Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

ЗАСІБ ДЛЯ ВБУДУВАННЯ ПОВІДОМЛЕННЯ У ФОТОГРАФІЇ ІЗ ЗБЕРЕЖЕННЯМ СТРУКТУРИ КОНТЕЙНЕРА

Виконав студент 2 курсу, групи 2КІ–22м
спеціальності 123 — Комп'ютерна інженерія
Гонца А.В.
Керівник д.т.н., проф. каф. ОТ
Азаров О.Д.
"07" 12 2023р.
Опонент к.т.н., доц. зав. каф. МБІС
Карпінець В.В.
"08" 12 2023р.

Допущено до захисту
Завідувач кафедри
обчислювальної техніки
д.т.н., проф. Азаров О.Д.

" 11 " 12 2023 р.

ВНТУ 2023

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки
Освітньо-кваліфікаційний рівень магістр
Спеціальність 123 Комп'ютерна інженерія

Затверджую
Завідувач кафедри
обчислювальної техніки
проф., д.т.н. О. Д. Азаров
_____
«_26__»_вересня_2023 р.

З А В Д А Н Н Я

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Гонці Андрію Владиславовичу

1 Тема роботи **«Засіб для вбудування повідомлення у фотографії із збереженням структури контейнера»**, керівник роботи Азаров Олексій дмитрович, д.т.н.,професор,затверджені наказом вищого навчального закладу від 18.09.223_року № 247.

2 Строк подання студентом роботи 9.12.2023 р.

3 Вихідні дані до роботи: симетричний алгоритм шифрування AES, мова об'єктно-орієнтованого програмування C# середовище програмування Visual Studio.

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): вступ, загальний аналіз стеганографічних методів приховування інформації та стеганоконтейнерів, розробка алгоритму роботи програмного додатку на основі обраного методу, практична реалізація засобу для вбудовування даних, тестування та дослідження засобу для вбудовування даних, висновки.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): структурна схема стеганосистеми, основні компоненти розроблювального програмного засобу, алгоритм роботи розроблюваного додатку для приховування даних, алгоритм роботи розроблюваного додатку для вилучення даних, UML — діаграма класів розробки.

6 Консультантів розділів роботи представлено в табл. 1.

Таблиця 1 — Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	Завдання прийняв
1,2, 3,4	Азаров О.Д., д.т.н., проф. каф. ОТ		
5	Небава М.І., к.е.н., проф. каф. ЕП і ВМ		

7 Дата видачі завдання 09.09.2023 р.

8 Календарний план наведено в табл. 2.

Таблиця 2 — Календарний план

№ з/п	Назва етапів виконання магістерської роботи	Строк виконання етапів роботи	
1	Постановка мети та задач роботи	20.10.23	Вик
2	Аналіз стеганографічних методів, стеганоконтейнери	27.10-30.10.23	Вик
3	Розробка алгоритму роботи програмного додатку	01.11-09.11.23	Вик
4	Вибір інструментарію розробки	10.11-17.11.23	Вик
5	Налаштування середовища розробки	18.11-.22.1123	Вик
6	Розробка програмного засобу	23.11-26.11.23	Вик
7	Тестування якості роботи засобу	27.11-31.11.23	Вик
8	Дослідження засобу для вбудовування даних	01.12-04.12.23	Вик
9	Розрахунок економічної частини роботи	01.12-04.12.23	Вик
10	Оформлення МКР	05.12.23-09.12.23	Вик
11	Аналіз виконання роботи, висновки, додатки	05.12.23-09.12.23	Вик
12	Перевірка якості виконання МКР	05.12.23-09.12.23	Вик

Студент _____ Гонца А.В.

Керівник роботи _____ Азаров О.Д.

АНОТАЦІЯ

УДК 004.9

Гонца А.В. Засіб для вбудування повідомлення у фотографії із збереженням структури контейнера. Магістерська кваліфікаційна робота зі спеціальності 123 — комп'ютерна інженерія, освітня програма — комп'ютерна інженерія. Вінниця: ВНТУ, 2023, 113 с.
На укр. мові. Бібліогр.: 29 назв, рис 39, табл. 8.

В магістерськіц кваліфікаційній роботі розглянуто методи для вбудування повідомлення у фотографії із збереженням структури контейнера.

Проблема приховування інформації в нешкідливих контейнерах з метою секретної передачі полягає, наприклад, коли є потреба захистити мережевих записувачів, іноді змушує користувачів мережі використовувати методи комп'ютерної стеганографії (CS), щоб приховати правду листування. Більшість стеганографічних алгоритмів дозволяють приховати тільки невеликі обсяги інформації, а на практиці часто виникає потреба у негласній передачі великих обсягів даних.

У роботі розроблено засіб для приховування великих обсягів даних у відомих графічних форматах для подальшої їх передачі.

Ключові слова: захист, стеганографія, контейнер, алгоритм, програмне забезпечення, графічні формати.

ANNOTATION

Gontsa A.V.

A tool to embed a message in a photo while preserving the container structure. Master's qualification route in the specialty 123 — computer engineering, educational program computer engineering. Vinnitsa, VTNU, 2023, 113 p.

In the Ukr. leng. Libr. name 29, figure 39, table 8.

This master's thesis is devoted to the development of a tool for embedding a message in a photograph while preserving the container structure.

The problem with hiding information in innocuous containers for the purpose of secret transmission is, for example, when there is a need to protect network recorders, sometimes forcing network users to use Computer Steganography (CS) techniques to hide the truth. correspondence.

Most steganographic algorithms allow you to hide only small amounts of information, and in practice there is often a need for covert transmission of large amounts of data.

Currently, we are working on the development of a tool for hiding large volumes of data in known graphic formats for further transmission.

Keywords: protection, steganography, container, algorithm, software, graphic formats.

ЗМІСТ

ВСТУП

Актуальність теми полягає в тому, що в зв’язку зі стрімким розвитком інформаційних технологій виникають проблеми, пов’язані із захистом даних та приховуванням факту передачі [1, 2]. Проблема приховування інформації в нешкідливих контейнерах з метою секретної передачі полягає, наприклад, коли є потреба захистити мережевих записувачів, іноді змушує користувачів мережі використовувати методи комп’ютерної стеганографії (CS), щоб приховати правду. листування.

Для захисту даних, а точніше, для забезпечення, власне, факту захисту безпеки в цифровій стеганографії використовуються контейнери — цифрові об'єкти, в яких розміщується інформація, що часто призводить до структурних змін даних в контейнерах.

Використання цифрової графіки у форматі стегоконтейнера залежить від таких причин:

— високий рівень поширення цифрової графіки;

— популярність і легкість обміну та публікації цифрових зображень в Інтернеті;

— зручний розмір контейнера з точки зору операцій з файлами (аудіофайли та відеофайли зазвичай у середньому більші за цифрові зображення);

— особливості системи зору людини, які не дозволяють візуально виявляти невеликі зміни в контейнері.

Однак більшість стеганографічних алгоритмів дозволяють приховати тільки невеликі обсяги інформації. Але на практиці часто виникає потреба у негласній передачі великих обсягів даних. Тому, дослідження у напрямку розробки програмного забезпечення важливо приховувати великі обсяги даних у відомих графічних форматах для подальшої їх передачі, є актуальними.

Детальне вивчення конкретних алгоритмів наведено в роботах Джордана (F. Jordan), Квісквотера (JJ Quisquater), Е. Коxf (E. Koch), Куттера (M. Kutter), Дж. Жао (J. Zhao), Делейгла (J. - F. Delaigle), Боси (F. Bossen), Дармстедтера (V. Darmstaedter), Хсу (Chiou-Ting Hsu) і Ву (Ja-Ling Wu) та інших відомих вченних. Отже, виходячи з вищесказаного тема роботи є важливою та актуальною.

Об'єктом дослідження є процес використання алгоритмів Дармстедтера (V. Darmsteadter), Delaigle (J.-F. Delaigle), Quisquater (JJ Quisquater) та Mack (B. Macq) для прихованого передавання даних у зображенні.

Предметом дослідження є методи та алгоритми вбудовування даних у стегоконтейнери для секретної передачі, а також вивчення змін у цих контейнерах.

Метою цієї магістерської роботи є підвищення інформаційної безпеки шляхом вбудовування повідомлень у зображення зі збереженням структури контейнера.

Для досягнення поставленої мети були визначені та вирішені такі **задачі:**

— вивчити концепцію змісту цифрової стеганографічної системи, провести загальний аналіз властивостей системи зору людини; проаналізувати значення растрових і векторних зображень як контейнерів;

— порівняти існуючі методи та обґрунтувати вибір способу вбудовування даних у зображення;

— розробка алгоритму програмної реалізації методу та аналізу параметрів дослідження контейнеризації;

— обрати інструментарій для розробки програмного продукту;

— програмна реалізація запропонованого способу та розробка інтерфейсу користувача програмного засобу.

Практична цінність роботи полягає у створенні програмного продукту, який дозволяє вбудовувати повідомлення у фотографію зі збереженням структури контейнера.

Методи дослідження — методи об’єктно-орієнтованого програмування, які є фундаментальними для розробки програмного забезпечення.

Апробацію результатів наукової роботи було проведено на науковій конференції — «LIII Науково–технічна конференція факультету інформаційних технологій та комп'ютерної інженерії (2024)», доповідь на тему “Засіб для вбудування повідомлення у фотографії із збереженням структури контейнера”

1 ЗАГАЛЬНИЙ АНАЛІЗ СТЕГАНОГРАФІЧНИХ МЕТОДІВ ПРИХОВУВАННЯ ІНФОРМАЦІЇ ТА СТЕГОКОНТЕЙНЕРІВ

1.1 Аналіз концепції стеганографії та практичне застосування

Глобальне поширення та постійне вдосконалення комп'ютерних і телекомунікаційних систем супроводжується збільшенням пропускної здатності, інтеграцією сервісів, використанням нових технологій обробки та зберігання даних, ускладненням забезпечення інформаційної безпеки. Вирішення проблем забезпечення конфіденційності, доступності та цілісності разом досягається використанням криптографічних і стеганографічних методів захисту.

При цьому криптографія має на меті зберегти в таємниці семантику надісланих повідомлень, а стеганографія — зберегти в таємниці фвкт надсилання такого повідомлення.

Сьогодні в стеганографії виділяють кілька напрямків:

— класична стеганографія, що включає «некомп'ютерні методи» приховування повідомлень неелектрнного характеру;

— комп'ютерна стеганографія передбачає використання властивостей форматів даних, що обробляються та передаються в інфокомунікаційних мережах;

— цифрова стеганографія базується на перевазі використання мультимедійних даних, представлених у цифровій формі, які спочатку є аналоговими (зображення, відео, аудіо) [1,3].

Проаналізувавши сучасну наукову літературу, можна виділити чотири напрямки стеганографії — це класична, цифрова, лінгвістична та квантова стеганографія (рис. 1.1). Стандартна стеганографія — це метод приховування інформації за допомогою техніки ШІ [1].

На сьогоднішній день, на жаль, відсутній нормативний документ, який би регулював класифікацію методів, принципів і термінології в галузі стеганографії, тому в різних джерелах по різному називається один і той же

термін або автори вводять власний тезаурус. Далі представлені основні терміни цифрової стеганографії, складені шляхом узагальнення та уточнення понять, які зазвичай використовуються в публікаціях у цій галузі [4].

Шифрування даних (стеганографія) — це процес, який реалізує методи передачі даних, які дозволяють передавати додаткову інформацію в структурі даних, представлених у цифровій формі та використовуваних як контейнер.

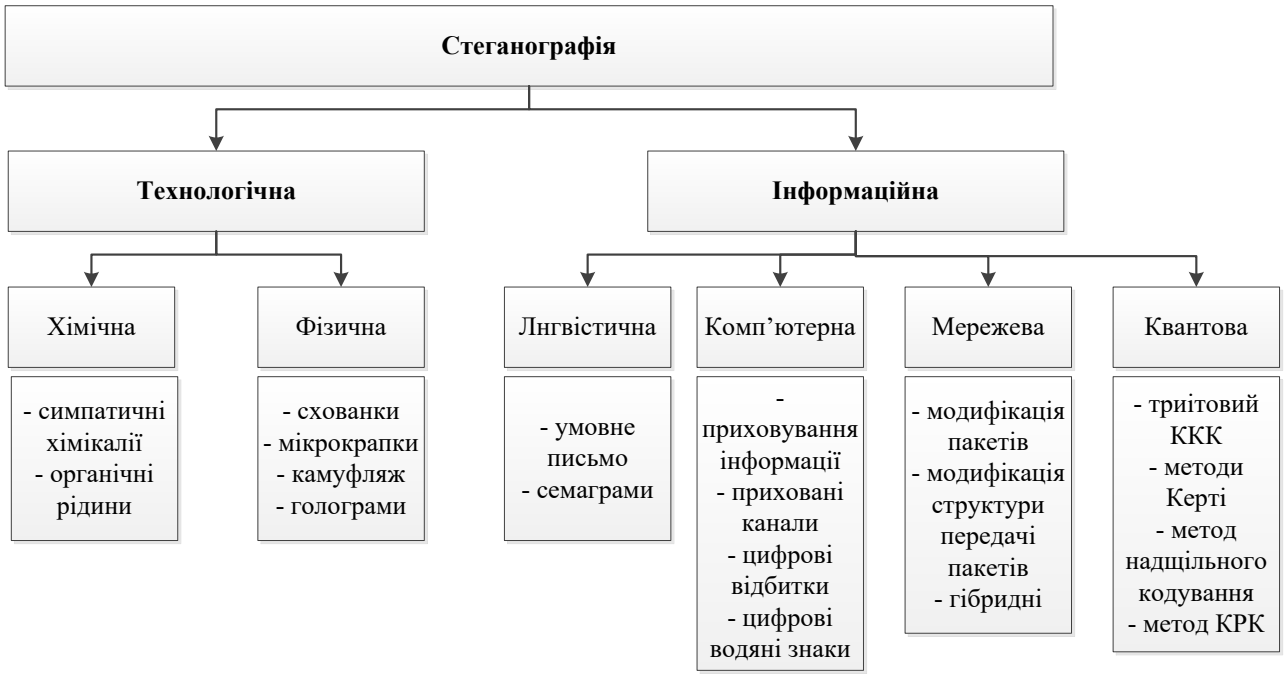


Рисунок 1.1 — Класифікація стеганографічних засобів

Під контейнером (в якому міститься об'єкт) розуміються цифрові дані, які дозволяють надавати додаткову інформацію, не розкриваючи факт використання привілею. Контейнер, що містить додаткову інформацію, називається порожнім, інакше — заповненим (або стего) [5].

Сукупність методів і засобів виявлення та додавання та отримання додаткової інформації споживачем без порушення цілісності контейнера дозволяє говорити про формування секретного (стеганографічного) каналу передачі інформації.

Стеганографічна система (стегосистема) — це сукупність засобів і

методів відправлення та отримання порожнього контейнера (рисунок 1.2, рисунок 1.3), пов'язаних із засобами і методами, що використовуються для створення секретного каналу передачі інформації.

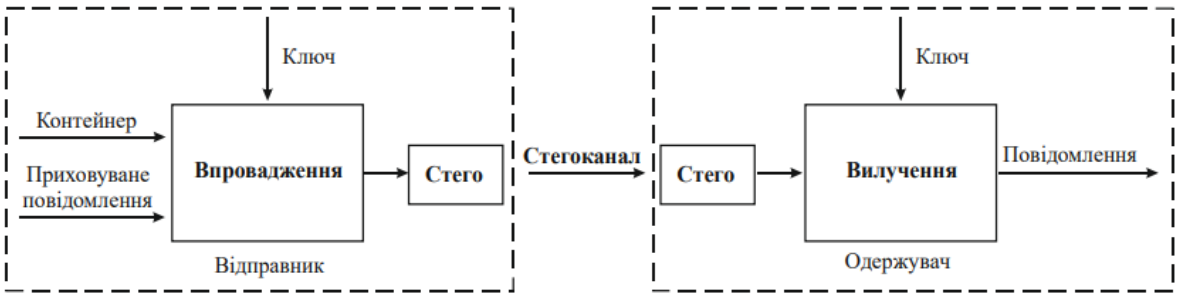


Рисунок 1.2 – Узагальнена модель стегосистеми

Прихованість [6,7] (стеганографічна стійкість) визначається можливими діями (атаками) зловмисника на систему стеганографії. Залежно від цілей організації стеговкладення, під прихованістю розуміють стійкість спроби видалити або знищити стеговкладення, коли воноа перестає бути секретом для порушника. При цьому напади, застосовувані злочинцем у першому випадку, будуть пасивними і можуть виражатися в наступних заходах:

— візуальний огляд з метою суб'єктивної оцінки якості;

— об'єктивний контроль даних за одним чи кількома параметрами оцінки якості виділеного зображення;

— гістограмна атака.

У іншому випадку зловмисник буде виконувати геометричні атаки на стего: повертати, масштабувати, стискати тощо.

Загалом методи цифрової та комп’ютерної стеганографії можна класифікувати відповідно до типу контейнера, вибраного для вбудовування, та структури контейнера для цілей застосування.

Рисунок 1.3 — Структурна схема стеганосистеми

Загальновизнаними є три напрямки використання цифрової та комп’ютерної стеганографії:

— вбудовування прихованих каналів мовлення, метою вбудовування є приховування реальності мовлення;

— впровадження цифрових водяних знаків (DWA), метою впровадження є підтвердження автентичності поданих даних та запобігання несанкціонованому доступу до них;

— розміщення ідентифікаторів (цифрових відбитків пальців) з метою секретного опису та перевірки переданих даних.

За типом контейнера, обраного для реалізації стеганографії, стеганографічні методи поділяються на варіативні методи, дані та програми, текстові, аудіо та відео типи. Організація прихованих додатків в основному зумовлена надмірністю типу даних, обраного користувачем, через очевидну популярність, для цієї мети використовують аудіо- та відеоданих. Стегометоди організації секретними каналами в основному використовуються як контейнери для аудіо- та відеоданих.

Форматні та неформатні стегометоди відрізняються тим, яке поле структури контейнера потрібно змінити. Використання першого обмежено низькою пропускною здатністю високої стеганографічної стабільності та часто використовується для вбудовування ЦВЗ. Другий напрямок виявився більш перспективним і заснований на зміні параметрів прихованого простору файлу, який безпосередньо описує зображення або звукові дані. Стеганографічні алгоритми, які протистоять розкриттю, добре розроблені в цій галузі та забезпечують достатню місткість контейнера для розміщення зашифрованих повідомлень або програм. Зокрема, це офіційно визнані стеганографічні алгоритми F5 та OutGess. Оскільки прихована смуга пропускання безпосередньо пов’язана з перевагою контейнера, зображення, які діють на користь прихованої передачі, нерухомі і рухомі зображення є оптимальними.

Аналіз відкритих публікацій у галузі стеганографії дозволяє говорити про появу нових галузей і напрямків. Як правило, удосконалення стегометодів відбувається за рахунок підвищення прихованості або збільшення пропускної здатності [8].

Таким чином, у цьому розділі представлено систематизацію та класифікацію сучасних стеганографічних галузей. На цьому етапі виділяють чотири напрямки стеганографії: класичну, цифрову, лінгвістичну та квантову. Кожна сфера представлена певними методами приховування конфіденційної інформації. Систематизація стеганографічних методів ЗІ значно спрощує пошук і дозволяє повноцінно оцінити наявний рівень успіху для більш ефективного використання. Виконана робота залишає широке поле для більш ґрунтовних досліджень і дозволяє підвищити ефективність створення нових стеганографічних систем СС (стійких проти різноманітних атак) [9].

1.2 Контейнери в стеганографічних системах

1.2.1 Типи контейнерів

Контейнер — це відкритий об'єкт, який використовується для приховування конфіденційної інформації. Будь-який матеріальний об’єкт, носій даних, джерело даних, текст, зображення, файл тощо може виконувати роль контейнера. Залежно від типу даних у контейнері розробляються більш стабільні стеганографічні методи або вибираються більш відповідні чи доступні методи. Ефективність методів шифрування даних залежить від призначення, структури та типу контейнерів. Стеганографічні методи базуються на характеристиках форматів та структур контейнерів. Це пояснюється тим, що вміст контейнера повинен мати перевагу, яка дозволяє виконувати великі додаткові повідомлення або допускати певні модифікації вмісту контейнера, до яких важко отримати доступ.

Звичайно, легше та безпечніше приховати конфіденційні дані в інших відкритих даних у більших обсягах. Тому сучасні стеганографічні методи, як правило, базуються на принципах приховування секретної інформації в інших великих обсягах контенту та змісту.

Зауважимо, що багато експертів вважають за доцільніше називати «носієм» об’єкт, який містить приховану інформацію, але в багатьох наукових джерелах такі об’єкти називаються «контейнером». Через свою популярність термін «контейнер» був прийнятий як загальний [10].

Залежно від принципів шифрування контейнери поділяються на дві категорії: потокові та фіксовані [11]. Потокові контейнери складаються з послідовності бітів. Оскільки приховані дані вводяться, коли контейнер надходить у режимі реального часу, розмір контейнера, необхідного для передачі повної інформації, невідомий кодеру. Ви можете опублікувати кілька повідомлень в одному великому контейнери. Інтервали між доданими (або зміненими) бітами визначаються генератором випадково розподілених псевдовипадкових послідовностей.

У той же час основними проблемами є досягнення синхронізації між відправником і одержувачем, а також визначення початку і кінця послідовності. Дані контейнера включають біти синхронізації, заголовки пакетів тощо. за наявності конфіденційної інформації можна відстежити їх. Велике значення для забезпечення конфіденційності передачі має складність організації синхронізації. Прикладом поточного контейнера є стандартна вузол стегоприставки, приєднаний до звичайного телефону. Під прикриттям звичайної телефонної розмови, можна приховати іншу розмову, конфіденційну інформацію тощо. Тоді, не знаючи секретного ключа, неможливо визначити вміст зашифрованої передачі або навіть правдивість передачі.

Розміри та властивості фіксованих контейнерів відомі заздалегідь, і ця інформація дозволяє відповідне (оптимальне) приховування. Надалі, під поняттям контейнер, буде матися на увазі фіксований контейнер. Контейнери можуть бути вибраними, випадковими чи навязаними. Вибраний контейнер залежить від характеру або призначення даних, що зберігаються. Випадкові контйнери використовуються в повсякденній діяльності, включаючи комп’ютерні технології, Інтернет, тощо. широко і вільно використовується, часто такі контейнери використовуються на практиці. Навязаний контейнер — це контейнер, який нав'язується відправнику, коли зловмисник (порушник) підозрює можливість обміну конфіденційною інформацією між сторонами.

Щоб більш надійно приховати секретні дані в контейнері, його розмір повинен бути більшим за розмір даних, які потрібно приховати. І навпаки, якщо розмір секретного повідомлення більший за розмір контейнера, контейнер є обов’язковим. Для цього необхідно стиснути контейнер, щоб значно зменшити його розмір перед вставленням зашифрованих даних. Крім того, рекомендується попередньо шифрувати за допомогою спеціального криптографічного алгоритму, щоб більш надійно захистити вміст конфіденційної інформації у разі виявлення в контейнері [3,4].

Графіка, аудіо, відео тощо втрачаються під час процесу перенесення. обсяг, розмір, формат та інші характеристики. контейнери можуть змінюватися. Для усунення цього недоліку використовуються методи забезпечення цілісності транспортного контейнера за допомогою кодів для виправлення помилок (кодування проти помилок). Слід зазначити, що початкова обробка часто виконується за допомогою закритого ключа для надійного захисту конфіденційної інформації.

Процес вставки інформації в контейнери виконується стеганокодером шляхом зміни форми або незначної зміни вмісту, і це визначається стеганографічним методом. Після введення даних модифікований контейнер надсилається одержувачу по відкритому каналу зв’язку [12]. Зрозуміло, що стеганограми вразливі до пасивних або активних атак зловмисника, коли вони відбуваються через відкриті канали зв’язку. Під час пасивної атаки зловмисник, який виявляє наявність конфіденційної інформації, перехоплює всі контейнери, надіслані по каналах зв’язку, потім аналізує їх окремо, а потім у пакеті. Правильно визначивши наявність конфіденційних даних, зловмисник намагається отримати їх із контейнера. У той же час, якщо секретні дані попередньо зашифровані, потрібно зламати додатковий код. Якщо пасивний порушник неспроможен визначити факт існування прихованої інформації, то природно, не зможе її вилучити.

Таким чином, при пасивній атаці контейнер не змінюється. При активній атаці зловмисник модифікує або знищує контейнер. Активний зловмисник може змінити контейнер, надісланий по каналу зв'язку, без відома відправника або одержувача. При цьому він видаляє незаконну чи приховану інформацію або вносить до неї значні зміни, створюючи фальшиву стеганограму. У той же час контейнери, що відправляються в канал зв’язку, спочатку повинні бути проаналізовані, щоб змінити або видалити дані [13].

