

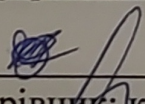
Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

БАКАЛАВРСЬКА ДИПЛОМНА РОБОТА

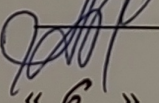
на тему:

«Розробка програми захисту авторського права цифрових зображень дизайну
інтерерів»

Виконав: студент 4 курсу, гр. 2КІТС-206
спеціальності 125– Кібербезпека
Освітня програма – Кібербезпека
інформаційних технологій та систем

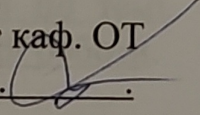
 Шевчук Б. В.

Керівник: к.т.н., доцент. каф. МБІС

 Адлер О. О.

« 6 » сервія 2024 р.

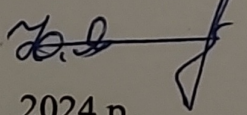
Рецензент: к.т.н., доц., доцент каф. ОТ

 Тарновський М. Г.

« 6 » сервія 2024 р.

Допущено до захисту

Голова секції УБ кафедри МБІС

д.т.н., проф. Яремчук Ю. Є. 

« 6 » сервія 2024 р.

Вінниця ВНТУ – 2024

Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

Рівень вищої освіти I-й (бакалаврський)

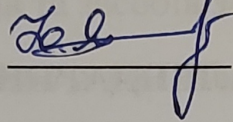
Галузь знань 12 – Інформаційні технології

Спеціальність 125 – Кібербезпека

Освітньо-професійна програма - Кібербезпека інформаційних технологій
та систем

ЗАТВЕРДЖУЮ

Голова секції УБ, кафедра МБІС



д.т.н., проф. Яремчук Ю.Є.

“ 22 ” березня 2024 р.

З А В Д А Н Н Я

на бакалаврську дипломну роботу студенту

Шевчуку Богдану Вікторовичу

(прізвище, ім'я, по-батькові)

1. Тема роботи Розробка програми захисту авторського права цифрових зображень дизайну інтер'єрів

Керівник роботи к.т.н., доцент кафедри МБІС Адлер Оксана Олександрівна
(прізвище, ім'я, по-батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “ 11 ” березня 2024 року
№ 80

2. Термін подання студентом роботи _____

3. Вихідні дані до роботи Електронні джерела та наукові статті по темі. Програмне забезпечення, яке стосується теми бакалаврської дипломної роботи.

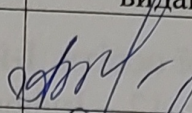
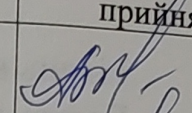
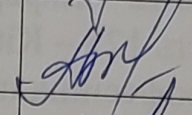
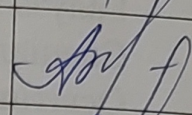
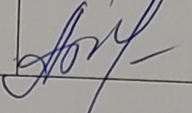
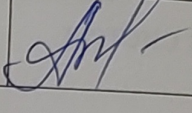
4. Зміст текстової частини:

Аналіз існуючих рішень та обґрунтування теми бакалаврської дипломної роботи. Вибір методів захисту авторського права. Розробка та тестування програми.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень)

Блок-схема алгоритму. Тест завантаження зображень. Тест генерації та перевірки цифрових підписів. Перевірка підпису. Програма для захисту авторського права цифрових зображень дизайну інтер'єрів.

6. Консультанти розділів роботи

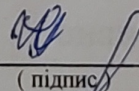
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ I	Адлер О.О., к.т.н., доц. каф. МБІС		
Розділ II	Адлер О.О., к.т.н., доц. каф. МБІС		
Розділ III	Адлер О.О., к.т.н., доц. каф. МБІС		

7. Дата видачі завдання 22 березня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

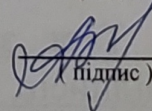
№	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Аналіз предметної області	22.03.2024	30.03.2024	Виконано
2	Аналіз методів розв'язання задачі	01.04.2024	10.04.2024	Виконано
3	Постановка задачі розробки програми	11.04.2024	18.04.2024	Виконано
4	Дослідження інформаційних джерел	19.04.2024	30.04.2024	Виконано
5	Аналіз варіантів роботи програми	01.05.2024	10.05.2024	Виконано
6	Розробка алгоритму роботи	11.05.2024	18.05.2024	Виконано
7	Реалізація програми	19.05.2024	25.05.2024	Виконано
8	Тестування програми	26.05.2024	28.05.2024	Виконано
9	Оформлення матеріалів до захисту БДР	29.05.2024	07.06.2024	Виконано

Студент


(підпис)

Шевчук Б.В.
(прізвище та ініціали)

Керівник роботи


(підпис)

Адлер О.О.
(прізвище та ініціали)

АНОТАЦІЯ

Бакалаврська дипломна робота складається з 98 сторінок формату А4, на яких є 4 рисунка, список використаних джерел містить 44 найменувань.

Метою роботи є розробка програми захисту авторського права цифрових зображень дизайну інтер'єрів .

У загальній частині роботи розглянуто аналіз предметної області виконано детальний аналіз існуючих методів та засобів захисту авторських прав. Описано процес створення алгоритму для програми, яка забезпечує захист авторських прав на цифрові зображення. Викладено кроки щодо розробки та тестування програмного забезпечення для захисту авторських прав. Надано технічне завдання, що включає мету, область застосування, вимоги до функціональних характеристик програми та перелік необхідних ресурсів. Розділи охоплюють основні теоретичні та практичні аспекти, необхідні для успішного захисту авторських прав.

Програма включає в себе механізми контролю доступу, генерації та перевірки цифрових підписів для зображень, а також зручний інтерфейс користувача для управління захистом авторських прав.

Ключові слова: захист авторського права, цифрові зображення, дизайн інтер'єрів, програмна розробка, інформаційна безпека.

ABSTRACT

The Bachelor's thesis consists of 98 A4 pages, containing 4 figures. The list of references includes 44 sources.

The objective of the work is to develop a program for the protection of copyright for digital images in interior design.

The general section of the work includes an analysis of the subject area, providing a detailed review of existing methods and means of copyright protection. It describes the process of creating an algorithm for a program that ensures copyright protection for digital images. The steps for the development and testing of the software for copyright protection are outlined. The technical specification is provided, including the purpose, scope of application, functional requirements of the program, and a list of necessary resources. The sections cover the main theoretical and practical aspects required for the successful protection of copyrights.

The program includes mechanisms for access control, generation and verification of digital signatures for images, as well as a user-friendly interface for managing copyright protection.

Keywords: copyright protection, digital images, interior design, software development, information security.

ЗМІСТ

ВСТУП.....	7
1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ У ПРОГРАМАХ ЗАХИСТУ АВТОРСЬКОГО ПРАВА	12
1.1. Аналіз існуючих програм захисту авторського права	12
1.2. Основні методи захисту цифрових зображень	23
1.3. Принципи роботи з алгоритмами захисту	28
1.4. Висновки до розділу.....	33
2 ВИБІР МЕТОДІВ ЗАХИСТУ АВТОРСЬКОГО ПРАВА	35
2.1. Вибір методів захисту та технічні вимоги до розробки	35
2.2. Обґрунтування мови програмування.....	37
2.3. Алгоритм роботи програми.....	39
2.4. Висновки до розділу.....	40
3 РОЗРОБКА ТА ТЕСТУВАННЯ ПРОГРАМИ.....	46
3.1. Розробка програми захисту авторського права цифрових зображень дизайну інтер'єрів	46
3.2. Тестування та оптимізація програми	54
3.3. Практичне застосування програми в дизайні інтер'єрів	59
ВИСНОВКИ.....	75
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	76
ДОДАТКИ.....	81

ВСТУП

Актуальність дослідження. В сучасному цифровому світі захист авторського права на цифрові зображення в галузі дизайну інтер'єрів стає все більш актуальним питанням. З ростом кількості професіоналів, які працюють у цій галузі, а також зі збільшенням кількості цифрових платформ для публікації та обміну дизайнерськими роботами, ризик порушення авторських прав зростає. Україна, яка має потужний потенціал у галузі ІТ та дизайну, потребує ефективних рішень для захисту інтелектуальної власності своїх спеціалістів. Розробка програми захисту авторського права для цифрових зображень в галузі дизайну інтер'єрів може стати важливим кроком у розвитку цієї сфери в Україні, сприяючи стимулюванню творчості та інновацій. Додатково, важливо враховувати, що відсутність ефективного захисту авторських прав може гальмувати інноваційний розвиток в галузі дизайну, адже творці можуть перейматись та боятись публікувати свої роботи або брати участь у колективних проектах через страх незаконного використання їхніх творінь.

Розробка програми захисту авторського права для цифрових зображень в галузі дизайну інтер'єрів може стати важливим кроком у розвитку цієї сфери в Україні. Це не лише захистить інтелектуальну власність українських дизайнерів, але й сприятиме стимулюванню творчості та інновацій, що є ключовим для динамічного розвитку галузі. Захист авторського права також зможе привернути інвестиції в український дизайн, роблячи його більш конкурентоспроможним на міжнародному ринку. Без сумніву, захист авторських прав є важливим елементом для будь-якої креативної індустрії, і галузь дизайну інтер'єрів не є винятком. Захист інтелектуальної власності дозволить дизайнерам відчувати себе захищеними, роблячи їхню роботу більш впевненою та стабільною. Це також сприятиме підвищенню репутації українських спеціалістів на світовому ринку, що в свою чергу може відкрити нові можливості для експорту українського дизайну.

Варто зазначити, що найбільш ефективним засобом активного захисту дизайну інтер'єрів як об'єктів авторського права є, поза сумнівом, судові позови із вимогою матеріальної компенсації. В цьому випадку, провідна роль в процесі доведення фактів порушення прав відводиться професіоналам судової експертизи. Необхідно акцентувати, що експертизи таких об'єктів авторського права як дизайн інтер'єрів в нашій країні на даний час належать до числа тих справді рідкісних досліджень, що трапляються далеко не кожен рік. Втім, у переважній більшості випадків експертизи інтер'єрів приміщень є дослідженнями високого рівня складності, їхнє проведення вимагає значних витрат часу на ознайомлення та аналіз численних фотографічних матеріалів, будівельних планів, схем і креслень.

Як наслідок, вартість проведення подібних досліджень може компенсувати усі витрати, необхідні для відповідного підготовки науково-методичного інструментарію. Водночас, з огляду на відсутність офіційної методики дослідження дизайну інтер'єрів як об'єктів авторського права, фахівці, які вже зустрічалися із дослідженнями дизайнів внутрішнього простору приміщень як об'єктів авторського права, змушені щоразу витрачати час та зусилля на пошуки тих науково-методичних засобів та фахової довідкової літератури, які потрібні для проведення дослідження. Певна річ, досвід за кордонних фахівців в цьому випадку становить чималу практичну цінність. Аналіз останніх досліджень та публікацій. З огляду на порівняну новизну теми публікації слід зазначити, що в середовищі фахівців з інтелектуальної власності тема дослідження об'єктів дизайну інтер'єрів як об'єктів авторського права все ще залишається на маргінесах наукових інтересів.

Разом з тим, серед закордонних спеціалістів вже існує ряд професійних довідкових публікацій, присвячених даній темі. Серед них необхідно зазначити розвідки «Architecture and copyright: a quick survey of the law» А.Адріана; «A New Remedy for Copyright Infringement» А.Арнольда; довідник «Conditions of Engagement» для інженерів та дизайнерів приміщень колективу авторів з «Association of Consulting Engineers»; «Case Comment – United States: copyright

– copyright infringement for architectural works» Б.Дейтца; «It's mine, not yours: Intellectual Copyright, Construction Law» П.Ньюмана, Copyright Issues on the Protection of Architectural Works and Designs П. Манн й Дж.Денонкоурт та ряд інших.

Окремо зауважимо, що публікації за темою дослідження дизайну інтер'єрів приміщень як об'єктів авторського права навіть у середовищі закордонних фахівців Сполучених Штатів, Великобританії та Європейського союзу належить до числа тих наукових напрямків, які, так би мовити, все ще перебувають в процесі становлення на рівнях теорії й практики проведення судових експертиз.

Важливо також враховувати, що в епоху цифровізації доступ до інформації стає все простішим. Це, з одного боку, сприяє швидшому розповсюдженню творчих ідей і зразків, а з іншого – може призвести до незаконного використання і копіювання. Тому існування надійної системи захисту авторських прав є критично важливим для забезпечення справедливого винагородження та визнання роботи творців.

Також варто зазначити, що розробка такої програми може послужити прикладом для інших галузей, де захист інтелектуальної власності є актуальною проблемою. Таким чином, наше дослідження може мати ширший вплив на інформаційні технології та інновації не тільки в Україні, але й у всьому світі. Наше дослідження зосереджено на розробці програмного рішення для захисту авторського права цифрових зображень, що відповідає науковим програмам з інформаційної безпеки та захисту інтелектуальної власності. Воно взаємодіє з темами сучасних наукових досліджень в області комп'ютерної техніки, програмування та дизайну.

Об'єкт дослідження: цифрові зображення в галузі дизайну інтер'єрів.

Предмет дослідження: методи захисту авторського права для цифрових зображень.

Мета дослідження: розробка програми захисту авторського права для цифрових зображень в галузі дизайну інтер'єрів.

Задачі дослідження:

1. Критичний аналіз існуючих методів захисту авторського права.
2. Розробка нового алгоритму захисту.
3. Тестування та оптимізація розробленої програми.

Методи дослідження: для виконання дослідження будуть використані наукові методи аналізу, порівняння та експериментального тестування програмного забезпечення, статистичний аналіз результатів для оцінки ефективності та надійності розробленої програми.

Наукова та практична новизна одержаних результатів. Наукова новизна полягає в розробці нового алгоритму захисту авторського права для цифрових зображень в галузі дизайну інтер'єрів. Практична новизна визначається можливістю застосування розробленої програми для захисту інтелектуальної власності професіоналів у цій галузі, що може стати важливим кроком у розвитку цифрового дизайну в Україні.

1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ У ПРОГРАМАХ ЗАХИСТУ АВТОРСЬКОГО ПРАВА

1.1. Аналіз існуючих програм захисту авторського права

Аналіз існуючих програм захисту авторського права включає огляд різноманітних програмних рішень, які призначені для захисту цифрових зображень від несанкціонованого використання та порушень авторських прав. Розглянемо деякі основні аспекти, які можна включити до такого аналізу [41]:

Функціональність. Оцінка того, які функції та можливості пропонує програма захисту авторського права. Що, включає наявність водяних знаків, шифрування, контроль доступу, аналіз метаданих, виявлення копіювання та інші функції. Коли оцінюється функціональність програми захисту авторського права, важливо враховувати різноманітні функції та можливості, які вона може пропонувати. Приділимо увагу ключовим функціям, які зазвичай включаються у такі програми:

- Вбудовання водяних знаків. Програма може дозволяти вбудовувати видимі або невидимі водяні знаки безпосередньо в цифрові зображення. Це дозволяє ідентифікувати та відстежувати авторство зображення, а також захищати від несанкціонованого копіювання.
- Шифрування. Програма надає можливість шифрувати цифрові зображення для забезпечення конфіденційності та захисту від несанкціонованого доступу [4].
- Контроль доступу. Така функція дозволяє обмежувати доступ до цифрових зображень залежно від рівня прав доступу, які надаються користувачам. Вона може включати механізми аутентифікації, авторизації та аудиту доступу.
- Аналіз метаданих. Програма аналізує метадані зображень для виявлення інформації про їхнє походження, авторство та інші важливі характеристики.

