**Руководство программиста.**

**Руководство программиста на разработку программы "Договоры на продажу жилья".**

**1. Общее описание**

Программа предназначена для управления договорами на продажу жилья. Она позволяет загружать, сохранять, добавлять, редактировать и удалять записи, а также визуализировать данные с помощью графиков. Интерфейс реализован с использованием библиотеки Tkinter.

**2. Назначение**

Программа предназначена для агентов по недвижимости и менеджеров, которые хотят эффективно управлять своими договорами, анализировать данные и визуализировать результаты работы.

**3. Описание функциональных блоков**

**3.1. Класс Contract**

* **Назначение**: Представляет один договор.
* **Атрибуты**:
  + property\_type: тип жилья (строка).
  + manager: имя менеджера (строка).
  + price: цена (число с плавающей точкой).

**3.2. Класс ContractManager**

* **Назначение**: Управляет коллекцией договоров.
* **Методы**:
  + load\_from\_file(filename): Загружает данные из файла.
  + save\_to\_file(filename): Сохраняет данные в файл.
  + add\_contract(property\_type, manager, price): Добавляет новый договор.
  + delete\_contract(index): Удаляет договор по индексу.
  + update\_contract(index, property\_type, manager, price): Обновляет данные договора.
  + segment\_by\_property\_type(): Сегментирует данные по видам жилья.
  + segment\_by\_manager(): Сегментирует данные по менеджерам.
  + plot\_pie\_chart(data, title): Строит круговую диаграмму.
  + get\_contracts\_info(): Возвращает информацию о всех договорах.

**3.3. Класс ContractApp**

* **Назначение**: Основной интерфейс приложения.
* **Методы**:
  + load\_data(): Загружает данные из выбранного файла.
  + open\_add\_dialog(): Открывает диалог для добавления нового договора.
  + update\_table(): Обновляет отображаемую таблицу данных.
  + show\_pie\_chart(fig): Отображает график в новом окне.
  + segment\_property(): Сегментирует данные по видам жилья и отображает график.
  + segment\_manager(): Сегментирует данные по менеджерам и отображает график.
  + on\_item\_double\_click(event): Обрабатывает двойное нажатие для редактирования договора.

**3.4. Класс AddContractDialog**

* **Назначение**: Диалоговое окно для добавления нового договора.
* **Методы**:
  + add\_contract(): Добавляет новый договор, проверяя заполненность полей и корректность данных.

**3.5. Класс EditContractDialog**

* **Назначение**: Диалоговое окно для редактирования существующего договора.
* **Методы**:
  + save\_contract(): Сохраняет изменения в договоре.
  + delete\_contract(): Удаляет договор.

**4. Структура кода**

Код организован по принципу модульности, каждый класс выполняет свою специфическую задачу. Используется обработка исключений для обработки ошибок при работе с файлами и пользовательскими вводами. Графический интерфейс построен с использованием стандартных виджетов Tkinter и более сложных элементов из ttk.

**5. Использование**

Для запуска программы необходимо иметь установленный Python и необходимые библиотеки (tkinter, matplotlib). Программа запускается из главного файла, который создает основное окно и запускает главный цикл обработки событий.

**6. Пример работы**

1. При запуске программы открывается основное окно с кнопками для загрузки данных, добавления новых записей и сегментации.
2. Пользователь может выбрать файл для загрузки данных, после чего информация отобразится в таблице.
3. Для добавления нового договора открывается отдельное диалоговое окно.
4. Доступна функциональность для редактирования и удаления записей через двойное нажатие на строку в таблице.
5. Сегментация данных позволяет пользователю визуализировать информацию в виде круговых диаграмм.

**7. Заключение**

Программа предоставляет удобный инструмент для работы с договорами на продажу жилья, включая функционал для анализа и визуализации данных, что делает ее полезной для специалистов в сфере недвижимости.