

УДК 004.054

РАЗРАБОТКА И ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЯ ДЛЯ УПРАВЛЕНИЯ МЕТАДАННЫМИ

Глушакова Д. А., студент гр. ПМР-221, III курс, Прохоренко А. В.,
студент гр. ПМР-221, III курс

Научный руководитель: Зайченко Е.А., старший преподаватель кафедры
«Программное обеспечение информационных технологий»

Белорусско-Российский университет

Метаданные автоматически приписываются каждому файлу, помогают систематизировать и искать данные. Простая аналогия: чтобы получить в библиотеке конкретное произведение, необходимо сообщить библиотекарю автора и название. Непосредственно текст произведения – это сами данные, а метаданные, т.е. «данные о данных» – это имя автора, название произведения, количество страниц в книге, издательство и т.п. Аналогично, посты в блогах имеют свои метаданные: заголовок, автор, дата публикации, теги и любая другая информация о публикации. Метаданные помогают установить лицензионные ограничения на распространение информации, указывая на автора контента [1]. Встроенные в веб-сайты метаданные включают в себя описание сайта, ключевые слова, мета-теги – все, что имеет значение для поиска. Они идентифицируют контент и дают специальную характеристику, которая отличает его от другого контента. Идентификация метаданных включает множество технических элементов, таких как уникальный идентификатор ресурса (или URL), справочный номер файла или имя файла, заголовок и автора [2].

Метаданные в аудио, видео и фотофайлах могут содержать скрытую информацию, которую преступники используют в своих целях. Например, в фотографиях часто сохраняются GPS-координаты, по которым можно определить, где и когда был сделан снимок, что позволяет злоумышленникам отследить местоположение человека [3]. Видеофайлы могут содержать данные о камере, дате съемки и даже информации о владельце устройства. Аудиофайлы нередко хранят сведения о кодеке, авторе записи и времени создания. Анализируя эти данные, преступники могут собирать цифровые следы, вычислять привычки и маршруты жертв, подделывать мультимедийные файлы или использовать их для шантажа. Именно поэтому перед публикацией или отправкой медиафайлов важно проверять и удалять метаданные, чтобы не оставлять лишней информации, которая может быть использована во вред.

Столь популярные сегодня мессенджеры нередко оказываются под прицелом, как исследователей безопасности, так и злоумышленников. Вот одно из недавних сообщений о том, что они раскрывают персональные данные пользователей. У каждого из четырех популярных мессенджеров (Telegram, WhatsApp, Viber, Messenger) есть функция передачи вложения в виде файла со всеми сопутствующими данными. Передача фото и видео через функцию

«Камера» или с помощью галереи нередко убирает все метаданные, однако это вызывает потерю качества изображения.

Злоумышленники, обладая доступом к таким данным, могут проанализировать активность пользователя, определить его маршруты передвижения, время посещения определённых мест и даже установить личность по встроенной информации. Это создаёт серьёзные риски для конфиденциальности и безопасности. Чтобы минимизировать возможные угрозы, перед публикацией изображений рекомендуется удалять или изменять метаданные с помощью специальных инструментов или встроенных настроек камер и фоторедакторов. Это поможет избежать нежелательного раскрытия персональной информации и защитить себя от потенциальных угроз.

Создание приложения для управления метаданными актуально из-за угроз, связанных с мошенниками и киберпреступниками. Например, анализ метаданных позволяет преступникам отслеживать перемещения жертвы или подделывать доказательства. Кроме того, утечка таких данных в корпоративной среде может привести к кибератакам. Разрабатываемое приложение позволит пользователям легко удалять или изменять метаданные, минимизируя риски и повышая уровень цифровой безопасности.

Для создания приложения на языке программирования Python будем использовать основные библиотеки для работы с метаданными в форматах фото, видео, аудио. Будем использовать основные библиотеки для работы с метаданными в форматах фото, видео и аудио.

Поскольку мы берем ограниченный ряд файлов, с которыми наша программа будет взаимодействовать: аудио, видео, и изображения, можно ограничиться одной библиотекой, которая позволяет подробно рассматривать все теги для данных типов, в данном случае ruexiftool [4].

