

Шмідт Валерій,
здобувач другого (магістерського) рівня вищої освіти
фізико-математичного факультету
Науковий керівник: **Постова Світлана,**
кандидат педагогічних наук, доцент,
доцент кафедри комп'ютерних наук та інформаційних технологій,
Житомирський державний університет імені Івана Франка
м. Житомир, Україна

ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРИХОВАНОГО КОДУВАННЯ В QR CODE

У сучасному світі обмін інформацією набув ключової ролі в повсякденному житті. Сучасні інформаційні технології забезпечують швидку та зручну передачу даних між різними куточками планети, але разом з тим зростають ризики втрати, перехоплення або пошкодження інформації під час передачі. Тому безпека інформації є однією з найактуальніших проблем сьогодення.

Ефективним методом захисту даних при передачі та зберіганні є їх кодування, яке дозволяє захистити інформацію від несанкціонованого доступу та забезпечити її цілісність. Особливу важливість має використання QR-кодів,

Секція 4. Технології розробки інформаційних систем

які сьогодні є дуже поширеними в логістиці, маркетингу, медицині та інших сферах. Ці матричні коди здатні містити великий обсяг інформації та використовуються для швидкого і надійного обміну даними.

Однак, процес кодування даних у форматі QR-коду має низку викликів. Основним є питання безпеки інформації, адже неавторизований доступ може призвести до втрати даних чи їхнього несанкціонованого використання. Крім того, QR-коди часто використовуються у друкованому вигляді, і це означає, що важливо забезпечити збереження даних навіть при фізичних пошкодженнях, таких як подряпини, потертості чи дефекти друку.

Для розробки програмного забезпечення для кодування даних у QR-коді було обрано архітектурний патерн MVVM для графічної підсистеми WPF [1; 2].

Архітектура MVVM (Model-View-ViewModel) є патерном проектування програмного забезпечення, що застосовується для створення додатків із графічним інтерфейсом користувача. Основною метою цього патерну є відокремлення бізнес-логіки додатка від її відображення в інтерфейсі.

Model – компонент, що відповідає за бізнес-логіку програми та взаємодію з джерелами даних, як-от бази даних, веб-сервіси чи файли. Model не залежить від інших компонентів програми, зокрема від View та ViewModel.

View – компонент, відповідальний за показ даних користувачу та обробку його дій, наприклад, натискань кнопок і введення тексту. View пов'язаний із ViewModel.

ViewModel – компонент, який забезпечує зв'язок між Model і View. Він отримує дані з Model і перетворює їх у формат, зручний для відображення у View. Також ViewModel обробляє дії користувача в View і передає їх у Model.

Схематичне зображення архітектури MVVM наведено на рисунку 1.

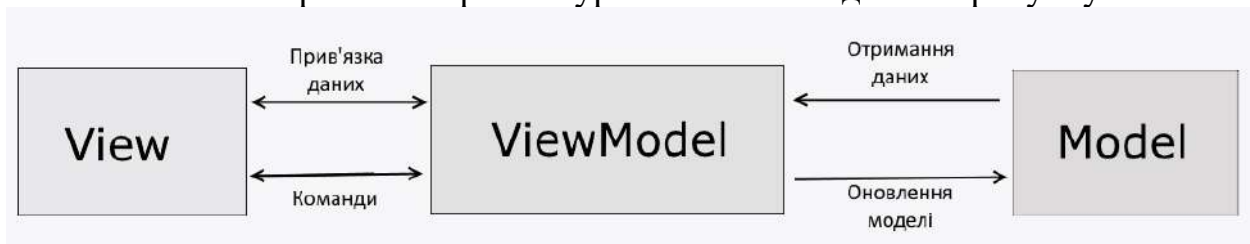


Рис. 1. Схематичне зображення архітектури MVVM

Відповідно до вимог до алгоритму кодування даних у QR-коді, було вдосконалено алгоритм для підвищення захищеності прихованих даних [7].

Вдосконалений алгоритм включає етап додавання ЕСС (корекції помилок) до прихованих даних під час кодування, а також етап декодування ЕСС [17] для відновлення даних при їх екстракції.

На рисунку 2 зображено схему роботи вдосконаленого алгоритму кодування даних у QR-коді для етапів кодування та вилучення прихованих даних відповідно.