- Виявлення копіювання. Ця функція дозволяє виявляти незаконне копіювання або використання цифрових зображень без дозволу власника авторських прав.
- Система моніторингу та аналітики. Деякі програми можуть надавати можливість моніторити активність користувачів та аналізувати дані для виявлення підозрілих дій або порушень безпеки.

Ефективність. Необхідно проаналізувати, наскільки ефективно програма захисту авторського права запобігає несанкціонованому використанню та порушенням авторських прав. Що, включає оцінку рівня захисту, зручність використання та широке застосування в різних галузях. При аналізі ефективності програми захисту авторського права важливо враховувати декілька ключових аспектів, щоб оцінити її здатність ефективно запобігати несанкціонованому використанню та порушенням авторських прав [46]:

- Програма повинна надати високий рівень захисту для цифрових зображень, забезпечуючи надійні механізми для вбудовування водяних знаків, шифрування, контролю доступу та інших методів захисту. Важливо, щоб програма ефективно запобігала несанкціонованому доступу та копіюванню зображень.
- Програма повинна бути легкою у використанні та налаштуванні для користувача. Це включає інтуїтивний інтерфейс користувача, зрозумілі налаштування та швидкий доступ до ключових функцій захисту.
- Ефективна програма захисту авторського права має бути придатною для використання в різних галузях та сферах діяльності. Вона повинна враховувати специфіку потреб та вимог різних користувачів, включаючи фотографів, дизайнерів, видавців та інших творчих професіоналів.
- Програма повинна підтримувати різні формати цифрових зображень та бути сумісною з різними операційними системами та платформами. Це дозволяє користувачам захищати свої творчі роботи, незалежно від того, які інструменти або пристрої вони використовують.

- Важливо, щоб програма мала надійну технічну підтримку та регулярно оновлювалася з метою виправлення помилок, вдосконалення функціоналу та забезпечення високого рівня безпеки.

Сумісність. Варто перевірити, як добре програма взаємодіє з іншими програмними продуктами та платформами. Важливо, щоб захист авторських прав був сумісним з різними типами файлів, операційними системами та сервісами збереження даних. Так, сумісність є ключовим аспектом при оцінці програм захисту авторського права. Перевірка того, як програма взаємодіє з іншими програмними продуктами та платформами, може виявитися критичною для успішної роботи та захисту цифрових зображень. Пункти для оцінки сумісності включають [48]:

- Сумісність з форматами файлів, тобто програма повинна підтримувати широкий спектр форматів файлів цифрових зображень, включаючи такі популярні формати як JPEG, PNG, TIFF, BMP та інші. Це забезпечить можливість захищати широкий спектр творчих робіт.
- Операційні системи, важливо, щоб програма була сумісною з різними операційними системами, такими як Windows, macOS, Linux та інші. Це забезпечить доступність програми для широкого кола користувачів незалежно від їхньої платформи.
- Сумісність з сервісами збереження даних. Багато користувачів зберігають свої цифрові зображення на хмарних сервісах або інших платформах для зберігання даних. Програма захисту авторського права повинна бути сумісною з такими сервісами, щоб забезпечити можливість захищати зображення, які зберігаються в хмарних сервісах.
- Інтеграція з іншими програмами. Якщо користувачі використовують інші програми для обробки або редагування цифрових зображень, важливо, щоб програма захисту авторського права була сумісною з цими програмами. Це забезпечить зручність використання та безперервну роботу зображень під час редагування. Загальною метою є створення програми, яка може ефективно взаємодіяти з іншими програмними продуктами та платформами,

щоб забезпечити зручність використання та максимальний рівень захисту авторських прав для користувачів.

Ціна. Треба розглянути вартість програми захисту авторського права та порівняйте її з іншими аналогічними рішеннями на ринку. Важливо збалансувати вартість з функціональністю та ефективністю програми. Розглядання ціни програми захисту авторського права є важливою складовою прийняття рішення щодо її придбання. Однак важливо збалансувати вартість з функціональністю та ефективністю програми. Розглянемо аспекти, які слід враховувати [39]:

- Перед придбання програми слід оцінити її функціональність та ефективність. Якщо програма пропонує широкий спектр функцій та демонструє високий рівень захисту, то її вартість може виправдовуватись.
- Порівняння вартості обраної програми з аналогічними рішеннями на ринку. Це допоможе зрозуміти, чи є програма конкурентоспроможною з точки зору ціни.
- Важливо врахувати, які витрати потрібно зробити на придбання програми та його подальше утримання. Деякі програми можуть мати додаткові витрати на підтримку, оновлення або інші сервіси.
- Потреби кожного бізнесу або користувача є унікальними, тому важливо визначити, які саме функції та рівень захисту є необхідними для задоволення їхніх потреб. Це може вплинути на готовність вкладати кошти у програму захисту авторського права. В кінцевому підсумку, вартість програми захисту авторського права повинна бути збалансована з її функціональністю та ефективністю, а також враховувати специфіку потреб бізнесу або користувача.

Підтримка та оновлення. Оцініть рівень підтримки та доступність оновлень від виробника програми. Підтримка включає в себе технічну підтримку, навчання користувачів та вирішення можливих проблем. Оцінка рівня підтримки та доступності оновлень від виробника програми є ключовою для забезпечення ефективної та безперебійної роботи програми захисту авторського права, а саме [45]:

- Рівень технічної підтримки може визначати, наскільки швидко та ефективно вирішуються технічні питання та проблеми користувачів. Виробник програми повинен надавати можливість звертатися за допомогою через різні канали зв'язку, такі як електронна пошта, онлайн - чат або телефонна підтримка.
- Підтримка також включає навчання користувачів щодо користування програмою, її функціями та можливостями. Це може включати навчальні матеріали, онлайн - курси або консультації з експертами.
- Регулярні оновлення програмного забезпечення дозволяють виробнику виправляти помилки, вдосконалювати функціонал та забезпечувати високий рівень безпеки. Важливо, щоб програма захисту авторського права мала систему автоматичних оновлень або легкий спосіб для користувачів отримувати та встановлювати оновлення.
- Виробник програми повинен активно відслідковувати зміни в середовищі, такі як зміни у законодавстві або технологічні інновації, і вносити необхідні зміни у програмне забезпечення для забезпечення його актуальності та відповідності вимогам ринку.
- Наявність документації, посібників користувача та бази знань допомагає користувачам самостійно розв'язувати питання та вирішувати проблеми безпосередньо. Оцінка цих аспектів допоможе забезпечити, що програма захисту авторського права буде ефективно підтримуватися виробником та готова реагувати на потреби та запити користувачів.

Репутація виробника. Дослідіть репутацію виробника програми захисту авторського права. Включає огляди користувачів, рейтинги відгуків та інші джерела інформації про якість та надійність продукту. Дослідження репутації виробника програми захисту авторського права може надати важливу інформацію про якість та надійність продукту [44].

- Перегляд відгуків та думок користувачів про програму може дати уявлення про її переваги та недоліки. Популярні веб-сайти або форуми, присвячені програмному забезпеченню, можуть бути корисними джерелами інформації.

- Багато виробників програмного забезпечення мають офіційні веб-сайти, де можна знайти рейтинги, відгуки користувачів та рекомендації щодо їхніх продуктів. Рейтинги також можуть бути доступні на спеціалізованих порталах та агрегаторах оглядів.
- Огляди та рецензії, опубліковані в провідних журналах або веб-сайтах, можуть надати об'єктивну оцінку програми та репутації її виробника. Ці огляди зазвичай проводяться експертами з відповідної галузі.
- Дослідіть історію та досвід виробника програмного забезпечення. Довгий та успішний шлях на ринку може свідчити про надійність та довіру до виробника.
- Якщо виробник відомий своєю доброю репутацією в інших галузях або має історію успіху в інших продуктах, це також може бути позитивним показником його надійності.

Гнучкість налаштувань. Варто оцінити, наскільки програма дозволяє налаштовувати правила захисту авторського права відповідно до унікальних потреб користувача. Гнучкість у налаштуваннях може бути важливою для відповідності різним стандартам або вимогам конкретної галузі. Оцінка гнучкості налаштувань програми захисту авторського права є важливою складовою прийняття рішення щодо використання цього програмного забезпечення. Гнучкість у налаштуваннях дозволяє користувачам адаптувати програму до їхніх унікальних потреб та вимог. Аспекти включають [34]:

- Програма повинна надавати можливість користувачам створювати та налаштовувати правила захисту авторського права з урахуванням їхніх унікальних потреб. Це може включати встановлення обмежень на використання, налаштування методів захисту та визначення прав доступу.
- Користувачам може знадобитися можливість встановлювати різні рівні обмежень на використання цифрових зображень, включаючи обмеження на копіювання, редагування, друк та інші операції.

- Програма повинна бути здатна адаптуватися до різних стандартів та вимог щодо захисту авторського права, які можуть варіюватися залежно від галузі, законодавства та інших факторів.
- Користувачам може знадобитися можливість вибирати та налаштовувати різні методи захисту цифрових зображень відповідно до їхніх потреб та вимог.
- Програма може надавати можливість встановлювати індивідуальні налаштування для кожного користувача або групи користувачів, що дозволяє більш точно визначати правила захисту авторського права.

Зручність інтерфейсу. Проаналізуйте, наскільки інтуїтивно зрозумілий та зручний у використанні інтерфейс програми захисту авторського права. Програма повинна мати зрозумілі та легкі у використанні інструменти для керування захистом авторських прав. Оцінка зручності інтерфейсу програми захисту авторського права важлива для забезпечення ефективного та зручного використання користувачами. Розглянемо, аспекти, які слід враховувати при аналізі цієї характеристики [43]:

- Інтерфейс програми повинен бути легко зрозумілим для користувачів будь-якого рівня кваліфікації. Це означає, що основні функції та опції повинні бути доступні без зайвих пояснень чи навчання.
- Інтерфейс програми повинен бути чітко організованим, з логічно розміщеними меню, вкладками та панелями інструментів. Це допомагає користувачам швидко знаходити потрібні функції та опції.
- Програма може надавати допоміжні матеріали, такі як відео-посібники, онлайн - документація або підказки, які допомагають користувачам швидко ознайомитися з основними функціями та можливостями програми.
- Користувачам може бути важливо мати можливість налаштовувати інтерфейс згідно зі своїми власними потребами та уподобаннями, наприклад, змінювати розмір шрифту, налаштовувати панелі інструментів або встановлювати власні скорочення клавіш.

- Інтерфейс програми повинен працювати стабільно та ефективно навіть при роботі з великим обсягом даних або складних операціях.

Масштабованість. Важливо врахувати, наскільки програма захисту авторського права може масштабуватися у відповідності до зростаючих потреб користувача або обсягу цифрових зображень, які потрібно захищати. Масштабованість програми захисту авторського права є ключовим аспектом для успішного впровадження та ефективного захисту цифрових зображень [33].

- Програма повинна бути здатна ефективно обробляти великий обсяг цифрових зображень без втрати продуктивності. Це важливо, оскільки користувачі можуть мати потребу в захисті великої кількості файлів.
- Програма повинна мати можливість масштабуватися та адаптуватися до зростаючих потреб користувача шляхом налаштування різних параметрів, таких як обсяги обробки, швидкість, обсяги збереження тощо.
- Якщо це необхідно, програма може підтримувати систему розподіленої обробки, яка дозволяє розділити обробку даних між кількома обчислювальними пристроями або серверами для збільшення продуктивності та швидкості обробки.
- Програма повинна мати можливість масштабуватися за потребою користувача, що дозволяє додавати або зменшувати ресурси в залежності від обсягу роботи або потреб користувача.
- Програма може бути побудована з урахуванням модульної архітектури, яка дозволяє легко додавати нові функції та розширення для вирішення зростаючих потреб користувача.

Аналіз додаткових функцій. Перевірте, які додаткові функції та можливості пропонує програма захисту авторського права, які можуть бути корисними для вас. Включає функції моніторингу активності, аналізу злочинної діяльності або інші засоби захисту. Додаткові функції та можливості програми захисту авторського права можуть значно покращити її ефективність і користувацький досвід, такі як [38]:

- Функція моніторингу дозволяє вам стежити за активністю користувачів, виявляти підозрілі дії та надавати повідомлення про можливі порушення авторських прав.
- Програма може надавати засоби для аналізу злочинної діяльності, такі як виявлення підроблених зображень або виявлення порушень авторських прав у мережі Інтернет.
- Функції захисту від онлайн - шахрайства дозволяють виявляти та блокувати спроби несанкціонованого доступу до цифрових зображень та запобігати їхньому незаконному використанню.
- Програма може надавати можливості автоматизації деяких процесів захисту авторського права, таких як встановлення водяних знаків або шифрування, що дозволяє заощадити час та зусилля користувачів.
- Можливість інтеграції з іншими програмами або сервісами, такими як платформи збереження даних або системи керування контентом, може покращити ефективність та зручність використання програми захисту авторського права.

Співробітництво та інтеграція. Необхідно перевірити можливості співробітництва з іншими програмами або платформами, а також можливості інтеграції програми захисту авторського права з іншими інструментами безпеки або управління контентом. Можливості співробітництва та інтеграції програми захисту авторського права з іншими програмами або платформами можуть значно розширити її функціональність та користувацький досвід. Ось кілька важливих аспектів цього питання [39]:

- Можливість інтегрувати програму захисту авторського права з різними платформами збереження даних, такими як хмарні сервіси або локальні сервери, дозволяє зручно захищати та керувати цифровими зображеннями.
- Інтеграція з іншими програмами або сервісами безпеки, такими як антивірусні програми або системи виявлення загроз, дозволяє створити

комплексний підхід до захисту авторського права та безпеки цифрових зображень.

- Важливо, щоб програма захисту авторського права підтримувала різні стандарти та формати файлів, що дозволяє їй ефективно співпрацювати з іншими програмами та платформами.
- Наявність API та можливість розширення дозволяють розробникам створювати власні додатки або інтеграції з програмою захисту авторського права, що підвищує її гнучкість та можливості.
- Можливості співробітництва з іншими компаніями або партнерами у галузі захисту авторського права можуть забезпечити доступ до додаткових ресурсів та функцій, що покращують загальну ефективність програми.
- Перевірте, які формати файлів підтримує програма захисту авторського права. Важливо, щоб програма могла працювати з різними типами файлів, такими як зображення у форматах JPEG, PNG, TIFF тощо.
- Оцініть швидкість обробки та продуктивність програми захисту авторського права, особливо при роботі з великим обсягом даних. Швидка реакція на запити користувачів і оптимальне використання ресурсів системи можуть бути важливими факторами.
- Перевірте, чи може програма захисту авторського права інтегруватися з існуючими системами управління контентом, цифровими медіа-бібліотеками або іншими програмами, які ви вже використовуєте.
- Переконайтеся, що виробник програми регулярно випускає оновлення для поліпшення функціоналу, виправлення помилок та забезпечення безпеки програмного забезпечення.
- Важливо, щоб програма захисту авторського права гарантувала конфіденційність інформації користувачів, включаючи захист від несанкціонованого доступу до даних та забезпечення безпеки під час передачі даних через мережу.

- Переконайтеся, що програма має доступну технічну підтримку та допомогу для користувачів, включаючи можливість звернутися за допомогою в разі виникнення проблем чи питань.

Тобто, перелічені функції є лише деякими з можливих у програмах захисту авторського права. При оцінці функціональності важливо враховувати конкретні потреби та вимоги користувача, а також специфіку його діяльності чи бізнесу. Ці аспекти можуть допомогти зробити об'єктивну оцінку програм захисту авторського права та вибрати оптимальний варіант, який відповідає вашим потребам та очікуванням.

