**Пояснительная записка**

1. Импортированные библиотеки:
   1. Стандартные: csv, sqlite3, sys
   2. Нестандартные: PyQT
2. Класс Main
   1. \_\_init\_\_
      1. Происходит инициализация окна из UI файла
      2. Определяется вызов какой функции будет на какое действие
   2. Set connection
      1. Открывается стандартный диалог выбора файла
      2. Происходит соединение с базой данных типа SQLite3
   3. Default query
      1. Происходит чтение из базы данных со стандартным запросом для этой программы
   4. Update table widget
      1. Обновление таблицы на экране с информацией, передающейся при вызове функции
   5. Item add
      1. Начинает работу при нажатии кнопки «Добавить»
      2. Открывается стандартный диалог ввода информации с Combo Box, в котором нужно выбрать, что хочет добавить пользователь. Также, представлена инструкция, как правильно добавлять произведение. Почему произведение нужно добавлять так, а не как иначе будет объяснено в пункте (4.1)
      3. После того, как пользователь нажал ОК в диалоге происходит инициализация «кастомного» диалога. (3)
   6. Item changing
      1. При двойном нажатии на ячейку в таблице открывается тот же «кастомный» диалог, что и в предыдущем пункте, но с предустановленными значениями в полях ввода (за исключением combo box)
   7. Item del
      1. Начинает работу при нажатии кнопки «Удалить»
      2. Открывается Message Box, который в зависимости от выбранного поля выводит разную информацию:
         1. Произведение: «Вы действительно хотите удалить это произведение?»
         2. Автор: «Вы действительно хотите удалить этого автора?»
         3. [Книга; ISBN]: «Вы действительно хотите удалить эту книгу?»
      3. Важно понимать, что при Удалении
         1. Произведения – удаляется произведение,
         2. Книги – удаляется книга, вместе с произведениями
         3. Автора – все произведения и книга автора + сам автор.
   8. Save 2 CSV (2 – to на сленге)
      1. Сохранение таблицы в CSV файл
   9. Sort columns
      1. Сортирует строки в таблице по выбранному параметру в Combo Box
   10. Log
       1. На лэйбле в самом низу пишет текст, что операция прошла успешно/неуспешно
   11. Search
       1. Поиск происходит по выбранному в Combo Box параметре. В том самом, в котором происходит выбор, как будут сортироваться строки, за исключением: Года, Страниц, Места. (4.2.2). Сам поиск происходит со стандартным запросом в базу данных + идёт выбор похожих на текст в поиске поля, соответствующего выбранному, в строках
   12. Close event
       1. При сохранении происходит commit с базой данных
3. Классы Add/Edit Author/Book/Work
   1. Разница Add и Edit классов в том, что в Edit все поля ввода заполнены, для того чтобы их изменить
   2. Add классы добавляют авторов/книги/работы в базу данных
   3. Edit классы изменяют авторов/книги/работы в базе данных
   4. В обоих классах операции происходят на прямую с базами данных
4. Ответы на возможные вопросы
   1. Почему произведения нужно добавлять именно таким образом?
      1. Можно сначала добавить книгу, потом автора, но логичнее вариант указанный в диалоге.
      2. Это связано с устройством таблицы. Связующим звеном является таблица произведений. В ней есть ссылка на таблицу с книгами и таблицу с авторами. Это сделано для удобства при выводе информации и просто логичнее. Допустим, я хочу прочитать произведение, что я о нём знаю? Автора, название произведения, а в какой книге оно лежит? Или же: Я хочу узнать какие произведения конкретного автора лежат в конкретной книге. Или: Я знаю произведение, в какой книге оно написано. А какой автор?
   2. Про Combo Box для поиска и сортировки таблицы одновременно
      1. Логично, что если пользователь уже отсортировал таблицу по книгам, то его интересует найти книгу. Аналогично с другими параметрами в Combo Box.
      2. Мне кажется, что не очень логично искать что-то в таблице по конкретному количеству страниц + поиск происходит от 3х символов в строке, а больше, чем 1000 страниц, может быть, будет в какой-нибудь энциклопедии Оксфорда, но не в Мертвых Душах Гоголя. Искать по конкретному месту/году можно отсортировав таблицу по нужному параметру.
   3. Почему классы типа Add и Edit не объединить в один?
      1. Это сделано для удобства программирования. Так программа выглядит красивее, чем если бы классы были с кучей флагов, как что сохранить, изменить, вывести, и т.п.
   4. Оставшиеся вопросы пишите на почту y.valentin123@yandex.ru