Секція 4. Технології розробки інформаційних систем

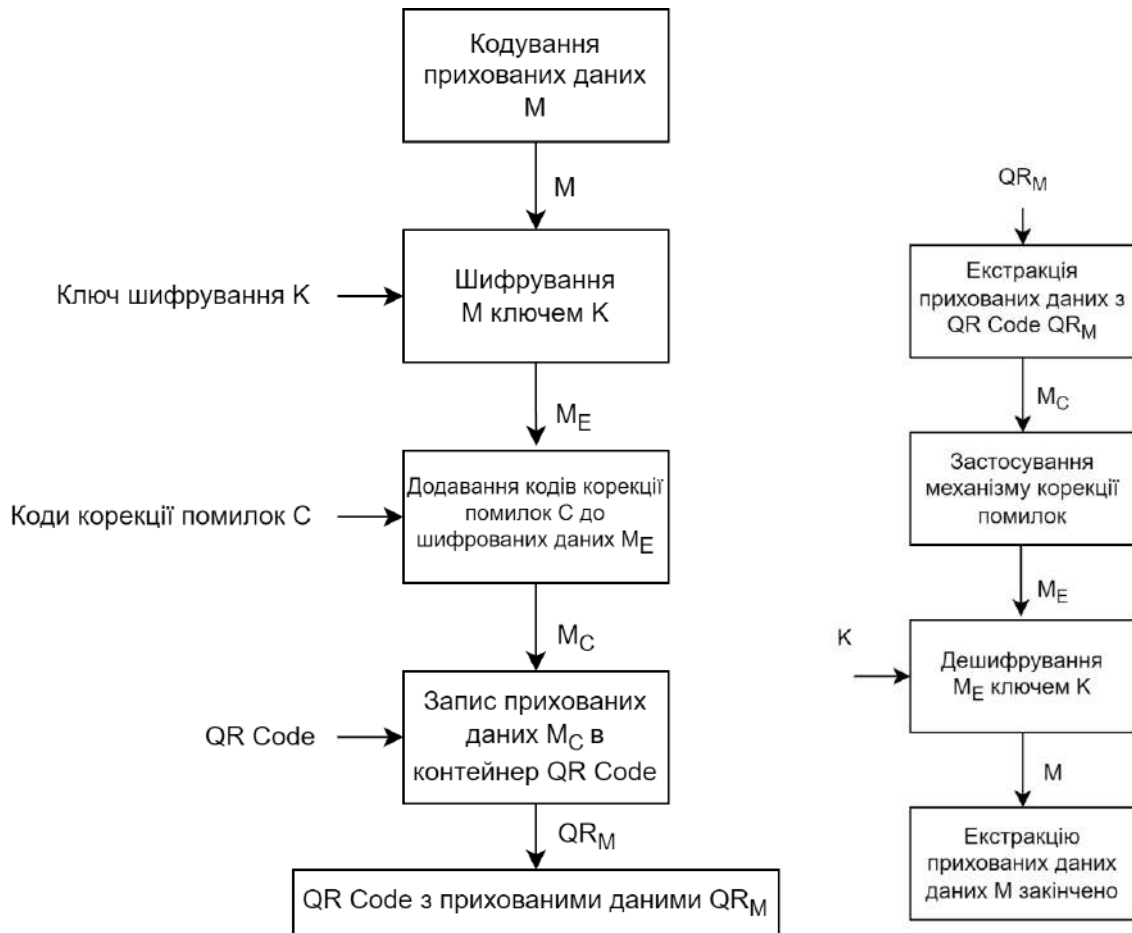


Рис. 2. Схема роботи покращеного алгоритму кодування даних в контейнері QR Code; етапи кодування прихованих даних та екстракції прихованих даних

Для реалізації покращеного алгоритму, реалізуючи архітектурний патерн MVVM, було створено структурну схему класів, зображену на рисунку 3.

Надамо детальний опис класів програмного забезпечення для кодування даних у QR-коді. Клас MainWindow: публічний клас модуля View, який відповідає за ініціалізацію інтерфейсу головного вікна програми та здійснює прив'язку даних до модуля ViewModel. У таблиці 1 наведено детальний опис методів цього класу.

