



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
НА ТЕМУ:

«Метод автоматического определения ритмического рисунка и
темпа цифровой музыкальной записи с использованием
байесовского иерархического моделирования»

Студент ИУ7-86Б
(Группа)

(Подпись, дата)

А. А. Петрова
(И.О. Фамилия)

Руководитель

(Подпись, дата)

К. А. Кивва
(И.О. Фамилия)

2023 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Основная часть	5
1.1 Постановка задачи	5
1.2 Этапы работы метода	5
1.2.1 Определение темпа	5
1.2.2 Определение ритма	8
1.3 Исходные коды и интерфейс	11
ЗАКЛЮЧЕНИЕ	13
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	14

ВВЕДЕНИЕ

Во время выполнения выпускной квалификационной работы был разработан метод автоматического определения переменного ритмического рисунка и переменного темпа цифровой музыкальной записи на основе байесовского иерархического моделирования.

1 Основная часть

1.1 Постановка задачи

Необходимо разработать метод определения переменного ритма и переменного темпа музыки на основе байесовского иерархического моделирования. Для этого требуется составить модели, которые будут использовать статистические данные о музыке, такие как темп и тактовый размер. Модель для определения темпа должна также учитывать жанр анализируемой музыки. Составленные модели должны быть обучены на наборе данных, включающих в том числе различные жанры [1]. После обучения моделей, они должны быть протестированы на новых данных, чтобы оценить их точность и эффективность.

Входные данные:

- аудиофайл;
- жанр музыки (строка).

Выходные данные:

- набор темпов (целые числа, в BPM);
- набор тактовых размеров (обыкновенные дроби в формате «a/b»).

Оба набора выходных данных должны сопровождаться временем, соответствующим каждому темпу или размеру.

Ограничения:

- загружаемые аудиофайлы должны быть в формате mp3;
- знаменатель тактового размера принимается равным 4.

1.2 Этапы работы метода

1.2.1 Определение темпа

Ниже представлены IDEF0-диаграммы для алгоритма определения переменного темпа музыки [2, 3].