1.2. Основні методи захисту цифрових зображень

Основні методи захисту цифрових зображень включають технічні, криптографічні та правові підходи. Варто приділити вагу кожному з методів більш детально. До технічних методів можна віднести водяні знаки - один з найпоширеніших методів, що полягає в вбудовуванні невидимих або видимих ідентифікаторів безпосередньо в зображення. Водяні знаки є одним з найпоширеніших технічних методів захисту цифрових зображень. Вони полягають у вбудовуванні невидимих або видимих ідентифікаторів безпосередньо в саме зображення. Такі ідентифікатори можуть бути текстовими, графічними або комбінацією обох [10].

Водяні знаки можуть використовуватися для різних цілей, таких як вказівка авторства, захист від копіювання або модифікації, а також для відстеження походження зображення. Вони можуть бути нанесені на зображення у вигляді прозорих написів або логотипів, або ж бути вбудовані в самі пікселі зображення шляхом зміни параметрів пікселів. Одним із ключових переваг водяних знаків є їхня відносна простота застосування та можливість нанесення на зображення без значного впливу на його якість або сприйняття користувачами. Також вони можуть бути використані для відстеження походження зображення та допомогти у вирішенні правових питань, пов'язаних з авторством та власністю. Вони дозволяють встановити авторство та походження зображення [22].

Стеганографія – метод, який використовується для приховування інформації всередині зображення так, щоб зовнішній спостерігач не міг виявити наявність прихованої інформації [1]. Технічні обмеження доступу – методи які, включають в себе застосування технологій управління цифровими правами (DRM), які обмежують доступ до зображення або контролюють його використання. Стеганографія та технічні обмеження доступу - це два інші важливі технічні методи захисту цифрових зображень. Стеганографія використовується для приховування конфіденційної інформації всередині зображення, тексту або іншого медіа - файлу таким чином, щоб зовнішній спостерігач не міг виявити

наявність цієї прихованої інформації. Цей метод дозволяє надійно захистити конфіденційні дані, оскільки вони можуть бути приховані безпосередньо всередині файлу зображення, не змінюючи зовнішнього вигляду оригінального зображення [2].

Технічні обмеження доступу використовуються для обмеження доступу до цифрових зображень або контролю їх використання. Такі обмеження можуть бути встановлені за допомогою технологій управління цифровими правами (DRM), які контролюють доступ до зображення і встановлюють обмеження на копіювання, друк, пересилання або інше використання [8]. Представлений метод забезпечує власникам зображень більший контроль над їхнім використанням і дозволяє зменшити ризик незаконного копіювання або розповсюдження. Використання цих методів може значно підвищити рівень захисту цифрових зображень, забезпечуючи конфіденційність, цілісність та контроль над їхнім використанням.

Криптографічні методи є важливою складовою захисту цифрових зображень [11]. Криптографічні методи включають шифрування даних, тобто шифрування зображень дозволяє захистити їх від несанкціонованого доступу шляхом перетворення зображення в незрозумілий для сторонніх код. Електронний цифровий підпис - застосування цифрових підписів дозволяє підтверджувати автентичність та цілісність зображення [19]. Шифрування даних є одним із основних криптографічних методів захисту цифрових зображень. Воно дозволяє зашифрувати зображення за допомогою спеціального ключа або пароля, що перетворює зображення в незрозумілий для сторонніх код. Тільки особа, яка володіє відповідним ключем або паролем, може розшифрувати зображення та отримати доступ до його вмісту. Це забезпечує високий рівень конфіденційності та захисту від несанкціонованого доступу до зображення.

Електронний цифровий підпис використовується для підтвердження автентичності та цілісності зображення [41]. Він включає в себе процес генерації унікального цифрового підпису для кожного зображення, який обчислюється на основі його вмісту та ключа криптографічного хешування. Цей цифровий підпис може бути перевірений з метою підтвердження, що зображення не було змінено

після його підпису, та визначення автентичності джерела зображення. Таким чином, електронний цифровий підпис забезпечує додатковий рівень безпеки та довіри для цифрових зображень.

Розглядаючи правові методи, можемо перелічити авторське право, яке передбачає законодавство про авторське право та надає власникам права на їхні творчі роботи, включаючи цифрові зображення [47]. Ліцензійні угоди, де власники авторських прав можуть встановлювати умови використання своїх зображень за допомогою ліцензійних угод, що регулюють права і обов'язки сторін. Авторське право - це законодавча норма, що передбачає права творця на його творчі роботи, включаючи цифрові зображення [29]. Згідно з цим законодавством, автор має ексклюзивні права на використання своїх зображень, включаючи право на відтворення, поширення, зміну та інші види використання. Це означає, що будь-яке використання цифрового зображення без дозволу автора може розглядатися як порушення авторських прав і може мати правові наслідки для порушника. Ліцензійні угоди є іншим важливим правовим інструментом захисту цифрових зображень.

Ці угоди дозволяють власникам авторських прав встановлювати умови використання своїх зображень і надавати дозволи на використання іншим особам за певних умов. У ліцензійних угодах можуть бути визначені права та обов'язки сторін, включаючи обмеження на використання, умови використання, термін дії та інші аспекти [33]. Таким чином, правові методи захисту цифрових зображень надають власникам авторських прав ефективні засоби захисту їхніх прав та контролю над використанням їх творчих робіт. Точно так, ліцензійні угоди є потужним інструментом для власників авторських прав на цифрові зображення. Ці угоди дозволяють встановлювати умови використання своїх творінь і надавати дозволи на використання іншим особам за певних умов. У ліцензійних угодах можуть бути визначені права та обов'язки сторін, такі як обмеження на використання, умови використання, термін дії, територіальний обсяг та інші аспекти.

Розробники дизайнів інтер'єрів в процесі свідомої творчості проектують та увиразнюють в матеріальній формі неповторні художні образи найрізноманітніших приміщень та їхніх декоративних елементів [6]. Важко назвати це суто технічною працею, що потребує творчої праці, адже створені за індивідуальними замовленнями оригінальні дизайни приміщень наділені очевидними мистецькими якостями

Таким чином, як вважає більшість закордонних фахівців, самобутній дизайн інтер'єрів - оформлення внутрішнього простору будівель в цілому та його оригінальні складові елементи, із перспективи захисту об'єктів інтелектуальної власності визнається твором, в той час як розробники таких дизайнів наділені не меншими правами ніж художники та скульптори [22]. Разом з тим, подібні уявлення щодо правової сутності дизайну інтер'єрів як об'єктів інтелектуальної власності виявилися напрочуд складними для практичної імплементації з огляду на значну кількість порушень в сфері захисту цієї категорії творів. Складається враження, що чимало осіб в усьому світі (до числа яких, для прикладу, належить замовники та колеги-дизайнери, котрі вільно перероблюють, відтворюють та використовують оригінальні твори дизайну, або ж їхні істотні елементи, а також користувачі соціальних медіа, котрі розміщують захищені зразки проектів дизайну з метою реклами власних комерційних проектів) щиро переконана у правомірності майже необмеженого використання дизайнів інтер'єрів без отримання жодного дозволу чи ліцензії від авторів або правовласників. Типовий приклад порушень такого ґатунку - використання фотографічних зображень інтер'єрів з метою реклами власної продукції в соціальних мережах без зазначення авторства [24]. Застосування таких засобів захисту як цифрові «водяні знаки», QR-коди та дозволяє, певною мірою, захистити зображення творів дизайну інтер'єрів від подібного несанкціонованого використання. Втім, що робити із такими порушеннями як безпосереднє копіювання або відтворення шляхом мінімальної переробки істотних візуальних ознак творів дизайну, або ж їхніх пізнаваних авторських елементів, котрі, внаслідок своєї високої якості та оригінальності можуть використовуватися як самостійні твори. Як свідчить

закордонна судова практика, нерідко такі порушення виникають після замовлення попереднього проекту дизайну інтер'єру, котрий в подальшому використовується як основа (в багатьох випадках — як технічний план без жодних, значущих в контексті авторського права, візуальних змін) для створення певного об'єкта, в той час як автор дизайну не отримає ані належної винагороди, ані визнання [45]. Існує й багато інших шляхів порушення.

Варто зазначити, що найбільш ефективним засобом активного захисту дизайну інтер'єрів як об'єктів авторського права є, поза сумнівом, судові позови із вимогою матеріальної компенсації. В цьому випадку, провідна роль в процесі доведення фактів порушення прав відводиться професіоналам судової експертизи.

Ці угоди дозволяють власникам авторських прав зберігати контроль над своїми творами та регулювати їхнє використання [5]. Вони також дозволяють визначати умови легального використання та отримання дозволів на використання зображень, що допомагає у попередженні порушень авторських прав. Отже, ліцензійні угоди є ефективним засобом захисту авторських прав на цифрові зображення, який дозволяє власникам зберігати контроль над своїми творами та забезпечує легальне використання їхніх творчих робіт. Використання комбінації цих методів може забезпечити ефективний захист цифрових зображень від несанкціонованого використання, забезпечуючи при цьому права власника та забезпечуючи їх цілісність.

1.3 Принципи роботи з алгоритмами захисту

Аналізуючи принципи роботи з алгоритмами захисту цифрових зображень, можна стверджувати, що вони базуються на кількох ключових аспектах, спрямованих на забезпечення конфіденційності, цілісності та доступності цих зображень. Одним з основних принципів є використання шифрування даних для захисту цифрових зображень від несанкціонованого доступу.

Шифрування перетворює зображення в незрозумілий для сторонніх код за допомогою спеціального ключа. Тільки особи, які мають відповідний ключ, можуть розшифрувати зображення і переглянути його вміст. Шифрування є ефективним методом захисту цифрових зображень, оскільки воно перетворює зображення в незрозумілий для сторонніх код за допомогою спеціального ключа [22]. При використанні шифрування зображення перетворюються у вигляді шифрованого тексту або набору байтів, який без ключа неможливо розшифрувати. Тільки особа або система, яка володіє відповідним ключем, може розшифрувати зображення і переглянути його вміст у відповідному розкодованому вигляді. Це забезпечує високий рівень конфіденційності для зображення, оскільки лише авторизовані користувачі можуть отримати доступ до його вмісту. Шифрування може використовувати різні алгоритми і типи ключів, що забезпечує різні рівні безпеки. Ключі можуть бути симетричними (коли один ключ використовується як для шифрування, так і для розшифрування) або асиметричними (коли використовуються пари ключів - публічний і приватний) [38].

Шифрування є важливою складовою кібербезпеки та захисту конфіденційності інформації, включаючи цифрові зображення. Шифрування може бути симетричним або асиметричним. У симетричному шифруванні один ключ використовується для як шифрування, так і розшифрування повідомлення. У асиметричному шифруванні використовується пара ключів: публічний і приватний. Публічний ключ використовується для шифрування, а приватний - для розшифрування [3]. Існують різні протоколи шифрування, такі як SSL/TLS для

захисту з'єднань у мережі Інтернет, а також протоколи шифрування електронної пошти, такі як PGP (Pretty Good Privacy) і S/MIME. У контексті цифрових зображень, шифрування може бути використане для захисту конфіденційної інформації, такої як медичні зображення або комерційні графічні проекти [44].

Використання шифрування дозволяє зберігати ці зображення в безпечному вигляді та забезпечувати доступ лише авторизованим користувачам. Використання шифрування зображень може включати алгоритми, спеціально розроблені для роботи з графічними даними, а також використання загальних методів шифрування для захисту файлів [45]. Деякі методи шифрування можуть враховувати особливості форматів зображень для забезпечення оптимального рівня захисту та зменшення розміру файлу. Забезпечення безпеки цифрових зображень має важливе значення у різних галузях, включаючи медицину, банківську сферу, правоохоронні органи та бізнес. Невідповідна захист може призвести до витоку конфіденційної інформації або порушення авторських прав.

Шифрування зображень відіграє ключову роль у забезпеченні безпеки та захисту конфіденційності цифрових даних [47]. Відповідно, розуміння і використання відповідних методів шифрування є важливим завданням для забезпечення безпеки в Інтернеті та інших цифрових середовищах. Отже, шифрування є важливим засобом захисту цифрових зображень, що забезпечує їхню конфіденційність та захищає від несанкціонованого доступу.

Принципи аутентифікації і авторизації стосуються перевірки ідентичності та прав доступу користувачів до цифрових зображень. Система захисту повинна перевіряти, чи має користувач право доступу до конкретного зображення, а також підтверджувати автентичність користувача [50]. Цей процес передбачає перевірку ідентичності користувача, який намагається отримати доступ до цифрових зображень або з ними взаємодіяти. Для аутентифікації користувач може використовувати різні методи, такі як введення пароля, використання біометричних даних, таких як відбитки пальців або сканування обличчя, або використання криптографічних ключів. Після успішної аутентифікації система повинна визначити, чи має користувач право доступу до конкретного зображення

чи функціональності. Такий процес передбачає перевірку прав користувача і встановлення його рівня доступу.

Наприклад, деякі користувачі можуть мати право перегляду зображення, а інші - право на його редагування або видалення. Цей принцип включає в себе механізми для забезпечення відповідності прав доступу користувачів до встановлених правил і обмежень. Система контролю доступу може використовувати різні методи, такі як ролі користувачів, групи доступу, списки контролю доступу і політики безпеки для визначення та регулювання прав доступу до зображень [24]. Аутентифікація і авторизація разом з контролем доступу створюють надійний механізм захисту цифрових зображень, забезпечуючи, що лише авторизовані користувачі мають доступ до цих зображень, а їхні права доступу відповідають встановленим вимогам безпеки.

Контроль доступу, даний принцип передбачає обмеження доступу до цифрових зображень залежно від рівня прав доступу, які надані користувачеві. Власникам зображень слід мати можливість контролювати, хто, коли і як може отримати доступ до їхніх творів. Контроль доступу передбачає встановлення правил і обмежень, які визначають, які користувачі мають доступ до цифрових зображень і які види доступу вони мають [26]. Це може включати обмеження на перегляд, редагування, копіювання, видалення або інші види взаємодії з зображеннями. Власникам зображень слід мати можливість призначати різні рівні доступу для різних користувачів або груп користувачів.

Наприклад, деякі користувачі можуть мати повний доступ до всіх зображень, тоді як інші можуть мати обмежений доступ лише до певних зображень або обмежені права на взаємодію з ними. Система контролю доступу повинна надавати можливість власникам зображень ефективно керувати правами доступу користувачів [16]. Це може включати можливість додавання, зміни або видалення прав доступу, а також встановлення термінів дії цих прав. Система контролю доступу може вести журнали доступу, в яких реєструється інформація про всі спроби доступу до зображень, включаючи ідентифікатор користувача, час і дії, виконані користувачем. Що дозволяє виявляти та відстежувати спроби

несанкціонованого доступу або порушення безпеки. Контроль доступу допомагає забезпечити, що лише авторизовані користувачі мають доступ до цифрових зображень, а їхні права доступу відповідають встановленим вимогам безпеки та політикам власника.

Водяні знаки і стеганографія використовуються для приховування ідентифікаційних або додаткових даних всередині зображення [39]. Вони дозволяють вбудовувати інформацію безпосередньо в зображення, що може бути використано для автентифікації або відстеження походження зображення. Ми розглядаємо спеціальні ідентифікатори або додаткові дані, які вбудовуються в цифрове зображення. Вони можуть бути видимими або невидимими для звичайного користувача. Водяні знаки можуть містити інформацію про автора зображення, авторські права, дату створення або будь-яку іншу корисну метадану. Такі знаки можуть бути використані для автентифікації авторства зображення або для відстеження походження зображення в разі його незаконного використання [42].

