

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ім. ІГОРЯ СІКОРСЬКОГО»**

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено:

Андрушкевич М. Т.

Завідувач кафедри

_____ Олександр Коваль

« __ » _____ 2020 р.

Дипломна робота

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Інформаційні технології
моніторингу довкілля»**

спеціальності 122 «Комп'ютерні науки та інформаційні технології»

на тему: «Формування кільцевих кодів для надійності передачі даних»

Виконав (-ла):

студент (-ка) IV курсу, групи ТМ-61

Андрушкевич Марія Тарасівна _____

Керівник:

Професор кафедри АПЕПС, доктор технічних наук,

Отрох Сергій Іванович _____

Рецензент:

Доцент кафедри ТК, кандидат технічних наук,

Зенів Ірина Онуфріївна _____

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів без
відповідних посилань.

Студент (-ка) _____

Київ – 2020 року

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший рівень

Напрямок підготовки 122 Комп'ютерні науки та інформаційні технології

Спеціалізація Інформаційні технології моніторингу довкілля

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр Коваль

(підпис)

” ___ ” _____ 2020р.

ЗАВДАННЯ

на дипломну роботу студенту

Андрушкевич Марії Тарасівній

(прізвище, ім'я, по батькові)

1. Тема роботи Формування кільцевих кодів для надійності передачі даних

керівник роботи Отрох Сергій Іванович, професор, доктор технічних наук

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

затверджена наказом вищого навчального закладу від "25" травня 2020р. №
1168-с

2. Строк подання студентом роботи 15.06.2020 року

3. Вихідні дані до роботи Sublime Text 3, Javascript, HTML

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) розробити систему кодування інформації; проаналізувати проблему надійності даних; проаналізувати використання кільцевих кодів для кодування;

5. Перелік ілюстративного матеріалу

мета роботи; актуальність; кільцеві коди; приклад кільцевого коду; структури кільцевого коду; засоби розробки; інтерфейс; робота з системою; висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання ” 8 ” січня 2020 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Затвердження теми роботи	08.01.2020	
2.	Вивчення та аналіз задачі	01.03.2020	
3.	Розробка архітектури та загальної структури системи	01.04.2020	
4.	Розробка структур окремих підсистем	15-30.04.2020	
5.	Програмна реалізація системи	03.05.2020	
6.	Оформлення пояснювальної записки	10-20.05.2020	
7.	Захист програмного продукту	08.06.2020	
8.	Передзахист	08.06.2020	
9.	Захист	15.06.2020	

Студент

(підпис)

Андрушкевич М.Т.

(прізвище та ініціали,)

Керівник роботи

(підпис)

Отрох С.І.

(прізвище та ініціали,)

АНОТАЦІЯ

Метою даної роботи є створення системи кодування інформації. Основною задачею є демонстрація використання кільцевих кодів в системах. Використовуючи формули та методи створення кільцевого коду, система має можливість кодувати будь-які дані та перевіряти достовірність інформації. Проаналізувати особливості кільцевих кодів, а саме:

- Ефективність з точки зору безпеки
- Передача в канал зв'язку образу інформації
- Передача інформації в 2,7 разів менше, в порівнянні з об'ємом вхідної.

Загальний обсяг роботи: 70 сторінок, 27 ілюстрацій та 10 бібліографічних назв.

Ключові слова: передача даних, кодування, група кільцевих кодів, кодова послідовність, вектор зсуву.

ABSTRACT

The purpose of this work is to create an information coding system. The main task is to demonstrate the use of ring codes in systems. Using formulas and methods of creating a ring code, the system has the ability to encode any data and verify the accuracy of information. Analyze the features of ring codes, namely:

- Safety efficiency;
- Transmission of an image of information to the communication channel;
- The transmission of information is 2.7 times less than the amount of input.

Total volume of work: 70 pages, 27 illustrations and 10 bibliographic titles.

Keywords: data transmission, coding, ring code group, code sequence, shift vector.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ.....	7
ВСТУП.....	8
1 ЗАДАЧА СТВОРЕННЯ СИСТЕМИ КОДУВАННЯ ІНФОРМАЦІЇ	10
1.1 Мета створення системи.....	10
1.2 Компоненти системи.....	10
1.3 Потенційні користувачі	10
2 АНАЛІЗ ПРОБЛЕМИ СТВОРЕННЯ СИСТЕМИ КОДУВАННЯ ІНФОРМАЦІЇ ..	12
2.1 Поняття системи кодування інформації	12
2.2 Принципи та функції СКІ.....	13
2.3 Структура СКІ.....	14
Висновки	14
3 КІЛЬЦЕВІ КОДИ	16
3.1 Опис поняття та структура кільцевих кодів	16
3.2 Характеристики векторів показників зсуву.....	19
3.3 Формування векторів показників зсуву кільцевих кодів з попереднім спотворенням	23
3.4 Основні властивості стандартного кільцевого коду	27
3.5 Сутність поняття сімейства кільцевого коду	29
3.6 Формування сімейства кільцевого коду типу 011100	31
3.7 Формування сімейства кільцевого коду типу 010101	33
Висновки	37
4 ЗАСОБИ РОЗРОБКИ	38
4.1 MVC.....	38
4.2 Середовище розробки Sublime Text 3	40
4.3 Мова програмування Javascript.....	43

