

Київський національний торговельно-економічний університет
Кафедра інженерії програмного забезпечення та кібербезпеки

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЄКТ

на тему:

«Проектування програмного забезпечення визначення академічного плагіату в роботах здобувачів освіти»

Студента 2 курсу, 2м групи,
спеціальності 121 «Інженерія
програмного забезпечення»
спеціалізації «Інженерія
програмного забезпечення»
Науковий керівник кандидат
технічних наук,
доцент кафедри інженерії
програмного забезпечення та
кібербезпеки
Гарант освітньої програми доктор
економічних наук, професор
кафедри інженерії програмного
забезпечення та кібербезпеки

підпис студента

Гудкова Валерія
Свгенівна

підпис керівника

Сашн'ова Мар'яна
Василівна

підпис гаранта

Токар Володимир
Володимирович

КИЇВ — 2021

Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь магістр

Спеціальність 121 «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри інженерії програмного
забезпечення та кібербезпеки

Криворучко О. В.

«29» грудня 2020 р.

Завдання

на випускний кваліфікаційний проєкт студентів

Гудковой Валерії Євгенівні

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проєкту «Проектування програмного
забезпечення визначення академічного плагіату в роботах здобувачів освіти»

Затверджена наказом ректора від «28» грудня 2020 р. № 3923

2. Строк здачі студентом закінченого проєкту 25 листопада 2021

3. Цільова установка та вихідні дані до проєкту

Мета проєкту: підвищення якості нових текстів, зокрема робіт здобувачів вищої освіти, за рахунок зменшення частки запозичених матеріалів, чого можна досягти шляхом проектування та розробки програмного забезпечення, яке могло би із достатньою точністю визначати наявний рівень оригінальності тексту.

Об'єкт дослідження: процес пошуку плагіату у текстових даних.

Предмет дослідження: методи та засоби для пошуку плагіату у текстових даних.

4. Консультанти проєкту із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проєкту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ТА ЗАСОБІВ ДЛЯ ВИЗНАЧЕННЯ ЗАПОЗИЧЕНЬ У ТЕКСТОВИХ ПРАЦЯХ

1.1 Опис особливостей галузей застосування систем перевірки на антиплагіат

1.1.1 Рекламні тексти та їх застосування для пошукової оптимізації SEO

1.1.2 Особливості аналізу на антиплагіат наукових публікацій у спеціалізованих журналах

1.1.3 Дослідження навчальних робіт здобувачів вищої освіти на наявність плагіату

1.2 Основні підходи до визначення текстових запозичень

1.2.1 Використання пошукових систем загального призначення

1.2.2 Робота зі спеціалізованими інформаційними системами відповідного призначення

1.2.3 Застосування готових систем-аналогів для пошуку плагіату, аналіз переваг та недоліків

1.3 Виокремлення невирішених проблем у засобах-аналогах та постановка задачі дослідження

1.4 Висновки по розділу

РОЗДІЛ 2 ОБҐРУНТУВАННЯ ПРОЄКТНИХ РІШЕНЬ ПРИ РОЗРОБЛЕННІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВИЗНАЧЕННЯ АКАДЕМІЧНОГО ПЛАГІАТУ В РОБОТАХ ЗДОБУВАЧІВ ОСВІТИ

2.1 Створення алгоритмічної складової проєктованого засобу

2.2 Аналіз необхідності використання СУБД та проєктування сховища даних

2.3 Обґрунтування вибору засобів розробки

2.3.1 Вибір технології програмування

2.3.2 Вибір мови програмування

2.3.3 Обґрунтування вибору засобів розробки

2.3.3.1 Вибір інтегрованого середовища розробки

2.3.3.2 Вибір інструментального засобу для збереження даних у системі

2.3.3.3 Додаткові засоби розробки

2.4. Висновки по розділу

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВИЗНАЧЕННЯ АКАДЕМІЧНОГО ПЛАГІАТУ В РОБОТАХ ЗДОБУВАЧІВ ОСВІТИ

3.1 Прототипування інтерфейсу користувача програмного забезпечення

3.2 Особливості реалізації сховища даних продукту

3.3 Опис реалізації основних алгоритмічних складових продукту

3.4 Документування створеного ПЗ (розробка керівництва користувача)

3.5 Тестування створеного програмного продукту та аналіз його ефективності

3.6 Висновки по розділу

ВИСНОВКИ

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

6. Календарний план виконання проєкту

№ пор.	Назва етапів випускного кваліфікаційного проєкту	Строк виконання етапів проєкту	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускного кваліфікаційного проєкту</i>	<i>21.09.2020</i>	<i>21.09.2020</i>
2.	<i>Розробка та затвердження завдання на проєкт магістра (заоч)</i>	<i>29.12.2020</i>	<i>29.12.2020</i>
3.	<i>Вступ та перелік літературних джерел</i>	<i>27.02.2021</i>	<i>27.02.2021</i>
4.	<i>Розробка технічного завдання</i>	<i>20.03.2021</i>	<i>20.03.2021</i>
5.	<i>Розділ 1. Аналіз існуючих методів та засобів для визначення запозичень у текстових працях</i>	<i>16.04.2021</i>	<i>16.04.21</i>
6.	<i>Розділ 2. Обґрунтування проєктних рішень при розробленні програмного забезпечення визначення академічного плагіату в роботах здобувачів освіти</i>	<i>24.05.2021</i>	<i>24.05.2021</i>
7.	<i>Розділ 3. Реалізація програмного забезпечення визначення академічного плагіату в роботах здобувачів освіти</i>	<i>21.06.2021</i>	<i>21.06.2021</i>
8.	<i>Розробка програми та методики тестування</i>	<i>18.10.2021</i>	<i>18.10.2021</i>
9.	<i>Написання наукової статті</i>	<i>22.05.2021</i>	<i>22.05.2021</i>
10.	<i>Керівництво користувача</i>	<i>21.10.2021</i>	<i>21.10.2021</i>
11.	<i>Висновки та пропозиції</i>	<i>01.11.2021</i>	<i>01.11.2021</i>
12.	<i>Здача випускного кваліфікаційного проєкту на кафедру (перша перевірка)</i>	<i>03.11.2021</i>	<i>03.11.2021</i>
13.	<i>Підготовка автореферату та презентації доповіді</i>	<i>03.11.2021</i>	<i>03.11.2021</i>
14.	<i>Попередній захист випускного кваліфікаційного проєкту</i>	<i>22.11.2021 – 25.11.2021</i>	<i>22.11.2021</i>
15.	<i>Здача зброшурованої випускного кваліфікаційного проєкту</i>	<i>25.11.2021</i>	<i>25.11.2021</i>
16.	<i>Зовнішнє рецензування випускного кваліфікаційного проєкту</i>	<i>26.11.2021</i>	<i>26.11.2021</i>
17.	<i>Підготовка до публічного захисту випускного кваліфікаційного проєкту</i>	<i>за розкладом роботи ЕК</i>	

7. Дата видачі завдання _____ «29» грудня 2020 р.
8. Науковий керівник випускного кваліфікаційного проєкту _____
Сашнєва М. В.
(прізвище, ініціали, підпис)
9. Гарант освітньої програми _____ Токар В. В.
(прізвище, ініціали, підпис)
10. Завдання прийняв до виконання студент _____ Гудкова В. Є.
(прізвище, ініціали, підпис)

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins or other markings visible.

(*nidnuc*, *dama*)

(ПИБ, *nidnuc*, *data*)

Випускний кваліфікаційний проєкт студента _____ Гудкової В. Є.
(прізвище, ініціали)

Гарант освітньої програми _____ Токар В. В.
(прізвище, ініціали, підпис)

Завідувач кафедри _____ Криворучко О. В.
(підпис, прізвище, ініціали)

« » 20 p.

АНОТАЦІЯ

У випускній кваліфікаційній роботі розглядається проектування програмного забезпечення визначення академічного плагіату в роботах здобувачів освіти.

У роботі виконано аналіз існуючих методів та засобів для визначення запозичень у текстових працях, обґрунтовано проєктні рішення при розробленні програмного забезпечення визначення академічного плагіату в роботах здобувачів освіти. Розглянуто переваги та недоліки готових систем-аналогів для пошуку запозичень.

Програмне забезпечення може використовуватися у реальній практичній діяльності особами, що створюють тексти (для самоконтролю) та їх перевіряють.

Для реалізації обрано мову Java та середовище IntelliJ IDEA. Розроблено інтерфейс користувача, проведено програмну реалізацію і виконано тестування створеного ПЗ, в результаті чого встановлено його цілком задовільні експлуатаційні показники та досить добрий збіг результатів роботи програми з іншими популярними аналогами.

Ключові слова: академічний плагіат, засоби автоматичного пошуку плагіату, програмне забезпечення, Java, IntelliJ IDEA.

ABSTRACT

This diploma project is dedicated to the design of software for the definition of academic plagiarism in the students' works.

The paper analyzes the existing methods and tools for determining replications in text works, substantiates project solutions in the development of software for the definition of academic plagiarism in the students' works. The advantages and disadvantages of ready comparable systems for replications are considered.

The software can be used in real practice by people who create texts (for self-control) and check them.

Java language and IntelliJ IDEA environment were chosen for implementation. The user interface was developed, software implementation was carried out and the newly developed software was tested. Entirely satisfactory performance indicators and good concurrence of program work results with other popular analogues were established.

Keywords: academic plagiarism, the tools for automatic plagiarism determination, software, Java, IntelliJ IDEA.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ – програмне забезпечення

MVC (англ. Model-View-Controller) – архітектурний шаблон, який використовується під час проєктування та розробки програмного забезпечення.

ПК – персональний комп'ютер

ООП – об'єктоорієнтоване програмування

СУБД – система управління базами даних

					<i>КНТЕУ 121 02з-7.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Проєктування програмного забезпечення визначення академічного плагіату в роботах здобувачів освіти <i>Перелік умовних скорочень</i>			
Зав. каф.	Криворучко О. В.			29.09.20		<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник	Сашиньова М. В.			29.09.20		<i>ПС</i>	<i>2</i>	<i>49</i>
Гарант	Токар В. В.			29.09.20		<i>Факультет інформаційних технологій</i> <i>2 курс, 2м група</i>		
Розробив	Гудкова В. Є.			29.09.20				

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ТА ЗАСОБІВ ДЛЯ ВИЗНАЧЕННЯ ЗАПОЗИЧЕНЬ У ТЕКСТОВИХ ПРАЦЯХ.....	7
1.1 Опис особливостей галузей застосування систем перевірки на антиплагіат.....	7
1.1.1 Рекламні тексти та їх застосування для пошукової оптимізації SEO.....	7
1.1.2 Особливості аналізу на антиплагіат наукових публікацій у спеціалізованих журналах.....	10
1.1.3 Дослідження навчальних робіт здобувачів вищої освіти на наявність плагіату.....	12
1.2 Основні підходи до визначення текстових запозичень.....	13
1.2.1 Використання пошукових систем загального призначення.....	13
1.2.2 Робота зі спеціалізованими інформаційними системами відповідного призначення	15
1.2.3 Застосування готових систем-аналогів для пошуку плагіату, аналіз переваг та недоліків.....	19
1.3 Виокремлення невирішених проблем у засобах-аналогах та постановка задачі дослідження.....	20
1.4 Висновки по розділу.....	22
РОЗДІЛ 2 ОБҐРУНТУВАННЯ ПРОЄКТНИХ РІШЕНЬ ПРИ РОЗРОБЛЕННІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВИЗНАЧЕННЯ АКАДЕМІЧНОГО ПЛАГІАТУ В РОБОТАХ ЗДОБУВАЧІВ ОСВІТИ.....	23
2.1 Створення алгоритмічної складової проєктованого засобу.....	23

					<i>КНТЕУ 121 023-7.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата	Проектування програмного забезпечення визначення академічного плагіату в роботах здобувачів освіти	Стадія	Аркуш	Аркушів
Зав. каф.		Криворучко О. В.		22.12.20		3	3	49
Керівник		Сашньова М. В.		22.12.20		Факультет інформаційних технологій 2 курс, 2м група		
Гарант		Токар В. В.		22.12.20				
Розробив		Гудкова В. С.		22.12.20	Зміст			

2.2	Аналіз необхідності використання системи управління базами даних та проектування сховища даних.....	25
2.3	Обґрунтування вибору засобів розробки.....	28
2.3.1	Вибір технології програмування.....	28
2.3.2	Вибір мови програмування.....	31
2.3.3	Обґрунтування вибору засобів розробки.....	34
2.3.3.1	Вибір інтегрованого середовища розробки.....	34
2.3.3.2	Вибір інструментального засобу для збереження даних у системі.....	35
2.3.3.3	Додаткові засоби розробки.....	35
2.4.	Висновки по розділу.....	36
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВИЗНАЧЕННЯ АКАДЕМІЧНОГО ПЛАГІАТУ В РОБОТАХ ЗДОБУВАЧІВ ОСВІТИ.....		37
3.1	Прототипування інтерфейсу користувача програмного забезпечення.....	37
3.2	Особливості реалізації сховища даних продукту.....	39
3.3	Опис реалізації основних алгоритмічних складових продукту.....	40
3.4	Документування створеного ПЗ (розробка керівництва користувача).....	42
3.5	Тестування створеного програмного продукту та аналіз його ефективності...44	
3.6	Висновки по розділу.....	46
ВИСНОВКИ		47
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....		48

					КНТЕУ 121 023-7.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		4

ВСТУП

Бурхливий розвиток Інтернету вносить у наше життя чимало нововведень та багато нових можливостей. Нажаль, деякі з них не дуже чесні на руку люди можуть використовувати з метою омані інших та отримання власної вигоди. Саме так відбувається коли одна особа знаходить в Інтернеті текст, написаний іншою, і видає за власну працю. Із повсюдним впровадженням електронного документообігу та цифрової обробки інформації такі дії стають особливо простими у реалізації (уся робота відповідно зводиться до операцій «скопіювати»-«вставити»). Значного поширення така практика отримала у:

- навчальній діяльності (зокрема, при підготовці рефератів, курсових робіт, а іноді навіть і дипломних проєктів);
- науковій роботі (коли копіюються роботи закордонних науковців, що вимагає перекладу між мовами, або навіть із віддалених міст однієї і тієї ж країни);
- видавничій справі (коли хороші статті з маловідомих джерел передруковуються один-в-один на інших ресурсах навіть без будь-яких посилань на першоджерело);
- у наповненні текстовим контентом вебсайтів (де розробнику потрібно досить швидко десь узяти значні обсяги більш-менш якісної тематичної текстової інформації), тощо.