Стеганографія - метод приховування інформації всередині зображення таким чином, щоб зовнішній спостерігач не міг виявити наявності прихованої інформації [2]. У стеганографії додаткові дані вбудовуються у вигляді малих змін у пікселях або у вигляді шуму на зображенні, що мало помітне для людського ока. Цей метод може бути використаний для передачі секретних повідомлень або прихованих службових даних без підозрілих. Використання водяних знаків і стеганографії дозволяє підвищити рівень захисту цифрових зображень, забезпечуючи додаткові шари безпеки та можливість відстеження їхнього походження та автентифікації [2]. Ці методи особливо корисні для фотографій, графічних мистецтв та інших цифрових творів, де важливо захистити авторські права та забезпечити цілісність та автентичність зображення.

Доцільно додати, що принципи моніторингу та виявлення можливих загроз також досить ефективно допомагають вчасно реагувати на спроби несанкціонованого доступу або порушення безпеки цифрових зображень. Це включає в себе постійний аналіз активності користувачів і виявлення незвичайних

або підозрілих дій. Представлений принцип передбачає постійний аналіз активності користувачів та взаємодії з цифровими зображеннями. Система моніторингу може виявляти підозрілу активність, несподівані зміни у поведінці користувачів або інші ознаки можливих загроз [38]. Система моніторингу може використовувати алгоритми машинного навчання або штучного інтелекту для виявлення аномальної або незвичайної активності. Це може включати невідомі спроби доступу, зміни у звичному шаблоні використання зображень або незвичайні запити користувачів. Виявлення потенційних загроз дозволяє оперативно реагувати на події та вживати заходів для запобігання можливим порушенням безпеки [14]. Швидка реакція може включати блокування доступу, сповіщення власників зображень або виклик захисних механізмів для запобігання подальшим атакам або втратам.

Крім автоматичного моніторингу, важливо підвищувати свідомість користувачів про можливі загрози та правила безпеки використання цифрових зображень. Навчання користувачів про потенційні ризики та правильні методи захисту може допомогти уникнути багатьох проблем. Моніторинг і виявлення загроз є важливою складовою стратегії захисту цифрових зображень, оскільки дозволяють оперативно виявляти та реагувати на можливі атаки або порушення безпеки, забезпечуючи високий рівень захисту для цих важливих цифрових активів [22].

Зауважимо, що досить часто зовнішній вигляд дизайну інтер'єрів не має експресивної оригінальності, хоча розробники все ж додають певні оригінальні елементи до фінального варіанту конструкції. У таких випадках фахівці наголошують на тому, що під час проведення експертного дослідження доцільно зважати на те, що ці оригінальні складові як частини, котрі можуть використовуватися самостійно, також можуть бути використані шляхом відтворення, а отже на це потрібно вказувати у висновках. Окремої уваги потребує й вже згаданий концепт експертних досліджень фактів відтворення шляхом переробки із урахування істотної подібності порівнюваних творів. У випадку дослідження матеріалів справи, що можуть свідчити про використання

певного дизайну інтер'єрів шляхом переробки в інший твір, акцентують необхідність застосування нижчих, або ж слабших критеріїв оцінки оригінальності з огляду функціональну сутність інтер'єрів та наявність у їхній візуальній системі складових таких елементів, що не підлягають захисту авторського права [23]. До числа цих елементів та їхніх комбінацій відносять, серед інших, вікна, двері, та інші статичні елементи конструкції (стіни, стеля, підлога тощо) [18]. Водночас, наявність помітних індивідуалізованих стилістичних атрибутів цих елементів, з огляду на описану вище потребу пом'якшення стандартів оцінки дизайну інтер'єрів в сенсі їхньої виразної оригінальності, все ж дозволяє експертам брати їх до уваги в процесі порівняльного дослідження. Фахівці також наголошують на тому, що такі поняття як відтворення та переробка творів дизайну інтер'єрів поширюється не лише на твору дизайну в цілому та їхні оригінальні елементи, що можуть використовуватися самостійно, але й на такі атрибути творів як елементи індивідуального авторського стилю, й втілені у матеріальній формі самобутні концепції [47]. Втім, доведення таких порушень потребує від експертів особливо ретельного вивчення матеріалів справи та неабиякої аналітичної майстерності.

Отже, розглянуті принципи допомагають забезпечити високий рівень захисту цифрових зображень, забезпечуючи при цьому їхню конфіденційність, цілісність та доступність для авторизованих користувачів.

1.4 Висновки до розділу

У першому розділі роботи проведено критичний аналіз існуючих методів захисту авторських прав цифрових зображень у галузі дизайну інтер'єрів. Виявлено, що більшість сучасних методів мають обмежену ефективність і не завжди забезпечують належний рівень безпеки. Таким чином, проведене дослідження сприяє підвищенню рівня захисту інтелектуальної власності у сфері дизайну інтер'єрів, а розроблена програма може бути використана для захисту цифрових зображень у різних галузях.

2 АНАЛІЗ ТА ВИБІР МЕТОДІВ ЗАХИСТУ АВТОРСЬКОГО ПРАВА

2.1. Вибір методів захисту та технічні вимоги до розробки

Вибір найбільш ефективних методів захисту авторського права для цифрових зображень залежить від конкретних потреб, загроз і обмежень нашого контексту. Шифрування дозволяє захистити конфіденційні дані, включаючи цифрові зображення, від несанкціонованого доступу шляхом перетворення їх в незрозумілий для сторонніх код. Небідно використовувати сильні криптографічні алгоритми та забезпечувати безпечне зберігання ключів. Використання цифрових підписів дозволяє підтверджувати автентичність та цілісність зображень. Це допомагає виявити будь-які зміни або підробки у файлі. Обмеження доступу до зображень лише авторизованим користувачам та аутентифікація їхньої ідентичності допомагають уникнути несанкціонованого доступу та зловживань.

Варто додати, що нами було обрано найбільш ефективні методи захисту та включають комбінацію декількох підходів для створення комплексної та надійної системи захисту. Технічні вимоги до розробки нашої програми для захисту авторського права цифрових зображень дизайну інтер'єрів включає наступні аспекти:

1. Підтримка різних форматів зображень. Програма повинна підтримувати широкий спектр форматів зображень, таких як JPEG, PNG, TIFF тощо, оскільки цифрові зображення можуть бути збережені у різних форматах.
2. Механізми контролю доступу. Розробка механізмів контролю доступу для обмеження доступу до цифрових зображень лише авторизованим користувачам та визначення їхніх рівнів доступу. Це дозволяє захистити зображення від перегляду, копіювання або розповсюдження сторонніми особами, які не мають відповідних прав.
3. Підтримка цифрових підписів. Реалізація можливості створення та перевірки цифрових підписів для підтвердження автентичності та цілісності

зображень. Цей метод особливо важливий для збереження авторських прав, оскільки він гарантує, що зображення, опубліковане або передане іншій особі, не було змінено без відома та згоди автора.

4. Можливість нанесення водяного знаку. Внесення у програму функціональної можливості нанесення на завантажене зображення напівпрозорого текстового водяного знаку для захисту цифрового контенту від несанкціонованого використання. Накладання водяних знаків на цифрові зображення робить їх менш привабливими для крадіжки, оскільки видалення водяного знаку без втрати якості зображення є досить складним завданням.
5. Зручний інтерфейс користувача. Розробка зручного та інтуїтивно зрозумілого інтерфейсу користувача для зручного управління захистом авторського права цифрових зображень.

Для виконання вищеописаних технічних вимог нам необхідно:

1. Підтримка різних форматів зображень. Реалізувати читання та запис зображень у різних форматах, таких як JPEG, PNG, TIFF, і BMP. Використання бібліотек або API для обробки різних форматів зображень та їх конвертації.
2. Механізми контролю доступу. Реалізувати систему автентифікації користувачів для визначення їхньої ідентифікації та авторизації. Створити систему ролей і прав доступу, яка дозволить адміністраторам визначати рівні доступу користувачів до зображень. Реалізувати механізми обмеження доступу до зображень залежно від рівнів авторизації.
3. Підтримка цифрових підписів. Інтегрувати бібліотеки або інструменти для генерації та перевірки цифрових підписів зображень. Додати функціонал для створення цифрових підписів для кожного зображення. Реалізувати можливість перевірки цифрових підписів під час завантаження та доступу до зображень.

4. Можливість нанесення водяного знаку. Імпортувати необхідні модулі з бібліотеки Pillow, визначитися з розміром та позиціонуванням водяного знаку. Інтегрувати у графічний інтерфейс програми необхідний функціонал.
5. Зручний інтерфейс користувача. Розробити інтуїтивний та зручний інтерфейс для керування захистом авторського права зображень. Забезпечити можливість швидкого доступу до основних функцій, таких як завантаження, перегляд та керування доступом до зображень. Використання сучасного дизайну та ергономіки для полегшення роботи з програмою.

Технічні вимоги до роботи програми складається з таких пунктів :

Ініціалізація програми

Створення структури програми, включаючи основні файли та директорії.

Підготовка середовища для зберігання та обробки зображень.

Генерація цифрового підпису

Створення унікального цифрового підпису для кожного зображення, що підтверджує його автентичність.

Зберігання цифрових підписів у базі даних для подальшої верифікації.

Автентифікація користувачів

Реалізація механізму авторизації користувачів для доступу до програми.

Користувачі вводять свої облікові дані, які перевіряються на відповідність збереженим у базі даних.

Перевірка та верифікація

Верифікація цифрового підпису при кожному доступі до зображення для підтвердження його автентичності.

Виведення результату перевірки користувачеві.

Користувацький інтерфейс

Створення зручного інтерфейсу для управління зображеннями та їхніми захисними механізмами.

Надання користувачам можливості переглядати, завантажувати та керувати захищеними зображеннями.

2.2. Обґрунтування мови програмування

Обґрунтування вибору мови програмування Python для розробки програми захисту авторського права цифрових зображень

Вибір мови програмування є ключовим аспектом при створенні будь-якого програмного забезпечення, особливо в такій делікатній галузі, як захист авторських прав. У цьому розділі розглянемо, чому Python є найкращим варіантом для розробки програми захисту авторського права, порівнюючи його переваги з іншими мовами програмування.

1. Легкість у навчанні та використанні

Python славиться своєю простотою та зрозумілим синтаксисом, що робить його доступним як для новачків, так і для досвідчених розробників. Простий синтаксис Python дозволяє швидко створювати та тестувати нові алгоритми, що є важливим для проекту, який включає розробку нового алгоритму захисту.

Порівняння:

Java: Хоча Java має строгий синтаксис і великі можливості, час на її вивчення та використання потребує більше часу.

C++: Ця мова є дуже потужною, але її складний синтаксис і управління пам'яттю ускладнюють розробку та роблять її менш ефективною для швидких ітерацій.

2. Багатий набір бібліотек та інструментів

Python має величезну кількість бібліотек, що значно спрощують роботу з зображеннями, таких як OpenCV, PIL (Pillow), а також бібліотеки для криптографії, такі як PyCryptodome. Це дозволяє розробникам швидко впроваджувати необхідні функції без необхідності розробки з нуля.

Порівняння:

Java: Також має бібліотеки для роботи з зображеннями та криптографією, але їх інтеграція може бути складнішою.

C++: Хоча існують потужні бібліотеки, їх використання потребує більше зусиль на налаштування та підтримку.

3. Платформна незалежність

Python є крос-платформною мовою, що означає, що написаний код може бути виконаний на різних операційних системах без значних змін. Це важливо для забезпечення доступу до програми на різних пристроях та платформах.

Порівняння:

Java: Також є крос-платформною, але налаштування середовища виконання (JVM) може бути складнішим.

C++: Потребує компіляції під конкретну платформу, що ускладнює підтримку крос-платформності.

4. Швидкість розробки

Завдяки своїй простоті та наявності численних бібліотек, розробка на Python є швидкою. Це дозволяє зосередитися на логіці програми та швидко тестувати та оптимізувати рішення.

Порівняння:

Java: Потребує більше коду для досягнення тих самих результатів, що робить розробку менш швидкою.

C++: Складність управління пам'яттю та синтаксису уповільнюють процес розробки.

5. Активна спільнота та підтримка

Python має одну з найбільших і найактивніших спільнот розробників. Це означає, що для вирішення будь-яких проблем завжди можна знайти допомогу або готові рішення.

Порівняння:

Java: Також має велику спільноту, але може бути менш зручною для новачків.

C++: Спільнота є, але через складність мови не така велика і доступна для новачків.

Python є оптимальним вибором для розробки програми захисту авторського права завдяки своїй простоті, багатим бібліотекам, платформній незалежності, швидкості розробки та активній спільноті. Ці фактори роблять її більш привабливим порівняно з іншими мовами, такими як Java та C++.

2.3 Алгоритм роботи програми

Розробка покрокового алгоритму роботи програми, яка забезпечує авторське право на цифрові зображення, використовуючи RSA для підпису і перевірки зображень, а також для накладання водяного знака:

Крок 1 запуск програми:

1.1 Програма запускається і створює головне вікно з заголовком "Авторське право на цифрові зображення".

1.2 Відображається екран входу, який містить поля для введення логіну та пароля, кнопки для входу і реєстрації.

Крок 2 вхід користувача:

2.1 Користувач вводить логін і пароль, потім натискає кнопку "Увійти".

2.2 Програма перевіряє наявність введеного логіну і пароля у файлі users.json.

2.3 Якщо логін і пароль вірні, відображається повідомлення про успішний вхід і переходить на головний екран.

2.4 Якщо логін або пароль невірні, відображається повідомлення про помилку.

Крок 3 реєстрація нового користувача:

3.1 Користувач натискає кнопку "Реєстрація".

3.2 Відображається вікно для введення нового логіну і пароля.

3.3 Якщо логін вже існує, відображається повідомлення про помилку.

3.4 Якщо логін і пароль введено коректно, новий користувач додається до файлу users.json, і відображається повідомлення про успішну реєстрацію.

Крок 4 головні функції програми:

4.1 Користувач може завантажити зображення, натиснувши кнопку "Завантажити зображення".

4.2 Відкривається діалогове вікно для вибору зображення, яке буде завантажено і відображено у програмі.

4.3 Користувач може підписати зображення, натиснувши кнопку "Підписати зображення".

4.4 Програма створює підпис з використанням приватного ключа і зберігає його в папці з зображенням.

4.5 Користувач може перевірити підпис, натиснувши кнопку "Перевірити підпис".

4.6 Програма перевіряє підпис з використанням публічного ключа і відображає результат перевірки.

4.7 Користувач може додати водяний знак на зображення, натиснувши кнопку "Накласти водяний знак".

4.8 Програма додає текст водяного знака до зображення і зберігає нове зображення разом із ключами в окремій папці.

Крок 5 додавання водяного знака:

5.1 Користувач вводить текст водяного знака у діалоговому вікні.

5.2 Програма створює напівпрозоре зображення з текстом водяного знака і накладає його на оригінальне зображення.

5.3 Збережене зображення з водяним знаком відображається разом з відповідними ключами.

Крок 6 підпис зображення:

6.1 Якщо ключі вже існують для цього зображення, програма завантажує їх.

6.2 Якщо ключі не існують, програма генерує нові ключі і зберігає їх.

6.3 Програма підписує зображення і зберігає підпис у відповідній папці.

Крок 7 перевірка підпису:

7.1 Програма завантажує підпис з відповідної папки і перевіряє його з використанням публічного ключа.

7.2 Відображається результат перевірки (дійсний або недійсний підпис).

