МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Измерительно-вычислительные комплексы»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | |  | |  |  |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | |  | | ТЕХНИЧЕСКОЕ ЗАДАНИЕ  на лабораторную работу №8  по дисциплине «Алгоритмы и структуры данных»  Тема «Управление договорами на аренду автомобилей» | | | | | | |
|  | | |  | |  | | | |
|  | | Исполнитель  студент гр. ИСТбд-21  Пирогов П.Е.  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2024 г. | | | |
|  | | 2024 | | | | | | |

**Введение**

Данный документ предназначен для программистов, которые будут поддерживать или расширять функциональность приложения "Управление договорами на аренду автомобилей". Программа разработана на языке Python с использованием библиотеки Tkinter для графического интерфейса и библиотеки Matplotlib для построения диаграмм. Руководство описывает архитектуру проекта, детали реализации основных классов и методов, а также инструкции для запуска и тестирования.

**Среда разработки**

Для работы с проектом необходимы следующие инструменты и библиотеки:

1. Python 3.x – основной язык программирования.

2. Tkinter – для создания графического интерфейса (входит в стандартный дистрибутив Python).

3. Matplotlib – для построения круговых диаграмм. Устанавливается с помощью команды pip install matplotlib.

**Структура проекта**

Проект состоит из одного файла с кодом, который включает:

- Классы для хранения и управления договорами на аренду автомобилей.

- Логику загрузки данных из файла.

- Сегментацию данных по категориям.

- Построение круговых диаграмм для визуализации данных.

**Описание основных классов и модулей**

Класс RentalContracts:

**Описание**: Этот класс управляет данными о договорах на аренду автомобилей. Он загружает данные из CSV-файла, хранит их и выполняет сегментацию по заданным критериям. Также включает методы для визуализации результатов.

**Основные методы**:

* load\_contracts(): загружает данные из CSV файла. Для каждого договора сохраняется его идентификатор, имя клиента и тип автомобиля. Также обновляется список типов автомобилей.
* segment\_by\_car\_type(): сегментирует договоры по типам автомобилей, возвращая словарь, где ключи — это типы автомобилей, а значения — количество договоров для каждого типа.
* segment\_by\_clients(): сегментирует договоры по клиентам, возвращая словарь, где ключи — это имена клиентов, а значения — количество договоров для каждого клиента.
* visualize\_pie\_chart(): строит круговую диаграмму на основе переданных данных. Визуализирует сегментацию по заданному критерию (например, по типам автомобилей или по клиентам).

Класс App:

**Описание**: Этот класс представляет графический интерфейс приложения. Он реализует кнопки для загрузки данных, сегментации данных по типам автомобилей и клиентам, а также отображения диаграмм.

**Основные методы**:

* load\_file(): открывает диалоговое окно для выбора файла и загружает данные с помощью класса RentalContracts.
* segment\_by\_car\_type(): выполняет сегментацию данных по типам автомобилей и отображает результат в виде круговой диаграммы.
* segment\_by\_clients(): выполняет сегментацию данных по клиентам и отображает результат в виде круговой диаграммы.

**Детали реализации**

**Интерфейс загрузки данных:**

- При нажатии на кнопку "Загрузить данные" вызывается метод load\_file. Этот метод открывает окно выбора файла, загружает данные и передает их в класс RentalContracts для дальнейшей обработки.

- Метод load\_contracts проверяет корректность данных: каждая строка должна содержать три значения, разделенных запятыми (идентификатор договора, имя клиента и тип автомобиля). Если данные не соответствуют этому формату, выводится сообщение об ошибке через окно messagebox.showerror.

Сегментация данных:

- Метод segment\_by\_car\_type формирует сегментацию данных по типам автомобилей (например, "седан", "внедорожник" и т.д.), возвращая словарь, где ключи — это типы автомобилей, а значения — количество договоров для каждого типа.

- Метод segment\_by\_clients сегментирует договоры по клиентам, возвращая словарь, где ключи — это имена клиентов, а значения — количество договоров для каждого клиента.

**Построение диаграмм:**

- Для визуализации сегментированных данных используются круговые диаграммы с помощью метода plt.pie из библиотеки Matplotlib.

- Метод visualize\_pie\_chart строит круговую диаграмму распределения договоров по выбранной категории (например, по типам автомобилей или по клиентам), передавая данные и заголовок диаграммы.

**Тестирование и отладка**

**Тесты загрузки данных:**

- Проверьте, что программа правильно обрабатывает файл с корректным и некорректным форматом данных.

- Убедитесь, что при ошибках ввода выводится соответствующее сообщение.

**Тесты сегментации:**

- Проверьте, что сегментация по типу автомобиля корректно распределяет договоры на основе их атрибута car\_type.

- Тестируйте сегментацию по продолжительности аренды для значений, находящихся на границах категорий ("краткосрочные", "среднесрочные", "долгосрочные").

**Тесты визуализации:**

- Убедитесь, что построение диаграмм успешно работает при наличии данных и адекватно отображает распределение.

- Проверьте корректность названий категорий на диаграммах.

**Заключение**

Этот документ описывает основные компоненты и функции проекта "Управление договорами на аренду автомобилей". Классы и методы реализованы с учетом модульности и могут быть расширены для добавления новых функций, таких как сортировка договоров или дополнительная фильтрация данных.