

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЧОРНОМОРСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ПЕТРА МОГИЛИ

**СТОВМАНЕНКО ВЛАДИСЛАВ ОЛЕГОВИЧ**

УДК 004.4

**Інформаційна система для роботи з тегами ID3v2 для mp3 файлів.  
Декодування двійкових даних, використання алгоритмів  
системного аналізу.**

Напрямок підготовки 6.050101 – «Комп'ютерні науки»

ДР - 403.1510321

Автореферат

дипломної роботи на здобуття освітньої кваліфікації

«Бакалавр комп'ютерних наук»

Миколаїв – 2019

Дипломна робота є рукопис.

Робота виконана в Чорноморському національному університеті імені Петра Могили Міністерства освіти і науки України на кафедрі інтелектуальних інформаційних систем.

Науковий керівник: кандидат технічних наук, доцент  
Давиденко Євген Олександрович.

Рецензент: д.т.н., проф., завідувач кафедри  
інтелектуальних інформаційних систем  
Кондратенко Юрій Пантелійович.

Захист відбудеться 26 червня 2019 р. о 9<sup>30</sup> год. на засіданні екзаменаційної комісії (ауд. 2-403) у Чорноморському національному університеті імені Петра Могили за адресою: 54003, м. Миколаїв, вул. 68-ми Десантників, 10.

З дипломною роботою можна ознайомитися в бібліотеці Чорноморського національного університету імені Петра Могили за адресою: 54003, м. Миколаїв, вул. 68-ми Десантників, 10.

Автореферат представлений «18» червня 2019 р.

Секретар  
екзаменаційної комісії,

ст.викл.

І.О. Кандиба

## ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Робота полягала у формуванні збірки, яка б надавала інкапсулювала особливості роботи з метаданими аудіоданих для формату MP3. При цьому необхідно було спроектувати класи, які б репрезентували наявні сутності, які можна було виділити із стандарту. Технології для розробки можна було обирати самостійно. В результаті виконання було обгрунтовано актуальність теми, обрано стек технологій, розглянуто та обрано формат аудіоданих для метаданих якого буде створюватися збірка, спроектовано та розроблено бібліотеку, яка дозволяла працювати із метаданими, надаючи програмний інтерфейс для роботи з ними.

Окрім того було реалізовано алгоритм kmeans для роботи із зображенням-обложкою, яке зберігається у метаданих. Завдяки цьому можна певним чином стилізувати інтерфейс користувача під час демонстрації інформації про трек. Використано можливості багатопотокового програмування для підвищення ефективності виконання алгоритму.

Було отримано додатковий досвід написання великих звітних робіт, які пояснюють характер наукового дослідження. Опановано поняття “пояснювальна записка”, “унікальність тексту”, “автореферат” та інші пов’язані поняття.

**Актуальність роботи.** В реальному світі доволі просто описати природу тих чи інших даних. Оглянувши їх, можна зробити висновок про їх природу та походження. Проте таке завдання ускладнюється у разі збільшення обсягів. Очевидним рішення є маркування даних у певний спосіб. Найбільш очевидним способом подіного маркування є присвоєння назви та авторства. В цьому разі на автора покладається вимога створення додаткових даних. Історично такі дані механічно постачалися разом із самими даними. На книгах були палітурки з назвою, на платівках були конверти з назвою, на картинах були відповідні написи. Такий спосіб, не зважаючи на свою очевидну простоту мав один

недолік: існувала можливість роз'єднання контейнеру з даними та додаткової інформації. Для відновлення оригінальних зв'язків треба було використовувати знання експертів і витратити час. Для мінімізації впливу цього недоліку постачальники контейнерів з даними (мається на увазі фізичне сховище даних у будь-якому вигляді, наприклад книга чи полотно) експериментували зі структурою контейнерів, шукаючи можливості вбудувати інформацію про дані в сам контейнер без модифікації самих даних. Так у книгах з'явилася інформація про видавця та автора, на платівках етикетка з коротким описом, а на катушках із відеоплівкою з'явилися наліпки з маркуванням.