Секція 4. Технології розробки інформаційних систем

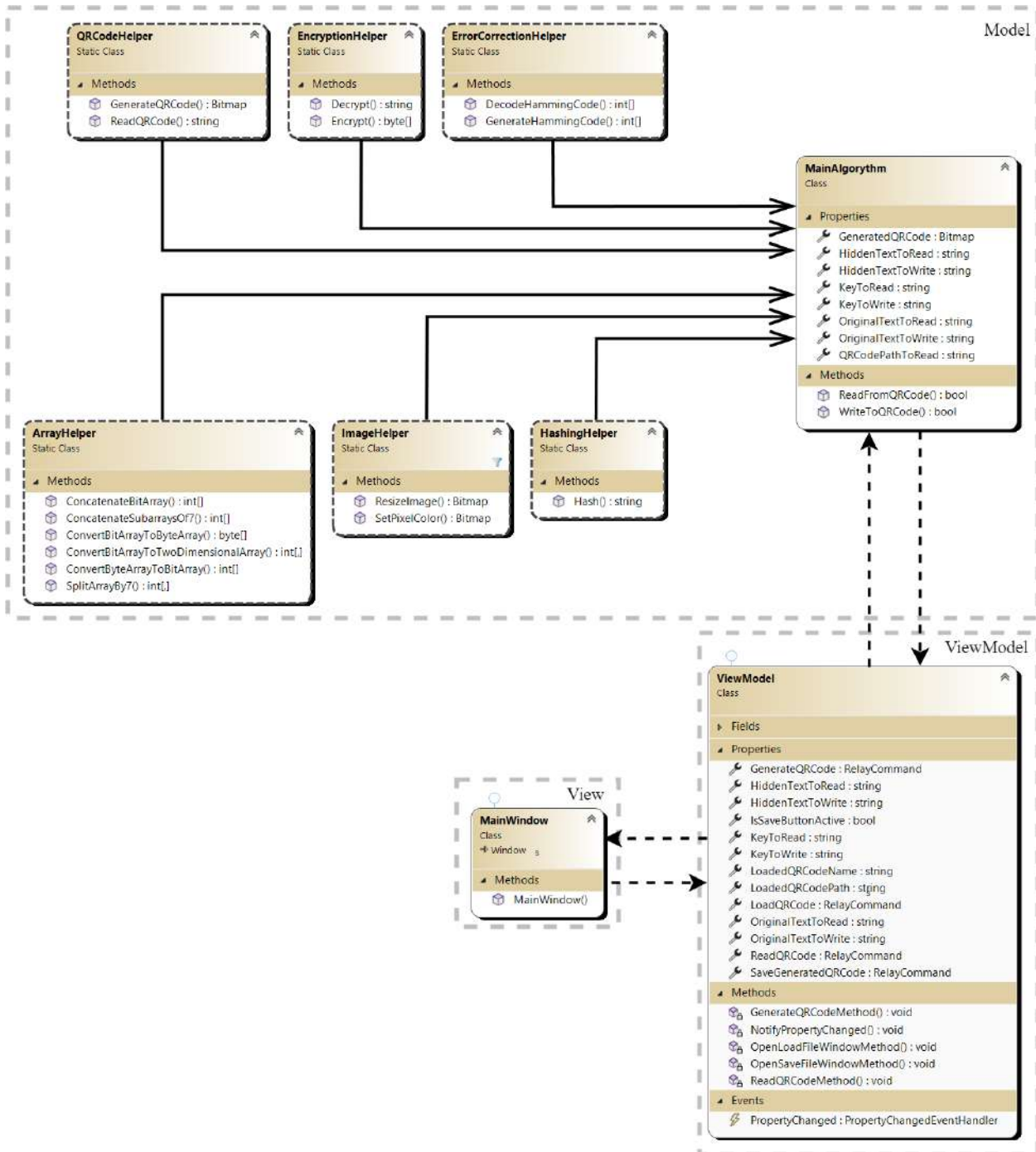


Рис. 3. Структурна схема класів

Таблиця 1

Опис методів класу MainWindow

Назва	Тип	Опис
MainWindow	Конструктор класу	Конструктор класу MainWindow. Ініціалізує представлення головного вікна програми. Проводить прив'язку даних до модуля ViewModel

Клас ViewModel: публічний клас модуля ViewModel, який забезпечує взаємодію між Model та View. Він обробляє дії користувача у View, передаючи

Секція 4. Технології розробки інформаційних систем

їх у Model, а також отримує дані з Model і передає їх у View для відображення. У таблиці 2 наведено детальний опис методів та властивостей цього класу.

