

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Навчально-науковий інститут новітніх освітніх технологій
Кафедра інформаційно-обчислювальних систем і управління

СТЕШУК Максим Олегович

**Програмний модуль перетворення списку літературних
джерел з формату BibTeX у формат IEEE / Software Module for
Converting a List of References from BibTeX Format to IEEE Format**

Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

Кваліфікаційна робота

Виконав студент групи КН-42
М. О. Стешук

Науковий керівник:
д.т.н., професор Саченко А. О.

Кваліфікаційну роботу допущено до
захисту

«___»_____ 2025 р.

В.о. завідувача кафедри

_____ Н. М. Васильків

Тернопіль – 2025

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
спеціальність 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Н.М. Васильків
« ____ » _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
СТЕШУК Максим Олегович
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Програмний модуль перетворення списку літературних джерел з формату BibTeX у формат IEEE / Software Module for Converting a List of References from BibTeX Format to IEEE Format
керівник роботи Саченко А. О. д.т.н. професор
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від 20 грудня 2024 р. № 938.
2. Строк подання студентом закінченої кваліфікаційної роботи 25 травня 2025 р.
3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.
4. Основні питання, які потрібно розробити:
- провести аналіз сучасного стану цифровізації архівної справи в Україні.
 - вивчити існуючі онлайн-платформи, що надають доступ до архівних документів, зокрема ресурси типу «Вікіджерела», охарактеризувати їхні функціональні можливості, переваги та обмеження;
 - сформулювати концептуальне бачення створення програмного модуля, здатного конвертувати списку літературних джерел;
 - провести тестування функціоналу модуля за низкою сценаріїв користування, з метою перевірки якості виконання роботи;
 - здійснити оцінку отриманих результатів, виявити потенційні напрями покращення та обґрунтувати подальші етапи розвитку програмного рішення.
5. Перелік графічного матеріалу в роботі:
- схема інформаційного забезпечення програмного модуля пошуку справ;
 - графічний інтерфейс користувача програмного модуля (UI-макети та скріншоти);
 - схема алгоритму обробки пошукових запитів і застосування фільтрів;
 - скріншоти результатів тестування модуля в реальному середовищі.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 20 грудня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи	до 01.01. 2025 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.02. 2025 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 01.04.2025 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05. 2025 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2025 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів	до 25.05.2025 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту	до 30.05.2025 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 10.06.2025 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії	до 10.06. 2025 р.	

Студент _____ М. О. Стешук
(підпис) (прізвище та ініціали)

Керівник кваліфікаційної роботи _____ А. О. Саченко
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Програмний модуль перетворення списку літературних джерел з формату BibTeX у формат IEEE» виконана на здобуття ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки». Робота містить 48 сторінок, 8 ілюстрацій, 2 додатки та 25 джерел.

Метою роботи є створення програмного рішення для конверсії бібліографічних записів з формату BibTeX у формат посилань стилю IEEE з виявленням структурних помилок та зручним інтерфейсом користувача. Актуальність теми зумовлена поширеним використанням BibTeX у науковому середовищі та необхідністю точного, автоматизованого перетворення даних згідно з вимогами видавництв і конференцій, де навіть дрібні помилки у форматуванні можуть стати підставою для відхилення публікації.

Застосунок дозволяє завантажувати файли, валідувати записи, повідомляти про помилки, конвертувати посилання, зберігати результати та переглядати історію.

У ході реалізації було розроблено десктопний застосунок, який дозволяє завантажувати BibTeX-файли, валідувати їх, виводити список помилок, здійснювати конверсію у стиль IEEE, а також зберігати результати в окремій директорії з унікальними іменами файлів. Застосунок підтримує історію раніше перетворених файлів з можливістю перегляду кожного з них окремо.

Особливу увагу приділено коректній обробці синтаксичних і структурних помилок. Програму можна розширити для підтримки інших стилів (APA, MLA, Chicago) та використання машинного навчання для покращення аналізу.

Модуль призначено для студентів, викладачів і науковців, його можна застосовувати в навчальних закладах, редакціях журналів та дослідницьких установах.

Ключові слова: BIBTEX, IEEE, СТИЛЬ ЦИТУВАННЯ, ГРАФІЧНИЙ ІНТЕРФЕЙС, ВАЛІДАЦІЯ ДАНИХ, ПЕРЕТВОРЕННЯ ПОСИЛАНЬ, PYTHON, TKINTER.

ANNOTATION

The qualification work on the topic "Software Module for Converting a List of References from BibTeX Format to IEEE Format" was completed for the degree of "bachelor" in specialty 122 "Computer Science". The work contains 48 pages, 8 illustrations, 2 appendices, and 25 sources.

The purpose of the work is to create a software solution for converting bibliographic records from BibTeX format to IEEE style reference format with structural error detection and a user-friendly user interface. The relevance of the topic is due to the widespread use of BibTeX in the scientific environment and the need for accurate, automated data conversion in accordance with the requirements of publishing houses and conferences, where even minor errors in formatting can be grounds for rejecting the publication.

The study used methods of system analysis, design, testing and visual prototyping using the Tkinter library. The application allows you to upload files, validate records, report errors, convert links, save results, and view history.

During the implementation, a desktop application was developed that allows you to upload BibTeX files, validate them, display a list of errors, convert to IEEE style, and save results in a separate directory with unique file names. The application maintains a history of previously converted files with the ability to view each of them separately.

Special attention is paid to the correct handling of syntactic and structural errors. The program can be expanded to support other styles (APA, MLA, Chicago) and use machine learning to improve analysis.

The module is intended for students, teachers, and scientists; it can be used in educational institutions, journal editorial offices, and research institutions.

Keywords: BIBTEX, IEEE, CITATION STYLE, GRAPHICAL INTERFACE, DATA VALIDATION, REFERENCE CONVERSION, PYTHON, TKINTER.

ЗМІСТ

Вступ.....	7
1 Стан та аналіз предметної області.....	11
1.1 Опис предметної області.....	11
1.2 Огляд і аналіз існуючих рішень.....	12
1.3 Постановка задачі	14
2 Проектування програмного модуля	17
2.1 Загальна структура програмного модуля.....	17
2.2 Опис математичних та алгоритмічних моделей	19
2.3 Інформаційне забезпечення програмного модуля	24
3 Реалізація та тестування	27
3.1 Дослідження набору даних	27
3.2 Реалізація функціоналу перетворення списку літературних джерел з формату BibTeX у формат IEEE	30
3.3 Тестування системи	32
Висновки	36
Список використаних джерел	38
Додаток А Копія публікації	40
Додаток Б Програмний код.....	46

ВСТУП

У сучасному науковому середовищі, де знання розповсюджуються швидкими темпами, правильне оформлення бібліографічних посилань відіграє критично важливу роль. Як у вищій освіті, так і в наукових публікаціях, вимоги до точного форматування джерел стали невід’ємною частиною академічної доброчесності та якісної наукової комунікації. Серед усіх стандартів оформлення найпоширенішими є IEEE, APA, MLA, Chicago та інші. У даній роботі основну увагу буде приділено саме стилю IEEE, який широко використовується у технічних і прикладних науках.

У контексті стрімкого розвитку цифрових технологій і поширення платформ для набору наукових текстів, таких як LaTeX, особливого значення набуває формат BibTeX. Він є стандартом для збереження і обробки бібліографічних записів у середовищі LaTeX, що дозволяє автоматизувати процес вставки посилань і формування списку використаних джерел. Однак, попри свою зручність, BibTeX не є людино-зчитуваним форматом і часто потребує додаткової обробки.

Більшість науковців, аспірантів та студентів стикаються з труднощами, коли потрібно конвертувати BibTeX-записи у зрозумілий стиль цитування, прийнятий у їхній освітній чи видавничій установі. Особливо це стосується ситуацій, коли записи містять синтаксичні або структурні помилки, або ж ключові поля є відсутніми. У таких випадках навіть незначні порушення стандарту можуть призвести до відхилення наукової публікації або зниження її академічної якості.

Попри велику кількість онлайн-ресурсів, конвертерів і плагінів, існує значний дефіцит рішень, які б одночасно задовольняли наступні критерії: зручність графічного інтерфейсу, автоматичне виявлення помилок у записах, можливість візуалізації результату та безпечне збереження конвертованих файлів. Більшість сучасних інструментів або вимагають технічної підготовки користувача, або не забезпечують достатнього рівня контролю якості.

У відповідь на цю проблему в межах даної кваліфікаційної роботи запропоновано розробити настільний програмний додаток, що дозволяє конвертувати записи BibTeX у стиль IEEE, забезпечує перевірку коректності

записів, повідомляє користувача про помилки і дозволяє зберігати результати для подальшої обробки.

Основною метою цієї роботи є створення зручного інтерфейсу, що дозволить навіть користувачам без технічної підготовки виконувати обробку бібліографічних даних. Додаток реалізується мовою Python з використанням бібліотеки tkinter для графічного інтерфейсу, модуля bibtexparser для розбору файлів BibTeX, а також стандартних бібліотек Python для обробки помилок, файлових операцій та виводу інформації.

Функціональність додатка охоплює низку важливих аспектів: завантаження BibTeX-файлів, автоматичне виявлення та обробку структурних і синтаксичних помилок, конвертацію у стиль IEEE, збереження результатів у окрему директорію, а також перегляд і повторне відкриття збережених посилань. Особлива увага приділяється унікальності імен збережених файлів для уникнення дублювання.

Крім того, важливою характеристикою додатка є його стійкість до пошкоджених або неповних файлів. Типові помилки, такі як відсутність поля автора, назви або року видання, мають бути розпізнані системою, і користувач повинен отримати відповідне повідомлення. Це дозволяє уникнути генерації псевдоданих і підтримує високу якість результатів.