Після появи можливостей для збереження великих обсягів даних на електронних пристроях запам'ятовування постало питання про відповідне маркування даних. Спочатку для цього намагалися використовувати ім'я файлу з даними, але швидко зрозуміли, що в ньому неможливо у повній мірі ідентифікувати дані. Від підходу із розділеним зберіганням даних також довелося швидко відмовитися. Під час маніпуляцій над файлами, деякі з них легко могли бути помилково видалені чи змінені. Тому розробники відразу пішли шляхом вбудовування даних безпосередньо у файли.

Навіть у наш час, коли багато людей віддають перевагу хмарним сховищам та сервісам і не утримують музичні бібліотеки в домашніх умовах, є ще багато користувачів, які воліють зберігати музику на переносимому і постійно доступному пристрої, вибираючи тільки те, що потрібно в тому чи іншому випадку. В той же час, наявність відкритого програмного інтерфейсу в певному хмарному сервісі дозволить не обмежуватися локальною бібліотекою і проводити маніпуляції з файлами так, немов вони знаходяться на локальному пристрої. Знаходження потрібного файлу, конвертація у потрібний формат може зайняти багато часу, якщо файли розміщені хаотично. А у разі, коли метадані недоступні, таке завдання стає практично невиконуваним.

Для вирішення проблеми управління локальними музичними колекціями можна використовувати редактор метаданих, який додатков надає інструментарій для

сортування і фільтрації. Простого перегляду назви пісні та імені автора недостатньо, тому велика кількість аудіоплеєрів не підходить для вирішення поставленої задачі. Наявні ж редактори часто не мають потрібного функціоналу. Здебільшого це програми, які дозволяють сформувати список файлів і оглядати інформацію про них. Проте тільки перегляду цих даних недостатньо для забезпечення належної зручності роботи з ними. Це все іще вимагає від користувача вручну виконувати кожну бажану операцію із файлом. Чим більшими є обсяги даних, які аналізуються, тим доречніше буде впровадити систему автоматизації, яка зможе відсотрувати файли відповідно до певного критерію, здійснити їх обробку чи перетворення або просто перемістити їх у потрібне місце.

**Мета бакалаврської дипломної роботи.** Метою роботи було продемонструвати отримані за час навчання в університеті навички та вміння, отримати досвід створення наукової роботи і водночас розробки складних програмних продуктів для вирішення певної прикладної задачі.

**Для досягнення поставленої мети необхідно вирішити такі задачі:** спроектувати та реалізувати збірку для роботи з метаданими ID3v2 використовуючи довільний стек технологій.

**Об'єкт дослідження.** Об'єктом дослідження є дані про дані або метадані аудіофайлів.

**Предмет дослідження.** Предметом дослідження є стандарти аудіоформатів та стандарти метаданих, сумісні з відповідними стандартами.

Під час виконання роботи було написано декілька статей, які було роздруковано у різних наукових виданнях.

**Практичне значення отриманих результатів.** Реалізовану збірку можна використовувати у будь-яких застосунках (веб-сайти, сервери, застосунки для робочого столу з графічним інтерфейсом користувача, консольні застосунки, мобільні застосунки). Приклади реалізованого WPF застосунку:



Рис. 1.1. Робота алгоритму підбору палітри кольорів

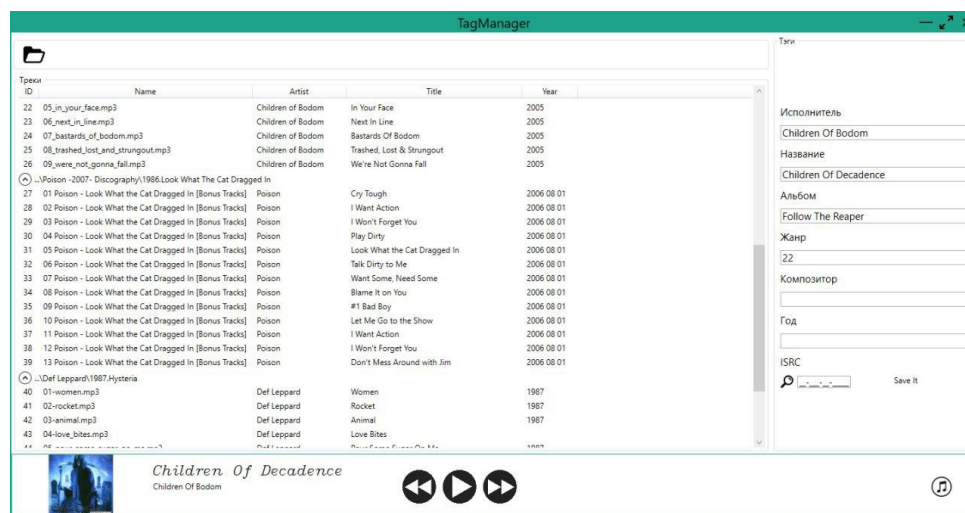


Рис. 1.2 Метадані різних аудіофайлів згруповані відповідно до місцезнаходження

**Публікації та апробація.** Дану роботу було апробовано на двох конференціях та представлено статтю у фаховому виданні.

- Давиденко Є. О., Стовманенко В. О., Григор'єв Д. О. Робота з метаданими аудіофайлів. Могилянські читання – 2018 : досвід та тенденції розвитку суспільства в Україні: глобальний, національний та регіональний аспекти : XXI Всеукр. наук.-метод. конф. : тези доповідей Комп'ютерні

- науки. Технічні науки, Миколаїв, 12-17 листоп. 2018 р. / ЧНУ ім. Петра Могили. – Миколаїв : Вид-во ЧНУ ім. Петра Могили, 2018. – С. 127-128.
- Стовманенко В. О., Григор'єв Д. О., Давиденко Є. О. Використання алгоритмів системного аналізу для роботи із медіа. Комп'ютерна інженерія і кібербезпека : досягнення та інновації : матеріали Всеукр. наук.-практ. конф. здобувачів вищої освіти й молодих учених (м. Кропивницький, 27-29 листоп. 2018 р.). – Кропивницький : ЦНТУ, 2018. – С. 114-115.
  - Стовманенко В. О., Григор'єв Д. О., Боровльова С. Ю., Давиденко Є. О. Робота з метаданими mp3-файлів. Наукові праці : наук. журн. – Вип. 308. Т. 320. Комп'ютерні технології. – Миколаїв : Вид-во ЧНУ ім. П. Могили, 2018. – С. 86-90.

**Структура дипломної роботи.** Дипломна робота складається із вступу, трьох розділів, висновків, спеціальної частини та додатків. Загальний обсяг роботи складає шістдесят чотири сторінки, вісімнадцяти рисунків, однієї таблиці та двадцятьох п'ятьох посилань на літературні джерела.

## ОСНОВНИЙ ЗМІСТ РОБОТИ

**У вступі** описано дипломний проект та базові характеристики наукової роботи (мета, об'єкт та предмет).

**У першому розділі (Актуальність проекту)** доводиться актуальність теми, яку обрано для дипломного проекту.

Для вирішення проблеми управління локальними музичними колекціями можна використовувати редактор метаданих, який додатков надає інструментарій для сортування і фільтрації. Простого перегляду назви пісні та імені автора недостатньо, тому велика кількість аудіоплеєрів не підходить для вирішення поставленої задачі. Наявні ж редактори часто не мають потрібного функціоналу. Здебільшого це програми, які дозволяють сформувати список файлів і оглядати інформацію про них. Проте тільки перегляду цих даних недостатньо для забезпечення належної зручності роботи з ними. Це все іще вимагає від користувача вручну виконувати кожну бажану операцію із файлом. Чим більшими є обсяги даних, які аналізуються, тим доречніше буде впровадити систему автоматизації, яка зможе відсотрувати файли відповідно до певного критерію, здійснити їх обробку чи перетворення або просто перемістити їх у потрібне місце.

