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*** |  | |  | | Пояснительная записка  на лабораторную работу  по дисциплине «Алгоритмы и структуры данных»  Тема «Управление банковскими вкладами» | | | | | | |
|  | | |  | |  | | | |
|  | | Исполнитель  студент гр. ИСТбд-21  Клянченков М.А.  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2024 г. | | | |
|  | | 2024 | | | | | | |

**1. Введение**

Данный проект представляет собой объектно-ориентированное приложение с графическим интерфейсом для управления банковскими вкладами. Программа разработана на языке Python с использованием библиотеки Tkinter для создания графического интерфейса и библиотеки Matplotlib для визуализации данных. Цель проекта — автоматизировать работу с банковскими вкладами, а также предоставить удобный способ сегментации и визуализации вкладов по типу и сумме.

**2. Обоснование выбора технологий**

**Python** — основной язык разработки. Выбран за его простоту, удобство работы с файлами и наличие мощных библиотек для создания интерфейсов и визуализации данных.

**Tkinter** — встроенная библиотека Python для создания графического интерфейса. Она позволяет быстро и легко разрабатывать GUI-приложения, поддерживающие работу с файлами и простые формы.

**Matplotlib** — библиотека для построения графиков и диаграмм. Она используется для визуализации сегментации вкладов в форме круговых диаграмм, что позволяет наглядно представить распределение вкладов по категориям.

**3. Структура программы**

Программа состоит из трех основных классов:

* **BankDeposit** — класс, описывающий отдельный банковский вклад. Включает следующие атрибуты:
  + deposit\_id — идентификатор вклада.
  + deposit\_type — тип вклада (например, срочный, накопительный).
  + amount — сумма вклада.
* **DepositManager** — класс, выполняющий управление списком вкладов и их сегментацию. Реализует следующие методы:
  + load\_from\_file(filename) — метод для загрузки данных из файла с контролем правильности ввода. Данные должны быть разделены запятыми.
  + segment\_by\_type() — метод для сегментации вкладов по типу.
  + segment\_by\_amount() — метод для сегментации вкладов по размеру суммы: "мелкие" (до 20 000), "средние" (от 20 000 до 50 000) и "крупные" (свыше 50 000).
* **App** — класс, отвечающий за графический интерфейс. Создает основные элементы интерфейса и связывает кнопки с функциями обработки данных и визуализации. Включает:
  + Кнопку загрузки данных (load\_button).
  + Кнопки для сегментации и визуализации вкладов по типу и по сумме (segment\_type\_button, segment\_amount\_button).
  + Методы для визуализации сегментации в форме круговых диаграмм.

**4. Процесс разработки**

Разработка проекта включала следующие этапы:

1. **Определение структуры классов** — проектирование классов и их методов на основе функциональных требований.
2. **Реализация основной логики** — разработка классов BankDeposit и DepositManager с методами загрузки и сегментации данных.
3. **Создание графического интерфейса** — создание класса App с использованием библиотеки Tkinter и настройка элементов интерфейса.
4. **Реализация визуализации** — интеграция библиотеки Matplotlib для отображения данных в виде круговых диаграмм.
5. **Тестирование** — проверка корректности работы функций загрузки данных, сегментации и визуализации, а также тестирование обработки ошибок при чтении файла.

**5. Тестирование**

Проект был протестирован в следующих направлениях:

* **Корректность обработки файлов** — проверено чтение данных из файлов с корректным и некорректным форматом. При обнаружении ошибок в файле программа выводит сообщение пользователю.
* **Сегментация вкладов** — проверена правильность разделения вкладов по категориям, а также корректность отображения данных на диаграммах.
* **Интерфейс** — протестирована работа всех элементов GUI, в том числе кнопок и окон сообщений.

**6. Выводы**

Программа успешно реализует требования задания по управлению банковскими вкладами и визуализации данных. Она представляет собой легковесное решение для загрузки, обработки и отображения вкладов. В будущем проект может быть дополнен функционалом для работы с более сложными форматами данных, добавлением фильтров и другими визуальными средствами.