	6
4.4 HTML.....	45
Висновки	46
5 МЕТОДИКА РОБОТИ КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ	47
Висновки	50
ВИСНОВОК.....	51
СПИСОК ДЖЕРЕЛ	52
ДОДАТОК 1	53
ДОДАТОК 2.....	55
ДОДАТОК 3.....	63

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

СКИ	— система кодування інформації.
ПЗ	— програмне забезпечення.
СОД	— система обробки даних.
ASCII	— American Standard Code for Information Interchange.
КТП	— кінцевий термінальний пристрій.
ЕОМ	— електронно-обчислювальна машина.
ПК	— персональний комп'ютер.
DTE	— Data terminal equipment.
ВПЗ	— Вектор показників зсуву.
MVC	— Модель – вигляд – контролер.

ВСТУП

Роки паперового діловодства створили своєрідну систему. Одним із численних недоліків системи є необхідність експлуатації фізичних об'єктів, які значно сповільнюють більшість взаємопов'язаних процесів (виробництво, обмін інформацією, прийняття рішень, надання послуг тощо). Побудова кращої системи в інформаційному суспільстві, тобто у віртуальному світі, має базуватися на найкращих досягненнях "паперового світу", мінімізуючи його недоліки. Такий підхід спростить та пришвидшить перехід до нового етапу розвитку електронного документообігу - електронного ведення документації. [1]

Впровадження електронних систем обміну даними у всіх секторах відкриває можливість використання великої гнучкості в обробці та зберіганні інформації, а також змушує організації чи структури працювати швидше та ефективніше. Використання цих систем обміну даними є важливим і сьогодні для її подальшого існування в цифровому світі. Ці технології також дозволяють підвищити ефективність роботи та заощадити час на щоденну діяльність. У той же час використання новітніх технологій, спрямованих на підвищення ефективності роботи, створює нові ризики, які можуть призвести до розкриття офіційної чи конфіденційної інформації.

Широке впровадження комп'ютерів у всі види діяльності, постійне збільшення їх обчислювальної потужності, використання комп'ютерних мереж різного масштабу призвели до загрози втрати конфіденційної інформації в системах обробки даних, які стали невід'ємною частиною майже кожної діяльності.

Будь-яка система, яка оголошує себе титулом "захищена", повинна включати механізм захисту, щоб: забезпечити безпеку документів, забезпечити безпечний доступ, забезпечити справжність документів, записати діяльність користувачів. Ці вимоги є основою безпеки будь-якої системи, і якщо всі ці критерії будуть дотримані, ми зможемо зберігати інформацію в повній безпеці.

Система обміну даними повинна забезпечувати не тільки передачу інформації,

але й її захист від крадіжок або змін, і повинна мати можливість її швидко відновити.

Захист інформації повинен певною мірою забезпечуватися будь-якою системою обміну даними. Організація та консолідація інформації, упорядкування обігу документів дає шанс створити кращу систему захисту.

1 ЗАДАЧА СТВОРЕННЯ СИСТЕМИ КОДУВАННЯ ІНФОРМАЦІЇ

Надійність кожної системи визначається надійністю її компонентів. А надійність елементів задається часом відмови або ймовірністю виходу з ладу протягом визначеного періоду часу. Надійність різних елементів може істотно відрізнятися. В результаті середні значення надійності та розподіл ймовірності відмов різних мережевих пристроїв можуть змінюватися в дуже широких межах. У багатьох випадках надійність та розподіл надійності визначаються емпірично. [2]

У цьому розділі розкривається характер роботи та завдання, які мали бути виконані в цьому процесі. Описано також мету, проблеми з її досягненням, вимоги до оригінального програмного продукту.

1.1 Мета створення системи

Метою даної роботи є створення системи, яка формує інформацію в кільцевий код для подальшого використання в процесі обміну даними.

Це завдання визначає наступні етапи роботи:

- підбір інформації, яка буде доступна для кодування;
- створення інтерфейсу для зручного використання;
- створення системи кодування.

Програмний продукт надає можливості для перетворення інформації в її образ. Це дозволить користувачеві надійно передавати інформацію та забезпечувати її точність. Програмний продукт повинен:

- отримувати вхідні дані;
- кодувати інформацію;
- знайти помилку та виправити її;
- декодувати інформацію.

