

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
« ____ » _____ 2026 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
“Програмне забезпечення системи приховування інформації на
базі використання генеративно-змагальних мереж”

Виконав здобувач вищої освіти
IV курсу, групи КН-22
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Теліга А.В.
« ____ » _____ 2026 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Коваленко А.С.
« ____ » _____ 2026 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь бакалавр
Галузь знань . 12 “Інформаційні технології”
Спеціальність 122 “Комп’ютерні науки”
Освітньо-професійна (освітньо-наукова) програма “Комп’ютерні науки”

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 15 » січня 2026 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Телізі Антону Володимировичу

(прізвище, ім’я, по батькові)

1. Тема роботи *Програмне забезпечення системи приховування інформації на базі використання генеративно-змагальних мереж*

2. Керівник роботи *Коваленко Анна Степанівна, канд. техн. наук, доцент*

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 116-02 від 06.02.2026 року

3. Строк подання студентом роботи до захисту *23.05.2026 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою роботи є розробка програмного забезпечення системи приховування інформації на базі використання генеративно-змагальних мереж*

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.

2. Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію.

6. Висновки

6. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

Структурна схема системи *1 аркуш*

Функціональна схема системи *1 аркуш*

Діаграма процесів *1 аркуш*

Блок-схема алгоритму роботи додатку *2 аркуша*

7. Дата видачі завдання « 15 » січня 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.03.2026 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2026 р.	
3.	Розробка моделі компонента	20.03.2026 р.	
4.	Розробка структур даних	25.03.2026 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2026 р.	
6.	Програмування алгоритмів	10.04.2026 р.	
7.	Оформлення ПЗ	17.04.2026 р.	
8.	Попередній захист роботи	23.05.2026 р.	

Дата видачі завдання
« 15 » січня 2026 р.

Підпис керівника

Коваленко А.С.
(прізвище та ініціали)

Завдання прийнято до виконання
« 15 » січня 2026 р.

Підпис здобувача

Теліга А.В.
(прізвище та ініціали)

АНОТАЦІЯ

Теліга А.В. Програмне забезпечення системи приховування інформації на базі використання генеративно-змагальних мереж. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи приховування інформації на базі використання генеративно-змагальних мереж.

Метою розробки є програмне забезпечення системи приховування інформації на базі використання генеративно-змагальних мереж.

Результат роботи – програмна реалізація системи приховування інформації на базі використання генеративно-змагальних мереж.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі Python.

Ключові слова: комп'ютерні науки, приховування інформації, генеративно-змагальні мережі

ABSTRACT

Teliha A.V. Software for an information hiding system based on the use of generative adversarial networks. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

In this final qualification work for the first (bachelor's) level of higher education, software has been developed, which is intended for an information hiding system based on the use of generative adversarial networks.

The purpose of the development is software for an information hiding system based on the use of generative adversarial networks.

The result of the work is a software implementation of an information hiding system based on the use of generative adversarial networks.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software are provided.

The program can be used on a PC with Windows 10/11.

The program is developed in Python.

Keywords: computer science, information hiding, generative adversarial networks

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	5
1.1 Призначення системи.....	5
1.2 Область застосування.....	6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	7
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	7
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	15
2.3 Розгорнута постановка завдання	17
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	19
3.1 Опис функціонування системи	19
3.2 Розробка структурної схеми.....	22
3.3 Розробка функціональної схеми	28
3.4 Розробка діаграми процесів.....	31
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	33
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	33
4.2 Захист розробленого програмного забезпечення.....	47
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	49
6 ОСНОВНІ ВИСНОВКИ.....	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	53

					ВКРБ-122.26.0030.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	Програмне забезпечення системи приховування інформації на базі використання генеративно- змагальних мереж	Літ.	Аркуш	Аркушів
Розроб.	Теліга А.В.					Б	1	59
Перев.	Коваленко А.С.							
Н.контр.	Коваленко А.С.					ЦНТУ КН-22		
Затв.	Смірнов О.А.							

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ВДТ	–	відео-дисплейні термінали
ЕОМ	–	електронно-обчислювальна машина
ЕПТ	–	електроннопроменева трубка
ЕЦП	–	електронний цифровий підпис
ЗІ	–	захист інформації
ПЗ	–	програмне забезпечення
ПК	–	персональний комп'ютер
СБ	–	служба безпеки
ТЗ	–	технічне завдання
ЦВЗ	–	цифрові водяні знаки
DES	–	стандарт шифрування США
DSA	–	Digital Signature Algorithm
ECDSA	–	Elliptic Curve Digital Signature Algorithm
EGSA	–	El Gamal Signature Algorithm
IDEA	–	International Data Encryption Algorithm – алгоритм шифрування
IP	–	Internet Protocol
LSB	–	Least Significant Bits – метод стеганографії
PGP	–	Pretty Good Privacy – міжнародний криптографічний стандарт
RSA	–	алгоритм асиметричного шифрування
SHA-1	–	Secure Hash Algorithm 1 – алгоритм криптографічного хешування
TDES	–	Triple DES – модифікація DES з трьома незалежними підключами
JPEG	–	Joint Photographic Experts Group – растровий формат зображення

ВСТУП

Актуальність теми. Стеганографія широко використовується в галузі захисту конфіденційності зображень. Однак, з розвитком методів стеганалізу, моделі на основі глибокого навчання тепер здатні точно виявляти зміни в стего-зображеннях, що створює значну загрозу для традиційної стеганографії. Щоб вирішити цю проблему, ми пропонуємо підхід на основі GAN для посилення змагальної стеганографії зображень. Цей метод використовує кодер як генератор разом з дискримінатором для формування генеративно-змагальної мережі (GAN), тим самим підвищуючи стійкість стегозображень до інструментів стеганалізу. Крім того, фреймворк GAN зменшує розрив між оригінальним секретним зображенням та вилученим зображенням, тоді як декодер ефективно витягує секретне зображення зі стего-зображення, досягаючи мети захисту конфіденційності зображень. Експериментальні результати показують, що метод не тільки забезпечує високоякісні секретні зображення, але й ефективно знижує точність класифікаторів нейронних мереж, що призводить до неправильних класифікацій та значно збільшує можливості вбудовування стеганографічної системи. Наприклад, під час PGD-атаки, збройні стегозображення, згенеровані GAN, за вищих рівнів збурень, успішно підтримують якість секретного зображення, досягаючи при цьому рівня неправильної класифікації 84,73% при виявленні за допомогою нейронної мережі. Порівняно із зображеннями з такою ж візуальною якістю, наш метод збільшив рівень неправильної класифікації на 23,31%.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи приховування інформації на базі використання генеративно-змагальних мереж.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

- Огляд існуючих систем приховування інформації на базі використання генеративно-змагальних мереж.
- Дослідження системи приховування інформації на базі використання генеративно-змагальних мереж.
- Програмна реалізація системи приховування інформації на базі використання генеративно-змагальних мереж.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі приховування інформації на базі використання генеративно-змагальних мереж.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи приховування інформації на базі використання генеративно-змагальних мереж, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Узагальнена модель стегосистеми може бути представлена як проблема ув'язненого із трьома учасниками. Аліса й Боб утримуються в окремих камерах. Звичайно ж, їм дозволено спілкуватися один з одним, однак усі їх комунікації контролюються наглядачем Венді. У сучасній стеганографії кожний канал між Алісою й Бобом містить п'ять елементів: контейнер, алгоритм впровадження / видобування повідомлення, ключі, секретне повідомлення й канал зв'язку.

Контейнер – це будь-яка інформація, призначена для передачі таємних повідомлень. Стегоключ або просто ключ – секретний ключ, необхідний для приховання інформації. Як і в криптографії ключ може бути секретним або відкритим.

Основна вимога полягає в тому, що стегозображення повинне бути невідрізняємо від вихідного зображення або набору вихідних зображень.

Безпека у свою чергу залежить від методів доступних злоумисникові. Залежно від роботи Венді, вона може бути активною або пасивною наглядачкою. Якщо Венді тільки перевіряє, є чи стегозображення звичайним або «модифікованим» при передачі по каналу, то її називають пасивним наглядачем. Якщо Венді може витягати уроки з попередніх атак і нагромадити знання про стегосистему, то вона буде активним доглядачем. У такому випадку вона могла б спробувати небагато змінити передані об'єкти або виявити наявність схованого зв'язку.

Багато статей по стеганографії присвячені пасивному режиму наглядача. Але, як показує практика, Венді звичайно виконує як активні, так і пасивні обов'язки. Згідно із принципом Керкгоффса, систем безпеки Венді повністю знає алгоритм, який можуть використовувати Аліса й Боб.

У випадку активного наглядача безпека звичайно зв'язується зі складністю видобування повідомлень. У випадку пасивної атаки ключовою проблемою

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

безпеки стеганографії є тонка грань, по якій складно відрізнити вихідне зображення й стегозображення. Це поняття містить у собі малопомітні як для людського зору, так і для машинного статистичного аналізу відмінності.

1.2 Область застосування

Більшість існуючих стеганографічних методів в основному розроблені для зображень у градаціях сірого. При безпосередньому застосуванні до кольорових зображень без врахування міжканальних кольорових взаємодій їхня безпека може бути значно скомпрометована. У даній роботі ми пропонуємо новий метод стеганографії кольорових зображень, заснований на генеративно-змагальних мережах (GAN). Наша структура включає генератор з двома гілками та дискримінатор, оснащений кількома стеганалітичними мережами, кожна з яких зосереджена на певному кольоровому каналі. Ця архітектура дозволяє поступово навчатися асиметричним витратам вбудовування в канали з нуля. Ми також представляємо поетапну стратегію навчання, яка сприяє навчанню міжканальних кольорових взаємодій та оптимізує безпеку кожного кольорового каналу на різних фазах, покращуючи загальну безпеку. Крім того, ми пропонуємо нову адаптивну стратегію оновлення та вводимо функцію втрат середнього абсолютного відхилення (MAD) для підтримки динамічного балансу між генератором та дискримінатором, тим самим поступово покращуючи стеганографічну продуктивність генератора під час навчання. Широкі порівняльні експерименти демонструють, що запропонований метод досягає найсучаснішого рівня безпеки від стеганоаналізу кольорових зображень. Комплексні дослідження абляції додатково підтверджують ефективність та обґрунтованість нашого підходу.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи приховування інформації на базі використання генеративно-змагальних мереж, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти

QuickStego

QuickStego дозволяє приховувати текст у знімках, щоб тільки інші користувачі QuickStego могли витягати й читати сховані секретні повідомлення.

Після того, як текст буде схований у зображенні, збережене зображення як і раніше є «зображенням», воно буде завантажуватися так само, як і будь-яке інше зображення. Зображення можна зберегти, відправити по електронній пошті, завантажити в Інтернеті, як і колись, єдина відмінність полягає в тому, що воно містить схований текст.



Рисунок 2.1 – Інтерфейс користувача QuickStego

Xiao Steganography

Інструмент використовується для приховання текстового повідомлення усередині зображення. Ви можете також сховати зображення усередині іншого зображення.



Рисунок 2.2 – Інтерфейс користувача Xiao Steganography

Openstego

Використовуючи це програмне забезпечення, ви можете або сховати дані (файл) усередині зображення, або витягти дані із зображення.