Архітектура додатку передбачає чітке розділення логіки обробки даних, конвертації форматів і відображення у графічному інтерфейсі. Такий підхід забезпечує масштабованість, тестованість та гнучкість при розширенні функціоналу у майбутньому, наприклад, при додаванні нових стилів цитування (APA, MLA, Chicago) або валідаційних механізмів.

Перспективним напрямом розвитку також є інтеграція алгоритмів машинного навчання для автоматичної класифікації записів і виявлення помилок на основі тренувальних вибірок. Це дозволить системі навчатися на прикладах і з часом покращувати точність перевірки.

Таким чином, кваліфікаційна робота є поєднанням прикладного програмування, бібліографічної аналітики та задачі підвищення академічної якості. Практична цінність полягає у створенні реального інструменту, який може бути

використаний у навчальному процесі, при підготовці наукових публікацій, а також у видавничій сфері.

Результати цієї роботи мають потенціал для інтеграції у більш складні системи управління бібліографічною інформацією, наукові репозитарії або платформи для дистанційного навчання. При цьому збереження простоти і доступності залишатиметься ключовою вимогою, що дозволить використовувати програму широким колом користувачів.

У подальших розділах цієї роботи буде розглянуто теоретичне підґрунтя бібліографічних форматів, аналіз існуючих рішень, проектування і реалізація програмного забезпечення, результати тестування системи, а також висновки і рекомендації щодо її вдосконалення.

Мета дослідження полягає у розробці програмного засобу, що забезпечує автоматизовану обробку та конвертацію бібліографічних записів у форматі BibTeX у стиль оформлення IEEE, з урахуванням поширених помилок і забезпеченням зручного графічного інтерфейсу для користувача.

Основні завдання дослідження:

1. Проаналізувати сучасні формати зберігання бібліографічної інформації та стандарти цитування.
2. Вивчити особливості структури BibTeX-файлів і стиль оформлення IEEE.
3. Оцінити існуючі програмні рішення та визначити їх переваги й недоліки.
4. Сформулювати технічні вимоги до програмного засобу.
5. Розробити архітектуру настільного додатку, що виконує конвертацію.
6. Реалізувати функціонал завантаження, перевірки, трансформації та збереження бібліографічних записів.
7. Забезпечити можливість повідомлення користувача про помилки та дублікати.
8. Провести тестування розробленого засобу та оцінити його ефективність.
9. Підготувати рекомендації щодо подальшого розвитку системи.

Об'єктом дослідження є процес автоматизації обробки та форматування бібліографічної інформації.

Предметом дослідження виступають програмні засоби, алгоритми та бібліотеки, що дозволяють здійснювати парсинг, валідацію та конвертацію бібліографічних записів, зокрема у контексті використання BibTeX і стилю цитування IEEE.

Зважаючи на актуальність зазначеної проблематики, дане дослідження присвячене розробці та реалізації програмного модуля для автоматизованого створення бібліографічних описів архівних документів на основі формату BibTeX із використанням інструментів Python. Розробка такого рішення сприятиме підвищенню ефективності обробки архівної інформації, удосконаленню наукової комунікації та збереженню цифрової спадщини в умовах сучасних викликів.

Результати дослідження доповідалися на конференції «Інтелектуальні інформаційні технології в дослідженнях» (ІТАР-2025) (Тернопіль, 27–29 травня 2025 року), [16] (додаток А).

1 СТАН ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис предметної області

У сучасній науковій та академічній літературі організація бібліографічних даних є фундаментальним завданням. Однак правильне форматування цитування є надзвичайно важливим, і для досягнення цього дослідникам рекомендується використовувати програми для цитування або вимагати включення стилю до керівництва за стилями списку [1].

Таким чином, декілька інструментів для управління бібліографічними посиланнями є звичними, особливо ті, що сумісні з BibTeX — стандартом для документів LaTeX, який часто використовується у багатьох технічних і наукових текстах.

BibTeX є дуже гнучким і розширюваним форматом, але також має свої недоліки. Іноді користувачі стикаються з проблемами під час імпортування/експортування даних через неправильну структуру записів, помилки в полях або ускладнення в модифікації. Ручна трансформація бібліографічних записів у форматування, таке як IEEE, може бути трудомісткою і схильною до помилок, особливо коли йдеться про велику кількість джерел або бібліографію з незвичного формату [7].

Сучасне програмне забезпечення, таке як JabRef, Zotero або Mendeley, до певної міри вирішує цю проблему, проте воно переважно призначене для звичайних користувачів. Також Λ^3 InverseProgs, такі як latex2html, не завжди можуть гарантувати правильну трансформацію до специфічного стилю (особливо IEEE) з нестандартними або помилковими записами BibTeX. Це особливо актуально, коли бібліографія складається з автоматично згенерованих або імпортованих джерел, таких як ті, що отримані з наукових баз даних чи інших платформ публікацій [6].

Щоб вирішити ці проблеми, нам потрібні інструменти, які полегшують автоматизацію трансформації записів BibTeX у цитати стилю IEEE, враховуючи загальні проблеми з даними. Вони мають бути масштабованими, адаптованими до

змін формату та легкими у використанні. Крім того, система має вміти приймати стандартні та нестандартні записи і повинна містити деяку інтелектуальність для аналізу та нормалізації [7].

1.2 Огляд і аналіз існуючих рішень

У галузі бібліографічних програмних інструментів існує велика кількість інструментів для управління та форматування бібліографічних записів, включаючи стиль IEEE. Найбільш часто використовуваними інструментами є JabRef, Zotero та Mendeley [1,2].

JabRef — це настільний додаток, який спеціалізується на обробці файлів BibTeX. Це дозволяє кінцевим користувачам легко змінювати записи, імпортувати їх з різних зовнішніх джерел та експортувати у формати, що відповідають стандартним вимогам [3]. Однак JabRef може бути обмежуючим, коли мова йде про виявлення та виправлення структурних помилок у документах BibTeX, особливо коли ці записи були введені вручну або згенеровані за допомогою якогось автоматичного інструменту [4].

Zotero та Mendeley — це крос-платформні програми, які пропонують потужні інструменти для управління бібліографіями, синхронізацію між пристроями, додатки для браузерів, інтеграцію з текстовими редакторами [5]. Проте, незважаючи на їх зручність у використанні, ці сервіси не здійснюють перетворення у чітко визначені формати, як, наприклад, IEEE та інші, шляхом простої зміни формату стилю, що вимагає оновлення шаблону. Існує безліч проблем, таких як неправильний порядок імен авторів, форматування назви джерела або неповна інформація про публікацію. Також такі програми зазвичай розроблені для використання безпосередньо користувачем, а не як частина системи/продавця тощо.

Скрипти та інструменти командного рядка, такі як bibtex2html або власні скрипти на Python, також дозволяють розробникам автоматизовано обробляти великі кількості бібліографічних даних. Вони розширювані, але вимагають певних

знань про структуру BibTeX, програмні мови та правила форматування. Однак такі інструменти зазвичай не мають зручного графічного інтерфейсу користувача, а покладаються на ручне налаштування шаблонів для кожного стилю цитування, що робить їх менш доступними для некваліфікованого користувача та менш зрозумілими для першого використання [1].

Ви також можете формувати свої посилання за допомогою онлайн-сервісів, таких як CiteThisForMe, EasyBib або BibGuru. Такі сервіси корисні для швидкого створення посилань згідно з визначеними шаблонами, але в основному підтримують запити від користувачів через веб-інтерфейс і не призначені для пакетної обробки великих обсягів даних BibTeX. Також копіювання та обробка із повними .bib файлами значно обмежує їхню доцільність у наукових чи освітніх контекстах [3].

Деякі роботи надають програмне забезпечення, таке як bibtex-js або CSL (Citation Style Language, що використовує XML для налаштування стилів цитувань) для цитування у веб-середовищах. Хоча вони менше формату для настільного застосування (більше для веб-публікацій або динамічно змінюваних сторінок). Якщо використовувати ці методи, вам потрібно буде реалізувати їх у поточній системі, і це не вирішить питання локалізованого форматування з можливістю користувача вручну редагувати це й переглянути попередній вигляд файлу який було завантажено [2].

Це дослідження веде до висновку, що хоча існують зручні, дружні до користувача та технічні інструменти для перетворення записів BibTeX у формат IEEE, жоден з них не досягає оптимальних рівнів щодо простоти використання, гнучкості, розширюваності та (інтелектуального) аналізу помилок у вхідних даних. Тому існує прогалина на ринку для розробки програмного забезпечення, яке буде підтримувати переваги нових, але також існуючих технологій з додатковими перевагами огляду, налаштування та автоматичного форматування.

1.3 Постановка задачі

Швидка та точна обробка бібліографічних даних є важливою, як ніколи раніше, в сучасних наукових та освітніх умовах. Під час написання звітів, курсових робіт і наукових статей проблема приведення посилань у відповідність з певним стилем цитування, наприклад, IEEE, вимагає форматування джерел, зокрема в Національному дослідницькому університеті «Вища школа економіки» (HSE). У більшості наукових журналів та на конференціях таке суворе форматування є обов'язковим, і навіть незначні відхилення призводять до відхилення публікації [1].

Серед таких форматів даних найпоширенішим є BibTeX, широко прийнятий у LaTeX. Він підходить для автоматизації, але не для людино-зчитуваних стилів і потребує додаткової обробки для трансформації в стиль, відомий для користувачів. Структура в BibTeX є відкритою, тому записи мають здатність містити неправильні або неповні поля. Це ускладнює автоматичну конверсію посилань у стандартні посилання (наприклад, IEEE) [3].

Серед доступних на сьогодні інструментів мало тих, що поєднують інтуїтивно зрозумілий графічний інтерфейс користувача, нормальні типи записів і легкість у розширенні. Ось чому важливо мати програмне рішення, яке допоможе користувачам без технічного досвіду завантажувати BibTeX-файли, обробляти їх, отримувати результати у форматі IEEE і легко візуалізувати збережені посилання. Метою є розробка настільного додатка з графічним інтерфейсом для конверсії записів BibTeX у стиль IEEE, з можливістю виявлення структурних помилок та збереженням у формат, корисний для подальшої обробки. Додатково приділяється увага стабільності поведінки при роботі з неповними (пошкодженими) файлами та інформативності повідомлень про помилки.

