

РЕАЛИЗАЦИЯ ЦИФРОВЫХ КОМПЕТЕНЦИЙ В МУЗЫКАЛЬНОМ ОБРАЗОВАНИИ СТУДЕНТОВ ВУЗОВ НА ПРИМЕРЕ ИСПОЛЬЗОВАНИЯ ЗВУКОВЫХ БИБЛИОТЕК ЯЗЫКА PYTHON В КУРСЕ «ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ»

Герасимов А.Д., Фирсова С.А.

ФГБОУ ВО «Мордовский государственный университет им. Н.П. Огарёва», Саранск,
e-mail: gera13sa@gmail.com, karpushkinasa@yandex.ru

В соответствии с требованиями профессионального стандарта «Педагог дополнительного образования детей и взрослых», вступившего в силу в 2021 году, неотъемлемым условием выполнения трудовых действий для такого специалиста является владение информационными технологиями. Поэтому возникла необходимость формирования у будущих педагогов, в том числе педагогов-музыкантов, компетенций, связанных с использованием средств цифровых технологий, так называемых цифровых компетенций. Кроме того, в связи с широким распространением методов искусственного интеллекта и машинного обучения, а также языка программирования Python, реализующего эти методы с помощью многочисленных свободно распространяемых библиотек, стало доступно их изучение и применение на практике студентами гуманитарных специальностей. Цель исследования: разработка блока лабораторных занятий по дисциплине «Информационные технологии в профессиональной деятельности» на основе применения звуковых библиотек языка программирования Python для решения типовых задач профессиональной деятельности будущих специалистов в области музыкального образования. В частности, в статье рассмотрен пример автоматизированного составления партитуры для гитары на основе звукового анализа аудиофайла. Применение методов математического моделирования, музыкальных компьютерных технологий, современных языков программирования позволит повысить уровень цифровых компетенций студентов, а также поможет им активно применять освоенные методы в будущей профессиональной деятельности.

Ключевые слова: музыкальное образование, математическое моделирование в музыке, звуковой анализ аудиофайла, музыкальные компьютерные технологии, автоматизированное формирование партитуры, язык программирования Python

IMPLEMENTATION OF DIGITAL COMPETENCIES IN MUSIC EDUCATION OF UNIVERSITY STUDENTS ON THE EXAMPLE OF USING PYTHON SOUND LIBRARIES IN THE COURSE “INFORMATION TECHNOLOGIES IN PROFESSIONAL ACTIVITY”

Gerasimov A.D., Firsova S.A.

Ogarev Mordovian State University, Saransk,
e-mail: gera13sa@gmail.com, karpushkinasa@yandex.ru

In accordance with the requirements of the professional standard “Teacher of additional education for children and adults”, which entered into force in 2021, an essential condition for the performance of labor actions for such a specialist is the possession of information technologies. Therefore, there was a need for future teachers, including music teachers, to develop competencies related to the use of digital technologies, the so-called “digital competencies”. In addition, due to the widespread use of artificial intelligence and machine learning methods, as well as the Python programming language, which implements these methods with the help of numerous freely distributed libraries, their study and application in practice by students of humanities has become available. The purpose of the study: to develop a block of laboratory classes on the discipline “Information technology in professional activity” based on the use of sound libraries of the Python programming language to solve typical tasks of professional activity of future specialists in the field of music education. In particular, the article considers an example of automated compilation of a guitar score based on sound analysis of an audio file. The use of mathematical modeling methods, musical computer technologies, modern programming languages will increase the level of students’ digital competencies, and will also allow them to actively apply the mastered methods in their future professional activities.

Keywords: music education, mathematical modeling in music, musical computer technologies, audio file sound analysis, automated score formation, Python programming language

В настоящее время одним из наиболее эффективных средств повышения качества обучения музыкальному искусству является применение компьютерных технологий. Информационные технологии в музыкальном образовании становятся не только средством, открывающим новые формы

организации образовательного процесса, но и фактором, диктующим необходимость в изменении содержания образования [1].