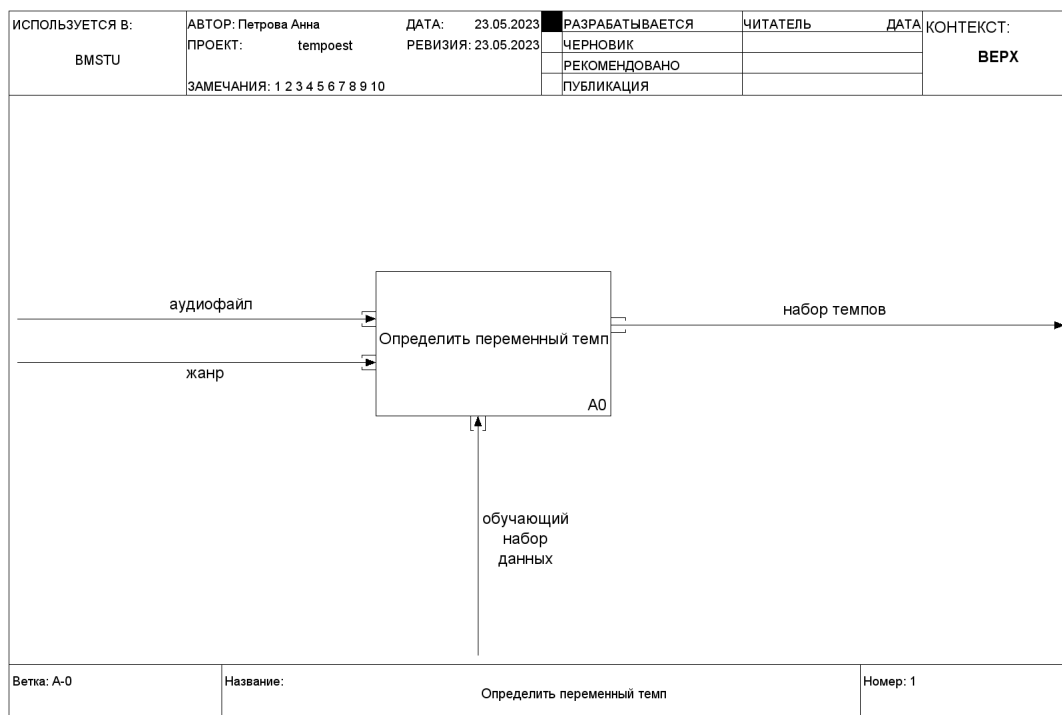


Рис. 1.1 – IDEF0 нулевого уровня

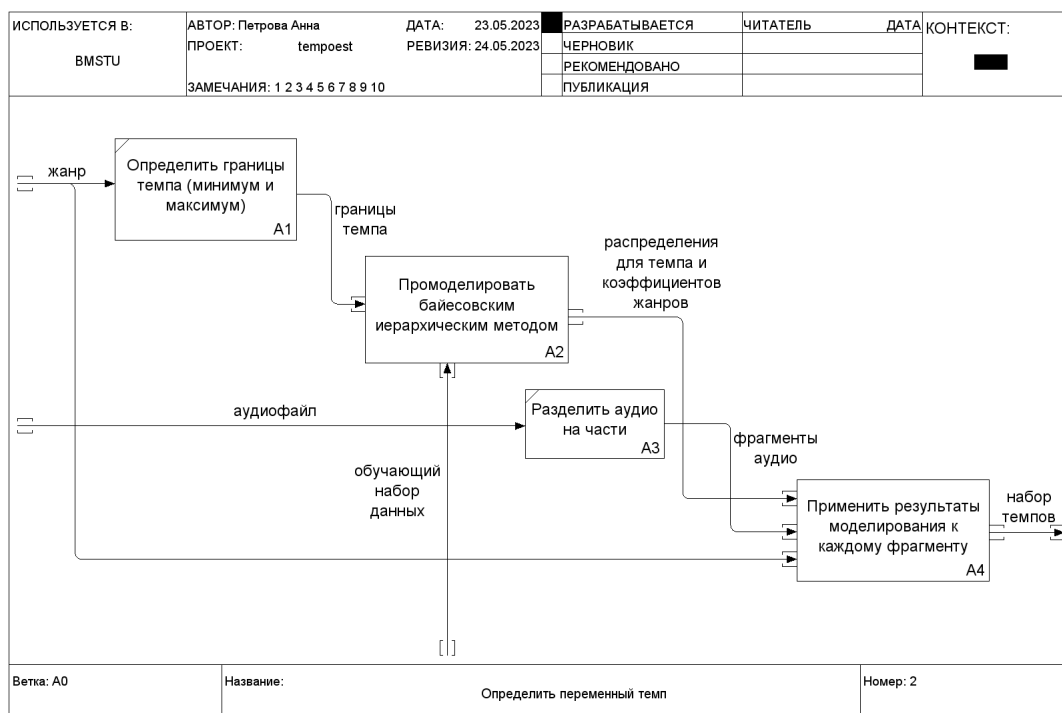


Рис. 1.2 – Определение переменного темпа

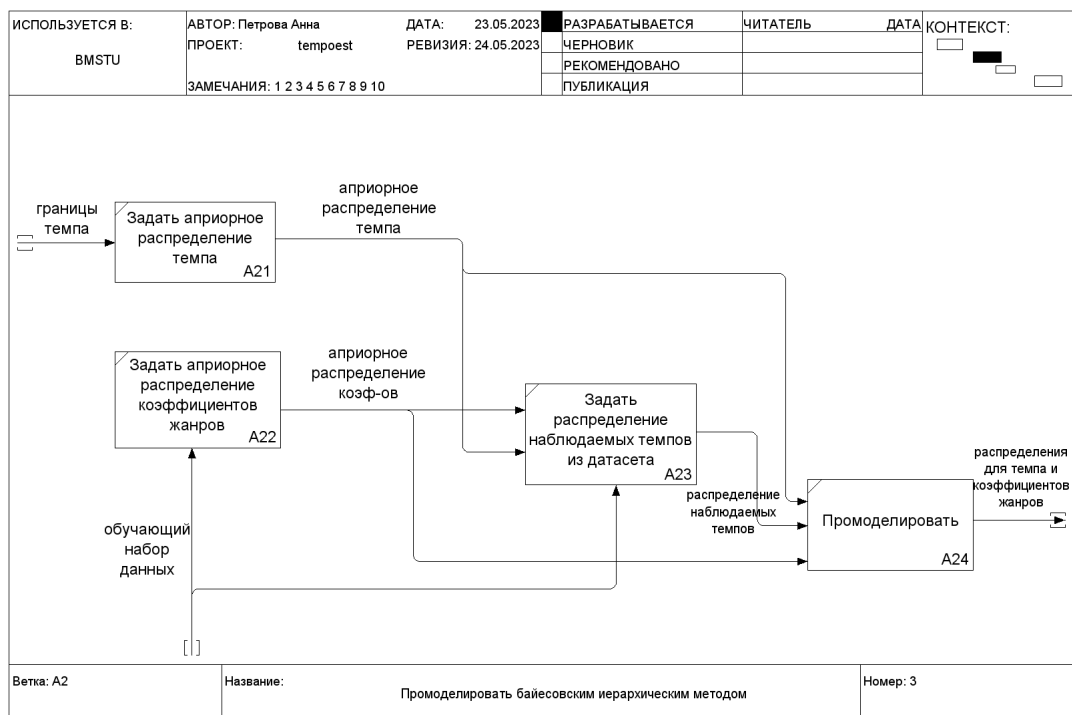


Рис. 1.3 – Байесовское моделирование

