**Проект: МП3 плеер**

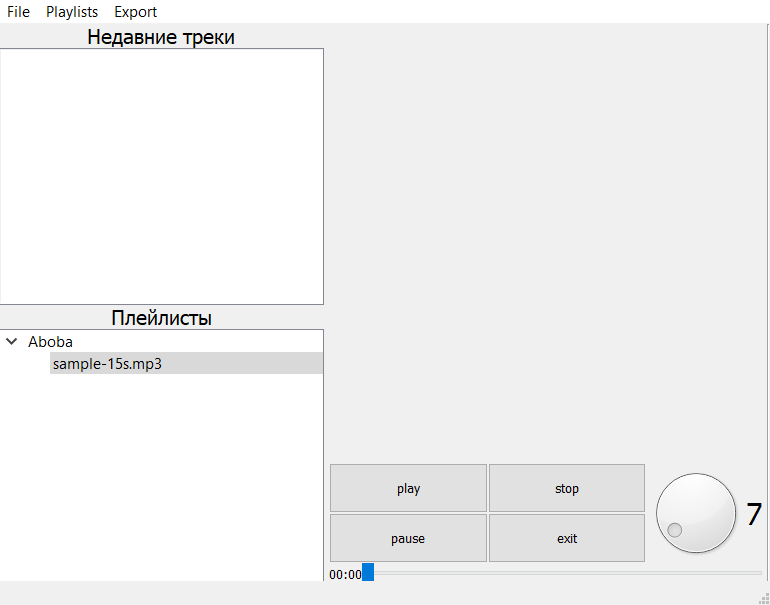
**ФИО автора/авторов:**

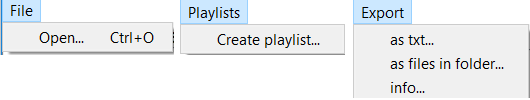
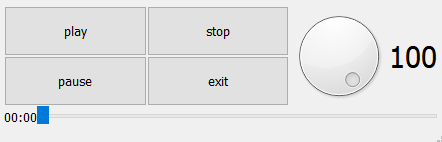
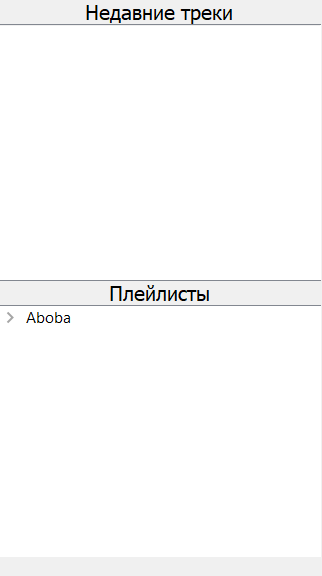
Хайруллин Азат

**Введение:**

Идея проекта состоит в создании МП3 плеера, который позволяет воспроизводить аудиофайлы различных форматов и управлять ими. Плеер разработан для решения задач прослушивания музыки, создания плейлистов, а также экспорта информации о композициях.

**А как пользоваться то?**

Запускаем exe-файл программы и видим перед собой подобное меню: [](https://private-user-images.githubusercontent.com/113846311/289568099-a14547d8-58ab-4834-b942-6f2a9db775a6.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmF3LmdpdGh1YnVzZXJjb250ZW50LmNvbSIsImtleSI6ImtleTEiLCJleHAiOjE3MDIzMTAyOTUsIm5iZiI6MTcwMjMwOTk5NSwicGF0aCI6Ii8xMTM4NDYzMTEvMjg5NTY4MDk5LWExNDU0N2Q4LTU4YWItNDgzNC1iOTQyLTZmMmE5ZGI3NzVhNi5wbmc_WC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQy1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBSVdOSllBWDRDU1ZFSDUzQSUyRjIwMjMxMjExJTJGdXMtZWFzdC0xJTJGczMlMkZhd3M0X3JlcXVlc3QmWC1BbXotRGF0ZT0yMDIzMTIxMVQxNTUzMTVaJlgtQW16LUV4cGlyZXM9MzAwJlgtQW16LVNpZ25hdHVyZT1iZmE5OTk0ZDFkYzg3MmMwYjFiNTMwOGIyZjdjNzY2NWEwMTIyZTgzMTU4YTk3M2Q3NGEyNjBkYTlhNzMwMWJmJlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCZhY3Rvcl9pZD0wJmtleV9pZD0wJnJlcG9faWQ9MCJ9.HB-5DGInV1HLYwSYQhnsMxfiXAGPTaMkAC2Lvzhw070)  
Рассмотрим повнимательнее:

* Тут мы видим toolbar-меню с различными заголовками:File, Playlists, Export. [](https://github.com/Azatish/Finally/blob/0c452fe8f7f6eff5c19c8d7fdda52570f1d340f9/image_explanations/toolbar.png)
  + File. Нажав, видим кнопку Open\_file, которая добавляет трек в db с playlist\_id = 1 и в QWidgetListWidget для удобного переключения между вашими недавно-добавленными треками
  + Playlist. Нажав, видим кнопку Create\_playlist, которая создает плейлист в db в таблицу playlists; создаёт плейлист с указанным пользователем именем. Элемент добавляется в QTreeWidget, где находятся все плейлисты, которые создал пользователь
  + Export.
    - as txt. Позволяет переместить ваши треки из выбранного плейлиста в выпадающем QWidget в txt файл
    - as file. Позволяет переместить ваши треки из выбранного плейлиста в выпадающем QWidget в отдельную папку, которую легко можно перенести на другие ваши устройства:)
    - info. Подробная информация о функции export.
* А тут... Тут же меню управлением воспроизведением вашего трека!  
  [](https://github.com/Azatish/Finally/blob/0c452fe8f7f6eff5c19c8d7fdda52570f1d340f9/image_explanations/play_pause_etc.png)  
  + play. Воспроизводить трек.
  + stop. Перейти в начало трека.
  + pause. Остановить воспроизведение трека.
  + exit. Закрыть трек.
  + Переключатель громкости воспроизводимой музыки(dial) с показом текущего значения громкости(label)
* Ну и наконец: окно плейлистов.  
  [](https://github.com/Azatish/Finally/blob/0c452fe8f7f6eff5c19c8d7fdda52570f1d340f9/image_explanations/playlists.png)  
  + В "Недавние треки" попадают все треки, которые вы импортировали в программу нажав на File->Open... Попробуйте нажать на добавленный трек и воспроизведите его. Наслаждайтесь музыкой
  + В "Плейлисты" попадают все плейлисты, которые вы импортировали в программу нажав на Playlists->Create Playlist... "А как ведь добавить трек в плейлист" - задаетёсь вы вопросом. Все просто. Достаточно нажать правой кнопкой мыши на плейлист и нажать на "Добавить трек".

**Описание реализации:**

Приложение реализовано на языке программирования Python с использованием библиотек PyQt5 и mutagen. Основная структура приложения состоит из класса MainWindow, который наследуется от класса QMainWindow. В приложении используются различные виджеты, такие как QTreeWidget, QMediaPlayer, QSlider и другие. Технологии, используемые в приложении:

* PyQt5 - библиотека для создания графических интерфейсов;
* sqlite3 - модуль для работы с базами данных SQLite;
* mutagen - библиотека для работы с метаданными аудиофайлов.

Код приложения охватывает функции управления воспроизведением, загрузки и экспорта аудиофайлов, создания и управления плейлистами. Приложение также предоставляет графический интерфейс, включающий различные элементы управления для удобства пользователя. Описание функций по коду:

* except\_hook(cls, exception, traceback) Функция, которая перехватывает исключения и отображает их в консоли.
* MainWindow(QMainWindow) Класс главного окна приложения. Инициализирует графический интерфейс, настраивает виджеты и подключает обработчики событий.
* init\_database(self) Заполняет QTreeWidget актуальными плейлистами из базы данных.
* init\_UI(self) Инициализирует пользовательский интерфейс, создает меню, подключает обработчики кнопок и действий.
* Check\_File(self) Проверка на присутствие файла по его директории в db
* Open\_File(self) Обработчик события открытия файла. Открывает диалоговое окно для выбора файла и добавляет его в список предыдущих треков.
* load\_mp3(self) Загрузка музыкального файла в плеер, обработка изображения(при наличии)
* set\_volume(self) Устанавливает громкость музыки в плеере в соответствии со значением настроенного диала.
* play\_music(self) Запускает воспроизведение музыки.
* pause\_music(self) Приостанавливает воспроизведение музыки.
* stop\_music(self) Останавливает воспроизведение музыки и обнуляет текущую позицию трека.
* check\_isMedia\_now(self) Проверяет, занят ли плеер в данный момент и выполняет соответствующие действия в зависимости от состояния плеера.
* enable\_message\_box\_isMedia\_button(self) Включает кнопку в сообщении MessageBox.
* export\_tracks\_as\_txt(self) Выполняет экспорт треков плейлиста в текстовый файл.
* export\_tracks\_as\_files(self) Выполняет экспорт треков плейлиста в папку.
* infoExport(self) Отображает информацию о функции экспорта треков в MessageBox.
* on\_item\_clicked(self, item) Обработчик события клика на QListWidget previous\_tracks. Загружает выбранный трек в плеер.
* create\_new\_playlist(self) Создает новый плейлист с введенным пользователем названием.
* track\_clicked(self, item, column) Обработчик события клика на трек в плейлисте. Загружает выбранный трек в плеер.
* show\_context\_menu(self, position) Отображает контекстное меню для добавления треков в плейлист.
* do\_action(self) Выполняет действие при выборе пункта контекстного меню для добавления трека в плейлист.
* exit\_music(self) Очищает текущий трек из плеера и обновляет информацию о времени воспроизведения.
* duration\_ch(self) Устанавливает максимальное значение слайдера времени в соответствии с длительностью трека.
* position\_ch(self) Устанавливает значение слайдера времени в соответствии с текущей позицией, также обновляет отображение текущего времени (NowTime\_label).
* slider\_triggered(self) Устанавливает новую позицию проигрывания трека в соответствии с новым положением слайдера.
* rename\_playlist(self) Переименование плейлиста

Это основные функции в коде. Они реализуют основной функционал приложения для работы с плейлистами и воспроизведением музыки

**Доработки:**

* Улучшение дизайна и пользовательского интерфейса для более удобного пользовательского опыта.
* Реализация функционала поиска и фильтрации музыкальных композиций в базе данных.
* Расширение возможностей управления плейлистами, такие как переименование, удаление и перемещение треков между плейлистами.