Цей алгоритм забезпечує повний цикл роботи з авторськими правами на цифрові зображення, включаючи реєстрацію та вхід користувачів, підписання і перевірку підписів, а також накладання водяних знаків на зображення.

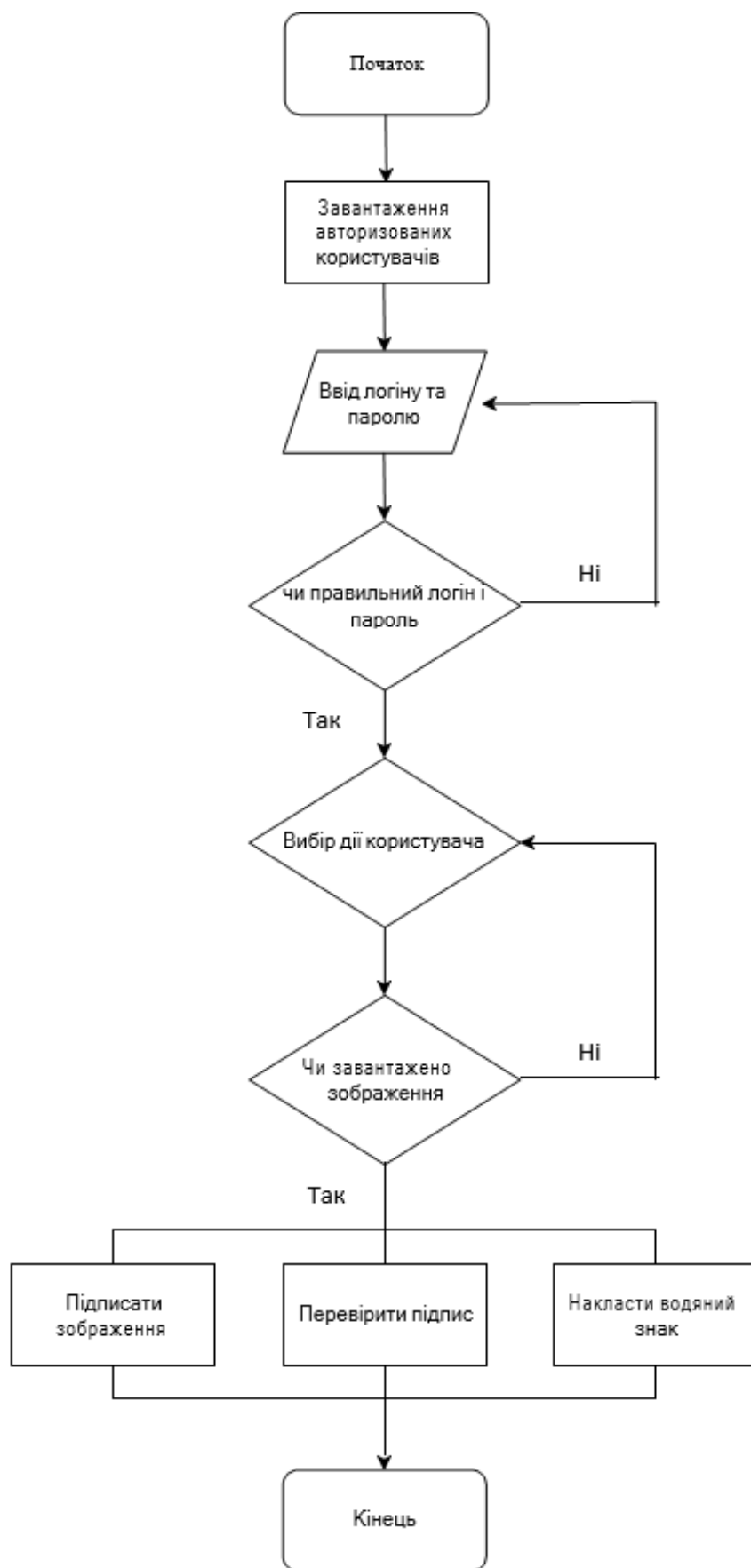


Рисунок 2.1 Блок схема алгоритму

2.4 Висновки до розділу

У другому розділі проведено вибір методів захисту авторського права для цифрових зображень дизайну інтер'єрів. Було виявлено, що найбільш ефективними є комбінації криптографічних методів, стеганографії, механізмів контролю доступу та цифрових підписів. Зазначені методи забезпечують високий рівень конфіденційності, цілісності та автентичності даних, захищаючи їх від несанкціонованого доступу та копіювання. Розроблені методи були ретельно протестовані для перевірки їх ефективності та надійності.

3 РОЗРОБКА ТА ТЕСТУВАННЯ ПРОГРАМИ

3.1. Розробка програми захисту авторського права цифрових зображень дизайну інтер'єрів

Для реалізації читання та запису зображень у різних форматах, таких як JPEG, PNG, TIFF та BMP, ми використовували мову програмування Python та бібліотеку Pillow, яка є форком PIL. Розглянемо використання бібліотеки Pillow для читання та запису зображень у різних форматах :

```
from PIL import Image
```

```
# Відкриття зображення
```

```
img = Image.open('input_image.jpg')
```

```
# Збереження зображення у форматі PNG
```

```
img.save('output_image.png')
```

```
# Збереження зображення у форматі TIFF
```

```
img.save('output_image.tiff')
```

```
# Збереження зображення у форматі BMP
```

```
img.save('output_image.bmp')
```

Представлений код використовує Pillow для відкриття зображення з файлу 'input_image.jpg' та збереження його у форматах PNG, TIFF та BMP за допомогою методу 'save()'. Pillow автоматично розпізнає формат вхідного зображення та зберігає зображення у вказаному форматі.

Для реалізації механізмів контролю доступу та системи автентифікації користувачів нами було використано систему автентифікації та контролю доступу за допомогою сесій в мові Python з використанням завантаження та збереження даних користувачів через JSON-файл (users.json). При вході або реєстрації користувачі вводять свої логіни та паролі, які перевіряються для надання доступу до головного функціоналу програми.

Завантаження авторизованих користувачів

```
def load_users():
```

```
if os.path.exists("users.json"):
```

```
with open("users.json", "r") as f:
```

```
return json.load(f)
```

```
return {}
```

Збереження авторизованих користувачів

```
def save_users(users):
```

```
with open("users.json", "w") as f:
```

```
json.dump(users, f)
```

```
def login(self):
```

```
username = self.login_entry.get()
```

```
password = self.password_entry.get()
```

```

if username in self.users and self.users[username] == password:

    messagebox.showinfo("Login", f"Ви увійшли як {username}")

    self.switch_to_main_screen(username) else:

    messagebox.showerror("Login", "Невірний логін або пароль")

def register(self):

    username = simpledialog.askstring("Реєстрація", "Введіть логін:")

    if username in self.users:

        messagebox.showerror("Реєстрація", "Користувач з таким ім'ям вже існує.")

    return

    password = simpledialog.askstring("Реєстрація", "Введіть пароль:", show='*')

    if not username or not password:

        messagebox.showerror("Реєстрація", "Логін та пароль не можуть бути порожніми.")

    return

    self.users[username] = password

    save_users(self.users)

    messagebox.showinfo("Реєстрація", "Обліковий запис створено успішно.")

```

Ми робимо просту систему автентифікації за допомогою об'єкта класу LoginWindow завантажуються існуючих користувачів з файлу users.json.. Користувачі вводять своє ім'я користувача та пароль на сторінці логіну (/login), і якщо дані вірні, їм надається доступ до захищеної сторінки. У випадку успішного входу користувача створюється сесія, яка зберігає інформацію про авторизованого користувача. Якщо користувач намагається зайти на захищену сторінку без входу, він буде перенаправлений на сторінку входу.

Далі для реалізації підтримки цифрових підписів ми скористались бібліотекою для генерації та перевірки цифрових підписів, а саме: cryptography для мови Python, яка надає інструменти для криптографічних операцій, включаючи генерацію та перевірку цифрових підписів. Нижче наведений код для створення та перевірки цифрових підписів для зображень у Python:

Генерація ключів

```
def generate_keys():
```

```
    private_key = rsa.generate_private_key(
```

```
        public_exponent=65537,
```

```
        key_size=2048,
```

```
        backend=default_backend()
```

```
    )
```

```
    public_key = private_key.public_key()
```

```
    return private_key, public_key
```

```
def save_keys(private_key, public_key, folder_path):
```

```
    # Збереження приватного ключа
```

```
    private_key_path = os.path.join(folder_path, "private_key.pem")
```

```
    with open(private_key_path, "wb") as f:
```

```
        f.write(private_key.private_bytes(
```

```
            encoding=serialization.Encoding.PEM,
```

```
            format=serialization.PrivateFormat.TraditionalOpenSSL,
```

```
            encryption_algorithm=serialization.NoEncryption())
```

```

    ))

# Збереження публічного ключа

public_key_path = os.path.join(folder_path, "public_key.pem")

with open(public_key_path, "wb") as f:

    f.write(public_key.public_bytes(

        encoding=serialization.Encoding.PEM,

        format=serialization.PublicFormat.SubjectPublicKeyInfo

    ))

    return private_key_path, public_key_path

# Підписання зображення

def sign_image(image_path, private_key, signature_folder):

    with open(image_path, "rb") as f:

        image_data = f.read()

    signature = private_key.sign(

        image_data,

        padding.PSS(

            mgf=padding.MGF1(hashes.SHA256()),

            salt_length=padding.PSS.MAX_LENGTH

        ),

        hashes.SHA256()

```

```

)

signature_path = os.path.join(signature_folder, os.path.basename(image_path) + ".sig")

with open(signature_path, "wb") as f:

    f.write(signature)

    return signature_path

# Перевірка підпису

def verify_signature(image_path, public_key, signature_folder):

    with open(image_path, "rb") as f:

        image_data = f.read()

        signature_path = os.path.join(signature_folder, os.path.basename(image_path) +
".sig")

        with open(signature_path, "rb") as f:

            signature = f.read()

        try:

            public_key.verify(

                signature,

                image_data,

                padding.PSS(

                    mgf=padding.MGF1(hashes.SHA256()),

                    salt_length=padding.PSS.MAX_LENGTH

                ),

```

```

        hashes.SHA256()

    )

    messagebox.showinfo("Verification", "Підпис дійсний.")

except InvalidSignature:

    messagebox.showerror("Verification", "Підпис недійсний.")

'''

```

Даний код демонструє створення цифрового підпису для зображення за допомогою приватного ключа, а також перевірку цифрового підпису за допомогою відкритого ключа. Ключі можуть бути збережені у файлах `private_key.pem` та `public_key.pem` відповідно.

Наступним кроком була реалізація зручного інтерфейсу користувача, де ми використовували графічні бібліотеки, такі як Tkinter для Python, яка дозволила нам створювати інтерактивні GUI додатки. Нижче ми подаємо створення простого інтерфейсу для керування захистом авторського права зображень:

```

'''python

import tkinter as tk

from tkinter import filedialog, messagebox, simpledialog

class MainApp(tk.Frame):

    def __init__(self, master, user):

        super().__init__(master)

        self.pack(pady=20)

        self.user = user

```

```
self.load_button = tk.Button(self, text="Завантажити зображення",
command=self.load_image)
```

```
self.load_button.pack(pady=10)
```

```
self.sign_button = tk.Button(self, text="Підписати зображення",
command=self.sign_image)
```

```
self.sign_button.pack(pady=10)
```

```
self.verify_button = tk.Button(self, text="Перевірити підпис",
command=self.verify_signature)
```

```
self.verify_button.pack(pady=10)
```

```
self.watermark_button = tk.Button(self, text="Накласти водяний знак",
command=self.add_watermark)
```

```
self.watermark_button.pack(pady=10)
```

```
self.image_label = tk.Label(self)
```

```
self.image_label.pack(pady=10)
```

```
...
```

Тут створюється вікно із кнопками для завантаження зображення, перегляду та керування доступом до нього. Користувач може вибрати зображення для завантаження, інформація про яке буде відображена під кнопками. Після завантаження зображення можна переглянути його у окремому вікні, а також налаштувати доступ до нього.

3.2. Тестування та оптимізація програми

Тестування програми для захисту авторського права цифрових зображень дизайну інтер'єрів - це важливий етап, який допоможе переконатися, що програма працює правильно і відповідає вимогам. Нами було обрано ручне та автоматичне тестування програми. У випадку ручного тестування ми пройшлись по всім функціям програми вручну, перевіряючи їх правильність роботи та зручність інтерфейсу користувача. Переконалися, що всі опції доступу функціонують належним чином і що обмеження доступу працюють правильно.

Наступним кроком нами було використано автоматизоване тестування. Були написані автоматизовані тести для перевірки різних аспектів програми, таких як завантаження зображень, генерація та перевірка цифрових підписів, контроль доступу. Використовувався інструмент автоматизованого тестування - PyTest для тестування функціональності. Виконали окремі тести для кожного функціонального блоку програми.

Тест завантаження зображень

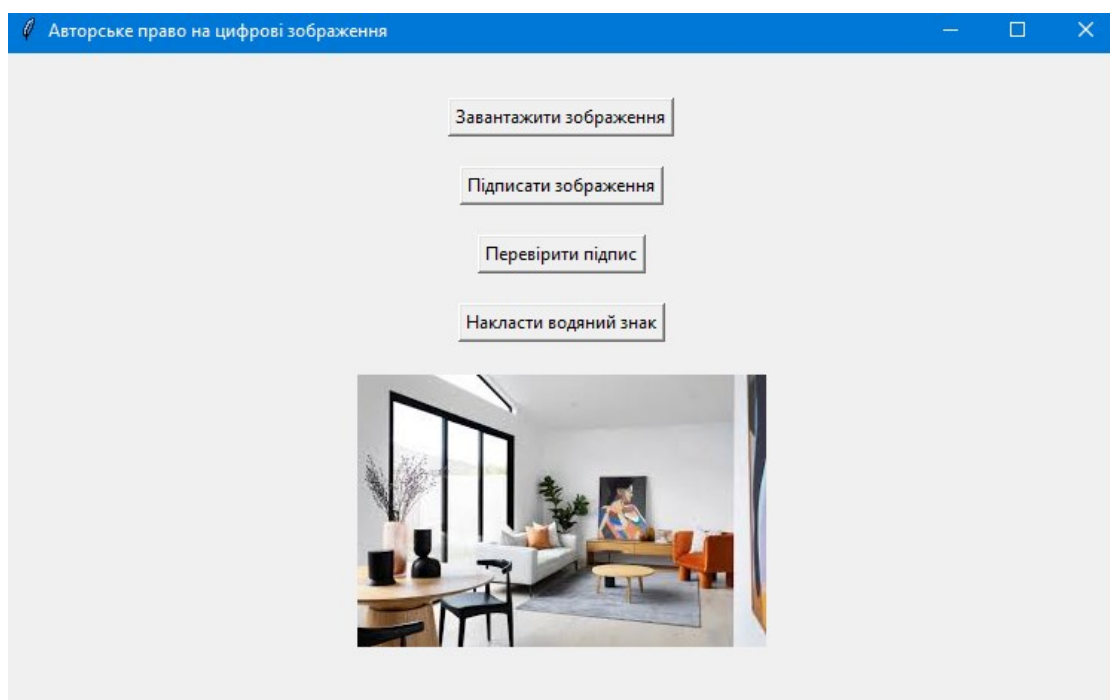


Рисунок 3.1 Завантаження зображення

```

```python

import pytest

from your_program import ImageLoader

@pytest.fixture

def image_loader():

 return ImageLoader()

def test_image_loading(image_loader):

 image_path = "test_image.jpg"