Особую роль в формировании так называемых цифровых компетенций следует отнести практическим занятиям по дисциплине «Информационные технологии в про-

фессиональной деятельности» [2], которые являются важнейшим звеном в процессе профессиональной подготовки как профессиональных музыкантов, так и будущих педагогов-музыкантов в рамках реализации основной профессиональной образовательной программы по различным профилям направления подготовки 53.00.00 «Музыкальное образование», а также по направлению 44.03.01 «Педагогическое образование», профиль «Музыка».

С применением компьютерных информационных технологий стали доступны почти все известные методы моделирования, например имитационное, математическое, аналоговое, алгоритмическое моделирование [3]. В профессиональном обучении музыке разнообразное моделирование звуковой среды особенно ценно, так как не только воссоздает ранее недоступные процессы и явления, но и позволяет активно участвовать в них [3].

Получили широкое распространение методы искусственного интеллекта и машинного обучения при решении различных задач, связанных с обработкой и анализом звуковых файлов, например определение жанра музыки [4], генерация музыкальных композиций [5; 6], распознавания аккордовой последовательности [7]. Кроме того, в связи с широким распространением языка программирования Python, благодаря его простому синтаксису и множеству доступных библиотек, музыканты, не являющиеся программистами, могут использовать многочисленные аудиомодули этого языка для своей профессиональной работы при решении вышеперечисленных и других задач.

Целью данной работы является демонстрация возможностей библиотек Python при решении задачи автоматизированного формирования партитуры для гитары на основе анализа аудиофайла. Данная работа может быть использована при разработке блока лабораторных занятий по дисциплине «Информационные технологии в профессиональной деятельности», как пример решения типовых задач профессиональной деятельности будущих специалистов в области музыкального образования с помощью современных информационных и интеллектуальных технологий.

Материалы и методы исследования

В качестве материалов для проведения данного исследования выступили рабочие программы дисциплины «Информационные технологии в профессиональной деятельности», где приведен набор компетенций, в том числе и «цифровых», освоение которых должно носить системный харак-

тер. Также при подготовке приведенной в статье лабораторной работы по вышеуказанной дисциплине использовались описания функций звуковых библиотек языка программирования Python, приведенные в различных интернет-источниках.

Методами исследования, которые применяются в рамках данной работы, являются: математическое моделирование звуковой волны с помощью дискретного преобразования Фурье, алгоритмическое моделирование процесса преобразования аудиофайла в нотную запись с помощью поэтапного выполнения необходимых алгоритмов и соответствующих им функций языка Python.

Результаты исследования и их обсуждение

При подготовке студентов к выполнению лабораторной работы, направленной на закрепление студентами навыков применения алгоритмического моделирования для построения процесса автоматизированного формирования партитуры для гитары на основе анализа аудиофайла, необходимо привести небольшое теоретическое описание основных методов звукового анализа.

Так, одним из наиболее распространенных методов звукового анализа является анализ спектра частот. Этот метод позволяет определить основные частоты, которые присутствуют в звуковом сигнале, и использовать эту информацию для различных задач, например определения высоты звука или нахождения гармоник. Результаты применения анализа спектра частот, как правило, отображаются на спектрограмме, которая обычно изображается в виде тепловой карты. При этом по горизонтали откладывается время, по вертикали частоты звукового сигнала, а цветом отображается интенсивность (амплитуда) той или иной частоты в каждый момент времени.

Анализ спектра не является самостоятельным методом, он служит основой для других способов звукового анализа, которые работают с полученной спектрограммой.

Другим методом звукового анализа является метод вейвлет-анализа. Этот метод основан на представлении звукового сигнала в виде суммы вейвлетов – математических функций, которые имеют свойства анализировать сигналы в различных масштабах и частотах. Метод широко используется для анализа сложных составных звуковых сигналов, таких как музыкальные композиции, и позволяет выделить различные характеристики звука, например высоту, длительность, громкость и тембр. При изменении параметров вейвлет-анализа можно получить информацию о разных

составляющих первоначального звукового сигнала, например только информацию о высоких частотах, составляющих звуковую волну, либо только о низких.

Наиболее популярным методом звукового анализа спектра является Фурье-анализ, который основывается на преобразовании дискретных значений звуковой волны из амплитудно-временной зависимости в амплитудно-частотную зависимость и позволяет анализировать спектральный состав звукового сигнала. С этой целью в обработке звука используется дискретное преобразование Фурье (ДПФ).