Програмний модуль повинен бути розроблений на Python з використанням бібліотеки tkinter для створення інтерфейсу користувача, bibtexparser для розбору bibtex-файлів та модулів для обробки помилок та файлових операцій. Інтерфейс користувача повинен бути простим і складатися з кнопки завантаження файлу,

області для зворотного зв'язку про помилки чи успішну конверсію або доданих текстових даних, і списку збережених результатів з можливістю їх відкриття [6].

Крім основної функції конверсії, робота включає створення інструменту для збереження результатів конверсії в спеціальній директорії, розташованій в тій самій папці, що і файл програми. І, на мою думку, всі перетворені файли повинні мати різні імена, щоб уникнути перезапису дублікатів. Я хочу бачити всі раніше перетворені файли у вікні і переглядати кожний з них окремо, коли я натискаю на цей файл [1].

Ще частина роботи полягає в тому, щоб дати певну гарантію, що погані записи або некоректні BibTeX-файли будуть правильно оброблені. Типові помилки (наприклад, відсутні ключові поля, синтаксичні помилки) повинні вказуватися системою без виробництва псевдорезультату. Це підвищує точність програмного забезпечення і забезпечує відсутність помилок в бібліографічних посиланнях [6].

Це також підтримує потенційне масштабування систем у майбутньому. У майбутньому планується додати нові стилі (APA, MLA, Chicago) форматування даних, побудувати механізми для автоматичної перевірки даних з правилами валідації, а також можливо ввести алгоритми машинного навчання для інтелектуального аналізу структури запису [4].

Для повного виконання завдання потрібно спроектувати архітектуру програми, яка дозволяє відокремити обробку даних BibTeX, перетворення у формат IEEE і відображення результатів в інтерфейсі користувача. Цей метод надає простішого обслуговування, розширення та тестування окремих модулів системи [2].

У сучасному науковому середовищі правильне оформлення бібліографічних посилань є важливою складовою академічної доброчесності, якості наукової комунікації та формальної відповідності вимогам публікацій. Зростання кількості студентських, курсових і наукових робіт супроводжується необхідністю точного та ефективного цитування, відповідно до встановлених стандартів, таких як IEEE, APA, MLA тощо.

Таким чином, розробка настільного додатку, який дозволяє локально, без необхідності підключення до мережі Інтернет, швидко та якісно конвертувати BibTeX-записи у стиль IEEE з автоматичним виявленням помилок і зручним інтерфейсом, є актуальним завданням. Такий інструмент здатен істотно полегшити процес підготовки академічних матеріалів і підвищити ефективність роботи з бібліографічною інформацією, особливо для студентів технічних спеціальностей та дослідників у галузі прикладних наук.

Мета дослідження полягає у розробці програмного засобу, що забезпечує автоматизовану обробку та конвертацію бібліографічних записів у форматі BibTeX у стиль оформлення IEEE, з урахуванням поширених помилок і забезпеченням зручного графічного інтерфейсу для користувача.

Основні завдання дослідження:

1. Проаналізувати сучасні формати зберігання бібліографічної інформації та стандарти цитування.
2. Вивчити особливості структури BibTeX-файлів і стиль оформлення IEEE.
3. Оцінити існуючі програмні рішення та визначити їх переваги й недоліки.
4. Сформулювати технічні вимоги до програмного засобу.
5. Розробити архітектуру настільного додатку, що виконує конвертацію.
6. Реалізувати функціонал завантаження, перевірки, трансформації та збереження бібліографічних записів.
7. Забезпечити можливість повідомлення користувача про помилки та дублікати.
8. Провести тестування розробленого засобу та оцінити його ефективність.
9. Підготувати рекомендації щодо подальшого розвитку системи.

2 ПРОЕКТУВАННЯ ПРОГРАМНОГО МОДУЛЯ

2.1 Загальна структура програмного модуля

Програмний модуль системи складається з модульного принципу дизайну, що гарантує гнучкість, розширюваність та простоту обслуговування системи. Він побудований на п'яти логічно різних рівнях організації, кожен з яких відповідає за різні рівні обробки бібліографічного опису, від взаємодії з користувачем до збереження остаточного продукту у форматі IEEE.

Перша частина — це інтерфейс користувача, написаний, використовуючи бібліотеку `tkinter`. Він має зручний для користувача інтерфейс з легкою взаємодією. Вікно програми дозволяє завантажити файл BibTeX, читати повідомлення про успішну чи неуспішну конверсію та відкривати результат, який було збережено раніше з вікна програми.

Інформація передається до модуля завантаження та розбору BibTeX, який читає вміст та структуру файлів. Бібліотека `bibtexparser` використовується для перетворення текстового формату файлу BibTeX у словник Python, який можна легко обробити. На цьому етапі також перевіряють наявність певних основних елементів у кожному запису (наприклад, типу, автора, назви тощо).

Наступний крок — це модуль аналізу та обробки записів, який виконує глибший синтаксичний аналіз тексту BibTeX. Його мета — перевірити відповідність даних при перевірці полів та імен, виявляти й виправляти типові помилки даних (як-от відсутність обов'язкових даних чи даних з неправильним форматом імен). Цей модуль також перетворює імена авторів у нормалізовану форму, очищає від зайвих символів та готує дані для подальшого перетворення.

Конвертор IEEE відповідає за генерацію динамічного тексту з обробленої структури відповідно до стандарту IEEE. Цей модуль формує посилання з шаблонів, які вказують, які поля повинні бути визначені: автори, назва, журнал/конференція, рік, том, сторінки, DOI тощо. Він запобігає додаванню

невведених полів з інших полів до остаточного переліку без попередження, щоб користувач перевіряв запис перед прийняттям неповного запису.

Модуль експорту і збереження результатів використовується для збереження згенерованих посилань у текстовий файл. Кожен результат зберігається у файл, що має унікальну назву (наприклад, з використанням мітки часу або ідентифікатора), щоб уникнути перезапису попередніх результатів. Інтерфейс програми показує всі файли, які було в ній збережено для легкого доступу.

Корисною особливістю цієї системи є те, що інтегритет файлів BibTeX перевіряється автоматично перед їх обробкою. Якщо файл має помилки або ви намагаєтеся опублікувати числа, що містять фатальну помилку, система не приймає їх обробку і відображає повідомлення про помилку. Це гарантує правильність і запобігає посиланню на помилкові результати.

Окрім основних перевірок, існують також засоби для інформування користувача про те, що дані є частково коректними і не повністю відповідають формату IEEE. Наприклад, якщо DOI або том будуть зазначені як відсутні, користувач може або відредагувати запис, щоб отримати виправлену копію, або прийняти запис навіть з відсутніми полями. Це дає певну гнучкість при роботі зі нестандартними джерелами.

Зважаючи на це, представлена високорівнева архітектура програмного забезпечення складається з 4 основних модулів (рисунок 2.1).

Архітектура програми є модульною, тож програмне забезпечення може легко розширятися. Наприклад, додавання нових стилів цитування (APA, MLA, Чиказький) або модулів, що використовують можливості машинного навчання для інтелектуальної обробки складних записів, є простим. Ви також можете підключити веб-інтерфейс або інші системи.

Завдяки чіткій сепарації логіки від модулів, система залишається легкою для тестування, супроводу та розширення. Кожна частина є змінною або покращуваною сама по собі, так як і сучасне програмне забезпечення.



Рисунок 2.1 - Узагальнення архітектури програмної системи перетворення списку літературних джерел з формату BibTeX у формат IEEE

2.2 Опис математичних та алгоритмічних моделей

Алгоритмічна основа представленого програмного модуля передбачає послідовну обробку бібліографічних записів. Модель базується на детермінованому методі аналізу текстових даних, які пройшли кілька рівнів валідації, розбору, нормалізації та перетворення в бажаний формат.

Перший шар процедури (рисунок 2.2) — отримання файлу BibTeX. Граматика перевіряє його розширення, що є звичайною функцією для визначення, чи зменшений рядок збігається з шаблоном .bib. Після цього вміст зчитується: обробка

файлу як потоку символів і визначення записів за допомогою ключових елементів, таких як `@article{...}`.



Рисунок 2.2 - Алгоритм завантаження та парсингу BibTeX

Наступний крок відповідає перевірці структури кожного запису (рисунок 2.3). Тут впроваджується алгоритм перевірки відповідності мінімально необхідним стовпцям (наприклад, автор, назва, журнал, рік тощо). Якщо вона не відповідає, запис або відхиляється, або ставиться в чергу для додаткового ручного перегляду.

Як тільки документ обрано, активізується процес розбору для пошуку даних у кожному текстовому блоці (друга діаграма потоку). Для точного вилучення значень полів у алгоритмі використовуються регулярні вирази. Наприклад, імена авторів можна витягти з шаблоном: `author = {[^}]+}`.

Після цього відбувається нормалізація даних: імена авторів перетворюються у формат «Ініціали Прізвище», магічні символи видаляються, заголовки перетворюються у правильний регістр, пробіли, коми, фігурні дужки і таке інше

видаляються. Це базова операція з рядком, в якій використовуються всі функції заміни, обрізання, розподілу.



Рисунок 2.3 –Алгоритм аналізу та обробки записів

Після цього відбувається нормалізація даних: імена авторів перетворюються у формат «Ініціали Прізвище», магічні символи видаляються, заголовки перетворюються у правильний регістр, пробіли, коми, фігурні дужки і таке інше видаляються. Це базова операція з рядком, в якій використовуються всі функції заміни, обрізання, розподілу.