1.2.2 Аналіз растрових і векторних зображень як стегоконтейнерів

Вибір контейнера має значний вплив на надійність стегосистеми та здатність виявляти передачу зашифрованих повідомлень.

Можливі такі види контейнерів:

— контейнер створюється самою стегосистемою. Це може бути програма MandelSteg, у якій фрактал Мандельброта створюється як контейнер для розміщення повідомлення такий підхід можна назвати конструктивною стеганографією;

— контейнер вибирається з набору контейнерів. У цьому випадку створюється багато альтернативних контейнерів, щоб вибрати найбільш підходящий для приховування повідомлення;

— контейнер надходить ззовні, у цьому випадку немає можливості вибрати контейнер, і перший найкращий контейнер для приховування повідомлення не завжди відповідає вбудованому повідомленню.

У сучасній поліграфічній промисловості всі зображення та елементи дизайну представлені різними типами цифрових зображень. Цифрові зображення поділяються на растрові, векторні та складені формати на основі оригінального методу визначення. Класифікація форматів цифрових зображень показана на рисунку 1.4.

Більш швидкісні зображення містять двовимірні масиви (піксельні матриці), кожен елемент яких представляє свій вихідний компонент із середнім показником кольору [14]. Глибина кольору – властивість, яка визначає якість передачі кольору, кількість відтінків, які можуть відображати елементи піксельної матриці. Кожен елемент масиву даних (матриці) є числом у двійковій системі числення.

Його розмір виражається в бітах. Глибина кольору — це кількість біт на піксель зображення. Один байт (8 біт) може зберігати 256 кольорів (переважно чорний і білий).

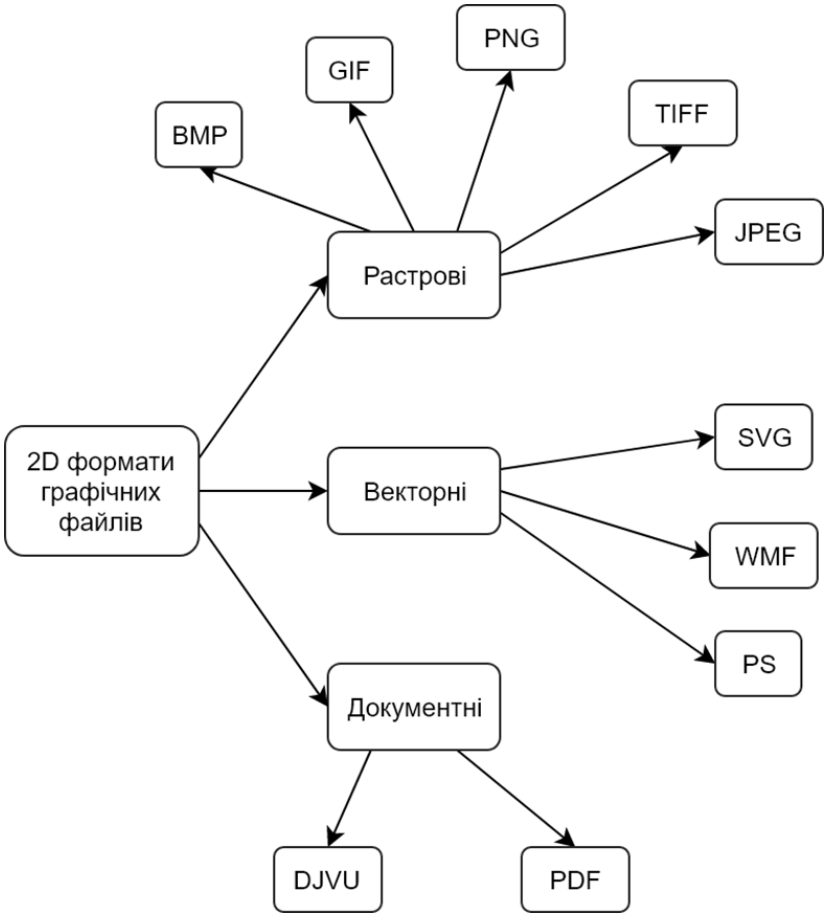


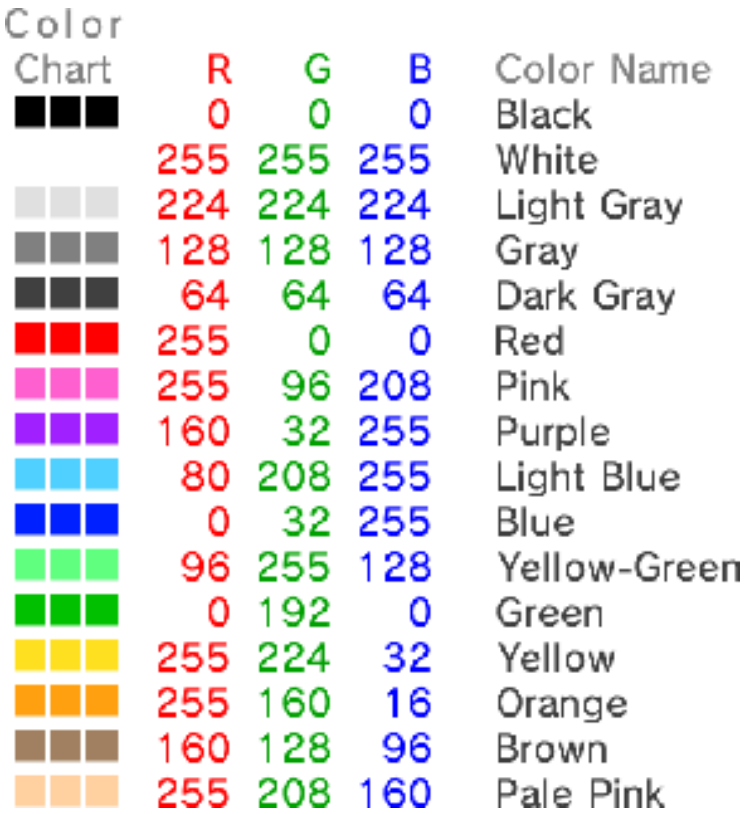
Рисунок 1.4 — Класифікація форматів графічних файлів

Колір пікселя визначається як комбінація трьох кольорів (червоний, зелений, синій) різних кольорів (рисунок 1.5), схема RGB [14].

Деякі дані завжди втрачаються під час аналого-цифрового перетворення, оскільки вибірка завжди виконується шляхом усереднення та підсумовування оригінального потоку аналогових даних. Тому основним недоліком растрових цифрових зображень є те, що їх неможливо збільшити без втрати якості.

Основна сфера застосування растрових зображень — це фотографічні ілюстрації. Коли вам потрібно відтворити аналоговий оригінал, будь то фотографія, лінія чи складний елемент дизайну, який легко конвертується у вектор, використовуються растрові зображення.

Векторні зображення — це зовсім інший тип цифрових зображень. Найменшими елементами векторного зображення є вектор і крива Безьє



Color Chart	R	G	B	Color Name
	0	0	0	Black
	255	255	255	White
	224	224	224	Light Gray
	128	128	128	Gray
	64	64	64	Dark Gray
	255	0	0	Red
	255	96	208	Pink
	160	32	255	Purple
	80	208	255	Light Blue
	0	32	255	Blue
	96	255	128	Yellow-Green
	0	192	0	Green
	255	224	32	Yellow
	255	160	16	Orange
	160	128	96	Brown
	255	208	160	Pale Pink

Рисунок 1.5 — Приклади представлення кольорів пікселя через три кольорових компоненти (схема RGB)

Векторні зображення отримують двома способами — оригінальним ручним трасуванням і автоматичним трасуванням.

Основна перевага векторного зображення — можливість масштабування без втрати якості. Ще однією перевагою векторних зображень є невеликий розмір файлів, які їх містять. Це робить зручним передачу векторних зображень по електронних каналах зв'язку [14].

Поняття шарів лежить в основі вертикальної структури векторно-растрових зображень. Шар — це поле даних, яке містить інформацію про окремий елемент вертикальної структури зображення.

Векторні зображення — це зовсім інший тип цифрових зображень. Найменшими елементами векторного зображення є вектор і крива Безьє (рис. 1.6).

Векторні зображення фіксуються двома способами - оригінальним і автоматичним відстеженням.

Головна перевага векторного зображення в тому, що його можна збільшувати без втрати якості. Ще однією перевагою векторних зображень є невеликий розмір файлів, які їх містять. Це робить зручним передачу векторних зображень по електронних каналах зв'язку [14].

Концепція шарів лежить в основі вертикальної структури векторно-растрових зображень. Шар — це поле даних, яке містить інформацію про окремий елемент вертикальної структури зображення.

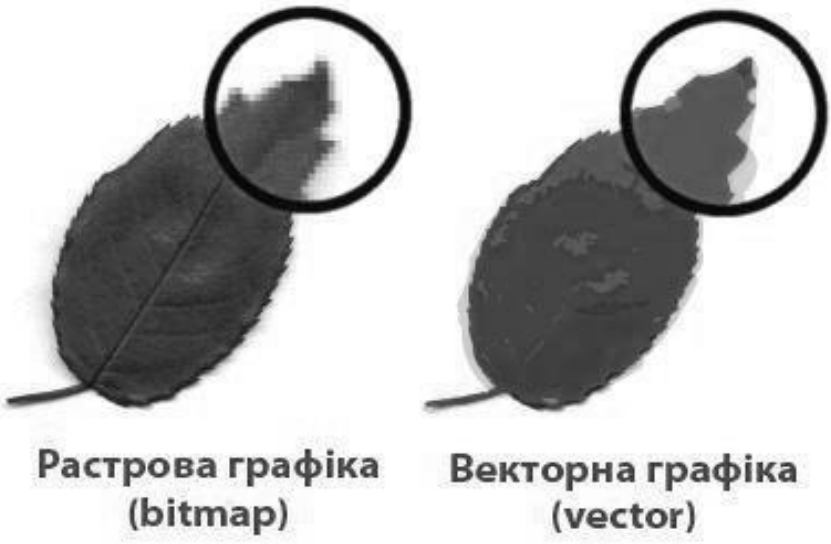


Рисунок 1.6 — Приклад растрового та векторного зображень

Отже, незважаючи на певні недоліки растрових зображень, вони є найпоширенішими — растрова графіка використовується практично всюди від іконок до плакатів. Цей вид графіки дозволяє створювати практично будь-яке зображення незалежно від складності, на відміну від векторної, де неможливо точно досягти ефекту плавного переходу від одного кольору до іншого без втрати кольору файлу [14].

Таким чином, об’єкт безпеки в інформаційній системі – це інформація обмеженого доступу, яка зберігається у вигляді інформації, команд, повідомлень, для інформації, яка є цінною та має обмежене поширення та для

інформації, як для її власника, так і для можливого потенційного порушника технічного захисту інформації.

1.3 Аналіз властивостей зорової системи людини та їх практичне значення в стеганографії

Характеристики зорової системи людини поділяються на дві категорії низького рівня (або фізіологічні) і високого рівня (або психофізіологічні) [15]. Серед властивостей низького рівня є три важливі, які впливають на появу шуму на зображенні: чутливість до змін яскравості (контрастності) зображення; частотна чутливість; ефект маски. [16] Рис. 1.7 показана залежність мінімального контрасту ΔI/I від яскравості.

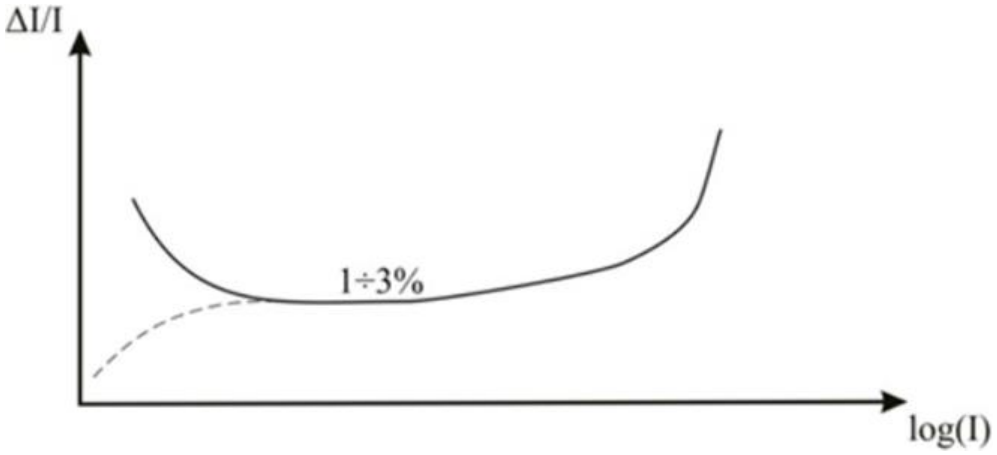


Рисунок 1.7 — Чутливість до зміни контрасту і поріг непомітності ΔI

Як бачимо, середній діапазон яскравості змінюється, контраст приблизно постійний, а величина невидимого порогу (ΔI) зростає як для малих, так і для великих значень яскравості. Визначено, що ΔI≈ (0,01 ÷ 0,03) · I для середніх значень освітленості [15].

Частотна чутливість ЗСЛ виникає, коли люди більш чутливі до низькочастотного (НЧ), ніж до високочастотного (ВЧ) звуку. Це пов'язано з нерівномірністю амплітудно-частотної характеристики ЗСЛ. [30] Елементи ZSL поділяють вхідний сигнал на окремі частини, кожна з яких збуджує закінчення зорового нерва через кілька невеликих каналів. Візуально відмінні

компоненти мають різні просторові та частотні характеристики, а також різні просторові напрямки (горизонтальні, вертикальні та діагональні) [16].

Частотна чутливість тісно пов'язана з яскравістю. Відомий також вираз для визначення порогів маски на основі певної світлочутливості, що дозволяє знайти міру спотворення зображення, яка враховуватиме властивості ЗСЛ. У цьому випадку були розроблені детальні математичні моделі для оцінки коефіцієнтів ДКП, оскільки це використовується в стандарті JPEG.

Високорівневі властивості ЗСЛ відрізняються від низькорівневих, оскільки вони є «вторинними» — після обробки основної інформації з ЗСЛ наш мозок дає команди зоровій системі «налаштувати» зображення (рис. 1.8).

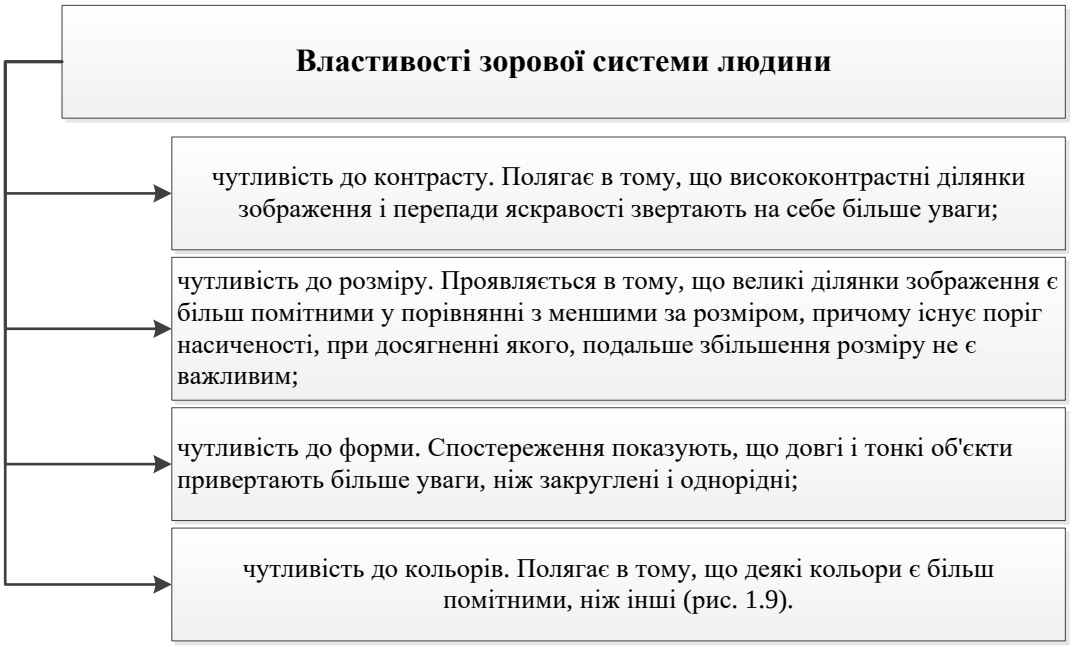


Рисунок 1.8 — Властивості зорової системи людини

Дотримуючись порад роботи ЗСЛ, можна запобігти виявленню конфіденційної інформації методом візуальної атаки. Це пояснюється тим, що він заснований на здатності ЗСЛ аналізувати візуальні зображення та виявляти невідповідності в зображеннях [16].

1.4 Аналіз існуючих програмних аналогів для введення зображень

Далі ми розглянемо та проаналізуємо програмне забезпечення, доступне для вбудовування даних у растрові зображення (рис. 1.9).

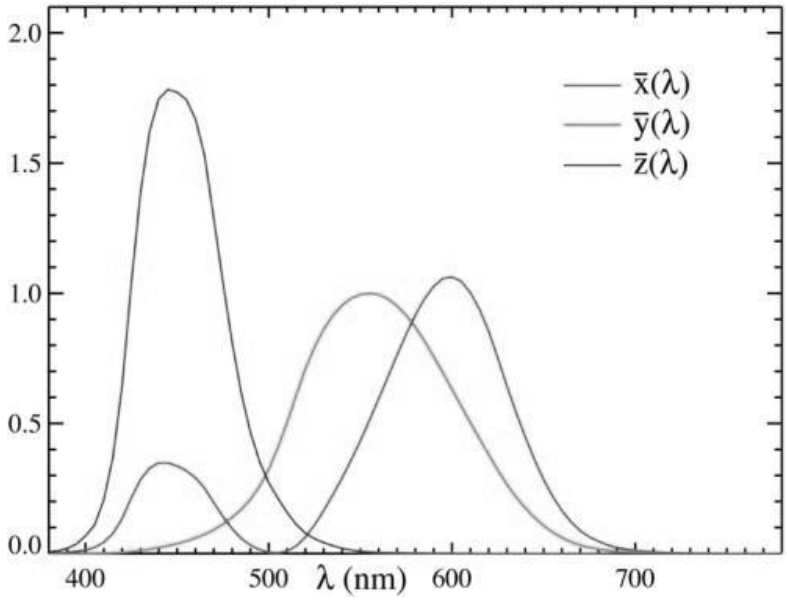


Рисунок 1.9 — Функції колірної відповідності стандартного колориметричного спостерігача

Програма Steganos Privacy Suite 11 (рис. 1.10, рис. 1.11) розроблена компанією Steganos Software. Ціна $69,95. Дозволяє приховати дані в растрових зображеннях bmp і jpg. Для шифрування даних використовується метод НЗБ [17].

Рисунок 1.10 — Програма Steganos Privacy Suite

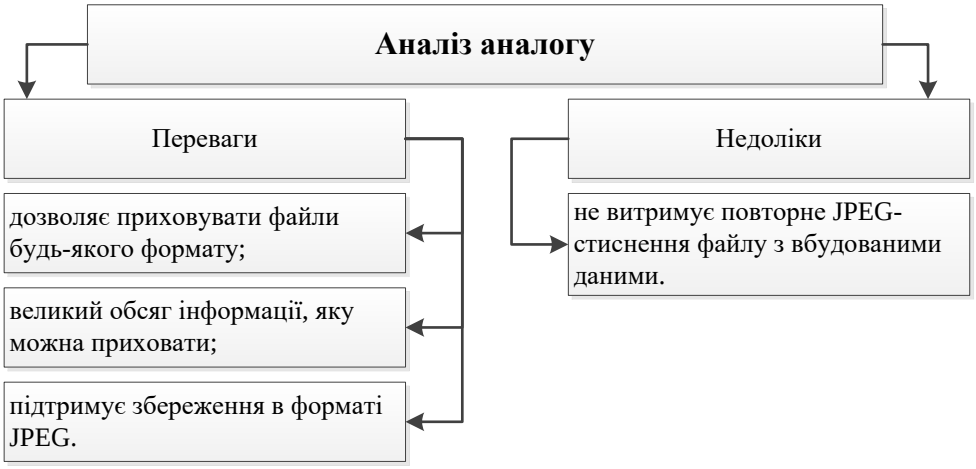


Рисунок 1.11 — Переваги та недоліки Steganos Privacy Suite

Програма S-Tools (рис. 1.12, рис.1.13). розроблена Енді Брауном. Дозволяє приховати дані в зображеннях формату bmp і gif. Для шифрування даних використовується метод NZB. Вільно поширюється [18].

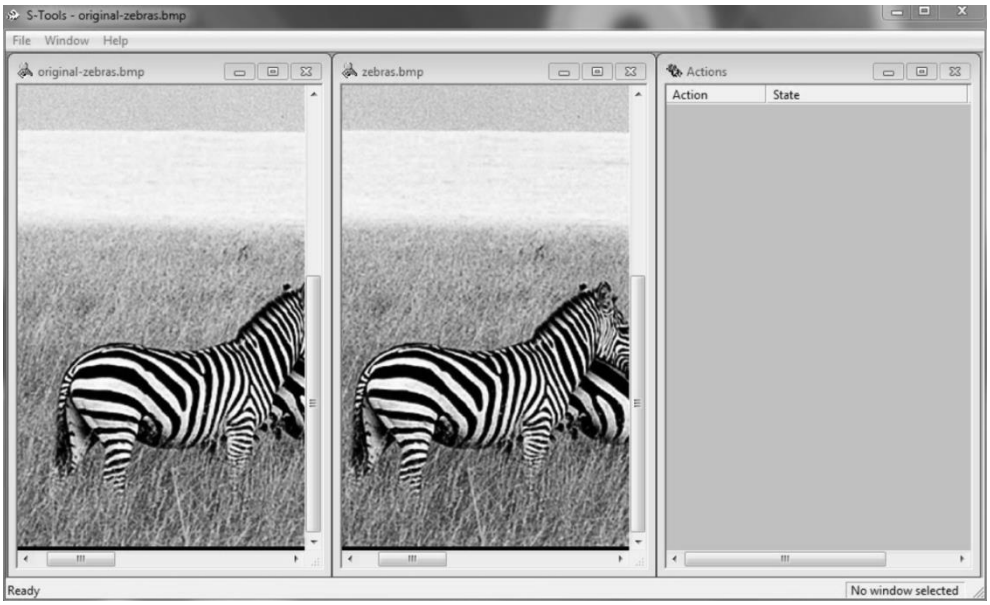


Рисунок 1.12 — Програма S-Tools

Програмне забезпечення ImageSpyer 2009 (рис. 1.14, рис.1.15). дозволяє приховувати дані в зображеннях формату bmp і tiff. Властивість методу NZB використовується для приховування даних. Вільно поширюється [19].

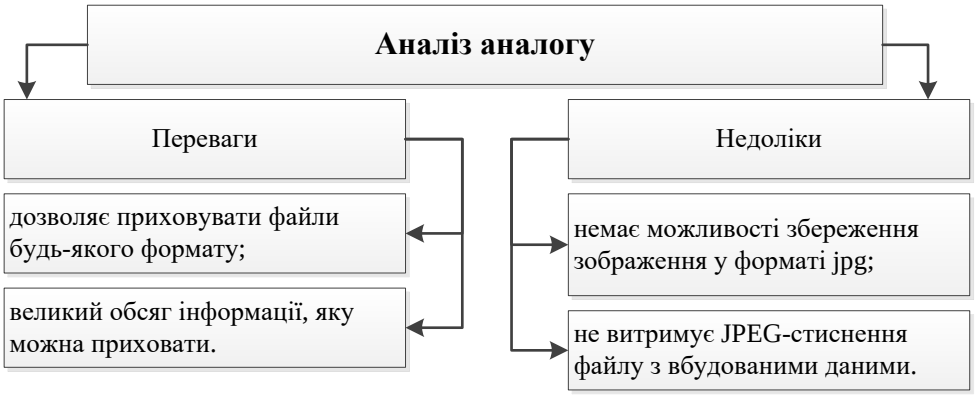


Рисунок 1.13 — Переваги та недоліки S-Tools

Рисунок 1.14 — Програма ImageSpyer

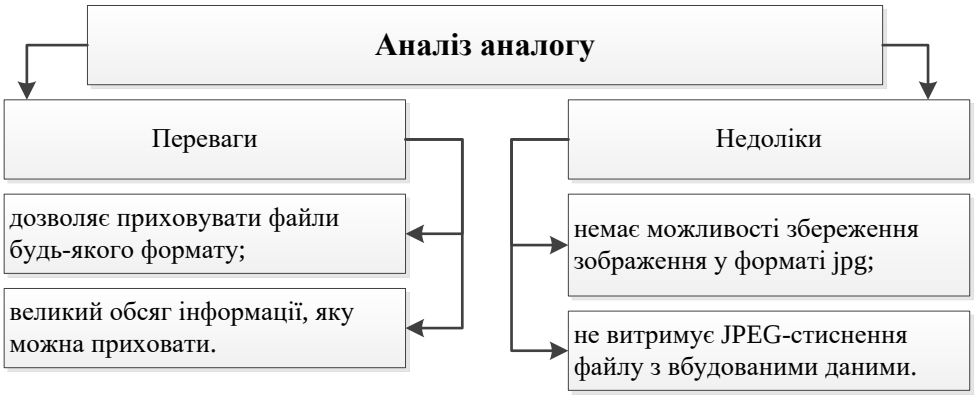


Рисунок 1.15 — Переваги та недоліки ImageSpyer

Програма JSTEG (рис.1.16, рис.1.17) дозволяє приховувати дані у файлах формату jpg. Він приховує дані в молодших бітах, крім нульових коефіцієнтів блоків зображення. Вільно поширюється [20].