Однак, там де виникають нові загрози, там завжди породжуються і нові можливості, що справедливо і у випадку поширення доступу все більших мас населення до мережі Інтернет. Відповідно, для боротьби з вказаними ситуаціями було запроваджено перші системи пошуку плагіату або запозичень,

					КНТЕУ 121 023-7.МР			
Зм.	Аркуш	№ докум.	Підпис	Дата	Проектування програмного забезпечення визначення академічного плагіату в роботах здобувачів освіти	Стадія	Аркуш	Аркушів
Зав. каф.		Криворучко О. В.		27.02.21		В	5	49
Керівник		Сашньова М. В.		27.02.21		Факультет інформаційних технологій 2 курс, 2м група		
Гарант		Токар В. В.		27.02.21				
Розробив		Гудкова В. Є.		27.02.21	Вступ			

тобто текстових фрагментів, які уже були раніше опубліковані іншими людьми.

З використанням подібних систем боротьба з плагіатом вийшла на новий рівень, адже тепер доступним є порівняння тексту із матеріалами, розміщеними у величезних базах класичних творів (включаючи художні, технічні та інші джерела – що визначається предметною галуззю), останніх періодичних видань, та навіть особистих сторінок (щоденників типу блогів). Слід сказати, що існуючі системи мають свої особливості, які у певних умовах перетворюються на суттєві недоліки, тому розробка нових оригінальних систем боротьби з плагіатом є актуальною задачею для фахівця в галузі інформаційних технологій.

					КНТЕУ 121 023-7.МР	Аркуш
						6
Зм.	Аркуш	№ докум	Підпис	Дата		

Мета дослідження: підвищення якості нових текстів, зокрема робіт здобувачів вищої освіти, за рахунок зменшення частки запозичених матеріалів, чого можна досягти шляхом проєктування та розробки програмного забезпечення, яке могло би із достатньою точністю визначати наявний рівень оригінальності тексту.

Задачами роботи є:

- аналіз існуючих систем визначення плагіату, основних принципів їх роботи та важливих особливостей, виокремлення їх недоліків та формування технічного завдання на розробку, що нейтралізує дані недоліки;
- розробка алгоритмів та структур даних, які мають бути покладені в основу програмного забезпечення для визначення академічного плагіату;
- обґрунтування вибору засобів та технологій розробки та проведення реалізації розроблених алгоритмів у робочому програмному продукті;
- тестування розробленого програмного забезпечення, визначення його показників ефективності, виконання висновків по роботі та її перспектив.

Об'єкт дослідження – процес пошуку плагіату у текстових даних.

Предмет дослідження – методи та засоби для пошуку плагіату у текстових даних.

Практичне значення роботи полягає у розробці оригінального програмного продукту, за допомогою якого можна проводити аналіз текстів на предмет наявності у них текстових запозичень із загальнодоступних джерел інших авторів, розміщених у мережі Інтернет.

Методи дослідження: методи галузі технологій програмування та загальнонаукові методи аналізу та синтезу.

У *перспективі* програмний продукт може розвиватися шляхом залучення додаткових пошукових систем, окрім тих, що вже використані у роботі, а також за допомогою формування інших ознак для визначення авторства тексту, таких, що наприклад, авторський стиль людини, що писала текст.

					КНТЕУ 121 023-7.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		7

РОЗДІЛ 1

АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ТА ЗАСОБІВ ДЛЯ ВИЗНАЧЕННЯ ЗАПОЗИЧЕНЬ У ТЕКСТОВИХ ПРАЦЯХ

1.1. Опис особливостей галузей застосування систем перевірки на анти плагіат

В сучасному світі, де майже в кожному домі користуються послугами мережі Інтернет, а переважна більшість звітів, навчальних та наукових робіт виконуються в електронному цифровому вигляді, текстова інформація набуває нової цінності, а до її якості висувається чимало нових вимог, обумовлених створенням та впровадженням засобів автоматизованої обробки текстів. Одним із таких засобів автоматизованої перевірки є системи антиплагіату, робота яких спрямована на унеможливлення видачі чужого тексту за свій недобросовісними авторами (мається на увазі запозичення без згадування джерела, що у свою чергу називається цитуванням і не є забороненим прийомом у сучасній літературі). Слід сказати, що у роботі таких систем можуть бути виділені певні відмінності, які у значній мірі залежать від особливостей предметної галузі, для якої здійснюється перевірка, тому розглянемо їх докладніше.

1.1.1 Рекламні тексти та їх застосування для пошукової оптимізації SEO

Якщо подивитися на відносну частку джерел інформації, які можуть поставляти у суспільство нові тексти, то ще у 80-х роках XX століття це були книги, газети та журнали. Однак, у середині 90-х до масового користувача починає доходити мережа Інтернет і на початку 2000-х різко зростає число сайтів, які починають використовувати усі без виключення компанії, відомі фізичні особи, справжні прогресивні фахівці своєї справи (у різних галузях – починаючи від

					КНТЕУ 121 02з-7.МР			
Зм.	Аркуш	№ докум	Підпис	Дата				
Зав. каф.		Криворучко О. В.		16.04.21	Проектування програмного забезпечення визначення академічного плагіату в роботах здобувачів освіти	Стадія	Аркуш	Аркушів
Керівник		Сашньова М. В.		16.04.21		Р1	8	49
Гарант		Токар В. В.		16.04.21		Факультет інформаційних технологій 2 курс, 2м група		
Розробив		Гудкова В. Є.		16.04.21	Аналіз існуючих методів та засобів для визначення запозичень у текстових працях			

програмістів і закінчуючи сантехніками). Наприкінці 2010-х рр., коли соціальні мережі іще не вийшли на передній план мережі Інтернет чи не кожна домогосподарка прагнула отримати свій власний сайт. На сьогоднішній день власний вебсайт став обов'язковою компонентою будь-якого успішного бізнесу, а компанії, які його не мають, де-факто викликають підозру, чи не фіктивний бізнес вони ведуть.

Очевидно, що наповнення текстовим контентом вебсайтів має чимало різноманітних аспектів, однак дуже укрупнено, якщо слідувати концепції MVC (Model-View-Controller) можна сказати, що ефективний сайт складається із трьох компонентів [3]:

- смислового наповнення сайту, його семантики, змісту, тобто тієї інформації, яку він хоче донести до користувача (Model);
- зовнішнього вигляду сайту, тобто всього, що відноситься до дизайну, краси рішення (View);
- активної компоненти, «поведінки» сайту, тобто наскільки повноцінно і корисно він може реагувати на дії користувача на даному сайті (Controller).

Перша частина – змістовне наповнення, звичайно може забезпечуватися і графічною, і текстовою, і мультимедійною інформацією, однак такі варіанти є більше виключенням із правил, а при традиційному підході основна частина змістовного наповнення сайту відноситься саме до текстової інформації. При цьому, як відомо, усі користувачі приходять на сайт двома шляхами:

- за посиланнями з інших сайтів, причому частка таких користувачів залежить від того, наскільки активно сайт оплачує власну рекламу, і, якщо реклама не замовляється, то дана частка користувачів для переважної більшості сайтів автоматично наближається до нуля;
- із результатів видачі пошукових систем (Google та Yandex в першу чергу) – і це єдиний відносно безкоштовний спосіб отримання трафіку на сайт, тому він і є основним для більшості не сильно комерціалізованих ресурсів.

Ясно, що у кожній предметній галузі на сьогоднішній день існує чимало

					КНТЕУ 121 023-7.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		9

достатньо якісних вебресурсів, тому конкуренція є досить високою, а одним із основних показників, за рахунок яких сайт займає вищі позиції у результатах видачі пошукових систем, є унікальність та якість текстової інформації, яка на ньому присутня (і чим більше там унікального тематичного тексту, насиченого ключовими словами галузі, тим краще будуть результати сайту і тим більше відвідувачів він отримає).

Таким чином, ті, хто створює сайти, весь час потребують значних обсягів нової текстової інформації, що відноситься до їхніх товарів та послуг. Відповідно, виникла ціла нова професія – копірайтер (SEO-копірайтер, автор текстів), основним завданням якої є генерація текстів із заданими параметрами (тематика, стиль, обсяг, наявність запозичень, цитат і т.п.). Очевидно, що замовники повинні мати певні інструменти для перевірки якості отриманого тексту і основним із них є програмне забезпечення для визначення плагіату у ньому. Таким чином, програми для пошуку запозичень є основним інструментом контролю роботи SEO-копірайтера, що використовується при прийомці виконаних ним текстів, а також ним самим у процесі роботи над завданням [4].

Слід відмітити, що сучасним вебсайтам можуть бути притаманні досить широкі рамки тематики та різноманітні стилі. Наприклад, на ресурсах для молоді «18+» можуть допускатися (або навіть бути бажаними) жаргонні та нецензурні вислови, що очевидно не є припустимим у літературному або академічному стилях. Однак і такі тексти мають перевірятися на плагіат, тому що, якщо вони є вкраденими, то це значно зменшить позицію даного сайту у видачі пошукової машини серед інших аналогічних сайтів. Втім, часто копірайтери працюють і з досить серйозними тематиками, що потребують заглиблення у предметну галузь, що схоже із процесом написання навчальних робіт або навіть і наукових творів. Однак, в цілому аналіз текстів для розміщення на вебсайтах є більш широкою задачею, аніж аналіз навчальних робіт на наявність академічного плагіату. Це означає, що відповідні програмні продукти взагалі кажучи можна використовувати для задач пошуку академічного плагіату, однак ефективність їх буде нижчою, аніж

					КНТЕУ 121 023-7.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		10

при використанні спеціально призначеного для цього програмного забезпечення (ПЗ).

Як відомо, наявність зайвої функціональності у програми (такої, що ніколи не буде використана користувачем) є недоліком, що є причиною для вибору інших програмних продуктів, які мають аналогічну затребувану функціональність та мінімум зайвої. Зайва функціональність – це завжди надмірно використані ресурси персонального комп'ютера (ПК), яких ніколи не буває забагато, і які можна було б виділити під інші, дійсно корисні програми.

1.1.2 Особливості аналізу на антиплагіат наукових публікацій у спеціалізованих журналах

Ще однією галуззю, де є важливим пошук на антиплагіат, є наукові публікації, адже нажалі іноді трапляються випадки, коли роботи маловідомих вчених передруковуються як власні. Також усі автори іноді (навіть і не свідомо) можуть допускати самоплагіат, що їм самим може здаватися природним, але для наукових журналів є не дуже гарною ситуацією, особливо якщо першоджерело є прямим конкурентом, як це часто буває серед передових видавництв.

Пошук у науковій сфері сильно відрізняється від попереднього випадку, в першу чергу тим, що наукові публікації часто розмішують у спеціалізованих форматах типу Portable Document Format (PDF), PostScript (PS), TeX. У ручному режимі це можна зробити за допомогою розширеного пошуку Google – рис. 1.1. В принципі, якщо вручну перебрати значну кількість посилань, що їх видає пошукова машина за основними ключовими словами наукового тексту, то професіонал у даній галузі зможе встановити, чи є схожою запропонована стаття на інші, уже опубліковані роботи. Однак, очевидно, що така робота потребуватиме дуже великих затрат робочого часу і має автоматизуватися за допомогою відповідного програмного забезпечення.