Результатом ДПФ будет преобразование дискретного ряда звуковой волны в ряд фундаментальных частот, образующих эту волну. По вертикали этого ряда откладывается частота гармоник, составляющей звуковую волну, а по вертикали – относительная амплитуда. Имея данные о составляющих гармониках и их амплитудах, можно выявить главную частоту, она будет наиболее выраженной среди других.

Таким образом, комбинируя различные методы звукового анализа, можно получить наиболее полную информацию как об исполняемом произведении, так и о технике его исполнения на различных музыкальных инструментах.

На основе приведенных теоретических сведений можно построить алгоритм преобразования аудиофайла в ноты.

Этапы работы алгоритма представлены на рисунке 1.

Каждый этап данного алгоритма подразумевает реализацию на языке програм-

мирования Python, при этом в качестве среды разработки удобно использовать сервис Google Colab. Данный сервис предоставляет возможность работать с кодом, написанным на языке Python через Jupyter Notebook, при этом на свой компьютер не нужно устанавливать никаких дополнительных программ. В Google Colab можно подключать различные библиотеки, написанные для Python и расширяющие его возможности, загружать и запускать файлы, проводить анализ данных и получать результаты в браузере.

Рассмотрим основные команды, которые потребуются изучить и выполнить студентам на каждом этапе реализации алгоритма.

Вначале выполняется импорт библиотеки Librosa, которая предназначена для анализа и обработки звуковых файлов. Она позволяет осуществлять работу со звуковыми сигналами различных форматов и предоставляет множество функций для извлечения спектральных признаков и работы с музыкальными данными. Librosa очень популярна среди исследователей музыки и аналитиков звуковых данных, так как она обладает множеством полезных функций и удобным интерфейсом.

Загрузка аудиофайла осуществляется с помощью функции `load()` (рис. 2), данная функция также позволяет получить значения временного ряда, содержащего значения амплитуды звукового сигнала, которые записываются в массив, и частоту дискретизации, с которой происходит чтение аудиофайла (рис. 2).

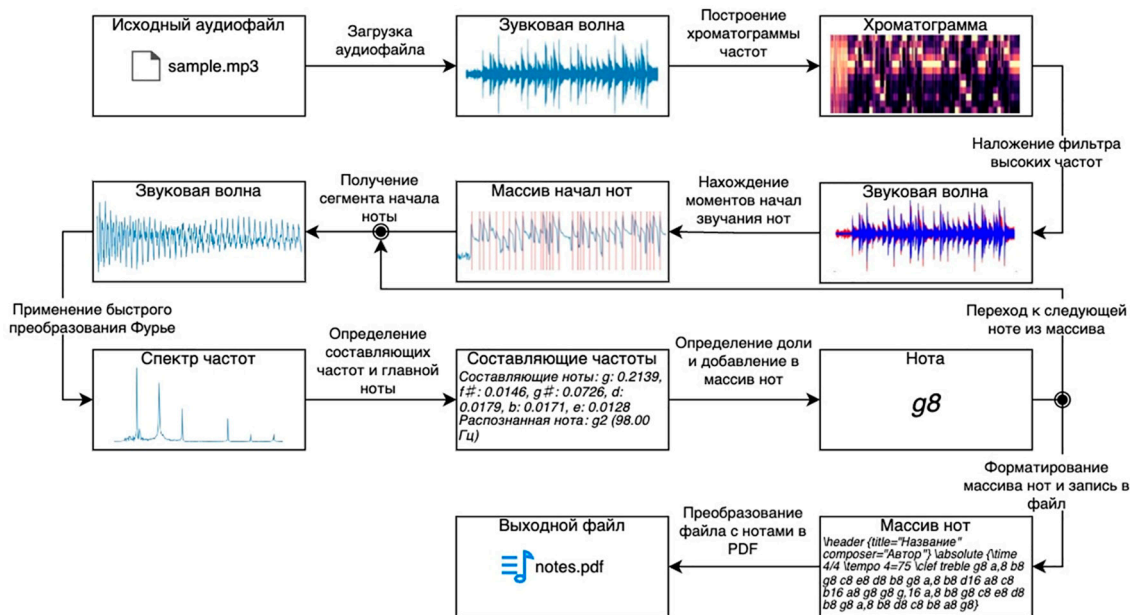


Рис. 1. Этапы работы алгоритма

```
import librosa

#Загрузка аудиофайла
file = "sample.mp3"
y, sr = librosa.load(file)
print("Временной ряд:")
print(y)
print("Частота дискретизации:")
print(sr)
```

