**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение

высшего образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Кафедра «Измерительно-вычислительные комплексы»

**Лабораторная работа**

**По дисциплине «Алгоритмы и структуры данных»**

**Тема Лабораторная работа №9**

|  |  |
| --- | --- |
| **Подп. и** |  |
| **Инв** |  |
| **Вза** |  |
| **Подп. и** |  |
| **Инв.** |  |

**Пояснительная записка**

**дата**

Р.02069337. <23/721>-<04> ПЗ-<2-зн. номер

редакции> Листов 6

**м. инв. №**

**. № дубл.**

**Исполнитель**: студент гр. ИСТбд-23

**дата**

*Зуев С. А.*

« 15 » ноября 2024 г.

2024

**№ подл.**

**Введение**

В данном проекте разрабатывается приложение для управления списком экспонатов. В приложении реализована функциональность для добавления, сегментации и визуализации данных о экспонатах. Используемый подход основан на объектно-ориентированном программировании с применением библиотеки Tkinter для графического интерфейса и Matplotlib для релевантной визуализации данных.

1. **Проектная часть**
   1. **Постановка задачи на разработку приложения**

Определяется заданием на курсовую работу. Детализируется в разработанном техническом задании.

* 1. **Математические методы**

Применяемая математическая модель для сегментации экспонатов основывается на классификации по двум параметрам: категориям и фирмам. Выбор этого подхода обусловлен необходимостью быстрого анализа и визуализации распределения экспонатов, что обеспечивает удобство для пользователей приложения.

* 1. **Архитектура и алгоритмы**
     1. Архитектура

Архитектура приложения включает следующие основные структуры данных и функции:

* Структуры данных:
  + items: список экспонатов (название, категория, фирма).
  + categories: словарь, хранящий сегментированные по категориям экспонаты.
  + firms: словарь, хранящий сегментированные по фирмам экспонаты.
* Функции:
  + load\_items(): загрузка экспонатов из файла.
  + save\_items(): сохранение экспонатов в файл.
  + add\_item(): добавление нового экспоната.
  + segment\_by\_category(): сегментация по категориям.
  + segment\_by\_firm(): сегментация по фирмам.
  + show\_pie\_chart(): отображение круговой диаграммы для визуализации данных.
    1. Алгоритм обработки данных

Пример алгоритма, который выполняется при добавлении нового экспоната:

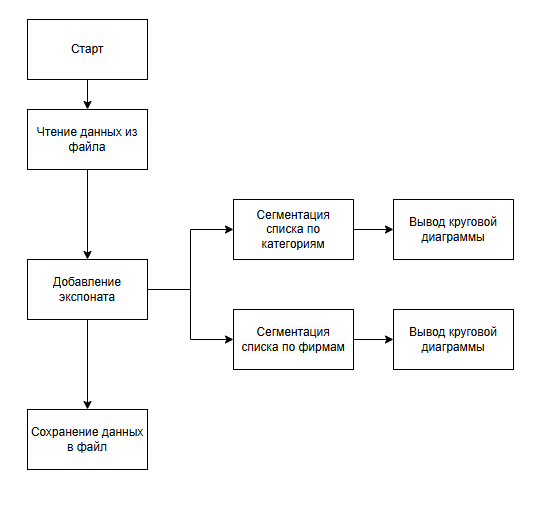
1. Запрос данных о новом экспонате (название, категория, фирма).
2. Проверка корректности введенных данных.
3. Если данные корректны:
   * Добавить экспонат в список items.
   * Обновить сегментацию по категориям и фирмам.
   * Визуализировать результат с помощью круговой диаграммы, если требуется.

Схема алгоритма оформляется в соответствии с ГОСТ 19.701-90, с описанием каждого шага без упоминания языка программирования или имен объектов.

* + 1. Алгоритм сегментации данных

Этот алгоритм обеспечивает сегментацию данных по категориям или фирмам для визуализации. Он состоит из следующих шагов:

1. Очистка текущих сегментов.
2. Пройдите по списку экспонатов.
3. Для каждого экспоната:
   * Добавьте его в соответствующую категорию или фирму.
4. Сформируйте данные для визуализации.
   1. **Тестирование**

****

* + 1. Описание отчета о тестировании

Отчет о тестировании предназначен для документирования всех тестовых сценариев и их результатов, а также для выявления проблем и уязвимостей в приложении.

* + 1. Цель тестирования

Проверить работу всех функций приложения, включая загрузку файлов, добавление экспонатов, сегментацию и визуализацию данных.

* + 1. Методика тестирования

Методы тестирования включают ручное тестирование интерфейса, автоматизированное тестирование обновлений и стресс-тестирование для проверки устойчивости к нагрузкам.

* + 1. Проведенные тесты

| **Тестовый случай** | **Описание** | **Ожидаемый результат** | **Фактический результат** | **Статус** |
| --- | --- | --- | --- | --- |
| TC1 | Тестирование загрузки файлов | Данные загружены и отображены в таблице. При выборе некорректного файла программа отобразит сообщение об ошибке. | Данные успешно загружены и отображены в таблице. При выборе некорректного файла программа отображает сообщение об ошибке. | Пройден |
| ТС2 | Тестирование добавления экспонатов | Экспонат добавлен в таблицу. | Экспонат и данные о нём успешно занесены в таблицу. | Пройден |
| ТС3 | Тестирование сегментации | Отображается круговая диаграмма с корректными данными. | Круговая диаграмма отображается с корректными данными. | Пройден |
| ТС4 | Тестирование сохранения данных | Данные сохраняются корректно и в правильном формате. | Файл успешно сохранён, данные сохранены корректно и в правильном формате. | Пройден |

* + 1. Выводы

Все тестовые случаи были успешно пройдены, что подтверждает корректность работы приложения.

Программа демонстрирует стабильную работу и устойчивость к ошибкам.

Пользовательский интерфейс удобен и интуитивно понятен.

1. **Источники, использованные при разработке**

https://metanit.com/python/tkinter/ Руководство по Tkinter