Програма Gifshuffle (рис.1.18, рис.1.19) дозволяє приховувати інформацію у файлах формату gif. Інформація приховується шляхом зміни порядку кольорів у палітрі. Вільно поширюється [21].

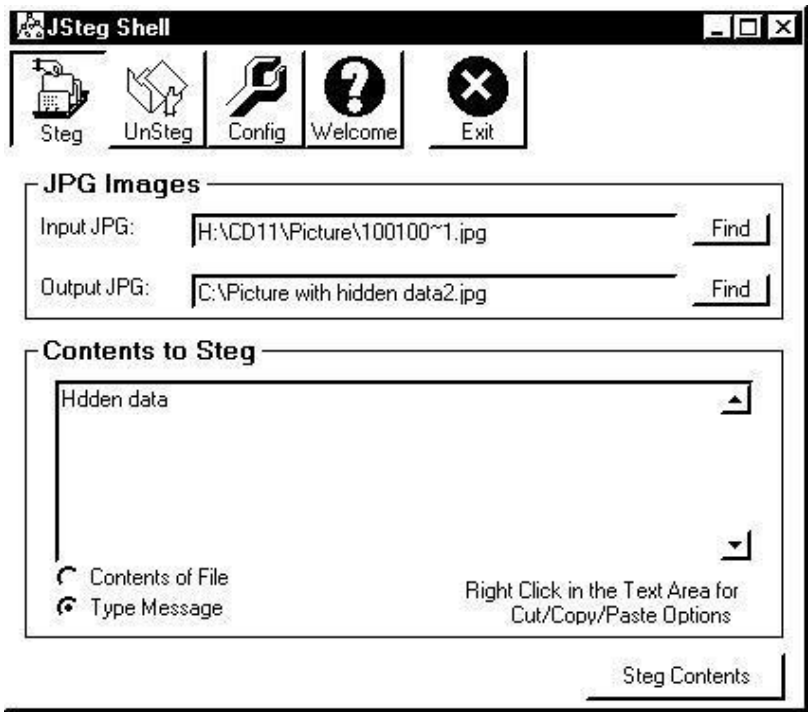


Рисунок 1.16 — Програма JSTEG

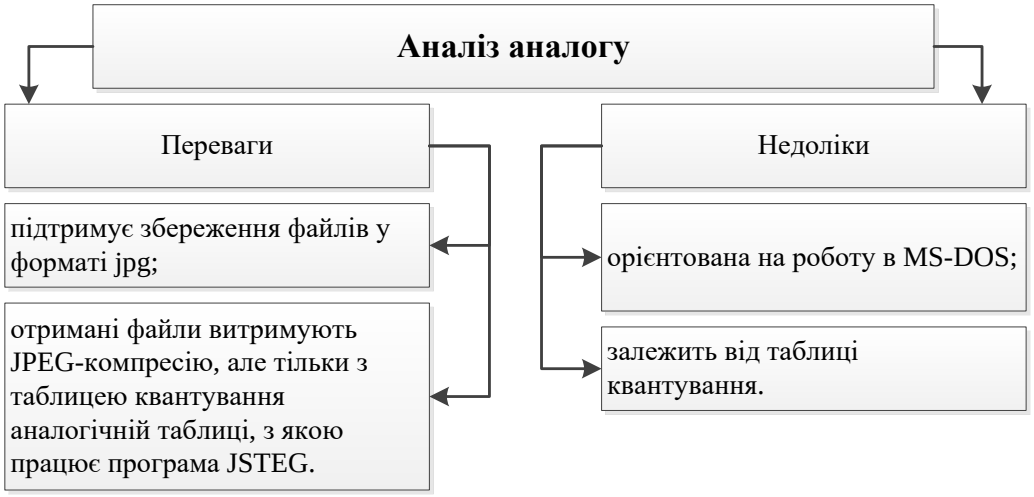


Рисунок 1.17 — Переваги та недоліки JSTEG

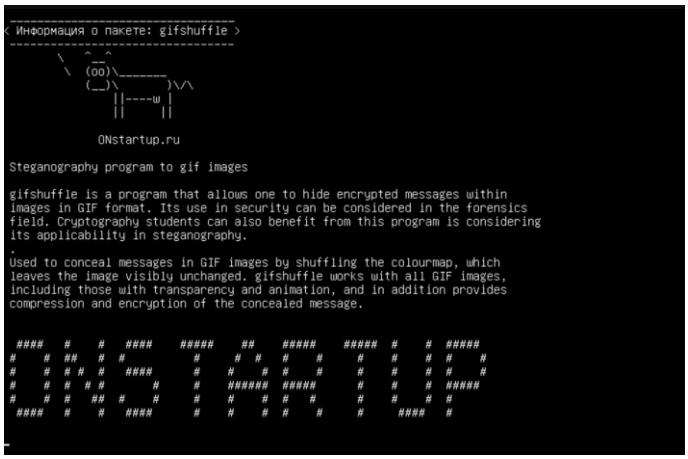


Рисунок 1.18 — Програма Gifshuffle

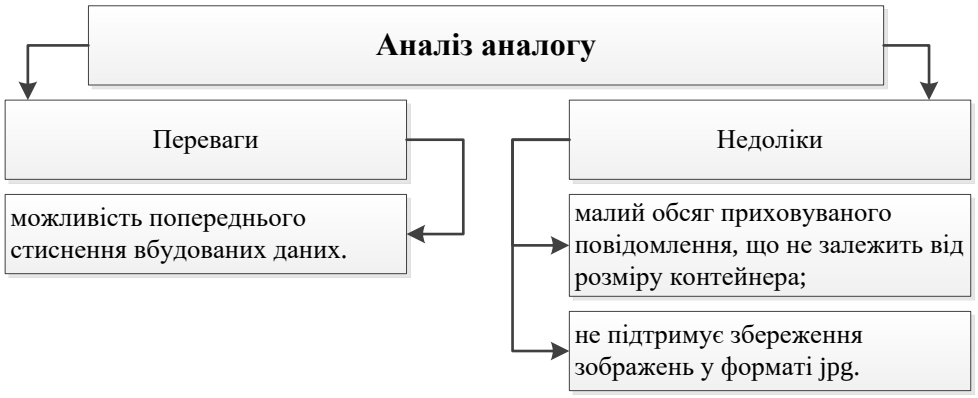


Рисунок 1.19 — Переваги та недоліки Gifshuffle

Узагальнимо наведені порівняння розглянутих програмних засобів у вигляді таблиці 1.1.

Таблиця 1.1 — Порівняння програмних засобів для вбудовування даних у фотографію

Назва	Формати, що підтримуються	Ціна	Метод приховування
Steganos Privacy Suite 11	*.bmp	69,95$	НЗБ
S-Tools 4	*.bmp, *.gif	-	НЗБ
ImageSpyer 2009	*.bmp, *.tiff	-	НЗБ (власна реалізація)
JSTEG	*.jpg	-	НЗБ у відмінних від нуля квантованих коефіцієнтах блоків зображення
Gifshuffle	*.gif	-	Метод заміни палітри

Аналізуючи ринок супутніх програмних продуктів, легко помітити, що більшість із них використовують різні варіації методу НЗБ для приховування повідомлень. Має низьку стеганографічну стійкість до атак пасивних і активних зловмисників (злочинців). Але основним його недоліком є висока чутливість до найменших деформацій контейнера.

Програми, призначені для стеганографічного приховування графічних даних, часто дозволяють приховати лише невелику або обмежену кількість даних.

Таким чином, актуальною науковою проблемою теоретичного та практичного значення є створення нового програмного забезпечення, яке задовольнятиме всім необхідним вимогам, зокрема забезпечуватиме надійний захист конфіденційної інформації зі збереженням структури контейнера.

Отже, слід уточнити та врахувати наступні особливості програмного засобу, що розробляється. Алгоритм вбудовування даних у растрове зображення повинен вносити мінімальні спотворення в зображення та враховувати великі обсяги даних.

Компоненти, необхідні для приховування секретного повідомлення в зображенні (стегоконтейнер) з палітрою не більше 128 кольорів; секретне повідомлення в текстовому вигляді; stegokey (збережено у файл); алгоритм шифрування.

Секретне повідомлення має бути у текстовому форматі. Рекомендується використовувати симетричний алгоритм шифрування для шифрування зашифрованого повідомлення, рекомендовано AES. Цей алгоритм дуже надійний і не має відомих недоліків.

Кількість бітів у зображенні має бути достатньою, щоб приховати зашифроване повідомлення. Якщо розмір даних, які потрібно приховати, занадто великий, розмір зображення також збільшиться. Щоб приховати або відобразити лише одне повідомлення, слід використовувати один стегоключ. Для різних повідомлень можна використовувати різні стегоключі.

Стабільність обраного методу та розробленого програмного забезпечення слід перевірити тестуванням основної вимоги — непомітності передачі даних.

2 РОЗРОБКА АЛГОРИТМУ РОБОТИ ПРОГРАМНОГО ДОДАТКУ НА ОСНОВІ ОБРАНОГО МЕТОДУ

2.1 Варіантний аналіз методів для вбудовування даних

У сучасній стеганографії існує багато способів зберігання інформації в різних типах контейнерів. У цій роботі увага зосереджена на вбудовуванні інформації саме у зображення. Методи молодших бітів, широкосмугові та статичні методи найбільш широко використовуються для цього типу контейнерів. Розглянемо їх детальніше [22].

Методи заміни в просторовій області найчастіше представляється методом заміни молодших бітів (LSB method), який заснований на тому, що молодші біти графічних, аудіо- та відеоформатів несуть мало інформації і їх зміна практично не впливає на якість передачі. зображення чи звук. Це дозволяє використовувати їх для кодування конфіденційної інформації.

Головною перевагою цього методу є простота реалізації та можливість анонімної передачі великих обсягів даних. Однак, вводячи додаткову інформацію, статистичні властивості файлу-контейнера спотворюються, і приховане повідомлення легко виявляється статистичними атаками, такими як оцінка ентропії та коефіцієнти кореляції. Необхідно відкоригувати статистичні характеристики, щоб зменшити достовірні характеристики. Недоліком методу є його чутливість до операцій цифрової обробки: стиснення, застосування фільтрів, перетворення кольорів, геометричні зміни, додатковий шум і зміна формату контейнера [3].

У методах, що працюють у частотній області, інформація прихована в коефіцієнтах частотного представлення контейнера. Для цього часто використовуються модифікації, які використовуються в сучасних алгоритмах стиснення з втратами (ексклюзивне косинусне перетворення в стандарті JPEG).

Дані можуть бути приховані у вихідному зображенні, а також одночасно зі стисненням зображення-контейнера. Важливо, що стегосистеми,

які враховують специфікації алгоритму стиснення, не чутливі до стиснення контейнера. Він також має більшу стійкість до геометричних змін і виявлення каналів (порівняно з методом LSB), а якість стисненого зображення може значно відрізнятися, що унеможливлює визначення походження його спотворення [24].

Суть широкосмугових методів полягає в розширенні частотної смуги сигналу до ширини спектру, необхідної для передачі реальних даних. Існує два способи розширення діапазону: метод прямого розширення спектру, метод псевдовипадкової послідовності та метод частоти зрізу. При цьому корисна інформація розподіляється по всьому діапазону, тому при втраті сигналу в одних діапазонах частот інформації в інших діапазонах її достатньо для відновлення. Принцип роботи широкосмугових методів схожий на проблеми, які вирішують стегосистеми, вони намагаються заховати секретне повідомлення в контейнер і ускладнити його пошук.

Оскільки сигнал поширюється по всьому спектру, його важко розрізнити. Важливою перевагою цих методів є їх стійкість до випадкових і навмисних спотворень. Тому він використовується в комунікаційних технологіях для забезпечення високої захищеності та складності процесу зберігання та виявлення. Однак його недоліком є наявність стеганалізу за рахунок цифрової обробки за допомогою шумопоглинаючих фільтрів.

Статистичні методи приховують дані, змінюючи деякі статистичні властивості зображення. Наприклад, ідея алгоритму Patchwork базується на припущенні, що значення пікселів є незалежними та рівномірно розподіленими. Одночасно генерується секретний ключ для активації генератора псевдовипадкових чисел, який відображає зображення, де введено водяний знак.

Для цього у відповідності із стегоключем вибирається n пар пікселів (bi,ci) в яких значення яскравості змінюється у такий спосіб:

$$\bar{b} = bi + 1, \bar{c} = ci + 1,$$

$$Sn = \sum_{i=1} n(\bar{b}i - \bar{c}i)$$

Якщо Sn значно відрізняється від нуля, вважається, що є вбудовані дані. Цей метод дуже стійкий до цифрової обробки. У широкосмугових і статистичних методах з використанням псевдовипадкового кодування наявність секретного ключа підвищує їх надійність.

Тому, враховуючи особливості, переваги та недоліки кожного типу методу, доцільніше використовувати метод вбудовування зображення в частотну область на основі дискретно-косинусного перетворення.

2.2 Вибір методу візуалізації даних

Дармстедтер (V. Darmsteadter), Делейгл (J.-F. Delaigle), Квісквотер (JJ Quisquater) і Макк (B. Macq) запропонували використовувати блоковий метод позиціонування контейнерного простору. Розроблений ними метод дозволяє досягти компромісу між стійкістю стеганосистеми до спотворень, якістю реалізації і, звичайно, обчислювальною складністю алгоритму [26]. Цей метод заснований на початковому тактильному (візуальному) відчутті та дозволяє регулювати розміщення блоків-контейнерів відповідно до їх поточного вмісту.

Перед вбудовуванням сенсорні дані перетворюються на вектор двійкових даних. Кожен біт розміщується в окремому блоці. У цьому випадку розмір блоку становить 8x8 пікселів. Головною особливістю цього простору є його відношення до блоків, які використовуються для стиснення JPEG. Таким чином, ефект стиснення буде однаково застосований до кожного імплантованого біта [6]. Крім того, інформація розміщується у верхньому положенні, підвищуючи загальну стабільність стеганосистеми.

Загалом, процес вбудовування бітів повідомлення виконується в чотири етапи (рис. 2.1).

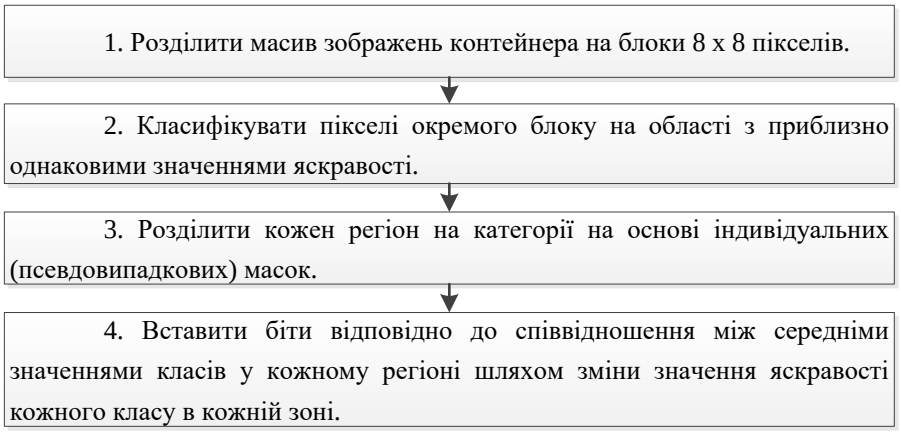


Рисунок 2.1 — Процес вбудовування бітів повідомлення

Розглянемо останні три кроки більш детально. Класифікація на зони [6].

Мета полягає в тому, щоб розділити пікселі в блоці на групи приблизно однакової яскравості. Ця класифікація враховує властивості блоків, що цікавлять з точки зору видимості та стабільності. При класифікації виділяють три заперечення (рис. 2.2).

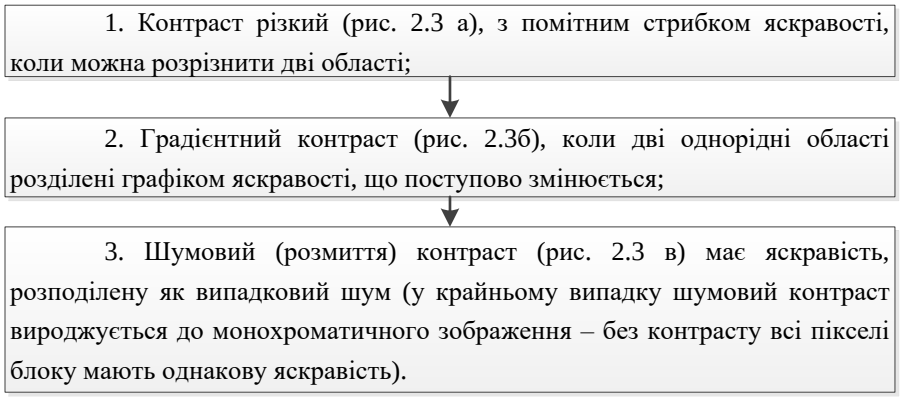


Рисунок 2.2 — Класифікація пік селів всередині блоку

Відсортовуючи за зростанням значення яскравості пікселів блоку можливо створити зростаючою функцією F (i) , де F (1) — це найменше значення яскравості серед усіх, які використовуються у цьому блоці, а F (N 2) — найбільше серед присутніх у блоці значень яскравості, де (N — розмірність квадратного блоку). Тип контрасту блоку визначається крутизною функції F

(i) , яку позначимо через S (i) .

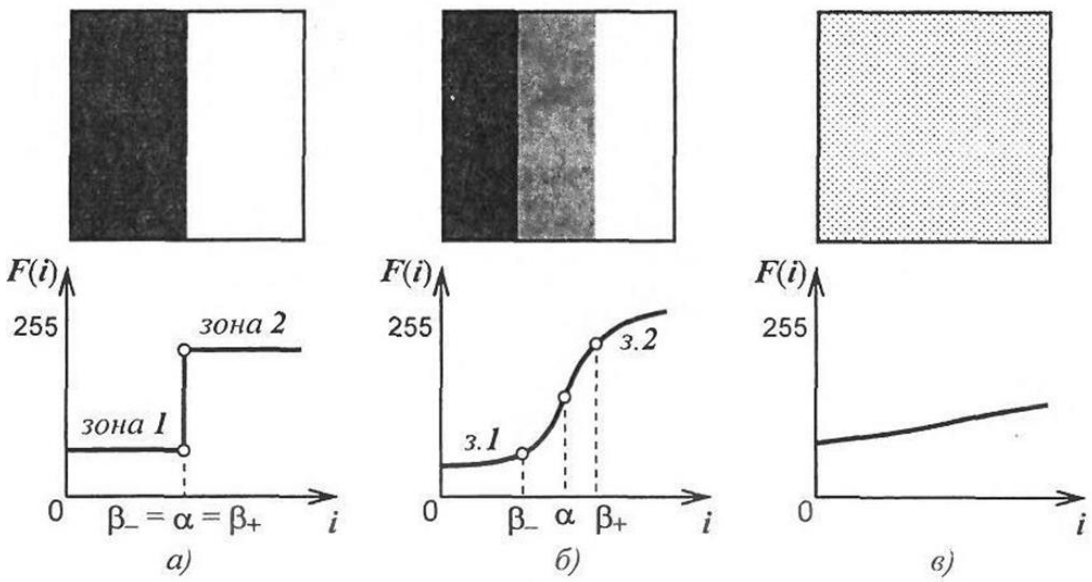


Рисунок 2.3 — Класифікація на зони: а) різко виражений контраст; б) поступовий контраст; в) шумовий контраст

Вважаємо, що S_{max} — максимальна крутизна функції F при i = a . Тоді, S_{max} нижче заданого порогу T_1 і вважатемо, що блок має шумовий контраст. В ситуації, якщо S_{max} перевищує поріг T_1, то блок має чи поступовий, чи різко виражений контраст. При цьому, в такому випадку додатково визначають такі параметри, як β_+ та β_- — індекси у найближчому околі точки, і, відповідно, вище і нижче її, які задовольняють нерівностям [26].

$$S(a) - S(\beta_+) > T_2 \quad \text{і} \quad S(a) - S(\beta_-) > T_2,$$

де T_2 — задане значення порогу.

При цьому, якщо контраст різко виражений, то, відповідно, $\beta + \approx a$ і $\beta \approx a$.

Далі, якщо контраст поступовий, тоді інтервал [ ] β + β — , є перехідною зоною поступового контрасту.

За наступними правилами відбувається класифікація пікселів p(x,y) на дві

зони.

Перша зона .для різкого і поступового вираженого контрастів:

— якщо $p(x, y) \leq \mathrm{F}\ \beta$, тоді піксель $p(x, y)$ належить до зони 1;

— якщо $p(x, y) \leq \mathrm{F}\ \beta$, тоді піксель $p(x, y)$ належить до зони 2;

— якщо $p(x, y) \leq \mathrm{F}\ \beta$, тоді піксель $p(x, y)$ належить до перехідної зони.

У другій зоні відбувається розподіл шумового контрасту пікселей на дві зони однакової розмірності:

— якщо $p(x, y) < \mathrm{F}\ (\mathrm{N}^2/2)$, тоді піксель $p(x, y)$ стосується зони 1;

— якщо $(x, y) < \mathrm{F}\ (\mathrm{N}^2/2)$, тоді піксель $p(x, y)$ стосується зони 2.

У блоках першого та другого типів зони з різною яскравістю не завжди повинні розміщуватися поряд один із одним та не обов'язково повинні містити однакову кількість пікселів. Крім того, деякі пікселі можуть взагалі не належати до вищезгаданих з цих зон. У блоках третього типу класифікація дещо складніша.

Після розбиття на попередні зони необхідно передбачити вбудовування додаткового біта шляхом модифікації деяких характеристик зон. Інколи буває, що безпосередній вплив на зони призводить до певних результатів, які або недостатньо стійкі до певних перетворень, або ж є незадовільняють очікуваних результатів, виходячи з показників деяких візуальних спотворень вихідного зображення.

При поошукові оптимального для вбудовування пікселя, насамперед, відбувається розподіл зони на дві категорії (A та Z). Далі для сортування пікселів по цих категоріях на блоки зображення накладаються маски, причому, бажана індивідуальність масок для кожного з конкретних блоків. Призначення масок полягає у забезпеченні таємності вбудовування. Нижче наведено приклади масок для двох зон, які представлені на рис. 2.4.

A	A	A	A	Z	Z	Z	Z
A	A	A	A	Z	Z	Z	Z
A	A	A	A	Z	Z	Z	Z
A	A	A	A	Z	Z	Z	Z
Z	Z	Z	Z	A	A	A	A
Z	Z	Z	Z	A	A	A	A
Z	Z	Z	Z	A	A	A	A
Z	Z	Z	Z	A	A	A	A

а)

A	A	Z	Z	A	A	Z	Z
A	A	Z	Z	A	A	Z	Z
Z	Z	A	A	Z	Z	A	A
Z	Z	A	A	Z	Z	A	A
A	A	Z	Z	A	A	Z	Z
A	A	Z	Z	A	A	Z	Z
Z	Z	A	A	Z	Z	A	A
Z	Z	A	A	Z	Z	A	A

б)

Рисунок 2.4 — Приклади масок, які використовуються: а) розмірами 4× 4 , б) розмірами 2× 2

Далі рекомендується використовувати більш складні комбінації та змінювати маску при переході до приховування слідуючого наступного біта повідомлення. Категорія, до якої далі буде віднесено той або інший піксель; залежить від наступних чинників:

— простір розміщення пікселя у масиві блоку;

— номери зони, до якої віднесений піксель.

Слід зазначити, що алгоритм формування масок має триматися у таємнмці, оскільки знання конфігурації масок суттєво знижує стійкість стеганосистеми уцілому.