Временной ряд:
[8.4898872e-08 1.5094116e-07 ... 1.0351254e-13 7.9779689e-14]

Частота дискретизации:
22050

Рис. 2. Значения временного ряда и частоты дискретизации

```
from IPython.display import Audio
#Воспроизведение аудиофайла
print("Прослушать файл:")
display(Audio(file))
```

Прослушать файл:

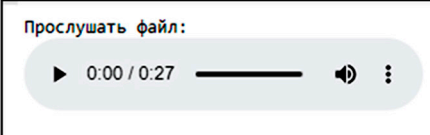


Рис. 3. Воспроизведение аудиофайла

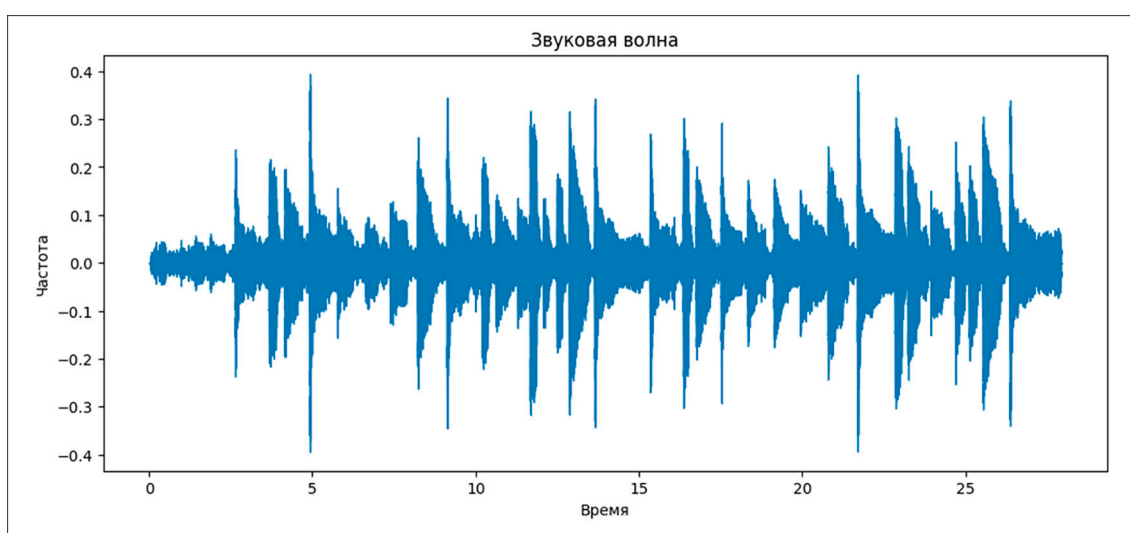


Рис. 4. График звуковой волны

Для воспроизведения аудиофайла необходимо подключить модуль Audio, затем воспользоваться функцией display() (рис. 3).

Вывод графика звуковой волны можно выполнить с помощью функции waveshow() библиотеки Librosa, в качестве параметров функции необходимо указать массив значений временного ряда и частоту дискретизации (рис. 4).

Для получения хроматограммы необходимо применить функцию chroma_stft(), тепловую карту мощностей для каждого из 7 классов нот, а затем функцию specshow() для её графического отображения (рис. 5).

Наложение фильтра высоких частот производится с помощью функции high_pass_filter() модуля AudioSegment, аргументами которой являются: 1) пороговая частота фильтрации, частоты ниже кото-

рой будут заглушаться; 2) сила фильтрации. Пример использования этой функции, а также график звуковой волны до и после проведенного преобразования приведены на рисунке 6.