Таблиця 2

Опис методів та властивостей класу ViewModel

Назва	Тип	Опис
OriginalTextToRead	string властивість	Містить текст, отриманий з QR-коду.
OriginalTextToWrite	string властивість	Містить текст, введений користувачем.
HiddenTextToRead	string властивість	Зберігає прихований текст, зчитаний з QR-коду.
KeyToRead	string властивість	Містить ключ для дешифрування, введений користувачем.
KeyToWrite	string властивість	Містить ключ для шифрування, введений користувачем.
LoadedQRCodeName	string властивість	Зберігає назву файлу, який було завантажено.
LoadedQRCodePath	string властивість	Зберігає повний шлях до завантаженого файлу.
IsSaveButtonActive	bool властивість	Визначає, чи активна кнопка 'Зберегти...!'.
GenerateQRCode-Method	private void метод	Приватний метод, що викликає метод WriteToQRCode класу MainAlgorhythm, передає введені дані та зберігає результат.
ReadQRCodeMethod	private void метод	Приватний метод, що викликає метод ReadFromQRCode класу MainAlgorhythm, передає введені дані та зберігає результат.
OpenSaveFileWindow-Method	private void метод	Приватний метод, що відкриває діалогове вікно для збереження файлу.
OpenLoadFileWindow-Method	private void метод	Приватний метод, що відкриває діалогове вікно для завантаження файлу.
NotifyPropertyChanged	private void метод	Приватний метод, що реалізує інтерфейс INotifyPropertyChanged

Секція 4. Технології розробки інформаційних систем

		для оновлення даних у представленні.
--	--	--------------------------------------

MainAlgorithm: публічний клас модуля Model, який реалізує покращений алгоритм кодування даних у QR-коді, зокрема кодування та вилучення даних. У таблиці 3 подано детальний опис методів та властивостей цього класу.

Таблиця 3

Опис методів та властивостей класу MainAlgorithym

Назва	Тип	Опис
WriteToQRCode	public bool метод	Публічний метод, що виконує алгоритм кодування даних у QR-код.
ReadFromQRCode	public bool метод	Публічний метод, що виконує алгоритм екстракції даних з QR-коду.
OriginalTextToWrite	string властивість	Зберігає основний текст, який буде закодовано в QR-код.
OriginalTextToRead	string властивість	Зберігає основний текст, який було декодовано з QR-коду.
HiddenTextToWrite	string властивість	Зберігає прихований текст, який буде закодовано в QR-код за допомогою алгоритму.
HiddenTextToRead	string властивість	Зберігає прихований текст, який було декодовано з QR-коду за допомогою алгоритму.
KeyToWrite	string властивість	Зберігає ключ для шифрування прихованих даних.
KeyToRead	string властивість	Зберігає ключ для дешифрування прихованих даних.
GeneratedQRCode	Bitmap властивість	Зберігає зображення згенерованого QR-коду у форматі Bitmap.
QRCodePathToRead	string властивість	Зберігає повний шлях до QR-коду, який необхідно декодувати.

Секція 4. Технології розробки інформаційних систем

Програмне забезпечення також включає допоміжні статичні класи модуля Model, які підтримують роботу методів класу MainAlgorithm. Нижче подано опис цих статичних класів.

QRCodeHelper – містить методи для створення та зчитування QR-коду. Клас MainAlgorithm звертається до методів цього класу під час генерації QR-коду для подальшого кодування прихованих даних і при декодуванні основних даних. У таблиці 4 наведено опис методів цього класу.

Таблиця 4

Опис методів класу QRCodeHelper

Назва	Тип	Опис
GenerateQRCode	public static Bitmap метод	Статичний метод, що забезпечує генерацію QR Code з основними даними. Повертає зображення у форматі Bitmap
ReadQRCode	public static string метод	Статичний метод, що забезпечує зчитування QR Code. Повертає строку, що містить основні дані.