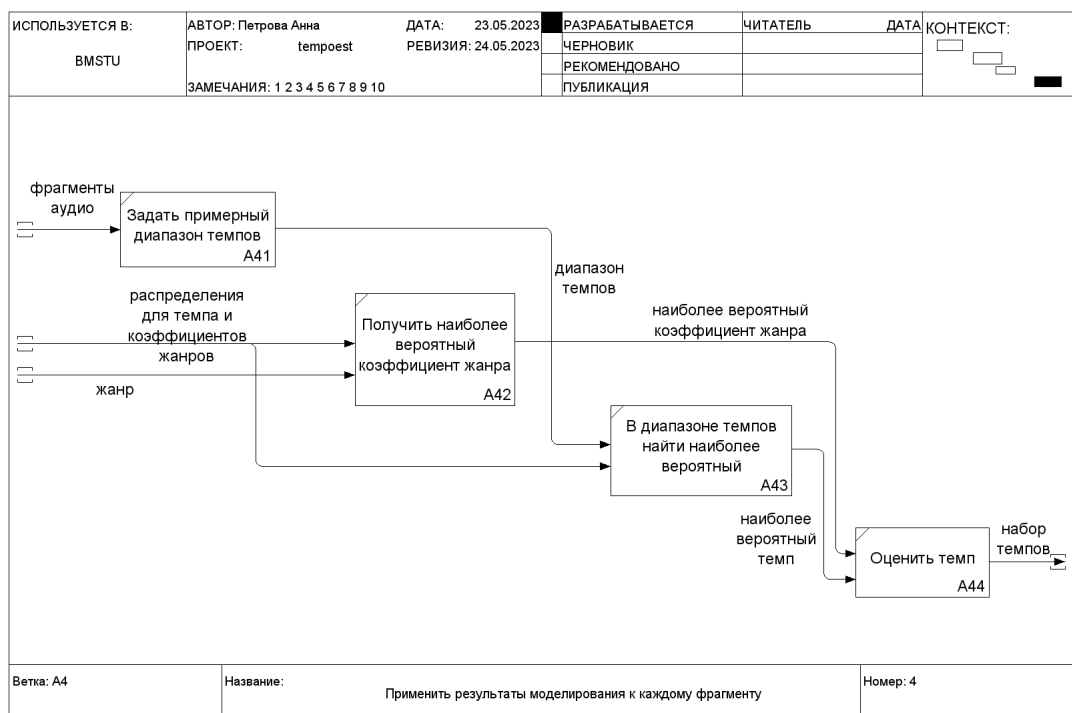


Рис. 1.4 – Применение результатов к фрагментам аудио

1.2.2 Определение ритма

Ниже приведены IDEF0-диаграммы для алгоритма определения переменного ритма (тактового размера) музыки.