Для нахождения моментов начала звучания нот выделим из аудиофайла массив сегментов с шагом 50 мс, содержащий громкости в децибелах полной шкалы, затем найдем «пики», или максимумы, в этом массиве, используя функцию find_peaks. Данная функция принимает следующие параметры: volume – массив громкостей звучания, height – пороговое значение громкости, prominence – выраженность пиков, distance – минимальное расстояние между двумя пиками. График звуковой волны с найденными моментами начала звучания нот представлен на рисунке 7.

```
# Построение хроматограммы частот аудиофайла
chromagram = librosa.feature.chroma_stft(y=y, sr=sr)
librosa.display.specshow(chromagram, y_axis='chroma', x_axis='time')
```

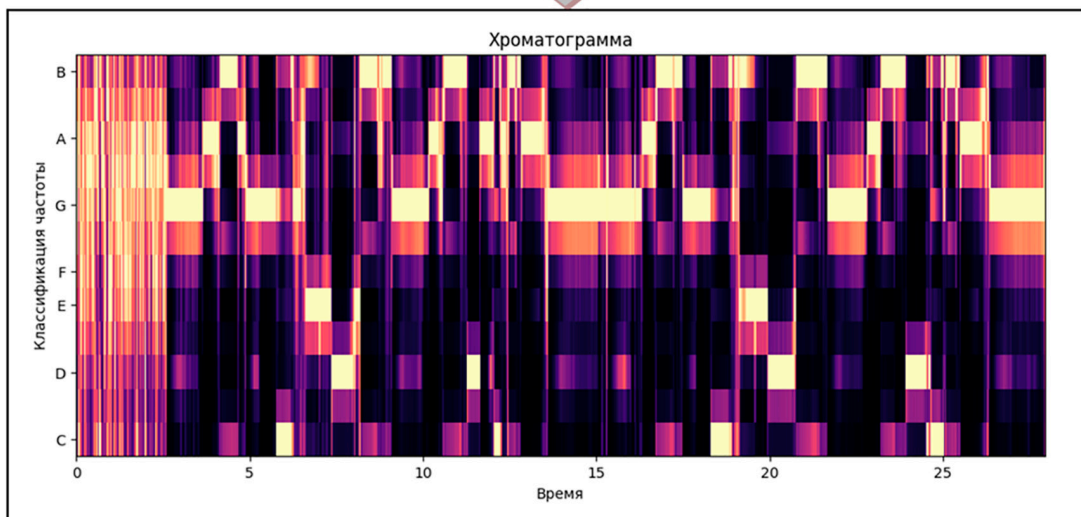


Рис. 5. Хроматограмма частот аудиофайла

```
from pydub import AudioSegment
#Получение звуковой волны и наложение фильтра высоких частот
song = AudioSegment.from_file(file)
song = song.high_pass_filter(80, order=4)
```

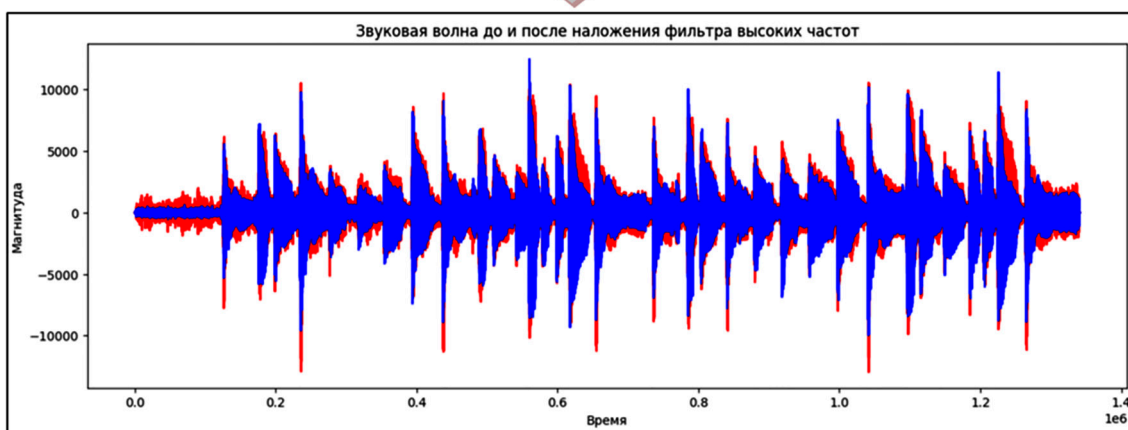


Рис. 6. Наложение фильтра высоких частот

Для каждого фрагмента аудиофайла от момента начала звучания ноты до момента следующей ноты выполняется быстрое преобразование Фурье (БПФ), которое позволяет перейти от амплитудно-временного