**У другому розділі (Теоретична частина)** наведено дані про структуру метаданих для різноманітних аудіоформатів та можливості для їх застосування.

Підхід із вбудовуванням метаданих у аудіофайл можна застосовувати не для всіх форматів даних. Його неможливо застосувати через особливості механізму декодування даних. Не всі стандарти надають можливість зберігати дані різного роду поруч. Наприклад кодек формату G.722 не здатний розрізняти блоки з аудіоданими та блоки із іншими даними. В той час як формати на кшталт MP3 не проектувалися з думкою про метадані, але в них використовується маркер синхронізації, що дозволяє відрізняти аудіодані від будь-яких інших даних у файлі.



У третьому розділі (Практична частина) надано опис засобів, які було використано для реалізації, створено вимоги до збірки та описано її структуру та функцій. Описано підходи, які було використано для створення бібліотеки класів, які відповідали створеній структурі, шаблони багатопотокового програмування та алгоритми кластерного аналізу, які було використано.

Для роботи з метаданими варто завантажити відповідну ділянку файлу в пам'ять, декодувати її і, на вимогу, віддавати декодовані шматки даних. Блок-схема роботи представлена нижче.

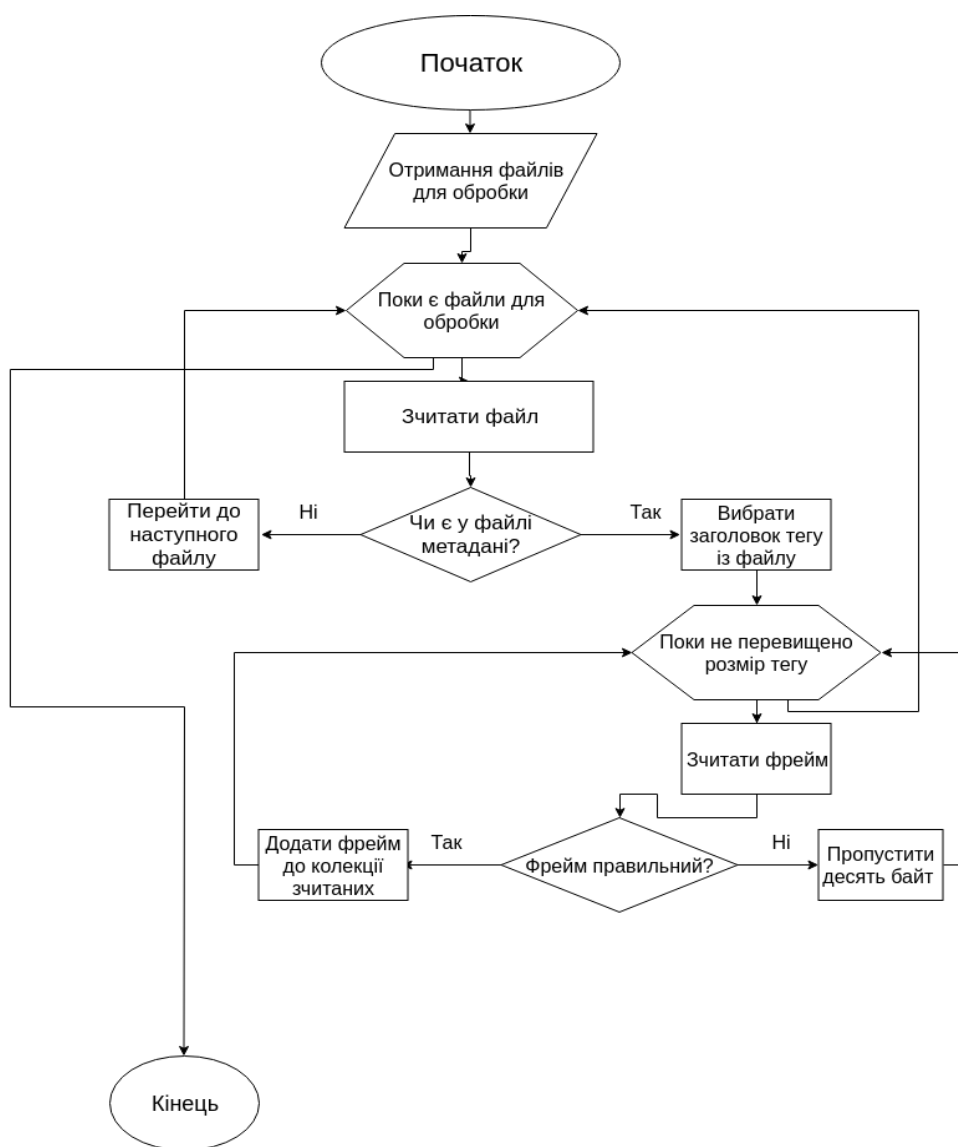


Рис. 3.1. Блок-схема роботи алгоритму зчитування метаданих ID3.

Для ефективної роботи було створено класи, що представляють основні сутності відповідно до стандарту (тег, фрейм, їх заголовки та інше). UML діаграма представлена нижче.