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

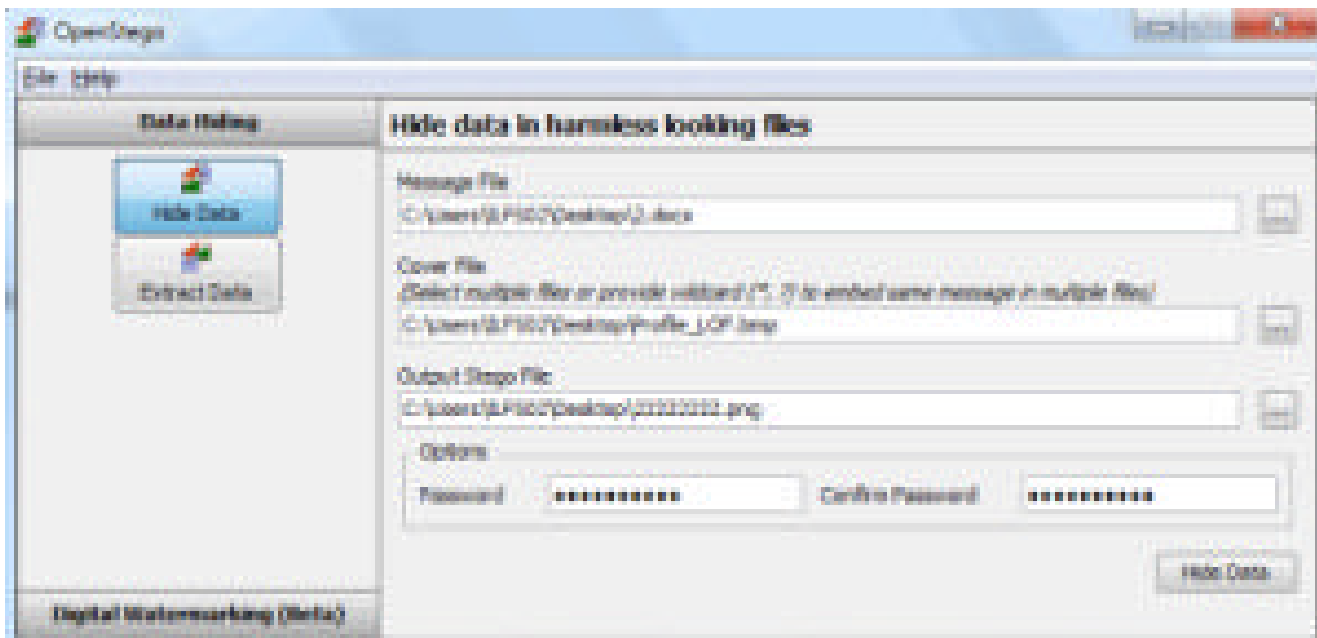


Рисунок 2.3 – Інтерфейс користувача Openstego

Camouflage

Ви можете сховати секретний TXT-файл у стандартному JPG-зображенні.

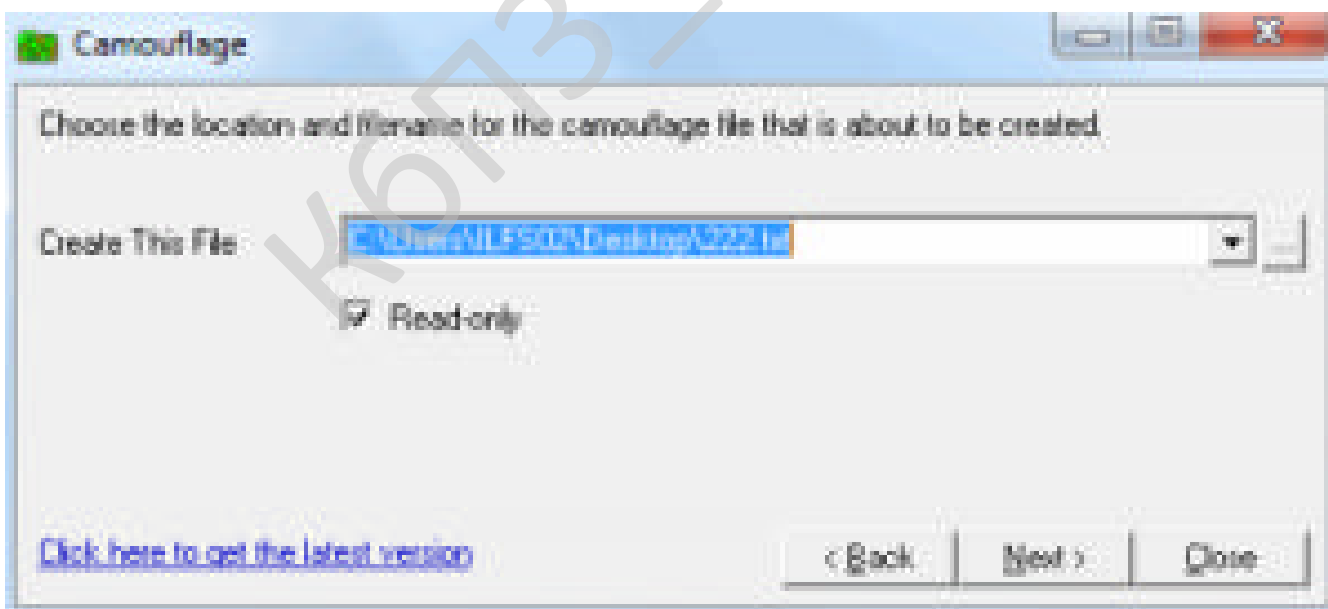


Рисунок 2.4 – Інтерфейс користувача Camouflage

SilentEye

SilentEye – проста у використанні програма крос-платформної стеганографії, яка дозволяє приховувати конфіденційне повідомлення в зображенні або в аудіофайле

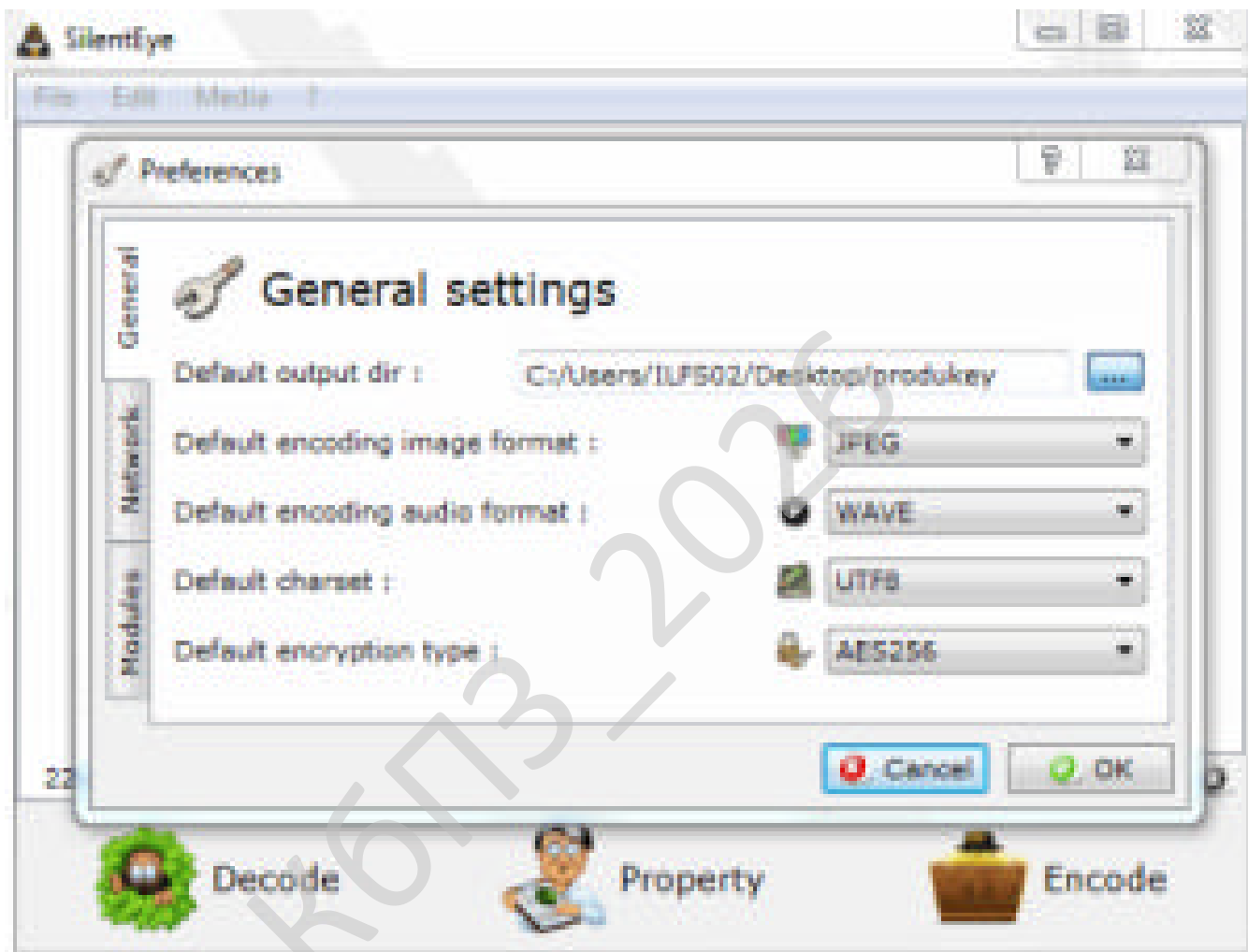


Рисунок 2.5 – Інтерфейс користувача SilentEye

Openpuff



Рисунок 2.6 – Інтерфейс користувача Openpuff

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

OurSecret

Oursecret – безкоштовне й просте у використанні програмне забезпечення для стеганографії, яке дозволяє приховувати секретні дані у файлах зображень.

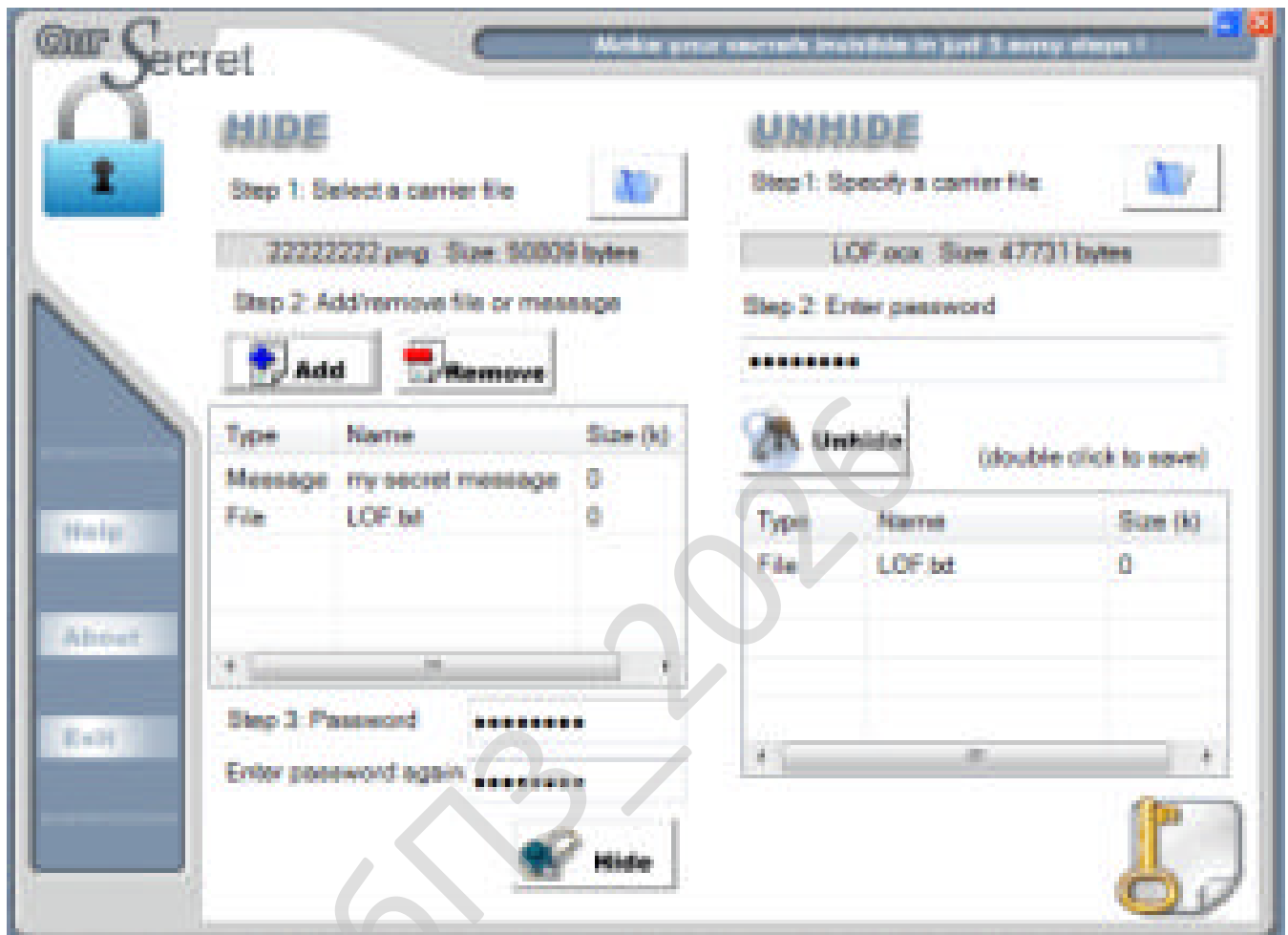


Рисунок 2.7 – Інтерфейс користувача Oursecret

Image Steganography

Image Steganography – безкоштовне програмне забезпечення стеганографії для приховання чутливого тексту або файлів усередині файлів зображень.

Ви можете легко сховати текст або файли різних типів усередині файлів зображень. Він також показує ємність файлу зображення контейнера.