					КНТЕУ 121 023-7.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		11

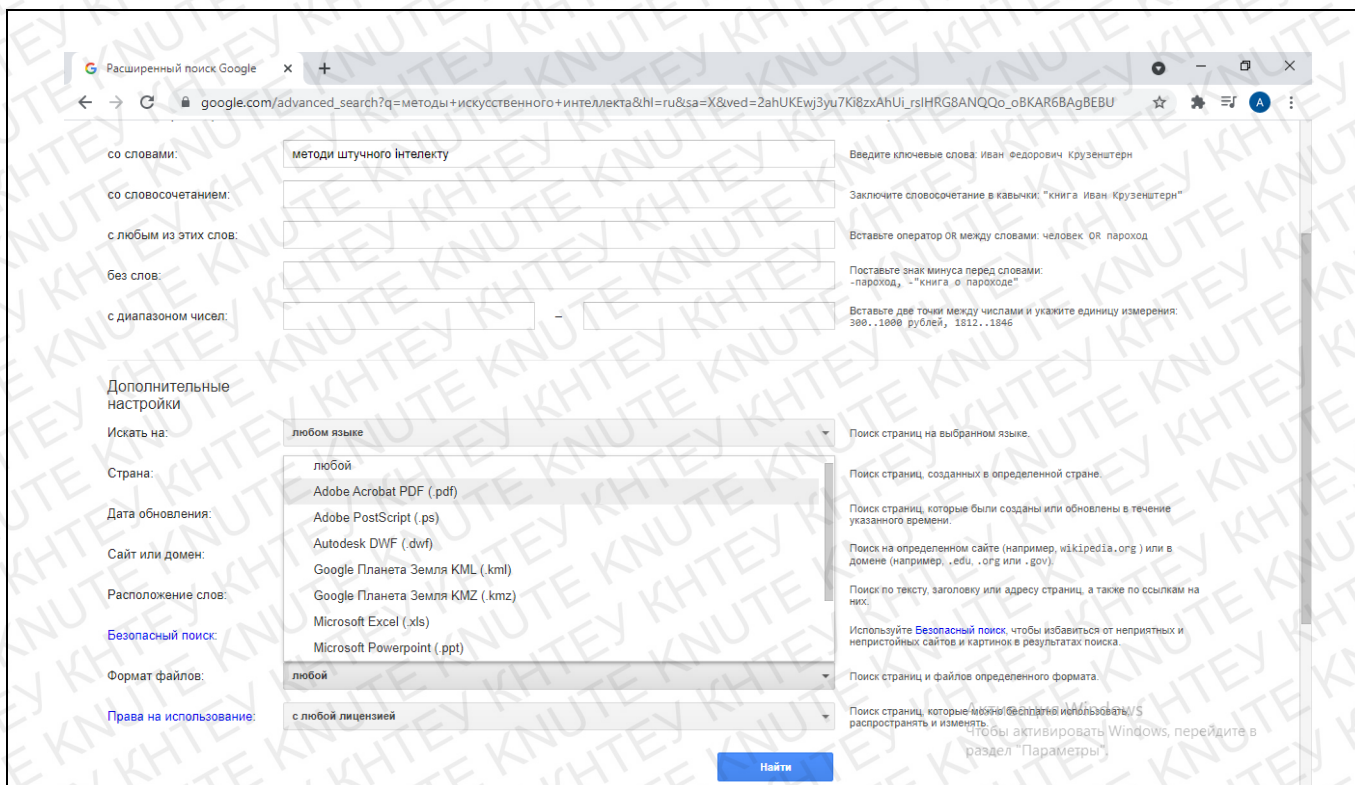


Рис. 1.1 Розширений пошук наукової інформації в системі Google по форматам PDF та PostScript

При аналізі наукових текстів доцільно використовувати спеціалізовані словники, оскільки велика кількість побутових слів не може бути використана у наукових публікаціях (особливо не літературно-мовного, а технічного напрямку), і в той же час лексика наукових публікацій значно розширена у бік використання спеціалізованих термінів, що і взагалі можуть бути незнайомими широкому загалу.

Також при аналізі наукових текстів допускається значно більша довжина речень (що має враховуватися і при пошуку запозичень) та перевантаження прикметниками та іменниками, що можуть розміщуватися багато разів підряд.

В цілому можна сказати, що наукові тексти набагато ближчі до текстів навчальних робіт, аніж продукція SEO-копірайтерів, а у випадку магістерських дисертацій взагалі є майже ідентичними за параметрами, тому системи пошуку плагіату у наукових текстах добре пристосовані і до роботи з текстами здобувачів

					КНТЕУ 121 023-7.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		12

вищої освіти. Конкретні приклади таких систем будуть розглянуті нижче докладніше.

1.1.3 Дослідження навчальних робіт здобувачів вищої освіти на наявність плагіату

Нарешті, аналіз текстів, що пропонуються здобувачами вищої освіти, є основним об'єктом дослідження у даній роботі. Даний процес має багато спільного із описом аналізу наукових текстів у попередньому пункті, але повинен включати також і пошук по веб-ресурсам у форматі HTML, а не тільки строгих форматах PDF (PS, TeX). Справа в тому, що на сьогоднішній день в Інтернеті доступно чимало бірж рефератів, де у вільному доступі можна продивитися роботи студентів попередніх років, які добровільно були виставлені ними на широкий загал (можливо із наданням певних незначних винагород). Усі ці роботи розміщені у звичайному форматі мови розмітки гіпертексту і, зважаючи на вище наведене у п.1.1.1, сприяють просуванню даної біржі у результатах видачі пошукової системи по словам «реферат» та аналогічних.

Таким чином, аналіз студентських робіт на академічний плагіат характеризується наступними рисами:

- повинен здійснюватися у всіх форматах (HTML, PDF, PS, DOC);
- не повинен включати побутові слова, жаргон, нецензурну лексику, метафори, художні засоби прикрашення тексту (якщо звичайно це не навчальна праця з літератури) та ін.;
- повинен включати довгі послідовності ключових слів (оскільки сам текст містить довгі речення);
- має базуватися на дослівному пошуку (який при ручному пошуку забезпечується взяттям пошукового запиту в лапки, а при автоматизованому – шляхом аналогічного формування запиту GET: [https://www.google.com/search?q="](https://www.google.com/search?q=)точний збіг слів").

					КНТЕУ 121 023-7.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		13

1.2 Основні підходи до визначення текстових запозичень

1.2.1 Використання пошукових систем загального призначення

Очевидним шляхом визначення того, що текст уже опублікований в мережі Інтернет, є використання пошукових машин типу Google, основною функціональністю яких якраз і є пошук заданих текстових рядків. Отже, говорячи простими словами, у найпростішому випадку, який не потребує взагалі жодної додаткової розробки, людина у ручному режимі може копіювати із роботи, що перевіряється, відносно довгі фрагменти (від 5 слів), вставляти їх у відповідне поле пошукової системи та аналізувати перші N результатів видачі, поступово відкриваючи їх одне за одним і порівнюючи з заданим текстом.

Загальновідомо, що на сьогодні існують і інші пошукові системи, окрім Google (яка є найбільш популярною у всіх без виключення країнах), які втім мають незначні долі ринку (дані наведено на 2021 рік з сервісу Statcounter [2]):

- Microsoft Bing посідає друге місце серед усіх пошукових систем і займає від 3% до 12% долі ринку по різних країнах;

- Yahoo – пошукова система, яка була значно популярнішою на початку 2000-х, а на сьогоднішній день займає біля 1% ринку по всьому світу із незначним випередженням у західних країнах та певним відставанням у східних;

- Baidu є найпопулярнішою пошуковою системою у Китаї, за рахунок великої кількості жителів якого і займає 4 місце у світовому рейтингу із близько 1% ринку (зі значним випередженням у Китаї, де її доля наближається до 12%);

- Yandex, що займає 5 місце у світі із долею близькою до 0,5%, в основному через дуже високу популярність у Росії. Як відомо, дана система знаходиться в Україні під санкціями, тому її використання у подальшій роботі розглядати не будемо.

Як бачимо, навіть друга по популярності у світі система Bing має настільки низькі рейтинги у порівнянні з Google, що без особливих втрат загальності можна

					КНТЕУ 121 023-7.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		14

усі альтернативні системи не розглядати, а сконцентруватися на роботі виключно з пошуковою машиною Google.

Слід відмітити, що робота у ручному режимі має наступні недоліки:

- її ефективність (тобто факт знаходження плагіату у тексті, якщо він там дійсно є) сильно залежить від кваліфікації особи, яка виконує аналіз (тобто, навіть якщо звичайна людина дивиться на два майже ідентичних тексти, один з яких був дещо модифікований по формі, то встановити факт плагіату їй буде досить важко, і для цього потрібно бути висококваліфікованим спеціалістом);

- для різних предметних галузей потрібні різні спеціалісти, тобто вказаний спосіб перевірки (якщо, наприклад, його реалізує наукове видавництво чи журнал) вимагає залучення значної кількості різних за профілем фахівців;

- вартість робочого часу висококваліфікованих вчених є дуже високою, тому такий спосіб перевірки є досить дорогим;

- продуктивність перевірки навіть при залученні найкращих спеціалістів не може бути високою через обмежені можливості людського організму;

- завжди існує не дуже мала імовірність, що людина може пропустити одне з багатьох джерел (адже при ручній рутинній праці часто виникають помилки), або не дійти до нього, рано зупинившись у процесі пошуку. Навіть при суворому дотриманні інструкцій людина може продивитися перші N джерел, а текстовий збіг буде знайдено у $N+1$ випадку. Ясно, що боротися з цією можливістю можна шляхом збільшення числа N , що легко зробити при автоматизованому пошуку, але важко (через суттєве подорожчання всього процесу) при ручній праці;

- пошукові системи загального призначення видають чимало непотрібної інформації, яка може бути схожою по ключовим словам із текстом, що аналізується, але розміщена у блогах (і її відповідно не потрібно аналізувати), художніх текстах, розважальних ресурсах і т.п. Для економії такого дорогого часу особи, що проводить дослідження, усі такі посилання не слід відкривати та аналізувати. Пошук потрібно здійснювати в основному по науковій інформації, що

					КНТЕУ 121 023-7.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		15

не враховують загальні системи пошуку, але може бути враховано при використанні спеціалізованих систем, початково призначених для науковців.

1.2.2 Робота зі спеціалізованими інформаційними системами відповідного призначення (сайти бібліотек, наукових спільнот, видавництв, тощо)

Більш кращі можливості для пошуку академічного плагіату в роботах здобувачів освіти надають спеціалізовані системи на окремих сайтах, що включають наукові праці. Це можуть бути наступні ресурси:

а) сайти крупних бібліотек, як, наприклад, Національної бібліотеки України імені В.І. Вернадського – рис. 1.2, російської державної бібліотеки, наукової електронної бібліотеки «КіберЛенінка» - рис.1.3 (сайт не знаходиться під санкціями, тому доступний для використання в Україні), тощо.

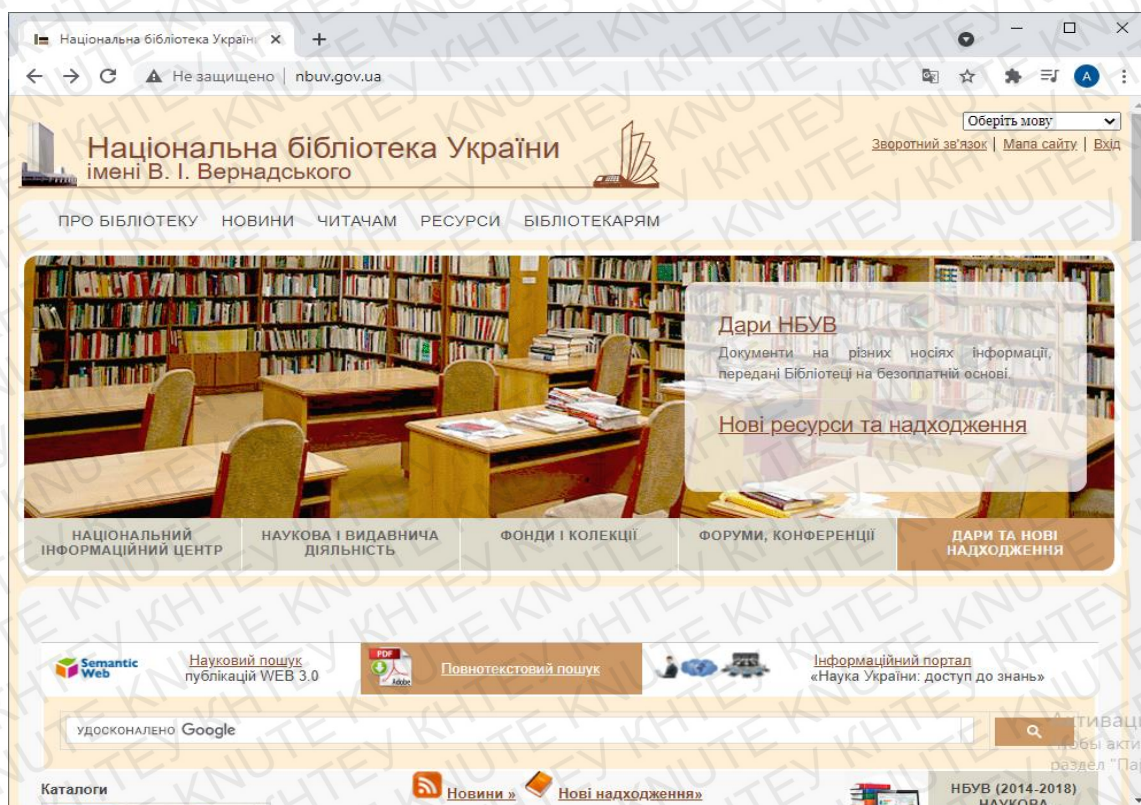


Рис. 1.2. Сайт Національної бібліотеки України ім. В.І. Вернадського, який надає відмінні можливості з пошуку текстів наукового напрямку (розосереджені по різним екранам, а тут наведено посилання на них)