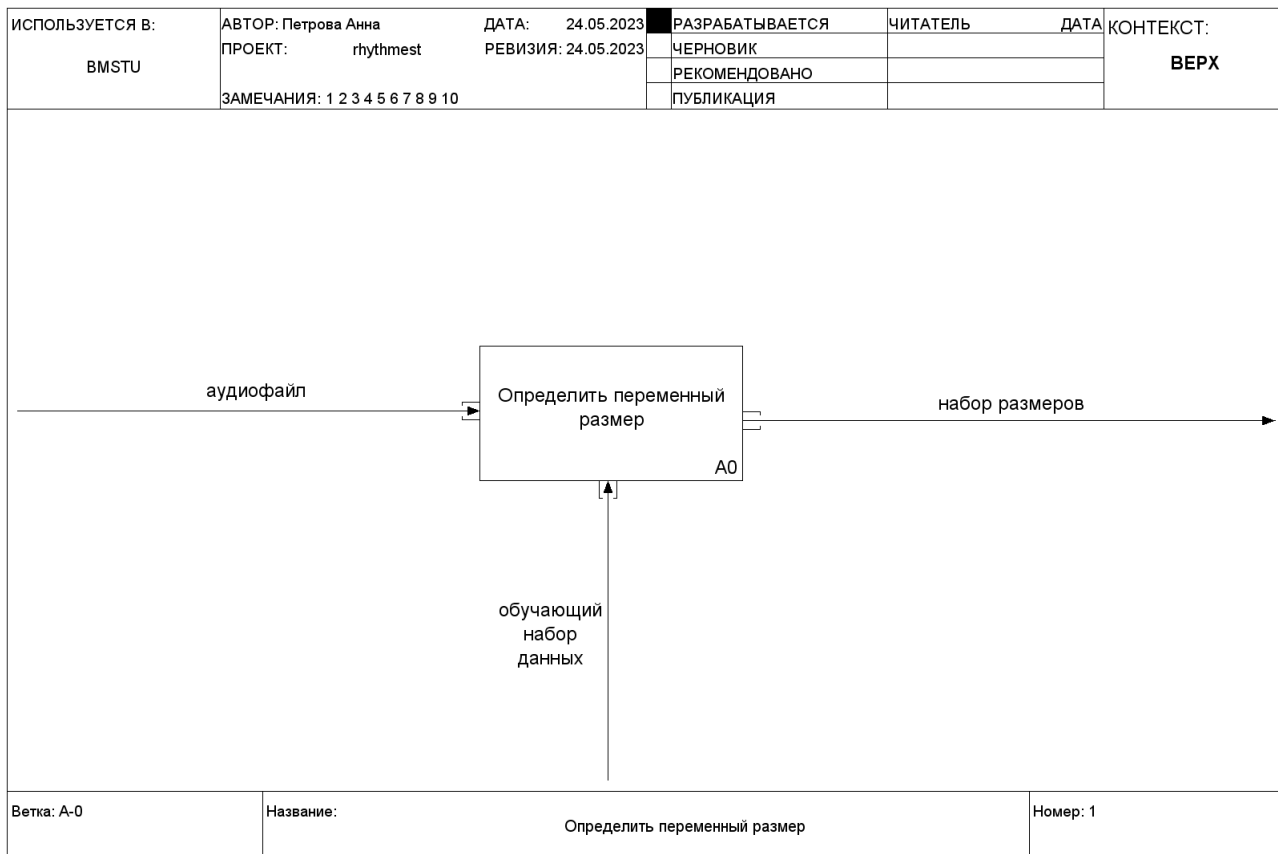


Рис. 1.5 – IDEF0 нулевого уровня

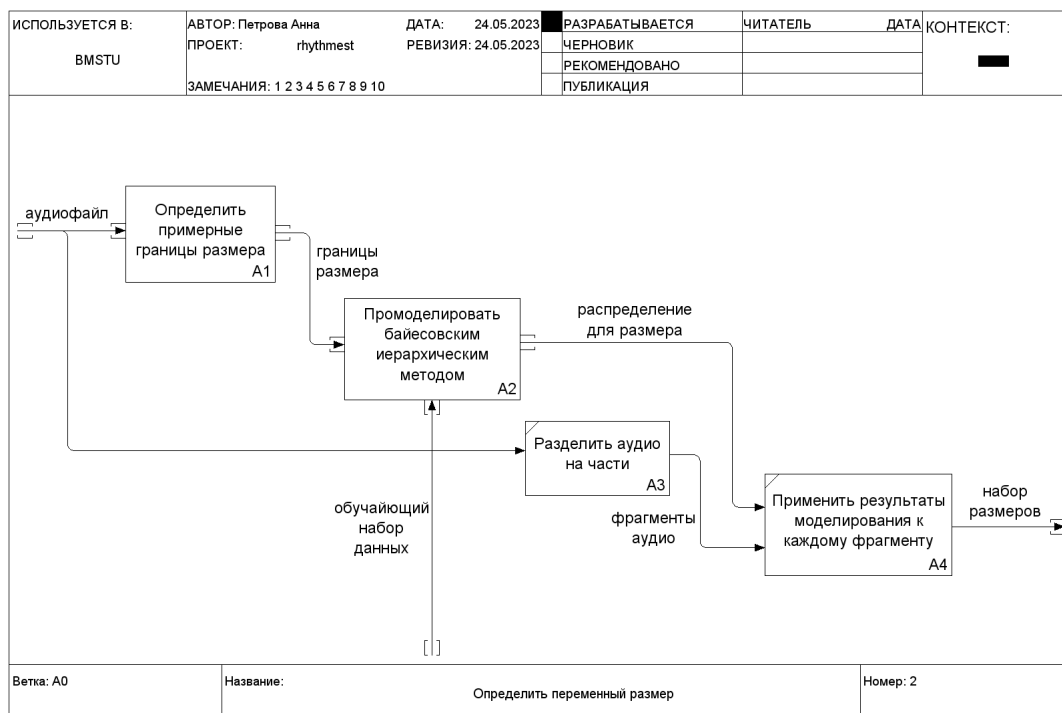


Рис. 1.6 – Определение переменного ритма

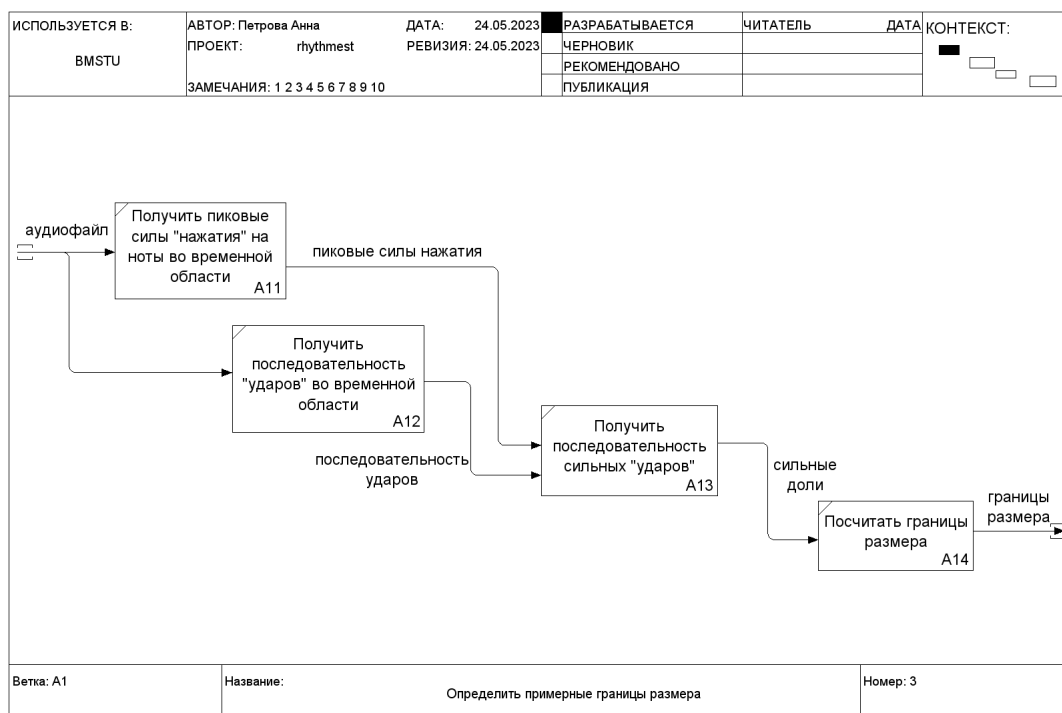


Рис. 1.7 – Определение границ размера

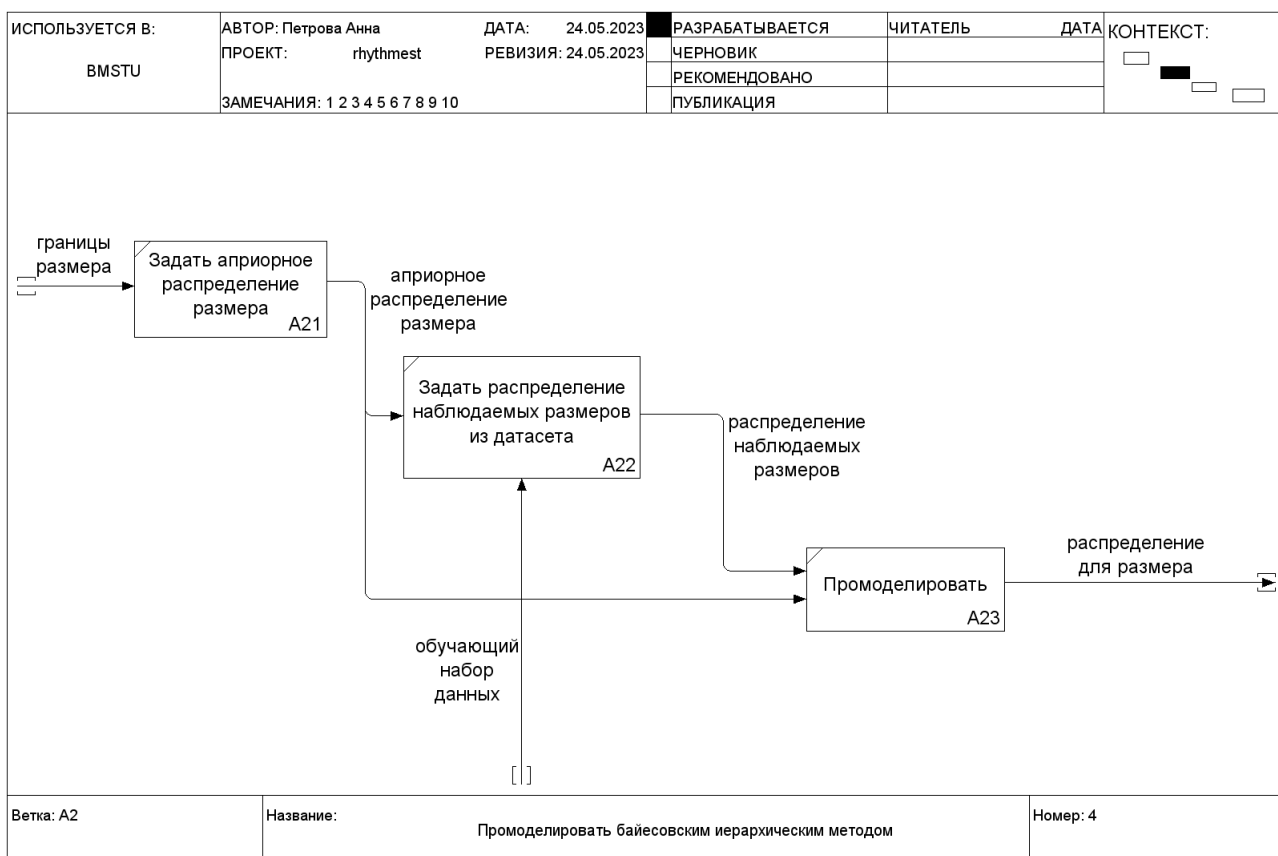


Рис. 1.8 – Байесовское моделирование

1.3 Исходные коды и интерфейс

В листингах ниже представлены реализации байесовских иерархических моделей для определения темпа и ритма (тактового размера) музыки. Для разработки моделей использовалась библиотека PyMC3 [4], а для работы с аудио – библиотека librosa [5].