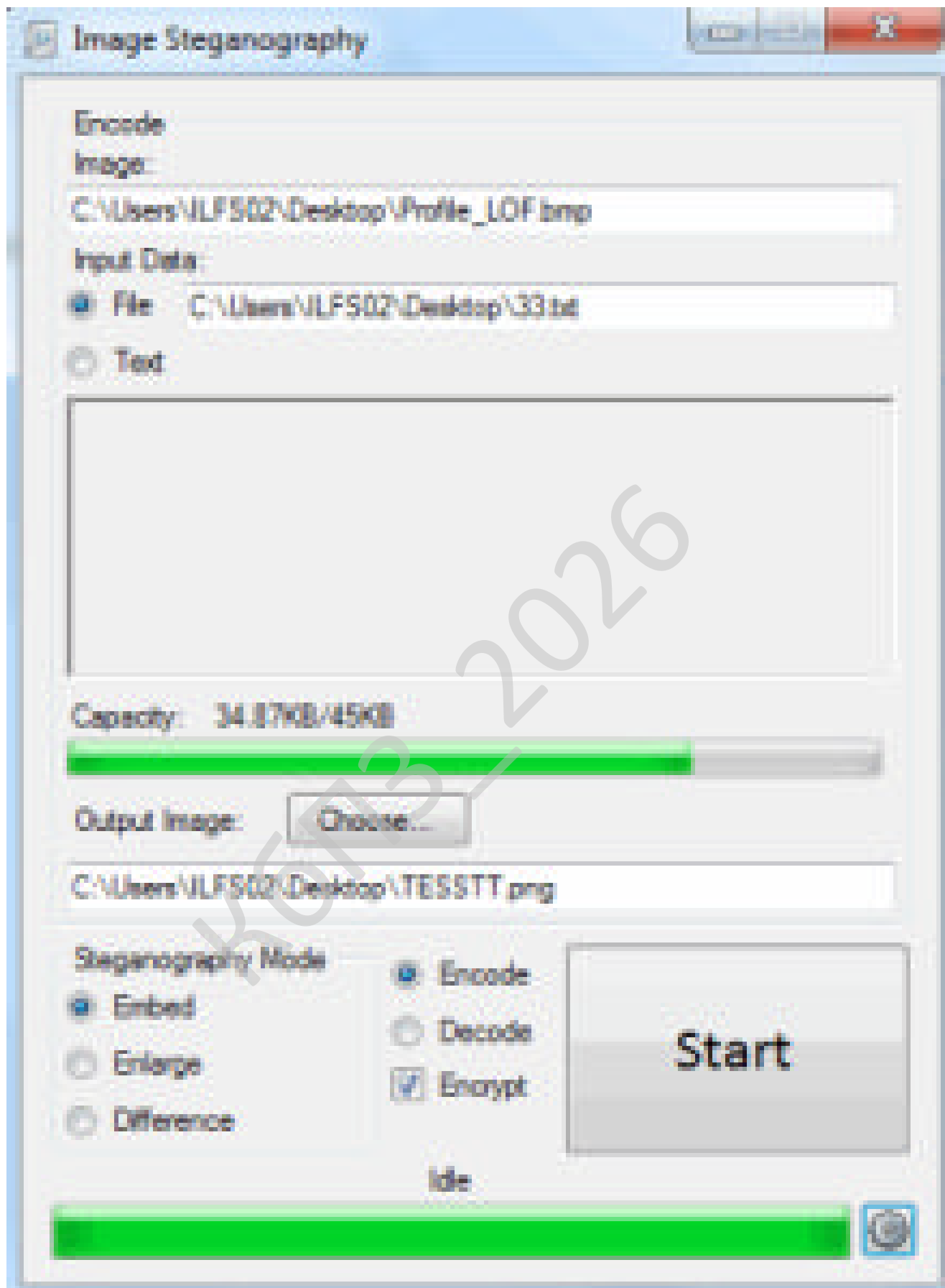


Рисунок 2.8 – Інтерфейс користувача Image Steganography

Steganofile

Steganofile дозволяє приховувати ваш секретний файл в одному або декількох файлах хосту.

Він має кнопки Encode і Decode на своєму інтерфейсі.

Ви можете сховати свій файл в іншому файлі (-ах) хосту.

Виберіть папку призначення й укажіть пароль по вашому вибору.

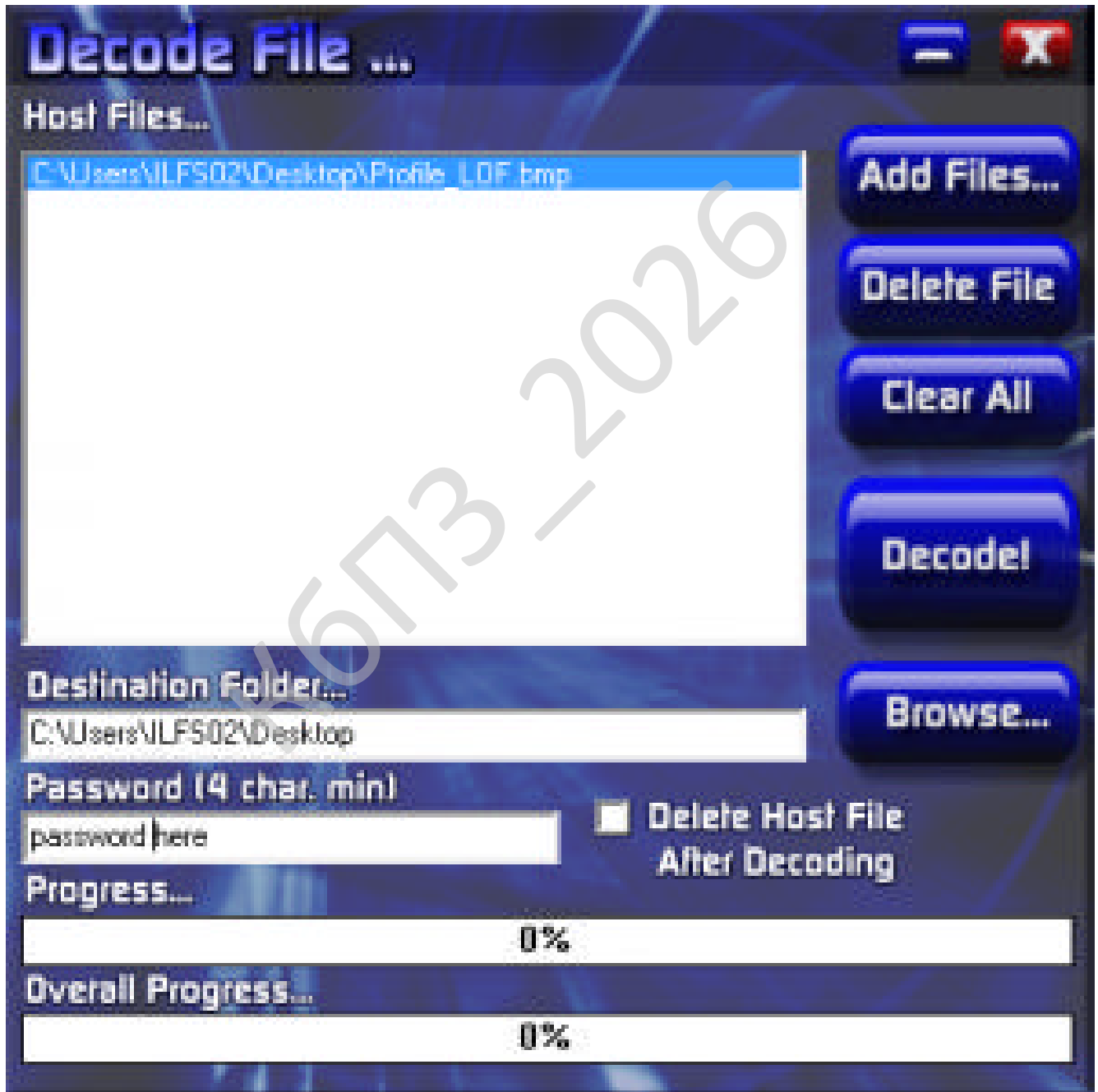


Рисунок 2.9 – Інтерфейс користувача Steganofile

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Python – високорівнева мова програмування, яку називають другою за популярністю в світі. Її використовують для розробки вебзастосунків, програмного забезпечення, машинного навчання. Python застосовують для вирішення робочих завдань у компаніях Google, Instagram, Facebook, IBM, NASA, Dropbox, Netflix та інших. Розробники цінують цю мову програмування за простоту у вивченні, ефективність та мультиплатформність.

Python – скриптова мова програмування з досить простим синтаксисом. Для розуміння достатньо порівняти принципи написання найпростішої програми, яка виводить на екран текстове повідомлення. Саме тому мова програмування Python більш доступна для новачків, а професіонали встигли адаптувати її для вирішення великої кількості завдань. Це мультиплатформне рішення, тому знання Python дає можливість працювати у різних сферах: від розробки мобільних застосунків до ігрової індустрії та штучного інтелекту.

У мови програмування динамічна типізація: є можливість передавати до функцій будь-який тип даних без попереднього вказання. Інтерпретованість дозволяє знаходити помилки у коді ще до повної збірки у робочий застосунок. При цьому Python дуже чітко дає зрозуміти, де та через що виникла помилка.

Це мова об'єктноорієнтованого програмування (ООП). Програмне забезпечення на Python оформлене у вигляді моделей, які можуть бути зібраними у пакети. Тип та структуру кожного об'єкта можна запитати під час виконання програми. Для кожного з об'єктів можна отримати всю інформацію щодо його внутрішньої структури. Окрім того:

- у мови логічний синтаксис, завдяки чому вихідний код легко читати та розуміти;
- гнучкість та масштабованість Python дозволяє адаптувати високорівневу логіку та розширяти складні застосунки, як тільки виникне така необхідність;

- розробка на Python у більшості випадків проходить швидше, ніж на інших мовах програмування;
- Python – інтерпретована мова програмування. Це значить, що код можна написати у будь-якому текстовому файлі на будь-якій платформі, і потім успішно запустити;
- у Python – колосальна спільнота однодумців. Тож будь-які складнощі конкретних розробників вирішуються колективно.

Проте є декілька особливостей, які можна віднести до недоліків. Це повільність (ця мова програмування хоч і універсальна, проте повільніша за інші), велика кількість ресурсів, необхідних для роботи та «прив’язаність» до системних бібліотек.

Мова програмування Python використовується у наступних сферах:

1. Розробка програмних застосунків будь-якого напрямку.
2. Розробка серверної частини мобільних застосунків (найпопулярніший напрямок).
3. Ігри. Багато сучасних ігор для комп’ютерів (наприклад, World of Tanks) частково чи повністю написані на Python.
4. Вбудовані системи для різних пристроїв. Дуже часто Python використовують для написання внутрішніх платформ управління банкоматами.
5. Скрипти та плагіни до уже реалізованих програм для автоматизації процесів чи створення інших рішень.
6. Тестування (автоматизація цього процесу).
7. Машинне навчання. – основна мова для написання алгоритмів і аналітичних застосунків у сфері Machine Learning.

Бібліотеки Python

Різні бібліотеки Python використовують для виконання конкретних завдань. Наприклад, Matplotlib підходить для відображення даних у двовимірній та тривимірній графіці. Pandas підходить для зручної роботи з даними. NumPy дозволяє створювати масиви та керувати ними. Requests використовується для

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

веброзробки. OpenCV-Python відкриває можливості для обробки зображень з метою оптимізації систем «машинного зору».

Найвідоміші фреймворки для мови програмування Python

Фреймворки Python допомагають створити зручне та функціональне середовище для розробки. У них міститься набір інструментів, модулів та бібліотек, корисних для виконання конкретних завдань. Це значно полегшує роботу: наприклад, дає змогу не витрачати час на розписування дій, які повторюються, а використати релевантний інструмент. Тож є можливість позбутися рутинних процесів та сконцентруватися на логіці проекту.

Серед найпопулярніших фреймворків для Python:

- Django – найстаріший та найвідоміший. Створений для реалізації великих інтерактивних проєктів;

- Pyramid – зручний у налаштуваннях, і дає можливість реалізувати складні нестандартні ідеї;

- Web2py – підходить в першу чергу для вебзастосунків і може використовуватись на будь-яких архітектурах.

Популярні Python IDE

IDE або інтегровані середовища розробки – це програмне забезпечення, яке надає розробникам необхідні інструменти для написання, редагування, тестування та налаштування коду. Для розробки на Python найчастіше використовують IDE PyCharm, IDLE, Spyder та Atom.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи приховування інформації на базі використання генеративно-змагальних мереж.