Наступний важливий етап — це правила вбудовування біт повідомлення. За результатами виконання попередніх трьох етапів отримані наступні чотири різні групи пікселів у певних блоках

Залежно від зони (1 або 2) і категорій (A або Z), при чому слід зазначити, що існує ще і п'ята група пікселів, а саме, ті, які не ввійшли в жодну із зон, при чому, останні не беруть участі в подальшому аналізі.

Для того, щоб результат вбудовування зробити якомога непомітнішими, потрібно бути зберігати низькі частоти, до них найчутливіша ЗСЛ.

Інтенсивності кожної зони зберігаються із врахуванням середніх значень, які забезпечуються виконанням наступних умов.

$$\frac{n_{1A} \cdot \lambda^*_{1A} + n_{1Z} \cdot \lambda^*_{1Z}}{n_{1A} + n_{1B}} = \Lambda_1;$$

$$\frac{n_{2A} \cdot \lambda^*_{2A} + n_{2Z} \cdot \lambda^*_{2Z}}{n_{2A} + n_{2B}} =$$

Вилучення вбудованої інформації із контейнера вимагає наявності інформації про розмірності блоків, на розбиття зображення, і ще, також про конфігурацію масок, що використовувались при вбудовуванні.

Цей процес вилучення складається з таких етапів:

— розбиття зображення на блоки розмірністю N × N ;

— класифікація пікселів одного блоку на зони;

— розподіл кожної зони на категорії;

— порівняння середніх значень яскравості при визначенні значення вбудованого біта даних.

А для підвищення завадостійкості потрібно використовувати один з циклічних кодів корекції помилок, як варіант, — Боуза-Чоудхурі-Хоквенгема.

2.3 Особливості побудови засобу

В загальному випадку, стеганографічну систему розроблюваного програмного засобу можна представити як комплекс таких складових [7]:

— обробник запитів — складова, яка забезпечує передачу інформації між додатком і користувачем для подальшої обробки;

— стегокодер — складова, яка відповідає за вбудовування даних в

зображення;

— стегодетектор — складова, якаа використовується при виявленні вбудованих даних у зображення;

— обробник файлів — складова, яка відповідає за обробку файлів.

Схематично, наведені вище компоненти переставлені на рисунку 2.5.

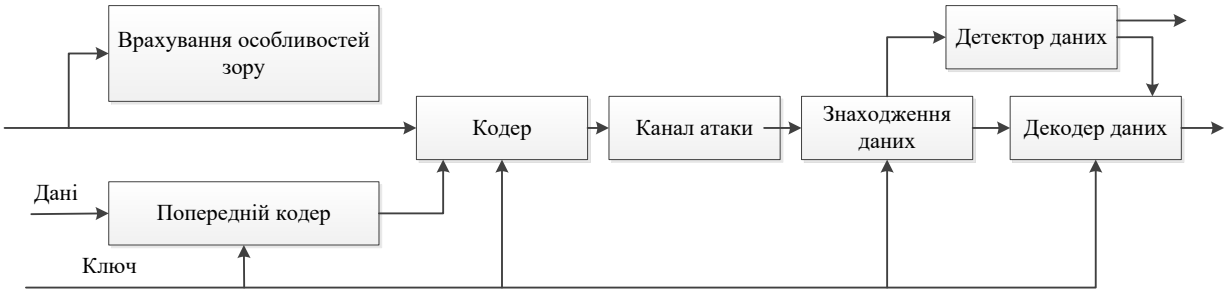


Рисунок 2.5 – Основні компоненти розробки програмного засобу

Далі наведено архітектуру програмного засобу, який складається з наступних модулів:

— інтерфейс користувача;

— модуль отримання даних;

— модуль вбудовування прихованого повідомлення у зображення;

— модуль вилучення прихованого повідомлення.

Щодо інтерфейсу користувача, то він відображає форми при введенні даних користувачем і відображення результатів роботи.

Щодо модуля отримання даних, то він відповідає за обробку введених даних і передачу їх у інші модулі для виконання наступних розрахунків.

Далі у модулі вбудовування прихованого повідомлення зображення вбудовують послідовності біт прихованого зображення.

У модулі вилучення повідомлення виконується прогноз яскравості вибраного кольору, після чого вилучаються послідовності біт вбудованого повідомлення.

Вбудовування одного біта повідомлення відбувається в один піксель зображеннятоді може змінюватись яскравість червоного, синього, чи зеленого кольору, які вибирає користувач, інші кольори залишаються без змін. Далі розглядається приклад вибору користувачем синього кольору, отже:

— R — яскравість червоного кольору;

— G — яскравість зеленого кольору;

— B —яскравість синього кольору;

— m — вбудований біт ('1' чи '0');

— x, y — координати пікселя.

Тоді $B^*_{x,y}$ — змінена залежно від біта, яка, вбудовується, яскравість синього кольору, обчислюється за наступною формулою

$$B^*_{x,y} = \begin{cases} B_{x,y} + 0.1 * (0.3 * R_{x,y} + 0.59 * G_{x,y} + 0.11 * B_{x,y}), \text{при } m_i = 1 \\ B_{x,y} - 0.1 * (0.3 * R_{x,y} + 0.59 * G_{x,y} + 0.11 * B_{x,y}), \text{при } m_i = 0 \end{cases}$$

При вилученні прогнозується відповідна яскравість синього кольору по сусіднім пікселям.

$$\overline{B_{x,y}} = \frac{\sum_{i=1}^{\sigma}(B_{x,y+i}+B_{x,y-i}+B_{x+i,y}+B_{x-i,y})}{4\sigma},$$

де $\sigma = 1 \div 3$.

Для безпосереднього отримання повідомлення, яке приховується використовується формула.

$$m_i = \begin{cases} 1, \text{при } B^*_{x,y} > \overline{B_{x,y}} \\ 0, \text{при } B^*_{x,y} < \overline{B_{x,y}} \end{cases}.$$

2.4 Розробка алгоритму роботи

У алгоритмі роботи передбачено забезпечення можливості користувача здійснювати приховування потрібних даних для подальшої їх передачі з врахуванням забезпечення захисту од витоку і несанкціонованого доступу із боку сторонніх користувачів.

Засіб надає можливість приховувати текстові дані у зображення. Це забезпечує зручність і ефективність використання розробки, тому, що різноманітні матеріали часто містять саме текстову і фото інформацію, яку доцільно зберігати разом для захисту від витоку.

Далі покроково розглядається алгоритм роботи:

— запуск виконуваного exe — файлу для вбудовування даних;

— вибір даних, які потрібно приховувати;

— вибір файлу, що слугуватиме контейнером;

— процес приховування даних;

— збереження результатів стеганографічного вбудовування;

— вибір файлу із якого потрібно вилучити дані;

— здійснення процесу вилучення, які були у стегоконтейнері;

— перегляд отриманих даних, які вилучені із стегоконтейнера;

— збереження отриманих даних, отриманих із стегоконтейнера;

— порівняння зображення до і після вбудовування інформаційних даних у стегоконтейнер;

— демонстрація результатів порівняння.

— завершення роботи з програмою.

Схематично даний алгоритм представлений на рисунку 2.6 та рисунку 2.7.

Вище наведений метод приховування інформації працює завдяки наведеними

вище алгоритмами у комплексі і забезпечує спільне зберігання фото і тексту в одному файлі зображення.

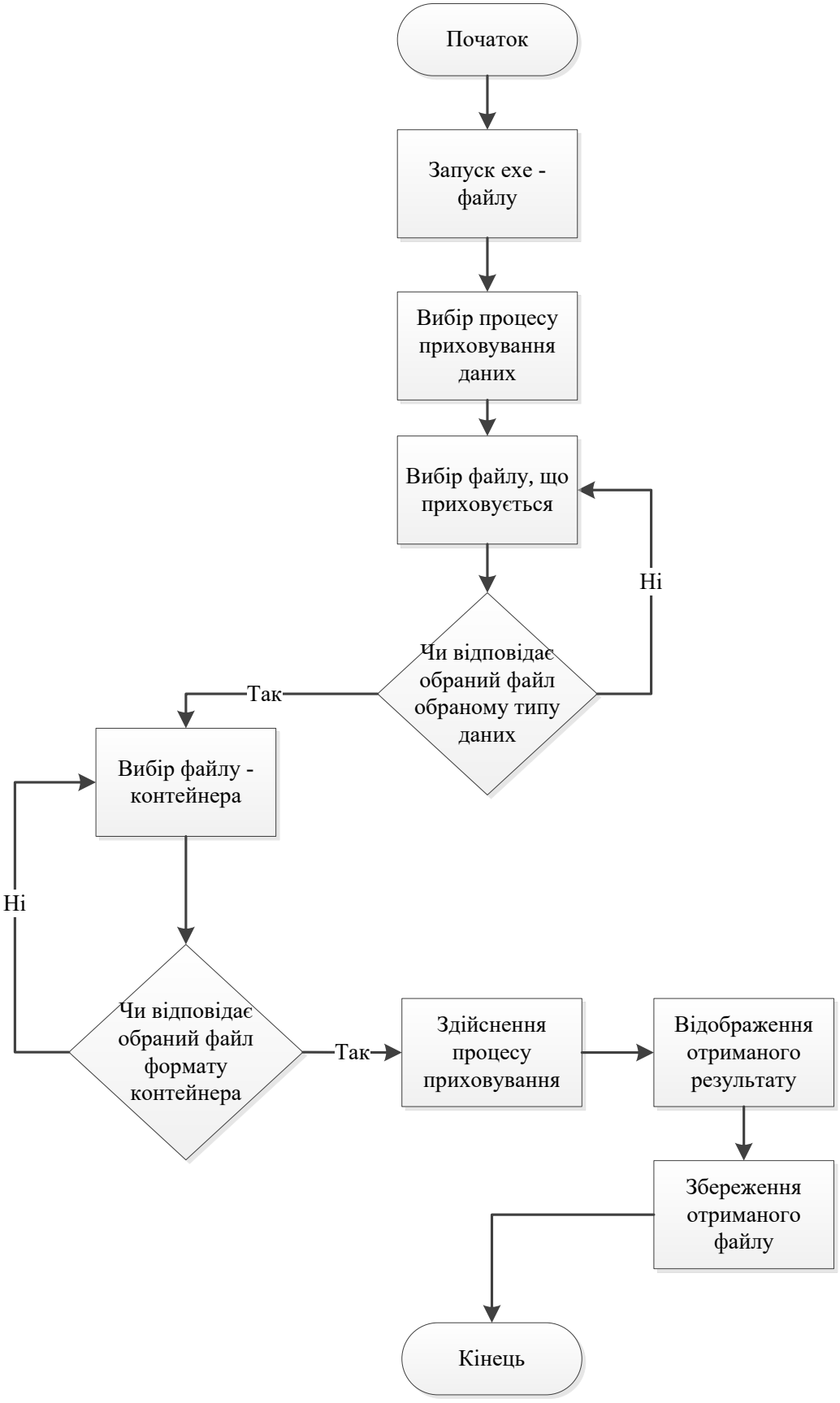


Рисунок 2.6 — Алгоритм роботи розроблюваного додатку

Рисунок 2.7 — Алгоритм роботи розроблюваного додатку

Крім того, для практичної реалізації розроблюваного додатку, спроектовано UML — діаграму класів (рис. 2.8).

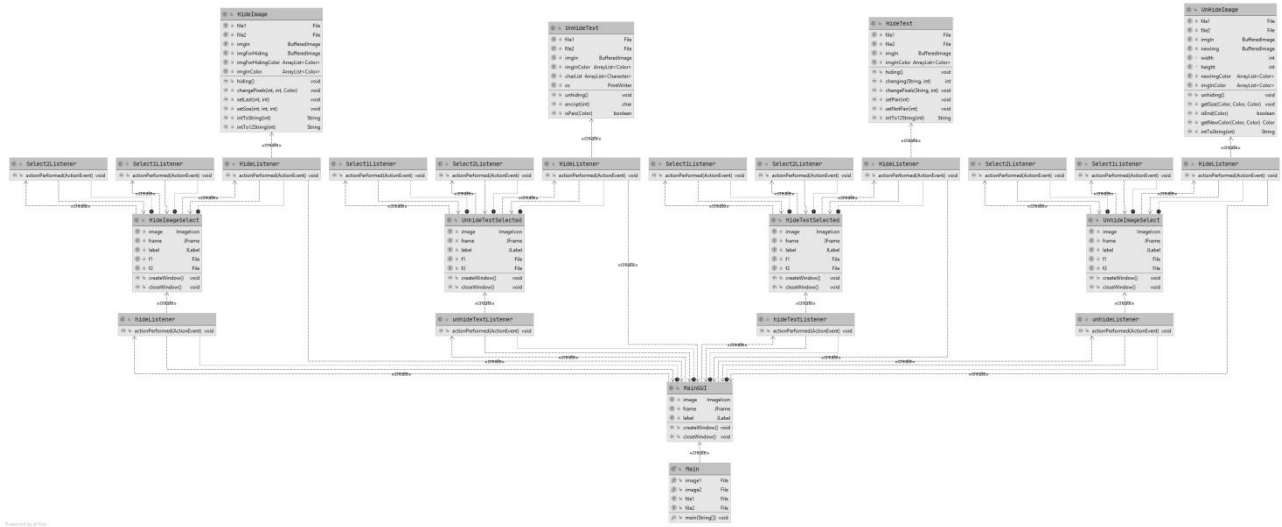


Рисунок 2.8 — UML – діаграма класів розробки

Із наведеної діаграми можна зазначити, що у практичній реалізації передбачаються такі класи для зображень, які приховуються, також класи для текстової інформації, які приховується, і також класи стегокотейнерів, в яких за допомогою певного методу приховуються та зберігаються дані.

Виходячи з цього, розроблений алгоритм роботи додатку є основою для його практичної реалізації, що який описаний нижяе.

2.5 Показники дослідження при зміні контейнеру

При розробці алгоритмів має місце врахування можливих модифікаці1 контейнера. У цій роботі розглядається приховання даних у зображеннях, тому, відповідно, буде матиме місце акцент на пошкодженні зображень.

Тому, одним з важливих показників, що повинні бути досліджені після реалізації додатку — це дослідження співвідношення рівня сигнал/шум, для того, щоб зрозуміти чи є погіршення певного показника та чи допустимі вони відносно відповідної ємкості контейнеру.

Співвідношення сигнал/шум (чи ССШ ВСШ, англ. SNR чи S/N, Signal-tonoise ratio) — міра відмінності, яка застосовується при визначенні того, наскільки сигнал спотворений шумом [8].

Співвідношення сигнал/шум визначається як відношення потужності сигналу

до потужності фонового шуму:

$$SNR = \frac{P_{signal}}{P_{noise}}$$

де P — середня потужність.

Сигнал та шум повинні бути виміряні в тій самій чи еквівалентній точці в системі, і в межах тієї ж самої смуги пропускання системи [8].

SNR, крім іншого, можна обчислити як відношення амплітуд:

$$SNR = \frac{P_{signal}}{P_{noise}} = \left(\frac{A_{signal}}{A_{noise}}\right)^2$$

де A — середньоквадратичне значення однієї амплітуди (англ. RMS – root mean square), її, зазвичай, беруть, як середньоквадратичне значення напруги.

Серед важливих показників можна ще виділити середню різницю значень пікселів, нормовану середню різницю значень пікселів і максимальне відношення «сигнал/шум» [8].

В подальшому наведено список вище зазначених показників спотворення і формули для їх обчислення.

Середня абсолютна різниця

$$AD = \frac{1}{XY}\sum_{x,y}\left|C_{x,y} - S_{x,y}\right|$$

Нормована середня абсолютна різниця

$$NAD = \frac{\sum_{x,y}\left|C_{x,y} - S_{x,y}\right|}{\sum_{x,y}\left|C_{x,y}\right|}$$

Якість зображення

$$IF = 1 - \frac{\sum_{x,y}\left(C_{x,y} - S_{x,y}\right)^2}{\sum_{x,y}\left(C_{x,y}\right)^2}$$

В рамках дослідження далі буде застосовано наведені вище показники для визначення змін у структурі контейнера до і після вбудовування даних у нього.

Таким чином, основна функція розробленого програмного забезпечення полягає не тільки в тому, щоб забезпечити можливість вставляти дані в зображення, але також забезпечити можливість секретної передачі, а також дозволити користувачеві проаналізувати стан контейнеру, вбудовану текстову або графічну інформацію. На основі даних, отриманих з таблиці, можна визначити, що обраний метод є найбільш придатним для вирішення цього завдання.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ЗАСОБУ ДЛЯ ВБУДОВУВАННЯ ДАНИХ

3.1 Вибір інструментарію

Для того, щоб реалізувати запропонований продукт, необхідно обрати засоби розробки. З цілей розроблюваного додатку та функціональних вимог можна визначити вимоги до його розробки, а саме:

— підтримка об'єктно-орієнтованого програмування;

— засоби для побудови графічного інтерфейсу;

— засоби для створення файлу, який виконується, що прив'язані до

— середовищ розробки.

Сьогодні на ринку операційних систем Windows існує безліч графічних середовищ обробки з різними інтерфейсами та методами програмування різними мовами. Серед них найпопулярніші інструменти: Microsoft Visual C++, Python, Java, C#.

Об’єктно-орієнтована технологія стала однією з головних розробок програмного забезпечення промислового масштабу. У всьому світі об’єктно-орієнтоване програмування використовується в таких різноманітних галузях, як управління банківськими транзакціями, автоматизація боулінгу, управління комунальними послугами та дослідження генетики людини. У багатьох випадках нові покоління операційних систем, систем керування базами даних, послуг телефонії, систем авіоніки та мультимедійних програм написані в об’єктно-орієнтованому стилі. У більшості таких проектів перевага віддавалася використанню об'єктно-орієнтованої технології просто тому, що не було іншого способу створити достатньо надійну та життєздатну систему. Кожна мова програмування має свої переваги та недоліки.

Python [29] є логічною та відносно простою мовою з мінімалістичним синтаксисом. У ньому невеликий набір основних правил, мова легко читається і нею не складно писати. Розробники написали для Python багато бібліотек, тому є можливість використовувати готові рішення у

розроблюваних проектах. Головний недолік у Python — невисока швидкість. Тому програми на Python працюватимуть дещо повільніше, ніж на інших мовах.

Java [29] — кросплатформна мова з великою кількістю бібліотек і великою спільнотою розробників. Кросплатформенність – це можливість один раз написати програму і одразу використовувати її на кількох операційних системах: Windows, Linux і MacOS.

C++ [29] — це кросплатформна мова сімейства C із широкою функціональністю. Часто операційні системи, драйвери та утиліти написані мовою C++. Вони створюють популярні настільні додатки серії Adobe і Office. Завдяки своїй швидкості та високій продуктивності C++ використовується для розробки комп'ютерних ігор.

C# (si-sharp) [29] — мова, винайдена Microsoft для створення програм для Windows. Це об'єктно-орієнтована мова – її важче вивчити, але простіше використовувати, наприклад, писати менше коду. За допомогою C# можна працювати з фреймворком WPF, який допомагає створювати «красиві» віконні програми. Наприклад, останні версії MS Office.

C# часто використовується для написання програм для Windows і створення комп'ютерних ігор. Наприклад, популярний двигун Unity працює на C#. Крім того, є можливість створювати системні програми та бібліотеки для C++. Ви можете використовувати C# для створення віконної програми для Windows, наприклад калькулятора або невеликої гри. Але його складніше вивчити, ніж мови для створення мобільних додатків.

Виходячи з поставлених в роботі задач, доцільно використовувати мову об'єктно-орієнтованого програмування C#.

Середовищем розробки обраної мови програмування було обрано Visual Studio [30], що має редактор вихідного коду з підтримкою технології IntelliSense та можливістю найпростішого рефакторинга коду. Вбудований засіб налагодження може працювати як налагоджувач на рівні вихідного коду, так і на рівні машини. Інші вбудовані інструменти включають редактор

форм для спрощення створення GUI програми, веб-редактор, конструктор класів і конструктор схем бази даних. VisualStudio дозволяє створювати та підключати сторонні програми (плагіни) для розширення функціональності на всіх рівнях, включаючи додавання підтримки систем контролю версій вихідного коду (таких як Subversion і VisualSourceSafe), додавання нових інструментів (наприклад, для редагування коду) . та візуальне оформлення).

Головною перевагою Visual Studio 2022 є продуктивність. Середовище дає змогу будувати різні програми на основі однакового набору набутих умінь і навичок.

Таким чином, надалі в роботі на основі обраних засобів програмування буде здійснено їх налаштування та реалізовано програмну розробку інструменту розміщення повідомлення на фотографіях зі збереженням структури контейнера.

3.2 Налаштування середовища розробки

Щоб полегшити написання, а також тестування та налагодження програмного коду, нерідко використовують спеціальні середовища розробки, зокрема Visual Studio.

Для створення програми на C++ нам потрібні, як мінімум, дві речі: текстовий редактор для набору коду та компілятор для перетворення цього коду на додаток. Для компіляції необхідно запускати консоль або термінал. Однак є і зручніший спосіб — використання різних середовищ розробки або IDE. Вони зазвичай містять вбудований текстовий редактор, компілятор і дозволяють скомпілювати і запустити програму по одному кліку миші, а також мають ще безліч різних допоміжних можливостей.

Для програмування під Windows найбільш популярним середовищем розробки, якщо говорити про C++, є Visual Studio. Це середовище можна знайти за посиланням https://visualstudio.microsoft.com/ru/vs/community/ .

Після завантаження та запуску інсталятора Visual Studio в ньому необхідно відзначити пункт Розробка класичних додатків на C++ (рис. 3.1.).

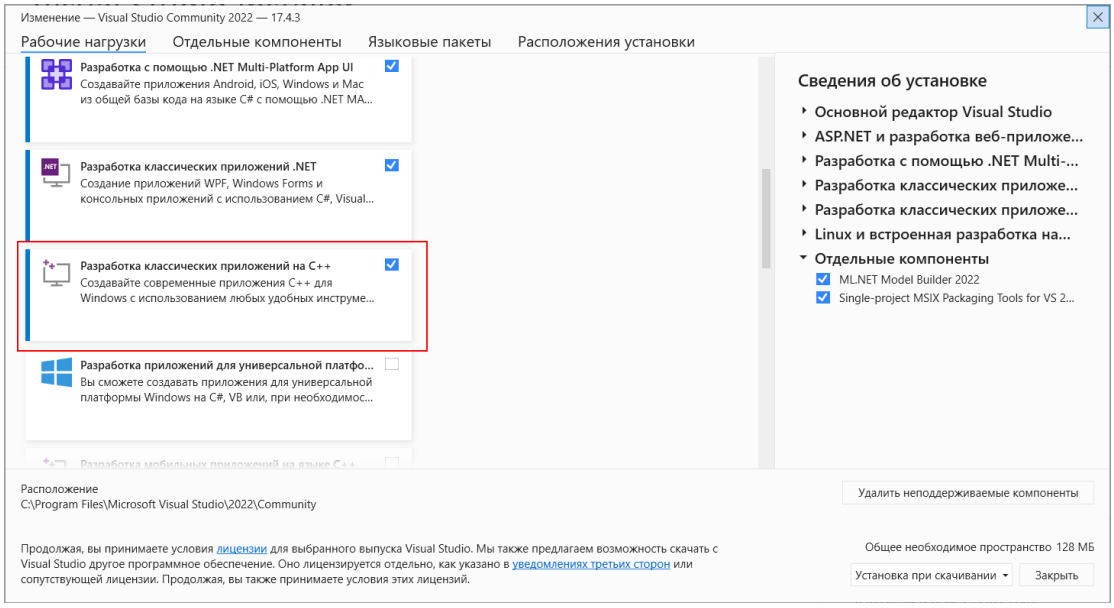


Рисунок 3.1 — Вікно розміщення "Розробка класичних додатків на C++»

Вибравши всі необхідні пункти, натисніть кнопку ОК для запуску установки. Після інсталяції Visual Studio створимо проект. Для цього відкриємо Visual Studio. На стартовому екрані серед шаблонів проектів для мови C++ виберемо тип Console App , який представляє шаблон консольної програми (рис. 3.2).

