Лабораторная работа 2  
Документация

# Тема:

Создание веб-приложения для чтения и анализа метаданных графических файлов.

# Инструменты разработки:

HTML — для создания структуры интерфейса.

CSS — для стилизации элементов.

JavaScript — для обработки пользовательского ввода, работы с изображениями, загрузки файлов, извлечения и отображения метаданных.

DataTables — библиотека для создания таблиц с функциями сортировки, фильтрации и пагинации.

JSZip — библиотека для работы с ZIP-архивами.

GitHub Pages — для размещения веб-приложения в общем доступе.

PyCharm — среда разработки.

FileReader API — для чтения содержимого файлов в браузере.

# Использованные библиотеки и технологии:

1. HTML — разметка веб-страницы для создания элементов интерфейса:

- <input> — для загрузки изображений или ZIP-архивов с изображениями.

- <button> — для отправки изображений на обработку и очистки таблицы.

- <table> — для отображения полученной информации о файлах (формат, размер, глубина цвета и др.).

2. CSS — для стилизации интерфейса:

- Использование внешнего CSS-файла для оформления элементов интерфейса, таких как кнопки и таблицы.

- Интеграция стилей из библиотеки DataTables для удобного представления данных.

3. JavaScript — логика обработки файлов, получения их метаданных и взаимодействие с интерфейсом:

- FileReader API — чтение загружаемых файлов и обработка их содержимого.

- JSZip — обработка ZIP-архивов, извлечение изображений для дальнейшего анализа.

- Image — определение размеров изображения.

- Определение характеристик изображения: размер, глубина цвета, цветовой режим, тип сжатия.

- Управление интерфейсом: динамическое обновление таблицы, валидация файлов.

- DataTables — библиотека для интерактивной таблицы с возможностью сортировки, фильтрации и пагинации данных.

4. GitHub Pages — для размещения приложения:

- Публикация HTML, CSS и JS файлов, что позволяет сделать приложение доступным для пользователей.

# Функционал приложения:

1. Загрузка изображений:

- Пользователь может загружать как отдельные изображения (JPEG, PNG, GIF, BMP, TIFF), так и ZIP-архивы, содержащие изображения.

- Приложение автоматически извлекает изображения из архива и обрабатывает их.

2. Анализ изображений:

- Определение формата изображения, его разрешения (размеров в пикселях), глубины цвета, цветового режима и типа сжатия.

- Все данные отображаются в интерактивной таблице с возможностью сортировки по любому из столбцов.

3. Очистка таблицы:

- Пользователь может очистить таблицу с помощью кнопки "Очистить таблицу", что сбрасывает все данные без необходимости перезагружать страницу.

4. Работа с ZIP-архивами:

- Если загружен архив, приложение автоматически распознает изображения внутри архива и обрабатывает их так же, как и одиночные файлы.

# Теоретические аспекты:

1. Форматы изображений:

- Приложение поддерживает форматы JPEG, PNG, GIF, BMP и TIFF. Каждый из этих форматов имеет свои особенности в плане глубины цвета, сжатия и цветовых режимов.

- JPEG — формат с потерей качества, обычно 24-битный RGB, используется для фотографий.

- PNG — поддерживает как 24-битный RGB, так и 32-битный (с альфа-каналом для прозрачности), использует сжатие без потерь.

- GIF — 8-битный индексированный цвет, использует LZW сжатие, поддерживает анимации.

- BMP — не использует сжатие, поддерживает глубины цвета от 8 до 32 бит.

- TIFF — формат для сохранения изображений высокого качества, поддерживает различные методы сжатия и цветовые модели (включая многоканальные).

2. Цветовая глубина:

- Глубина цвета определяется количеством бит, которые используются для представления одного пикселя изображения.

- 8 бит — монохромные или индексированные изображения.

- 24 бит — изображение с тремя цветами (RGB), по 8 бит на каждый.

- 32 бит — изображение с четырьмя каналами (RGBA), включая альфа-канал для прозрачности.

3. Цветовые режимы:

- Определяют способ кодирования цветов в изображении:

- RGB — основной режим для большинства изображений.

- RGBA — RGB с альфа-каналом для прозрачности.

- Indexed — используется в GIF для сжатия изображения за счет ограничения количества цветов.

4. Сжатие:

- JPEG использует сжатие с потерями, что позволяет значительно уменьшить размер файлов, но снижает качество изображения.

- PNG и GIF используют сжатие без потерь (Deflate и LZW соответственно), что сохраняет исходное качество.

- BMP обычно не использует сжатие.

- TIFF поддерживает несколько типов сжатия, включая LZW и JPEG.

# Структура файлов:

1. HTML:

- Содержит разметку веб-страницы: форму для загрузки файлов, кнопки для взаимодействия с пользователем и таблицу для отображения информации о загруженных изображениях.

2. CSS:

- Оформление интерфейса страницы, включая таблицу, кнопки и поля для загрузки файлов.

- Подключение стилей DataTables для улучшенного отображения данных.

3. JavaScript:

- Основная логика приложения: обработка загружаемых изображений, чтение их метаданных, работа с ZIP-архивами, динамическое обновление данных в таблице и обработка событий интерфейса.

- Обработка изображений через FileReader и JSZip для работы с архивами.

- Применение библиотек DataTables для создания интерактивной таблицы с возможностью сортировки и фильтрации данных.

4. GitHub Pages:

- Использование GitHub Pages для публикации веб-приложения и предоставления его в общий доступ через веб-браузер.

# Вывод:

Приложение позволяет пользователям удобно загружать и анализировать изображения различных форматов, а также архивы, содержащие изображения. Оно автоматически выводит информацию о файлах (размеры, глубина цвета, цветовой режим и тип сжатия) в формате таблицы, которая поддерживает сортировку и фильтрацию данных. Приложение может обрабатывать как одиночные файлы, так и ZIP-архивы, что делает его полезным для массового анализа изображений.