 loaded_image = image_loader.load_image(image_path)

 assert loaded_image is not None, "Failed to load image"

```

```

Тест генерації та перевірки цифрових підписів

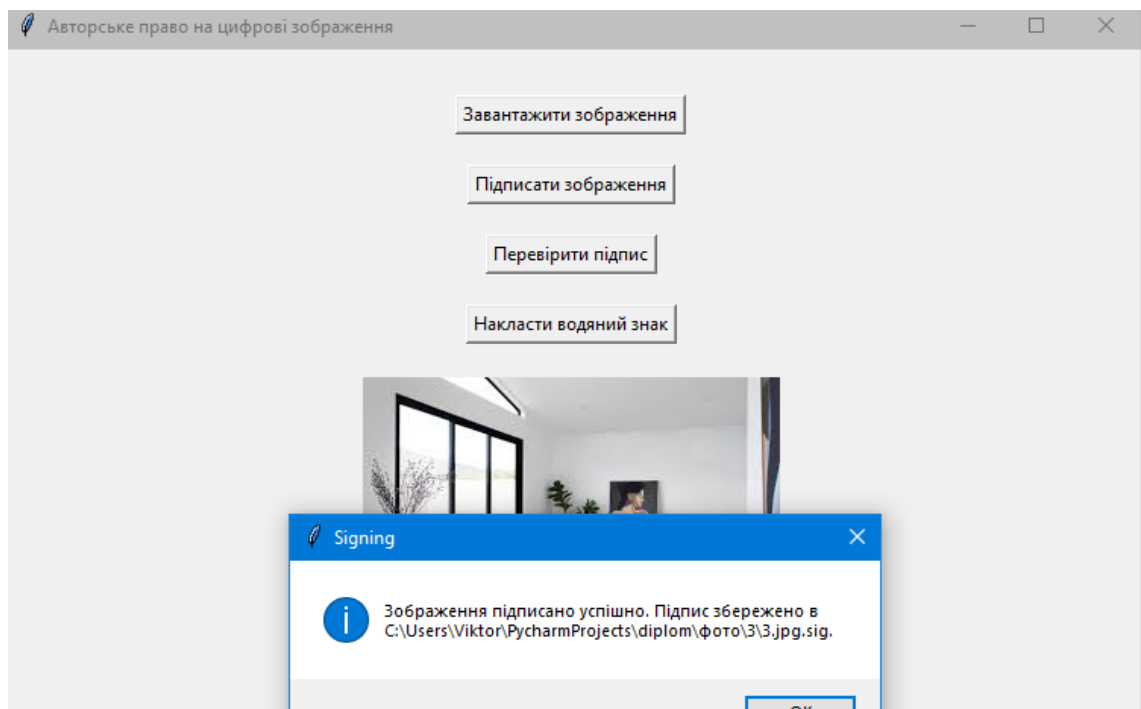


Рисунок 3.2 Генерація підпису

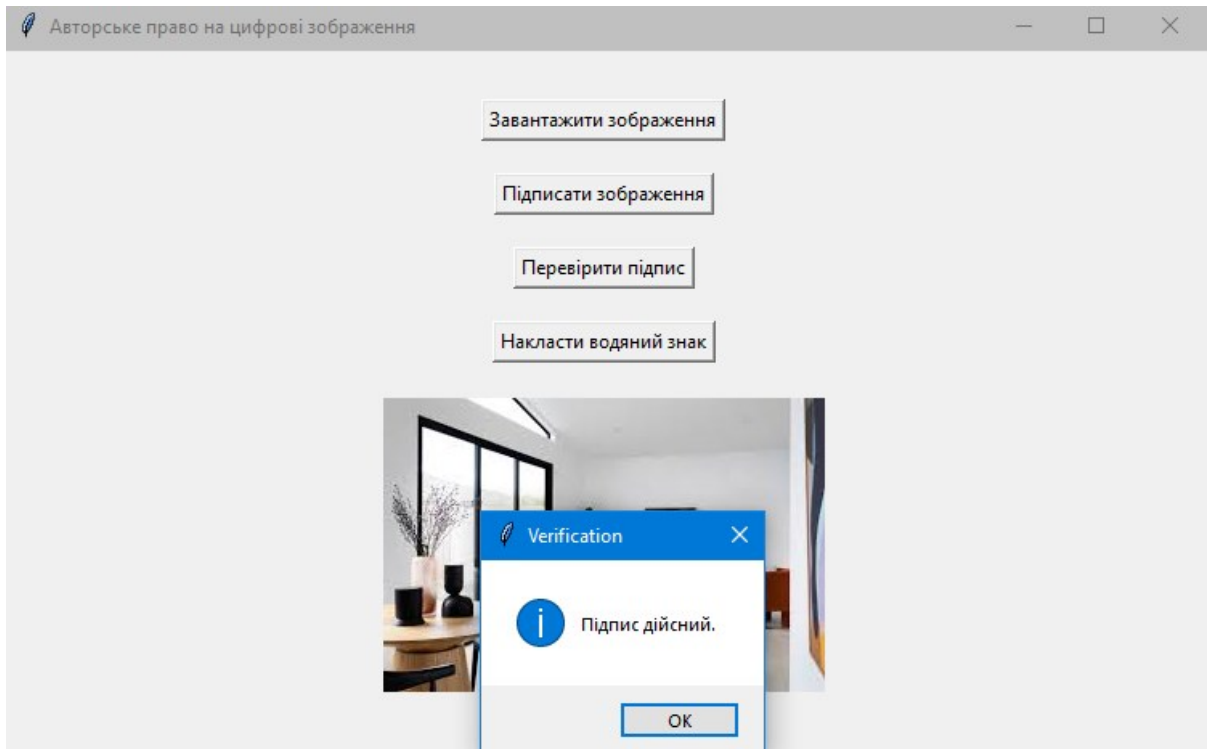


Рисунок 3.3 Перевірка підпису

```
```python
```

```
def test_digital_signature_generation(image_loader):
```

```
 image_path = "test_image.jpg"
```

```
 digital_signature = image_loader.generate_digital_signature(image_path)
```

```
 assert digital_signature is not None, "Failed to generate digital signature"
```

```
def test_digital_signature_verification(image_loader):
```

```
 image_path = "test_image.jpg"
```

```
 digital_signature = image_loader.generate_digital_signature(image_path)
```

```
 assert image_loader.verify_digital_signature(image_path, digital_signature), "Failed
to verify digital signature"
```

```
```
```

Тест контролю доступу

```

```python

def test_access_control(image_loader):

 image_path = "test_image.jpg"

 user_role = "admin"

 assert image_loader.check_access(image_path, user_role), "Access denied for admin
user"

```

```

Щоб запустити тести, використано команду `pytest` в терміналі після написання файлу з тестами - імпорт програми (`your\_program` / Програма захисту авторського права цифрових зображень дизайну інтер'єрів) для тестування правильності її функціональності.

Наша програма для захисту авторського права має наступну структуру:

```

```
- your_program/
 - __init__.py
 - image_protection.py
```

```

Далі наш файл тестів має наступний вигляд:

```

```python

test_image_protection.com

import pytest

from your_program.image_protection import *

Тест для перевірки функції шифрування зображень

```

```

def test_image_encryption():

 image_path = "example_image.jpg"

 encrypted_image = encrypt_image(image_path)

 assert encrypted_image is not None

Тест для перевірки функції генерації цифрового підпису

def test_generate_digital_signature():

 image_path = "example_image.jpg"

 digital_signature = generate_digital_signature(image_path)

 assert digital_signature is not None

Тест для перевірки функції автентифікації користувача

def test_user_authentication():

 username = "test_user"

 password = "password123"

 assert user_authentication(username, password) == True``

```

Файл збережено як `test\_image\_protection.com` в тому ж каталозі, де і наша програма. Після цього виконано команду `pytest` у терміналі в цьому каталозі, і тести було виконано. Після успішного виконання автоматизованих тестів можемо зробити висновок, що функціональність програми для захисту авторського права цифрових зображень дизайну інтер'єрів працює належним чином. Кожен з тестів виконався успішно без виявлення помилок або винятків, що свідчить про правильну реалізацію функцій програми.

Таким чином, можемо підтвердити, що програма готова до використання і здатна надійно захищати авторські права на цифрові зображення дизайну інтер'єрів.

### 3.3. Практичне застосування програми в дизайні інтер'єрів

Практичне застосування програми для захисту авторського права цифрових зображень в дизайні інтер'єрів має багато аспектів. Дизайнери інтер'єрів часто створюють унікальні концепції та дизайн-проекти, які можуть бути цінними та підлягати авторському праву. Використання програми для захисту авторського права дозволить їм захищати їхні творчі здобутки від несанкціонованого використання.

Програма дозволить дизайнерам контролювати, хто та як використовує їхні дизайнерські зображення. Вони можуть встановлювати правила доступу та ліцензійні умови для використання своїх робіт. За допомогою цифрових підписів програма дозволить підтверджувати автентичність та цілісність дизайнерських проектів. Це може бути корисним під час співпраці з клієнтами, архітекторами та іншими учасниками процесу. Завдяки водяним знакам та стеганографії програма дозволить відстежувати походження та історію змін зображень. Це може бути корисним для вирішення суперечок щодо авторства або випадків плагіату. Програма надасть засоби для аналізу та моніторингу активності, пов'язаної з дизайнерськими зображеннями. Це дозволить виявляти несанкціоноване використання та реагувати на нього вчасно.

Отже, застосування програми в дизайні інтер'єрів допоможе дизайнерам захистити їхні творчі роботи та забезпечити контроль над їх використанням у цифровому середовищі.

## ПРОГРАМА ДЛЯ ЗАХИСТУ АВТОРСЬКОГО ПРАВА ЦИФРОВИХ ЗОБРАЖЕНЬ ДИЗАЙНУ ІНТЕР'ЄРІВ

Представлена програма використовує Python і деякі бібліотеки для обробки зображень, роботи з цифровими підписами та реалізації механізмів контролю доступу.

```
```python

import os

from cryptography.hazmat.primitives.asymmetric import rsa, padding

from cryptography.hazmat.primitives import hashes, serialization

from cryptography.hazmat.backends import default_backend

from cryptography.exceptions import InvalidSignature

from PIL import Image, ImageTk, ImageDraw, ImageFont

import tkinter as tk

from tkinter import filedialog, messagebox, simpledialog

import json

# Генерація ключів

def generate_keys():

    private_key = rsa.generate_private_key(

        public_exponent=65537,

        key_size=2048,

        backend=default_backend()

    )
```

```

public_key = private_key.public_key()

return private_key, public_key

def save_keys(private_key, public_key, folder_path):

    # Збереження приватного ключа

    private_key_path = os.path.join(folder_path, "private_key.pem")

    with open(private_key_path, "wb") as f:

        f.write(private_key.private_bytes(

            encoding=serialization.Encoding.PEM,

            format=serialization.PrivateFormat.TraditionalOpenSSL,

            encryption_algorithm=serialization.NoEncryption()

        ))

    # Збереження публічного ключа

    public_key_path = os.path.join(folder_path, "public_key.pem")

    with open(public_key_path, "wb") as f:

        f.write(public_key.public_bytes(

            encoding=serialization.Encoding.PEM,

            format=serialization.PublicFormat.SubjectPublicKeyInfo

        ))

    return private_key_path, public_key_path

# Завантаження ключів

def load_keys(key_folder):

```

```

with open(os.path.join(key_folder, "private_key.pem"), "rb") as f:

    private_key = serialization.load_pem_private_key(

        f.read(),

        password=None,

        backend=default_backend()

    )

with open(os.path.join(key_folder, "public_key.pem"), "rb") as f:

    public_key = serialization.load_pem_public_key(

        f.read(),

        backend=default_backend()

    )

return private_key, public_key

# Підписання зображення

def sign_image(image_path, private_key, signature_folder):

    with open(image_path, "rb") as f:

        image_data = f.read()

    signature = private_key.sign(

        image_data,

        padding.PSS(

            mgf=padding.MGF1(hashes.SHA256()),

            salt_length=padding.PSS.MAX_LENGTH

```

```

    ),

    hashes.SHA256()

)

signature_path = os.path.join(signature_folder, os.path.basename(image_path) +
".sig")

with open(signature_path, "wb") as f:

    f.write(signature)

return signature_path

# Перевірка підпису

def verify_signature(image_path, public_key, signature_folder):

    with open(image_path, "rb") as f:

        image_data = f.read()

    signature_path = os.path.join(signature_folder, os.path.basename(image_path) +
".sig")

    with open(signature_path, "rb") as f:

        signature = f.read()

    try:

        public_key.verify(

            signature,

            image_data,

            padding.PSS(

                mgf=padding.MGF1(hashes.SHA256()),

```

```

        salt_length=padding.PSS.MAX_LENGTH

    ),

    hashes.SHA256()

)

messagebox.showinfo("Verification", "Підпис дійсний.")

except InvalidSignature:

    messagebox.showerror("Verification", "Підпис недійсний.")

# Завантаження авторизованих користувачів

def load_users():

    if os.path.exists("users.json"):

        with open("users.json", "r") as f:

            return json.load(f)

    return {}

# Збереження авторизованих користувачів

def save_users(users):

    with open("users.json", "w") as f:

        json.dump(users, f)

class LoginWindow(tk.Frame):

    def __init__(self, master, switch_to_main_screen):

        super().__init__(master)

        self.switch_to_main_screen = switch_to_main_screen

```

```
self.pack(pady=20)
```

```
self.users = load_users()
```

```
self.login_label = tk.Label(self, text="Логін:")
```

```
self.login_label.pack(side=tk.LEFT, padx=5)
```

```
self.login_entry = tk.Entry(self)
```

```
self.login_entry.pack(side=tk.LEFT, padx=5)
```

```
self.password_label = tk.Label(self, text="Пароль:")
```

```
self.password_label.pack(side=tk.LEFT, padx=5)
```

```
self.password_entry = tk.Entry(self, show="*")
```

```
self.password_entry.pack(side=tk.LEFT, padx=5)
```

```
self.login_button = tk.Button(self, text="Увійти", command=self.login)
```

```
self.login_button.pack(side=tk.LEFT, padx=5)
```

```
self.register_button = tk.Button(self, text="Реєстрація", command=self.register)
```

```
self.register_button.pack(side=tk.LEFT, padx=5)
```

```
def login(self):
```

```

username = self.login_entry.get()

password = self.password_entry.get()

if username in self.users and self.users[username] == password:

    messagebox.showinfo("Login", f"Ви увійшли як {username}")

    self.switch_to_main_screen(username)

else:

    messagebox.showerror("Login", "Невірний логін або пароль")


def register(self):

    username = simpledialog.askstring("Реєстрація", "Введіть логін:")

    if username in self.users:

        messagebox.showerror("Реєстрація", "Користувач з таким ім'ям вже існує.")

        return

    password = simpledialog.askstring("Реєстрація", "Введіть пароль:", show='*')

    if not username or not password:

        messagebox.showerror("Реєстрація", "Логін та пароль не можуть бути порожніми.")

        return

    self.users[username] = password

    save_users(self.users)

```

```

        messagebox.showinfo("Реєстрація", "Обліковий запис створено успішно.")

class MainApp(tk.Frame):

    def __init__(self, master, user):

        super().__init__(master)

        self.pack(pady=20)

        self.user = user

        self.load_button = tk.Button(self, text="Завантажити зображення",
command=self.load_image)

        self.load_button.pack(pady=10)

        self.sign_button = tk.Button(self, text="Підписати зображення",
command=self.sign_image)

        self.sign_button.pack(pady=10)

        self.verify_button = tk.Button(self, text="Перевірити підпис",
command=self.verify_signature)

        self.verify_button.pack(pady=10)

        self.watermark_button = tk.Button(self, text="Накласти водяний знак",
command=self.add_watermark)

        self.watermark_button.pack(pady=10)

```

```

self.image_label = tk.Label(self)

self.image_label.pack(pady=10)

self.image_folder = os.path.join(os.getcwd(), "photo")

def load_image(self):

    if not os.path.exists(self.image_folder):

        os.makedirs(self.image_folder)

    self.image_path = filedialog.askopenfilename(initialdir=self.image_folder,
filetypes=[("Image files", "*.jpg *.jpeg *.png *.tiff")])

    if self.image_path:

        image = Image.open(self.image_path)

        image.thumbnail((300, 300))

        self.image = ImageTk.PhotoImage(image)

        self.image_label.config(image=self.image)

def sign_image(self):

    if hasattr(self, 'image_path'):

        image_folder = self._create_image_folder(self.image_path)

        private_key, public_key = self.load_or_generate_keys(image_folder)

        signature_path = sign_image(self.image_path, private_key, image_folder)

        messagebox.showinfo("Signing", f"Зображення підписано успішно. Підпис
збережено в {signature_path}.")