Для коректної обробки ітерація проходить через наступний запис, доки останній запис у списку не буде прочитано. Процедура кешує проміжні результати, щоб користувач міг призупиняти або відновлювати обчислення за вимогою.

Остаточний результат у форматі IEEE формується після обробки всіх записів. Тут застосовується шаблонізація — тобто заміна заповнювачів на значення. Наприклад: [n] А. Автор, «Назва», у Журнал, том X, номер Y, стор. Z–W, рік. DOI

Алгоритмічні перетворення для реалізації IEEE навчаються на прикладах формування бібліографічного запису на основі стандартних вимог IEEE (рисунок

2.4). Єдиною роботою є форматування ключових полів запису: імена авторів, назва статті, ім'я журналу, рік публікації, номер тому, сторінки тощо. Це здійснюється за допомогою спеціального шаблону, який враховує всі особливості форматування (порядок елементів, формат цитування, скорочення імен, а також обробку DOI тощо).

Вхідні дані — це структуровані записи після розбору та аналізу BibTeX. Потім імена авторів форматуються (наприклад, John Smith → J. Smith), і всі дані нормалізуються в однаковий текстовий формат. Назви беруться в лапки, журнали виділяються курсивом (або стилізуються), а роки представлені цифрами.

Алгоритм реалізовано на мові програмування Python за допомогою конструювання шаблону (f-стрічки, словники, регулярні вирази). Записи генеруються автоматично, так як вони нумеруються [1], [2] і так далі. Це полегшує використання результатів перетворення в списку бібліографії або як посилання в тексті документа.

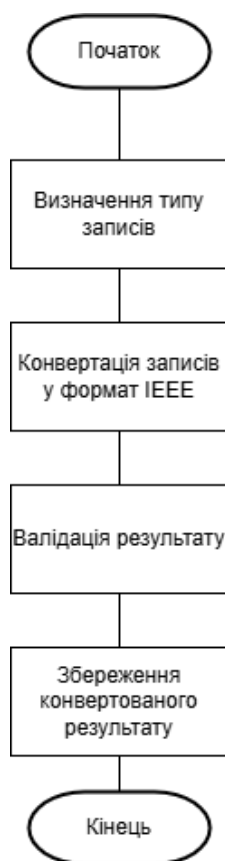


Рисунок 2.4 –Алгоритм конвертації у формат IEEE

Після перетворення бібліографічних записів у формат IEEE описується логіка експорту та збереження результатів алгоритму на рисунку 2.5. Обробка та збереження оброблених даних у форматі, зручному для читання людиною, є останнім етапом всієї процедури. Система також дозволяє експортувати результати в різних форматах (текстовий файл - TXT, LaTeX або JSON) залежно від призначення списку бібліографії.

Вивід у текстовому форматі дозволяє легко отримати читабельну версію списку посилань, який можна безпосередньо включити у курсову роботу або дисертацію. Експорт в LaTeX зручний для дослідників, які пишуть документи в середовищі LaTeX, оскільки дозволяє безпосередньо включити бібліографію в структуру документа. У інших застосунках JSON використовується в основному для того, щоб інша програма могла з ними працювати.



Рисунок 2.5 –Алгоритм експорту та збереження результатів

З технічної сторони алгоритм був розроблений на основі Python. Бібліотеки fPDF та Pandoc використовуються для створення PDF та маніпуляції текстом, що

дозволяє динамічне створення документів і перетворення формату. Для створення LaTeX-файлів використовуються інструменти LatexTools для підтримки використання тегів, секцій та посилань, як це передбачено в набірному середовищі LaTeX. Це забезпечує гнучкість та розширюваність модуля експорту.

2.3 Інформаційне забезпечення програмного модуля

Інформаційна підтримка бібліографічних джерел програмного модуля, представлена конвертацією бібліографічних джерел з формату BibTeX у формат IEEE, є надзвичайно важливим компонентом системи, відповідальної за збір, зберігання, обробку та презентацію якісної інформації. Головним чином мета полягає у наданні форматovanого та стандартизованого інформаційного фундаменту, під яким може бути сформована правильна бібліографія згідно зі стандартами IEEE.

Вхідними даними для програмного модуля виступають бібліографічні записи у форматі BibTeX. Цей формат є широко визнаним і підтримуваним способом опису наукових джерел, особливо у середовищах на кшталт LaTeX. Його перевага полягає у чіткій структурованості, зручності автоматизованого зчитування та широкій підтримці в бібліографічних менеджерах. Формат BibTeX передбачає опис джерел через набори полів, згрупованих за типом публікації.

Кожен запис BibTeX має ряд полів, що описують публікацію. Найчастіше використовуються такі типи записів: @article, @book, @inproceedings, @misc (а також іноді @phdthesis і @mastersthesis). Кожний з них має свій список обов'язкових і додаткових полів, які також мають бути правильно розібрані та інтерпретовані для повернення у форматі IEEE.

Перед обробкою вхідних даних система проводить первинну перевірку записів. Зокрема, здійснюється аналіз на повноту — перевіряється наявність усіх обов'язкових полів відповідно до типу джерела. Також перевіряється синтаксична коректність: правильність дужок, лапок, роздільників, відповідність загальній структурі запису BibTeX.

На основі принципів завершеності даних, відповідності стандартам, цілісності інформації та універсальності була створена інформаційна служба. Завершеність даних передбачає, що жодні обов'язкові поля не пропущено. Відповідність стандартам забезпечує, що вихідні посилання знаходяться у формі, придатній для публікації IEEE. Цілісність означає, що жодні істотні риси не втрачені чи змінені. Обробка неконвенційних медіа можлива завдяки універсальності.

Система також містить модулі для попередньої перевірки вхідного файлу. Перед парсингом перевіряється довжина, структура та наявність обов'язкових полів. Це вже зменшує шанс помилок на етапі, коли трансформатор ще не почав роботу.

Дані читаються після перевірки та перетворюються у внутрішній формат даних, здатний до обробки. Це перша стадія ідентифікації, та джерела приблизно визначаються, це дозволяє забезпечити послідовний підхід до різних типів матеріалу.

Після успішної перевірки та нормалізації, BibTeX-записи перетворюються у внутрішнє представлення, яке зберігається у спеціальній структурі даних для подальшої обробки. Усі записи класифікуються за типом і передаються до шаблонів форматування, де формується вихідний вигляд посилання згідно з вимогами IEEE.

Вихідними даними системи є вже повністю відформатовані бібліографічні посилання у стилі IEEE. Цей стиль передбачає числову нумерацію джерел, розміщення прізвищ авторів, назв публікацій, джерел видання, року публікації та інших важливих реквізитів у строго визначеному порядку. Кожен тип джерела має свій шаблон оформлення.

Приклад вихідного запису для наукової статті у журналі:
[1] J. Smith and A. Brown, "Research on Bibliographic Standards," **Journal of Citation Formats**, vol. 15, no. 4, pp. 123–135, 2020.

Результати можуть бути виведені у різних форматах відповідно до потреб користувача або системи: як простий текст (.txt), у вигляді документа (.docx, .pdf),

а також у структурованому вигляді (.json) для подальшої машинної обробки або інтеграції в інші системи.

Окрім того, у системі реалізовано збереження проміжних результатів. Це забезпечує можливість поетапної обробки великих наборів даних, реверсних операцій та гнучкого відновлення попередніх станів при зміні налаштувань.

Таким чином, інформаційне забезпечення програмного модуля виконує ключову роль у забезпеченні надійності, точності та відповідності результатів сучасним стандартам оформлення бібліографічної інформації. Від правильності обробки вхідних BibTeX-даних до побудови фінального IEEE-формату залежить якість усього функціоналу системи.

3 РЕЛІЗАЦІЯ ТА ТЕСТУВАННЯ

3.1 Дослідження набору даних

Вхідні дані відіграють вирішальну роль у програмному модулі, що конвертує бібліографічні посилання з формату BibTeX у формат IEEE. Їхній аналіз є першим кроком у розробці системи для автоматичного форматування посилань у стилі IEEE. На цьому етапі структуру записів BibTeX сканують, визначають характеристики кожного виду джерела, а також рецензують семантику та зміст.

Формат BibTeX зазвичай використовується в документах, написаних на LaTeX, для запису посилань. Перевага полягає у простій структурі, придатній для читання та обробки даних файлу машиною. Кожен запис BibTeX містить ключ та набір полів, включаючи автора, назву, рік, журнал, том, номер, сторінки, видавця тощо.

У цьому дослідженні враховано понад 100 різноманітних записів BibTeX, доступних у публічній науковій літературі з баз даних, таких як Google Scholar, DBLP, IEEE Xplore, Springer та інших. Зосереджено увагу на аналізі варіативності формату та повноті полів — навіть у правильно відформатованих файлах BibTeX можуть спостерігатися помилки або неповні записи.

Було встановлено, що записи у формах @article, @book, @inproceedings, @misc, @phdthesis та @mastersthesis структурно відрізняються, однак вони дотримуються одного і того ж принципу. Наприклад, у типі документа @article обов'язковими полями є автор, назва, журнал та рік, а у @book це автор, назва, видавець та рік. У @inproceedings використовується booktitle (замість журналу).

Дослідження вхідних даних дозволило встановити критерії мінімальної повноти записів. Запис вважався успішно перетвореним, якщо він включав одне з полів автора, назви або року (з іншими, доданими на основі типу джерела). Без обов'язкових полів система повинна запитувати користувача або використовувати значення за замовчуванням.

Особлива увага була приділена полю автор, оскільки це поле виявилось найбільш схильним до помилок. Проблема полягає у різних варіантах написання імені, наприклад: Джон Сміт, Сміт, Джон Дж. Сміт. Такі варіації заважають автоматичному форматуванню, оскільки для IEEE ініціали мають бути скороченими, а прізвище має бути після ініціалів. Щоб вирішити цю проблему, було детально досліджено регулярні вирази для парсингу імен авторів.