Утилита ruexiftool, являясь обёрткой для ExifTool, позволяет извлекать большое количество тегов, которые могут быть использованы злоумышленниками для составления цифрового следа пользователей. При анализе изображения можно обнаружить данные, позволяющие установить точное время и место его создания. Метаданные часто содержат GPS-координаты, что делает возможным отслеживание местоположения человека. Кроме того, в файле могут сохраняться сведения о модели камеры или смартфона, с которого была сделана фотография, а также параметры съёмки, которые могут указывать на условия, в которых производилась фиксация кадра. Помимо технических характеристик, метаданные могут включать в себя информацию о владельце изображения, если она была предварительно добавлена. Например, в некоторых случаях фиксируется имя автора снимка, описание сцены или даже контактные данные. Также могут присутствовать сведения об использованных программах для редактирования, что позволяет определить, подвергалось ли изображение обработке [5].

Для создания графического интерфейса будем использовать библиотеку PyQt.

Программа предоставляет пользователю возможность просматривать, редактировать и очищать метаданные различных файлов, включая изображения, аудио и видео. Вид главного окна программы на рисунке 1.

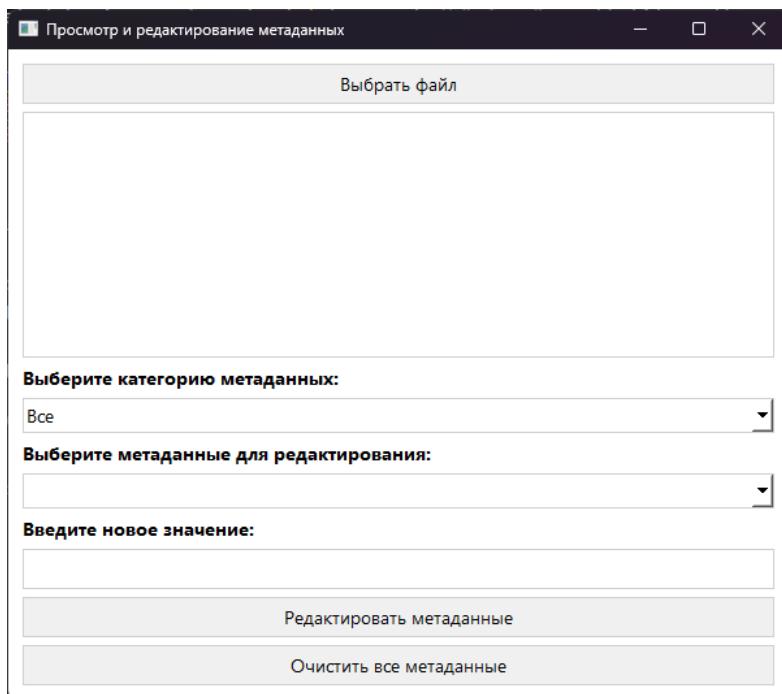


Рисунок 1 – Вид главного окна программы

Пользователь может выбрать файл, метаданные которого он хочет просмотреть. Программа поддерживает различные форматы, включая изображения (JPEG, PNG, TIFF и другие), аудио (MP3, WAV, FLAC и т.д.) и видео (MP4, AVI, MOV).

После выбора файла, программа автоматически извлекает метаданные с использованием инструмента ExifTool и отображает их в удобном текстовом формате, подробнее можно увидеть на рисунке 2.

Программа позволяет фильтровать метаданные по категориям, таким как «Все», «Изображение», «Файл», «Дополнительно» и «Аудио», что помогает быстро находить нужную информацию. Помимо этого, предоставляется возможность выбрать конкретное поле метаданных и отредактировать его значение. Пользователь может ввести новое значение, и программа обновит метаданные файла в реальном времени.

Для обеспечения конфиденциальности и безопасности программа также предлагает функцию очистки всех метаданных из файла. Это полезно, например, при подготовке изображений или аудиофайлов для публикации в интернете, чтобы избежать утечек личных данных, таких как GPS-координаты или информация о дате и времени съемки. Таким образом, программа предоставляет простое и эффективное средство для работы с метаданными файлов, позволяя пользователям как просматривать, так и редактировать или очищать их по мере необходимости.