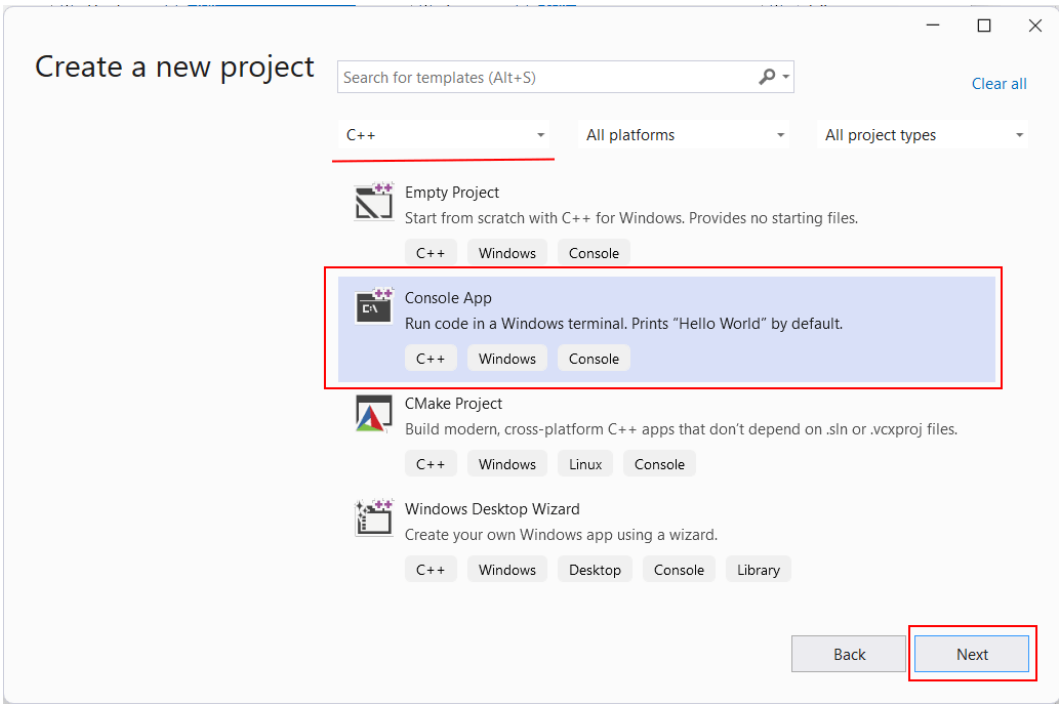


Рисунок 3.2 — Вікно вибору шаблон консольної програми Console App

На наступному екрані в полі для імені проекту дамо проекту ім'я, (наприклад HelloApp) і також можна вказати розташування проекту. І потім натиснемо Create для створення проекту (рис. 3.3). Після цього Visual Studio створить типовий проект консольної програми C++ (рис. 3.4).

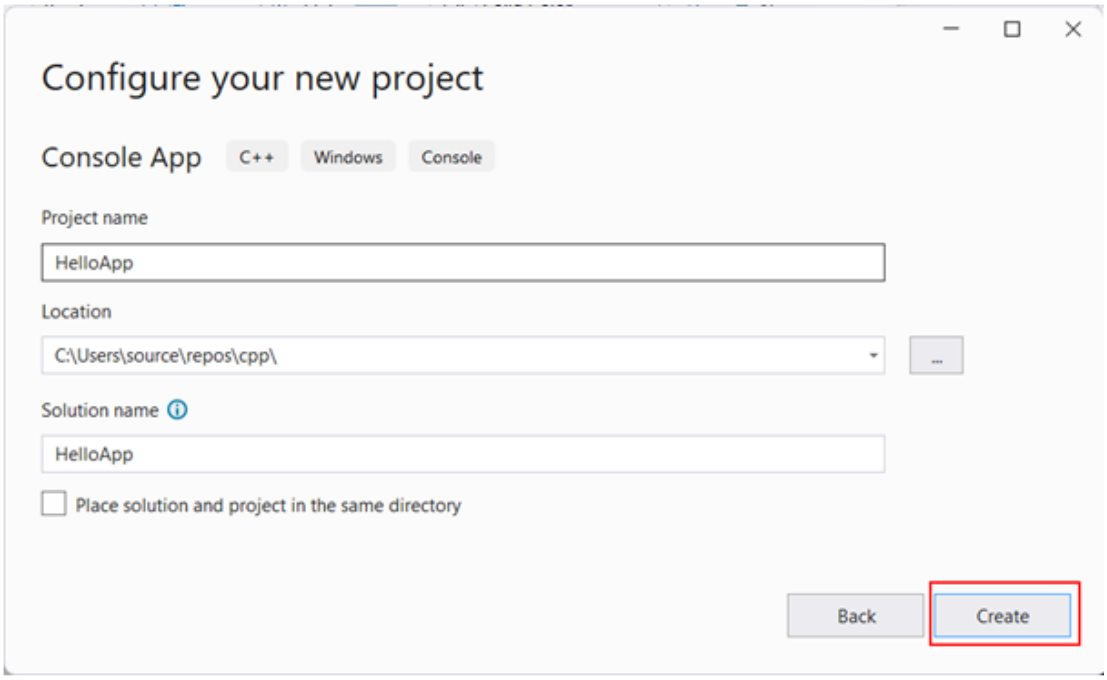


Рисунок 3.3 — Вікно налаштування імені та розташування проекту

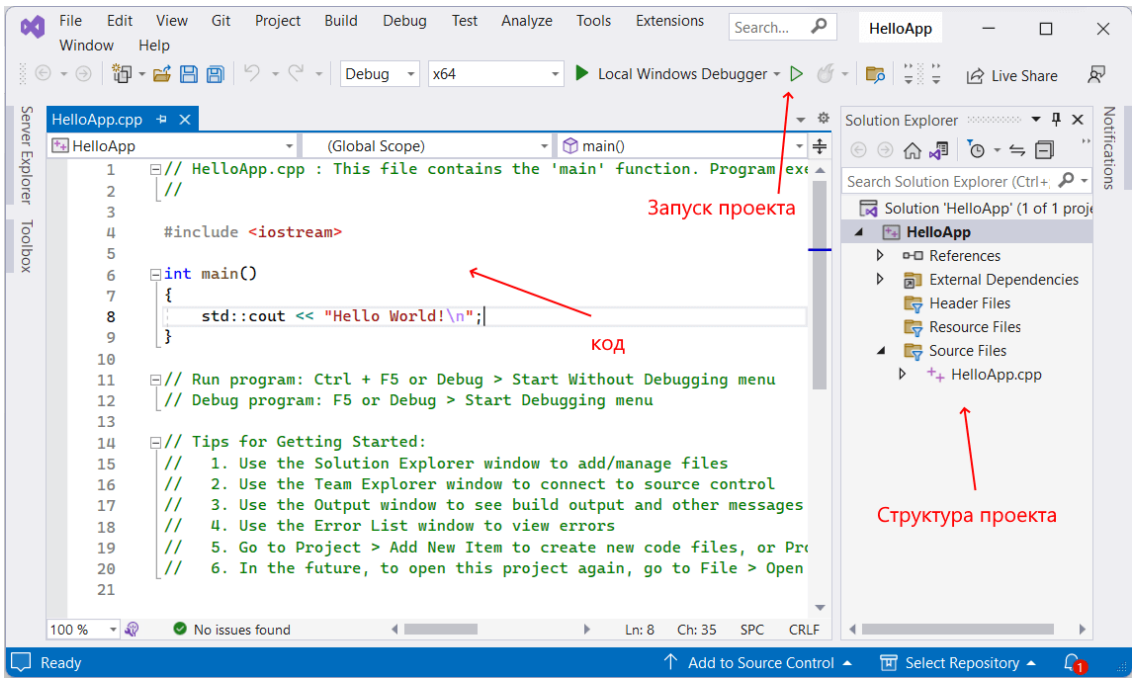


Рисунок 3.4 — Вікно типового проект консольної програми C++

Праворуч у вікні Solution Explorer відображається структура проекту. Насправді вікно Solution Explorer містить рішення. У цьому випадку воно називається HelloApp. Рішення може містити декілька проектів. За замовчуванням у нас один проект, який має те саме ім'я – HelloApp. У проекті є низка вузлів:

— External Dependencies — відображає файли, які використовуються у файлах вихідного коду, але не є частиною проекту;

— Header Files — призначена для зберігання заголовних файлів з розширенням .h;

— Resource Files — призначена для зберігання файлів ресурсів, наприклад зображень;

— Source Files — зберігає файли з вихідним кодом.

За замовчанням каталог Source Files містить один файл з вихідним кодом — HelloApp.cpp (назва проекта+ розширення файлу .cpp — зазвичай вихідні файли на C++ мають розширення .cpp).

HelloApp.cpp містить код мовою C++, і саме цей код можемо побачити ліворуч у текстовому редакторі Visual Studio.

Тепер запустимо програму. Для цього в Visual Studio натискаємо на поєднання клавіш Ctrl+F5 або виберемо пункт меню Debug -> Start Without Debugging (рис. 3.5).

І в результаті Visual Studio передасть вихідний код компілятору, який скомпілює з коду виконуваний файл exe, який потім буде запущений на виконання (рис. 3.6).

Після цього на жорсткому диску в папці рішення в каталозі \x64\Debug скомпільований файл exe, який можемо запускати незалежно від Visual Studio.

У цьому випадку файл HelloApp.exe якраз і представляє скомпільований виконуваний файл. І, крім того, в тій же папці автоматично генерується допоміжний файл — HelloApp.pdb , який містить інформацію налагодження.

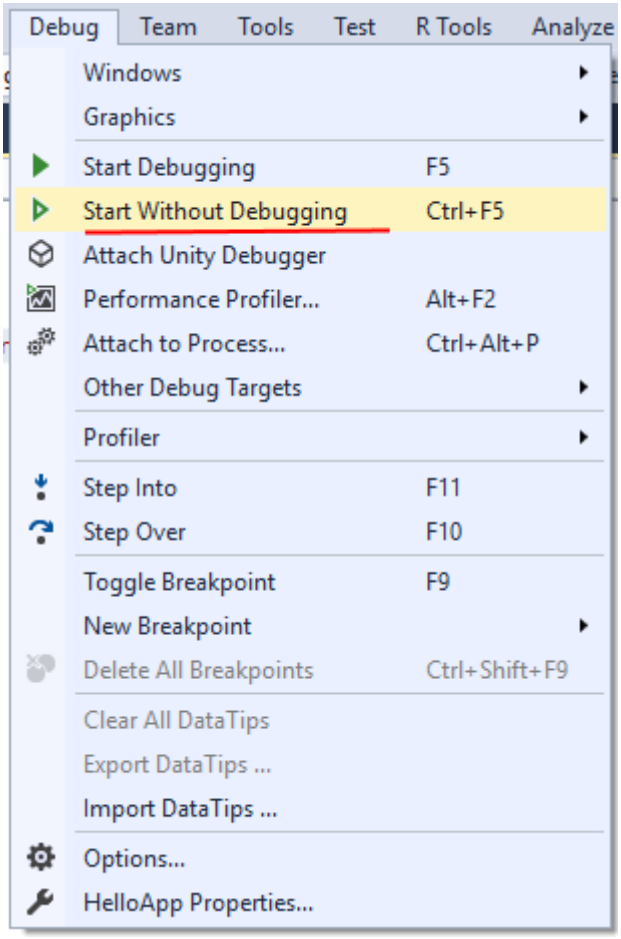


Рисунок 3.5 — Вікно запуску компіляції програми

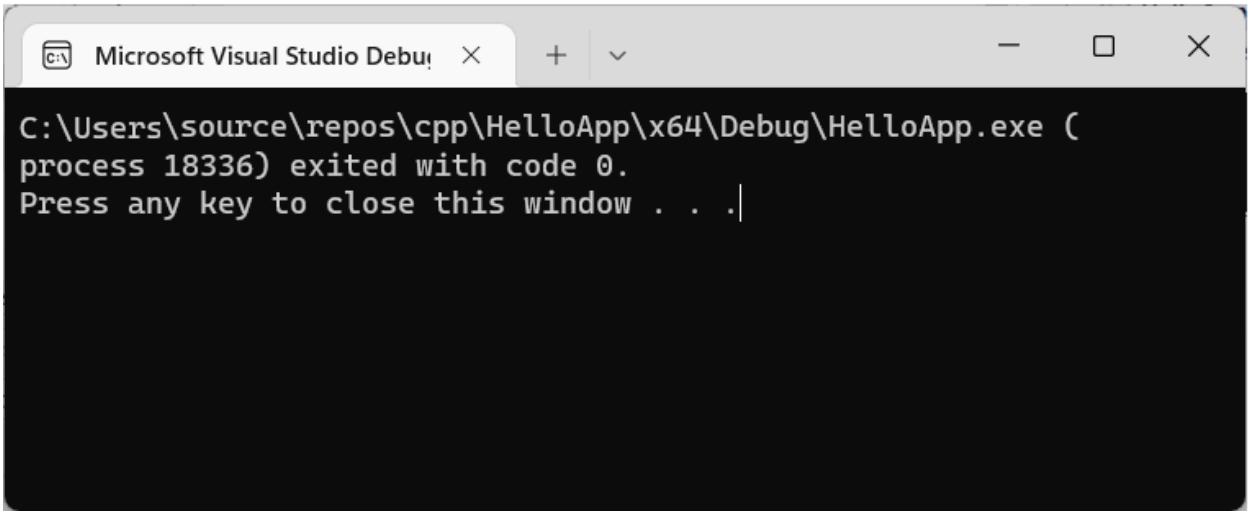


Рисунок 3.6 — Вікно запуску виконуваного файлу

Для мови C++ є кілька стандартів, кожен із яких додає деякі додаткові можливості. І Visual Studio дозволяє встановити стандарт, який буде

використовуватися при компіляції програми. Для цього перейдемо до властивостей проекту (рис. 3.7, рис. 3.8).

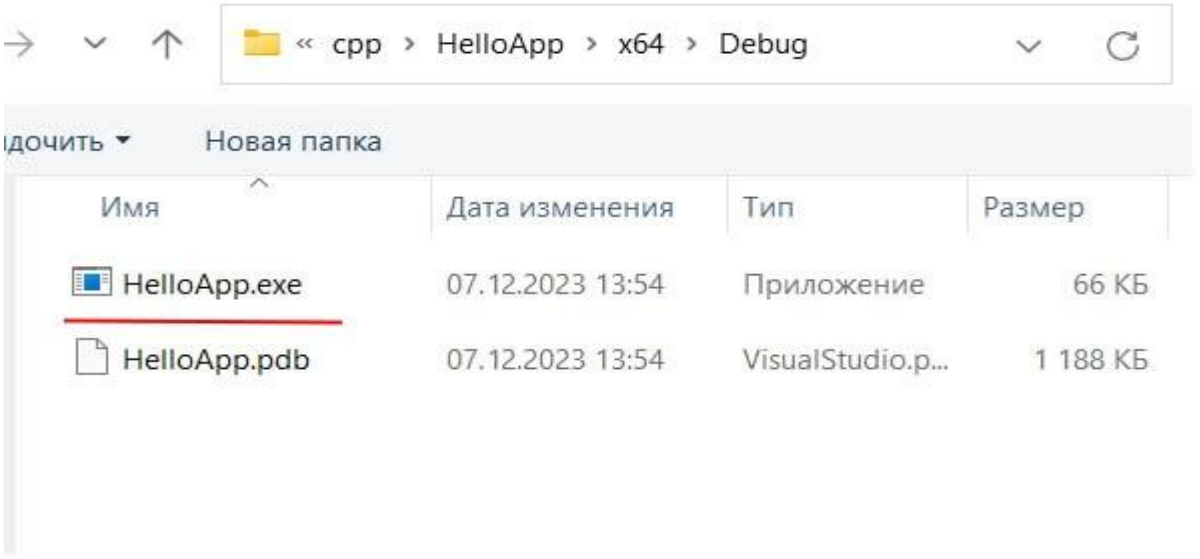


Рисунок 3.7 — Скомпільований файл *.exe каталозі \x64\Debug

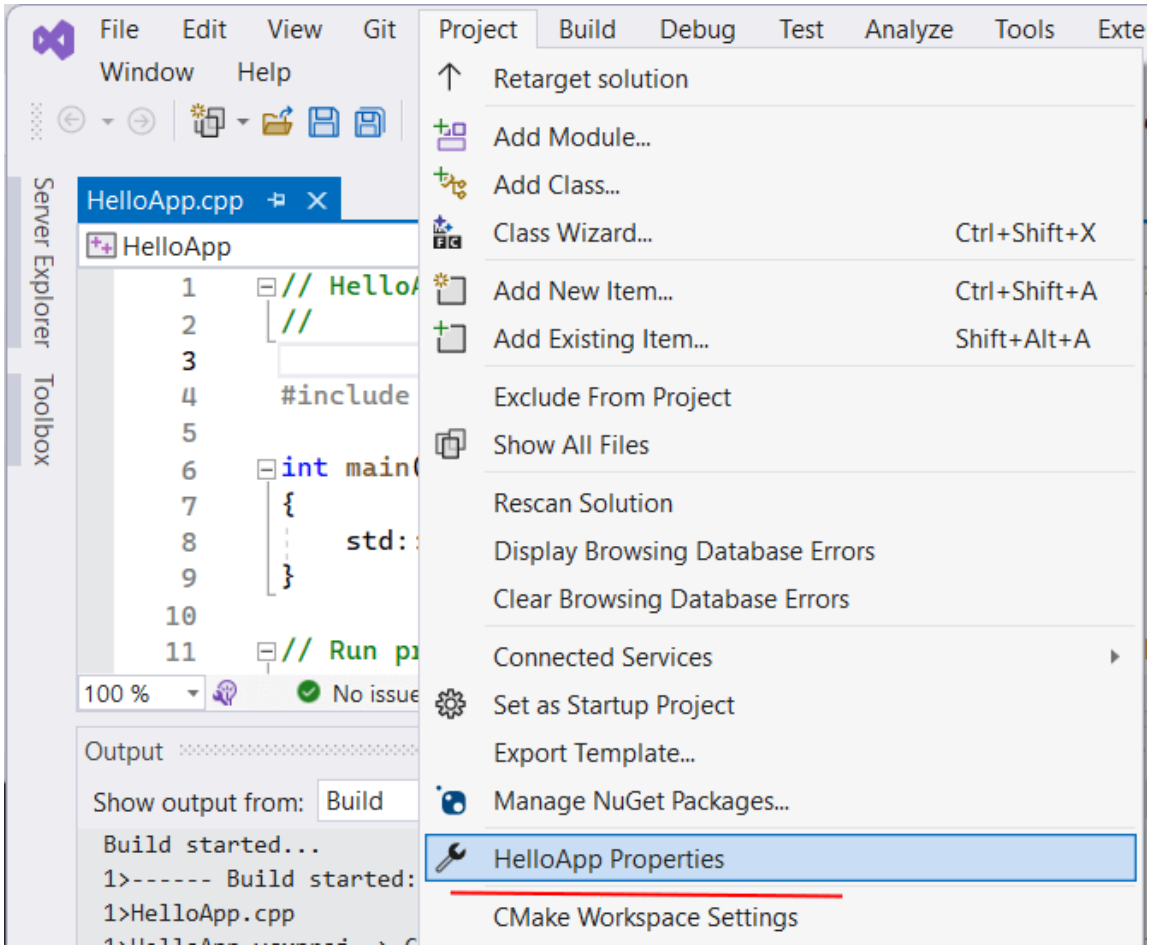


Рисунок 3.8 — Меню властивостей проекту

А у вікні властивостей перейдемо до пункту Configuration Properties -> C/C++ -> Language . На вікні властивостей за допомогою опції C++ Language Standard можна задати стандарт мови, який хочемо використовувати (рис. 3.9).

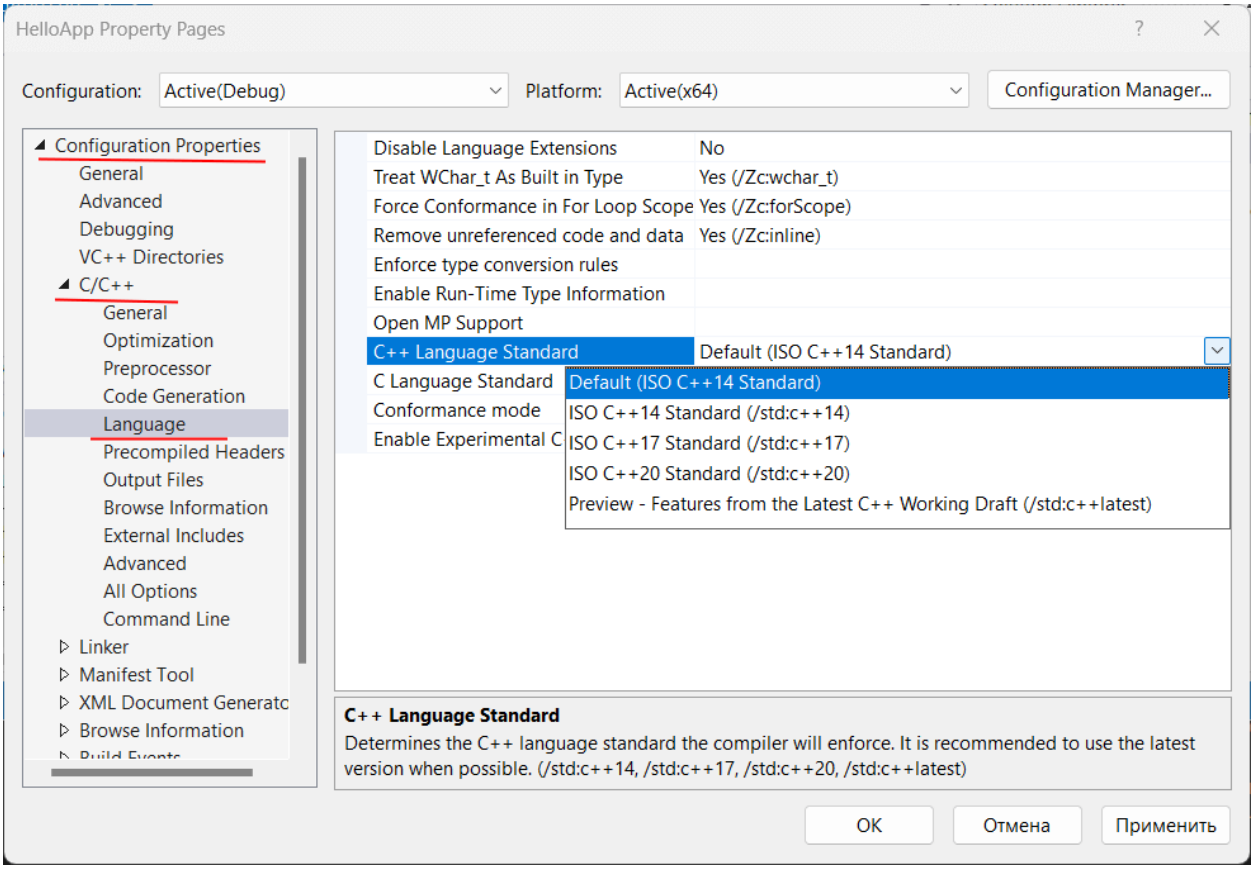


Рисунок 3.9 — Меню задання стандарту мови

3.3 Розробка програмного забезпечення

Вибравши засоби програмування та налаштувавши їх для реалізації програмного забезпечення, призначеного для розміщення повідомлення на фотографіях зі збереженням структури контейнера та аналізом робочих завдань, переходимо до практичної розробки програмного забезпечення.

Отже, оскільки основним завданням програми є надання користувачам функцій для введення та вилучення даних із зображення, ми розробляємо програму за методом авторів Darmstedter, Delaigle, Quiswater та Mack, що

дозволяє досягти компромісу між стійкістю стеганосистеми до спотворень, якістю вбудовування і, звичайно ж, обчислювальною складністю алгоритму.

У розробці програмного забезпечення для відображення відповідних форм і діалогових вікон використовуються два основні класи:

public partial class Form1: Form;
public class Algorithm.