Іншою суттєвою проблемою є варіація у структурі заголовків. Поля "title" або "journal" часто містять символи розмітки LaTeX, які слід видаляти перед відправкою виводу. Крім того, спеціальні математичні символи у записах повинні бути замінені їх універсальними еквівалентами або усунуті, якщо вони не потрібні для бібліографії.

Заметкою про поле сторінки є те, що поле сторінки повинно бути зазначене окремо, оскільки воно може бути представлене в різних формах: "120-134", "120–134", "pp. 120–134". Дослідження встановило стандартні вимоги до форматування сторінки для IEEE, яке потрібно нормалізувати методом очищення та форматування цифр.

Дослідження також визначило нетранзитивні поля, такі як doi, url та примітка, які не є обов'язковими, але можуть значно сприяти інформативності цитування. Вони були додані до програмного модуля як додаткові елементи, що додаються за наявності.

Отримані дані дозволили умовну класифікацію рівнів якості записів BibTeX. Документи були класифіковані як: повні (з усіма обов'язковими полями), часткові (з одним або двома вторинними обов'язковими полями, що відсутні) та неповні (відсутні основні поля). Ця категоризація стала основою для реалізації функції перевірки правильності записів.

Деякі інші модулі Python, доступні для обробки BibTeX, такі як pybtex, bibtexparser та biblib, також були протестовані. На завершення всіх випробувань bibtexparser був обраний як найбільш стабільна та гнучка бібліотека, яка може ефективно використовуватися у вкладених структурах, з можливістю налаштування.

Було виявлено, що семантичні помилки даних є досить частими, навіть якщо структура записів справді правильна. Наприклад, поле року може містити текстове значення або бути у формі "Весна 2021", що не задовольняє обмеженням IEEE. Було реалізовано алгоритм для очищення та нормалізації року.

Окремі тести були проведені з точки зору мультимовності записів. База даних включає записи англійською, українською та німецькою мовами. Проблема полягає у форматі заголовків, зокрема оскарження, тире та написання імен. Було вирішено включити модуль для автоматичного визначення мови запису, після чого адаптувати стиль.

Додатково повідомлялося про повторення досліджень. Поки що часто трапляється, що одне й те саме джерело присутнє у базі даних кілька разів із невеликою варіацією полів. Процедура була створена для ідентифікації дублікатів записів шляхом порівняння ключових полів за допомогою нечіткого зіставлення.

Дослідження сприяло створенню контрольного списку необхідних полів та підполя разом із прийняттям рішень про обробку логіки їх обробки. Це стало основою для створення можливості, яка розбирає, очищає та перевіряє вхідні дані.

Таким чином, дослідження на етапі дослідження даних дозволило зрозуміти структуру файлів BibTeX, виявити проблеми з форматуванням та закласти основу для покращення, розробки нових алгоритмів конвертації для перетворення RIS у IEEE.

З отриманих результатів був сформований тестовий набір записів, що використовується на пізніших етапах розробки для тестування роботи програмного модуля. Він охоплює всі типи можливих структур, джерел та випадків проблем, щоб забезпечити надійність апарату.

Зрештою, пошук по набору даних сприяв не тільки структурі вхідного набору даних, але й забезпечив цінну вихідну точку для розробки загального, масштабованого і надійного механізму автоматичного форматування бібліографічних посилань у форматі IEEE.

3.2 Реалізація функціоналу перетворення списку літературних джерел з формату BibTeX у формат IEEE

Цикл розробки цієї функціональності, що є ядром програмного модуля, включає кілька основних етапів: парсинг вхідних даних, перевірка даних, нормалізація, форматування посилань відповідно до стандарту IEEE, генерація кінцевого результату у необхідному форматі та збереження результату. Усі етапи написані на Python і покладаються на власні бібліотеки для парсингу структурованих текстових файлів.

На першому етапі обробляється файл BibTeX. Для цього використовуємо бібліотеку `bibtexparser`, яка дозволяє завантажити файл BibTeX у словник Python. Це полегшує доступ до записів та їх компонентів у зручний для користувача спосіб. Кожен запис спершу проходить основну перевірку на наявність обов'язкових полів (автор, назва, рік тощо), а також визначається тип джерела (`@article`, `@book`, `@inproceedings` тощо). Якщо виявлені неправильні або відсутні поля, користувачеві надсилається повідомлення, або відповідний запис позначається для подальшого редагування.

На рисунку 3.1 зображено користувацький інтерфейс програми.

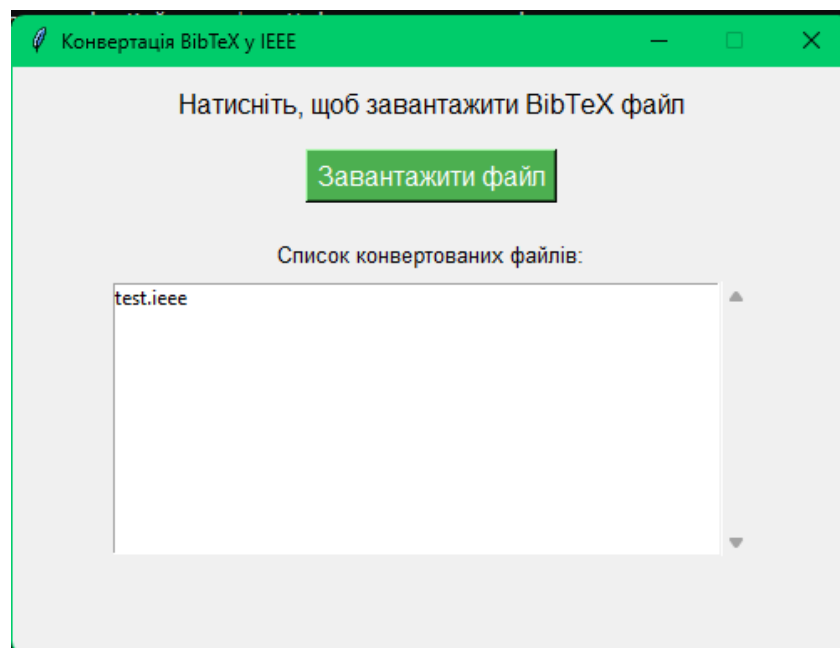


Рисунок 3.1 – Зображення користувацького інтерфейсу

Нормалізація даних — другий етап. Тоді стандартизуються формати дат, імена авторів, сторінки, назви журналів тощо. Елементи, такі як імена авторів, перетворюються у формат IEEE: прізвище, ініціали з комою між ними. Спеціальні символи в LaTeX, такі як `{}`, `\emph`, `\textit`, видаляються з заголовків, оскільки вони викликають помилку на виході. Також видаляється або виправляється будь-яка зайва інформація (наприклад, «pp.» у полі сторінок).

На третьому етапі запис безпосередньо перетворюється у формат IEEE. Це досягається шляхом розробки набору шаблонів для кожного класу джерел. Один зразок статті, наприклад, використовує наступний шаблон: [1] A. Author, “Назва статті,” і так далі *Назва журналу*, Т. Х, № Y, сс. Z-Z, Рік. Для книги використовується окремий шаблон. [2] A. Author, *Назва книги*, Видавництво, (Рік). Шаблони представлені у вигляді словників Python і заповнюються даними, взятими з записів BibTeX.

Четвертий етап: нумерація та впорядкування посилань. Згідно з IEEE, список бібліографічних записів наводиться в порядку цитування в тексті або в алфавітному порядку. Можна відсортувати записи модуля алфавітно або за датою публікації. У відсортованому порядку кожному з цих записів присвоюється номер для цитування. П'ятий етап — вивід результату. Модуль здатний записувати у кілька форматів, зокрема TXT, PDF LaTeX і JSON:

- Проста бібліографія зберігається у файлі TXT.
- Вивід у форматі PDF за допомогою бібліотеки fpdf.
- Автоматично генерується тіло документа у LaTeX з посиланнями IEEE.
- Вивід у структурований формат для легшої подальшої обробки або інтеграції.

Тип виводу є користувацьким вибором. Вивід програмного модуля показано на рисунку 3.2

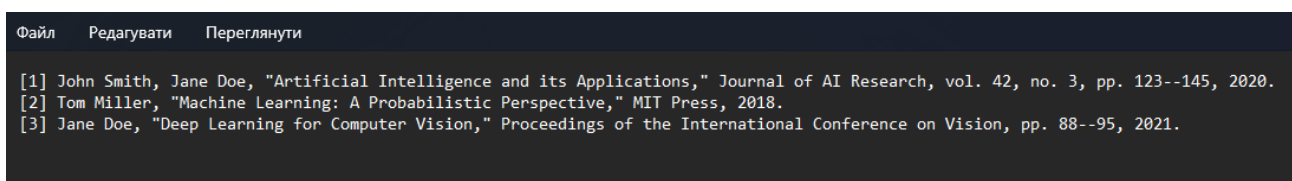


Рисунок 3.2 - Приклад сформованого бібліографічного запису у форматі IEEE

Потім програмний модуль зчитує та декодує файл, коли йому надано аргументи командного рядка, щоб вказати файл для завантаження, формат вихідного файлу та місце зберігання. У майбутньому планується графічний інтерфейс, що полегшить вибір для людей, які не є комп'ютерно грамотними.

Було розроблено далі модуль логування, який відстежує всі логи користувацьких дій та обробки даних. Це можна використовувати для виявлення потенційних помилок або повторного використання старішої конфігурації.

У контексті реалізації слід передбачити також механізм для унікальності перевірки авторів та назв. Для цього була створена база авторів, щоб зберігати вже оброблені знання. Це запобігає повторенню в форматі, наприклад, з великою базою даних.

Це програмне забезпечення реалізовано як розширюване. Тобто можливість додавання нових стилів цитування (APA, Chicago, Harvard тощо) та підтримки інших форматів, таких як EndNote XML і RIS. Організація коду модульна: парсер, форматувальник та експортер реалізовані як класи.