					КНТЕУ 121 023-7.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		16

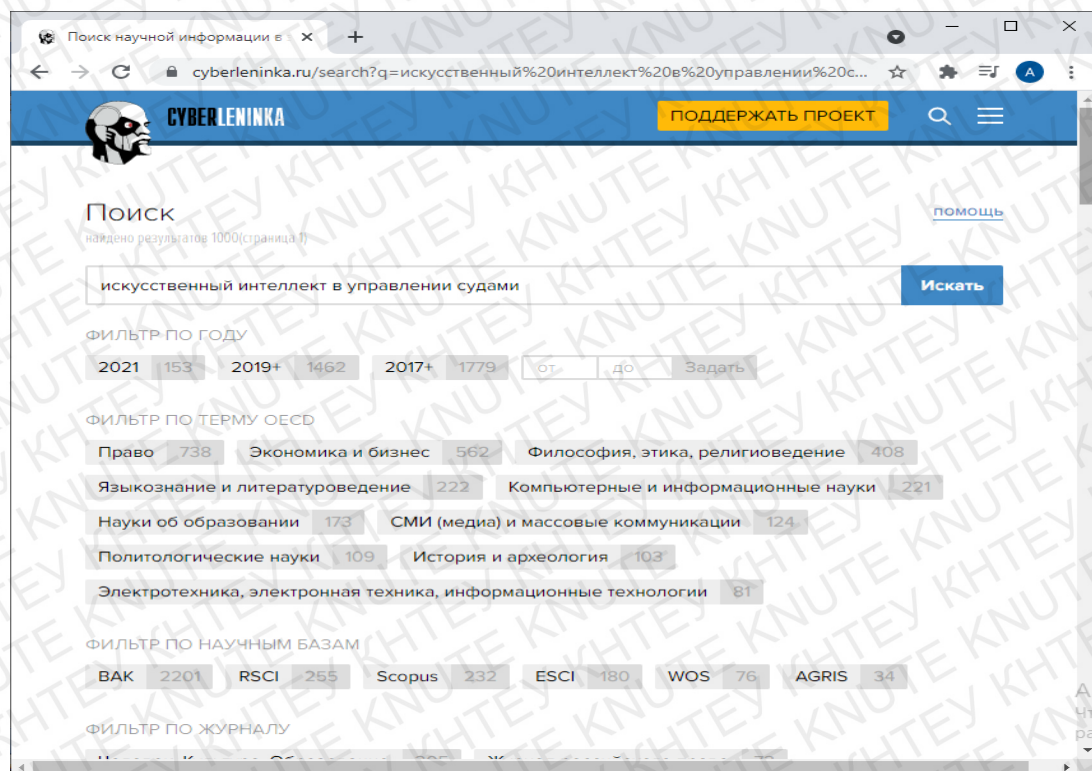


Рис. 1.3. Форма пошуку електронної бібліотеки «Кіберленінка»

Як видно з рисунків, а також, що підтверджується практикою їх використання, інтерфейс пошукового движка сучасних електронних бібліотек є досить потужним і дозволяє проводити досить ефективні пошуки для визначення запозичених текстових блоків.

б) загальносвітові наукові індексовані (наукометричні) бази даних, такі як Scopus, WebOfScience, Google Scholar, ПИНЦ, Index Copernicus, Ulrich's Periodicals Directory, Directory of Open Access Journals, SCImago Journal & Country Rank, тощо. Окрім своєї прямої функції – оцінювати наукові здобутки окремих осіб, колективів, і т.д. – вони дозволяють шукати самі наукові праці, переглядати їх тексти. Суттєвим недоліком цих систем є те, що вони надають свої послуги на платній основі, що, ймовірно, можливо для галузі високої науки де задіяні великі гроші, але абсолютно не припустимо для галузі навчання.

в) інші бази даних, пов'язані із наукою та навчанням, зокрема бази патентних документів, яскравим прикладом яких є система Укрпатенту – рис. 1.4

					КНТЕУ 121 023-7.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		17

Спеціалізована БД "Винаходи" x +

base.uip.org/searchINV/

Відомості про видачу патентів України на винаходи (корисні моделі)

Пошук

Останні результати пошуку (0)

Кошик (0)

Допомога

ліст адміністратору

Пошук

Записів на сторінці 10

Актуалізовано 2021-07-01 02:06:03

Шукати в знайденому

Ключові слова в назві (uk) велосипед ТА причіп

(10) Вид та номер охоронного документа СРСР 1314963

(11) Номер патенту 56

(20) Дата подання заявки, раніше поданої в патентне відомство СРСР 19.03.1993

(20) Номер заявки, раніше поданої в патентне відомство СРСР 93101140

(21) Номер заявки u200804180

(22) Дата подання заявки 15.01.2004

(24) Дата, з якої є чинними права 17.06.2002

(31) Номер попередньої заявки 60/171.906

(32) Дата подання попередньої заявки 22.05.2008

(33) Двобуквенний код держави UA

(46) Дата публікації відомостей про видачу патенту 15.08.2005

(46) Номер бюлетеня 11/2006

(51) Індекс МПК A01B 13/08

(56) Аналоги винаходу SU 716711

(86) Номер заявки РСТ РСТ/CN03/00069

(86) Дата подання заявки РСТ 01.12.2004

(71) Заявник Семенов Семен Семенович

(72) Винахідник Швець Іван Петрович

Хоріков Сергій

Акції
Чтобы
раздел

Рис. 1.4. Пошукова форма Укрпатенту, що дозволяє вести ефективний аналіз текстових запозичень із патентів на винаходи та корисні моделі

Слід відмітити, що у всіх таких системах пошук також ведеться у ручному режимі, що спричинює усі ті ж самі недоліки, що вже були озвучені вище у пункті 1.2.1. Очевидно, що представляє інтерес автоматизація пошуку у розглянутих спеціалізованих системах, що може бути проведено шляхом надсилання автоматичних, належним чином сформованих запитів типу GET, які здійснюються по протоколу HTTP, з подальшим аналізом вебсторінок, надісланих у відповідь цими системами (процес автоматичного аналізу тіла вебсторінки із виділенням необхідної інформації називається парсінгом і може виконуватися за кількома різними схемами, про що докладніше буде говоритися у подальшому викладі).

						Аркуш
					КНТЕУ 121 023-7.МР	18
Зм.	Аркуш	№ докум	Підпис	Дата		

Взагалі, повністю автоматичний аналіз текстів уже реалізовано в рамках програм відповідного класу, які розглянемо докладніше у наступному пункті.

1.2.3 Застосування готових систем-аналогів для пошуку плагіату; аналіз їх переваг та недоліків

На сьогоднішній день існує значна кількість пропозицій на ринку систем аналізу текстів на запозичення, однак усі вони мають свої особливості, декілька з яких розглянемо більш докладно.

Система Unichesk є міжнародно визнаною і надає високий рівень сервісу та багато опцій для пошуку/аналізу, однак має суттєвий недолік – перевірка є платною (окрім смішних обсягів тексту, які не можна розглядати при серйозній роботі) і за 20 сторінок студентові пропонується заплатити 5 доларів США, що є абсолютно неприпустимим. Жодних пільг чи безкоштовних планів система не має, тому в подальшому її розглядати не будемо.

Найбільш якісним та поширеним на сьогоднішній день рішенням є повністю безкоштовна програма ETXT AntiPlagiarism.NET – рис. 1.5.

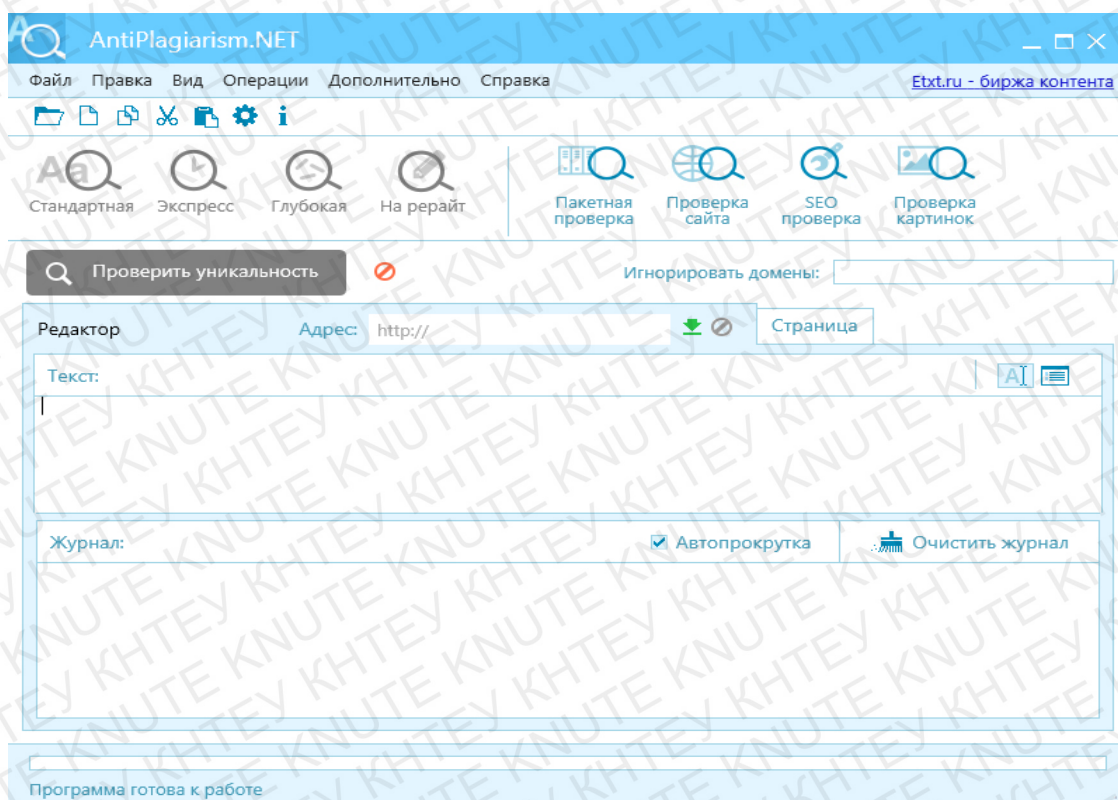


Рис. 1.5. Зовнішній вигляд вікна програми ETXT AntiPlagiarism.NET

					КНТЕУ 121 023-7.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		19

Особливістю програми є проведення процесу перевірки не на віддаленому сервері, що характерно для більшості аналогічних систем, а на комп'ютері користувача, що висуває певні вимоги до його потужності, а головне, до якості Інтернет-з'єднання. Також через це весь процес перевірки займає тривалий час (іноді – до півгодини для великих робіт на 100 сторінок). Такі особливості в окремих випадках (наприклад, нестабільного Інтернет-з'єднання, що періодично зникає) робить перевірку за допомогою продукту ETXT взагалі неможливою.

Також програма AntiPlagiarism.NET не завжди коректно обробляє файли з розширенням PDF, що якраз є суттєвим недоліком для аналізу академічних текстів (через причини, розглянуті вище).

Ще однією особливістю, яку при використанні виключно в рамках предметної галузі, що розглядається (аналіз робіт здобувачів вищої освіти), можна назвати недоліком, є наявна велика кількість опцій у програмі, які є непотрібними при аналіз виключно навчальних робіт. Як уже зазначалося вище, зайва функціональність є небажаною, оскільки ускладнює інтерфейс програми (а, отже, зменшує продуктивність роботи користувача із нею) та збільшує ресурси комп'ютера.

1.3 Виокремлення невирішених проблем у засобах-аналогах та постановка задачі дослідження

Таким чином, у попередньому розділі розглянуто можливі способи визначення плагіату в роботах здобувачів вищої освіти. Встановлено наступні недоліки таких способів:

- при використанні пошукових машин загального призначення виникає значна частка зайвої інформації, на яку може витратитися чимало часу при ручному способі аналізу;
- ручний спосіб характеризується дуже малою продуктивністю та високою вартістю всього процесу;

					КНТЕУ І21 023-7.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		20

- при використанні спеціалізованих програм переважна більшість із них є платними;

- існуючі безкоштовні програмні продукти перевантажені опціями та, відповідно, мають складний інтерфейс (що суттєво при використанні спеціалістами, далекими від галузі ІТ), а також потребують значних ресурсів комп'ютера та високоякісного каналу доступу до Інтернет.

Зважаючи на встановлені недоліки, доцільно перейти до розробки власного програмного продукту, що здійснював би пошук академічного плагіату в роботах здобувачів вищої освіти. Встановимо основні параметри відповідної розробки.

Тип продукту: програма для обробки текстової інформації.

Архітектура продукту: настільний програмний продукт.

Назва продукту: AntiplagiatEdu.

Призначення продукту: пошук текстових запозичень із опублікованої в Інтернет інформації у роботах здобувачів вищої освіти.

Цільова програмно-апаратна платформа: персональні комп'ютери, ноутбуки та інші апаратні рішення, що працюють під управлінням ОС Windows.

Цільова операційна система: ОС Windows, версії не нижче 7.

Сховище даних: найпростіше з точки зору системних вимог.

Мінімальні системні вимоги:

- процесор 1 ГГц;
- оперативна пам'ять 1 Гб;
- 50 Мб на диску;
- екран 800x600 пікселів.

Функціональність:

- можливість завантаження у програму текстових документів;
- пошук текстових запозичень за допомогою популярної пошукової системи Google;
- перегляд результатів визначення унікальності у зручній для користувача формі.

					КНТЕУ 121 023-7.МР	Аркуш
						21
Зм.	Аркуш	№ докум	Підпис	Дата		

Програма повинна мати зручний графічний інтерфейс, стандартний для пристроїв на базі ОС Windows.

