

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)

Факультет информационных технологий
Кафедра «Инфокогнитивные технологии»

КУРСОВОЙ ПРОЕКТ

на тему: « *Инструмент исследования данных из файлов с аудио*»

Направление подготовки 09.03.03 «Прикладная информатика»
Профиль «Корпоративные информационные системы»

Выполнил:

студент группы 211-362

Силаев Андрей Николаевич

30.06.2023

_____ (подпись)

Москва 2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ И ИНСТРУМЕНТЫ.....	4

2	ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА.....	6
2.1	Структура базы данных.....	6
2.2	Структура приложения.....	6
2.3	Реализация приложения.....	7
3	ЗАКЛЮЧЕНИЕ.....	11

ВВЕДЕНИЕ

Целью данного курсового проекта является разработка инструмента для исследования данных, извлеченных из аудиофайлов различных форматов. Данное приложение обеспечивает поддержку трех основных форматов аудиофайлов: mp3, aiff и flac. Оно предоставляет удобный способ анализа и извлечения метаданных из этих файлов, а также обеспечивает графический интерфейс пользователя для их наглядного отображения.

Одним из ключевых аспектов разработки данного приложения является его способность эффективно обрабатывать различные форматы аудиофайлов. Форматы mp3, aiff и flac широко распространены и используются для хранения и передачи аудиоданных. Поддержка всех трех форматов позволяет приложению быть универсальным инструментом для работы с аудиофайлами.

Для хранения метаданных из аудиофайлов использовалась реляционная база данных. Она позволяет пользователям хранить и управлять множеством аудиофайлов, а также облегчает дальнейший анализ и исследование этих данных. База данных обеспечивает удобный доступ к метаданным и позволяет пользователю выполнять различные запросы и фильтрации для получения нужной информации.

Графический интерфейс пользователя является важным компонентом приложения, так как он обеспечивает интуитивно понятный и удобный способ взаимодействия с функциональностью приложения. Пользователи смогут легко просматривать и анализировать метаданные, выполнять поиск по определенным критериям, фильтровать данные и просматривать результаты в удобном формате.

1 ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ И ИНСТРУМЕНТЫ

В рамках разработки инструмента для анализа данных из аудиофайлов, был проведен тщательный анализ доступных библиотек для работы с метаданными аудиофайлов. Были рассмотрены такие библиотеки, как `tinytag`, `eyeD3`, `ID3`, `songdetails`, `mp3-tagger` и `mutagen`. После проведения анализа и сравнения этих библиотек, было принято решение использовать `mutagen` по следующим причинам:

- обширная поддержка форматов. `Mutagen` обеспечивает поддержку широкого спектра форматов аудиофайлов, включая `mp3`, `aiff` и `flac`, которые требуются для нашего приложения. Это позволяет нам удобно работать с файлами различных форматов, не используя отдельные библиотеки для каждого формата и стандартизировать получение метаданных из файлов;
- простота использования. `Mutagen` предоставляет простой и понятный функционал для работы с метаданными аудиофайлов. Это упрощает процесс извлечения и обработки метаданных в приложении.

В процессе выбора инструмента для отрисовки спектрограммы в проекте, были рассмотрены несколько библиотек, включая `librosa` и `SciPy`. Однако, после проведения детального анализа и сравнения их функциональности, было принято решение остановиться на использовании библиотеки `librosa`. Одной из основных причин выбора `librosa` было ее специализация на анализе и обработке аудиофайлов. Библиотека `librosa` предоставляет широкий спектр функций для работы с аудио, включая возможность создания спектрограммы. Так же на выбор повлияла простота отрисовки спектрограммы при помощи одной функции без дополнительных вычислений и библиотек и поддержка всех используемых в проекте форматов аудиофайлов.

Для реляционной базы данных в проекте была выбрана СУБД `SQLite`. Этот выбор обусловлен несколькими факторами, включая простоту локального хранения данных и удобный синтаксис `SQL`. `SQLite` отличается возможностью

хранить базу данных в виде одного файла, что облегчает ее передачу и использование на разных системах. Это важно для нашего приложения, где данные будут храниться локально. Простота хранения данных в SQLite облегчает развертывание и управление базой данных в рамках проекта.