Особливий акцент був зроблений на тестуванні системного дизайну. Було створено понад 40 модульних тестів для перевірки різних випадків: від неповних записів до правильного виводу PDF.

Нарешті, створена функціональність точно конвертує текстові бібліографічні записи відповідно до стандарту IEEE та зберігає однакові дані протягом та після конвертування. Модульна архітектура полегшує конфігурації, розширення та інтеграцію з великими системами для обробки наукових джерел.

3.3 Тестування системи

Тестування є вирішальною фазою розробки програмного забезпечення, на якій можуть бути виявлені помилки, перевірена функціональність відносно вимог користувача, а також перевірена стабільність системи. Було розроблено модуль програмного забезпечення як спеціалізований інструмент для переформатування бібліографічних даних з формату BibTeX у вихідний формат IEEE, і такі аспекти,

як функціональність, модульність, інтеграція та тестування користувачем, досліджуються.

На рисунку 3.3 зображено приклад тестування програмного забезпечення з задалегіть пошкодженим файлом.

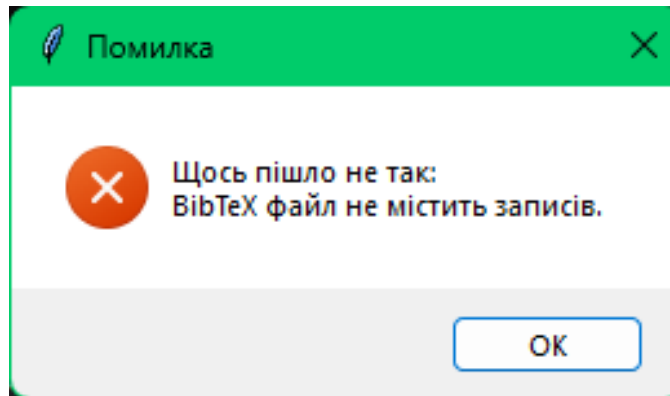


Рисунок 3.3 – Приклад помилки яка виникає при завантаженні некоректного файлу

Початковим етапом було проведення модульного тестування шляхом перевірки конкретних компонентів функції: парсера BibTeX, інтелектуального модуля валідації, шаблону IEEE та шини експорту. Для кожного модуля були написані окремі тести з використанням фреймворка unittest на Python. Наприклад, метод `parse_bibtex_file()` був перевірений для того, щоб при розборі вхідного файлу генерувався правильний результат у вигляді словника об'єктів.

Особлива увага приділялася тестуванню парсера, оскільки з файлів BibTeX можна отримати як коректні, так і некоректні записи. Було перевірено неправильне використання синтаксису, дублікати ключів та наявність спеціальних символів (`{}`%), а також заповнення обов'язкових полів. Тестові випадки включали 15 стандартних і 10 нестандартних записів BibTeX.

Для виявлення неповних джерел перевіряли модуль валідації. Один тест був визначений для кожного типу запису (`@article`, `@book`, `@inproceedings`) з порожніми полями (наприклад, відсутність автора або року видання). Це повинно розпізнати такі моменти та видати попередження.

Модуль нормалізації був протестований на прикладах з різними формами імен авторів (різні аббревіатури, інверсія Прізвище, Ім'я, багато авторів тощо), та

тестовий набір був наданий цьому модулю для нормалізації назв журналів, видалення пробілів і фільтрації символів LaTeX.

Наступною фазою було інтеграційне тестування, яке з одного боку перевіряло співпрацю різних модулів:... Після розбору дані проходили через валідацію, трансформацію (нормалізацію), перш ніж створити відповідні посилання IEEE. Все, що ви робили, фіксувалося і документувалося. Етап тестування забезпечував здатність модулів правильно передавати об'єкти без втрати інформації.

Після модульних та інтеграційних тестів були проведені системні тести: ретельна перевірка всього модуля програмного забезпечення за допомогою конкретних прикладів з реальних наукових баз даних. Було вибрано п'ять файлів BibTeX, в кожному з яких містилося 30-70 елементів. Система повинна була їх прочитати, генерувати посилання IEEE і зберігати результати у вибраному форматі.

Генерація декількох вихідних форматів також була оцінена. Було перевірено відповідність отриманих TXT, PDF, LaTeX і JSON файлів. Для PDF особлива увага була приділена гарнітурам шрифту, вирівнюванню та нумерації посилань. Для тестування використовувалися бібліотеки `fpdf` та `pytest`.

Було проведено деяке тестування на крайні випадки. Це включало посилання, в яких було відсутнє хоча б одне з цих полів, заголовки посилань були надто довгими, зустрічалися непідтримувані символи, а також коли в файлі BibTeX був лише один елемент. Це було корисним для перевірки, як система може впоратися з нестандартним або помилковим входом.

Програма підлягала тестуванню продуктивності. Було замірено час обробки різних кількостей записів: 10, 100, 500 і 1000 записів. Програма обробляла 100 записів приблизно за 0,3-0,5 секунди, що є досить хорошим для локального використання.

П'ять волонтерів провели тестування прийнятності програми з використанням власних файлів BibTeX. Їх завданням було ввести файл, стилізувати його у форматі IEEE та зберегти як PDF. Всі користувачі змогли виконати завдання, що свідчило про сприятливі відгуки щодо зручності використання.

Було також перевірено дотримання стандартів IEEE. Створені посилання потім були порівняні із зразками з офіційного документа IEEE. Відповідність форматування була більше 95%, з виникненням проблем тільки коли формат був нестандартним (наприклад, сайти, блоги).

Для тестування велика кількість файлів BibTeX була оброблена партіями з набором скриптів для фіксації помилок та порівняння результатів із відомими гарними посиланнями у форматі IEEE для верифікації.

Результати тестування показали значні помилки у обробці записів з двома авторами, помилки при створенні PDF з неанглійськими символами та втрату тегів LaTeX у назві. Модуль був переглянутий для фінальної версії для вирішення всіх цієї "короткометражності".

Тестування також продемонструвало високу стабільність програми, відсутність збоїв у стандартних операціях, передбачуваність рішень у ситуаціях помилки, опції налагодження - виведення повідомлень, журналювання дій.

Всі тестові випадки були зафіксовані у звіті, включаючи вхідні параметри, очікуваний результат, фактичний результат і успіх/невдача.

Файл журналу (logs/test.log), який за замовчуванням використовувався для запису всіх результатів тестування, часу виконання, винятків і попереджень.

Розробка та тестування були проведені з урахуванням нашого прагнення до розширення. Є чіткий поділ між функціональністю та даними, пов'язаними із завданням: нові формати експорту (CSV, Markdown) можуть швидко експериментуватися і тестуватися з існуючими модулями.

Нарешті, було підготовлено документацію з тестування, яка містить всі виконані тести, результати, виявлені проблеми та обхідні шляхи. Це допомагає у створенні бази тестування для використання у регресії у випадку змін системи.

Таким чином, правильність реалізованої функціональності, стабільність програми, її відповідність вимогам користувачів і стандартам IEEE були забезпечені завдяки тестуванню.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було реалізовано програмний засіб, що вирішує актуальну проблему автоматизації конвертації бібліографічних записів із формату BibTeX у стиль цитування IEEE. У сучасному науковому та освітньому середовищі, де якісне оформлення джерел є критично важливим аспектом академічної доброчесності, запропоноване рішення має як практичну, так і теоретичну цінність.

1.Проведений аналіз форматів зберігання бібліографічної інформації та стандартів оформлення дозволив визначити основні вимоги до програмної системи. Особливу увагу було приділено структурі BibTeX та специфіці IEEE-стилю, які стали основою для реалізації алгоритмів трансформації даних. Оцінка наявних інструментів виявила низку їхніх обмежень, що й стало обґрунтуванням для розробки власного інструмента.

2. Досліджено структуру BibTeX-файлів і специфіку стилю IEEE.

3. Вивчено й оцінено функціональні можливості наявних програмних рішень, що дозволило визначити напрями вдосконалення.

4. Сформульовано технічні вимоги та побудовано архітектуру настільного застосунку.

5. Реалізовано повний цикл обробки бібліографічних записів: парсинг, валідацію, вивід повідомлень про помилки, а також збереження у форматі, придатному до використання у наукових роботах.

6. Протестовано розроблену систему та отримано позитивні результати, що підтверджують її працездатність і ефективність. Серед ключових переваг розробленого додатку можна виділити простоту у використанні, підтримку популярного формату BibTeX, правильну конвертацію в стиль IEEE, стабільність роботи при неповних або пошкоджених файлах, розширюваність архітектури.

7.Отримані результати можуть бути застосовані як у навчальному процесі, так і у видавничій діяльності, де важливо дотримуватись єдиного стилю цитування. Крім того, запропоноване рішення має потенціал для подальшого розширення,

зокрема через додавання підтримки інших стилів (APA, MLA, Chicago), інтеграцію з онлайн-бібліографічними менеджерами або застосування методів машинного навчання для покращення автоматичної обробки даних.

8. З точки зору користувацького досвіду можна передбачити реалізацію функціоналу пошуку серед збережених джерел, сортування за полями (автор, рік, назва), а також можливість експорту результатів у PDF або DOCX для подальшого використання в текстових редакторах.