В процесі розробки випускної кваліфікаційної роботи за першим

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

(бакалаврським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Стеганографія зображень [17] використовує статистичну надмірність зображень обкладинок та перцептивні характеристики людського зору для вбудовування секретної інформації шляхом зміни певних одиниць вбудовування (наприклад, значень пікселів, коефіцієнтів DCT), що полегшує прихований зв'язок. Сучасні методи стеганографії базуються на системі мінімального спотворення [7], де основна увага приділяється розробці функцій спотворення для кожної одиниці вбудовування, а потім схемам кодування (наприклад, STC [6], SPC [26]) для вбудовування секретної інформації. Типовий метод включає HUGO [18], WOW [9], S-UNIWARD [10] та HILL [13]. Ці методи розподіляють нижчі витрати на вбудовування на багаті на текстури області та вищі витрати на більш гладкі, простіші області. Цей підхід вбудовує більше секретних даних в області, де зміни пікселів менш помітні та важче моделювати, що підвищує безпеку стеганографії. Вищезазначені методи є симетричними методами стеганографії, які однаково обробляють витрати на вбудовування модифікацій +1 та -1 пікселя. На противагу цьому, Лі та ін. і Денемарк та ін. запропонували стратегію асиметричної стеганографії, CMD[14] та SMD[4] відповідно. Ці підходи сегментують зображення обкладинки на неперекриваючі підзображення, послідовно вбудовують дані та оновлюють витрати в незмінених областях на основі напрямку змін пікселів. Це призводить до асиметричних витрат на вбудовування, що підвищує загальну безпеку стеганографії.

Однак усі вищезгадані методи розроблені для зображень у градаціях сірого. Безпосереднє застосування цих методів до окремих кольорових каналів кольорових зображень може призвести до помітних артефактів вбудовування, тим самим знижуючи їхню безпеку. Хоча деякі стеганографічні методи були

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

розроблені для кольорових зображень, такі як стратегія CMD-C [22] Танга та ін., яка розширює метод CMD на кольорові зображення, вона менш ефективна через нездатність враховувати кореляції між кольоровими каналами. Ляо та ін. запропонували стратегію ASCMP [16], яка агрегує зміни у вбудованому повідомленні, коригуючи початкове спотворення в точках модифікації по всіх трьох каналах, концентруючи зміни в областях текстури. Ван та ін. представили асиметричну стратегію GINA [23], натхненну CMD, яка враховує кореляції та відмінності між каналами. Цей підхід забезпечує синхронізацію між червоним (r) та синім (b) каналами із зеленим (g) каналом та вибірково оновлює витрати на вбудовування на основі принципу пріоритету складності, значно підвищуючи безпеку стеганографії кольорових зображень. Зверніть увагу, що згадані вище методи в основному спираються на існуючі алгоритми градацій сірого (наприклад, S-UNIWARD та HILL) для ініціалізації витрат на вбудовування для кожного каналу перед застосуванням запропонованих стратегій оновлення витрат. З розвитком глибокого навчання, змагальні приклади та генеративно-змагальні мережі (GAN) також були застосовані до стеганографії кольорових зображень. Наприклад, Qin та ін. представили змагальний метод на основі вибірки GEAP [19], який використовує існуючу кольорову стеганографію - CPV [20] для ініціалізації витрат на вбудовування, а потім оновлює їх за допомогою змагального механізму, щоб запобігти виявленню цільовими моделями стеганалізу. Однак його безпека виявилася слабшою порівняно зі стратегією GINA.

На відміну від вищезгаданих методів, Ляо та ін. запропонували перший підхід на основі GAN, CIS-GAN [15], для проектування стеганографічної функції спотворення для кольорових зображень. Цей метод дозволяє вивчати асиметричні витрати на вбудовування кольорових зображень з нуля. Однак він має два обмеження. По-перше, модель використовує відносно слабший стеганалізатор, Xu-Net [27], як дискримінатор, що, ймовірно, обмежує продуктивність генератора. По-друге, модель використовує стратегію одночасного вбудовування,

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

вбудовуючи інформацію в усі кольорові канали одночасно. Такий підхід ускладнює ефективне захоплення кореляцій між каналами, обмежуючи його здатність підвищувати безпеку стеганографічного методу. Як результат, його ефективність безпеки є незадовільною.

У даній роботі ми пропонуємо новий метод стеганографії кольорових зображень, заснований на генеративно-змагальних мережах (GAN). Метод використовує стратегію поетапного вбудовування, яка враховує кореляції між кольірними каналами, розділяючи процес навчання мережевої моделі на три етапи. Кожен етап вивчає витрати на вбудовування для певного кольорового каналу. З точки зору проектування моделі, кожен етап оснащений генератором з двома гілками та дискримінатором, що складається з кількох мереж стеганалізу. Крім того, ми вводимо просту, але ефективну втрату середнього абсолютного відхилення (MAD), яка динамічно коригує ваги втрат розрізнення зображень усіх мереж у дискримінаторі. Це допомагає збалансувати навчання генератора та дискримінатора, поступово покращуючи стеганографічну продуктивність генератора. Наш метод дозволяє крок за кроком вивчати асиметричні витрати на вбудовування для кожного кольорового каналу з нуля, не покладаючись на традиційні алгоритми стеганографії. Широкі експерименти демонструють, що наш підхід перевершує існуючі методи з точки зору безпеки. Підсумовуючи, основні досягнення цієї статті полягають у наступному:

Ми пропонуємо новий фреймворк на основі GAN для стеганографії кольорових зображень, який з нуля вивчає асиметричні витрати на вбудовування різних кольорових каналів. Цей фреймворк використовує поетапну стратегію вбудовування для ефективного фіксування міжканальних кореляцій та систематично досліджує вплив різних порядків вбудовування каналів на стеганографічну безпеку, що зрештою покращує її.

Ми вводимо функцію втрат MAD, яка динамічно регулює ваги втрат різних мереж у дискримінаторі. Цей механізм забезпечує збалансований процес навчання між генератором та дискримінатором, стабілізує збіжність моделі та

поступово покращує стеганографічну продуктивність генератора.

Широкі експерименти з наборами кольорових зображень, включаючи ALASKA II та BOSSBase, демонструють, що наш метод значно перевершує існуючі підходи з точки зору безпеки. Крім того, багато досліджень абляції підтверджують ефективність нашого мережевого дизайну та функції втрат у покращенні стеганографічної продуктивності.

3.2 Розробка структурної схеми

Узагальнена модель стegosистеми у вигляді структурної схеми представлена на рисунку нижче.



Рисунок 3.1 – Структурна схема системи

Контейнер – будь-яка інформація, призначена для приховання таємних повідомлень.

Порожній контейнер – контейнер без вбудованого повідомлення; заповнений контейнер або стегоконтейнер, що містить вбудовану інформацію.

Вбудоване (сховане) повідомлення – повідомлення, що вбудовується в контейнер.

Стежоключ або просто ключ – секретний ключ, необхідний для приховання інформації. Залежно від кількості рівнів захисту (наприклад, вбудовування попереднє зашифрованого повідомлення) у стегосистемі може бути один або трохи стегоключів.

У цьому розділі ми детально розглядаємо запропоновану модель GAN, охоплюючи поетапний процес навчання та вбудовування, проектування генераторів та дискримінаторів, а також адаптивну стратегію оновлення та функції втрат.

Генеративно-змагальна мережа (GAN)

Генеративно-змагальні мережі (GAN) допомагають машинам створювати нові, реалістичні дані, навчаючись на існуючих прикладах. Вони були представлені Ієном Гудфеллоу та його командою у 2014 році, і вони змінили те, як комп'ютери генерують зображення, відео, музику тощо.

На відміну від традиційних моделей, які лише розпізнають або класифікують дані, вони використовують креативний підхід, створюючи абсолютно новий контент, який дуже нагадує дані реального світу. Ця здатність допомогла різним галузям, таким як мистецтво, ігри, охорона здоров'я та наука про дані. У цьому розділі ми розглянемо більше про GAN та їхні основні концепції.

Архітектура GAN

GAN складається з двох основних моделей, які працюють разом для створення реалістичних синтетичних даних, а саме:

1. Модель генератора

Генератор – це глибока нейронна мережа, яка приймає випадковий шум як вхідні дані для генерації реалістичних зразків даних, таких як зображення або текст. Він вивчає основні шаблони даних, коригуючи свої внутрішні параметри

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

під час навчання за допомогою зворотного поширення . Його метою є створення зразків, які дискримінатор класифікує як реальні.

Функція втрат генератора: Генератор намагається мінімізувати ці втрати:

Генератор прагне максимізувати тобто він хоче, щоб дискримінатор класифікував його підроблені дані як справжні (ймовірність близька до 1).

2. Модель дискримінатора

Дискримінатор діє як бінарний класифікатор, що допомагає розрізняти реальні та згенеровані дані. Він навчається покращувати свої класифікаційні можливості шляхом навчання, уточнюючи свої параметри для точнішого виявлення підроблених зразків. Під час роботи з даними зображень дискримінатор використовує згорткові шари або інші відповідні архітектури, які допомагають виявляти ознаки та розширювати можливості моделі.

Функція втрат дискримінатора: Дискримінатор намагається мінімізувати ці втрати:

Дискримінатор хоче правильно класифікувати реальні дані як реальні і фальшиві дані як фальшиві.

Мін.Макс. Втрата

GAN навчаються з використанням MinMax Loss між генератором та дискримінатором:

Генератор намагається мінімізувати ці втрати (обдурити дискримінатор), а дискримінатор намагається максимізувати їх (точно виявляти підробки).

Навчання GAN

Навчання GAN відбувається за допомогою двох мереж: генератора (G) та дискримінатора (D), які конкурують та вдосконалюються разом. Ось покроковий процес.

1. Перший хід генератора

Генератор починає роботу з вектора випадкового шуму, такого як випадкові числа. Він використовує цей шум як відправну точку для створення

фальшивого зразка даних, такого як згенероване зображення. Внутрішні шари генератора перетворюють цей шум на щось, що виглядає як реальні дані.

2. Черга дискримінатора

Дискримінатор отримує два типи даних:

- Реальні зразки з фактичного навчального набору даних.
- Підроблені зразки, створені генератором.

Завдання D полягає в аналізі кожного вхідного сигналу та визначенні, чи це реальні дані, чи щось вигадане G. Він видає оцінку ймовірності від 0 до 1. Оцінка 1 показує, що дані, ймовірно, реальні, а 0 означає, що вони підроблені.

3. Змагальне навчання

- Якщо дискримінатор правильно класифікує реальні та підроблені дані, він краще виконує свою роботу.
- Якщо генератор обдурює дискримінатор, створюючи реалістичні підроблені дані, він отримує позитивне оновлення, і дискримінатор карається за прийняття неправильного рішення.

4. Покращення генератора

- Щоразу, коли дискримінатор помилково приймає фальшиві дані за справжні, генератор вчиться на цьому успіху.
- Шляхом багатьох ітерацій генератор покращується та створює переконливіші фальшиві зразки.

5. Адаптація дискримінатора

- Дискримінатор також постійно навчається, оновлюючись, щоб краще виявляти підроблені дані.
- Цей постійний обмін інформацією з часом зміцнює обидві мережі.

6. Прогрес у навчанні

- У міру продовження навчання генератор стає дуже вправним у створенні реалістичних даних.
- Зрештою, дискримінатор намагається відрізнити справжні дані від підроблених, що показує, що GAN досягла добре навченого стану.

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

– На цьому етапі генератор може створювати високоякісні синтетичні дані, які можна використовувати для різних застосувань.

Типи GAN

Існує кілька типів GAN, кожен з яких призначений для різних цілей. Ось деякі важливі типи:

1. Vanilla GAN

Vanilla GAN – це найпростіший тип GAN. Він складається з:

- Генератор і дискримінатор побудовані з використанням багатошарових перцептронів (MLP).
- Модель оптимізує свою математичну формулювання за допомогою стохастичного градієнтного спуску (SGD).

Хоча Vanilla GAN є базовою, вона може зіткнутися з такими проблемами, як:

- Згортання режиму: Генератор повторно створює обмежені типи вихідних сигналів.
- Нестабільне навчання: Генератор і дискримінатор можуть не покращуватися плавно.

2. Умовна GAN (CGAN)

Умовна GAN (CGAN) додає додатковий умовний параметр для керування процесом генерації. Замість випадкової генерації даних, вона дозволяє моделі створювати певні типи результатів.

Робота CGAN:

- Умовна змінна (y) подається як у генератор, так і в дискримінатор.
- Це гарантує, що генератор створює дані, що відповідають заданій умові (наприклад, генерує зображення певних об'єктів).
- Дискримінатор також отримує мітки, які допомагають розрізняти справжні дані від підроблених.