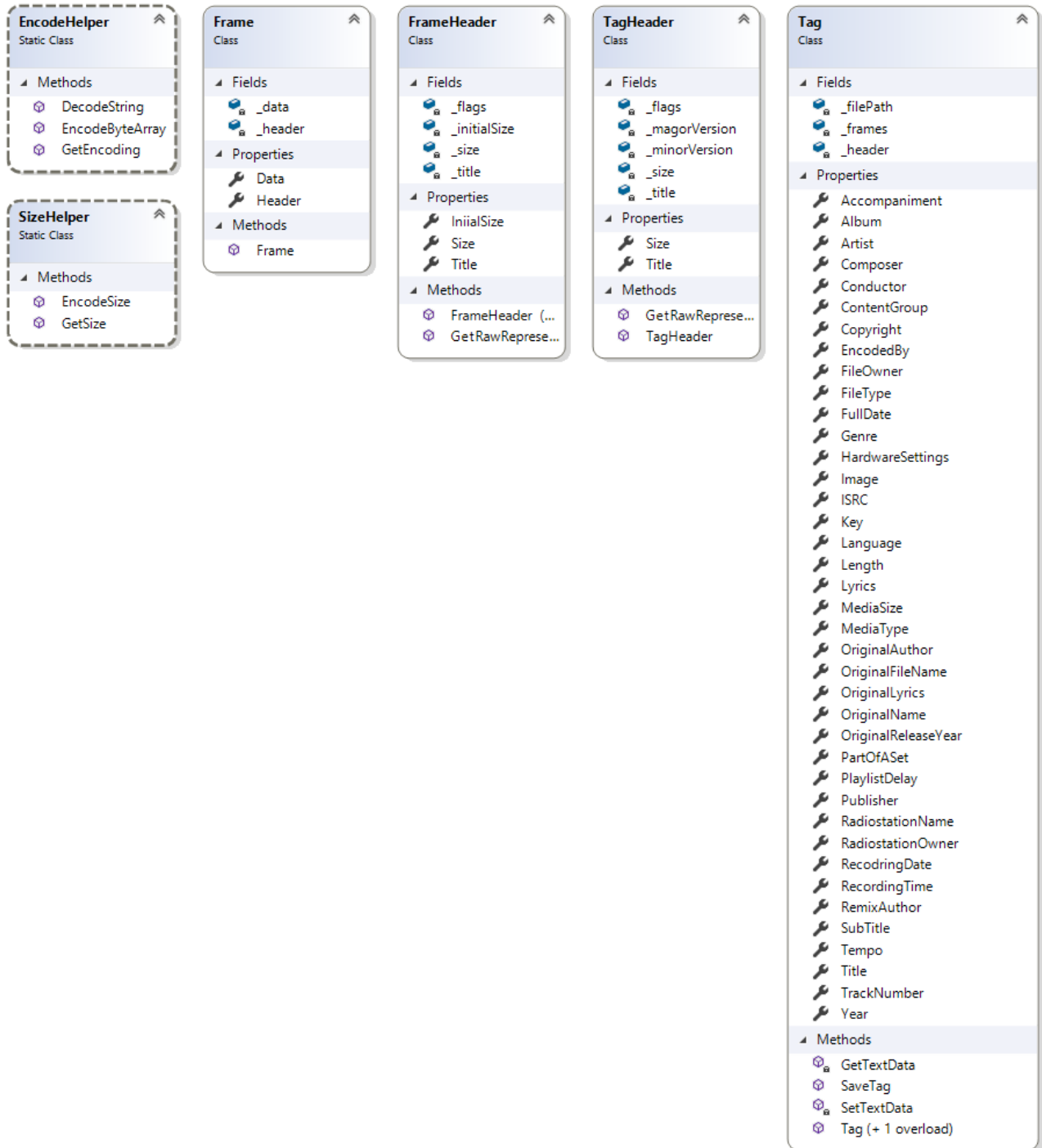


Рис. 3.2. UNL-діаграма реалізованих сутностей, що стосуються роботи з метаданими ID3.

Для реалізації було обрано платформу .Net, що забезпечує переносимість коду (збірку можна виконувати на платформах для яких існує реалізація Common

Language Runtime, на момент написання це різноманітні програмні платформи (MS Windows, Mac OS, сімейство операційних систем GNU/Linux, сімейство систем BSD, Android та iOS), а також апаратні платформи (i386 та похідні, x64\_86, IA-32, ARM, SPARC та інші) на рівні скомпільованої збірки. Як і технологія Java, середовище розробки .NET створює байт-код, призначений для виконання віртуальною машиною. Вхідна мова цієї машини в .NET називається CIL (Common Intermediate Language), також відома як MSIL (Microsoft Intermediate Language), або просто IL. Перед запуском збірки в середовищі виконання (CLR) байт-код перетворюється вбудованим в середовище JIT-компілятором (just in time, компіляція на льоту) в машинні коди цільового процесора.

Для знаходження палітри кольорів зображення було використано алгоритм kmeans. Кластеризація k-means є методом векторного квантування, який є популярним засобом кластерного аналізу і часто використовується для інтелектуального аналізу даних. k-means кластеризація спрямована на розбиття  $N$  спостережень на  $K$  кластерів, в яких кожне спостереження належить до кластера з найближчим середнім, що є його прототипом. Таким чином увесь простір даних розбивається на комірки Вороного.