Розробка програмного продукту ведеться в рамках написання кваліфікаційного проєкту магістра і буде захищатися під час представлення результатів її створення.

Керуючись даними вимогами можна переходити до процесу проєктування програмного продукту, вибору засобів розробки та проведення його програмної реалізації.

1.4 Висновки по розділу

Таким чином, у даному розділі розглянуто особливості предметної галузі аналізу текстової інформації на предмет запозичень із інших джерел, що вже були опубліковані раніше іншими авторами (тобто плагіату, якщо посилання на першоджерело немає). Розглянуто сфери застосування продуктів для боротьби з плагіатом і виявлено їх особливості. Розглянуто існуючі методи та засоби для перевірки текстів на плагіат, виявлено їх недоліки та, відповідно, встановлено доцільність власної розробки. Зафіксовано основні вимоги до відповідного програмного продукту.

					КНТЕУ 121 023-7.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		22

РОЗДІЛ 2

ОБГРУНТУВАННЯ ПРОЄКТНИХ РІШЕНЬ ПРИ РОЗРОБЛЕННІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВИЗНАЧЕННЯ АКАДЕМІЧНОГО ПЛАГІАТУ В РОБОТАХ ЗДОБУВАЧІВ ОСВІТИ

2.1 Створення алгоритмічної складової проєктованого засобу

У попередньому розділі встановлена доцільність власної розробки програмного продукту для аналізу текстів на наявність плагіату. Очевидно, що цей процес не є надзвичайно простим, а тому перед програмною реалізацією слід розглянути його алгоритм. Коротко він може бути виражений наступними кроками – рис. 2.1.

1. Завантаження у програму текстового файлу для аналізу.
2. Розбивка тексту на окремі невеликі фрагменти (шингли, як їх називають інші автори схожого програмного забезпечення), по яким буде здійснюватися пошук. Кількість слів у фрагменті має складати біля 5.
3. Для кожного шинглу слід виконати пошук по точним збігам у пошуковій машині (Google – відповідно до завдання на розробку).
4. Сторінку виводу результатів пошуку слід пропарсити і дістати з неї результати пошуку, якщо точні збіги знайдено.
5. Для кожного знову знайденого точного збігу розпочати процедуру визначення його масштабів, тобто пошук послідовних збігів із цим же джерелом наступних шинглів у тексті, що аналізується. Продовжувати такий вкладений пошук по одному даному джерелу поки не буде знайдено розходження і встановити який відсоток від тексту, що аналізується, складає знайдений збіг.
6. Якщо по даному шинглу є ще точні збіги, то перейти до процедури пошуку

					<i>КНТЕУ 121 023-7.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О. В.		24.05.21	Проектування програмного забезпечення визначення академічного плагіату в роботах здобувачів освіти	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник		Сашньова М. В.		24.05.21		<i>P2</i>	<i>23</i>	<i>49</i>
Гарант		Токар В. В.		24.05.21		Факультет інформаційних технологій 2 курс, 2м група		
Розробив		Гудкова В. Є.		24.05.21				
					Обґрунтування проєктних рішень при розробленні програмного забезпечення визначення академічного плагіату в роботах здобувачів освіти			

– п.5, а якщо усі точні збіги отримані на етапі 4 опрацьовано, то перейти до наступного шинглу оригінального тексту, тобто до п. 3.

7. Кінцево видати результати знайдених збігів на екран списком та підрахувати результуючу оригінальність тексту у відсотках (єдиним числом).



Рис. 2.1. Схема алгоритму аналізу оригінальності тексту

					КНТЕУ 121 023-7.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		24

8. Завершити виконання процесу.

Якщо говорити про представлення проектного рішення за допомогою мови уніфікованого моделювання UML, то діаграма варіантів використання в даному випадку є надзвичайно простою і складається всього лише з одного прецеденту – рис. 2.2.

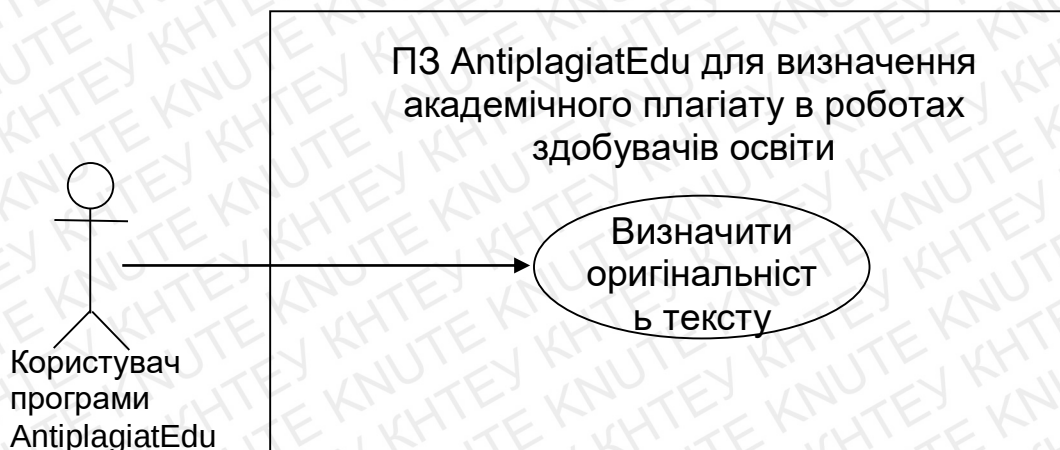


Рис. 2.2. Діаграма прецедентів (варіантів використання, Use Case Diagram) для програмного продукту, що розробляється

Для вказаного варіанту використання діаграма діяльності буде мати вид рис. 2.3.

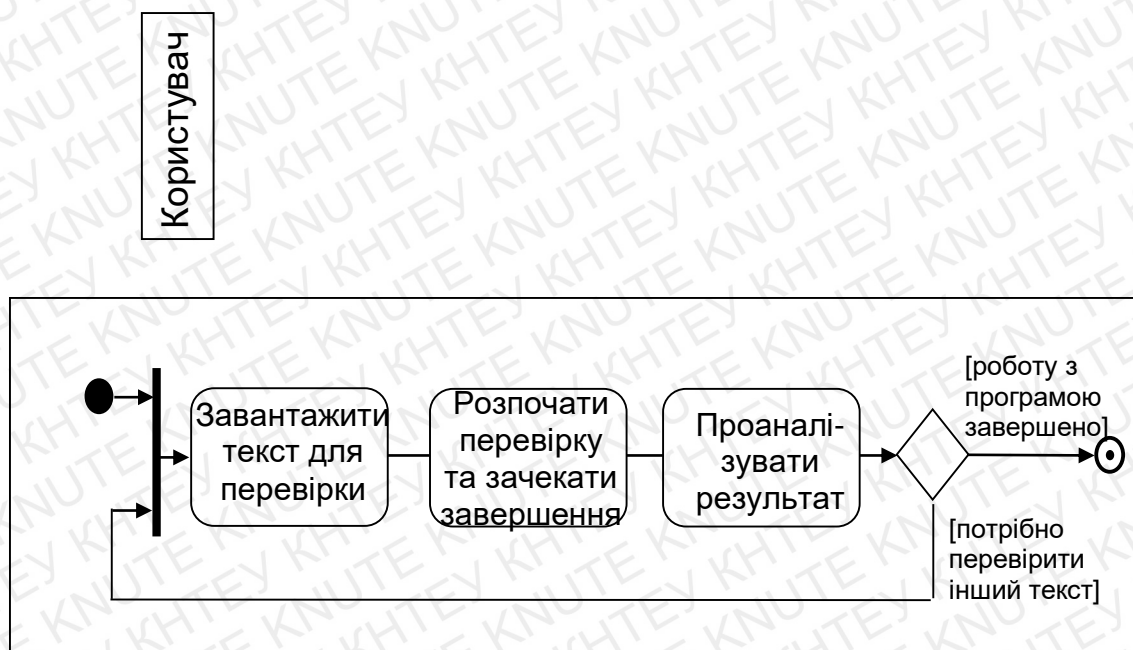


Рис. 2.3. Діаграма діяльності (Activity Diagram) для варіанту використання «Визначити оригінальність тексту»

					КНТЕУ 121 023-7.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		25

Керуючись даними алгоритмічними розробками, можна проводити програмну реалізацію процесу пошуку текстових запозичень, деталізуючи окремі його етапи уже безпосередньо у програмному коді.

2.2 Аналіз необхідності використання системи управління базами даних та проєктування сховища даних

Із вищенаведеної інформації можна зробити висновок, що сам процес пошуку текстових запозичень не є алгоритмічно простим, а відповідне програмне забезпечення має досить високий рівень складності. Як відомо, часто комплексні програмні продукти використовують сховища даних у вигляді систем управління базами даних, тому питання необхідності їх використання у даній роботі також є важливим і підлягає розгляду у даному підрозділі.

Усі СУБД можна розділити на реляційні, які дуже широко використовуються для переважної більшості стандартних програмних продуктів, а також бази даних NoSQL, які в основному використовуються в галузі штучного інтелекту, роботи із великими даними BigData, високо навантаженими системами, розподіленими програмними продуктами та іншим спеціалізованим програмним забезпеченням. Незважаючи на те, що пошук плагіату у текстах не є дуже простою задачею, все ж таки вона не має перелічених особливостей і тому використання особливих NoSQL-рішень тут не є потрібним.

Що ж стосується реляційних баз даних, то у них традиційно створюється таблична структура, куди записуються численні однотипні рядки інформації, які в подальшому будуть використовуватися для роботи програми. У випадку, що розглядається, кожний текстовий пошук є унікальним, оскільки різними є тексти, які підлягають аналізу. При цьому немає абсолютно ніякого сенсу зберігати у локальній базі даних результати пошуку по набору із, припустимо, $N = 5$ конкретних слів, оскільки імовірність повторення саме такої комбінації у наступних текстах є надзвичайно малою, а при зміні у ній хоча б одного слова результати пошуку уже будуть іншими і локально збережені результати для старої комбінації не знадобляться.

					КНТЕУ 121 023-7.МР	Аркуш
						26
Зм.	Аркуш	№ докум	Підпис	Дата		

Таким чином, як висновок можна сказати, що зберігати складні структуровані масиви даних на комп'ютері, де буде встановлена програма для аналізу текстів на плагіат, не є доцільним.

2.3 Обґрунтування вибору засобів розробки

2.3.1 Вибір технології програмування

Першим кроком при виборі методів програмування є вибір технології розробки. Такими, що на сьогодні активно використовуються, можна назвати структурний підхід до програмування та об'єктно-орієнтовану парадигму.

Суть технології, або як ще кажуть, методології об'єктно-орієнтованого програмування - в тому, що система розглядається, як сукупність окремих сутностей - об'єктів, які мають набір якихось своїх внутрішніх параметрів - властивостей, а також можуть взаємодіяти між собою за допомогою деяких дій - викликів методів (або трохи більше непрямим чином - шляхом надсилання повідомлень, оброблюваних методами об'єктів; для цього необхідна присутність активної сутності, яка роздає повідомлення адресатам, як, наприклад, менеджер вікон у більшості багатозадачних операційних систем (ОС)).

Якщо говорити про програмний код, то для того, щоб оперувати об'єктом, його спочатку потрібно створити. Об'єкти створюються як змінні, у яких типом виступає клас об'єкта. Клас - це просто опис, які властивості можуть мати об'єкти такого типу (тобто яку інформацію вони можуть зберігати), і які у них є методи (тобто які дії вони можуть виконувати). Об'єкт - це набір значень, чому саме рівні властивості даного об'єкта (свої методи кожен об'єкт отримує від свого класу, тобто методи однакові у всіх об'єктів, що належать даному класу).

Для чого потрібен цей специфічний підхід, адже самі по собі об'єкти не додають нічого корисного (навпаки, введення об'єктів ускладнює програму, вносить в неї нові сутності). Виявляється, реалізуючи всі сутності, необхідні, згідно з алгоритмом, для роботи програми, у вигляді класів і об'єктів, ми спрощуємо її розуміння для самих себе. Саме тому об'єктно-орієнтований підхід рекомендується до застосування для великих проєктів (більше десятків тисяч

					КНТЕУ 121 023-7.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		27

рядків коду), коли утримувати «в голові» всю систему цілком стає важко. Можна сказати, що розбиття (декомпозиція) програми на об'єкти і проєктування їх класів наближає розуміння предметної області до звичного людського образу мислення (в разі «великих» проєктів). Людина мислить класами, об'єктами і зв'язками між ними.

Крім розбиття (декомпозиції) всієї предметної області на об'єкти (класи) і співвідношення між ними, також ОО-підхід має на увазі дотримання трьох основних його принципів: інкапсуляція, наслідування, поліморфізм.

Під інкапсуляцією мається на увазі об'єднання даних (значення властивостей класу у деякого конкретного об'єкта) та засобів їх обробки (методи класу). Це знову ж таки зручно психологічно, так як дозволяє реалізовувати окремі завершені сутності - класи, які самі обробляють свої дані. Звернення до об'єктів цих класів відбувається за допомогою методів, що утворюють інтерфейс класу.

Наслідування дуже корисно, тому що дозволяє сильно скоротити обсяги повторюваного коду (до чого потрібно завжди прагнути при розробці будь-якого програмного забезпечення). Згідно з цим принципом виділяється клас, який має загальний набір властивостей і методів для декількох більш розширених класів. Цей клас оголошується батьком, базовим класом для декількох похідних від нього (нащадків, спадкоємців). Всі класи-нащадки успадковують від базового всі його властивості та методи, але до цього ще мають свої власні оригінальні властивості і / або методи.