Приклад: Замість генерації будь-якого випадкового зображення, CGAN може генерувати певний об'єкт, такий як собака чи кіт, на основі мітки.

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

3. Глибока згорткова GAN (DCGAN)

Глибока згорткова GAN (DCGAN) є одними з найпопулярніших типів GAN, що використовуються для створення зображень.

Вони важливі, тому що:

- Використовує згорткові нейронні мережі (CNN) замість простих багатошарових перцептронів (MLP).
- Шари максимального об'єднання замінюються згортковим кроком, що допомагає зробити модель ефективнішою.
- Повністю зв'язані шари видаляються, що дозволяє краще просторово розуміти зображення.

DCGAN успішні, оскільки вони генерують високоякісні, реалістичні зображення.

4. Лапласова піраміда GAN (LAPGAN)

Лапласова пірамідальна GAN (LAPGAN) розроблена для створення надвисокоякісних зображень за допомогою підходу з кількома роздільними здатностями.

Робота LAPGAN:

- Використовує кілька пар генератор-дискримінатор на різних рівнях піраміди Лапласа.
- Зображення спочатку знижують роздільну здатність на кожному шарі піраміди, а потім знову масштабують за допомогою умовного GAN (CGAN).
- Цей процес дозволяє зображенню поступово уточнювати деталі, допомагає зменшити шум і покращити чіткість.

Завдяки своїй здатності створювати високодеталізовані зображення, LAPGAN вважається найкращим підходом для створення фотореалістичних зображень.

5. GAN з надвисокою роздільною здатністю (SRGAN)

Надроздільна GAN (SRGAN) розроблена для збільшення роздільної здатності зображень низької якості, зберігаючи при цьому деталі.

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

Робота SRGAN:

- Використовує глибоку нейронну мережу в поєднанні з функцією змагальних втрат.
- Покращує зображення з низькою роздільною здатністю, додаючи дрібніші деталі, допомагаючи зробити їх чіткішими та реалістичнішими.
- Допомагає зменшити поширені помилки масштабування зображення, такі як розмиття та пікселізація.

3.3 Розробка функціональної схеми

Актуальність роботи: захист авторського права в даний момент відіграє більшу роль при розгляді питань інформаційної безпеки. Механізм цифрових водяних знаків популярний як для захисту різних зображень (фото, логотипи), так і для захисту відео-файлів і аудіо-файлів. Пропонована програмна система може використовуватися як засіб для захисту авторського права користувача при роботі з файлами у форматі .bmp і .png. Запропонований програмний продукт можна адаптувати для захисту авторського права у відео-файлах після створення додаткового модуля роботи з форматом .avi [1].

При використанні стеганографічних алгоритмів передана інформація ховається у файлі. Дані, усередині яких розміщаються сховані повідомлення, називаються стегоконтейнером, а самі сховані повідомлення – стегоповідомленнями. Канал передачі стегоконтейнера називається стеганографічним каналом або стегоканалом. До стеганографії ставляться безлічі різних методів: умовні розташування знаків, мікрофотознімки, засоби зв'язку на плаваючих частотах і таємні канали. Стеганографія є відмінним доповненням, але ніяк не заміною криптографії [2].

Використання авторського алгоритму разом з існуючими криптографічними алгоритмами шифрування дозволить більш надійно захистити

ее передачу, що вбудовується в графічні зображення інформацію й подальшу, по каналах зв'язку.

На сьогоднішній день існує величезна кількість відкрита розповсюджуваних програмних продуктів, що використовують стеганографічну техніку. Найбільш популярні інструменти розміщуються в Github-репозиторіях.

У рамках даного проекту розроблений стеганографічний веб-сервіс нового покоління, що дозволяє приховувати конфіденційну інформацію у вигляді непомітного шуму в JPEG-зображеннях, при цьому виділити дану інформацію можна тільки знаючи заданий при кодуванні секретний ключ-пароль. Сервіс використовує стійкі методи стеганографії для приховання самого факту приховання інформації разом зі стійкими методами криптографії для захисту даних, переданих по відкритих каналах [3].

Основні особливості:

- використання SHA-3 для генерації ключів;
- симетричне шифрування AES-256;
- JPEG (DCT LSB) стеганографія;
- підтримка RARJPEG і подвійного приховання;
- вибір випадкового контейнера;
- обчислення без участі сервера (на клієнтській стороні);
- гарантія повної конфіденційності [3].

Стеганографічний застосунок надає дві можливості:

1. Приховання даних (можна сховати будь-які дані усередині файлу-контейнера).

2. Водяні знаки (вбудовування невидимого підпису, наприклад у зображенні, з метою виявлення несанкціонованого копіювання).

Дана програма надає можливість надійного вбудовування водяних знаків при незначній зміні розміру або кадрування зображення [4].

При розробці системи насамперед необхідно визначити вимоги до програми. На рисунку 3.2 показана функціональна схема системи розроблювальної системи.

При проектуванні авторської стегосистеми можна виділити три етапи:

- одержання із цифрового документа байтів;
- перетворення байтів цифрового документа (підготовка даних для впровадження в контейнер);
- впровадження даних у зображення, використовуючи метод LSB [6].



Рисунок 3.2 – Функціональна схема системи

На першому етапі деякий цифровий об'єкт повинен бути представлений у вигляді потоку байт. Якщо це файл, то його можна відкрити в бінарному режимі. Якщо текстове повідомлення, то представити його у вигляді байтів, заданим кодуванням [7]. На другому етапі, на вхід подаються байти, отримані на першому етапі. Насамперед необхідно обчислити контрольну геш-суму [8] вихідних байт. Після цього стиснути вихідні байти, обраним користувачем алгоритмом стиску [9]. До стислих байтів у початок додається заголовок (метадані). Після чого ці байти шифруються симетричним алгоритмом шифрування [10].

Останній етап полягає в тому, що зашифровані байти впроваджуються в зображення.

3.4 Розробка діаграми процесів

Відповідно до методичних рекомендацій розроблення графічної частини кваліфікаційної бакалаврської роботи розглянемо розроблену діаграму процесів яка зображена на рисунку 3.3.



Рисунок 3.3 – Діаграма взаємодії процесів

Розроблена діаграма взаємодії процесів використовується для представлення та візуалізації процесів обробки даних тобто структурного проектування бакалаврської роботи.

Основні складові елементи діаграми взаємодії процесів це потоки даних:

- Репозиторії, потік сховища даних.
- Потоки зовнішні по відношенню до системи сутності.
- Процеси які являють собою трансформацію даних в рамках описуваної системи.
- Потоки даних гібридні між елементами трьох попередніх типів.

Відповідно до документації основна будова діаграми процесів полягає у графічному представленні складу сукупностей даних, що характеризуються як співвідношення різних частин кожної з сукупностей. Склад статистичної сукупності графічно може бути представлений як за допомогою абсолютних, так і відносних показників. Графічне зображення складу сукупності по абсолютними і відносними показниками сприяє проведенню більш глибокого аналізу і дозволяє проводити аналіз системи.

Для схематичного представлення системи що розробляється необхідно спочатку представити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи в цілому у подальшому. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

Розроблена діаграма взаємодії процесів системи в подальшому уточнюється шляхом деталізації процесів та потоків даних з метою показати систему що розробляється. Таким чином у результаті після розгляду, вищеописаної системи, схеми структурної, функціональної, діаграми взаємодії процесів перейдемо до опису та розгляду блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Розглянемо реалізацію бакалаврської дипломної роботи. Були проведені розрахунки і підібрані набори тестових даних для перевірки правильності реалізації проектних рішень.

Було створено блок-схеми роботи системи. Перед їх розглядом необхідно провести роз'яснення який саме тип блок-схем використовується. Блок-схеми показують весь процес роботи системи з підсистемами та частково доказують правильність вибраних проектних рішень. Тому від точності і детальної блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає високого рівня декомпозиції задач на класи.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підсистеми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ.

При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

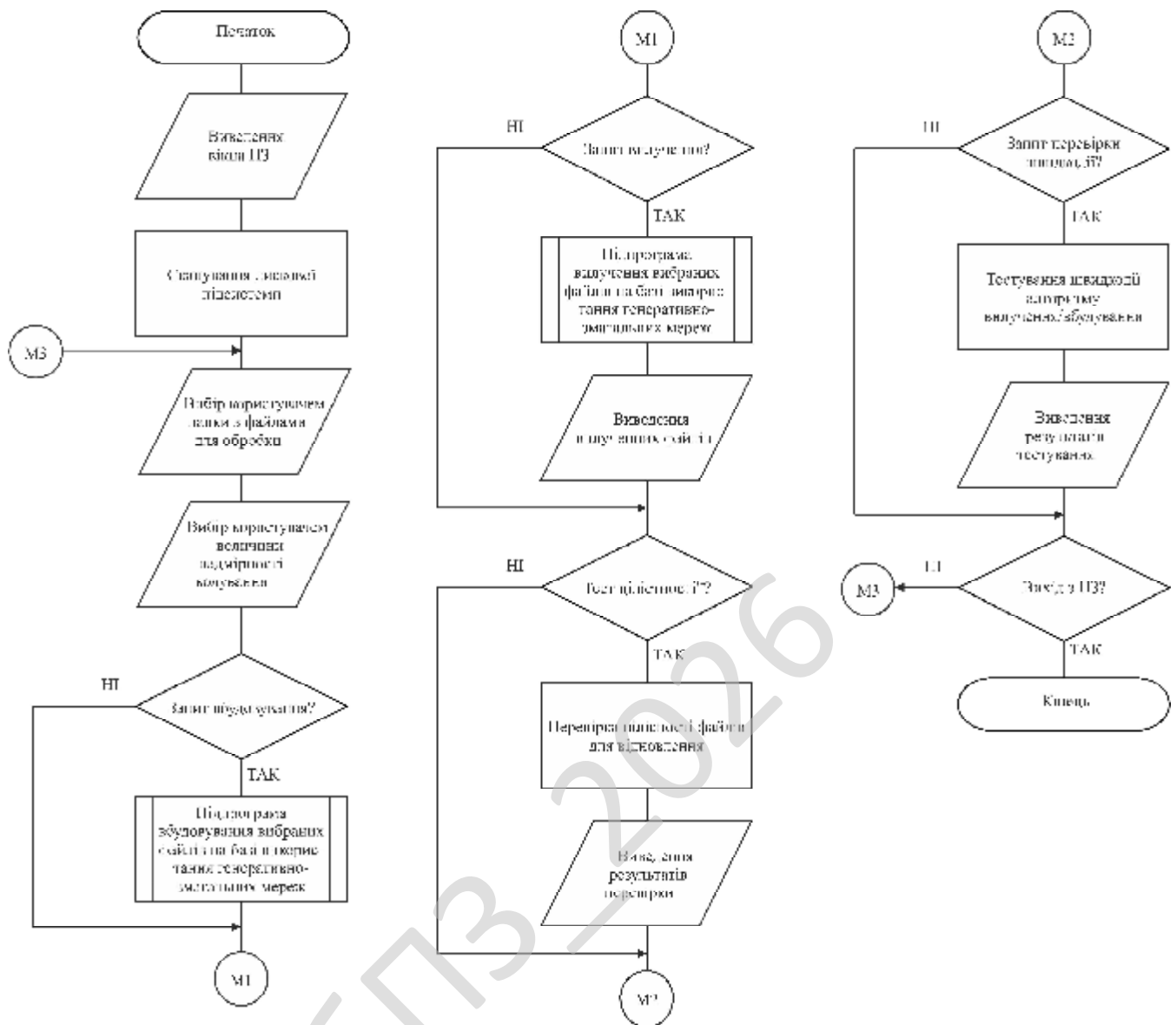


Рисунок 4.1 – Блок-схема основної програми

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю.

UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем.