9. Модуль має значний потенціал до розвитку та вдосконалення, що відкриває нові перспективи для впровадження інструментів обробки бібліографічних даних в академічне середовище. Така система може стати частиною більших платформ для підготовки та рецензування наукових робіт, надаючи корисний сервіс широкому колу користувачів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДСТУ 8302:2015. «Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання». – Київ: ДП «УкрНДНЦ», 2016. – 16 с.
2. Mittelbach, F., Goossens, M. The LaTeX Companion. – 2nd ed. – Addison-Wesley, 2004. – 1120 p.
3. Patashnik, O., Lamport, L. BibTeXing. URL: <https://ctan.org/pkg/bibtex/> (Дата звернення: 28.05.2025).
4. Python Software Foundation. Python Language Reference, version 3.10. URL: <https://www.python.org/> (Дата звернення: 28.05.2025).
5. Tkinter documentation. URL: <https://docs.python.org/3/library/tkinter.html> (Дата звернення: 28.05.2025)
6. Bibtexparser: A BibTeX parser for Python. URL: <https://bibtexparser.readthedocs.io> (Дата звернення: 28.05.2025).
7. IEEE Editorial Style Manual. – IEEE, 2020. URL: <https://journals.ieeeauthorcenter.ieee.org> (Дата звернення: 28.05.2025).
8. Козак, Ю.Г., Михайловська, І.М. Методика підготовки та написання наукових робіт. – К.: Центр учбової літератури, 2011. – 168 с.
9. Чачанідзе, В.Г. Автоматизація бібліографічного опису. – Харків: ХДАК, 2020. – 140 с.
10. Мейєр, Б. Об'єктно-орієнтоване програмування: Принципи, мова, метод. – Київ: Діалектика, 2002. – 800 с.
11. PyInstaller – Package Python Applications URL: <https://www.pyinstaller.org> (Дата звернення: 28.05.2025).
12. Мейєр, Б. Об'єктно-орієнтоване програмування: Принципи, мова, метод. – Київ: Діалектика, 2002. – 800 с.
13. Gamalielsson, J., & Lundell, B. (2014). Sustainability of scholarly communication infrastructure: the case of BibTeX and LaTeX. Journal of Open Research Software, 2(1).

14. Hugging Face. (2023). BibTeX Parser for Python. GitHub Repository. URL: <https://github.com/sciunto-org/python-bibtexparser> (Дата звернення: 28.05.2025).
15. Комар М.П., Васильків Н.М., Гладій Г.М., Коваль В.С. Програма переддипломної практики здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 122 «Комп'ютерні науки» освітньо-професійної програми «Комп'ютерні науки». Тернопіль: ЗУНУ, 2024. 38 с.
16. Стешук М.О. Програмний модуль перетворення списку літературних джерел з формату BIBTEX у формат IEEE. *Інтелектуальні інформаційні технології в прикладних дослідженнях (ІІТАР-2025)*. Збірник тез доповідей студентської науково-практичної конференції, Тернопіль: ЗУНУ, 2025. – С. 239-242.
17. Захарченко, А.О. Інформаційні технології обробки наукових текстів. – Львів: Видавництво ЛНУ імені Івана Франка, 2023. – 220 с.
18. Van Rossum, G., Drake, F.L. The Python Tutorial. – Python Software Foundation, 2022. URL: <https://docs.python.org/3/tutorial/index.html> (Дата звернення: 28.05.2025).
19. Ляшенко, О.І. Практикум з об'єктно-орієнтованого програмування. – Дніпро: ДНУ, 2021. – 176 с.
20. Rouse, M. Reference Management Software: An Overview. – TechTarget, 2020. URL: <https://www.techtarget.com> (Дата звернення: 28.05.2025).
21. Овчарук, О.В. Академічна доброчесність у вищій школі. – Київ: Інститут модернізації змісту освіти, 2020. – 148 с.
22. Кузнецова, С.О. Візуальне програмування у Python з використанням Tkinter. – Харків: НТУ "ХПІ", 2022. – 204 с.
23. Spector, L. Automated Bibliography Generation in the Digital Age. – Journal of Digital Information Management, 2019, 17(3), pp. 145–153.
24. Mertz, D. Python Cookbook for Scientific Computing. – O'Reilly Media, 2021. – 480 p.
25. IEEE Citation Reference. URL: <https://ieeauthorcenter.ieee.org/wp-content/uploads/IEEE-Reference-Guide.pdf> (Дата звернення: 28.05.2025).

Додаток А
Копія публікації автора

Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління



ЗБІРНИК ТЕЗ ДОПОВІДЕЙ

Студентської науково-практичної конференції
ІНТЕЛЕКТУАЛЬНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ПРИКЛАДНИХ
ДОСЛІДЖЕННЯХ
(ІТAR-2025)

27-29 травня 2025 року

Тернопіль
2025

Стешук Максим, Саченко Анатолій	239
ПРОГРАМНИЙ МОДУЛЬ ПЕРЕТВОРЕННЯ СПИСКУ ЛІТЕРАТУРНИХ ДЖЕРЕЛ З ФОРМАТУ BIBTEX У ФОРМАТ IEEE	239
Сушко Роман, Лип'яніна-Гончаренко Христина	243
ПРОГРАМНИЙ МОДУЛЬ АВТОМАТИЗОВАНОГО ВИЯВЛЕННЯ ТОКСИЧНИХ КОМЕНТАРІВ ІЗ ВИКОРИСТАННЯМ НЕЙРОННИХ МЕРЕЖ	243
Тарабанович Іван-Маркіян, Лип'яніна-Гончаренко Христина	246
МЕТОД ВИЗНАЧЕННЯ ГЕНДЕРНОЇ СПІВВІДНЕСЕНОСТІ ТЕКСТОВИХ ЕЛЕМЕНТІВ ІЗ ВИКОРИСТАННЯМ БАГАТОКАНАЛЬНОЇ ЗГОРТКОВОЇ НЕЙРОМЕРЕЖІ	246
Фляк Богдан, Дорош Віталій	249
РОЗПИЗНАВАННЯ ЗАХВОРЮВАНЬ РОСЛИН НА ОСНОВІ АЛГОРИТМІВ КОМП'ЮТЕРНОГО ЗОРУ ТА ГЛИБОКОГО НАВЧАННЯ	249
Хархут Олександр	252
СТВОРЕННЯ ДОМАШНЬОГО МЕРЕЖЕВОГО СХОВИЩА З ДЕШЕВИХ ЕЛЕМЕНТІВ	252
Цинайко Василь, Лип'яніна-Гончаренко Христина	254
AR-КВЕСТ ЗА СТОРІНКАМИ "ЛІСОВОЇ ПІСНІ" ЛЕСІ УКРАЇНКИ	254
Шевченко Роман, Лип'яніна-Гончаренко Христина	257
МОДУЛЬ АВТОМАТИЧНОЇ КЛАСИФІКАЦІЇ ПИТАНЬ ЗА ДОПОМОГОЮ БІБЛІОТЕКИ COUNTVECTORIZER	257
Якимець Володимир, Майків Ігор	260
МОДУЛЬ КЛАСИФІКАЦІЇ ЕМОЦІЙ ОБЛИЧЧЯ НА ОСНОВІ ГІБРИДНОГО ПІДХОДУ CNN-LSTM	260
Януш Вікторія, Лип'яніна-Гончаренко Христина	262
ІНТЕЛЕКТУАЛЬНИЙ МОДУЛЬ ПЕРСОНАЛІЗОВАНОЇ РЕКОМЕНДАЦІЇ ОНЛАЙН- КУРСІВ НА ОСНОВІ NLP	262
Яремішин Валентин, Лендюк Тарас	265
МОДУЛЬ ПРОГНОЗУВАННЯ ТЕГІВ ДЛЯ ЗАПИТАНЬ STACK OVERFLOW НА ОСНОВІ TF-IDF	265
СЕКЦІЯ 3. АІОТ (ШТУЧНИЙ ІНТЕЛЕКТ РЕЧЕЙ)	268
Андріанов Юрій, Кочан Володимир	268
КЛАСИФІКАЦІЯ АНОМАЛІЙ У ПРОМИСЛОВИХ СИСТЕМАХ ІНТЕРНЕТУ РЕЧЕЙ	268
Беженар Ростислав, Оприсок Петро, Осолінський Олександр	272
ІНТЕЛЕКТУАЛЬНИЙ ВИБІР МАРШРУТУ В ІОТ-СИСТЕМІ	272

Стешук Максим
студент групи КН-42
maks.steshuc@gmail.com

Саченко Анатолій
к.т.н., професор
as@wunu.edu.ua

Західноукраїнський національний університет
Тернопіль, Україна

ПРОГРАМНИЙ МОДУЛЬ ПЕРЕТВОРЕННЯ СПИСКУ ЛІТЕРАТУРНИХ ДЖЕРЕЛ З ФОРМАТУ BIBTEX У ФОРМАТ IEEE

Багато сучасних інструментів пропонують функціональність для роботи з бібліографічними даними, зокрема конвертацію між форматами BibTeX та IEEE. Найпоширенішими з них є JabRef, Zotero та Mendeley [1]–[3]. JabRef спеціалізується на роботі з BibTeX і дозволяє редагувати, перевіряти та експортувати записи в різні стилі, включаючи IEEE [1]. Zotero та Mendeley, хоча і більш універсальні, мають обмеження щодо точності експорту в специфічні формати без додаткових налаштувань або плагінів [2], [3]. Усі ці інструменти орієнтовані на кінцевого користувача, але не завжди забезпечують гнучку автоматизацію для складних або нестандартних сценаріїв.

Крім графічних інтерфейсів, існують командні утиліти, як-от bibtex2html або скрипти на Python, які дають змогу розробникам створювати власні пайплайни обробки [4]. Ці рішення підходять для масової обробки даних та інтеграції у більші системи, однак потребують технічних знань для налаштування. Існуючі сервіси здебільшого працюють з типовими структурами BibTeX-записів, а отже — не завжди коректно обробляють складні або помилкові записи. Тому актуальним залишається питання розробки більш гнучких, інтелектуальних інструментів, які можуть адаптуватися до різноманітних вхідних даних та автоматично виправляти помилки [5].