Наприклад, клас Студент є похідним від класу Людина, тому що кожна людина має властивість Ім'я, Прізвище, метод Відпочити(). Однак у Студента є свої специфічні властивості і / або методи, які як раз і відрізняють його від просто Людини: СереднійБал, НомерЗаліковки, ЗдатиЕкзамен(), і т.д.

При наслідуванні іноді методи батьківського і похідного класу мають однакове призначення, але реалізуються по-різному. Такі методи називаються перевантаженими. Наприклад, метод Відпочити() у класу Людина реалізується як відпочинок на дивані, а у класу Студент - як похід в клуб. При цьому ще раз

					КНТЕУ 121 023-7.МР	Аркуш
						28
Зм.	Аркуш	№ докум	Підпис	Дата		

підкреслимо, що призначення методу в обох випадках одне і той саме.

Поліморфізм є можливістю деякої функції приймати об'єкти як батьківського, так і похідних класів, і вміти викликати перевантажені методи саме того класу, об'єкт якого був переданий в функцію. Слід сказати, що це досить специфічна можливість і в загальному багато програмістів використовують ОО-підхід і без звернення до поліморфізму.

Нехай, наприклад, в нашій програмі є функція ПровестиВихідні(), припустимо яка не належить якомусь класу (хоча це не принципово). Нехай аргументом цієї функції є об'єкт класу Людина. Тоді в неї можна передавати об'єкти всіх похідних від Людини класів: Студент, Службовець, Пенсіонер, і т.д., тому що всі вони є Людиною (спадкоємці цього класу). Ясно, що ця функція повинна включати різні дії: ПрибратиКвартиру(), ПітиНаРинок(), і в тому числі Відпочити(). Так ось поліморфізм дозволяє всередині цієї функції просто вказати назву методу Відпочити(), не вказуючи якого саме класу він повинен бути викликаний, а вже в процесі виконання програми, якщо в функцію переданий об'єкт класу Студент, то викликається саме його метод Відпочити(), а якщо переданий об'єкт класу Пенсіонер, то автоматично викликається саме його метод Відпочити(), і т.д. Кажуть, що функція ПровестиВихідні() - поліморфна, і вона є такою завдяки тому, що реалізує принцип поліморфізму.

Важливими поняттями в ООП також є: статичні члени класу, абстрактні методи і класи, дружба функцій і класів, і т.д.

Таким чином, будемо використовувати об'єктно-орієнтований підхід як більш сучасний (тобто такий, що краще відповідає віянням у світі програмування) і такий, що підтримується майже всіма сучасними мовами програмування, зокрема, мовою Java.

2.3.2 Вибір мови програмування

У роботі для проведення програмної реалізації буде використовуватися сучасна повністю об'єктно-орієнтована мова програмування Java, особливості якої слід розглянути докладніше.

					КНТЕУ 121 023-7.МР	Аркуш
						29
Зм.	Аркуш	№ докум	Підпис	Дата		

Мову Java розробила компанія Sun Microsystems на початку 90-х років XX століття. Провідну роль у створенні мови зіграв канадський інженер Джеймс Гослінг (James Gosling). На ранніх етапах розробки мова називалася Oak, а потім її перейменували на честь сорту кави Java (зв'язок мови з напоєм відбивається також у її логотипі).

Джеймс Гослінг і його однодумці хотіли створити мову з С-подібним синтаксисом. У той же час він повинен бути простішим у порівнянні з С/С++. Творці планували використовувати Java для програмування побутової електроніки, проте практично відразу після випуску версії 1.0 в 1995 мову почали використовувати розробники серверного та клієнтського ПЗ. У 2010 році компанію Sun Microsystems купила Oracle, яка, відповідно, на даний час і займається розвитком цієї платформи (деякі прихильники вільного програмного забезпечення звинувачують Oracle у певній дискримінаційній політиці по відношенню до мови Java, через яку вона поступово втрачає позиції у деяких сегментах галузі розробки програмного забезпечення (зокрема, якраз при розробці мобільних додатків власниця ОС Android корпорація Google все інтенсивніше рекомендує мову Kotlin російської розробки, а не більш звичну Java). Тим не менше, на даний час Java все ще залишає за собою статус лідера як основного засобу для створення мобільних додатків.

Java - мова програмування загального призначення. Відноситься до об'єктно-орієнтованих мов програмування, причому з сильною типізацією. Творці мови реалізували принцип WORA: write once, run anywhere або «пиши один раз, запускай скрізь». Це означає, що написаний на Java додаток можна запустити на будь-якій платформі, якщо на ній встановлено середовище виконання Java (JRE, Java Runtime Environment), або по-іншому, Java-машина.

Це завдання вирішується завдяки компіляції написаного на Java коду у проміжний байт-код. Цей формат і виконує JVM або віртуальна машина Java. Якщо говорити більш точно, JVM - частина середовища виконання Java (JRE). Байт-код, який виконується віртуальною машиною не залежить від платформи і є однаковим для

					КНТЕУ 121 023-7.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		30

усіх можливих випадків. Сама ж віртуальна машина доробляє байт-код, проводячи його кінцеву компіляцію у машинні коди саме тієї платформи, на якій працює Java-машина. Такий підхід дозволяє поєднати переваги компільованих (висока швидкість виконання програм) та інтерпретованих (повна кросплатформеність) мов програмування та згладити їхні недоліки. Слід відмітити, що компіляція у проміжний код була застосована у мові Java вперше, але потім ця модель була скопійована і на інші платформи. Наприклад, на сьогодні код C# компілюється у так звану проміжну мову (IL – Intermediate Language), а потім уже на конкретній ЕОМ за допомогою платформи .NET доробляється до машинних кодів.

В Java реалізований механізм управління пам'яттю, який називається прибиральником сміття або garbage collector. Розробник створює об'єкти, а JRE за допомогою прибиральника очищує пам'ять, коли об'єкти перестають використовуватися.

Як зазначалося вище, синтаксис мови Java схожий на синтаксис інших С-подібних мов. Ось його деякі особливості:

- чутливість до регістру - ідентифікатори User і user в Java є різними сутностями;
- для іменування методів використовується lowerCamelCase. Якщо назва методу складається з одного слова, воно повинно починатися з малої літери. Приклад: firstMethodName ();
- для іменування класів використовується UpperCamelCase. Якщо назва складається з одного слова, воно повинно починатися з великої літери. Приклад: FirstClassName;
- назва файлів програми має точно збігатися з назвою класу з урахуванням чутливості до регістру. Наприклад, якщо клас називається FirstClassName, файл повинен називатися FirstClassName.java;
- ідентифікатори завжди починаються з літери (A-Z, a-z), знака \$ або нижнього підкреслення _.

					КНТЕУ 121 023-7.МР	Аркуш
						31
Зм.	Аркуш	№ докум	Підпис	Дата		

Java відноситься до мов програмування загального призначення. За даними компанії Oracle, програми на Java запускаються на 3 млрд девайсів. Це маркетингове повідомлення складно перевірити, проте точно відомо, що Java широко використовується і входить в число найбільш затребуваних мов.

Наприклад, переважна більшість великих компаній так, чи інакше використовують Java. Дуже багато серверних додатків для корпорацій написані на цій мові, зокрема, таких як програми для фінансових організацій, які забезпечують проведення транзакцій, фіксацію торгових операцій. Широке поширення Java у банках та інших фінансових інституціях обумовлене якраз стабільністю додатків на цій мові (відсутність «небезпечних» указників, які використовуються у мові C/C++, автоматичне прибирання сміття, розширена надійна обробка виключень exceptions і т.п.). Через це на Java написано дуже багато веб-додатків. Популярні фреймворки, в тому числі Spring, Struts, JSP, використовуються для створення різних додатків в Інтернет: від ecommerce-проектів до великих порталів, від освітніх платформ до урядових ресурсів. Популярна комп'ютерна гра Minecraft написана на Java.

Мобільна розробка - ще одна широка область використання Java, і цією мовою якраз пишуть програми для пристроїв, що працюють під управлінням ОС Android.

На Java створюють клієнтські програми. Простий і близький розробникам приклад: IDE NetBeans написано на Java.

Також Java застосовується для роботи з Big Data, розробки програм для наукових цілей, наприклад, обробки природних мов, програмування приладів - від побутових девайсів до промислових установок.

Тобто на Java можна писати різні типи додатків: веб, мобільний і десктопний софт, ігри і так далі. Традиційно у цієї мови сильні позиції в промисловому програмуванні, в сегменті великих компаній (т.зв. ентерпрайз).

Як підсумок можна сказати, що Java - мова програмування загального призначення, яка має C-подібний синтаксис і використовується для створення додатків в різних

					КНТЕУ 121 023-7.МР	Аркуш
						32
Зм.	Аркуш	№ докум	Підпис	Дата		

областях: від Інтернету до розробки ігор, від мобільного ПЗ до програм для корпорацій і наукових інститутів. Саме через ці причини дана мова і обрана у якості основного засобу для розробки у даній роботі.

2.3.3 Обґрунтування вибору засобів розробки

2.3.3.1 Вибір інтегрованого середовища розробки

Для мови Java на сьогоднішній день доступні різні варіанти вибору інтегрованих середовищ розробки (IDE, Integrated Development Environment). Історично раніше активно використовувалися такі продукти, як Eclipse та NetBeans, що і до сих пір активно використовуються досвідченими програмістами на Java. Однак, все більше осіб (особливо із числа тих, хто тільки розпочинає програмувати на цій мові) обирають потужне IDE IntelliJ IDEA, яке має чимало вбудованих інструментів для розробки та багато різних зручностей. Сюди відноситься якісне підсвічування коду, зручне авто доповнення коду для досить великої кількості ситуацій, пропозиції для виправлення виявлених помилок у коді, візуальне проектування інтерфейсу програми, тощо. Слід відмітити, що на основі даного IDE компанією JetBrains були створені рішення і для мобільної розробки для Android (Android Studio) і для веб-розробки (WebStorm) і т.п.

Єдиним недоліком даного програмного забезпечення є порівняно високі системні вимоги, однак, при використанні програмістом сучасних ПК, цей недолік не є суттєвим, тому для подальшого використання обираємо саме це середовище розробки.

2.3.3.2 Вибір інструментального засобу для збереження даних у системі

Раніше було встановлено, що незважаючи на усю складність процесу аналізу тексту на плагіат, у ньому відсутні великі обсяги інформації, яку слід зберігати на комп'ютері користувача (причому такої необхідності немає і безпосередньо під час сеансу перевірки, і між цими сеансами, оскільки кожного разу пошуку підлягає новий текст із іншими наборами фраз). Даний факт є причиною, через яку у програмному продукті, що розробляється, не є потрібним використання якоїсь системи управління базами даних.

					КНТЕУ 121 023-7.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		33

Усю інформацію, яку необхідно зберігати на комп'ютері користувача будемо розміщувати у звичайному текстовому файлі налаштувань. Таким чином, якщо говорити строго, в роботі буде використана проста текстова база даних.

2.3.3.3 Додаткові засоби розробки

Для повноцінного процесу розробки іноді є необхідним залучення інших програмних інструментів, окрім тих, що пропонує інтегроване середовище розробки (яким би якісним воно не було). Так, зокрема, інтерфейс користувача на сьогоднішній день часто проєктують (розробляють прототип) у загальновідомій по світу системі Figma, що також буде використана у даній роботі.

Під час альфа-тестування, яке виконується розробником програми ще в процесі розробки, іноді виникають помилки, які важко локалізувати, особливо, якщо вони мають логічний характер високого рівня абстракції. Для цього можуть знадобитися програми-відладчики із більшими можливостями, аніж стандартний засіб вбудований у IDE. У даному проєкті для відлагодження програми за необхідності будемо використовувати найкращий відладчик OllyDbg.

Також під час розробки використовуються і інші програмні продукти, наприклад, для документування – текстовий редактор типу Microsoft Word, і т.п., однак вони мало стосуються теми даної роботи, тому їх тут розглядати не будемо.

2.4 Висновки по розділу

Таким чином, у розділі проведено обґрунтування основних проєктних рішень по розробці програмного продукту для аналізу текстів здобувачів вищої освіти на академічний плагіат. Створено алгоритмічну складову проєкту, обрано основні технології та засоби розробки. Результати, висвітлені у даному розділі, дозволяють переходити до процесу реалізації відповідного програмного забезпечення, що буде описано у наступному розділі.

					КНТЕУ 121 023-7.МР	Аркуш
						34
Зм.	Аркуш	№ докум	Підпис	Дата		

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВИЗНАЧЕННЯ АКАДЕМІЧНОГО ПЛАГІАТУ В РОБОТАХ ЗДОБУВАЧІВ ОСВІТИ

3.1 Прототипування інтерфейсу користувача програмного забезпечення

В першу чергу, при створенні інтерфейсу користувача слід перелічити інформацію, яку користувачеві слід вводити у програму, та яку він може з неї отримати. Отже, для введення можуть бути передбачені наступні елементи управління:

- велике поле для відображення тексту, що підлягає поточному аналізу. Дане поле повинно мати можливість безпосереднього введення тексту, а також заповнення вмістом файлу;
- кнопка, що викликає діалог відкриття файлу, який повинен містити текст, що підлягає аналізу;
- кнопка, яка ініціює процес перевірки;
- елемент управління для налаштування числа слів у шинглі;
- головне меню, що є стандартом для програм з графічним інтерфейсом користувача.