Це бібліотеки, які використовуються при реалізації програмного засобу (для створення компонентів, отримання даних, обробки тексту, створення завдань, реалізації інтерфейсу):

```
using System;
using System.ComponentModel;
using System.Collections.Generic;
using System.Drawing;
using System.Data;
using System.Text;
using System.Linq;
using System.Windows.Forms;using System.Threading.Tasks;
```

Для початку роботи з програмою користувач завантажує образ контейнера для подальшої обробки, ця функція описана наступним чином:

```
private void trim(ref Bitmap image, int sizeSegment)
      {
         int x = image.Width % sizeSegment;
         int y = image.Height % sizeSegment;
         Size newSize = new Size(image.Width - x, image.Height - y);
         Bitmap b = new Bitmap(newSize.Width, newSize.Height);
         for (int i = 0; i < b.Width; i++)
         {
            for (int j = 0; j < b.Height; j++)
            {
               b.SetPixel(i, j, image.GetPixel(i, j)); } }
```

Для початку роботи з програмою користувач завантажує образ контейнера для подальшої обробки, ця функція описана наступним чином:

```
private void inliningMess(byte[] message, ref List<double[,]> DKP, int P)
      {
         int key = LengthOfMessage;
         List<int> possiblePositions = new List<int>();
         for (int i = 0; i < DKP.Count; i++)
         {
            possiblePositions.Add(i);
         }
```

Нижче наведено клас, який обробляє надсилання повідомлення про прогрес програмного

засобу користувачеві:

```
private void sendMessToForm(String mess)
      {
        if (_form.InvokeRequired)
          _form.Invoke((MethodInvoker)delegate { _form.sendMess(mess); });
        else
          _form.sendMess(mess);  }
```

Наступний етап роботи програми — нормалізація зображення, для подальшої ідентифікації компонентів і поділу на блоки. Ця функція реалізується наступним чином:

```
private double[,] normaliz(double[,] odkp)
      {    double min = Double.MaxValue, max = Double.MinValue;
        for (int i = 0; i < odkp.GetLength(0); i++)
        {
          for (int j = 0; j < odkp.GetLength(1); j++)
          {
            if (odkp[i, j] > max)
              max = odkp[i, j];
            if (odkp[i, j] < min)
              min = odkp[i, j];    }     }
```

Після завантаження основних компонентів для реалізації операції введення даних зображення та встановлення основних параметрів відбувається побайтовий аналіз повідомлення.

```
Byte res = Byte.MinValue;
        for (int j = 0, m = 1; j < 8; j++, m *= 2)
        { if (bits[j] == 1)
          { if (j == 0)
            { res = (byte)m;
            } else { res += (byte)m; }
          } if (bits[j] == 0)
          { if (j == 0)
            { res = (byte)0;
            } else { res += (byte)0; }    }
```

Процес визначення довжини повідомлення і розбиття його на блоки з подальшим застосуванням їх при введенні повідомлення записується так:

```
public int LengthOfMessage;
      Bitmap picture;
      int SizeOfSegment=8;
      int P=25;
      string ComponentOfEmbedding;
```

Вибір компонентів зображення для введення текстових даних базується на колірній моделі RGB і доступний користувачеві:public Algorithm(int size, int p, string component,Form1 form)

```
{
   P = p;
   SizeOfSegment = size;
   ComponentOfEmbedding = component;
   _form = form;
   DeterminePointsOfCoefficients();
}
```

Обробка користувачем коефіцієнтів при встановленні параметрів введення даних в зображення записується за такою кодовою послідовністю:

```
private void DeterminePointsOfCoefficients()
      {    if (SizeOfSegment == 2)
           {  p1 = new Point(1, 0);
              p2 = new Point(1, 1);     }
           else if (SizeOfSegment == 4)
           {  p1 = new Point(3, 2);
              p2 = new Point(2, 3);     }
           else
           {   p1 = new Point(6, 3);
    p2 = new Point(3, 6);    }
```

Обробка користувачем коефіцієнтів при встановленні параметрів введення даних в зображення записується відповідно до послідовності наступних кодів:

```
sendMessToForm("Почати вставляти повідомлення...");
      picture= new Bitmap(im);
      if ((picture.Width % SizeOfSegment) != 0 || (picture.Height % SizeOfSegment) != 0)
      {
        trim(ref picture, SizeOfSegment);
      }
```

Формування ключа, обов’язкового для для послідующого вилучення вбудованих даних із повідомлення реалізовано таким циклом:

```
for (int i = 0; i < message.Length; i++)
      {
         int [] bitsOfSymbol = ConvertToBits(message[i]);
         for (int j = 0; j < 8; j++)
         {   int currentBit = bitsOfSymbol[j];
            key = GetKey(key, possiblePositions.Count);
            int pos = possiblePositions[key];
            possiblePositions.RemoveAt(key);
```

Генерація ключа для подальшого вилучення включеної інформації з повідомлення здійснюється за такими циклами:

```
private double[,] odkp(double[,] dkp)
      { int n = dkp.GetLength(0);
        double[,] result = new double[n, n];
        double temp = 0;
        for (int v = 0; v < n; v++)
        { for (int u = 0; u < n; u++)
          { temp = 0;
            for (int i = 0; i < n; i++)
            { for (int j = 0; j < n; j++)
          { temp += FindCoefficient(i) * FindCoefficient(j) * dkp[i, j] * Math.Cos(Math.PI * i * (2 *
v + 1) / (2 * n)) * Math.Cos(Math.PI * j * (2 * u + 1) / (2 * n)); }
```

Далі завершується процес вбудовування даних у зображення:

```
modifImage.SetPixel(i, j, Color.FromArgb((byte)Math.Round(newModifArray[i, j]),
picture.GetPixel(i, j).G, picture.GetPixel(i, j).B));
```

Наступним кроком є створення зображення з вбудованим повідомленням, сповіщення про це в полі та відображення цього зображення користувачеві:

```
Double[,] newModifArray = new Double[x, y];
        BuildNewB(ODKP, ref newModifArray, x, y,SizeOfSegment);
        newModifArray = normaliz(newModifArray);
        Bitmap modifImage=new Bitmap(picture);
```

Розрахунок параметрів зображення з вбудованим повідомленням виводиться через такий масив:

```
int x = modifPicture.Width;
          int y = modifPicture.Height;
          Byte[,] ArrayForEmbedding = new Byte[x, y];
          for (int i = 0; i < x; i++)
```

Визначення елементів, до яких включено повідомлення, здійснюється за допомогою наступного методу:

```
if (ComponentOfEmbedding == "Синій")
                { ArrayForEmbedding[i, j] = modifPicture.GetPixel(i, j).B;
                } else if (ComponentOfEmbedding == "Зелений")
                { ArrayForEmbedding[i, j] = modifPicture.GetPixel(i, j).G;
                }       else
        { ArrayForEmbedding[i, j] = modifPicture.GetPixel(i, j).R;
}
```

Сегментація компонентів зображення:

separation(ArrayForEmbedding, C, x, y, SizeOfSegment);

Виконання дискретно-косинусного перетворення досягається за допомогою відповідного поля:

```
List<double[,]> DKP = new List<double[,]>();
            foreach (byte[,] b in C)
            {
               DKP.Add(dkp(b));
            }
```

Згенероване повідомлення створюється лише за допомогою ключа, згенерованого під час введення, для вказаної функції використовується наступний метод:

```
int key = LengthOfMessage;
            List<int> possiblePositions = new List<int>();
            for (int i = 0; i < DKP.Count; i++)
            {
               possiblePositions.Add(i);
            }
            for (int i = 0; i < LengthOfMessage; i++)
            {
               int[] bits = new int[8];
               for (int j = 0; j < 8; j++)
               {
                  key = GetKey(key, possiblePositions.Count);
```

Створення сповіщення користувача про готовність вихідного повідомлення за допомогою масиву даних і повідомлення виводиться у відповідному полі:

```
for (int i = 0; i < message.Count; i++)
            {
               char ch = Encoding.GetEncoding(1251).GetString(message.ToArray())[i];
               result += ch;

  }
```

Таким чином, на даному етапі роботи завершено програмну реалізацію розробленого програмного засобу для вставки повідомлення у фотографії зі збереженням структури контейнера. У наступному розділі ми обговоримо візуалізацію розробки програмного забезпечення для користувача програми.

3.4 Створення графічного інтерфейсу

Графічний інтерфейс програмного додатку є одним з обов'язкових компонентів великої кількості сучасних програмних продуктів, які в основному орієнтуються на напрямок кінцевого користувача такої розробки.

До зовнішнього вигляду графічного інтерфейсу висуваються високі вимоги, як з точки зору інженерного, так і естетичного представлення, а при проектуванні структури програмного забезпечення орієнтуються на можливості людини.

В абсолютній більшості графічний інтерфейс реалізовано в інтерактивному режимі роботи оператора з програмними продуктами, які функціонують у середовищі операційної системи Windows і представляють собою структуру випадаючого меню з використанням комп’ютерної миші. і клавіатура як інструмент управління розробкою програмного забезпечення.

Графічний інтерфейс передбачає, що робота розробника програмного забезпечення здійснюється в результаті взаємодії з екранними формами, які містять об'єкти управління, панелі інструментів з піктограмами режимів і командами обробки.

Графічний інтерфейс користувача, адаптований до вимог стандарту, повинен відповідати таким правилам:

— інформаційно-технологічне забезпечення роботи оператора з програмними продуктами, а тому має знайомі та зрозумілі пункти меню, що відповідають компетенціям обробки;

— для зручності використання оператором функції діалогових вікон усі елементи мають бути розміщені у звичній послідовності;

— розробка інтерфейсу повинна бути спрямована на кінцевого користувача, який взаємодіє з нею на зовнішньому рівні;

— задовольняти правило «шістки», яке вказує на те, що в одній стрічці може бути не більше шести пунктів, кожен з яких має не більше шести варіантів.

Дизайн графічних об'єктів, особливо інформаційної графіки, важливий при проектуванні інтерфейсу, тому що такі об'єкти повинні розміщуватися в діалоговому вікні, а не за його межами.Також слід зазначити, що основна перевага хорошого дизайну програмного інтерфейсу полягає в тому, що користувач завжди відчуває, що він контролює програмне забезпечення, а не навпаки.

Розглянемо більш детально конструктивні особливості діалогових вікон розробленої програми для введення повідомлення на фото зі збереженням структури контейнера. Таким чином, обробка в цілому має два

ключових діалогових вікна, які є невід'ємною частиною коректної роботи програми.

Отже, після запуску файлу додатку відкривається головне діалогове вікно, яке має назву «Робота із зображеннями». Розглянемо далі його структуру (рис. 3.1).

У цьому діалоговому вікні в його першій частині є поле для додавання оригінального зображення, яке згодом слугуватиме стегоконтейнером для вставки повідомлення.

Нижче наведено наш конфігураційний блок для розміщення даних. У відповідному полі користувач може вибрати шлях до файлу з даними для включення, налаштувати компоненти для включення, кількість блоків, різницю коефіцієнтів.

Область відображення окремої клавіші програмою і функціональної кнопки «Приховати повідомлення» ще нижче.

Подібно до конфігураційного блоку для введення інформації, є блок для вилучення даних із зображення (місце завантаження зображення з вбудованим повідомленням, параметри вилучення, поле для введення ключа та відповідна функціональна кнопка для вилучення повідомлення) .

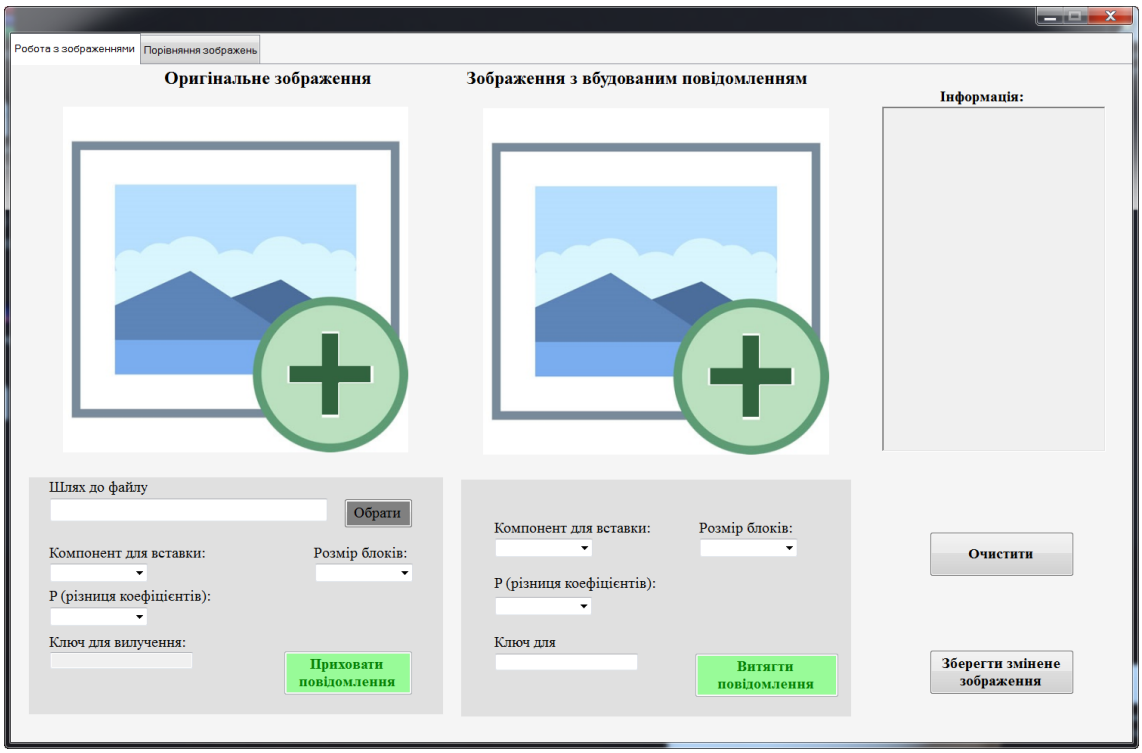


Рисунок 3.1 — Вигляд вікна програми для приховування повідомлення

Подібно до конфігураційного блоку для введення інформації, є блок для вилучення даних із зображення (місце завантаження зображення з вбудованим повідомленням, параметри вилучення, поле для введення ключа та відповідна функціональна кнопка для вилучення повідомлення) .

Після запуску процесів «Приховати повідомлення» і «Видалити повідомлення» вся історія виконаних в програмі дій буде відображена в окремому полі в лівій частині вікна.

У правій нижній частині вікна розташовані функціональні кнопки збереження обробленого зображення для подальшої передачі та кнопка очищення вікна повідомлень для подальшої роботи.

Наступне діалогове вікно програми призначене для порівняння зображень і призначене для показу змін, які відбуваються в зображенні контейнера порівняно з оригіналом.

Щоб побачити отримані зміни, користувач повинен завантажити оригінальне зображення та зображення з прихованим вмістом, а потім натиснути функціональну кнопку «Порівняти зображення».

У правій частині вікна також відображається хід розпізнавання подій, зображення змін із зображенням у полі «Зображення №3» у вигляді чорно-білого зображення з областями вхідних даних. Відображення описаного діалогового вікна показано на рисунку 3.2

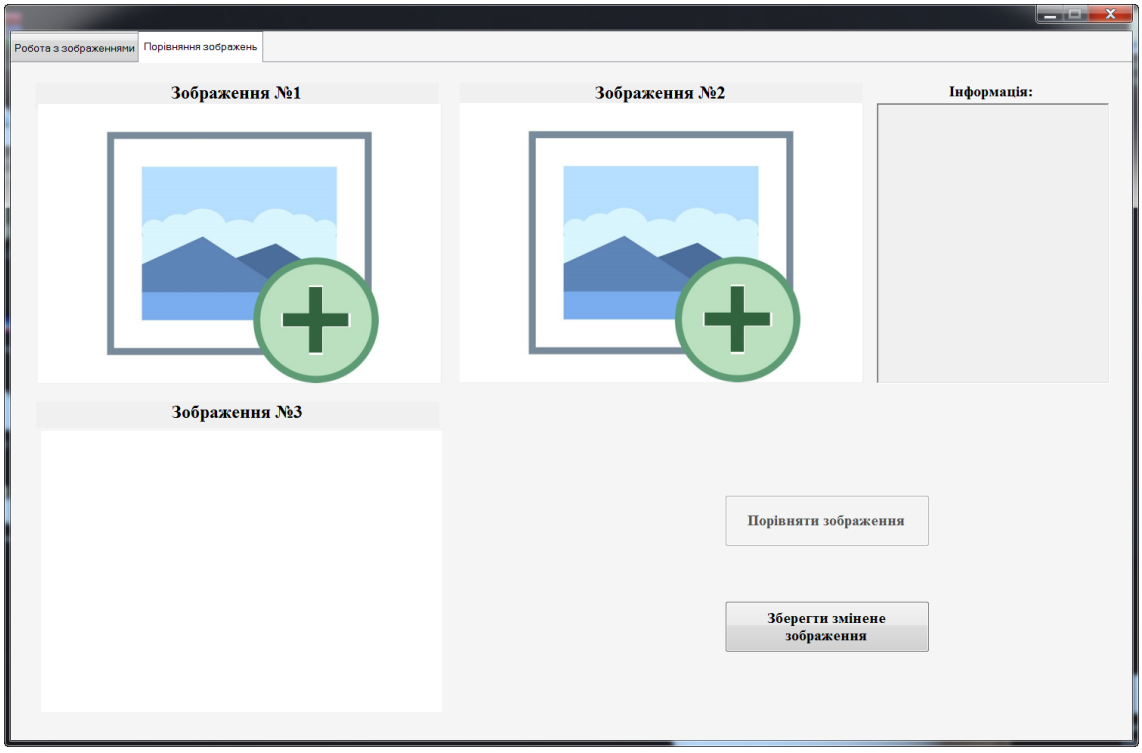


Рисунок 3.2 — Вигляд вікна програми для зрівнювання зображень

Таким чином, у цьому підрозділі представлено основні особливості розробки графічного інтерфейсу користувача розробленого програмного засобу. Більш детально процес роботи в програмі та досягнення практичних результатів буде описано в наступному підрозділі.

4 ТЕСТУВАННЯ ТА ДОСЛІДЖЕННЯ ЗАСОБУ ДЛЯ ВБУДОВУВАННЯ ДАНИХ

Для аналізу практичних результатів роботи розглянемо роботу розробленої програми на прикладі реалізації процесів включення та вилучення даних, а також порівняння зображень.

4.1 Розміщення даних

Відкрийте виконуваний файл програми і в діалоговому вікні «Робота з зображеннями» в поле вихідного зображення завантажте файл у форматах .png або .bmp для подальшої обробки.

Наступним кроком є вибір текстового файлу .txt із файлової системи комп’ютера та завантаження його в програму за допомогою функціональної кнопки «Вибрати файл».

Далі користувачеві необхідно встановити основні параметри введення даних в зображення, тобто вибрати компонент, який буде включено, визначити параметр P (різниця компонентів) і розмір блоків розподілу зображення.

Після виконання всіх дій необхідно натиснути кнопку «Сховати повідомлення» (рис. 4.1).

Після підтвердження цієї дії в правій частині діалогового вікна в полі «Інформація» для користувача відображається покрокова послідовність виконуваних подій (початок процесу вбудовування, сегментація забруднення, виконання дискретно-косинусне перетворення, фактичне зміщення повідомлення, виконання зворотного дискретного перетворення -косинус, форма зміненого зображення та повідомлення про те, що розміщення зображення завершено.

Важливо, що коли виконується функція вбудовування даних, генерується ключ для вилучення даних із обробленого зображення в майбутньому. Користувач повинен його знайти, оскільки без зазначеного параметра неможливо отримати інформацію (рис. 4.2).

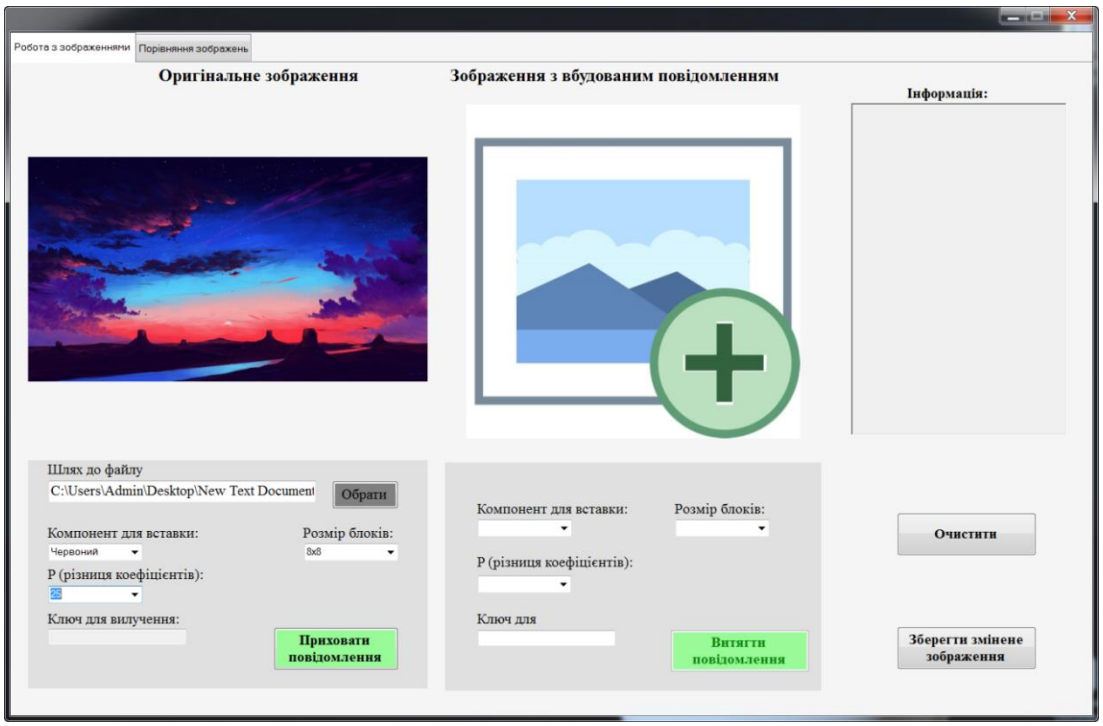


Рисунок 4.1 — Вигляд вікна при вбудовуванні даних

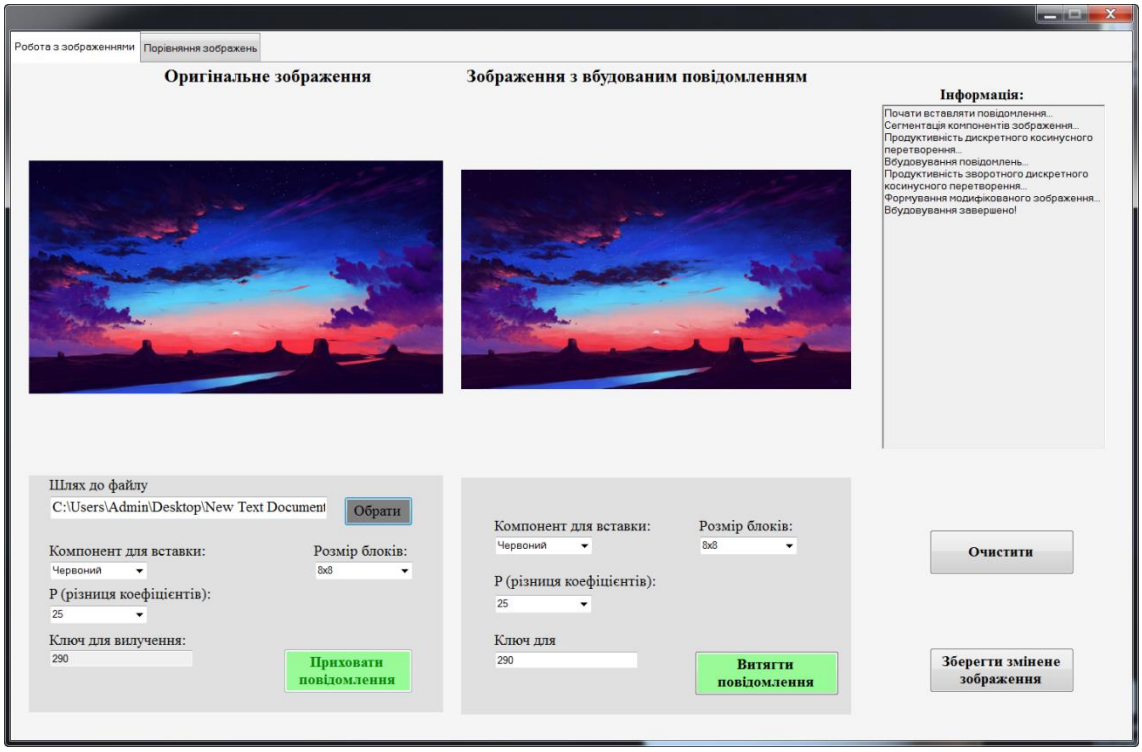


Рисунок 4.2 — Наступний вигляд вікна додатку при вбудовуванні даних

Після завершення цього процесу буде активовано кнопку «Зберегти зображення» (з прихованим текстом), щоб користувач міг перенести його далі, а також кнопку «Очистити» для видалення інформації з діалогового вікна та можливості обробки інші зображення.

Незважаючи на те, що алгоритм, реалізований у додатку, передбачає вставку великих текстових даних, якщо обраний для вставки файл перевищує допустимий розмір, користувач отримає відповідне системне сповіщення (рис. 4.3).

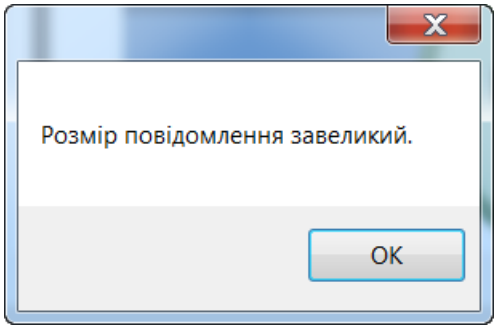


Рисунок 4.3 — Сповіщення про розміру файлу

У разі натискання користувачем функціональної кнопки для підтвердження початку процесу вставки (або вилучення) без заповнення обов’язкових полів система також виведе відповідне попередження (рис. 4.4.)

