**Модель приложения "Кредитный контрактный менеджер"**

**1. Структуры данных**

Основные структуры данных:

• Кредитный договор (класс CreditContract):

    - contract\_id: str  # идентификатор договора

    - amount: float     # сумма договора

    - manager: str      # менеджер договора

• Список договоров:

contracts = [CreditContract(...), CreditContract(...)]

• Сегментация по суммам:

segments = {

    'Малые': [контракты < 10000],

    'Средние': [контракты от 10000 до 50000],

    'Крупные': [контракты > 50000]

}

• Сегментация по менеджерам:

manager\_segments = {

    'Иванов': [контракты менеджера],

    'Петров': [контракты менеджера],

    ...

}

**2. Алгоритмы**

**2.1 Алгоритм загрузки данных:**

- Открытие файла

- Построчное чтение

- Парсинг строк по разделителю ','

- Валидация данных

- Создание объектов CreditContract

- Добавление в список contracts

**2.2 Алгоритм сегментации по суммам:**

- Обход всех договоров

- Распределение по категориям:

  \* < 10000 - малые

  \* 10000-50000 - средние

  \* > 50000 - крупные

- Подсчет количества в каждой категории

- Визуализация круговой диаграммой

**2.3 Алгоритм сегментации по менеджерам:**

- Обход всех договоров

- Группировка по полю manager

- Подсчет количества для каждого менеджера

- Визуализация круговой диаграммой

**3. Дополнительные элементы модели**

**3.1 Контроль данных:**

- Проверка формата файла

- Валидация сумм (> 0)

- Проверка наличия всех полей

**3.2 Обработка ошибок:**

- Ошибки чтения файла

- Неверный формат данных

- Некорректные значения

**4. Визуализация**

**4.1 Круговые диаграммы:**

- Отображение процентного соотношения

- Легенда с категориями

- Заголовок диаграммы

**4.2 Интерфейс:**

- Кнопки управления

- Статус-бар

- Область отображения графиков

**5. Итеративность разработки**

Возможные улучшения:

- Добавление новых видов сегментации

- Расширение функций анализа

- Оптимизация алгоритмов

- Улучшение визуализации