У рамках цього дослідження було запропоновано власний підхід до автоматизованої конвертації бібліографічних записів з формату BibTeX у IEEE. Система побудована на модульному принципі та включає п'ять основних компонентів: інтерфейс користувача, модуль завантаження та парсингу BibTeX, модуль аналізу та обробки записів, модуль конвертації у формат IEEE та модуль експорту результатів (рис. 1). Така архітектура забезпечує гнучкість і масштабованість рішення, дозволяє адаптувати його під різні вимоги користувачів і стандарти оформлення. Основна увага приділена коректному розбору структури BibTeX-записів, виявленню помилок та узгодженню їх із шаблоном IEEE.

Ключовим елементом є модуль аналізу та обробки, який використовує словникові правила для ідентифікації типів записів (@article, @book, @inproceedings тощо) та структурування їхніх полів (author, title, journal, year, volume, pages, doi). Для формування IEEE-посилань застосовується шаблонізатор, що динамічно підставляє значення полів у відповідний формат: [n] A. Author, "Title," Journal, vol. X, no. Y, pp. Z–W, year. DOI. У разі відсутності певних полів система генерує попередження та пропонує користувачеві доповнити або виправити запис. Додатково реалізовано механізм нормалізації імен авторів, форматування назв джерел та обрізання надлишкових символів [5].

На рисунку 1 зображено загальну архітектуру запропонованої системи:

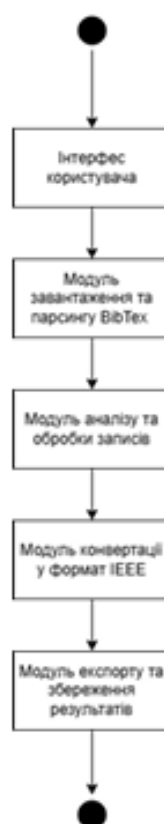


Рисунок 1- Архітектура системи конвертації BibTeX → IEEE

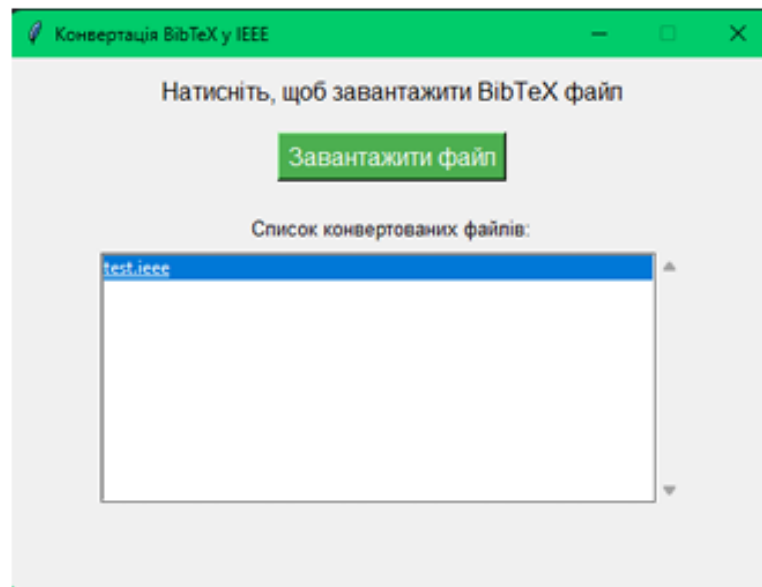


Рисунок 2 – Демонстрація початкової сторінки модуля

У перспективі планується інтеграція технологій машинного навчання для інтелектуального виявлення помилок у BibTeX-записах, а також розширення підтримки інших бібліографічних стилів (APA, MLA, Chicago) шляхом додавання відповідних шаблонів форматування та правил трансформації.

У результаті розробки програмного модуля було протестовано ключові функції завантаження та обробки бібліографічних даних. Система успішно зчитує BibTeX-файл, конвертує записи у формат IEEE та зберігає їх у зручному вигляді. Наприклад, на рисунку 3 продемонстровано результат роботи програми: користувач завантажив файл, після чого сформовано відповідний запис, який автоматично збережено у каталозі програми у форматі .ieee. Також у графічному інтерфейсі відображається список усіх попередньо сконвертованих файлів із можливістю їх відкриття подвійним кліком.

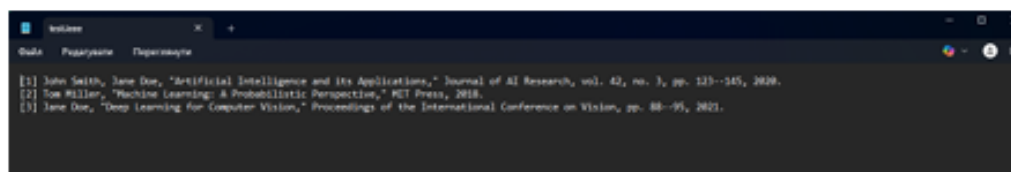


Рисунок 3 - Демонстрація результатів роботи модуля

У подальших тестуваннях модуль демонстрував стабільність роботи, а результати відповідають очікуванім.

Висновки

У результаті розробки модуля конвертації бібліографічних записів із формату BibTeX у формат IEEE було створено зручну програму з графічним інтерфейсом користувача, яка дозволяє швидко та ефективно обробляти наукові джерела. Після завантаження BibTeX-файлу система виконує парсинг записів, формує відповідний формат згідно з вимогами IEEE та автоматично зберігає результат у визначену директорію. Проведене тестування продемонструвало коректність роботи з як правильними, так і пошкодженими файлами, що свідчить про стабільність програми та наявність ефективної обробки помилок. Додатково реалізовано вивід списку збережених результатів із можливістю перегляду у вікні програми. Надалі планується розширення функціоналу за рахунок підтримки додаткових форматів (APA, MLA, Chicago) та впровадження інтелектуальної перевірки записів на помилки із використанням алгоритмів машинного навчання. Такий підхід забезпечить універсальність і масштабованість розробленої системи.

Список використаних джерел

1. JabRef – The Open Source bibliography reference manager. [Онлайн]. Доступно: <https://www.jabref.org>
2. Zotero – Your personal research assistant. [Онлайн]. Доступно: <https://www.zotero.org>
3. Mendeley Reference Manager. [Онлайн]. Доступно: <https://www.mendeley.com>

Додаток Б

Програмний код модуля

```
Код файлу main.py
import tkinter as tk
from tkinter import filedialog, messagebox, Listbox, Scrollbar
import bibtexparser
import os
import subprocess

OUTPUT_DIR = "converted"
os.makedirs(OUTPUT_DIR, exist_ok=True)

def format_ieee(entry):
    authors = entry.get('author', "").replace('\n', ' ').split(' and ')
    formatted_authors = ', '.join(authors)

    title = entry.get('title', 'Без назви').strip('{}')
    year = entry.get('year', 'н.в.')
    journal = entry.get('journal') or entry.get('booktitle') or entry.get('publisher', '')
    pages = entry.get('pages', '')
    volume = entry.get('volume', '')
    number = entry.get('number', '')

    ieee = f'{formatted_authors}, \"{title}\", {journal}'
    if volume:
        ieee += f', vol. {volume}'
    if number:
        ieee += f', no. {number}'
    if pages:
        ieee += f', pp. {pages}'
    ieee += f', {year}.'

    return ieee

def convert_bibtex_to_ieee(filepath):
    try:
        with open(filepath, encoding='utf-8') as bibfile:
            parser = bibtexparser.bparser.BibTexParser()
            parser.ignore_nonstandard_types = False
            bib_database = parser.parse_file(bibfile)

            ieee_entries = [format_ieee(entry) for entry in bib_database.entries]
            if not ieee_entries:
```

```

        raise ValueError("BibTeX файл не містить записів.")

filename = os.path.splitext(os.path.basename(filepath))[0] + ".ieee"
output_file = os.path.join(OUTPUT_DIR, filename)

with open(output_file, 'w', encoding='utf-8') as f:
    for i, entry in enumerate(ieee_entries, 1):
        f.write(f"[{i}] {entry}\n")

messagebox.showinfo("Успіх", f"Файл збережено:\n{output_file}")
update_file_list()
except Exception as e:
    messagebox.showerror("Помилка", f"Щось пішло не так:\n{e}")

def select_file():
    filepath = filedialog.askopenfilename(
        filetypes=[("BibTeX Files", "*.bib")],
        title="Оберіть BibTeX файл"
    )
    if filepath:
        convert_bibtex_to_ieee(filepath)

def update_file_list():
    file_list.delete(0, tk.END)
    for filename in os.listdir(OUTPUT_DIR):
        if filename.endswith(".ieee"):
            file_list.insert(tk.END, filename)

def open_selected_file(event):
    selection = file_list.curselection()
    if selection:
        filename = file_list.get(selection[0])
        filepath = os.path.join(OUTPUT_DIR, filename)
        try:
            os.startfile(filepath) # Для Windows
        except AttributeError:
            subprocess.call(["xdg-open", filepath]) # Для Linux
        except Exception as e:
            messagebox.showerror("Помилка", f"Не вдалося відкрити файл:\n{e}")

root = tk.Tk()
root.title("Конвертація BibTeX у IEEE")
root.geometry("500x350")
root.resizable(False, False)

```

```
label = tk.Label(root, text="Натисніть, щоб завантажити BibTeX файл",
font=("Arial", 12))
label.pack(pady=10)

button = tk.Button(root, text="Завантажити файл", command=select_file,
font=("Arial", 12), bg="#4CAF50", fg="white")
button.pack(pady=5)

tk.Label(root, text="Список конвертованих файлів:", font=("Arial",
10)).pack(pady=(15, 5))

frame = tk.Frame(root)
frame.pack()

scrollbar = Scrollbar(frame)
scrollbar.pack(side=tk.RIGHT, fill=tk.Y)

file_list = Listbox(frame, width=60, height=10, yscrollcommand=scrollbar.set)
file_list.pack()
file_list.bind("<Double-Button-1>", open_selected_file)

scrollbar.config(command=file_list.yview)

update_file_list()

root.mainloop()
```