EncryptionHelper – містить методи для шифрування та дешифрування прихованого тексту. Клас MainAlgorithm звертається до методів цього класу для шифрування початкових прихованих даних і їх дешифрування. Для забезпечення шифрування за алгоритмом Рейндаля використовується клас Aes з бібліотеки System.Security.Cryptography. Опис методів цього класу наведено в таблиці 5.

Таблиця 5

Опис методів класу EncryptionHelper

Назва	Тип	Опис
Encrypt	public static byte[] метод	Статичний метод, що забезпечує шифрування вхідних даних. Повертає зашифрований масив байтів
Decrypt	public static string метод	Статичний метод, що забезпечує дешифрування вхідних даних. Повертає строку, яку було дешифровано

ErrorCorrectionHelper – містить методи для додавання та декодування кодів Геммінга. Клас MainAlgorithm використовує методи цього класу для реалізації механізму корекції помилок. Опис методів цього класу наведено в таблиці 6.

Таблиця 6

Опис методів класу ErrorCorrectionHelper

Назва	Тип	Опис
-------	-----	------

Секція 4. Технології розробки інформаційних систем

GenerateHamming-Code	public static int[] метод	Статичний метод, що забезпечує додавання кодів Геммінга до вхідного масиву біт.
DecodeHamming-Code	public static int[] метод	Статичний метод, що забезпечує декодування кодів Геммінга із вхідного масиву біт

HashingHelper – містить метод для хешування. Клас MainAlgorithm використовує методи цього класу для хешування ключа шифрування.

ImageHelper – містить методи для роботи із зображеннями. Клас MainAlgorithm звертається до методів цього класу для обробки зображень, таких як зміна їх розміру або встановлення кольору пікселів у QR-коді. Опис методів цього класу наведено в таблиці 7.

Таблиця 7

Опис методів класу ImageHelper

Назва	Тип	Опис
ResizeImage	public static Bitmap метод	Статичний метод, що забезпечує зміну розміру вхідного зображення. Повертає зображення у форматі Bitmap
SetPixelColor	public static Bitmap метод	Статичний метод, що задає колір пікселя QR Code. Повертає зображення у форматі Bitmap

ArrayHelper – містить методи для роботи з масивами. Клас MainAlgorithm звертається до методів цього класу для обробки масивів даних до потрібного формату. Опис методів цього класу наведено в таблиці 8.

Таблиця 8

Опис методів класу ArrayHelper

Назва	Тип	Опис
ConcatenateBit-Array	public static int[] метод	Статичний метод, що об'єднує двовимірний масив біт у одновимірний.
ConcatenateSubArrays-Of7	public static int[] метод	Статичний метод, що об'єднує підмасиви з 7 елементів в один масив.
SplitArrayBy7	public static int[,] метод	Статичний метод, що ділить масив на підмасиви по 7 елементів.
ConvertByteArray-ToBitArray	public static int[] метод	Статичний метод, що конвертує масив байт у масив біт.

Секція 4. Технології розробки інформаційних систем

ConvertBitArray-ToByteArray	public static byte[] метод	Статичний метод, що конвертує масив біт у масив байт.
ConvertBitArrayTo-TwoDimensionalArray	public static int[,] метод	Статичний метод, що конвертує масив біт у двовимірний масив.

У даній розробці для захисту прихованих даних використовується шифрування за допомогою алгоритму AES-128, що є перевіреним і надійним методом забезпечення конфіденційності інформації. Крім цього, застосовуються коди корекції помилок Геммінга, які дозволяють виявляти та виправляти помилки, що можуть виникати під час передачі або обробки даних, підвищуючи їхню цілісність. Для додаткової безпеки та захисту від несанкціонованого доступу застосовано також методи прихованого кодування.

Наведені аспекти конструювання розробки створюють основу для подальшої практичної реалізації програмного забезпечення, що включатиме оцінку його якості та тестування для виявлення можливих удосконалень і перевірки відповідності всім заданим вимогам.

Список використаних джерел та літератури

1. QR Code bar code symbology specification. *ISO*: веб-сайт. URL: <https://www.iso.org/standard/62021.html> (дата звернення: 14.09.2024)
2. QR Steganography A Threat to New Generation Electronic Voting Systems. *Scite Press*: веб-сайт. URL: <https://www.scitepress.org/papers/2014/51204/51204.pdf> (дата звернення: 14.09.2024)