# Руководство разработчика к приложению

## **«Приложение для управления базой данных»**

**Разработчик**:

Рустамова Дарина Дмитриевна

Оглавление

[Руководство разработчика к приложению 1](#_Toc75461878)

[**«Приложение для управления базой данных»** 1](#_Toc75461879)

[Описание структуры базы данных 2](#_Toc75461880)

[Структура директории 2](#_Toc75461881)

[Технические требования 3](#_Toc75461882)

[Описание класса DataBase 4](#_Toc75461883)

[Описание классов и функций в файле main.py 4](#_Toc75461884)

[Описание классов и функций в файле widgets.py 4](#_Toc75461885)

# 

# Описание структуры базы данных

База данных «tracks» состоит из 157 записей и 12 атрибутов: 5 качественных и 7 количественных:

Качественные: Название, Название альбома, Название жанра, Имя артиста, Место создания

Количественные: Трек ID, Альбом ID, Жанр ID, Прослушиваний, Рейтинг, Артист ID, Стоимость.

База данных поделена на 4 справочника: Треки, Альбомы, Артисты и жанры:

**«Треки»** включает в себя атрибуты Трек ID, Альбом ID, Название и прослушиваний.

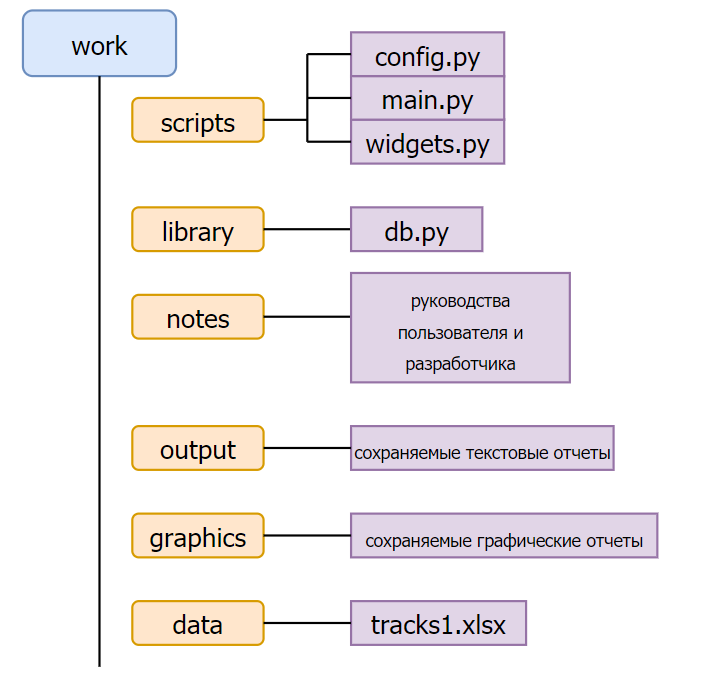
**«Альбомы»** включает в себя атрибуты Альбом ID, Название альбома, Жанр ID, Стоимость, Артист ID и Место создания

**«Артисты»** включает в себя атрибуты Артист ID, Имя артиста, Рейтинг

**«Жанры»** включает в себя атрибуты Жанр ID и Название жанра.

# Структура директории

Директория work выглядит следующим образом:



В папке

**scripts** содержатся файлы с кодом:

**main.py** – главный запускаемый файл

**widgets.py** – виджеты приложения, графический интерфейс

**config.py** – конфиг приложения, содержит путь

**library** – содержит файл **db.py**, который парсит базу данных по справочникам и содержит необходимые функции для редактирования

**notes** – руководства пользователей и разработчика

**output** – сохраняются текстовые отчеты

**graphics** – сохраняются графические отчеты

**data** – содержится файл формата xlsx базы данных

# Технические требования

ПК с операционной системой **Windows XP** или выше, или **MacOS Leopard** или выше, или **Linux**. Также требуется минимум 1.5 ГГц тактовой частоты процессора и разрешение экрана 1280x720 или выше.

# Описание класса DataBase

|  |  |
| --- | --- |
| \_\_init\_\_ | Инициализирует базу данных |
| Add() | Добавляет запись в базу данных |
| Delete(index\_df, row) | Удаляет запись с индексом row в справочнике index\_df |
| Define\_df | Делит базу данных на справочники |
| Transform\_type() | Изменяет типы атрибутов базы данных в соответствии с изначальными типам |
| Save\_default(format) | Возвращает файл в папке output c форматом format |
| Save\_as\_csv() | Сохраняет в формате csv |
| Save\_as\_xlsx() | Сохраняет в формате xlsx |
| Save\_as\_pic() | Сохраняет в формате pic |
| Get\_frame() | Возвращает текущую frame |
| @classmethod get\_full\_df | Возвращает базу данных |
| @classmethod get\_list\_names\_df | Возвращает лист справочников и их названий |
| @classmethod get\_df\_types | Возвращает лист типов |

# Описание классов и функций в файле main.py

|  |  |
| --- | --- |
| main() | Создает главное окно root, настраивает его и вызывает функцию build\_app() |
| Build app() | Строит графические виджеты, из них далее вкладки |