представления звуковой волны к её частотному спектру.

На рисунке 8 показана звуковая волна первой ноты, а на рисунке 9 – результат проведения БПФ для неё.

```
# Выделение массива громкости каждого фрагмента аудиофайла с шагом в 50 мс
volume = [segment.dBFS for segment in song[::50]]
from scipy.signal import find_peaks
# Нахождение пиков в выделенном массиве
peaks, _ = find_peaks(volume, height=-35, prominence=2.5, distance=6)
```

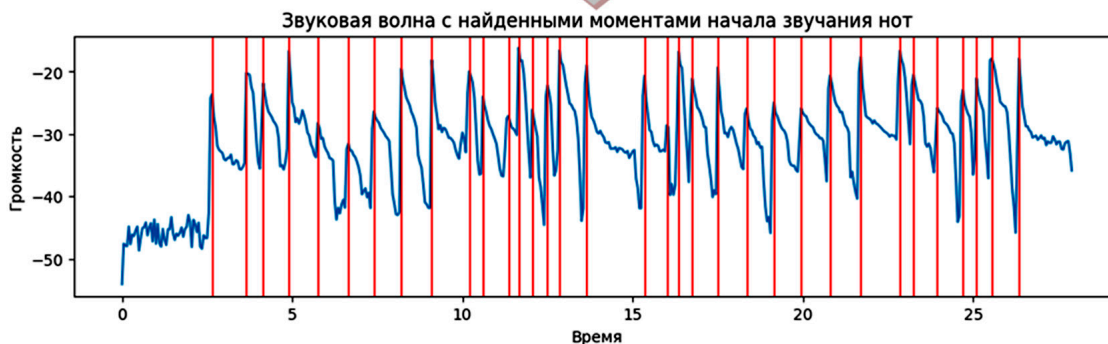


Рис. 7. Звуковая волна с найденными моментами начала звучания нот

```
# Получение сегмента аудиофайла с началом в from и концом в to
peaks=peaks*50
_from = peaks[0] + 50
_to = peaks[0] + 550
_sample = song[_from:_to]
_raw_audio_data = array.array(get_array_type(_sample.sample_width * 8), _sample.data)
```

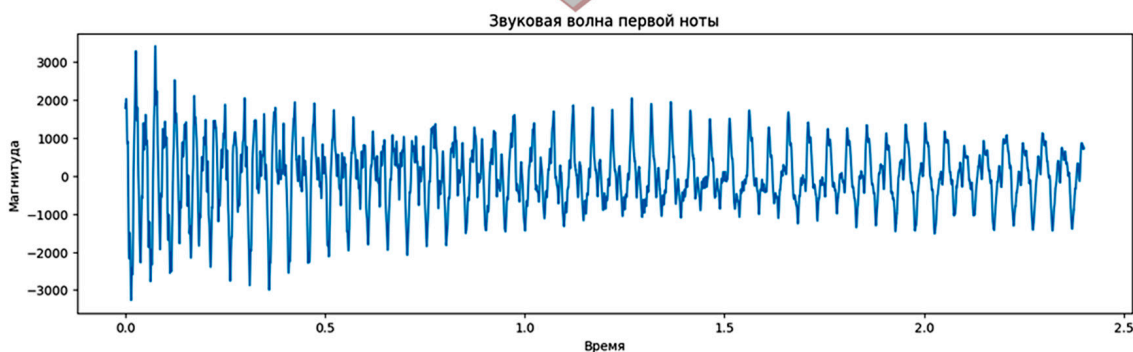


Рис. 8. Звуковая волна первой ноты

Для определения ноты и октавы на основе главной частоты в библиотеке Librosa предусмотрены соответственно функции `librosa.hz_to_note()` и `librosa.hz_to_octs()`. При нахождении знака октавы необходимо найти, насколько нота ниже или выше, чем вторая октава, и добавить к её записи соответствующее количество запятых либо

