TALLINNA TÖÖTUSHARIDUSKESKUS  
Noorem Tarkvaraarendaja

**Andmebaasi haldamine**Referaat

Õpilane: Aleksander Milishenko  
Rühm: TARpv23  
Juhendaja: Marina Oleinik

Tallinn  
2024

sisukord

# ****Sisesjuhatus****

Задача проекта является создание программы для упрощения работы с базой данных библиотек, используя упрощенный графический интерфейс для работы.

Программа будет полезна для администраторов или тех кто просто работает с базой данных.

Данная программа будет актуальна для тех, кто не хочет работать с базой данных с помощью профессиональных программ администрирования баз данных, а просто с программой которая может выполнять только лишь важные для администрирования функции.

Программа будет актуальна на текущий год тем, что внее встроен графический интерфейс и она не требовательна к железу.

Для создания программы были использованы язык программирования, среды разработки, которые будут описаны далее

За фундамент идеи программы используется концепция таблиц и полей с помощью **«SQL».**

# Teoreetiline osa

## Keel

Для программы был использован такой язык программирования как – «**Python 3.12.3**»

**Python** — это язык программирования, который широко используется в интернет-приложениях, разработке программного обеспечения, науке о данных и машинном обучении. Разработчики используют Python, потому что он эффективен, прост в изучении и работает на разных платформах. Программы на языке Python можно скачать бесплатно, они совместимы со всеми типами систем и повышают скорость разработки.

**Язык структурированных запросов (SQL) – это язык программирования для хранения и обработки информации в реляционной базе данных**. Реляционная база данных хранит информацию в табличной форме со строками и столбцами, представляющими различные атрибуты данных и различные связи между значениями данных. Инструкции SQL можно использовать для хранения, обновления, удаления, поиска и извлечения информации из базы данных. Можно также использовать SQL для поддержания и оптимизации производительности базы данных.

## Vahendid

Среда для разработки была выбрана – «**Visual Studio Code**», с использованием расширения от Microsoft под названием «**Python**», для коректной работы с **«Python»**

*(* *Версия* *расширения* ***2022.4.1*** *).*

«**Visual Studio Code**» (**VS Code**) — это редактор кода для разных языков программирования. Он относительно немного весит, гибкий и удобный. В нем можно писать, форматировать и редактировать код на разных языках.

**VS Code** не стоит путать с **Visual Studio** — это **IDE**, очень мощная и масштабная, но одновременно с этим тяжеловесная. Названия похожи, потому что у обоих средств разработки один и тот же создатель, но продукты разные по своей сути. **VS Code** меньше весит, проще в освоении и подходит в том числе для начинающих разработчиков.

Редактор кода существует для всех популярных операционных систем: **Windows**, **Linux** и **macOS**. Он бесплатный, в отличие от большинства версий «старшего брата» **Visual Studio.**

Хранение и поддержка кода осуществлялась с помощью «**GitHub».**

**«GitHub»** — это облачная платформа для хостинга IT-проектов и совместной разработки, под капотом которой находится популярная система контроля версий Git, а также полноценная социальная сеть для разработчиков.

## Moodulid

Для написания программы потребовались встроенные модули от языка программирования «Python», такие как – «**tkinter**», «**sqlite3**», «**os**» и собственный модуль «**sqlite-handler**» для упрощенной работы с модулем «**sqlite3**».

Модуль «**Tkinter»** – **библиотека, позволяющая создавать приложения с графическим интерфейсом**. Преимуществами **Tkinter** считаются встроенность в стандартный пакет **Python** (ничего дополнительно устанавливать не нужно) и кроссплатформенность, позволяющая писать приложения для разных операционных систем.

Модуль «**SQLite3»** - **предоставляет инструменты для работы с базами данных SQLite, легкой и встроенной системой управления базами данных**. SQLite позволяет хранить данные в локальных файлах, что делает его удобным выбором для многих приложений с небольшими объемами данных.

Модуль «**OS»** - **предоставляет множество функций для работы с операционной системой**, причём их поведение, как правило, не зависит от ОС, поэтому программы остаются переносимыми.

Модуль «**SQLite-Handler» –** модуль написанный в целях упрощения работы с **SQL** запросами, за основу модуля был взят модуль «**sqlite3**». Сам по себе модуль предоставляет возможность работы с помощью классов, то-есть каждая таблица / поле создается при помощи экземпляра классов. Данный модуль открыт для общего использования на **Git Hub.**

# pRAKTILINE OSA

## Loomise protsess

Процесс создания программы длился на протяжении недели. Были использованы абсолютно все перечисленные ранее модули.

Модуль «OS» , был использован для совместной работы с модулем «sqlite3», для поиска и создания файла где хранилась бы дата база с расширением «.db»

Для начала разработки программы потребовалось модернизировать модуль «sqlite-handler» для расширения нынешнего функционала модуля.