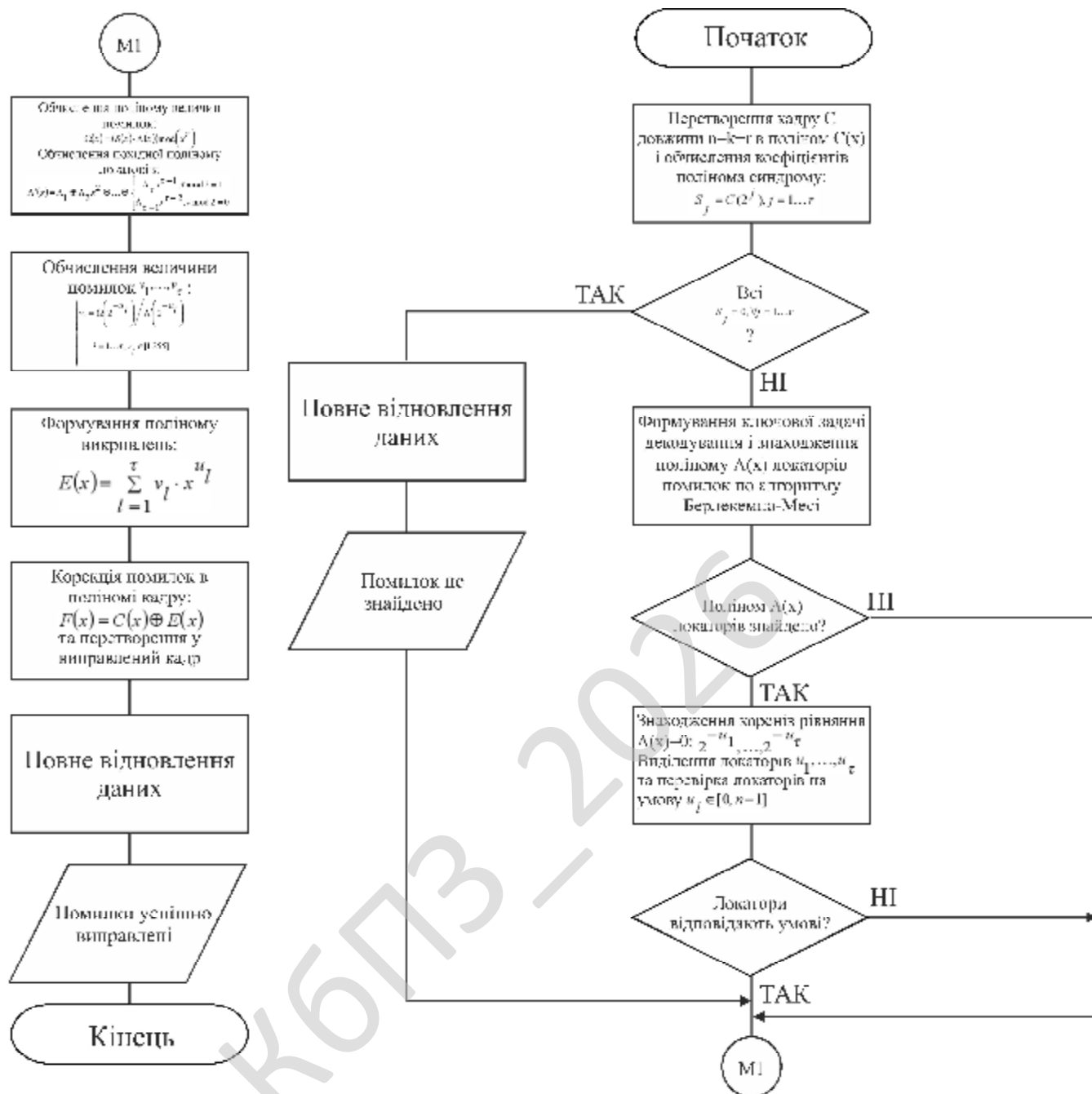


Рисунок 4.2 – Блок-схема роботи підпрограми

UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

Було використано наступні підходи UML: діаграма діяльності (діаграми поведінки типу); діаграма прецедентів (діаграми поведінки типу).

Діаграма діяльності. Це візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій. Дія є фундаментальною одиницею визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів.

Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії. Подібно до цього, виконання діяльності є виконанням окремої діяльності, буквально, включно із виконанням тих дій, що містяться в діяльності. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності. Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності.

Специфікація діяльності (на вищих рівнях сумісності) може дозволяти виконання декількох (логічних) потоків, та існування механізмів синхронізації для гарантування виконання дій у правильному порядку.

Діаграма прецедентів це діаграма, на якій зображено відношення між акторами та прецедентами в системі. Також, перекладається як діаграма варіантів використання.

Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених границею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання.

Суть даної діаграми полягає в наступному: проєктована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант використання (use case) використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором.

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою.

У мові UML є кілька стандартних видів відношень між акторами і варіантами використання:

- асоціації (association relationship);
- включення (include relationship);
- розширення (extend relationship);
- узагальнення (generalization relationship).

При цьому загальні властивості варіантів використання можуть бути представлені трьома різними способами, а саме — за допомогою відношень включення, розширення і узагальнення.

Відношення асоціації – одне з фундаментальних понять у мові UML і в тій чи іншій мірі використовується при побудові всіх графічних моделей систем у формі канонічних діаграм.

Включення (include) у мові UML – це різновид відношення залежності між базовим варіантом використання і його спеціальним випадком. При цьому відношенням залежності (dependency) є таке відношення між двома елементами моделі, при якому зміна одного елемента (незалежного) приводить до зміни іншого елемента (залежного).

Відношення розширення (extend) визначає взаємозв'язок базового варіанта використання з іншим варіантом використання, функціональна поведінка якого задіюється базовим не завжди, а тільки при виконанні додаткових умов.

При розробці ПЗ було використано підходи ризик-менеджменту – це система управління ризиками, яка включає в себе стратегію та тактику управління, направлені на досягнення основних цілей. Ефективний ризик-менеджмент включає:

- систему управління;
- систему ідентифікації і вимірювання;
- систему супроводження (моніторингу та контролю).

Сучасна наука представляє ризик як вірогідну подію, в результаті настання якої можуть відбутися позитивні, нейтральні або негативні наслідки. Якщо ризик припускає наявність як позитивних, так і негативних результатів, він відноситься до спекулятивних ризиків. Якщо ж наслідки негативні, або відсутні взагалі, такий ризик іменується чистим.

Мета ризик-менеджменту – підвищення конкурентоспроможності господарюючих суб'єктів за допомогою захисту від реалізації чистих ризиків.

Теорія ризик-менеджменту ґрунтується на трьох базових поняттях: корисності, регресії і диверсифікації.

У 1738 швейцарський математик Даніель Бернуллі доповнив теорію вірогідності методом корисності або привабливості того або іншого результату подій. Ідея Бернуллі полягала в тому, що в процесі ухвалення рішення люди приділяють більше уваги розміру наслідків різних результатів, ніж їх вірогідність.

В кінці XIX століття англійський дослідник Ф. Гальтон запропонував вважати регресію або повернення до середнього значення універсальною статистичною закономірністю. Суть регресії трактувалася ним як повернення явищ до норми з часом. Згодом було доведено, що правило регресії діє в найрізноманітніших ситуаціях, починаючи з азартних ігор та розрахунку вірогідності виникнення нещасних випадків, і закінчуючи прогнозуванням коливань економічних циклів.

У 1952 аспірант Університету Чикаго Гарі Марковіц в статті «Диверсифікація вкладень» («Portfolio Selection») математично обґрунтував стратегію диверсифікації інвестиційного портфеля, зокрема, він показав, як шляхом продуманого розподілу вкладень мінімізувати відхилення прибутковості від очікуваного показника. У 1990 Г. Марковіцу присуджена Нобелівська премія за розробку теорії і практики оптимізації портфеля фондових активів.

Етапи ризик-менеджменту

У ризик-менеджменті прийнято виділяти декілька ключових етапів:

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

- на першому етапі відбувається виявлення ризику з супутньою оцінкою вірогідності його реалізації і масштабу наслідків;
- на другому етапі здійснюється розробка ризик-стратегії з метою зниження вірогідності реалізації ризику і мінімізації можливих негативних наслідків;
- на третьому етапі вибираються методи і інструменти управління виявленим ризиком;
- на четвертому етапі проводиться безпосереднє управління ризиком;
- на завершальному етапі оцінюються досягнуті результати і коректується ризик-стратегія.

За ключовий етап ризик-менеджменту вважається етап вибору методів і інструментів управління ризиком.

Методи і інструментарій ризик-менеджменту

Базовими методами ризик-менеджменту є відмова від ризиків, зниження, передача і ухвалення.

Ризик-інструментарій значно ширший. Він включає політичні, організаційні, правові, економічні, соціальні інструменти, причому ризик-менеджмент як система допускає можливість одночасного застосування декількох методів і інструментів ризик-управління.

Найбільш часто вживаним інструментом ризик-менеджменту є страхування. Страхування припускає передачу відповідальності за відшкодування передбачуваного збитку сторонній організації (страхової компанії).

Прикладами інших інструментів можуть бути відмова від надмірно ризикової діяльності (метод відмови), профілактика або диверсифікація (метод зниження), аутсорсинг витратних ризикових функцій (метод передачі), формування резервів або запасів (метод ухвалення).

Незважаючи на те що я працював над ПЗ один в реалізації програми я використовував підходи пришвидшення розробки на основі методологій Agile – Extreme Programming.

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

Екстремальне програмування (Extreme Programming, далі XP) це методологія розробки програмного забезпечення, найпопулярніша серед так званих гнучких методологій. Має на меті поліпшення якості програмного забезпечення та чутливість до змін у вимогах замовників. Як вид гнучких методологій, XP радить часті "випуски" програми у коротких циклах розробки, що має на меті поліпшити продуктивність праці та покращити можливості виконання вимог замовника що змінюються. Авторами даної методології є Кент Бек, Ворд Каннінгем, Мартін Фаулер та інші.

Інші елементи екстремального програмування включають в собі: парне програмування, проведення обширної перевірки сирцевого коду, модульне тестування всього коду, уникання створення функціональності до того як вона дійсно необхідна, простота та ясність коду, очікування на зміну вимог замовників з плином часу та коли вимоги до продукту стають ясніші, досить часте спілкування із замовником та між самими програмістами.

Назва методології походить від ідеї застосувати корисні методи і практики розробки програмного забезпечення, піднявши їх до "екстремальних" рівнів.

Критики XP зауважують на потенційні недоліки цієї методології – нестабільні вимоги, незадокументовані компроміси конфліктів користувачів, відсутність загального документу дизайну програми.

Технологія екстремального програмування була розроблена Кентом Бекем, Уардом Каннінгемом та Роном Джеффріесом під час роботи над Chrysler Comprehensive Compensation System (C3). У 1996 Кент Бек став лідером проекту і почав вдосконалювати методи розробки, що застосовувалися в роботі над проектом. Свій метод він виклав у книзі «Extreme Programming Explained», котру було видано у жовтні 1999. Після купівлі Крайслера компанією Даймлер–Бенц проект C3 було скасовано у лютому 2000.

Хоча саме екстремальне програмування є відносно новим, багато її практик вже існували і використовувались протягом певного часу; однак, методологія підносить "найкращі практики" до екстремального рівня. Для

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

прикладу, практика по плануванню і написанню тестів перед написанням кожної маленької частини коду було використано раніше в проекті НАСА "Меркурій". Для зменшення часу на розробку ПЗ деякі формальні документи тестування (такі як приймальне тестування) писались паралельно (або й раніше) з написанням самого ПЗ. Незалежна група тестування НАСА може писати процедури тестування базуючись на формальних вимогах до продукту до того як програмне забезпечення розроблене та інтегроване в систему. В ХР ця концепція піднесена до "екстремального рівня" завдяки написанню автоматичних тестів які перевіряють поведінку навіть малих частинок коду, а не тільки значних функціональних частин ПЗ.

Посібник Extreme Programming Explained: Embrace Change описує Екстремальне Програмування, як:

- Спроба примирити гуманність і продуктивність.
- Механізм для соціальної зміни.
- Шлях до удосконалення.
- Стиль розвитку.

Дисципліна розробки програмного забезпечення.

Головною метою Екстремального Програмування є скорочення вартості неочікуваних змін. У традиційних методах розробки (на кшталт SSADM) вимоги до розвитку системи визначаються на початку роботи над проектом, і часто виправляються пізніше. Це означає, що вартість проекту через зміни буде більшою за заплановану (традиційна особливість для програмного забезпечення, що проектується).

ХР використовується для скорочення вартості змін, завдяки представленню простих значень, принципів і методів. При використанні екстремального програмування, проект повинен стати гнучкішим щодо змін.

Extreme Programming Explained описує екстремальне програмування як дисципліну розробки програмного забезпечення яка змушує людей створювати високоякісне ПЗ якомога швидше.

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

ХР намагається зменшити ціну зміни вимог до ПЗ завдяки малим циклам розробки, а не одним довгим циклом. Екстремальне програмування сприймає зміни до вимог як звичайні, неминучі та бажані аспекти розробки ПЗ, і ці зміни мають бути очікуваним. Основна ідея полягає в тому що неможливо розробити самодостатній пакет вимог до ПЗ, зміни в вимогах – неминучі.

Екстремальне програмування також вводить набір практик та принципів на основі методології гнучкої розробки програмного забезпечення.