Для виведення інформації користувачу потрібні наступні елементи:

- інформаційне поле, в якому у текстовій формі з різними елементами форматування має відображуватися стан системи під час процесу перевірки (сюди можна включати назви або адреси текстів, з якими знайдено співпадіння, видавати навколишній текст, який знаходиться поруч із знайденим шинглом, часові мітки цих операцій і т.п.);
- індикатор прогресу всього процесу, який можна об'єднати із попереднім елементом управління або зробити окремим;

					<i>КНТЕУ 121 02з-7.МР</i>		
Зм.	Аркуш	№ докум	Підпис	Дата			
Зав. каф.		Криворучко О. В.		21.06.21	Проектування програмного забезпечення визначення академічного плагіату в роботах здобувачів освіти	Стадія	Аркуш
Керівник		Сашньова М. В.		21.06.21		РЗ	35
Гарант		Токар В. В.		21.06.21		Архивів	
Розробив		Гудкова В. Є.		21.06.21	Реалізація програмного забезпечення визначення академічного плагіату в роботах здобувачів освіти	Факультет інформаційних технологій 2 курс, 2м група	

- поле для виведення кінцевого результату – числового значення оригінальності тексту.

В результаті розміщення вказаних елементів управління на формі отримаємо зовнішній вигляд вікна проєктованого програмного забезпечення, показаний на рис. 3.1.

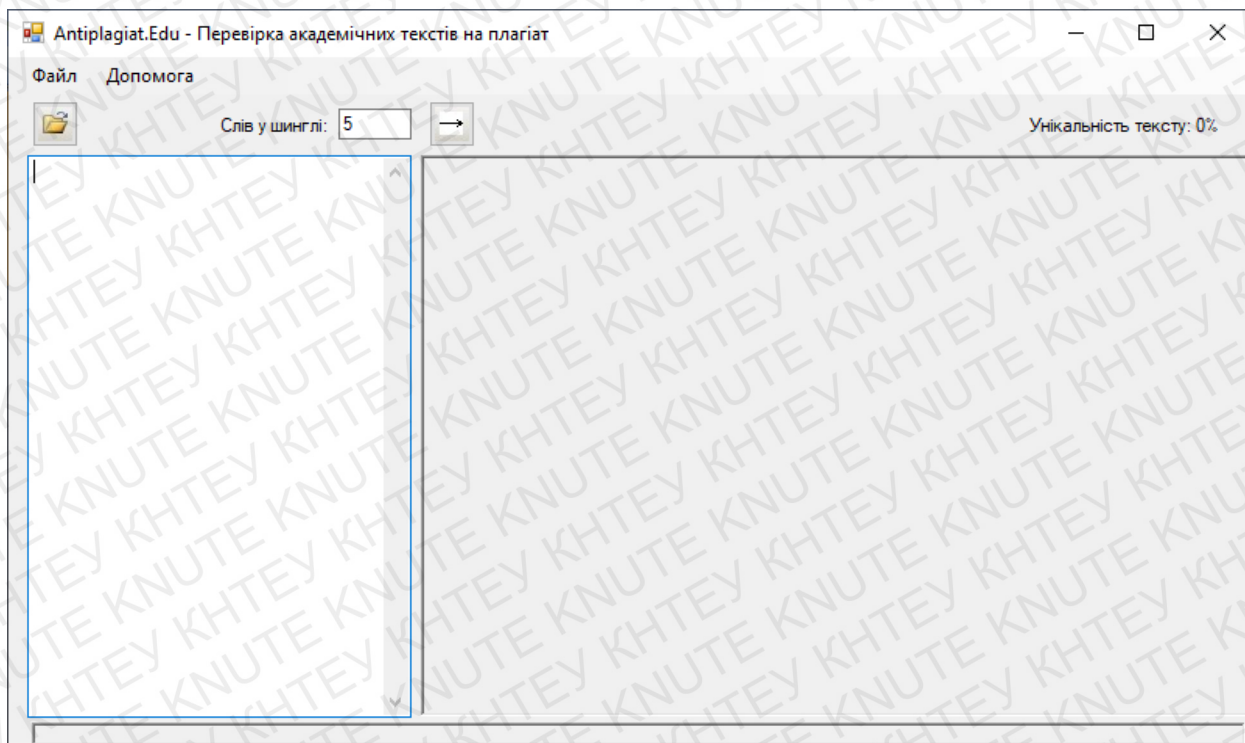


Рис. 3.1. Зовнішній вигляд головного вікна програмного забезпечення для перевірки оригінальності академічних текстів, що проєктується

Окрім головного вікна, передбачимо у програмі традиційне вікно About, тобто відомості про програму – рис. 3.2.

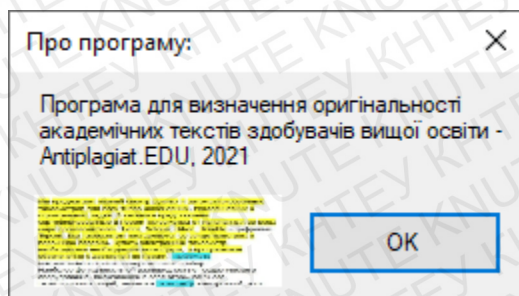


Рис. 3.2. Зовнішній вигляд вікна «Про програму»

					КНТЕУ 121 023-7.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		36

3.2 Особливості реалізації сховища даних продукту

Як уже було зазначено вище, програмі, що виконує перевірку на плагіат, не потрібно зберігати великі обсяги даних у базі даних, тому у 2.3.3.2 було вирішено використовувати не СУБД, а звичайний текстовий файл для зберігання простих налаштувань програми та, можливо, ще якоїсь додаткової інформації. Тут слід сказати, що користувачеві може знадобитися доступ до переліку уже виконаних операцій, тому, зважаючи на те, що це можливість, яка відносно просто реалізовується, запровадимо у даний файл ще і невеликий протокол перевірок, який включатиме назву файлу, дату та час перевірки та результат у відсотках.

Кінцево сховище даних проєктованого програмного забезпечення матиме вигляд як на рис. 3.3.

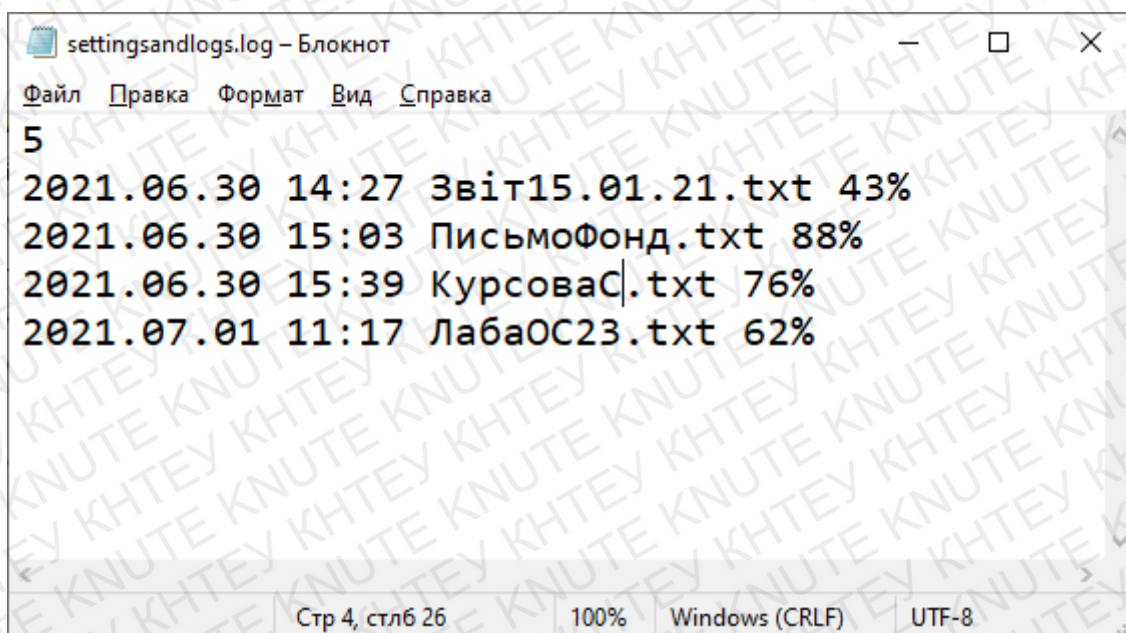


Рис. 3.3. Сховище даних програми після кількох використань (у першому рядку зберігається виставлене раніше число слів у шинглі)

Таким чином, сховище даних спроектовано і можна переходити до наступного етапу програмної реалізації, що описується у наступному підрозділі.

3.3 Опис реалізації основних алгоритмічних складових продукту

Для проведення програмної реалізації додатку з графічним інтерфейсом слід обрати якусь бібліотеку, за допомогою якої ця можливість була би реалізована. У

					КНТЕУ 121 023-7.МР	Аркуш
						37
Зм.	Аркуш	№ докум	Підпис	Дата		

даній роботі обрано JavaFX, що нативно підтримується обраним середовищем розробки IntelliJ IDEA.

Отже, при відкритті файлу використовується клас `FileChooser`, а код обробник кнопки «Відкрити файл...» міститиме наступний код, пов'язаний із завантаженням тексту в програму:

@FXML

```
private void hndlOpenFile(ActionEvent event) {  
    FileChooser fileChooser = new FileChooser();  
    fileChooser.setTitle("Open Document");  
    FileChooser.ExtensionFilter extFilter =  
        new FileChooser.ExtensionFilter("HTML files (*.html)", "*.html");  
    fileChooser.getExtensionFilters().add(extFilter);  
    File file = fileChooser.showOpenDialog(CodeNote.mainStage);  
    if (file != null) {  
        // операції по зчитуванню тексту, розміщеного у файлі;  
    }  
}
```

Саме зчитування можна проводити традиційно за допомогою класу `StringBuilder` і наступного коду:

```
BufferedReader reader = new BufferedReader( new FileReader (fileName));  
String line = null;  
StringBuilder stringBuilder = new StringBuilder();  
String ls = System.getProperty("line.separator");  
while( ( line = reader.readLine() ) != null ) {  
    stringBuilder.append( line );  
    stringBuilder.append( ls );  
}  
stringBuilder.deleteCharAt(stringBuilder.length()-1);
```

					КНТЕУ 121 023-7.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		38

Розбиття великого тексту на шингли можна проводити у циклі while-do поки не дійдемо до кінця файлу, причому для виділення конкретного шинглу використовуємо вкладений цикл який повторюється 5 разів (де 5 – обране число слів у шинглі). У такому циклі відраховуємо по 5 пробілів за допомогою методу indexOf(‘ ‘), який у Java існує у будь-якого рядка і у даній програмі викликається для рядка, що містить увесь текст, який підлягає аналізу (і який розбивається на шингли).

Безпосередній запит до системи Google звичайно виконуємо по протоколу HTTP, а саме з використанням простого (але ефективного у випадку, що розглядається) методу GET. Як відомо, другий метод POST є більш складним і його слід використовувати, коли передаються бінарні дані, прикріплені до HTTP-запиту, або у запиті міститься секретна інформація. В нашому випадку обидві умови не є справедливими, плюс формувати GET-запити набагато простіше, ніж POST, саме тому і обираємо варіант GET, який реалізується функцією:

```
String serverAnswer = getUrl("http://www.google.com.ua/search?q=\'" + current
Shingle + "\'");
```

Розбір відповіді serverAnswer реалізується звичайними функціями для роботи з рядками, але можливо використовувати виключно регулярні вирази (класи Pattern та Matcher) та код виду:

```
Pattern pattern = Pattern.compile(RegExp);
Matcher matcher = pattern.matcher(serverAnswer);
while (matcher.find()) {
    System.out.println(text.substring(matcher.start(), matcher.end()));
}
```

Таким чином, основні аспекти програмної реалізації тут розглянуто, а інші, зважаючи на їх відносну простоту, будуть реалізовуватися уже відразу в готових програмних кодах.

					КНТЕУ 121 023-7.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		39

3.4 Документування створеного програмного забезпечення (розробка керівництва користувача)

Обов'язковим етапом створення програмних продуктів є створення комплекту документації, що може бути достатньо обширним, але у даній роботі розглянемо тільки такий важливий документ, як керівництво користувача програми. Його можна створити у вигляді нумерованої інструкції, яка в цілому відповідає рис. 2.3 та частково рис. 2.1, а також враховує особливості інтерфейсу користувача рис. 3.1.