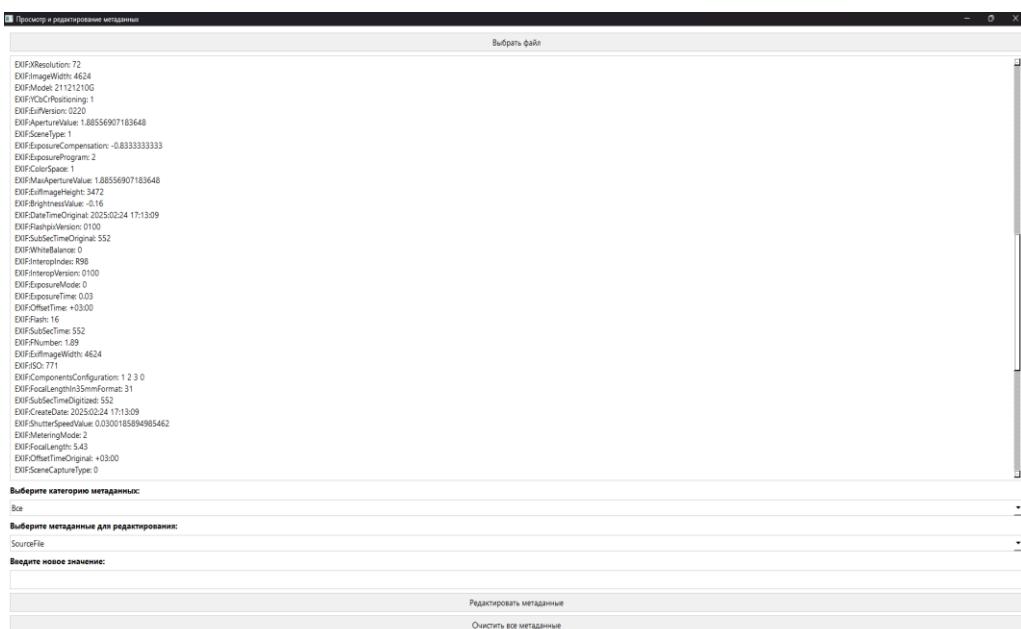


Рисунок 3 – Пример извлечения метаданных с помощью разработанной программы

Для обеспечения надежности и функциональности приложения была проведена серия тестов, направленных на проверку его ключевых возможностей. Был разработан набор из 5 модульных тестов, которые проверяют корректность работы основных функций, таких как просмотр, редактирование и удаление метаданных из различных типов файлов (изображений, аудио и видео). Тесты полностью охватывают функционал программы.

Для проведения тестов использовалась техника мок-объектов, что позволяет эмулировать взаимодействие с внешними зависимостями, в данном случае с инструментом ExifTool, отвечающим за извлечение и изменение метаданных файлов. Этот подход дает возможность тестировать логику работы программы без фактической манипуляции с реальными файлами, что значительно упрощает процесс тестирования и повышает его эффективность.

Рассмотрим подробно первый тест – проверку корректности обработки изображения.

Цель теста: проверить, что программа корректно обрабатывает ситуации, когда данные метаданных повреждены или отсутствуют.

Что тестируется: важно убедиться, что программа правильно обрабатывает ситуации, когда инструмент ExifTool возвращает пустой или некорректный JSON. В случае ошибок программа должна адекватно реагировать, например, вывести сообщение об ошибке, а не завершить работу аварийно.

Как проводился тест: используется мок-объект, который возвращает пустой JSON или некорректные данные. Проверяется, что программа выбрасывает исключение JSONDecodeError в случае поврежденных метаданных и корректно отображает ошибку в интерфейсе.

Значение теста: этот тест нужен для гарантии, что программа будет стабильной и устойчивой к ошибкам во внешнем инструменте или в метаданных.

Без такого теста программа может аварийно завершить работу при получении некорректных данных.

Код выполнения теста представлен на рисунке 3.

```
@patch("test3.exiftool.ExifTool")
def test_display_image_metadata(self, mock_exiftool):
    """Проверяем, что программа корректно обрабатывает изображения"""
    mock_et_instance = MagicMock()
    mock_exiftool.return_value.__enter__.return_value = mock_et_instance

    # Эмуляция нормального результата (корректный JSON)
    mock_et_instance.execute.return_value = json.dumps([
        {
            "File:Name": "test.jpg",
            "EXIF:DateTime": "2023:01:01 12:00:00",
            "Image:Width": "1920",
            "Image:Height": "1080"
        }
    ]).encode()

    self.viewer.file_path = "test.jpg"
    self.viewer.display_image_metadata()

    self.assertIn("File:Name", self.viewer.metadata["Все"])
    self.assertIn("EXIF:DateTime", self.viewer.metadata["Изображение"])
    self.assertEqual(self.viewer.metadata["Изображение"]["Image:Width"], "1920")
```

Рисунок 3 – Проверка корректной обработки изображения
(test_display_image_metadata)

Были проведены также следующие тесты.