Екстремальне програмування описує чотири базові активності що виконуються при розробці програмного забезпечення: написання коду, тестування, слухання та дизайн. Написання коду. Прихильники ХР заявляють що єдиним дійсно важливим результатом розробки ПЗ є код: без готового коду нема продукту.

Тестування. Методологія екстремального програмування заявляє, що якщо дрібне тестування може перевірити незначну частину функціональності, то багато дрібних тестів можуть перевірити набагато більше частинок і продукт в цілому.

Основні прийоми ХР. Дванадцять основних прийомів екстремального програмування (за першим виданням книги Extreme programming explained) можуть бути об'єднані в чотири групи:

1. Короткий цикл зворотного зв'язку (Fine scale feedback).
 - 1.1. Розробка через тестування (Test driven development).
 - 1.2 Гра в планування (Planning game).
 - 1.3. Замовник завжди поруч (Whole team, Onsite customer).
 - 1.4 Парне програмування (Pair programming).
2. Безперервний, а не пакетний процес.
 - 2.1 Безперервна інтеграція (Continuous Integration).
 - 2.2 Рефакторинг (Design Improvement, Refactor).
 - 2.3 Часті невеликі релізи (Small Releases).
3. Розуміння, що поділяється всіма учасниками.
 - 3.1 Простота (Simple design).

3.2 Метафора системи (System metaphor).

3.3 Колективне володіння кодом (Collective code ownership) або обраними шаблонами проектування (Collective patterns ownership).

3.4 Стандарт кодування (Coding standard or Coding conventions).

4. Соціальна захищеність програміста (Programmer welfare), а саме 40 годинний робочий тиждень (Sustainable pace, Forty hour week).

Архітектура клієнт-сервер є одним із архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених мережних програм і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти:

- набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;
- набір клієнтів, які використовують сервіси, що надаються серверами;
- мережа, яка забезпечує взаємодію між клієнтами та серверами.

Сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до серверів. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися то до одного сервера, то до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів.

Дуже важливо ясно уявляти, хто або що розглядається як «клієнт». Можна говорити про клієнтський комп'ютер, з якого відбувається звернення до інших комп'ютерів. Можна говорити про клієнтське та серверне програмне забезпечення. Нарешті, можна говорити про людей, які бажають за допомогою відповідного програмного та апаратного забезпечення отримати доступ до тієї чи іншої інформації.

Загальноприйнятим є положення, що клієнти та сервери – це перш за все програмні модулі. Найчастіше вони знаходяться на різних комп'ютерах, але бувають ситуації, коли обидві програми – і клієнтська, і серверна, фізично

розміщуються на одній машині; в такій ситуації сервер часто називається локальним.

Модель клієнт-серверної взаємодії визначається перш за все розподілом обов'язків між клієнтом та сервером. Логічно можна відокремити три рівні операцій:

- рівень представлення даних, який по суті являє собою інтерфейс користувача і відповідає за представлення даних користувачеві і введення від нього керуючих команд;
- прикладний рівень, який реалізує основну логіку ПЗ і на якому здійснюється необхідна обробка інформації;
- рівень управління даними, який забезпечує зберігання даних та доступ до них.

Дворівнева клієнт-серверна архітектура передбачає взаємодію двох програмних модулів – клієнтського та серверного. В залежності від того, як між ними розподіляються наведені вище функції, розрізняють:

- модель тонкого клієнта, в рамках якої вся логіка ПЗ та управління даними зосереджена на сервері. Клієнтська програма забезпечує тільки функції рівня представлення;
- модель товстого клієнта, в якій сервер тільки керує даними, а обробка інформації та інтерфейс користувача зосереджені на стороні клієнта. Товстими клієнтами часто також називають пристрої з обмеженою потужністю: кишенькові комп'ютери, мобільні телефони та ін.

Типовим прикладом клієнт-серверної взаємодії є WWW. Існує величезна кількість веб-серверів, на яких розміщується та чи інша інформація. У найпростішому випадку ця інформація являє собою набір веб-сторінок, які можуть зберігатися на сервері у вигляді файлів, розмічених за допомогою мови розмітки HTML. Але ситуація, як правило, є складнішою; значна частина веб-ресурсів на сучасному етапі є динамічними, тобто вони не існують в заздалегідь

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

підготовленому вигляді, а створюються безпосередньо в процесі обробки запиту від користувача.

Для того, щоб людина, яка працює в Інтернеті, могла переглянути ту чи іншу сторінку, на її комп'ютері повинно бути встановлено відповідне програмне забезпечення. Програми для перегляду веб-сторінок називаються браузером.

Але, крім браузерів, до серверів можуть звертатися і інші клієнти, а саме – автономні програми. Вони можуть передбачати взаємодію з людиною, а можуть працювати в цілком автоматичному режимі. Типовим класом таких програм є роботи, призначені для автоматичного перегляду веб-ресурсів. Зокрема, роботи є важливим елементом пошукових систем і використовуються ними для перегляду сторінок і збору інформації про них.

Для запиту до веб-сервера клієнтська програма повинна задати місцезнаходження комп'ютера, на якому розміщується серверна програма, назву потрібного документа і, можливо, інші дані, які специфікують запит. Мережа забезпечує знаходження сервера і передачу йому клієнтського запиту. Серверні програми обробляють цей запит, відповідь пересилається по мережі клієнтові.

Трирівнева клієнт-серверна архітектура, яка почала розвиватися з середини 90-х років, передбачає відділення прикладного рівня від управління даними. Відокремлюється окремий програмний рівень, на якому зосереджується прикладна логіка ПЗ. Програми проміжного рівня можуть функціонувати під управлінням спеціальних серверів ПЗ, але запуск таких програм може здійснюватися і під управлінням звичайного веб-сервера. Нарешті, управління даними здійснюється сервером даних.

Для роботи з системою користувач використовує стандартне програмне забезпечення –звичайний браузер. Це позбавляє його необхідності завантажувати та інстальювати спеціальні програми (хоча інколи така необхідність все-таки виникає).

Але користувачеві слід надати в розпорядженні інтерфейс, який дозволяв би йому взаємодіяти з системою і формувати запити до неї. Форми, що

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

визначають цей інтерфейс, розміщуються на веб-сторінках та завантажуються разом з ними.

Веб-оглядач формує запит та пересилає його до сервера, який здійснює обробку. При необхідності сервер викликає серверні програмні модулі, які забезпечують обробку запиту і в разі потреби звертаються до сервера даних. Сервер даних здійснює операції з даними, що зберігаються в системі та складають її інформаційну основу. Зокрема, він може здійснити вибірку з інформаційної бази відповідно до запиту та передати її модулю проміжного рівня для подальшої обробки. Дані, з якими працює сервер даних, найчастіше організовані як реляційна база даних.

Найчастіше веб-сервер і серверні модулі проміжного рівня розміщуються на одному комп'ютері, хоч і являють собою окремі і логічно незалежні програмні модулі.

На сучасному етапі для програмування модулів проміжного рівня використовується мова серверних сценаріїв PHP, а для управління даними – СУБД MySQL. Таким чином, зв'язку PHP-MySQL слід розглядати як стандартний інструмент для створення порівняно простих інтерактивних веб-сайтів та систем електронної комерції; близько 90% комерційних систем сьогодні створюється саме на цій основі. Водночас як засоби управління даними, так і middleware-засоби можуть бути найрізноманітнішими. Так, для створення серверних програм, крім PHP, широко застосовуються Java, Perl, Python, Delphi.

Взагалі, технології створення розподілених, зокрема веб-програм, стрімко розвиваються. Слід згадати про технології EJB (Enterprise Java Beans), CORBA, а також про .NET – порівняно нову ініціативу компанії Microsoft. Для зберігання даних та їх передачі часто використовується так звана розширювана мова розмітки XML (Extensible Markup Language).

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм ММВ, в основі якого лежить змішування операцій різних алгебраїчних груп. ММВ – ітеративний алгоритм, що складається з лінійних дій (XOR і використання ключа) і паралельного застосування чотирьох великих оборотних нелінійних підстановок. Ці підстановки визначаються за допомогою множення по модулю $2^{32}-1$ з постійними множниками. У підсумку з'являється алгоритм, що використовує 128-бітовий ключ і 128-бітовий блок.

Алгоритм ММВ оперує 32-бітовими підблоками тексту (x_0, x_1, x_2, x_3) і 32-бітовими підблоками ключу (k_0, k_1, k_2, k_3) . Це спрощує реалізацію алгоритму на сучасних 64-бітових процесорах. Чергуючись із операцією XOR, шість разів використовується нелінійна функція f . Запишемо операції алгоритму (всі операції з індексами виконуються по модулю 4):

$$x_i = x_i \oplus k_i \text{ для } i = 0..3$$

$$f(x_0, x_1, x_2, x_3)$$

$$x_i = x_i \oplus k_{i+1} \text{ для } i = 0..3$$

$$f(x_0, x_1, x_2, x_3)$$

$$x_i = x_i \oplus k_{i+2} \text{ для } i = 0..3$$

$$f(x_0, x_1, x_2, x_3)$$

$$x_i = x_i \oplus k_i \text{ для } i = 0..3$$

$$f(x_0, x_1, x_2, x_3)$$

$$x_i = x_i \oplus k_{i+1} \text{ для } i = 0..3$$

$$f(x_0, x_1, x_2, x_3)$$

$$x_i = x_i \oplus k_{i+2} \text{ для } i = 0..3$$

$$f(x_0, x_1, x_2, x_3)$$

Функція f виконується в три кроки:

1. $x_i = c_i * x_i$ для $i = 0..3$ (Якщо на вході множення одні одиниці, то на виході – теж одні одиниці).

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

2. Якщо молодший значущий біт $x_0 = 1$, то $x_0 = x_0 \oplus C$. Якщо молодший значущий байт $x_3 = 0$, то $x_3 = x_3 \oplus C$.

3. $x_i = x_{i-1} \oplus x_i \oplus x_{i+1}$ для $i = 0..3$.

Всі операції з індексами виконуються по модулю 4. Операція множення на кроці 1 виконується по модулі $2^{32}-1$. Спеціальний випадок для даного алгоритму: якщо другий операнд дорівнює $2^{32}-1$, результат теж дорівнює $2^{32}-1$. В алгоритмі використовуються наступні константи:

$$C = 2\text{aaaaaaa}, c_0 = 025\text{f1cdb}, c_1 = 2 * c_0, c_2 = 2^3 * c_0, c_3 = 2^7 * c_0.$$

Константа C – «найпростіша» константа без кругової симетрії, високою трійковою вагою й нульовим молодшим значущим бітом. У константи c_0 є інші особливі характеристики. Константи c_1 , c_2 і c_3 – зрушені версії c_0 , і служать для запобігання атак, заснованих на симетрії.

Розшифрування виконується у зворотному порядку, Етапи 2 і 3 інверсні їм самим. На етапі 1 замість c_i використовується c_i^{-1} . Значення $c_0^{-1} = 0\text{dad4694}$.

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Програма має простий та інтуїтивно зрозумілий інтерфейс, який зображений на рисунку 5.1. З рисунку головного вікна можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

- Функціональних кнопок ПЗ: Вибрати; Перевірити; Створити; Робота з ключом; Робота зі стегоданими; Збереження даних.
- Верхнього меню: Файл; Стегокодер/декодер; Ключі; Параметри; Довідка.
- Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші: Обробка даних; Налаштування; Довідка; Журнал роботи.
- Розділу виведення результату роботи системи с статистичними даними.