Было создано с помощью модуля «sqlite-handler» 3 экземпляра класса «SQLHTable», данный класс предоставлял возможности создания и редактирования таблицы.

Название для трех таблиц были следующими – „raamatud“, „autorid“, „žanrid“

Далее последовало создания полей с помощью класса «SQLHColumn» и дальнейшее привзяванние их к экземплярам класса «SQLHTable».

Таблица «**Autorid**» имела такие поля как –

* **autor\_id**
  + Данное поле хранит уникальные индификаторы авторов. Поле является первичным ключом таблицы
* **autor\_nimi**
  + Данное поле хранит в себе имена авторов.
* **sunnikuupaev**
  + Данное поле хранит в себе год, день, месяц рождение авторов

Данная таблицы является „вторичной“ для таблицы «Raamatud»

Таблица «**Zanrid**» имела такие поля как –

* žanr\_id
  + Данное поле хранит в себе уникальные индификаторы жанров. Поле является первичным ключом таблицы
* žanr\_nimi
  + Данное поле хранит в себе имена жанров.

Данная таблицы является „вторичной“ для таблицы «Raamatud»

Таблица «**Raamatud**» имела такие поля как –

* raamat\_id
  + Данное поле хранит в себе уникальные индификаторы для книг. Данное поле является первичным ключом таблицы
* pealkiri
  + Данное поле хранит в себе заголовки книг.
* valjaandmise\_kuupaev
  + Данное поле хранит в себе год выпуска книг
* autor\_id
  + Данное поле хранит в себе уникальные индификаторы авторов из таблицы «Autorid». Данное поле является вторичным ключом.
* žanr\_id
  + Данное поле хранит в себе уникальные индификаторы жанров из таблицы «Zanrid». Поле является вторичным ключом.

После создания таблиц и полей мы создаем экземпляр класса «SQLHDatabase», этот класс является дочерним классом, родительский класс – «SQLHandler».

Данный экземпляр помогает нам работать напрямую с базой данных с помощью функционала класса.

Работает класс таким образом, что он „образно“ привязывает к себе файл где хранятся данные и работает с ним с помощью модуля «sqlite3» через **SQL запросы** .

С помощью специального метода класса базы данных мы добавляем экземпляры таблиц в наш экземпляр.

При добавлении экземпляров таблиц, наша база данных проверяет на наличие сущевствующих записей и если таковы и имеются, то она добавляет к экземплярам таблиц их записи которые хранились в файле привязонном к экземпляру базы даных.

После этой подготовки можно работать с базой данных, а тоесть добавлять, изменять или удалять записи. Изменение / добавление полей не было нужно, поэтому данный функционал был вырезан из модуля «sqlite-handler».

Далее для создания графического интерфейса для упрощения работы потребовался модуль «tkinter» .

Был создан отдельный класс для работы с окном, в окно входило два главных компонента, а именно – меню кнопок, главный экран

Переключение окон в окне было продумано таким образом, что для начало абсолютно все элементы окна уничтожаются с помощью специальной функции «reset\_window» из созданного класса для управления интерфейсом -

  def reset\_window(self, buttons):

      if self.buttons\_frame is not None:

        self.clear\_frame(self.buttons\_frame)

      self.buttons\_frame = self.render\_menu\_buttons(buttons)

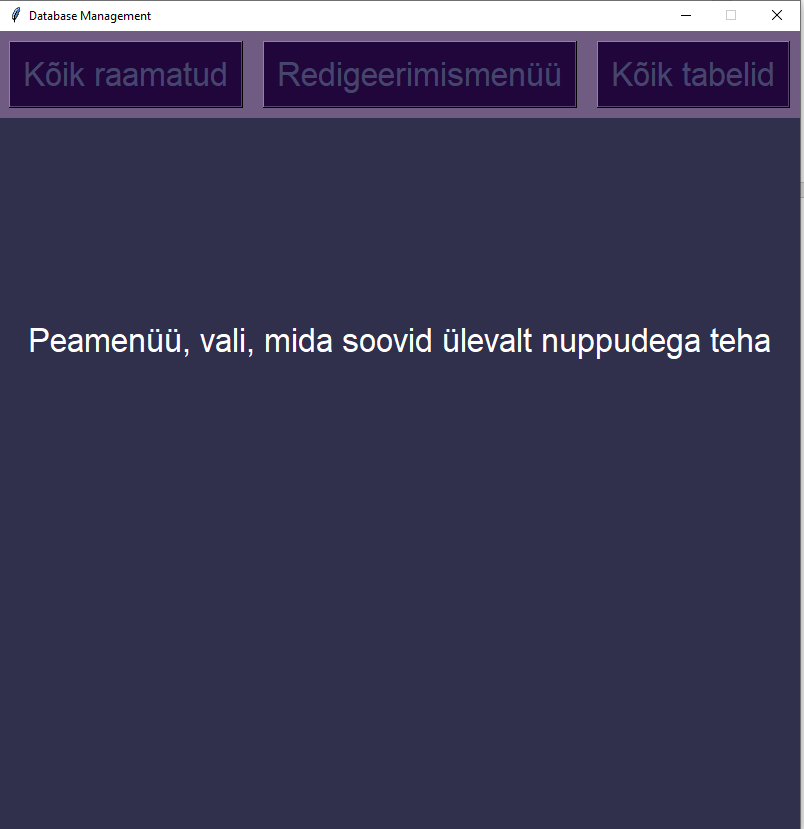
      if self.main\_frame is not None:

        self.clear\_frame(self.main\_frame)

      self.main\_frame = self.render\_main\_frame()

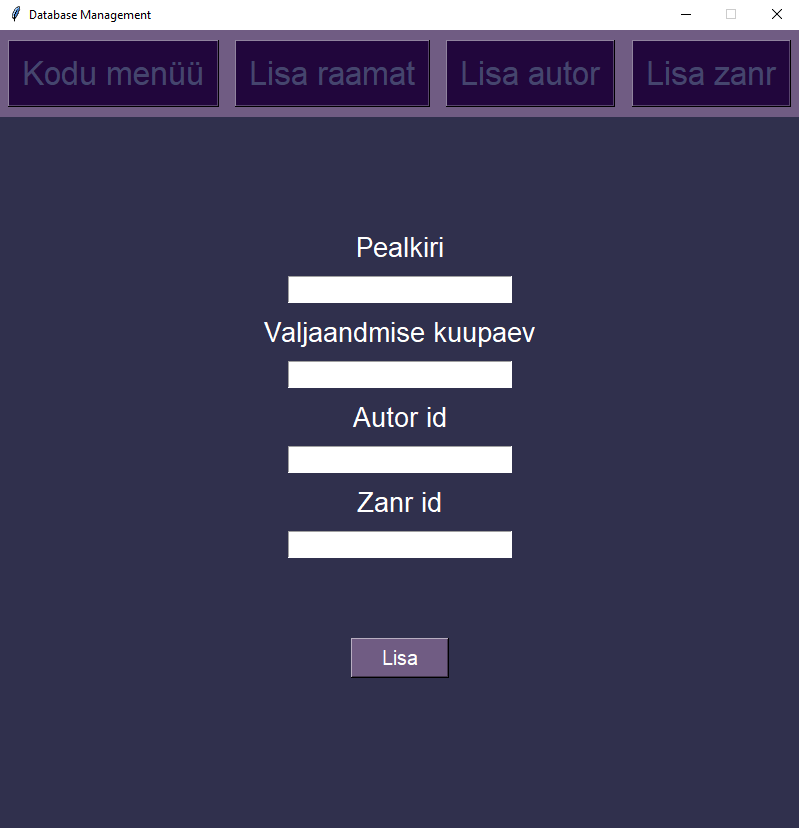
в данную функцию передается аргумент , в котором должен быть список из кнопок, а тоесть имя кнопки и функция которая будет запускаться при нажатии на кнопку.

После уничтожения элементов генерируется меню кнопок с помощью в функции «render\_menu\_buttons», которая была задействована как раз таки в «reset\_window» и основное окно в котором будут второстепенные элементы для работы с программой.