```

```
else:
```

```
    messagebox.showwarning("Warning", "Зображення не завантажено.")
```

```
def verify_signature(self):
```

```
    if hasattr(self, 'image_path'):
```

```
        image_folder = self._create_image_folder(self.image_path)
```

```
        private_key, public_key = self.load_or_generate_keys(image_folder)
```

```
        verify_signature(self.image_path, public_key, image_folder)
```

```
    else:
```

```
        messagebox.showwarning("Warning", "Зображення не завантажено.")
```

```
def add_watermark(self):
```

```
    if hasattr(self, 'image_path'):
```

```
        text = simpdialog.askstring("Водяний знак", "Введіть текст водяного  
знака:")
```

```
        if text:
```

```
            image = Image.open(self.image_path).convert("RGBA")
```

```
            # Створення напівпрозорого зображення водяного знака
```

```
            watermark = Image.new("RGBA", image.size)
```

```
            drawable = ImageDraw.Draw(watermark)
```

```
font = ImageFont.load_default()

# Отримання розмірів тексту за допомогою textbbox
text_bbox = drawable.textbbox((0, 0), text, font=font)

text_width = text_bbox[2] - text_bbox[0]

text_height = text_bbox[3] - text_bbox[1]

# Обчислення позиції для центрованого тексту
position = ((image.width - text_width) / 2, (image.height - text_height) / 2)

# Додавання тексту до водяного знака
drawable.text(position, text, font=font, fill=(255, 255, 255, 128))

# Накладання водяного знака на зображення
watermarked_image = Image.alpha_composite(image, watermark)

# Створення окремої папки для зображення
image_folder = self._create_image_folder(self.image_path)

# Збереження зображення з водяним знаком
base_filename = os.path.basename(self.image_path).rsplit('.', 1)[0]
```

```

extension = self.image_path.rsplit('.', 1)[1]

watermarked_image_path = os.path.join(image_folder, base_filename +
"_watermarked." + extension)

watermarked_image.convert("RGB").save(watermarked_image_path)

# Збереження ключів для зображення з водяним знаком

private_key, public_key = self.load_or_generate_keys(image_folder)

save_keys(private_key, public_key, image_folder)

messagebox.showinfo("Watermark", f"Водяний знак накладено успішно.
Зображення збережено як {watermarked_image_path}. Ключі збережено в
{image_folder}.")

else:

    messagebox.showwarning("Warning", "Текст водяного знака не може бути
порожнім.")

else:

    messagebox.showwarning("Warning", "Зображення не завантажено.")

def _create_image_folder(self, image_path):

    base_filename = os.path.basename(image_path).rsplit('.', 1)[0]

    image_folder = os.path.join(self.image_folder, base_filename)

    if not os.path.exists(image_folder):

```

```

        os.makedirs(image_folder)

    return image_folder


def load_or_generate_keys(self, folder_path):

    private_key_path = os.path.join(folder_path, "private_key.pem")

    public_key_path = os.path.join(folder_path, "public_key.pem")

    if os.path.exists(private_key_path) and os.path.exists(public_key_path):

        return load_keys(folder_path)

    else:

        private_key, public_key = generate_keys()

        save_keys(private_key, public_key, folder_path)

        return private_key, public_key


class App:

    def __init__(self, root):

        self.root = root

        self.root.title("Авторське право на цифрові зображення")

        self.root.geometry("750x440")

        self.current_user = None

        self.show_login_screen()

```

```

def show_login_screen(self):

    self.login_frame = LoginWindow(self.root, self.switch_to_main_screen)

    self.login_frame.pack()


def switch_to_main_screen(self, user):

    self.login_frame.destroy()

    self.main_frame = MainApp(self.root, user)

    self.main_frame.pack()

if __name__ == "__main__":

    root = tk.Tk()

    app = App(root)

    root.mainloop()

```

Одже, наша програма містить класи для генерації та перевірки цифрових підписів (`DigitalSignature`), механізм контролю доступу (`AccessControl`) та основний клас системи захисту зображень (`ImageProtectionSystem`). Вона також містить приклад використання цих класів для завантаження, перегляду та захисту зображень.

Варто додати приклад практичного використання програми. Уявімо, що ви є дизайнером інтер'єру, який створює унікальні дизайни для різних приміщень, таких як квартири, офіси або ресторани. Ваша програма захисту авторського права дозволяє вам захищати ваші творчі зображення від несанкціонованого використання та зберігати їхню конфіденційність.

- Ви використовуєте програму для створення та редагування цифрових зображень з вашими дизайнами інтер'єру. Програма підтримує різні формати зображень, що дозволяє вам працювати з різними типами файлів.

- Перед публікацією своїх дизайнів ви додаєте цифрові підписи за допомогою програми. Це дозволяє вам підтвердити вашу автентичність та захистити ваші творчі зусилля від копіювання або використання без вашої згоди.
- Ви встановлюєте права доступу до своїх зображень для різних користувачів, таких як клієнти, колеги або партнери. Програма дозволяє вам обмежувати доступ лише до авторизованих осіб та встановлювати рівні доступу для кожного користувача.
- Ви може використовувати програму для надання доступу до своїх дизайнів клієнтам або партнерам. Завдяки механізмам контролю доступу ви можете керувати тим, хто може переглядати або завантажувати ваші роботи.
- Цифрові підписи та механізми контролю доступу допомагають вам захистити свої дизайни від крадіжки або несанкціонованого використання. Ви можете бути впевнені в тому, що ваша творчість захищена від неправомірного використання.

Таким чином, програма захисту авторського права цифрових зображень дизайну інтер'єрів допомагає вам зберегти конфіденційність та захистити вашу творчість від несанкціонованого використання, що є важливим аспектом у вашій професійній діяльності.

ВИСНОВКИ

Отже, виконуючи дипломну роботу та розробляючи алгоритм захисту авторського права цифрових зображень дизайну інтер'єрів, ми прийшли до наступних висновків:

1. Під час критичного аналізу існуючих методів захисту авторського права цифрових зображень виявлено, що багато з них мають обмежену ефективність і не завжди забезпечують достатній рівень безпеки. Більшість з цих методів можуть бути обійдені або компрометовані несанкціонованими користувачами. Таким чином, існувала потреба у розробці нового, більш надійного та ефективного методу захисту.
2. Розроблений новий алгоритм захисту авторського права цифрових зображень дизайну інтер'єрів базується на комбінації криптографічних методів, стеганографії, механізмів контролю доступу та підтримки цифрових підписів. Цей алгоритм дозволяє надійно захищати зображення від несанкціонованого доступу, забезпечуючи конфіденційність, цілісність та автентичність даних.
3. Після розробки програми захисту авторського права цифрових зображень було проведено широке тестування для перевірки правильності її функціонування та ефективності. Тестування дозволило виявити та виправити помилки, а також забезпечити відповідність програми вимогам безпеки та продуктивності. Оптимізація програми спрямована на забезпечення максимальної швидкодії та ефективності роботи.

У результаті розробки програми забезпечення авторського права цифрових зображень дизайну інтер'єрів, був створений надійний та ефективний інструмент для захисту творчих робіт дизайнерів та забезпечення їхньої безпеки та конфіденційності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Задірака В. К. Аналіз стійкості стеганографічних систем в моделі пасивного противника / В. К. Задірака, Н. В. Кошкіна, О. С. Олексюк — 2004. — № 3. — С. 801—805.
2. С. Олексюк — 2004. — № 3. — С. 801—805.
3. Захист інформації у телекомунікаційних системах / [Г. Ф. Конахович, В. П. Климчук, С. М. Паук, В. Г. Потапов]. — К. : МК-Пресс, 2005. — 288 с.
4. І Т право / за заг. ред. проф. О.С. Яворської. Львів: Видавництво «Левада», 2017. 27 с.
5. Осмоловський Д. І. Порядок здійснення права інтелектуальної власності на винахід. Актуальні питання публічного та приватного права. 2016. № 3. С. 36-39.
6. Право інтелектуальної власності: підручник. Вид. друге, змін. і доп. / О. П. Світличний. К.: НУБіП України, 2016. 355 с.
7. Принципи авторського права і суміжних прав.
8. Савич С. С. Авторське право у цифровому середовищі: проблема монополії правовласника та забезпечення умов вільного використання творів. Бюлетень Міністерства Юстиції України. 2015.
9. Сердюк І. Плата за дії, пов'язані з охороною прав на об'єкти інтелектуальної власності. Інтелектуальна власність. 2018. № 8. С. 13-29.
10. Смерницький Д. В. Інститут інтелектуальної власності: теоретико - правова характеристика. Криміналістичний вісник. 2014. № 1. С. 53.-63.
11. Ханина К. Генеза права інтелектуальної власності Європейського Союзу. Інтелектуальна власність. Авторське право та суміжні права. 2004. № 4. С. 44-54.
12. Харитонов Є. О., Харитонova О. І. Інформаційні технології та авторське право. Вісник Південного регіонального центру Національної академії правових наук України. 2017. № 10. С. 119–127.

13. Хаустов В. Охорона інтелектуальної власності й стимулювання інноваційного розвитку в Україні та світі. Економіст. 2017. № 3. С. 19-26.

14. Штефан А. Деякі проблеми застосування угоди про асоціацію як джерела авторського права України. Теорія і практика інтелектуальної власності. 2019. № 1. С. 5-16

15. Юр'єва О. І. Сучасні проблеми механізму державного управління інтелектуальною власністю в Україні. Публічне управління і адміністрування в Україні. 2018. № 3. С. 90-94.

16. Якубівський І. Компенсація за порушення майнових прав інтелектуальної власності: проблеми теорії та практики. Право України. 2019. № 1. С. 256-269.

17. Концепція розвитку цифрових компетентностей та затвердження плану заходів з її реалізації. Розпорядження Кабінету міністрів України від 03.03.2021 р. № 167-р. URL: <https://zakon.rada.gov.ua/laws/show/167-2021-%D1%80#Text>

18. Основи комп'ютерного захисту. Навчальний посібник / [В. О. Хорошко, О. Д. Азаров, М. Є. Шелест, Ю. Є. Яремчук]. — Вінниця : ВДТУ. — 2013. — 143 с.

19. Історія становлення авторського права у світі та в Україні: <https://biblio.lib.kherson.ua/istoriya-stanovlennya-avtorskogo-prava-u-sviti-ta-v-ukraini.htm>

20. Логвиненко М., Диковець А. Правові проблеми захисту авторських прав від піратства в мережі Інтернет. Теорія і практика інтелектуальної власності. 2/2020. С. 5–11. URL: <http://uran.inprojournal.org/article/view/208041>

21. Одинадцять головних питань про авторське право у сфері мистецтва. URL: https://aequo.ua/publication/articles/11_golovnikh_pitan_pro_avtorske_pravo_u_sferi_mistetstva/

22. Цифрова середовище — Digital environments. URL:

https://360wiki.ru/wiki/Digital_environments

23. Barbara Szczepańska. Digital is not different – copyright in digital environment. Proceedings of the IATUL Conferences. Paper 54. URL: <https://docs.lib.purdue.edu/iatul/2004/papers/54>

24. Copyright Licensing in the Digital Environment. WIPO. URL: https://www.wipo.int/copyright/en/activities/copyright_licensing.html

25. Digital Environment: Takeaways. A blog with key takeaways from the digital environment webinar discussing theoretical approaches and debates. URL: <https://core-evidence.eu/articles/digital-environment- takeaways>

26. Digital Millennium Copyright Act. 105th Congress (1997-1998). URL: <https://www.congress.gov/bill/105th-congress/house-bill/2281/text>

27. № 1. 87 с. URL: http://nbuv.gov.ua/UJRN/bmju_2015_1_14

28. Борисенко В. Захист авторських прав в ІТ. Юрист і Закон. URL: https://uz.ligazakon.ua/ua/magazine_article/EA014137

29. Кислий Т. Інструкція. Як захистити авторські права в Інтернеті? 2020. URL: https://arzinger.ua/press-center/publications/instruktsiya_yak_zakhistiti_avtorski_prava_v_interneti/

30. Договір Всесвітньої організації інтелектуальної власності про авторське право від 20.12.1996. URL: https://zakon.rada.gov.ua/laws/show/995_770#Text

31. Доктрина інформаційної безпеки України. Указ Президента України від 25.02.2017 № 47/2017 (втратила чинність на підставі Указу Президента № 685/2021 від 28.12.2021). URL: <https://zakon.rada.gov.ua/laws/show/47/2017#Text>

32. Порядок державної реєстрації авторського права і договорів, які стосуються права автора на твір. Постанова Кабінету Міністрів України від 27.12.2001 р. № 1756. URL: <https://zakon.rada.gov.ua/laws/show/1756-2001-%D0%BF#Text>

33. Право інтелектуальної власності. URL: <http://sites.kpi.kharkov.ua/pravo/iv/2.html>

34. Про авторське право і суміжні права. Відомості Верховної Ради України (ВВР), 1994, № 13, ст. 64. URL: <https://zakon.rada.gov.ua/laws/show/3792-12?find=1&text=%D1%86%D0%B8%D1%84>

35. Про застосування судами норм законодавства у справах про захист авторського права і суміжних прав. Постанова Пленуму Верховного Суду. 04.06.2010 № 10. URL: <https://zakon.rada.gov.ua/laws/show/v0005700-10#Text>

36. Про затвердження мінімальних ставок винагороди (роялті) за використання об'єктів авторського права і суміжних прав. Постанова Кабінету Міністрів України від 18.01.2003 р. № 72. URL: <https://zakon.rada.gov.ua/laws/show/72-2003-%D0%BF#Text>

37. Прехорова Г. Історія становлення сфери інтелектуальної власності в Україні. Інтелектуальна власність. 2018. № 7. С. 4-12.

38. Рекомендації для Інтернет - провайдерів, контент-провайдерів та користувачів файло - обмінних мереж та інших веб - сервісів щодо правомірного використання об'єктів авторського права і суміжних прав у мережі Інтернет. URL: <http://sips.gov.ua/ua/ip.html?s=print>

39. Рекомендації Ради Європи щодо поваги, захисту та здійснення прав дитини в цифровому середовищі, 2020. URL: <https://www.coe.int/en/web/children>

40. Рекомендації щодо забезпечення правомірності створення і використання фотографічних творів. URL: <https://www.me.gov.ua/Documents/Print?lang=uk-UA&id=33cec514-d70e-4df5-adbc-c9fba22daea8>

41. Стратегія інформаційної безпеки. Указ Президента України від 28.12.2021 р. № 685/2021. URL: <https://zakon.rada.gov.ua/laws/show/685/2021#Text>

42. Тарасенко Л. Л. Цифрове середовище як місце здійснення прав інтелектуальної власності. Центр дослідження інтелектуального

права. URL: <http://ipcenter.lviv.ua>

43. Цивільний кодекс України [Електронний ресурс]. URL: <https://zakon.rada.gov.ua/laws/show/435-15/ed20081014> (дата звернення: 12.04.2020).

