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*** |  | |  | | Руководство программиста  на лабораторную работу №8  по дисциплине «Алгоритмы и структуры данных»  Тема «Разработка программы для управления кредитными договорами» | | | | | | |
|  | | |  | |  | | | |
|  | | Исполнитель  студент гр. ИСТбд-21  Саранцев С.А.  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2024 г. | | | |
|  | | 2024 | | | | | | |

**1. Введение**

Это руководство предназначено для программистов, которые будут поддерживать или расширять функциональность программы для управления кредитными договорами, разработанной с использованием Python,Tkinter и Matplotlib. В руководстве описываются основные архитектурные решения, структура проекта и порядок его запуска.

**2. Среда разработки**

Для работы с проектом требуется установить следующие инструменты:

* **Python 3.x**: основной язык программирования.
* **Tkinter**: библиотека для создания графических интерфейсов. Обычно включена в дистрибутив Python.
* **Matplotlib**: библиотека для создания графиков. Устанавливается с помощью pip install matplotlib.

**3. Структура проекта**

Проект состоит из одного основного файла, который содержит все необходимые классы и функции:

1. main.py: содержит классы CreditContract и ContractApp, а также основную логику программы.

**4. Описание основных классов и методов**

1. **CreditContract**
   * **Описание:** Класс, представляющий кредитный договор.
   * **Атрибуты:**
     + contract\_id: идентификатор договора.
     + amount: сумма договора.
     + manager: менеджер, ответственный за договор.
   * **Методы:**
     + \_\_init\_\_: конструктор класса.
     + \_\_repr\_\_: строковое представление объекта**.**
2. **ContractApp**
   * **Описание:** Основной класс приложения, управляющий графическим интерфейсом и логикой программы.
   * **Основные методы:**
     + \_\_init\_\_: инициализация приложения.
     + create\_widgets: создание элементов интерфейса.
     + load\_data: загрузка данных из файла.
     + segment\_and\_visualize\_by\_amount: сегментация и визуализация по суммам.
     + segment\_and\_visualize\_by\_manager: сегментация и визуализация по менеджерам.
     + visualize\_segmentation: отображение круговой диаграммы.
     + exit\_app: выход из приложения.

**5. Детали реализации**

1. **Загрузка данных**
   * Реализована через метод load\_contracts, который читает данные из выбранного пользователем файла.
   * Поддерживаются форматы .txt и .csv, где данные разделены запятыми.
2. **Сегментация данных**
   * Реализована в методах segment\_and\_visualize\_by\_amount и segment\_and\_visualize\_by\_manager.
   * Для сумм используются категории: малые (< 10000), средние (10000-50000), крупные (> 50000).
   * Для менеджеров подсчитывается количество договоров для каждого менеджера.
3. **Визуализация**
   * Использует библиотеку Matplotlib для создания круговых диаграмм.
   * Диаграммы встраиваются в интерфейс Tkinter с помощью FigureCanvasTkAgg.

**6. Расширение функциональности**

1. **Добавление новых типов сегментации**
   * Создать новый метод в классе ContractApp, реализующий логику сегментации.
   * Добавить новую кнопку в метод create\_widgets для вызова новой функции сегментации.
2. **Изменение критериев сегментации по суммам**

* Модифицировать логику в методе segment\_and\_visualize\_by\_amount, изменив пороговые значения для категорий.

1. **Добавление экспорта данных**

* Реализовать новый метод для сохранения результатов сегментации в файл.
* Добавить соответствующую кнопку в интерфейс.

**7. Тестирование и отладка**

1. **Основные тесты:**
   * Проверка корректности загрузки данных из файла.
   * Тестирование правильности сегментации по суммам и менеджерам.
   * Проверка корректности отображения круговых диаграмм.
2. **Отладка:**
   * Использовать print() для логирования промежуточных результатов.
   * Применять отладчик IDE для пошагового выполнения кода.

**8. Заключение**

Данное руководство описывает основные аспекты проекта по управлению кредитными договорами. Оно предоставляет необходимую информацию для понимания структуры кода, его функциональности и методов расширения. При возникновении вопросов обращайтесь к исходному коду программы или к автору проекта.