Листинг 1: реализация байесовской модели для определения темпа

```
1  def bpm_model(min_bpm: int, max_bpm: int, bpm_dataset, genre_dataset,
    genres_ints, progress):
2      with pm.Model() as model:
3          # hyperpriors (lvl 1)
4          tempo = pm.Uniform('tempo', lower=min_bpm, upper=max_bpm)
5          progress.setValue(20)
6          mu = (min_bpm + max_bpm) / 2.0
7          sigma = (max_bpm - min_bpm) / 12.0
8          genre_coef = pm.Normal('genre_coef', mu=0, sd=1, shape=len(
    genre_dataset.unique()))
9          progress.setValue(40)
10
11         # prior (lvl 2)
12         bpm_est = mu + genre_coef[genres_ints] * sigma
13         progress.setValue(60)
14
15         # likelihood (lvl 3)
16         bpm_obs = pm.Normal('bpm_obs', mu=bpm_est, sd=sigma, observed=
    bpm_dataset)
17         progress.setValue(80)
18
19         # get the samples
20         trace = pm.sample(1000, tune=500, chains=2, cores=1)
21         progress.setValue(100)
22
23     return trace
```

Листинг 2: реализация байесовской модели для определения ритма

```
1  def rhythm_model(measure_min, measure_max, rhythm_dataset, progress):
2      with pm.Model() as model:
3          # prior
```

```

4     measure = pm.Uniform('measure', lower=measure_min, upper=measure_max)
5     progress.setValue(20)
6     mu = (measure_min + measure_max) / 2.0
7     progress.setValue(40)
8     sigma = (measure_max - measure_min) / 12.0
9     progress.setValue(60)
10
11     # likelihood
12     measure_obs = pm.Normal('measure_obs', mu=mu, sd=sigma, observed=
rhythm_dataset)
13     progress.setValue(80)
14
15     trace = pm.sample(1000, tune=1000, chains=2)
16     progress.setValue(100)
17
18     return trace

```

На рисунке 1.9 представлен интерфейс разработанного приложения [6].

