

**Руководство разработчика к аналитическому  
приложению AnalyticsApp. Комментарии к  
функциям, определённым в коде.**

Разработчик и автор отчёта: Чанке И. С.

Москва 2020

Данный документ содержит описание структуры программного кода и комментарии к функциям, составляющим приложение. Основная преследуемая здесь цель – упростить понимание архитектуры программы.

## Используемые библиотеки

Программа написана на Python 3.7 с использованием перечисленных ниже библиотек для следующих целей:

- *tkinter* – графический интерфейс пользователя
- *pandas* – представление и обработка данных в табличной форме
- *pickle* – сохранение и загрузка базы данных в двоичном формате
- *matplotlib.pyplot* – построение графиков
- *seaborn* – высокоуровневое API для построения графиков на базе *matplotlib*
- *numpy* – математические вычисления
- *io* – управление потоками ввода/вывода

## Структура кода

Весь код за исключением импортирования и настройки библиотек работает внутри цикла для графического интерфейса tkinter (mainloop). Код внутри цикла можно разбить на следующие основные блоки:

- **БЛОК 1. Глобальные переменные (строки 21-55).** В данном блоке определены переменные, отвечающие за состояние тех параметров приложения, которые используются и изменяются различными функциями, вызывающимися в ходе работы программы. К таким переменным относятся:
  1. **database** – словарь, хранящий загруженную в приложение базу данных. Каждая таблица связана с соответствующим ключом словаря (stations, lines, metros, merged)

2. **current\_filtered** – хранит последнее построенное пользователем отображение для последующего экспорта
  3. **current\_file\_name** – хранит имя файла, в котором записана загруженная в приложение база
  4. **open\_tab** – имя текущей открытой вкладки таблицы
  5. **len\_1/len\_2/len\_3/len\_4** – количество строк в загруженных таблицах
  6. **sel\_1/sel\_2/sel\_3/sel\_4** – количество выбранных строк в каждой вкладке таблицы.
  7. **number\_of\_rows** – переменная класса tk.StringVar, хранящая данные о количестве строк в текущей открытой таблице в текстовом формате
  8. **number\_selected** - переменная класса tk.StringVar, хранящая данные о количестве выбранных строк в текущей открытой таблице в текстовом формате
- **БЛОК 2. События и управляющие функции (строки 56 - 2111).** Самый объёмный, основной блок кода, в котором определены следующие функции, вместе составляющие механизм работы приложения:
    1. **configure\_f** – функция, отвечающая за фильтрацию открытой таблицы. Также определяет окно настройки фильтров. Внутри имеет следующие функции:
      - **clear\_all\_entries** – очищает все строки окна настройки фильтров
      - **apply** – берёт значения фильтров из строк окна ввода и фильтрует открытую таблицу

2. **analysis\_function** – отвечает за инструменты анализа таблицы.

Определяет окно выбора столбцов для анализа. Внутри имеет следующие функции:

- **insert\_columns** – вставляет в окно выбора столбцов столбцы в зависимости от текущей открытой таблицы.
- **select\_cols\_to\_process** – выполняет анализ с использованием выбранных колонок. В зависимости от выбранного инструмента анализа строит и показывает графики или сводную таблицу
- **check\_possibility** – вспомогательная функция, на вход принимает список и множество, проверяет, все ли элементы списка встречаются в множестве. Используется функцией `select_cols_to_process`

3. **print\_info** – выводит информацию об открытой таблице в блок «Информация о таблице».

4. **export\_pivot** – экспортирует сводную таблицу, построенную функцией `select_cols_to_process` функции `analysis_function`. Вызывается кнопкой в окне сводной таблицы.

5. **export\_representation** – экспортирует в формате Excel последнее построенное пользователем отображение.

6. **export\_selected** – экспортирует выбранные строки открытой таблицы в формате Excel

7. **delete\_selected** – удаляет выбранные строки из базы. В функции прописан механизм поддерживания базы в 3NF – при удалении строки, содержащей внешней ключ, на который ссылаются строки другой таблицы, соответствующие строки также удаляются.

8. **add\_instance** – позволяет добавить строку в текущую открытую таблицу. Внутри функции определяется окно создания новой строки, а также механизмы, поддерживающие базу в 3NF – запрет повторения первичных ключей и ссылки на несуществующий внешний ключ. Внутри функции определены следующие вспомогательные:

- **clear\_all\_entries** – очищает все строки окна создания строки
- **create\_new\_instance** – делает основную работу по созданию строки: забирает значения из окна создания, строит строку и добавляет её в базу

9. **edit\_instance** – позволяет редактировать выбранную строку текущей таблицы. Внутри функции определяется окно редактирования строки, а также механизмы, поддерживающие базу в 3NF – запрет повторения первичных ключей и ссылки на несуществующий внешний ключ. Внутри функции определены следующие вспомогательные:

- **clear\_all\_entries** – очищает все строки окна редактирования строки
- **alter\_row** - делает основную работу по редактированию строки: забирает значения из окна редактирования, изменяет в соответствии с ними выбранную строку

Функции **add\_instance**, **edit\_instance**, **configure\_f** используют окна ввода информации пользователем, построенные по одной и той же логике практически одинаковым кодом.

10. **buttons\_state** – управляет состоянием (активностью/неактивностью) кнопок главного окна приложения. Вызывается каждый раз при смене таблицы или выборе функции анализа, также при выборе строк таблицы

11. **manage\_open** – управляет состоянием кнопок приложения (для чего ссылается на `buttons_state`) и отображаемой в данный момент на экране информацией. Вызывается каждый раз при смене вкладки таблицы.
12. **select** – управляет счётчиком выбранных в данный момент строк таблицы. Ссылается на `buttons_state` для управления активностью кнопок, работающих с выбранными строками.
13. **sort** – сортирует по возрастанию выбранный столбец текущей таблицы (создаёт отображение). Вызывается кликом левой кнопкой мыши по названию столбца в таблице.
14. **unsort\_f** – убирает сортировку, также создаёт отображение (неотсортированная таблица)

Следующие функции вызываются кнопками выпадающего меню «Управление базой»

15. **open\_file** – загружает базу из файла в приложение.
16. **new\_base** – инициализирует новую пустую базу.
17. **base\_from\_csv** – позволяет создавать базу в двоичном формате из файлов в формате csv. Определяет окно выбора файлов csv. Включает в себя следующие вспомогательные функции:
  - **select\_file** – отвечает за выбор файла с csv таблицей
  - **create\_database** – создаёт двоичную базу из трёх выбранных таблиц
18. **save** – сохраняет текущую базу. Вызывается кнопкой «Сохранить».

19. **save\_as** – вызывается кнопкой «Сохранить как». Сохраняет базу в указанном файле.

20. **populate\_table** – заполняет treeview данными pandas dataframe.

- **БЛОК 3. Основной графический интерфейс (строки 2111+).** Блок кода, отвечающий за графический интерфейс основного окна. В данном блоке определены treeview для таблиц, кнопки, а также все остальные элементы интерфейса.