Для реализации интерфейса в проекте была выбрана библиотека tkinter. Этот выбор обусловлен простым синтаксисом и удобством использования данной библиотеки. Tkinter является стандартной библиотекой Python для создания графического интерфейса пользователя (GUI). Она предоставляет разнообразные инструменты и виджеты, позволяющие разрабатывать интерактивные пользовательские интерфейсы. Также важно, что tkinter является платформонезависимой библиотекой, что обеспечивает переносимость нашего приложения между различными операционными системами. Это позволяет разрабатывать универсальное приложение, доступное для использования на различных платформах, без необходимости вносить значительные изменения в код.

2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА

2.1 Структура базы данных

В базе данных используется таблица "audiofile", которая имеет следующую структуру:

- id – уникальный идентификатор записи (целое число);
- name – уникальное имя аудиофайла (текст);
- cover – обложка песни (BLOB);
- title – название песни (текст);
- album – название альбома песни (текст);
- artist – имя артиста (текст);
- genre – жанр песни (текст);
- year – год выпуска песни (текст);
- audio_format – расширение аудиофайла (текст);
- track_duration - длительность песни (текст);
- count_channels – количество каналов (текст);
- bitrate – битрейт песни (целое число);
- sampling_rate – частота дискретизации (целое число);
- file_size – размер файла (текст);
- spectrogram – спектрограмма песни (BLOB).

2.2 Структура приложения

Приложение состоит из трех файлов:

- metaWave.py – это основной модуль приложения, который используется для получения метаданных из аудиофайлов;
- dirAnalysis.py – этот модуль используется для инициализации базы данных, анализа директории на наличие аудиофайлов подходящего формата и получения данных из них при помощи модулей из metaWave, добавления и

обновления всех данных в базе данных. Так же при помощи него можно выгрузить базу данных в csv файл;

- `metaWaveGUI.py` – этот модуль используется для создания графического приложения на `tkinter`. Он использует функции `dirAnalysis` для получения данных из бд и манипуляций с ними, а так же для анализа директорий и выгрузки базы данных в csv файл.

2.3 Реализация приложения

Файл `metaWave.py` состоит из класса `MetaWave`, который предназначен для извлечения и управления метаданными аудиофайлов. Для достижения этой цели, модуль использует несколько внешних библиотек, таких как `mutagen`, `matplotlib` и `librosa`.

Класс содержит методы для извлечения метаданных из аудиофайлов и обработки звуковых данных. Внутренние методы класса, такие как `__get_metadata_mp3_aiff()`, `__get_metadata_flac()` и `__get_additional_data()`, используют библиотеку `mutagen` для получения метаданных из аудиофайлов, таких как `mp3`, `aiff` и `flac`.

Для создания спектрограммы звукового файла, модуль использует внутренний метод `__get_spectrogram()`, который использует библиотеку `librosa` для построения спектрограммы и `matplotlib` для задания размера диаграммы и сохранения спектрограммы в файл `jpg`. Так же используется библиотека `io` для преобразования картинки в байты.

Так же есть вспомогательные функции, такие как `__file_size()` и `print_metadata()`, которые обеспечивают преобразование размера файла из байтов в удобочитаемый формат и вывод извлеченных метаданных в консоль.

Файл `dirAnalysis.py` состоит из класса `DirAnalysis`, который представляет инструмент для анализа папки с аудиофайлами, извлечения и сохранения их

метаданных в базе данных SQLite. В данном классе также реализованы функции для поиска, фильтрации и экспорта данных.

В этом классе используется модуль `metaWave`, создается экземпляр класса `MetaWave` с именем `meta`. Этот экземпляр будет использоваться для получения метаданных из аудиофайлов.

Класс `DirAnalysis` содержит несколько приватных переменных, таких как `__all_metadatas`, `__search_flag` и `__search_query`, которые используются внутри класса для хранения информации о метаданных, состоянии поиска и запросе поиска.

Метод `set_data()` является основной функцией класса, которая принимает путь к папке с аудиофайлами, путь к базе данных SQLite и опциональные параметры типов файлов и исключений. Этот метод вызывает другие внутренние методы для создания базы данных, анализа папки и добавления данных в базу данных.