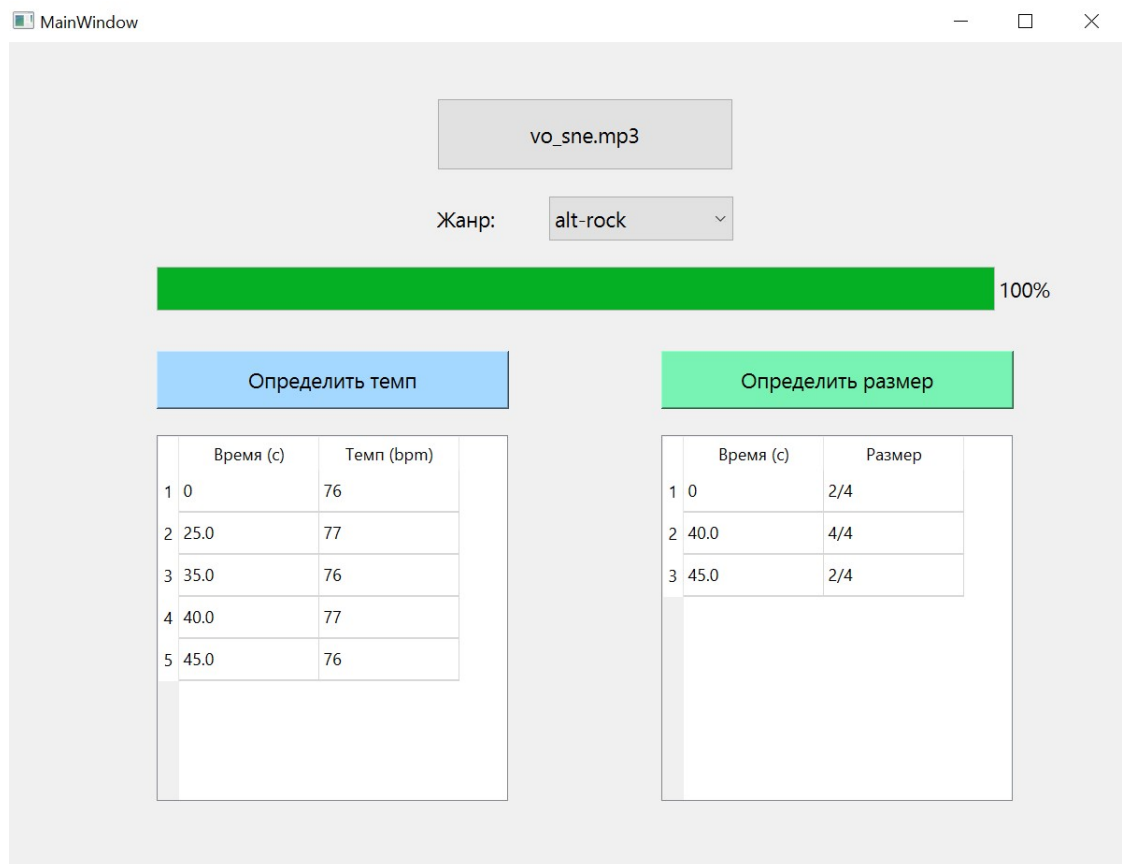


Рис. 1.9 – Интерфейс приложения

ЗАКЛЮЧЕНИЕ

Было разработано программное обеспечение, демонстрирующее практическую осуществимость спроектированного в ходе выполнения выпускной квалификационной работы метода автоматического определения переменного ритмического рисунка и переменного темпа цифровой музыкальной записи на основе байесовского иерархического моделирования.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Spotify Tracks Dataset [Электронный ресурс]. – Режим доступа: <https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset> (15.05.2023).
2. Kübler R. Bayesian Hierarchical Modeling in PyMC3. – 2021.
3. Nakamura E., Itoyama K., Yoshii K. Rhythm transcription of MIDI performances based on Hierarchical Bayesian Modelling of repetition and modification of musical note patterns. – 2016.
4. PyMC3 Documentation [Электронный ресурс]. – Режим доступа: <https://www.pymc.io/projects/docs/en/v3/> (дата обращения: 17.05.2023).
5. Librosa: audio and music processing in Python [Электронный ресурс]. – Режим доступа: <https://librosa.org/doc/latest/index.html> (дата обращения: 17.05.2023).
6. PyQt Documentation v6.5.0 [Электронный ресурс]. – Режим доступа: <https://www.riverbankcomputing.com/static/Docs/PyQt6/> (дата обращения: 20.05.2023).