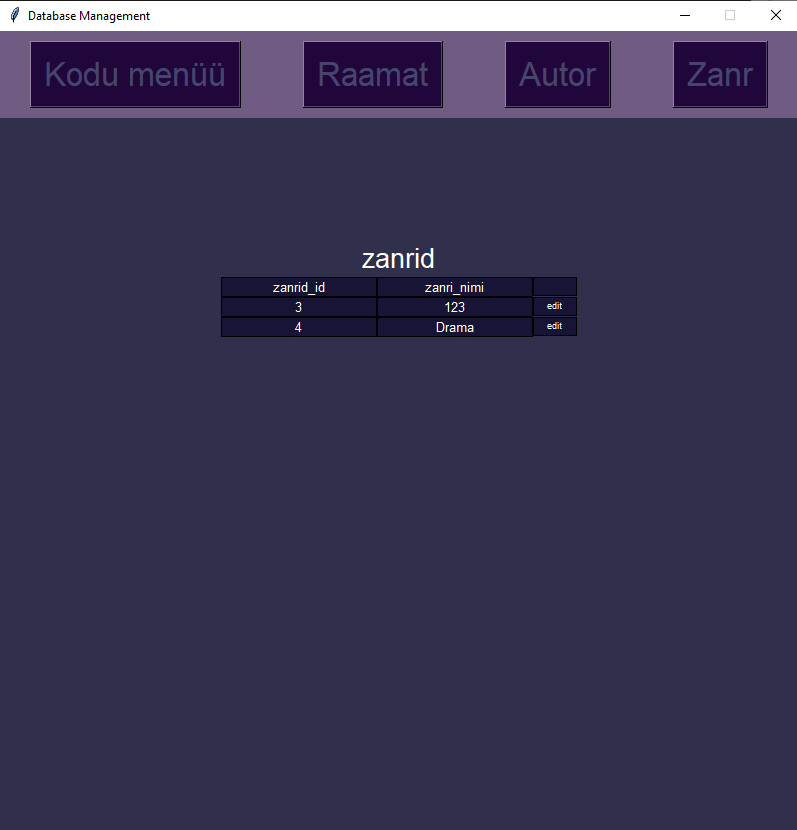


**Начальное окно.**

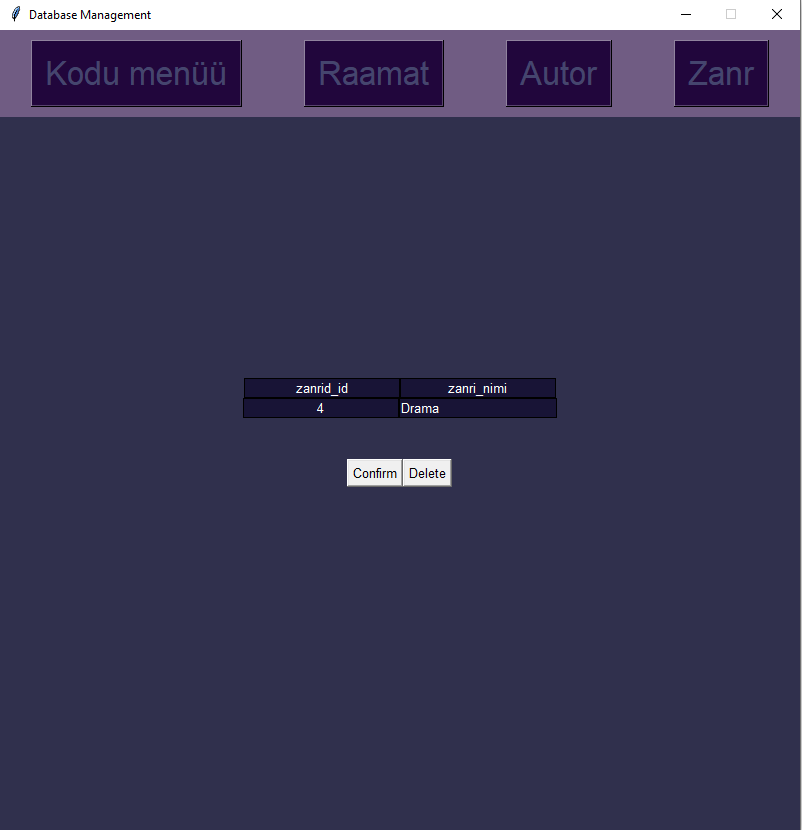
Пользователю предоставляются 3 кнопки для начала, первая кнопка должна отвечать за вывод абсолютно всех книг, вторая кнопка отвечает за добавление записей в таблицы, третья кнопка отвечает за вывод всех записей и возможности обновления/ удаления данных.

**Окно добавления .**

Оно одинаковое для каждой таблицы, но меняются поля для заполнения. Работает оно следующим образом – пользователь заполняет поля и нажимает на кнопку «Lisa» , при нажатии на кнопку срабатывает функция (уникальная для каждой таблицы), в которой генерируется SQL запрос для добавления в таблицу записи, после добавления пользователя перемещает окно где отображены все записи той таблицы в которую он добавлял поля.

**Окно вывода всех данных**.

В данном окне обрабатываются абсолютно все записи таблицы которой пользователь хочет увидеть. Рядом с записями добавляются отдельные кнопки для каждой записи , их функционал заключается в том, что при нажатии кнопки изменения определенной записи вас перекидывает на окно где вы можете либо изменить запись, либо удалить.



**Окно изменения записи.**

В данном окне у нас обрабатывается и показывается та запись, которую мы захотели изменить.

Мы можем удалить ее с помощью кнопки „delete“ либо изменить значение полей.

При нажатии «confirm» срабатывается функция «confirm\_change»

# alikad

<https://pythonworld.ru/moduli/>

<https://selectel.ru/blog/tutorials/tkinter-library-in-python/>

<https://aws.amazon.com/ru/what-is/>

<https://www.python.org/>

<https://code.visualstudio.com/>

https://skillbox.ru/