1. Завантажити програму стандартним для системи Windows способом.
2. Натиснути кнопку відкриття файлу АБО у меню «Файл» обрати пункт «Відкрити файл...».
3. У новому діалоговому вікні, що відкривається після виконання пункту 2, слід обрати файл формату *.txt із текстом, який підлягає аналізу та натиснути кнопку «ОК».
4. Перевірити, що після виконання пункту 3 у велике текстове поле, розміщене у лівій частині вікна програми, завантажився саме той текст, що підлягає аналізу.
5. Замість пунктів 2-4 можна помістити потрібний текст у велике текстове поле, розміщене у лівій частині вікна програми, шляхом простого редагування цього поля (набрати цей текст з клавіатури, або використати стандартну для ОС Windows операцію «Вставити»/«Paste».
6. Після того, як текст для аналізу з'явився у великому лівому текстовому полі слід задати число слів, яке буде використовуватися в одному шинглі під час пошуків або залишити те значення, яке там відображується за умовчанням (якщо його ніколи не міняли, то воно дорівнює 5, а якщо колись відбувалися пошуки з іншими значеннями цього числа, то у даному полі відображується те число слів у шинглі, яке використовувалося під час аналізу попереднього тексту.
7. Коли текст і число слів у шинглі задані, можна ініціювати процес визначення унікальності (або оригінальності – ці два слова використовуються як

					КНТЕУ 121 023-7.МР	Аркуш
						40
Зм.	Аркуш	№ докум	Підпис	Дата		

синоніми), для чого слід натиснути кнопку із зображенням стрілки «Вправо» або у меню «Файл» обрати пункт «Перевірити текст».

8. Після виконання пункту 7 потрібно зачекати завершення процесу аналізу тексту, прогрес якого буде відображатися у самому низу вікна програми.

9. Після завершення аналізу можна передивитися особливості його проведення, які відображуються у великому текстовому полі, розміщеному у правій частині вікна програми, а головне – дізнатися унікальність тексту, яка видається окремим діалоговим вікном - рис. 3.4, а також відображується у правій верхній частині вікна програми.

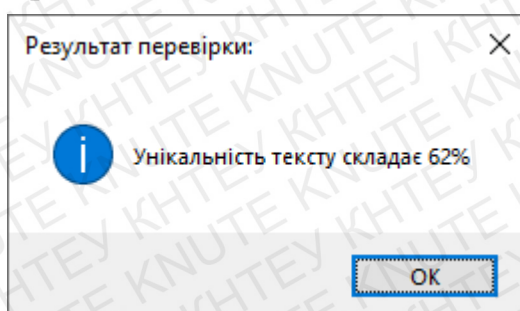


Рис. 3.4. Вікно з результатом перевірки оригінальності тексту

10. Якщо потрібно аналізувати інші тексти, то перейти до пункту 2, інакше закрити програму стандартним для ОС Windows способом.

3.5 Тестування створеного програмного продукту та аналіз його ефективності

Після створення усіх частин програмного продукту його робота була піддана тестуванню, що показало наступні результати:

- робота програмного забезпечення проходить стабільно, без виникнення системних помилок, аварійних завершень програми та інших нештатних ситуацій;
- продуктивність створеного програмного забезпечення знаходиться на задовільному рівні: зависання, зупинки або помітні паузи в роботі програми відсутні при проведенні будь-яких дозволених у ній операцій;
- в цілому програмний продукт адекватно виконує поставлену перед ним задачу по визначенню унікальності тексту – рис. 3.5: інформація добре сприймається (розміщена із відповідним форматуванням), що в цілому дозволяє

					КНТЕУ 121 023-7.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		41

особам, які бажають перевірити свій текст на плагіат і не є фахівцями в галузі ІТ, зробити це легко, швидко та максимально комфортно. Результати порівняння визначеної створеною програмою унікальності для трьох різних текстів із результатами інших програмних продуктів наведена у табл. 3.1.

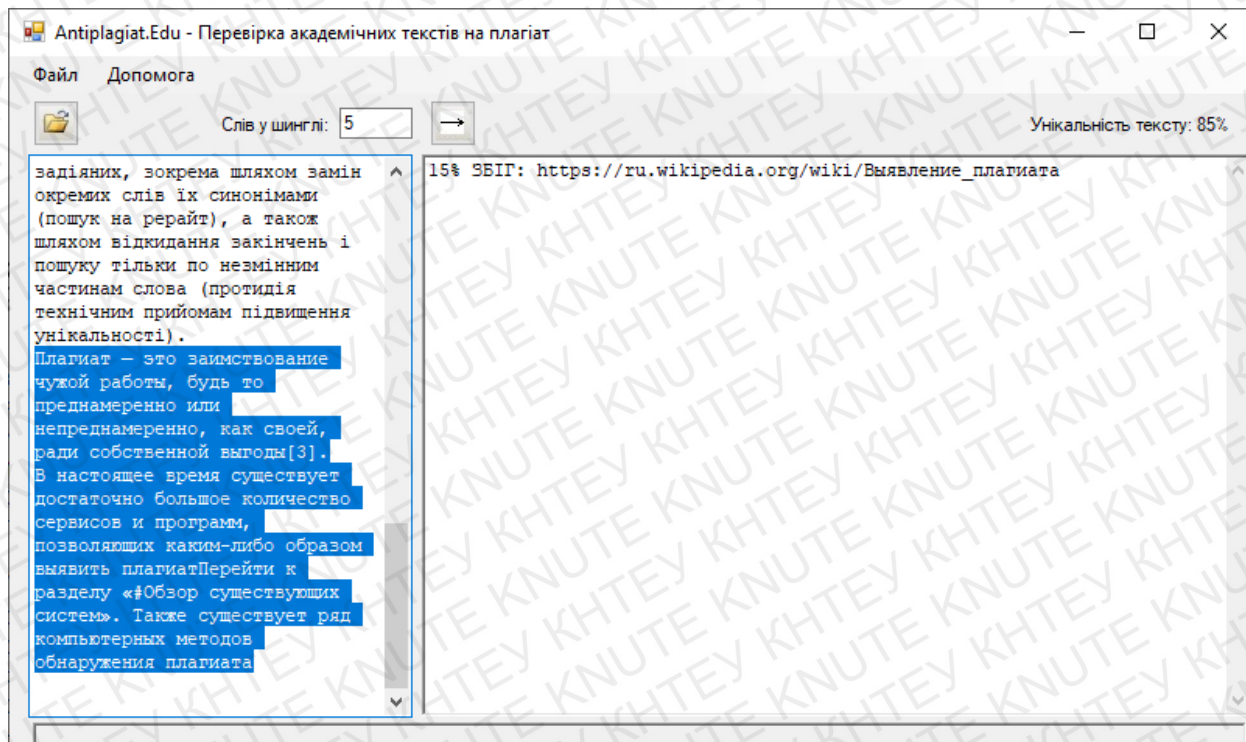


Рис. 3.5. Приклад роботи програми: збіг із результатами, що надані іншими продуктами з точністю до 1%

Таблиця 3.1. Порівняння результатів визначення унікальності із іншими системами

	Контрольний текст №1	Контрольний текст №2	Контрольний текст №3	Середнє
ETXT	86%	76%	58%	-
Antiplagiarism				
Розроблена система Antiplagiat.Edu	88%	76%	62%	-
Середнє відхилення від аналогів	+3%	+1,5%	+5%	+3,2%

					КНТЕУ 121 023-7.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		42

Аналізуючи табл. 3.1, можна сказати, що створена програма в середньому показує дещо вищу оригінальність, аніж інші аналоги (хоча у деяких випадках спостерігається навіть точний збіг результатів), однак середнє значення різниці, яке складає +3,2%, є досить малим, отже створений програмний продукт має досить високий рівень надійності рішень і його можна (доцільно) використовувати для подальшого практичного використання.

3.6 Висновки по розділу

Таким чином, у даному розділі проведено програмну реалізацію програмного продукту для визначення академічного плагіату в роботах здобувачів вищої освіти. Спроектовано інтерфейс користувача, який виконано мінімалістичним, а отже, максимально ефективним для осіб, що не є фахівцями в галузі ІТ. Описана структура сховища даних, яке містить налаштування та протоколи роботи програми. Пояснюються основні аспекти реалізації у програмних кодах алгоритмів роботи системи. Також у розділі розроблена інструкція по використанню (керівництво користувача програмного продукту) та описуються результати проведеного тестування. В цілому програма видає адекватні результати і може бути використана у реальній практичній діяльності.

					КНТЕУ 121 023-7.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		43

ВИСНОВКИ

Таким чином у даній роботі проведено проєктування та реалізацію програмного забезпечення для визначення академічного плагіату в роботах здобувачів освіти, що дозволяє підвищити якість їх виконання за рахунок неможливості залучення сторонніх текстів, а через необхідність їх створення власноруч.

Початково проведено аналіз існуючих систем визначення плагіату, основних принципів їх роботи та важливих особливостей, виділено їх недоліки та сформовано технічне завдання на дану розробку. Потім розроблено алгоритми та проведено опис даних, які були покладені в основу проєктованого програмного забезпечення.

В роботі обґрунтовано вибір об'єктно-орієнтованої технології програмування, мови Java та середовища IntelliJ IDEA. Розроблено інтерфейс користувача, проведено програмну реалізацію і виконано тестування створеного ПЗ, в результаті чого встановлено його цілком задовільні експлуатаційні показники та досить добрий збіг результатів роботи програми з іншими популярними аналогами. Це означає, що даний продукт може використовуватися у реальній практичній діяльності особами, що створюють тексти (для самоконтролю) та їх перевіряють.

У перспективі робота може розширюватися за рахунок залучення додаткових методів контролю, або покращення задіяних, зокрема шляхом заміни окремих слів їх синонімами (пошук на рерайт), а також шляхом відкидання закінчень і пошуку тільки по незмінним частинам слова (протиція технічним прийомом підвищення унікальності).

					КНТЕУ 121 023-7.МР			
Зм.	Аркуш	№ докум	Підпис	Дата				
Зав. каф.		Криворучко О. В.		01.11.21	Проектування програмного забезпечення визначення академічного плагіату в роботах здобувачів освіти Висновки	Стадія	Аркуш	Аркушів
Керівник		Сашньова М. В.		01.11.21		В	44	49
Гарант		Токар В. В.		01.11.21		Факультет інформаційних технологій 2 курс, 2м група		
Розробив		Гудкова В. Є.		01.11.21				

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Top 10 Search Engines In The World (2021 Update) [Електронний ресурс]. – RELIABLESOFT: Digital Marketing Agency, 2021. – Режим доступу: <https://www.reliablesft.net/top-10-search-engines-in-the-world>
2. Search Engine Market Share Worldwide (2021 Update) [Електронний ресурс]. – StatCounter, 2021. – Режим доступу: <https://gs.statcounter.com/search-engine-market-share>
3. Модель-вид-контролер [Електронний ресурс]. – Вікіпедія. Вільна енциклопедія. – Режим доступу: <https://uk.wikipedia.org/wiki/Модель-вид-контролер>
4. SEO-копірайтинг – Что это такое, для чего он нужен и где ему можно научиться? [Електронний ресурс]. – Режим доступу: <https://sergeysmirnovblog.ru/blogging/seo-kopirajting.html>
5. Блох, Дж. Java: эффективное программирование / Дж. Блох. - М.: Диалектика, 2019. - 464 с.
6. Chen, X. Shared information and program plagiarism detection [Text] / Xin Chen; Brent Francia; Ming Li, Brian McKinnon, Amit Seker. // IEEE Transactions on Information Theory. – 2004. – Vol. 50, № 7. – P. 1545-1551
7. . Features of Java [Електронний ресурс]. – Режим доступу: <https://www.javatpoint.com/features-of-java>
8. Bishop, J. and Bishop, N. Java Gently for Engineers and Scientists / J. Bishop and N. Bishop. - Austin, Texas, USA, 2000
9. Герман, О.В. Программирование на Java и С# / О.В. Герман. - СПб.: BHV, 2012. - 512 с.

					<i>КНТЕУ 121 023-7.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>	Проектування програмного забезпечення визначення академічного плагіату в роботах здобувачів освіти	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Зав. каф.		Криворучко О. В.		27.02.21		<i>ПВД</i>	45	49
Керівник		Сашньова М. В.		27.02.21		Факультет інформаційних технологій 2 курс, 2м група		
Гарант		Токар В. В.		27.02.21				
Розробив		Гудкова В. Є.		27.02.21				
					<i>Перелік використаних джерел</i>			

10. IntelliJ IDEA Review [Электронный ресурс]. – Режим доступа: <https://comparecamp.com/intellij-idea-review-pricing-pros-cons-features/>
11. Дашнер, С. Изучаем Java EE. Современное программирование для больших предприятий / С. Дашнер. - СПб.: Питер, 2018. - 384 с.
12. МакГрат, М. Программирование на Java для начинающих / М. МакГрат. - М.: Эксмо, 2016. - 192 с.
13. Мартин, К. Соломон Oracle. Программирование на языке Java / Мартин К. Соломон, Нирва Мориссо-Леруа, Джули Басу. - М.: ЛОРИ, 2010. - 512 с.
14. Comparison of Eclipse 3.6 and IntelliJ IDEA 10.5: Pros and Cons [Электронный ресурс]. – Режим доступа: <https://theholysjava.wordpress.com/2011/10/18/comparison-of-eclipse-3-6-and-intellij-idea-10-5-pros-and-cons/>
15. Нимейер, П. Программирование на Java / П. Нимейер, Д. Леук. - М.: Эксмо, 2018. - 448 с.
16. Нотон, П. Java. Справочное руководство. Все, что необходимо для программирования на Java / Нотон, Патрик. - М.: Бином, 2015. - 448 с.
17. Смоленцев, Н.К. MATLAB. Программирование на C++, C#, Java и VBA / Н.К. Смоленцев. - М.: ДМК, 2015. - 498 с.
18. Эмерик, Ч. Программирование в Clojure: Практика применения Lisp в мире Java / Ч. Эмерик. - М.: ДМК, 2015. - 816 с.

					КНТЕУ 121 023-7.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		46