Для зчитування даних із файлу використовується FileStream, клас, що реалізує базові можливості побайтового зчитування та більш специфічні (відкриття у певному режимі, блокування, тощо). Для запису великих обсягів даних (у випадку збереження даних після модифікації багатьох файлів) варто використовувати асинхронні методи, які надає клас FileStream. Вони будуть виконання без блокування основного потоку. Розгляд цієї продуктивності особливо у додатках, де робота потоку, яка займає багато часу, може заблокувати потік інтерфейсу користувача та зробити програму такою, що не працює. FileStream буферизує вхід і вихід для кращої продуктивності.

Окрім файлового потоку, для читання можна використовувати будь-який інший тип. Таким чином можна отримувати доступ до віддаленого файлу, що

знаходиться на сервері, якщо це дозволено відповідними налаштуваннями безпеки. Для цього використовується клас `NetworkStream`.

Для роботи з текстом у новітніх застосунках використовується кодунікод `Unicode` (.Net рядки за замовчуванням використовують `UTF8`) та його різновиди. Стандарт `ID3` передбачає збереження тексту в деяких інших кодуваннях, зокрема в `ASCII`. Більшість текстових полів та полів посилань містить мітку з інформацією про кодування даних. У деяких випадках це дозволяє зменшити розмір даних, які зберігаються.

