C# in Depth / Jon Skit – IBM Corporation. – 2008.
7. Clean Code / Roberto Ierusalimschy – 4<sup>th</sup> ed. – Robert Sesil Martin Digital Book Distribution, 2008

## ПРИЛОЖЕНИЕ А

(обязательное)

### Реализация класса Fourier Transform

```
using System;
using Accord.Compat;
using System.Numerics;
using Accord.Math.Transforms;

public static class FourierTransform
{
    public enum Direction
    {
        Forward = 1,
        Backward = -1
    };

    public static void DFT(Complex[] data, Direction direction)
    {
        int n = data.Length;
        double arg, cos, sin;
        var dst = new Complex[n];
        for (int i = 0; i < dst.Length; i++)
        {
            dst[i] = Complex.Zero;
            arg = -(int)direction * 2.0 * System.Math.PI * (double)i / (double)n;
            for (int j = 0; j < data.Length; j++)
            {
                cos = System.Math.Cos(j * arg);
                sin = System.Math.Sin(j * arg);
                double re = data[j].Real * cos - data[j].Imaginary * sin;
                double im = data[j].Real * sin + data[j].Imaginary * cos;
                dst[i] += new Complex(re, im);
            }
        }

        if (direction == Direction.Forward)
        {
            for (int i = 0; i < data.Length; i++)
                data[i] /= n;
        }
        else
        {
            for (int i = 0; i < data.Length; i++)
                data[i] = dst[i];
        }
    }
}
```

Рисунок А.1 – Дискретное преобразование Фурье

```

public static void FFT(Complex[] data, Direction direction)
{
    int n = data.Length;
    int m = Tools.Log2(n);
    ReorderData(data);
    int tn = 1, tm;

    for (int k = 1; k <= m; k++)
    {
        Complex[] rotation = FourierTransform.GetComplexRotation(k, direction);
        tm = tn;
        tn <<= 1;
        for (int i = 0; i < tm; i++)
        {
            Complex t = rotation[i];
            for (int even = i; even < n; even += tn)
            {
                int odd = even + tm;
                Complex ce = data[even];
                Complex co = data[odd];
                double tr = co.Real * t.Real - co.Imaginary * t.Imaginary;
                double ti = co.Real * t.Imaginary + co.Imaginary * t.Real;
                data[even] += new Complex(tr, ti);
                data[odd] = new Complex(ce.Real - tr, ce.Imaginary - ti);
            }
        }
    }

    if (direction == Direction.Forward)
    {
        for (int i = 0; i < data.Length; i++)
            data[i] /= (double)n;
    }
}

```

Рисунок А.2 – Быстрое преобразование Фурье

```

private static Complex[] GetComplexRotation(int numberOfBits, Direction direction)
{
    int directionIndex = (direction == Direction.Forward) ? 0 : 1;
    if (complexRotation[numberOfBits - 1, directionIndex] == null)
    {
        int n = 1 << (numberOfBits - 1);
        double uR = 1.0;
        double uI = 0.0;
        double angle = System.Math.PI / n * (int)direction;
        double wR = System.Math.Cos(angle);
        double wI = System.Math.Sin(angle);
        double t;
        Complex[] rotation = new Complex[n];

        for (int i = 0; i < n; i++)
        {
            rotation[i] = new Complex(uR, uI);
            t = uR * wI + uI * wR;
            uR = uR * wR - uI * wI;
            uI = t;
        }
        complexRotation[numberOfBits - 1, directionIndex] = rotation;
    }
    return complexRotation[numberOfBits - 1, directionIndex];
}

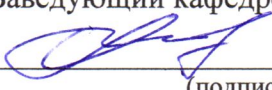
```

Рисунок А.3 – Метод поворота комплексного числа



УТВЕРЖДАЮ

Заведующий кафедрой ИВТ и ПМ

  
О.В. Валова  
(подпись, И.О.Ф)

«22» 06 2020 г.

### Заключение

о результатах проверки выпускной квалификационной работы в системе  
«Антиплагиат»

В выпускной квалификационной работе обучающегося

Молодого Никита Алексеевич

(фамилия, имя, отчество)

Энергетического факультета группы ВМК-16


Система для распознавания мелодии и генерация ее нотной записи

(название выпускной квалификационной работы)

доля оригинального текста составляет 61,36 процентов.

Расширенный отчет об источниках, с которыми были обнаружены совпадения  
фрагментов текста работы, находится на кафедре ИВТ и ПМ.

Менеджер кафедры

  
(подпись)

Р.С. Долгих  
(И.О.Ф.)

Дата «22» 06 2020 г.

**Отзыв**  
**о работе выпускника Забайкальского государственного университета**

Молодоженова Никиты Алексеевича

(фамилия, имя, отчество)

по выполнению выпускной квалификационной работы по направлению подготовки \_\_\_\_\_  
09.03.01 Информатика и вычислительная техника

Тема Система для распознавания мелодии и генерация её нотной записи

Объем работы:

количество листов пояснительной записки 60 стр.

количество листов чертежей \_\_\_\_\_

количество схем \_\_\_\_\_

Заключение о степени соответствия выполненной работы заданию работа выполнена в соответствии с техническим заданием

Проявленная выпускником самостоятельность при выполнении работы работа выполнена самостоятельно

Плановость и дисциплинированность в работе работа выполнена в соответствии с календарным планом

Умение пользоваться литературным, справочным, нормативным материалом при выполнении работы использовалась специальная литература, материалы сети Интернет

Умение представить результаты выпускной квалификационной работы (в виде плакатов, мультимедийной презентации и др.) результаты работы представлены в виде презентации в формате Microsoft PowerPoint

Индивидуальные способности выпускника проявлена полная самостоятельность при выполнении работы

Положительные стороны работы в работе решена задача, имеющая практическое значение

Недостатки работы имеется ряд замечаний по реализации работы

Характеристика общенаучной специальной подготовки выпускника подготовка в целом соответствует требованиям специальности

Заключение и предполагаемая оценка работа выполнена на уровне, соответствующем требованиям специальности, и заслуживает оценки «удовлетворительно» при соответствующей защите

Руководитель \_\_\_\_\_ Д.А. Макаров /

Дата « 19 » июня 2020 г.