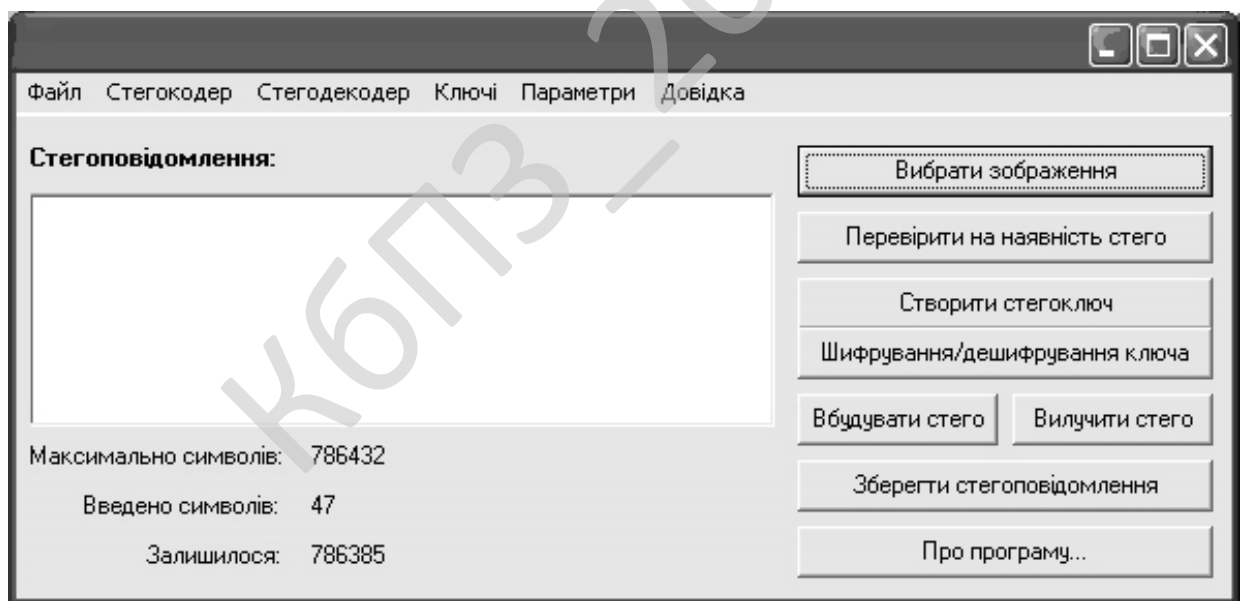


Рисунок 5.1 – Головне вікно ПЗ

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення. Розроблена програма має дуже простий і зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним

середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий. Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

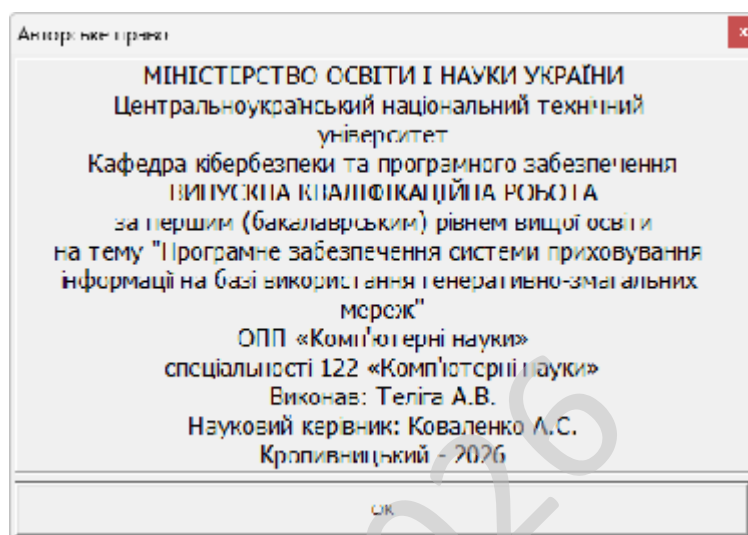


Рисунок 5.2 – Авторське право

Процес впровадження програмного забезпечення, це процес налаштування програмного забезпечення під певні умови використання, а також навчання користувачів роботі з програмним продуктом. Впровадження програмного забезпечення це усі дії, що роблять розроблену програмну систему готовою до використання. Даний процес є частинною життєвого циклу програмного забезпечення.

Таким чином у результаті вищерозглянутого можна стверджувати що розроблено інтерфейс системи у відповідності з вибраною метою роботи. Система містить максимальний необхідний набір функцій придатних для виконання будь-яких дій для забезпечення повноцінної роботи програми. Далі розглянемо висновки та використані літературні джерела.

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи приховування інформації на базі використання генеративно-змагальних мереж.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

Рішення завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем приховування інформації на базі використання генеративно-змагальних мереж.
- Досліджена система приховування інформації на базі використання генеративно-змагальних мереж.
- На основі отриманих результатів досліджень створена програмна реалізація системи приховування інформації на базі використання генеративно-змагальних мереж.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання приховування інформації на базі використання генеративно-змагальних мереж.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Python. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

системи приховування інформації на базі використання генеративно-змагальних мереж. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм ММВ.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Foreman J.W. Data Smart: Using Data Science to Transform Information into Insight 1st Edition. – Wiley, 2013. – 432 p.
2. Hurbans R. Grokking Artificial Intelligence Algorithms. – Manning, 2020. – 631 p.
3. Gusfield D. Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology 1st Edition. – Cambridge University Press, 2008. – 556 p.
4. Kotu V., Deshpande B. Data Science: Concepts and Practice. – Elsevier Science, 2018. – 953 p.
5. Knowledge Base A Complete Guide - 2021 Edition // The Art of Service - Knowledge Base Publishing, 2020. – 306 p.
6. Knuth D. The Art of Computer Programming, Vol. 1: Fundamental Algorithms, 3rd Edition 3rd Edition. – Addison-Wesley Professional, 2019. – 672 p.
7. Mattmann C. Machine Learning with TensorFlow, Second Edition. – Manning, 2020. – 1124 p.
8. Mueller J.P., Massaron L. Machine Learning For Dummies. – Wiley, 2016. – 714 p.
9. Teofili T. Deep Learning for Search. – Manning, 2019. – 695 p.
10. Rungta K. TensorFlow in 1 Day: Make your own Neural Network. – Publishdrive, 2019. – 587 p.
11. Weidman S. Deep Learning from Scratch: Building with Python from First Principles. – O'Reilly. 2019. – 252 p.
12. Rajasekaran S., Vijayalakshmi Pai G.A. Neural networks, fuzzy logic, and genetic algorithms: synthesis and applications (with cd-rom) Kindle Edition. – PHI, 2013. – 628 p.
13. Вінтенко Б.Ю., Миронець І.В., Смірнов О.А. «Модель комп'ютерно-орієнтованої процедури системи підтримки оперативного

персоналу АЕС». *IV Міжнародна науково-практична Інтернет-конференція «Інновації та перспективні шляхи розвитку інформаційних технологій (ІПШРІТ-2025)»* м.Черкаси 25 листопада 2025 року – Черкаси: ЧДТУ.– 2025. – С.101-103.

14. Kuznetsov O., Frontoni E., Kuznetsova Y., Chevardin V., Smirnov O. «Architectural foundations for adaptive security in edge computing systems». *Cybersecurity Defensive Walls in Edge Computing*, 2025. pp. 21-61.

15. Вінтенко, Б.Ю., Миронець, І.В., Смірнов, О.А., Коваленко, О.В., Усік, П.С., Буравченко, К.О., Лисенко, І.А. «Логіко-структурна модель комп'ютерно-орієнтованої процедури системи підтримки оперативного персоналу АЕС». *Кібербезпека: освіта, наука, техніка*. 2025. Том 2 № 30. С. 413-427, 2025.

16. Смірнова, Т.В. «Дослідження методів, моделей та сучасних ІТ-рішень для підтримки технологічних процесів у критичній інфраструктурі держави». *Кібербезпека: освіта, наука, техніка*. 2025. Том 2 № 30. С.195-208, 2025.

17. Усік, П.С., Смірнова, Т.В., Буравченко, К.О., Смірнов, О.А., Улічев, О.С., Смірнов, С.А. «Дослідження технологій забезпечення кібербезпеки банківських систем з використанням штучного інтелекту». *Кібербезпека: освіта, наука, техніка*. 2025. Том 1 № 29. С.704–716, 2025

18. Kuznetsov, O., Frontoni, E., Kuznetsova, K., Arnesano, M., Smirnov, O. «A secure biometric authentication architecture for blockchain-driven cyber-physical systems». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 193–224.

19. Kuznetsov, O., Smirnov, O., Akhmetov, B., Alimseitova, Z., Imoize, A.L. «Deep Learning Frontiers in Copy-Move Forgery Detection: Advances, Challenges, and Future Directions». *Advancements in Cybersecurity Next Generation Systems and Applications*, 2025. 202-229.

20. Вінтенко Б., Смірнов О., Миронець І., Смірнова Т., Смірнов С. «Імітаційна модель шляхів вхідних даних комп'ютерної інтелектуальної системи підтримки оператора енергоблоку АЕС». *Комбінаторні конфігурації та їхні*

застосування: Матеріали XXVII Міжнародного науково-практичного семінару, присвяченого 125-річчю Національного університету «Запорізька політехніка» (Запоріжжя-Кропивницький-Київ, 4-6 червня 2025 р.). Запоріжжя: НУ «Запорізька політехніка», 2025. С.82-91.

21. Al-Azzeh, J., Ayyoub, B., Mesleh, A., Smirnova, T., Gnatyuk, S., Drieiev, O., Smirnov, O., Dorenskyi, O. «Cloud-Based Information System for Evaluating Caverns in the Process of Blasting Metal Surfaces of Details». *International Review on Modelling and Simulations* 18 (1), 2025. pp. 32-42.

22. Вінтенко Б.Ю., Смірнов О.А., Миронець І.В., Смірнова Т.В., Коваленко О.В., Мацуї А.М. «Модель шляхів отримання вхідних даних комп'ютерної інтелектуальної системи підтримки оперативного персоналу АЕС». *Центральноукраїнський науковий вісник. Технічні науки*. 2025. Вип. 11(42), ч. II. С.52-62.

23. Вінтенко Б.Ю., Смірнов О.А., Миронець І.В., Смірнова Т.В. «Методи забезпечення відмовостійкості інтелектуальних систем підтримки оператора». *VIII міжнародна науково-практична конференція “Інформаційна безпека та комп'ютерні технології”*, м. Кропивницький. 24-25 квітня 2025 р. – Кропивницький: ЦНТУ. – 2025. – С. 44-46.

24. Смірнов, О.А., Константинова, Л.В., Коноплицька-Слободенюк, О.К., Козірова, Н.В, Якименко, Н.М., Доренський, О.П., Буравченко, К.О. «Дослідження інструментів штучного інтелекту для роботи з базами даних та аналізу даних». *Кібербезпека: освіта, наука, техніка*. 2025. №3(27), С. 429–448.

25. Lakhno, V., Malyukov, V., Smirnov, O., Bebeshko, B., Chubaievskiy, V., Zhumadilova, M., Malyukova, I., Smirnov, S. «Multifactorial Model for Targeted Attacks Counteracting Within the Framework of a Multi-Step Quality Game with Fuzzy Information». *8th International Symposium on Intelligent Informatics, ISI 2023*, 2025. vol 389. pp 377-389. Springer, Singapore.

26. Smirnov O., Fedorov E., Neskorodieva A., Neskorodieva T. «Intellectual Classification method of Gymnastic Elements Based on Combinations of

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

Descriptive and Generative Approache». *CEUR Workshop Proceedings* Volume 3664, 2024, Pages 11-23.

27. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.

28. Malyukov V., Bebeshko B., Lakhno V., Smirnov O., Malyukova I., Mohylnyi H. «Managing the Purchase-Sale Process of Digital Currencies Under Fuzzy Conditions». *Lecture Notes in Networks and Systems*, 2023, 729 LNNS, pp. 104–112.

29. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.

30. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchев, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.

31. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи*, 2023, том 7, № 2, С. 49-56.

32. Smirnov, O., Karapetyan, A., Fedorov, E., «Creating Neural Network and Single Solution Human-Based Metaheuristic Methods of Solving the Traveling Salesman Problem». *CEUR Workshop Proceedings*, Volume 3312, 2022, pp. 47-58.

33. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» *CEUR Workshop Proceedings*, Volume 3187, 2022, pp. 1-12.

34. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». *Sensors (Basel, Switzerland)* Volume 22, Issue 16, 6223, 2022.

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

35. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebesko, B. «Selection of a Rational Composition of Information Protection Means Using a Genetic Algorithm». In: Rajakumar, G., Du, KL., Vuppapalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies, vol 131. 2023. Springer, Singapore. pp. 21-34.

36. Kuznetsov, A., Oleshko, I., Chernov, K., Bagmut, M., Smirnova, T. «Biometric authentication using convolutional neural networks». Lecture Notes in Networks and Systems. Volume 152, 2021, Pages 85-98.

37. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. <https://doi.org/10.1007/s42979-021-00739-w>.

38. Smirnov O., Neskorodieva T., Fedorov E., Rymar P. «Neural Network Modeling Method of Transformations Data of Audit Production with Returnable Waste». CEUR Workshop Proceedings Volume 3101, 2021, Pages 192-207.

39. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

40. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114.

41. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

42. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In:

Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587.

43. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379.

44. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645.

45. Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019; Odessa; Ukraine; 9-13 September 2019. P.22-28.

46. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

47. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

48. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process». CEUR Workshop Proceedings, Vol 2588, P. 215-227, 2019.

49. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019.

50. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.

51. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 353-358.

К6ПЗ-2026

					ВКРБ-122.26.0030.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59