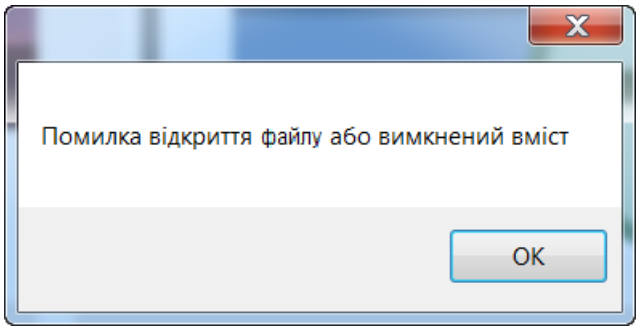


Рисунок 4.4 — Сповіщення про некоректно заповнені поля

4.2 Вилучення даних

Відкрийте виконуваний файл програми і в діалоговому вікні «Робота з

зображеннями» в полі зображення з вбудованим текстом завантажте файл у форматах .png або .bmp для подальшої обробки.

Далі користувач повинен встановити основні параметри для вставки даних в зображення, тобто вибрати компонент для вставки, визначити параметр P (різниця компонентів), розмір блоків розподілу зображення та ввести ключ.

Після виконання всіх дій необхідно натиснути кнопку «Надіслати повідомлення» (рис. 4.5).

Після підтвердження цієї дії в правій частині діалогового вікна в полі «Інформація» користувачеві відображається покрокова послідовність виконуваних подій (початок процесу вилучення, сегментація забруднення, виконання дискретного -косинусне перетворення, генерація повідомлень, сповіщення про закінчення досягнення введення зображення та виведення тексту, прихованого в зображенні (рис. 4.6).

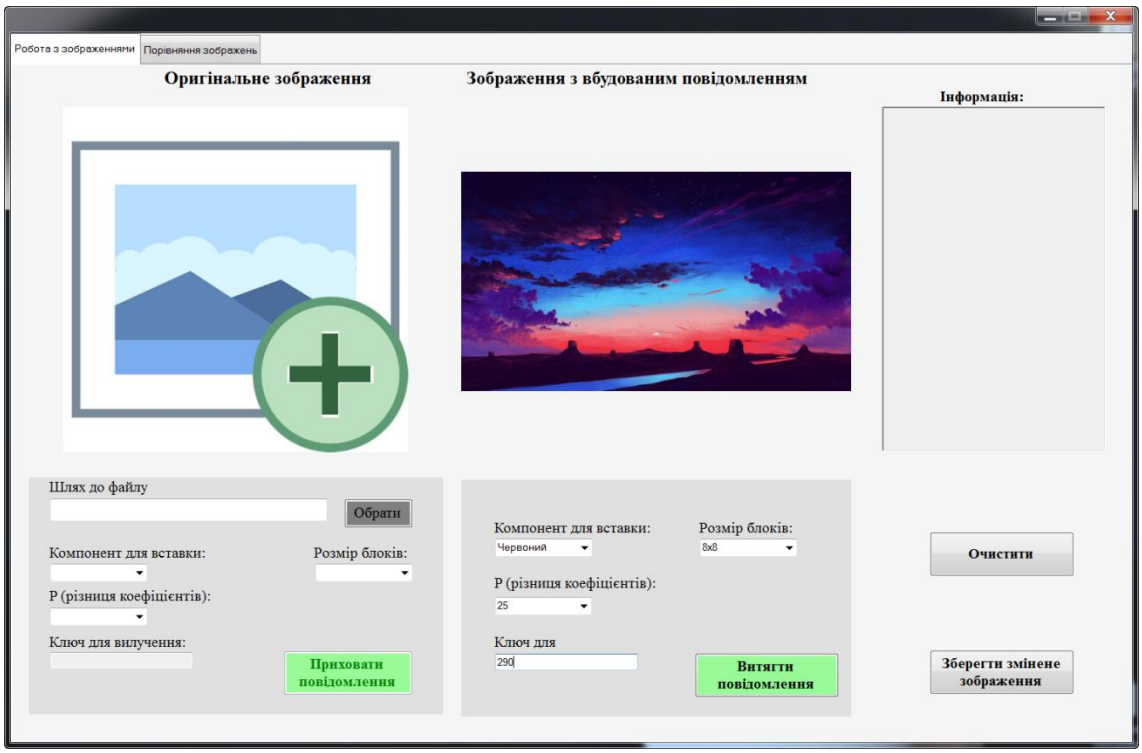


Рисунок 4.5 — Вигляд вікна додатку при вилученні даних

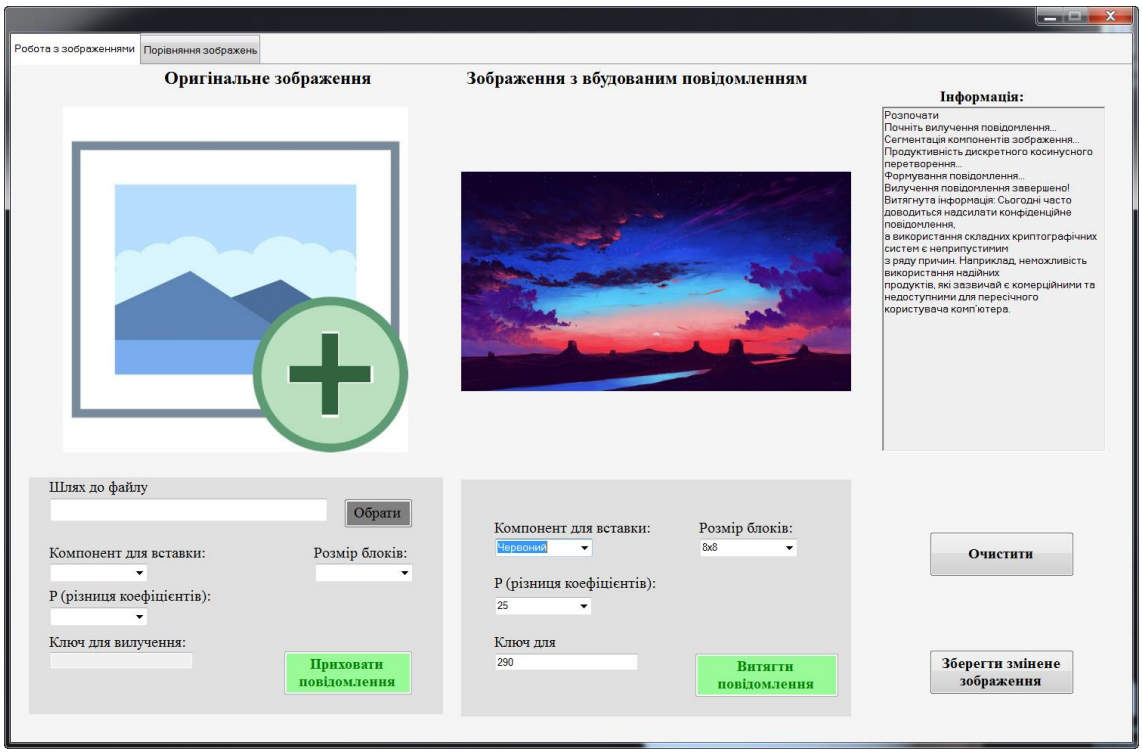


Рисунок 4.6 — Вигляд вікна додатку при вилученні даних

Наступний крок — порівняння зображень. Для завершення процесу перевірки змін, внесених в зображення, користувач повинен завантажити зображення в оригінальному вигляді і з повідомленням, введеним у відповідні поля на вкладці вікна програми «Порівняти зображення».

Зображення №1 — це поле для додавання вихідного зображення. Зображення №2 — це поле для додавання зображення з прихованою текстовою інформацією.

Поле даних також розміщено в правій частині вікна. Для початку процесу розпізнавання користувачеві необхідно натиснути функціональну кнопку «Порівняння зображень» (рис. 4.7).

Після завершення процесу порівняння зображень у полі «Зображення №3» відображається графік, що показує поле введення даних і те, як змінився вміст контейнера (зображення) після введення даних (рис. 4.8).

Слід зазначити, що при введенні коротких текстових повідомлень результат порівняння буде дуже маленьким, оскільки зображення зміниться лише найменшого розміру.

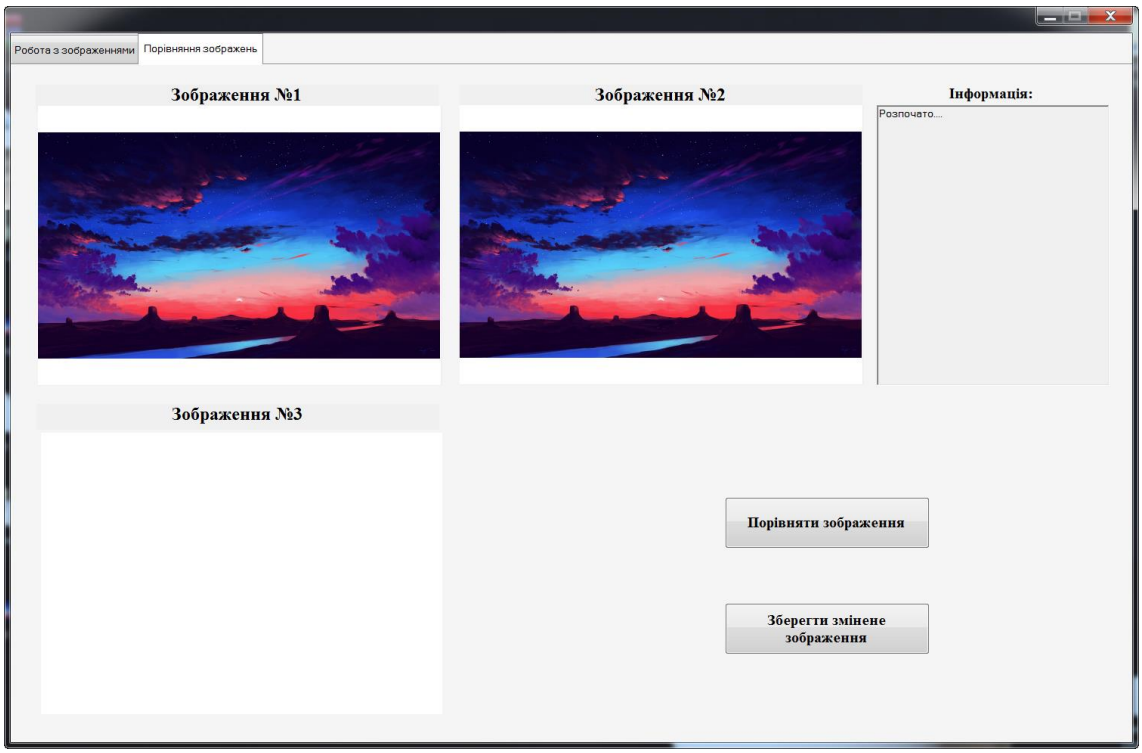


Рисунок 4.7 — Вигляд вікна при порівнянні зображень

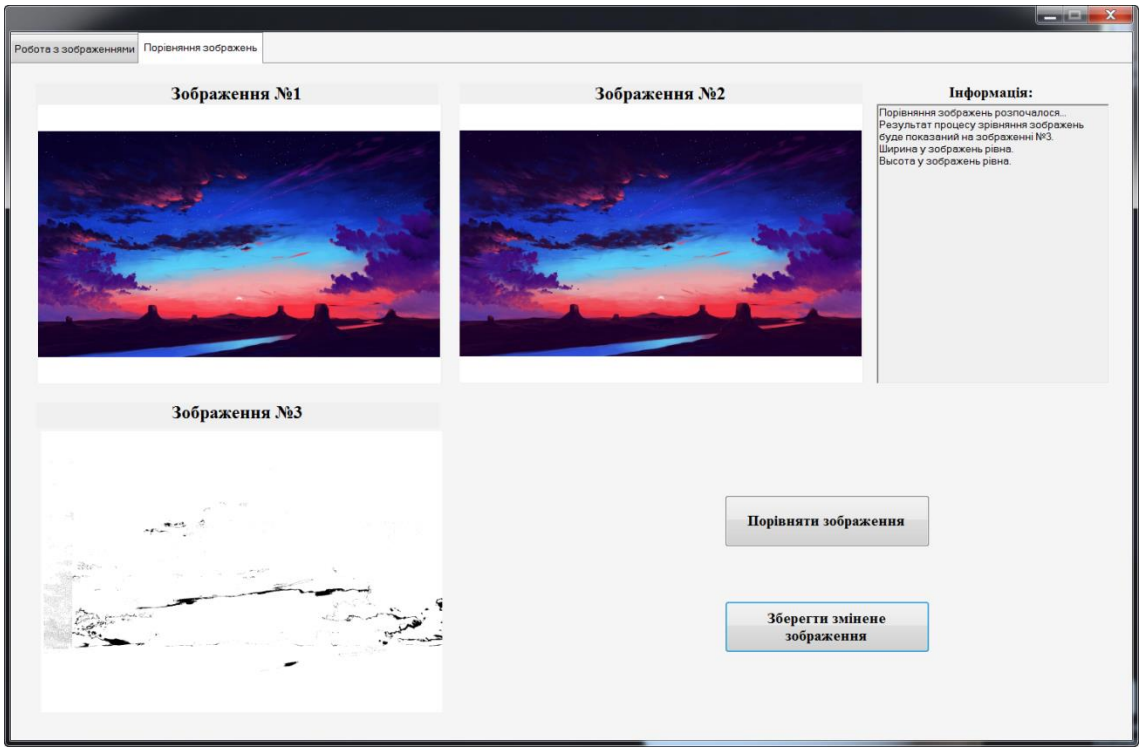


Рисунок 4.8 — Вигляд вікна при порівнянні зображень

Якщо подивитися на результати роботи програми і порівняти зображення до і після введення даних, можна зробити висновок, що зміни в зображенні, як

правило, непомітні для людського ока, навіть при введенні великих текстових повідомлень. Такий результат обумовлений доцільністю обраного методу і правильністю запиту.

4.3 Дослідження технічних показників

Далі розглянемо зміни з зображенням за допомогою дослідження технічного показника: співвідношення сигнал / рівень шуму, щоб зрозуміти, як змінилася структура контейнера.

Отже, визначити відношення сигнал/шум, тобто відношення потужності корисного сигналу до потужності шуму, можна за формулою:

$$SNR = \frac{P_{signal}}{P_{noise}}$$

де P — середня потужність.

Коефіцієнти сигналу та шуму вимірюються в тій самій еквівалентній точці системи в межах однієї смуги пропускання системи.

Це відношення (SNR) також можна розрахувати як квадрат відношення амплітуд:

$$SNR = \frac{P_{signal}}{P_{noise}} = \left(\frac{A_{signal}}{A_{noise}}\right)^2$$

де A — середньоквадратичне значення амплітуди.

Отже, розглянемо п’ять кольорових зображень у форматі .png для розрахунку кількісного результату за наведеними формулами (табл. 4.1).

На підставі отриманих результатів можна зробити висновок, що збільшення рівня шуму для обраного типу зображення коливається від 1 до майже 2%, що є цілком прийнятним при введенні великих даних.

Що стосується технічних системних вимог до розробленого додатку, то на практиці програмний засіб запускався на комп’ютері з операційними

системами Windows 7, 8.1 та 10, де вихідний код розробки показав належну продуктивність.

Таблиця 4.1 — Зміна рівня шуму

Рисунок	До вбудовування	Після вбудовування	Рівень шуму
Рисунок 1	25 dbi	28 dbi	1,3%
Рисунок 2	40 dbi	46 dbi	1,5%
Рисунок 3	35 dbi	38 dbi	1%
Рисунок 4	38 dbi	41 dbi	0,8%
Рисунок 5	29 dbi	33 dbi	1,2%

Спостереження за станом процесора комп'ютера під час роботи програмного забезпечення показало, що ця програма не потребує великої кількості ресурсів для роботи з нею.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку програмного засобу для вбудування повідомлення у фотографії із збереженням структури контейнера. Метою даного дослідження є підвищення інформаційної безпеки шляхом вбудовування повідомлень у зображення зі збереженням структури контейнера. Це досягається за допомогою використання технології збереження структури контейнера.

Аналогом може бути розробки можуть бути Steganos Privacy Suite - 2500 грн.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п’ятибальної системи оцінювання за 12-ма критеріями, у відповідності із табл. 5.1.

Таблиця 5.1 — Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздат-ність продук-кту в реаль-них умовах
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку

Продовження табл. 5.1

3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
Ринкові переваги					
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною .	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практик на здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так із комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

Продовження табл. 4.1

11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Усі дані по кожному параметру занесено в таблиці 5.2

Таблиця 5.2 — Результати оцінювання комерційного потенціалу

Критерії оцінювання	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	3	4	4
Наявність аналогів на ринку	3	3	4
Цінова політика	2	4	4
Технічні та споживчі властивості виробу	4	3	4
Експлуатаційні витрати	4	4	3
Ринок збуту	4	3	4
Конкурентоспроможність	3	4	3
Фахівці з технічної і комерційної реалізації	2	3	4
Фінансування	4	4	3
Матеріально-технічна база	3	3	3
Термін реалізації ідеї	4	4	2
Супровідна документація	4	3	3
Сума	40	42	41
Середньоарифметична сума балів	(40+42+41) / 3 = 41		

За даними таблиці 5.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 5.3.

Таблиця 5.3 — Рівні комерційного потенціалу розробки

Середньоарифметична сума балів, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0-10	Низький
11-20	Нижче середнього
21-30	Середній
31-40	Вище середнього
41-48	Високий

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є високим, що досягається за рахунок підвищення інформаційної безпеки шляхом вбудовування повідомлення у фотографії із збереженням структури контейнера.

5.2 Прогнозування витрат на виконання науково-дослідної роботи

5.2.1 Основна заробітна плата розробників, яка розраховується за формулою:

$$З_о = \frac{M}{T_р} \cdot t, \quad (5.1)$$

де М — місячний посадовий оклад конкретного розробника (дослідника), грн.;

$T_р$ — число робочих днів за місяць, 21 днів;

t — число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 5.4.

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту

Таблиця 5.4 – Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	26000	1238,10	32	39619,048
Програміст	24500	1166,67	32	37333,333
Всього				76952,38

5.2.2 Додаткова заробітна плата розробників, які брати участь в розробці обладнання/програмного продукту.

Додаткову заробітну плату прийнято розраховувати як 13 % від основної заробітної плати розробників та робітників:

$$З_д = З_о \cdot 13\ \% / 100\ \% \qquad (5.2)$$

$$З_д = (76952{,}38 \cdot 13\ \% / 100\ \%) = 10003{,}81 \ \text{(грн.)}$$

5.2.3 Нарахування на заробітну плату розробників.

Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$Н_з = (З_о + З_д) \cdot 22\ \% / 100\% \qquad (5.3)$$

$$Н_з = (76952{,}38 + 10003{,}81) \cdot 22\ \% / 100\ \% = 19130{,}36 \ \text{(грн.)}$$

5.2.4. Оскільки для розроблювального пристрою не потрібно витрачати матеріали та комплектуючи, то витрати на матеріали і комплектуючі дорівнюють нулю.

5.2.5 Амортизація обладнання, яке використовувалось для проведення розробки.

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді розраховується за формулою:

$$A = \frac{Ц}{Тв} \cdot \frac{t_{вик}}{12} \text{ [грн.].} \quad (5.4)$$

де Ц — балансова вартість обладнання, грн.;

Т — термін корисного використання обладнання згідно податкового законодавства, років

$t_{вик}$ — термін використання під час розробки, місяців.

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова вартість якого становить 34200 грн., термін його корисного використання згідно податкового законодавства — 2 роки, а термін його фактичного використання — 1,52 міс.

$$A_{обл} = \frac{34200}{2} \times \frac{1{,}52}{12} = 2171{,}429 \; грн.$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 5.5. Так як вартість ліцензійної ОС та спеціалізованих ліцензійних нематеріальних ресурсів менше 20000 грн, то даний нематеріальний актив не амортизується, а його вартість включається у вартість розробки повністю, $B_{нем.ак.}$ = 5400 грн (Microsoft Windows 10 — 5400 грн.).

Тарифи на електроенергію для непобутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас).

Таблиця 5.5 — Амортизаційні відрахування на матеріальні та нематеріальні ресурси для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортиза-ційні відрахуван-ня, грн.
Комп'ютер та комп'ютерна периферія	34200	2	1,52	2171,429
Офісне обладнання (меблі)	24000	4	1,52	761,905
Приміщення	980000	20	1,52	6222,222
Всього				9155,56

Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$В_е = В \cdot П \cdot Ф \cdot К_п, \quad (5.5)$$

де В — вартість 1 кВт-години електроенергії для 1 класу підприємства, В = 6,2 грн./кВт;

П — встановлена потужність обладнання, кВт. П = 0,4 кВт;

Ф — фактична кількість годин роботи обладнання, годин.

$К_п$ — коефіцієнт використання потужності, $К_п$ = 0,9.

$$В_е = 0,9 \cdot 0,4 \cdot 8 \cdot 32 \cdot 6,2 = 571,392 \text{ (грн.)}$$

5.2.6 Інші витрати та загальновиробничі витрати.

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за

статтею «Інші витрати» розраховуються як 50…100% від суми основної заробітної плати дослідників:

$$I_в = (З_о + З_р) \cdot \frac{Н_{ів}}{100\%}, \quad (5.6)$$

де $Н_{ів}$ — норма нарахування за статтею «Інші витрати».

$$I_в = 76952,38 * 90\% / 100\% = 69257,14 \text{ (грн.)}$$

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100…150% від суми основної заробітної плати дослідників:

$$Н_{нзв} = (З_о + З_р) \cdot \frac{Н_{нзв}}{100\%}, \quad (5.7)$$

де $Н_{нзв}$ — норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$Н_{нзв} = 76952,38 * 135 \% / 100 \% = 103886 \text{ (грн.)}$$

5.2.7 Витрати на проведення науково-дослідної роботи.

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$B_{заг} = 76952,38+10003,81+19130,36+9155,56+5400+571,39+69257,14+$$
$$+103886 = 294356,36 \text{ грн.}$$

5.2.8 Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів.

Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{B_{заг}}{\eta} \text{ (грн),} \qquad (5.8)$$

де η — коефіцієнт, який характеризує етап (стадію) виконання науково дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то η=0,1; технічного проектування, то η=0,2; розробки конструкторської документації, то η=0,3; розробки технологій, то η=0,4; розробки дослідного зразка, то η=0,5; розробки промислового зразка, то η=0,7; впровадження, то η=0,9. Оберемо η = 0,5, так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$ЗВ = 294356,36 / 0,5 = 588713 \text{ грн.}$$

5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових

коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

— вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;

— зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);

— кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;

— визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

—абсолютного економічного ефекту (чистого дисконтованого доходу);

—внутрішньої економічної дохідності (внутрішньої норми дохідності);

—терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

5.3.1 Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta П_i = (\pm \Delta Ц_0 \cdot N + Ц_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \quad (5.9)$$

де $\pm \Delta Ц_o$ — зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N — кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

$Ц_o$ — основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $Ц_o = Ц_б \pm \Delta Ц_o$;

Цб — вартість програмного продукту у році до впровадження результатів розробки;

ΔN — збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

λ — коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0{,}8333$.

p — коефіцієнт, який враховує рентабельність продукту;

ϑ — ставка податку на прибуток, у 2023 році ϑ =18%.

Припустимо, що при прогнозованій ціні 1500 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 100 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом

першого року – на 6000 шт., протягом другого року – на 7500 шт., протягом третього року на 9000 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$\Delta П_1$ = (0*100 + (1500 + 100)*6000)* 0,8333* 0,32) * (1 - 0,18) = 1967999,921 грн.

$\Delta П_2$ = (0*100 + (1500 + 100)*(6000+7500)* 0,8333* 0,32) * (1 - 0,18) = 4723199,811 грн.

$\Delta П_3$ = (0*100 + (1500 + 100)*(6000+7500+9000)* 0,8333* 0,32) * (1 - 0,18) = 7871999,685 грн.

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 14563199,42 грн.

5.3.2 Розрахунок ефективності вкладених інвестицій та періоду їх окупності.

Розраховуємо приведену вартість збільшення всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{1}^{T} \frac{\Delta П_i}{(1+\tau)^t}, \qquad (5.10)$$

де $\Delta П$ — збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

Т — період часу, протягом якою виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

τ — ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05...0,15$;

t — період часу (в роках).