апострофов, для вычисления доли на основе темпа и длительности используется формула, в которой длительность целой ноты для данного темпа, учитывая что при 60 уд./мин. целая нота длится 4 секунды, делится на фактическую длительность ноты в секундах и округляется к ближайшей степени двойки.

```
# Получение массива магнитуд составных частот в сегменте
# при помощи быстрого преобразования Фурье
freq_magnitude = scipy.fft.fft(_raw_audio_data)
```

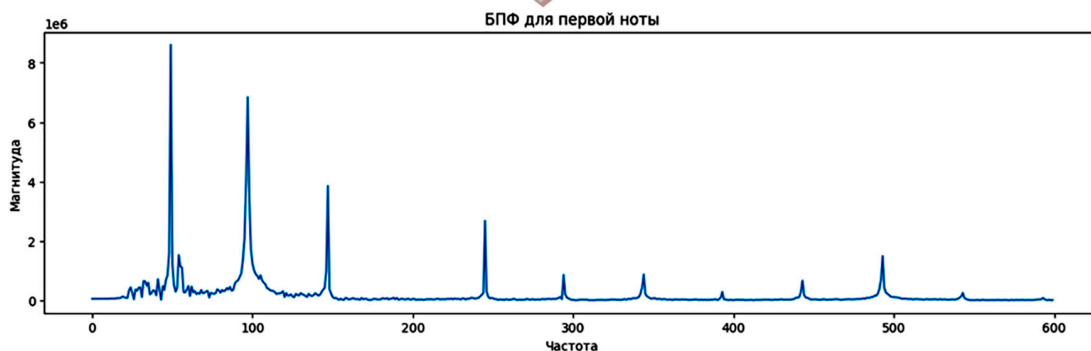


Рис. 9. Результат БПФ для первой ноты

```
# Классификация ноты и октавы на основе главной частоты
note = librosa.hz_to_note(freq, octave=False).lower()
octaveNumber = librosa.hz_to_octs(freq)
lastOctave = 2
octaveSign = ("","*abs(lastOctave-octaveNumber),
              ""*abs(lastOctave-octaveNumber) [lastOctave<octave]
#Темп песни, необходимый для расчета долей нот
ТЕМПО = 75
# Вычисление доли на основании темпа и длительности
duration = str(power_log(np.floor((60/ТЕМПО * 4)/((to - from)/1000)))
# Запись ноты, её октавы и доли в массив нот
predicted_notes.append(noteName+octaveSign+duration or "U")
```



Массив найденных нот и их долей:

```
['g8', 'a,8', 'b8', 'g8', 'c8', 'e8', 'd8', 'b8', 'g8', 'a,8', 'b8', 'd16', 'a8', 'c8',
 'b16', 'a8', 'g8', 'g8', 'g,16', 'a,8', 'b8', 'g8', 'c8', 'e8', 'd8', 'b8', 'g8', 'a,8',
 'b8', 'd8', 'c8', 'b8', 'a8', 'g8']
```

Рис. 10. Массив найденных нот, их октав и долей

На рисунке 10 четко представлен фрагмент кода, демонстрирующий применение описанных функций и формул для определения параметров первой ноты, а в качестве результата – приведен массив найденных нот, их октав и долей для всего аудиофайла.