Проверка выбора файла. Тест проверяет, правильно ли сохраняется путь к выбранному файлу. Это важный шаг, поскольку программа не может продолжать свою работу (например, извлечение метаданных или их редактирование) без правильного указания файла. В teste эмулируется процесс выбора файла. Через mock-объект имитируется вызов инструмента для извлечения метаданных. После выбора файла (в teste путем присваивания значения пути) проверяется, что путь к файлу сохранен верно.

Проверка обработки некорректных метаданных. Важно убедиться, что программа правильно обрабатывает ситуации, когда инструмент ExifTool возвращает пустой или некорректный JSON. В случае ошибок программа должна адекватно реагировать, например, вывести сообщение об ошибке, а не завершить работу аварийно. Используется mock-объект, который возвращает пустой JSON или некорректные данные. Проверяется, что программа выбрасывает исключение JSONDecodeError в случае поврежденных метаданных и корректно отображает ошибку в интерфейсе. Этот тест нужен для гарантии, что программа будет стабильной и устойчивой к ошибкам во внешнем инструменте или в метаданных. Без такого теста программа может аварийно завершить работу при получении некорректных данных.

Проверка редактирования метаданных. Важным аспектом является то, что изменение метаданных не должно нарушать структуру данных и должно быть

успешно записано в файл. Для теста имитируется ввод нового значения для одного из метаданных, например, даты съемки. Проверяется, что инструмент ExifTool был вызван с правильными параметрами (включая имя метаданных и новое значение). Этот тест проверяет ключевую функциональность программы — редактирование метаданных. Он важен для гарантии, что изменения в файле будут корректно обработаны и записаны.

Проверка удаления всех метаданных. Проверяется, что программа может корректно очистить все метаданные из выбранного файла, а затем обновить отображение данных (к примеру, после очистки файл больше не содержит метаданных). Это также тестирует вызов инструмента для очистки метаданных через команду -all=.

В процессе разработки и тестирования приложения по управлению метаданными аудио, видео и фото файлов были проверены ключевые функции программы, такие как выбор файла, извлечение и редактирование метаданных, а также очистка всех метаданных. Каждый тест был направлен на проверку определенного аспекта работы программы, что гарантирует ее стабильность и надежность. Эти тесты охватывают основные сценарии использования программы и обеспечивают уверенность в корректности работы системы.

Список литературы

- 1 Библиотека для открытой науки. Метаданные / Библиотека для открытой науки. – Текст: электронный // Библиотека для открытой науки : [сайт]. – Москва, 2022. – URL: <https://lib-os.ru/issledovatelyam/rukovodstvo-po-upravleniyu-dannymi-issledovanij/metadannye/> (дата обращения: 01.03.2025).
- 2 Элементы управления метаданными / ТАВИЗЕР. – Текст: электронный // ТАВИЗЕР: [сайт]. – Москва, 2025. – URL: https://www.tadviser.ru/index.php/Статья:Элементы_управления_метаданными (дата обращения: 02.03.2025).
- 3 Метаданные в компьютерных системах. – Текст: электронный // ИПР РАН. – 2017. – № 4. – С. 28-46. – URL: http://www.iprras.ru/old_site/articles/kogalov13-03.pdf (дата обращения: 04.03.2025).
- 4 Python - Задача в получении метаданных списка файлов из установленной директории с помощью сторонней утилиты (ExifTool) / Stack Overflow на русском. – Текст: электронный // Stack Overflow: [сайт]. – URL: <https://ru.stackoverflow.com/questions/615939/Задача-в-получении-метаданных-списка-файлов-из-установленной-директории-с-помощью> (дата обращения: 05.03.2025).
- 5 ExifTool: читаем и редактируем метаданные файлов / Астапчик М. – Текст: электронный // Домашняя страница ExifTool: [сайт]. – URL: <https://astapm.github.io/soft/exiftool.html> (дата обращения: 05.03.2025).