Збільшення прибутку ми отримаємо, починаючи з першого року:

$$ПП = (1967999,921/(1+0,1)^1)+(4723199,811/(1+0,1)^2)+(7871999,685/$$
$$/(1+0,1)^3) = 1789090,84 + 3903470,918 + 5914349,876 = 11606911,63 \text{ грн.}$$

Далі розраховують величину початкових інвестицій *PV*, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{інв} * ЗВ, \quad (5.11)$$

де $k_{інв}$ — коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію,це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{інв}=2...5$, але може бути і більшим;

ЗВ — загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 588713 = 1177425,43 \text{ грн.}$$

Тоді абсолютний економічний ефект $E_{абс}$ або чистий приведений дохід *(NPV, Net Present Value)* для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV, \quad (5.12)$$

$$E_{абс} = 11606911,63 - 1177425,43 = 10429486,20 \text{ грн.}$$

Оскільки $E_{абс} > 0$ то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої но-рми дохідності (IRR, Internal Rate of Return) вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $E_в$. Для цього використаємо формулу:

$$E_в = \sqrt[T_ж]{1 + \frac{E_{абс}}{PV}} - 1, \tag{5.13}$$

де $T_ж$ — життєвий цикл наукової розробки, роки.

$$E_в = \sqrt[3]{(1 + 10429486{,}20/1177425{,}43} - 1 = 1{,}144$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \tag{5.14}$$

де d $T_ж$ — середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні d = (0,09…0,14);

f — показник, що характеризує ризикованість вкладень; зазвичай, величина f = (0,05...0,5).

$$\tau_{\min} = 0{,}14 + 0{,}05 = 0{,}19.$$

Так як $E_в > \tau_{\min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_в}, \tag{5.15}$$

$$T_{ок} = 1 / 1{,}144 = 0{,}87 \text{ р.}$$

Оскільки $T_{ок}$ < 3-х років, а саме термін окупності рівний 0,87 роки, то фінансування даної наукової розробки є доцільним.

Економічна частина даної роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 588713 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є висококонкурентоспроможним. Період окупності складе близько 0,87 роки.

ВИСНОВКИ

У цій роботі проаналізовано методи розміщення даних у зображенні та розроблено програмний засіб для підвищення безпеки даних у зображенні та збереження структури контейнера.

Проведено аналіз варіантів способів включення інформації в зображення. Враховуючи результати попереднього аналізу можливих алгоритмів введення даних в зображення, в даній роботі використано метод розміщення блоків у просторовій області контейнера, розроблений В. Дармстедтером, Ж.-Ф. Делайглем. , Quisquater, було обрано для використання) та Makk (B. Macq. Розроблено програмний засіб, що включає обробку основних компонентів: процесор запитів – компонент, який забезпечує передачу інформації між програмою та користувачем для подальшої обробки. ; стегокодер — це компонент, який відповідає за вбудовування даних у зображення; стегодетектор — це компонент, який використовується для виявлення вбудованих даних під час обробки файлу зображення.

Для практичної реалізації поставленої мети роботи та розробки її програмного забезпечення в середовищі програмування Microsoft Visual Studio на базі Windows Form використовувалася мова програмування C#.

На практиці перевірено роботу розробленого програмного засобу для вставки даних в зображення зі збереженням структури контейнера. За результатами роботи програми та порівняння зображення до і після введення даних можна зробити висновок, що зміни, внесені в зображення, зазвичай непомітні для людського ока, навіть при розміщенні великих текстових повідомлень. Такий результат обумовлений доцільністю обраного методу і правильністю запиту.

Зміни в зображенні перевірялися за допомогою технічного індикатора:

співвідношення рівня сигнал/шум. Виходячи з отриманих результатів, можна припустити, що підвищення рівня шуму для обраного типу зображення коливається від 1 до майже 2%, що є прийнятним при введенні великих даних.

З наведених вище результатів було підтверджено, що початкова мета роботи, тобто розробка програмного засобу, який усуває недоліки розглянутих аналогів, була досягнута.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Heatherly R., Kantarcioglu M., Thuraisingham B. Preventing private information inference attacks on social networks. IEEE Transactions on Knowledge and Data Engineering. 2013. 25(8). P. 1849–1862.

2. Development of methodology for modeling the interaction of antagonistic agents in cybersecurity systems / O. Milov et al. Восточно-Европейский журнал передовых технологий. 2019. 2(9). С. 56–66.

3. Development of the model of the antagonistic agents behavior under a cyber conflict / O. Milov et al. Восточно-Европейский журнал передовых технологий. 2019. 4(9). С. 6–19.

4. Кошкина Н.В. Стеганоаналіз цифрових зображень із застосуванням контрольного вкраплення // Матеріали з Міжнар. наук.-техн. конф. «Захист інформації і безпека інформаційних систем», 5–6 черв. 2014. – Львів: Львівська політехніка, 2014. –С. 98–100.

5. Karampidis K., Kavallieratou E., Papadourakis G. A review of image steganalysis techniques for digital forensics. Journal of Information Security and Applications 2018. 40. P. 217–235.

6. Задирака В.К., Кудин А.М. Новые модели и методы определения стой кости систем защиты информации. Кибернетика и системный анализ. 2017. 53(6). С. 176–184.

7. Бобок І.І. Підвищення інформативності результатів виявлення клонування в цифровому зображенні. Збірник наукових праць ВІКНУ імені Тараса Шевченка. 2017. 58. С. 81–90.

8. Ansari M.D., Ghrera S.P., Tyagi V. Pixel-based image forgery detection: A Review. IETE Journal of Education. 2014. 55(1). P. 40–46.

9 Zheng L., Zhang Y., Thing V.L.L. A survey on image tampering and its detection in real-world photos. Journal of Visual Communication and Image Representation. 2019. 58. P. 380–399.

10. Швідченко І.В. Аналіз програмного забезпечення зі стеганоаналізу.

Штучний інтелект. 2012. 3. С. 487–495.

11. Хорошко В. О. Основи комп’ютерної стеганографії: Навчальний посібник / В. О. Хорошко, М. Є. Шелест, О. Д. Азаров, Ю. Є. Яремчук. – Вінниця: ВДТУ, 2003 – 143 с.

12. Карпінець, В. В. Методи захисту векторних зображень цифровими водяними знаками : монографія / В. В. Карпінець, Ю. Є. Яремчук. – Вінниця: ВНТУ, 2013. – 156 с.

13. Вовк О.О. Методи підвищення стійкості та пропускної здатності систем прихованої передачі інформації / Вовк О.О. – Харків, 2016 – 177с.

14. В.Г. Бабенко, В.М. Зажома, О.Б. Нестеренко. Метод вбудовування стегоповідомлення на основі ключового елемента / В.Г. Бабенко, В.М. Зажома, О.Б. Нестеренко. // Захист інформації. 2014. С. 53-58.

15. Кузнецов О. О. Стеганографія : навчальний посібник / О.О.Кузнецов, С. П. Євсеєв, О. Г. Король. // – Х. : Вид. ХНЕУ, 2011. – 232с.

17. S-Tools URL: https:// stools.com.ua (дата звернення: 04.06.2022).

18. Steganos Privacy Suite URL: https://steganos_privacy.com.ua (дата звернення: 04.06.2022).

19. ImageSpyer URL: https:// imagespyer.com.ua (дата звернення: 04.06.2022).

20. JSTEG URL: https://jsteg.com.ua (дата звернення: 04.06.2022).

21. Gifshuffle URL: https://gifshuffle.com.ua (дата звернення: 04.06.2022).

22. Коханович Г. Ф. Компьютерная стеганография. Теория и практика / Г. Ф. Коханович, А. Ю. Пузиренко. – Київ: МК-Пресс, 2006. – 288 с.

23. Chaeikar S.S., Ahmadi A. Ensemble SW image steganalysis: A low dimension method for LSBR detection. Signal Processing: Image Communication. 2019. 70. P. 233–245.

24. Sairam T.D., Boopathybagan K. Computational intelligence-based steganalysis comparison for RCM-DWT and PVA-MOD methods. Automatika. 2019. 60(3). P. 285–293.

25. Wang P., Liu F., Yang C., Luo X. Steganalysis aided by fragile detection of image manipulations. Multimedia Tools and Applications. 2019. 78. P. 23309–

23328.

26. Корченко О.Г., Васіліу Є.В., Гнатюк С.О. Сучасні квантові технології захисту інформації // Науково-технічний журнал "Захист інформації". – 2010, № 1. – С. 77-89.

27. Bilal A. Shaw. Quantum steganography and quantum error-correction // University of Southern California. – 2010. – P.137.

28. Mogos G. Stego Quantum Algorithm // International Symposium on Computer Science and its Applications. – 2018. – P. 187-190

29. Visual Studio URL: https://docs.microsoft.com/ru-ru/visualstudio/get-started/visual-studio-ide?view=vs-2019 (дата звернення: 05.06.2022).

30. Everything you need to know about C# URL: https://www.pluralsight.com/ (дата звернення: 05.06.2022).

ДОДАТОК А

Технічне завдання

Міністерство освіти та науки України

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

_____проф., д.т.н. О. Д. Азаров

«___» _____ 20__ р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи

«Засіб для вбудування повідомлення у фотографії із збереженням структури контейнера»

08-54.МКР.026.00.000 ПЗ

Науковий керівник

д.т.н., проф. каф. ОТ

_____ Азаров О.Д.

виконав:

магістрант 2 курсу,

_____Гонца А.В.

Вінниця 2023

1 Підстава виконання магістерської кваліфікаційної роботи

1.1 Для захисту інформації, а точніше забезпечення власне факту захисту, в цифровій стеганографії використовуються контейнери – цифрові об’єкти, куди вбудовується інформація, що часто зумовлює структурні зміни в даних контейнерах.

1.2 Наказ про затвердження теми МКР

2 Мета і призначенням МКР

2.1 Метою роботи є підвищення захищеності даних шляхом вбудовування повідомлення у фотографії із збереженням структури контейнера.

2.2 Призначення розробки — виконання магістерської кваліфікаційної роботи.

3 Вихідні дані для виконання МКР

Вихідні дані для виконання МКР: симетричний алгоритм шифрування AES, мова об’єктно — орієнтованого програмування C# середовище програмування Visual Studio.

4 Вимоги до виконання МКР

МКР повинна задовольняти такі вимоги: розробити програмне забезпечення для приховування великих обсягів інформації у відомих графічних форматах, для подальшої їх передачі.

5 Етапи МКР та очікувані результати

Етапи роботи та очікувані результати приведено в табл. А.1.

6 Матеріали, що подаються до захисту МКР

До захисту МКР подаються: пояснювальна записка МКР, ілюстративні та графічні матеріали, протокол попереднього захисту МКР на кафедрі, відзив наукового керівника, відзив опонента, протоколи складання державних екзаменів, анотації до МКР українською та іноземною мовами, довідка про відповідність оформлення МКР діючим вимогам.

Таблиця А.1 — Етапи МКР

з/п	Назва етапів виконання магістерської роботи	Строк виконання етапів роботи	.
	Постановка мети та задач роботи	20.10.23	
	Аналіз стеганографічних методів, стеганоконтейнери	27.10-30.10.23	
	Розробка алгоритму роботи програмного додатку	01.11-09.11.23	
	Вибір інструментарію розробки	10.11-17.11.23	
	Налаштування середовища розробки	18.11-.22.1123	
	Розробка програмного засобу	23.11-26.11.23	
	Тестування якості роботи засобу	27.11-31.11.23	
	Дослідження засобу для вбудовування даних	01.12-04.12.23	
	Розрахунок економічної частини роботи	01.12-04.12.23	
	Оформлення пояснювальної записки та ілюстративного матеріалу	05.12.23	
	Аналіз виконання роботи, висновки, додатки		
	Перевірка якості виконання магістерської роботи та усунення недоліків		

7 Порядок контролю виконання та захисту МКР

Виконання етапів розрахункової та графічної документації МКР контролюється науковим керівником згідно зі встановленими термінами.

Захист МКР відбувається на засіданні Державної екзаменаційної комісії, затвердженою наказом ректора.

8 Вимоги до оформлення МКР

8.1 При оформлюванні МКР використовуються:

— ДСТУ 3008: 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;

— ДСТУ 8302: 2015 «Бібліографічні посилання. Загальні положення та правила складання»;

— Методичні вказівки до виконання магістерських кваліфікаційних робіт зі спеціальності 123 — «Комп'ютерна інженерія». Кафедра обчислювальної техніки ВНТУ 2022.

8.2 Порядок виконання МКР викладено в «Положення про кваліфікаційні роботи на другому (магістерському) рівні вищої освіти СУЯ ВНТУ–03.02.02 П.001.01:21.

ДОДАТОК Б

Структурна схема стеганосистеми

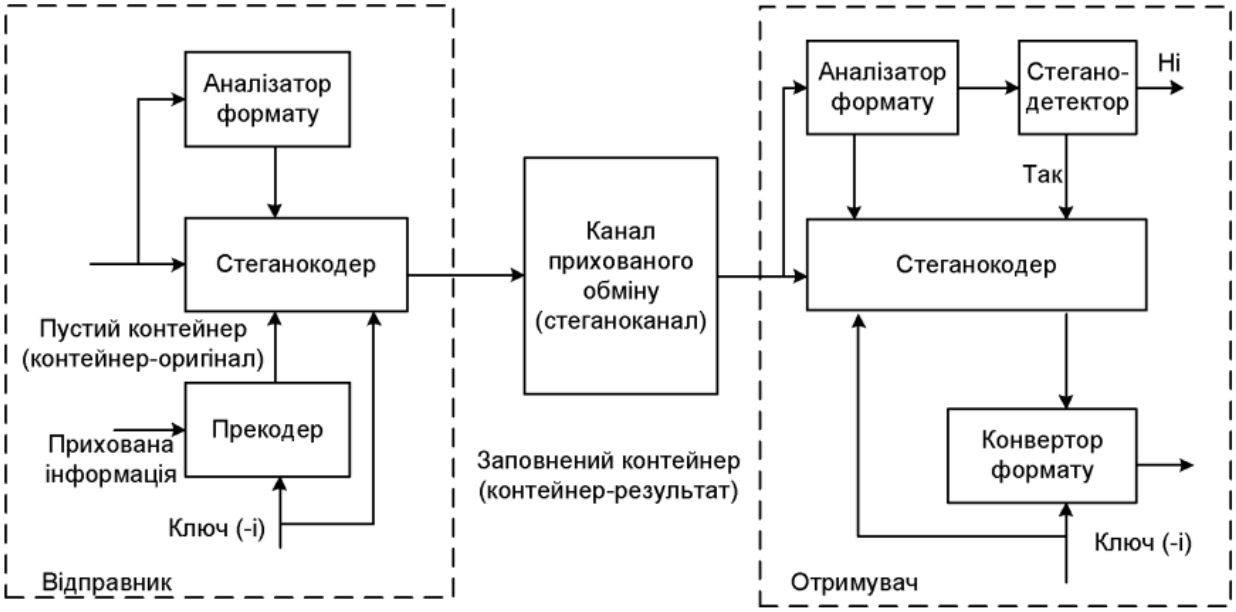


Рисунок Б.1 — Структурна схема стеганосистеми

ДОДАТОК В

Основні компоненти розроблюваного програмного засобу

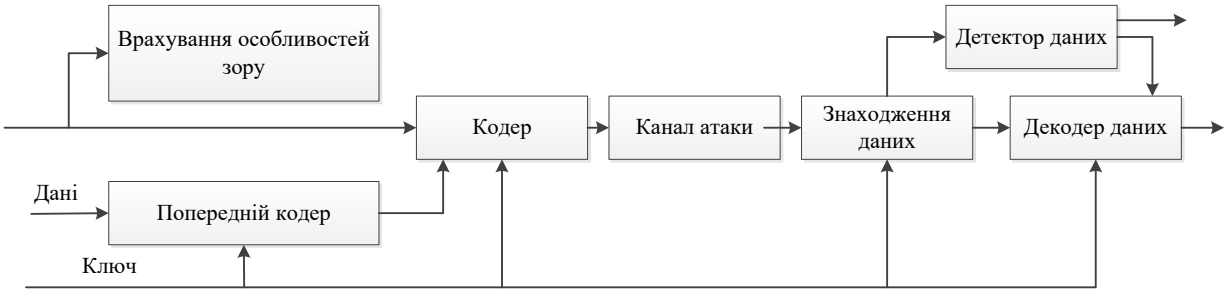


Рисунок В.1 — Основні компоненти розроблювального програмного засобу

ДОДАТОК Г

Алгоритм роботи розроблюваного додатку для приховування даних

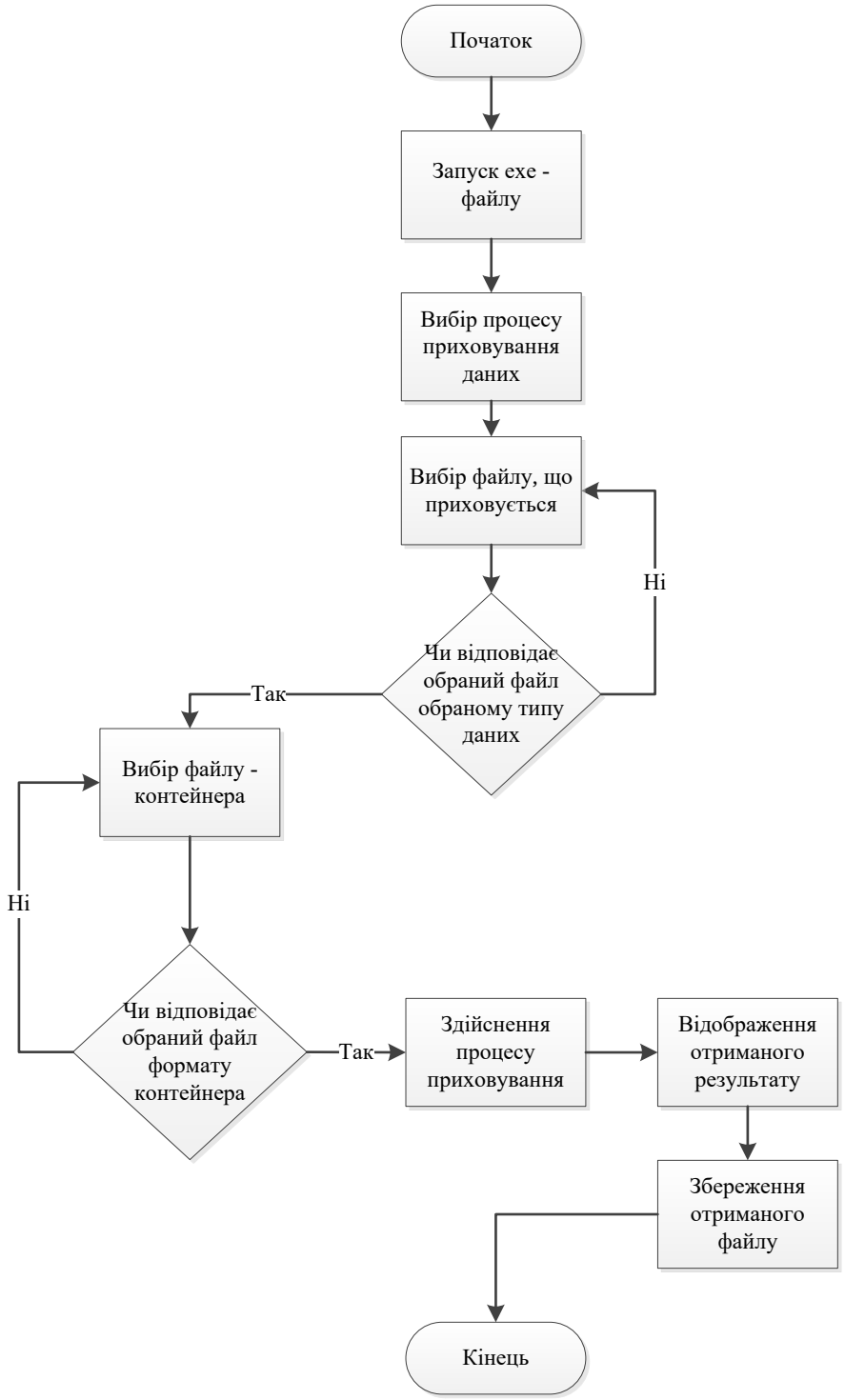


Рисунок Г.1 — Алгоритм роботи розроблюваного додатку для приховування даних

ДОДАТОК Д

Алгоритм роботи розроблюваного додатку для вилучення даних

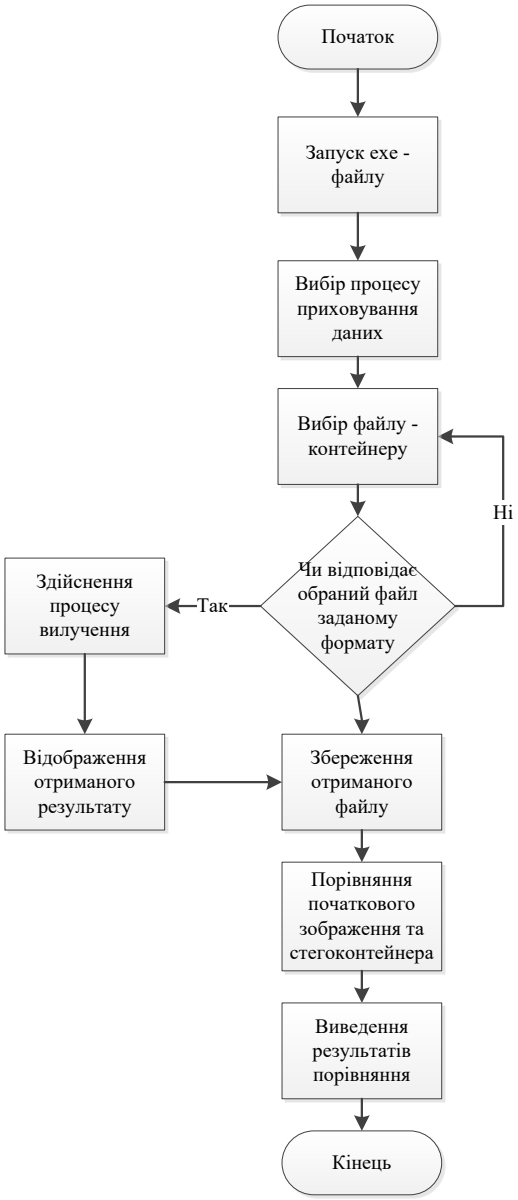


Рисунок Д.1 — Алгоритм роботи розроблюваного додатку для вилучення даних

ДОДАТОК Е

UML – діаграма класів розробки

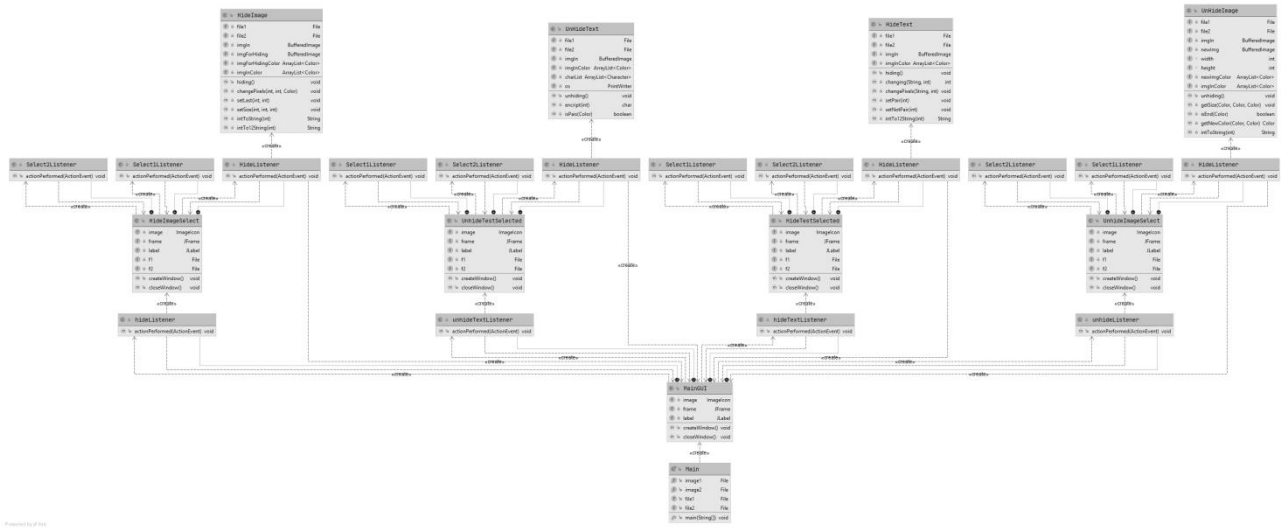


Рисунок Е.1 — Алгоритм роботи розроблюваного додатку для вилучення даних

ДОДАТОК Ж

ПРОТОКОЛ

ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Засіб для вбудування повідомлення у фотографії із збереженням структури контейнера

Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ кафедра обчислювальної техніки
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 91,5% Схожість 18,5%

Аналіз звіту подібності (відмітити потрібне):

✓ Запозичення, виявлення у роботі, оформлені коректно і не містять ознак плагіату.

□ 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.

□ 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 _____ Захарченко С.М.
(підпис) (прізвище, ініціали)

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

Автор роботи _____ Гонца А.В
(підпис) (прізвище, ініціали)

Керівник роботи _____ Азаров О.Д.
(підпис) (прізвище, ініціали)