Метод `creat_db()` используется для создания базы данных, если она еще не существует. Здесь используется модуль `sqlite3` для создания и управления базой данных.

Метод `__dir_analysis()` анализирует содержимое указанной папки и извлекает метаданные из аудиофайлов указанных типов. Если в параметрах передан список исключений, то файлы из этого списка будут пропущены. Полученные метаданные добавляются в переменную `__all_metadatas`.

Метод `__creat_or_updat_data_db()` добавляет или обновляет данные в базе данных. Он выполняет проверку наличия записи с таким же именем в базе данных и в зависимости от результата либо вставляет новую запись, либо обновляет существующую.

Методы `output_all_data_db()` и `output_name_data_db()` используются для вывода данных из базы данных. Первый метод возвращает все данные из базы данных, а второй выводит только имена файлов в виде таблицы с помощью библиотеки `prettytable`.

Методы `search()` и `cancel_search()` используются для поиска данных в базе данных. Метод `search()` принимает путь к базе данных, тип поиска и строку поиска, а затем выполняет запрос к базе данных и возвращает найденные данные. Метод `cancel_search()` сбрасывает состояние поиска.

Метод `filter()` используется для фильтрации данных в базе данных. Он принимает путь к базе данных, тип фильтрации и флаг, указывающий направление сортировки. Метод выполняет запрос к базе данных с заданными параметрами и возвращает отфильтрованные данные.

Метод `export_csv()` экспортирует данные из базы данных в формат CSV. Он извлекает данные из базы данных и записывает их в файл CSV с использованием модуля `csv`.

Файл `metaWaveGUI.py` состоит из класс `MetaWaveGUI`, который представляет графический интерфейс пользователя для инструмента исследования данных из файлов с аудио. Он использует библиотеку `tkinter` для создания окон и элементов интерфейса.

При запуске приложения происходит проверка создана база данных или нет при помощи функции `creat_db()`, а потом достается вся информация из базы данных при помощи функции `output_all_data_db()` из класса `DirAnalysis` и передается в функцию `load_songs()`. Эта функция загружает список песен в интерфейс приложения. Она очищает существующие виджеты в контейнере `self.songs_frame_inner`, настраивает сетку для размещения песен и преобразовывает байты обложки песни в картинки при помощи класса `io`. Функция также добавляет обработчик щелчка мыши для отображения деталей песни при клике на обложку.

Окно подробной информации о песне создается при помощи функции `show_song_details()`, которая создает новое окно `details_window` для отображения подробной информации о песне. Она устанавливает размер и позицию окна на экране, а затем размещает метки и данные о песне, включая название и обложку (если доступна), в окне.

Вверху главного окна располагается кнопка «Открыть папку» – эта кнопка отвечает за открытие проводника, для выбора анализируемой папки, после выбора папки откроется модальное окно созданное при помощи функции `open_folder()`, где нужно указать типы файлов и исключения. Эти данные передадутся в функцию `update_db_and_songs()`. Эта функция отвечает за обработку полученных данных, преобразования их в удобный формат для функции `set_data()` из класса `DirAnalysis` и передачи их ей. Затем обновленные данные будут получены при помощи функции `output_all_data_db()` и переданы в функцию `load_songs()`.

Внизу главного окна расположена кнопка «Экспорт в csv» – эта кнопка отвечает за открытие проводника, для выбора места, где создать файл csv. После выбора данные передадутся в функцию `export_csv()` из класса `DirAnalysis`.

Слева в главном окне расположены поиск и фильтр. Данные из них будут переданы в функции `search()` и `filter()` из класса `DirAnalysis`.

3 ЗАКЛЮЧЕНИЕ

В рамках курсового проекта было разработано приложение для исследования данных из аудиофайлов. Приложение поддерживает три формата аудиофайлов: mp3, aiff и flac. Основная цель приложения заключается в извлечении и хранении метаданных аудиофайлов в базе данных, а также в предоставлении графического пользовательского интерфейса для их отображения.

В будущем планируется реализовать поддержку других форматов аудиофайлов, добавить дополнительные функции анализа и обработки звуковых данных, а также улучшить пользовательский интерфейс с учетом отзывов и запросов пользователей.

Ссылка на удаленный репозиторий:

<https://github.com/Andrey222333/MetaWave.git>