44. Цивільний кодекс України. Відомості Верховної Ради України (ВВР), 2003, №№ 40-44, ст. 356. URL: <https://zakon.rada.gov.ua/laws/show/435-15?find=1&text=%D0%B0%D0%B2%D1%82%D0%BE%D1%80%D1%81%D1%8C%D0%BA#Text>

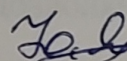
ДОДАТКИ

Додаток А. Технічне завдання

Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

ЗАТВЕРДЖУЮ

Голова секції “Управління інформаційною
безпекою” кафедри МБІС
д.т.н., професор



Юрій ЯРЕМЧУК

“ 22 ” березня 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

до бакалаврської дипломної роботи на тему:

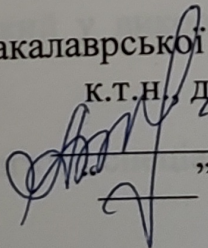
«Розробка програми захисту авторського права цифрових зображень дизайну
інтерєрів»

08-72.БДР.021.00.098.ТЗ

Керівник бакалаврської дипломної роботи

к.т.н., доцент, каф., МБІС:

Адлер О. О.



Вінниця – 2024 р.

1. Найменування та область застосування

Програма захисту авторського права цифрових зображень дизайну.

Область застосування: захист авторського права та захист творчих робіт дизайнерів.

2. Підстава для розробки

Розробка виконується на основі наказу ректора ВНТУ № 80 від 11.03.2024 р.

3. Мета та призначення розробки

3.1 Мета розробки: розробка програми захисту авторського права для цифрових зображень в галузі дизайну інтер'єрів.

3.2 Призначення: розроблена програма дозволяє захищати авторські права зображень дизайну інтер'єрів.

4. Джерела розробки

4.1. Digital Environment: Takeaways. A blog with key takeaways from the digital environment webinar discussing theoretical approaches and debates. URL: <https://core-evidence.eu/articles/digital-environment-takeaways>

4.2 Digital Millennium Copyright Act. 105th Congress (1997-1998). URL: <https://www.congress.gov/bill/105th-congress/house-bill/2281/text>

4.3 Copyright Licensing in the Digital Environment. WIPO. URL: https://www.wipo.int/copyright/en/activities/copyright_licensing.html

4.4 Борисенко В. Захист авторських прав в ІТ. Юрист і Закон. URL: https://uz.ligazakon.ua/ua/magazine_article/EA014137 2023. № 2(38). С. 335–343.

5. Вимоги до програми

5.1 Вимоги до функціональних характеристик:

5.1.1 Програма повинна мати зручний, легкий у використанні інтерфейс користувача;

5.1.2 Розробка програми не повинна вимагати спеціальних ліцензійних програмних додатків;

5.1.3 Програма повинна виконувати процес збереження авторського права;

5.2 Вимоги до надійності:

5.2.1 Програма повинна працювати без помилок, у випадку виникнення критичних ситуацій необхідно передбачити виведення відповідних повідомлень;

5.2.2 Програмний засіб повинен виконувати процес автентифікації користувачів у системі.

5.2.3 Програма повинна виконувати свої функції.

5.3 Вимоги до складу і параметрів технічних засобів:

- процесор – Pentium 1500 МГц і подібні до них;
- оперативна пам'ять – не менше 512 Мб;
- середовище функціонування – операційна система сімейство Windows;
- вимоги до техніки безпеки при роботі з програмою повинні відповідати існуючим вимогам та стандартам з техніки безпеки при користуванні комп'ютерною технікою.

6. Вимоги до програмної документації

6.1 Обов'язкова поетапна інструкція для майбутніх користувачів, наведена у пункті 3.3

7. Вимоги до технічного захисту інформації

7.1 Необхідно забезпечити захист розроблюваного програмного засобу від несанкціонованого використання.

7.2 Можливість отримання доступу користувачів до інформаційних ресурсів таких як ключі шифрування.

8. Техніко-економічні показники

8.1 Цінність результатів використання даного проекту повинна перевищувати витрати на його реалізацію.

8.2 Має бути реалізований таким чином, щоб підходити для використання широкого загалу.

Додаток Б. Лістинг програми

```
```python

from PIL import Image

import hashlib

class DigitalSignature:

 def generate_signature(self, image_path):

 with open(image_path, "rb") as file:

 image_data = file.read()

 hash_value = hashlib.sha256(image_data).hexdigest()

 return hash_value

 def verify_signature(self, image_path, signature):

 generated_signature = self.generate_signature(image_path)

 if signature == generated_signature:

 return True

 else:

 return False

class AccessControl:

 def __init__(self):

 self.users = {} # Dictionary to store user roles and access levels

 def add_user(self, username, role, access_level):

 self.users[username] = {"role": role, "access_level": access_level}
```

```

def check_access(self, username, required_access_level):

 if username in self.users:

 user_access_level = self.users[username]["access_level"]

 if user_access_level >= required_access_level:

 return True

 return False

class ImageProtectionSystem:

 def __init__(self):

 self.digital_signature = DigitalSignature()

 self.access_control = AccessControl()

 def upload_image(self, image_path):

 # Here you can implement image upload logic, including checking access control

 pass

 def view_image(self, image_path, username):

 if self.access_control.check_access(username, 1): # Example: Access level 1 is
required to view images

 image = Image.open(image_path)

 image.show()

 else:

 print("Access denied.")

 def add_user(self, username, role, access_level):

```

```

 self.access_control.add_user(username, role, access_level)

 def generate_signature(self, image_path):

 return self.digital_signature.generate_signature(image_path)

 def verify_signature(self, image_path, signature):

 return self.digital_signature.verify_signature(image_path, signature)

Приклад використання:

Створюємо екземпляр системи захисту зображень
image_protection_system = ImageProtectionSystem()

Додаємо користувачів та встановлюємо їх ролі та рівні доступу
image_protection_system.add_user("user1", "admin", 3)
image_protection_system.add_user("user2", "standard", 1)

Завантажуємо зображення
image_path = "example_image.jpg"

image_protection_system.upload_image(image_path)

Користувач переглядає зображення
image_protection_system.view_image(image_path, "user1")

Генеруємо та перевіряємо цифровий підпис
signature = image_protection_system.generate_signature(image_path)

print("Generated Signature:", signature)

print("Signature Verified:", image_protection_system.verify_signature(image_path,
signature))``

```

## Додаток В. Ілюстративний матеріал

### Розробка програми захисту авторського права цифрових зображень дизайну інтер'єрів

Виконав: студент 4 курсу гр. 2КІТС-206 Шевчук Б.В.

Керівник: доц. Адлер О.О.

Вінниця ВНТУ – 2024

#### Методи захисту



Контроль доступу



Цифрові підписи



Водяні знаки

## Мета роботи

Це розробка програмного забезпечення для захисту авторського права цифрових зображень, які використовуються в дизайні інтер'єрів. Це програмне забезпечення повинно забезпечувати ефективний захист інтелектуальної власності авторів, запобігаючи несанкціонованому використанню та копіюванню їхніх робіт.

### Об'єктом дослідження є

цифрові зображення, що використовуються в галузі дизайну інтер'єрів. Ці зображення є важливою складовою частиною сучасного дизайну, дозволяючи створювати візуалізації, прототипи та кінцеві продукти. В умовах цифровізації об'єктом стає також їх захист від несанкціонованого використання та копіювання, що важливо для збереження авторських прав та інтелектуальної власності.



### Предметом дослідження є

методи захисту авторського права для цифрових зображень у галузі дизайну інтер'єрів. Це включає вивчення та розробку технологій і алгоритмів, які забезпечують захист цифрових зображень від незаконного використання.

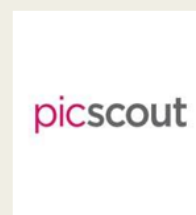
## Аналоги програми



Visual Watermark



Digimarc



PicScout

## Покрокові аспекти розробки



# Метод RSA

Метод RSA (Rivest-Shamir-Adleman) є одним з найпопулярніших алгоритмів асиметричного шифрування, який також використовується для створення цифрових підписів. Цифровий підпис забезпечує автентичність, цілісність та неможливість відмови від авторства підписаного документа або зображення. Нижче наведено детальний опис процесу підпису зображення методом RSA.

# Водяні знаки

Водяні знаки (watermarks) – це методи захисту цифрових медіа, таких як зображення, відео та аудіо, від несанкціонованого використання, копіювання та розповсюдження. Водяні знаки можуть бути видимими або невидимими і забезпечують додатковий рівень безпеки та ідентифікації власності на контент. Ось докладний опис процесу додавання водяних знаків до зображень.

## Використані технології та сервіси

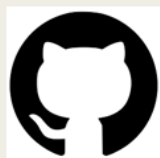
### Python

Мова програмування, що використовується для створення додатку



### GitHub

Тут можна знайти багато репозиторіїв з бібліотеками

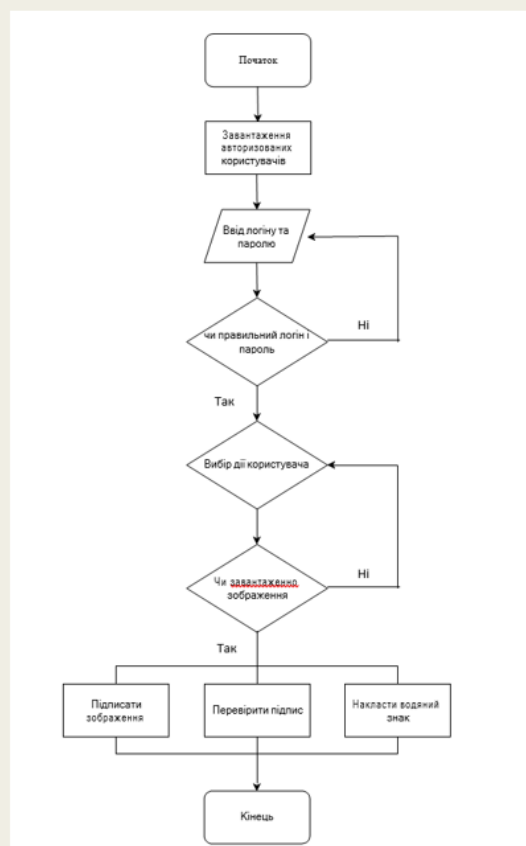


### PyCharm

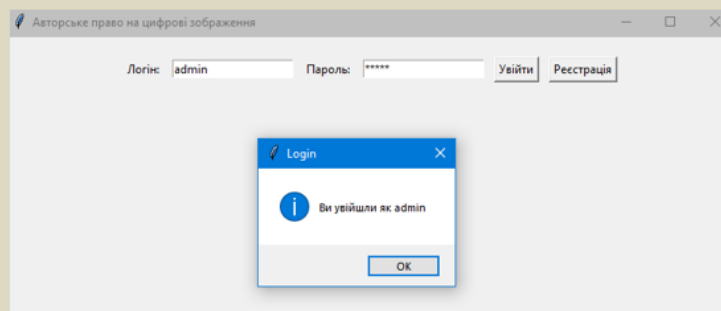
інтегроване середовище розробки (IDE), спеціально розроблене для програмування на Python



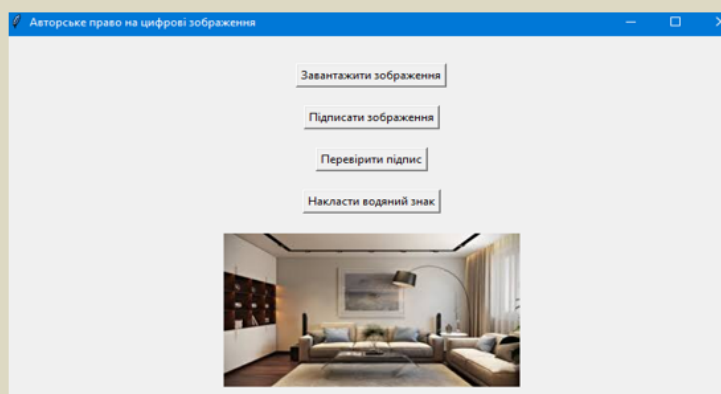
## Блок схема роботи програми



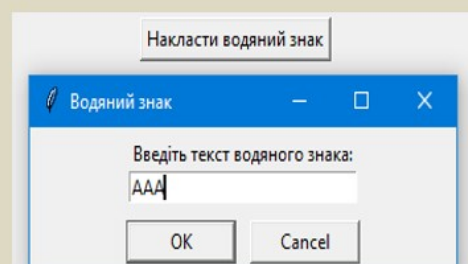
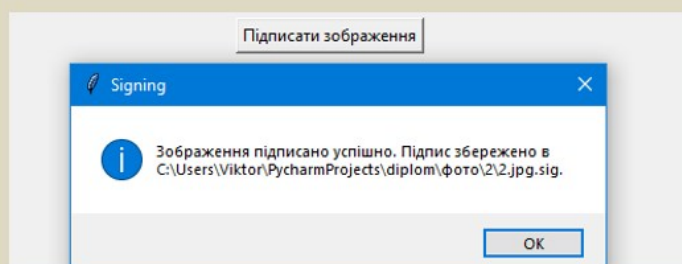
## Процедура аунтефікації



## Основний інтерфейс



## Демонстрація функціоналу



## Порівняння зображень



З правої сторони  
продемонстровано  
зображення із водяним  
знаком



# Висновки

У результаті виконання роботи було здійснено:

- Дослідження аналогів
- Розробка алгоритму роботи системи
- Реалізація програмного застосунку
- Тестування реалізованого додатку

Дослідження виявило, що більшість сучасних методів захисту авторських прав цифрових зображень у галузі дизайну інтер'єрів мають обмежену ефективність і не завжди забезпечують належний рівень безпеки. Розроблена програма має значний потенціал для впровадження у різних галузях, де необхідний захист цифрових зображень. Захист інтелектуальної власності у галузі дизайну інтер'єрів має ключове значення для підтримки творчості та інновацій.

Дякую за увагу !

ПРОТОКОЛ  
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА  
НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Розробка програми захисту авторського права цифрових зображень дизайну інтер'єрів

Тип роботи: бакалаврська дипломна робота  
(БДР, МКР)

Підрозділ: Кафедра менеджменту та безпеки інформаційних систем  
Факультет менеджменту та інформаційної безпеки  
(кафедра, факультет)

**Показники звіту подібності Unicheck**

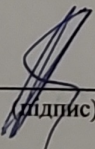
Оригінальність 90 %

Схожість 10 %

Аналіз звіту подібності (відмітити потрібне):

1. Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
2. Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
3. Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку

  
(підпис)

Коваль Н.П.  
(прізвище, ініціали)

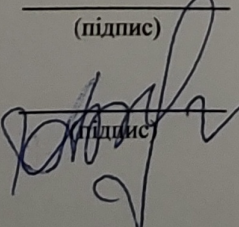
Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи

  
(підпис)

Шевчук Б.В.  
(прізвище, ініціали)

Керівник роботи

  
(підпис)

Адлер О.О.  
(прізвище, ініціали)