Для виконання кластерного аналізу із високим ступенем імовірності та малими затримками треба виконати розрахунок Евклідової відстані для зображення не менше ніж 256 на 256 пікселів. Така кількість операцій може заблокувати основний потік застосунку і унеможливити функціонування інтерфейсу користувача на деякий час. Для того аби уникнути цього та прискрити обробку даних можна скористатися перевагами багатопроцесорних систем. .Net підтримує роботу з потоками та багатопроцесорними системами. Для розуміння багатопотокового виконання варто усвідомлювати роботу однопотокового процесора та механізми, що використовуються для створення уявної багатопотоковості. Процесор надзвичайно “швидкий” пристрій і він встигає виконувати завдання швидше ніж результат його виконання можна буде відправити користувачеві, тому увесь процесорний час розділяється та планується між процесами. Процес — це екземпляр програми, що виконується на електронному обчислювальному пристрої зі своєю ділянкою оперативної пам’яті в якій містяться дані та програмні інструкції. Планувальник операційної системи розподіляє час між усіма процесами відподно до пріоритету та інших політик. В результаті подібного виконання процесор може виконувати ті шматки коду, які потребують негайної уваги. При тому самі по собі операції є не складними, особливо у випадках, коли більша частина їх обробки виконується периферійним пристроєм (наприклад перерахунок графіки на відеоадаптері). Втакому випадку процесом має перервати виконання поточного процесу,

виконати процедуру завантаження даних на відеоадаптер і повернутися до виконання. Фактично програми виконувалися конвеєрно, але дуже швидко. Справжня багатопотоковість з'явилася разом із процесорами із додатковими віртуальними або реальними ядрами. В таких системах в одну одиницю часу могло виконуватися два набори інструкцій на кожному із ядер. Саме в таких системах є сенс використовувати багатопотокові алгоритми та підходи.

**У спеціальній частині (Охорона праці)** розглядалися умови праці у великих ІТ-офісах, документи, які регламентують ці умови та необхідні правила техніки безпеки під час роботи з електронними обчислювальними приладами.

## ВИСНОВКИ

Під час виконання дипломної роботи було спроектовано та розроблено збірку .Net для роботи із метаданими аудіофайлів у форматі mp3 відповідно до стандарту ID3. Було проведено аналіз різних форматів збереження аудіоданих та стандартів збереження метаданих. Описано розвиток цих форматів та їх основні особливості. Опрацьовано стандарт збереження метаданих ID3. Отримано навички роботи із діаграмами функціонального проектування IDEF. Використано шаблони проектування такі як (“фабрика”, “стратегія”, “хранитель” та ін) для реалізації збірки. Описано засоби для роботи із файловою системою платформи .Net, класи, що виконують кодування та декодування двійкових даних, а також підходи до паралельного програмування і засоби синхронізації потоків. Адаптовано алгоритм кластерного аналізу kmeans для знаходження палітри кольорів довільного зображення. Для збільшення швидкості роботи алгоритму kmeans використано шаблони паралельного та асинхронного програмування. Набуто досвіду роботи із системами контролю версії (git) та розподіленими методологіями розробки. Код ліцензовано відповідно до Загальної громадської ліценсії третьої версії (GPL v3). Після впровадження все більшої кількості сервісів (YouTube Music, iTunes, Spotify,