# Описание классов и функций в файле widgets.py

|  |  |  |
| --- | --- | --- |
| Table(ttk.Frame) | \_\_init\_\_(parent, df) | Инициализирует таблицу данных dataframe, где df – dataframe, а parent – это master для текущей таблицы |
|  | get\_df() | Возвращает измененную dataframe, на основе вводимых в entry новых значений |
| TableManaget(DataBase) | \_\_init\_\_(root) | Инициализирует менеджер для редактирования таблицы, родительский класс DataBase, root – master для текущего frame |
|  | Init\_table() | обновляет таблицу данных |
|  | Check\_types(df) | Проверяет типы данных измененных записей, для соответствия типам данных dataframe  Возвращает true, если соответствует, иначе false |
|  | Add\_record() | Добавляет в dataframe новую запись |
|  | Delete\_record() | Удаляет запись |
|  | Init\_menu() | Обновляет\создает frame с кнопками\надписями |
|  | Get\_frame() | Возвращает текущую frame |
| SamleData(DataBase) | \_\_init\_\_(parent) | Инициализирует frame с master – parent, а также dataframe  Вкладка с выборкой данных |
|  | Init\_table(df) | Обновляет текущую таблицу для dataframe df |
|  | Change\_main\_base() | Меняет текущий справочник |
|  | Init\_manager() | Инициализирует\обновляет меню для выборки данных с кнопками\надписями |
|  | Save\_changes() | Сохраняет выбранные атрибуты и вызывает init\_table для визуализации |
|  | Save\_default(format) | Возвращает файл в папке output c форматом format |
|  | Save\_as\_csv() | Сохраняет в формате csv |
|  | Save\_as\_xlsx() | Сохраняет в формате xlsx |
|  | Save\_as\_pic() | Сохраняет в формате pic |
|  | Get\_frame() | Возвращает текущую frame |
| StaticData(ttk.Frame) | \_\_init\_\_(parent, file\_path) | Инициализирует класс и два класса QualityData и QuantityData, которые отвечают за меню для получения статистистика для качественных и количественных атрибутов  Parent – master для создания этих frame  File\_path – путь сохранения отчета |
| QualityData(DataBase) | \_\_init\_\_(parent, file\_path) | Инициализирует frame для создания статистики для качественных атрибутов |
|  | Change\_main\_base() | Изменяет текущий справочник |
|  | Init\_table(df) | Создает таблицу для dataframe df |
|  | Build\_statistic() | Строит статистику для текущей базы данных |
|  | Define\_quality\_attr()  Для QualityData() | Определяет качественные атрибуты |
|  | Init\_manager() | Строит менеджер для управления статистикой |
|  | Define\_quantity\_attr()  Для QuantityData() | Определяет количественные атрибуты |
|  | Save\_default(format) | Возвращает файл в папке output c форматом format |
|  | Save\_as\_csv() | Сохраняет в формате csv |
|  | Save\_as\_xlsx() | Сохраняет в формате xlsx |
|  | Save\_as\_pic() | Сохраняет в формате pic |
|  | Get\_frame() | Возвращает текущую frame |
| Pivot\_table(DataBase) | \_\_init\_\_(parent, file\_path) | Инициализирует класс и frame для сводной таблицы, file\_path для сохранения по умолчанию |
|  | Change\_main\_base() | Изменяет текущий справочник |
|  | Clear\_combobox() | Очищает entry |
|  | Define\_quality\_attr() | Определяет качественные атрибуты |
|  | Init\_table(df) | Создает таблицу данных df – dataframe |
|  | Change\_table() | Меняет текущую dataframe исходя из заданных параметров |
|  | Init\_manager() | Создает\обновляет меню управления и редактирования базой данных |
|  | Save\_default(format) | Возвращает файл в папке output c форматом format |
|  | Save\_as\_csv() | Сохраняет в формате csv |
|  | Save\_as\_xlsx() | Сохраняет в формате xlsx |
|  | Save\_as\_pic() | Сохраняет в формате pic |
|  | Get\_frame() | Возвращает текущую frame |
| BarPlot(DataBase) | \_\_init\_\_(parent, file\_path) | Инициализация вкладки для BarPlot – столбчатая диаграмма  Для HistPlot – гистограмма  BoxVisk – диаграмма Бокса-Уискера  Scatter – диаграмма рассеяния |
| Histplot(DataBase) | Transform\_type() | Изменяет тип базы данных dataframe в соответствии с ее типами данных |
| BoxVisk(DataBase) | Define\_quality\_attr() | Определяет качественные атрибуты |
| Scatter(DataBase) | Define\_quantity\_attr() | Определяет количественные атрибуты |
|  | Init\_graph() | Рисует график |
|  | Init\_manager() | Создает менеджер для настройки графика |
|  | Save\_default() | Сохраняет график по умолчанию в папку output/ |
|  | Save() | Сохраняет в указанную пользователем папку в формате jpg |
|  | Get\_frame() | Возвращает текущую frame |