Для преобразования созданного массива нот и их долей в партитуру применяется GNU LilyPond, который представляет собой свободно распространяемый нотный редактор со специальным языком разметки. Музыкальная партитура создается путём

компиляции из текстового файла, содержащего определенные элементы, размещенные в соответствующих блоках. Например, предусмотрен блок `\header`, в котором указываются название композиции и её автор, блок `\absolute` – для задания октав нот в абсолютных значениях, `\clef treble` – для обозначения скрипичного музыкального ключа, для задания нот применяется формат «нота,октава,длительность» (октава задётся серией запятых или апострофов, каждый из которых, соответственно, понижает или повышает ноту на одну октаву).

```
# Константы для хранения названия произведения, имени автора и временного размера
NAME = "Название"
COMPOSER = "Автор"
TIME = "4/4"

# Запись полученных нот в соответствии с форматом Lilypond
notes = "\\header { title = \"" + NAME + "\" composer = \"" + COMPOSER + "\" }
\\absolute { \\time " + TIME + " \\tempo 4=" + str(TEMPO) + " \\clef treble "
for note in predicted_notes:
    notes += note + " "
notes = notes+ " }"
```



```
\header { title = "Название" composer = " Автор" } \absolute { \time 4/4 \tempo
4=75 \clef treble g8 a,8 b8 g8 c8 e8 d8 b8 g8 a,8 b8 d16 a8 c8 b16 a8 g8 g8 g,16
a,8 b8 g8 c8 e8 d8 b8 g8 a,8 b8 d8 c8 b8 a8 g8 }
```

Рис. 11. Создание записи для компиляции в LilyPond

```
!sudo apt-get install -y lilypond
# Генерация файла с нотами и его преобразование в PDF-документ
f = open("/content/notes.ly", "w")
f.write(notes)
f.close()
os.system("sudo lilypond /content/notes.ly /content/notes.pdf")
os.remove("/content/notes.ly")
print("Файл notes.pdf успешно сохранен!")
```



Название

Автор

Рис. 12. Вызов утилиты lilypond и получение музыкальной партитуры в формате PDF

Программа создаёт сначала промежуточный файл (в формате TeX), который затем преобразуется в PDF-файл. На рисунке 11 показано формирование записи полученных нот согласно формату LilyPond для её последующей компиляции в партитуру.

На рисунке 12 показано, как производится запуск утилиты lilypond с правами суперпользователя «sudo», которая обрабатывает файл, содержащий сформированную запись, и выводит результат в файл с нотами формата PDF.

Заключение

Таким образом, в статье рассмотрен пример применения методов математического моделирования, музыкальных компьютерных технологий, современных языков программирования в блоке лабораторных занятий по дисциплине «Информационные технологии в профессиональной деятельности» для будущих специалистов в области музыкального образования. Представленная и подобные ей работы позволят

студентам повысить их уровень цифровых компетенций, а также применить освоенные методы и алгоритмы в будущей профессиональной деятельности.

Список литературы

1. Яцентковская Н.А. Информатизация музыкального образования: трактовка музыкального языка как информационной системы // Мир науки, культуры, образования. 2018. № 1(68). С. 257-259.
2. Сизова О.А., Карнаухова В.А. Ключевая роль профессиональной компетентности будущего педагога-музыканта, в условиях реализации профессиональных стандартов // Государственный советник. 2018. № 2(22). С. 54-58.
3. Приселков А.С., Терентьев Ю.Ю. Метод моделирования в музыкальном образовании // Проблемы современного педагогического образования. 2021. № 70-3. С. 124-128.
4. Солоденников Д.А. Определение жанра музыки по мелодии с помощью машинного обучения и параллельной обработки данных // Информационные системы и технологии в моделировании и управлении: сборник трудов VII Международной научно-практической конференции / Отв. редактор К.А. Маковойчук, Ялта, 24–25 мая 2023 года. Симферополь: Общество с ограниченной ответственностью «Издательство Типография «Ариал», 2023. С. 95-101.
5. Бурякова О.С., Решетникова И.В., Черкесова Л.В. Методы искусственного интеллекта в генерации алгоритмических музыкальных композиций // Современные наукоемкие технологии. 2022. № 8. С. 82-91. DOI: 10.17513/snt.39271.
6. Никитин Н.А., Розалиев В.Л., Орлова Ю.А., Заболева-Зотова А.В. Применение методов машинного обучения для задачи генерации музыкальных композиций // Прикаспийский журнал: управление и высокие технологии. 2018. № 2(42). С. 84-95.
7. Мелешкин А.А., Фирсова С.А. Разработка системы для автоматического распознавания аккордовой последовательности в цифровых аудиофайлах // Огарёв-Online. 2021. № 12(165).