Digger та ін.), які надають постійний доступ до потокового аудіо високої якості потреба у збереженні колекцій у форматі mp3 може поступово зменшуватися. Сервери інкапсулюють роботу з файлами у яких містяться аудіодані та відповідні метадані. Кінцевий користувач отримує доступ безпосередньо до інформації про трек. Потужностей переносних пристроїв вистачає для того або отримувати доступ до таких сервісів або відтворювати аудіофайли закодовані у одному з форматів, що дозволяють збереження “без втрат”. До того ж завдяки розвитку алгоритмів кодування даних з’являються інші алгоритми, які дозволяють стискати дані із більшим коефіцієнтом стиснення і мати кращу якість. Це все сприяє зменшенню популярності формату mp3 як для збереження приватних колекцій, так і для передачі даних через мережу. Не зважаючи на це, реалізовану збірку все ще можна використовувати із наявними колекціями створивши мінімальний консольний застосунок або використавши збірку в застосунку робочого столу Windows, перемосимого застосунку Avalonia, проекту ASP.NET або веб-сайті з Blazor та Web Assembly.

## АНОТАЦІЇ

**Стовманенко Владислав Олегович. Інформаційна система для роботи з тегами ID3v2 для mp3 файлів. Декодування двійкових даних.** — на правах рукопису.

В роботі було розглянуто питання управління файлами із медіаданими шляхом додавання до основної медіаінформації додаткових даних про ці дані та управління ними. Наведено приклади використання та можливості застосування таких метаданих. Описано та порівняно деякі стандарти метаданих для розповсюджених аудіоформатів, зокрема mp3. Розглянуто підходи до реалізації програмних засобів, інструментів та бібліотек для роботи з метаданими. Описано структуру додатку, що може використовуватися для маніпуляції метаданими. Описано та реалізовано алгоритм кластерного аналізу (kmeans)

для адаптації кольорової палітри графічного застосунку відповідно до зчитаних метаданих. Приведено шаблони багатопотокового програмування, які дозволяють прискорити виконання аналізу. Проаналізовано інші застосунки та засоби, що виконують подібні завдання.

*Ключові слова:* метадані, mp3, аудіо, обробка даних, C#, аудіофайли, ID3, фрейм.

## ABSTRACT

**Stovmanenko Vladyslav Olegovych. Informational system for ID3v2 tags processing in mp3 files. Decoding binary data. - manuscript copyright.**

Data identification process which uses special kind of information alongside with existing mediadata was reviewed in this article. Possibilities of its usage and examples of its applying was described. Some standards of audiometadata, mp3 in particular was mentioned and analyzed. Different approaches to implementation of programs that work with metadata was reviewed. Chosen and described approach for creating application that works with big amounts of audiofiles. Special algorithm for cluster analysis was described and implemented (kmeans) for colour palette of user interface according to read metadata. Some templates for multithreading was given. They were used to speed up analysis execution. Similar applications that gives opportunity to work with audio metadata was analyzed.

**The object of the research** — data about other data, or, so-called, metadata.

**The subject of the research** — different kinds of metadata for various audio formats.

The purpose of the research — Demonstrate skills and abilities acquired in university, get experience of creating science work production and, in the same time, get experience in creating complex application for solving some kind of tasks.

In the “Relevance” section was described usage of metadata for data processing, audio in particular, given some statistics about streaming services and its role in metadata development. Also Dublin group was mentioned as world wide metadata

standardization center. There is a brief history of metadata application in both mechanic and electronic ways.

“Theory” section gives characteristics of available approaches for storing metadata into audio files, provides information about audio-codecs with comparison table between them in terms of metadata application. Described different metadata standards, compared it’s structure. There is justification of choosing mp3 as a target format and ID3 as the best candidates for research.

“Design” section brings set of IDEF diagrams which describe general flow of execution. Also there is some additional schemes (Block schemes, UML, etc) for clarifying the processes or assembly structure.

“Practic” section has description of available technologies for assembly implementing and causes to choose .Net and C# as a primary ones. All BCL classes used in assembly was described and given short explanation of using them in exactly the same way they were used. There are also mentions about ways for creating adaptive colour palette for graphical user interface according to metadata. Kmeans algorithm was chosen and implemented for this purpose. A big characteristic of different multithreading approaches for accelerating analysis execution was described.

*Key words:* metadata, mp3, audio, data processing, C#, music library, ID3, frame.