1.2 Компоненти системи

У виконаній дипломній роботі ми можемо виділити основні частини, що забезпечують ефективність роботи системи:

- веб-інтерфейс програми;
- система кодування.

Розроблений веб-інтерфейс дозволяє користувачеві зручно працювати в системі, а решта системи кодує інформацію.

1.3 Потенційні користувачі

Система, розроблена в цій бакалаврській роботі, може бути корисною людям, яким потрібно передати конфіденційну інформацію. Система не вимагає від користувача спеціальних навичок роботи з комп'ютером, тому нею може користуватися людина, яка не використовує комп'ютери в повсякденному житті.

2 АНАЛІЗ ПРОБЛЕМИ СТВОРЕННЯ СИСТЕМИ КОДУВАННЯ

Для автоматизації роботи з даними, що відносяться до різних типів, важливим є приведення їх до єдиної форми. Для цього, як правило, використовують кодування, тобто представлення даних одного типу через дані іншого типу.

Кодування інформації - це спеціально розроблена система прийомів запису даних.

Особливостями кодування інформації є:

- подання інформації у більш компактній та зручній формі, яка потрібна під час запису даних у документ, передачі їх на комп'ютерний носій, обробки, передачі та інших операцій;
- один із ключових моментів у розробці політики інформаційної безпеки;
- підвищення захисту інформаційної інфраструктури від несанкціонованого доступу.

Суть кодування полягає в тому, що інформація, що зберігається в системі і передається по каналах зв'язку, перетворюється в закодований текст. У цій формі повідомлення також передається по незахищеному каналу зв'язку.

2.1 Система кодування інформації

Система кодування - це набір правил для генерації коду (Рисунок 2.1.1). Він використовується для заміни інформації кодами для забезпечення зручної та ефективнішої обробки. Для кожного коду встановлюється: алфавіт коду - система символів, прийнята для створення коду (цифровий, буквено-цифровий, буквено-цифровий); кодова база - кількість символів в алфавітному коді; кодові цифри -

- знаходження та виправлення помилок;
- простота кодування та декодування.

Спосіб кодування залежить від мети кодування: зменшення запису, шифрування повідомлення, простота обробки.

2.3 Структура СКІ

Практично всі системи кодування інформації, незалежно від сфери їх застосування, містять однаковий набір компонентів, що називаються типами програмного забезпечення. Система кодування має як програмне, так і технічне забезпечення. [5]

Програмне забезпечення - набір програмного забезпечення для створення та експлуатації СОД за допомогою комп'ютерної техніки. Програмне забезпечення містить основні та прикладні програмні продукти. Основні програмні засоби використовуються для автоматизації взаємодії людина-комп'ютер, організації стандартних процедур обробки даних, контролю та діагностики роботи апаратних засобів СОД. Прикладне програмне забезпечення - це сукупність програмних продуктів, призначених для автоматизованого вирішення функціональних завдань інформаційної системи. Вони можуть бути розроблені як універсальні інструменти, так і як спеціалізовані.

Технічне забезпечення - це група технічних засобів, що використовуються для управління системою обробки даних і включає пристрої, які виконують типові операції з обробки даних поза комп'ютером, а також на комп'ютерах різних видів.

Висновки

В цьому розділі розглянуто проблему безпеки передачі даних та продемонстровано систему кодування, її функції та структуру. З проведеної роботи над інформацією маємо виділити найголовніші особливості кодування інформації та

її системи. В першу чергу, це подання інформації у більш компактній та зручній формі, яка потрібна під час запису даних у документ, передачі їх на комп'ютерний носій, обробки, передачі та інших операцій, по-друге, кодування є одним із ключових моментів у розробці політики інформаційної безпеки, також підвищення захисту інформаційної інфраструктури від несанкціонованого доступу.

3 КІЛЬЦЕВІ КОДИ

Існує ціле сімейство завадостійких кодів, які забезпечують досить велику гнучкість з точки зору можливості реалізації коду з необхідною здатністю виявлення та виправлення помилок. Одним із цих сімейств є кільцеві коди. На даний час вони не є популярними і використовуються дуже рідко, хоча мають гарну характеристику і прості у використанні. Основними їх особливостями є:

- Ефективність з точки зору безпеки
- Передача в канал зв'язку образу інформації
- Передача інформації в 2,7 разів менше, в порівнянні з об'ємом вхідної.

3.1 Опис поняття та структура кільцевих кодів

Множина кільцевих кодів будується за принципом блокових циклічних кодів, рядки утворюючих матриць яких пов'язані між собою умовою циклічності. [7] Зсув елементів кодової послідовності циклічного коду здійснюється справа наліво, причому крайній лівий символ кожного разу переноситься праворуч в кінець кодової послідовності (Рисунок 3.1.1).

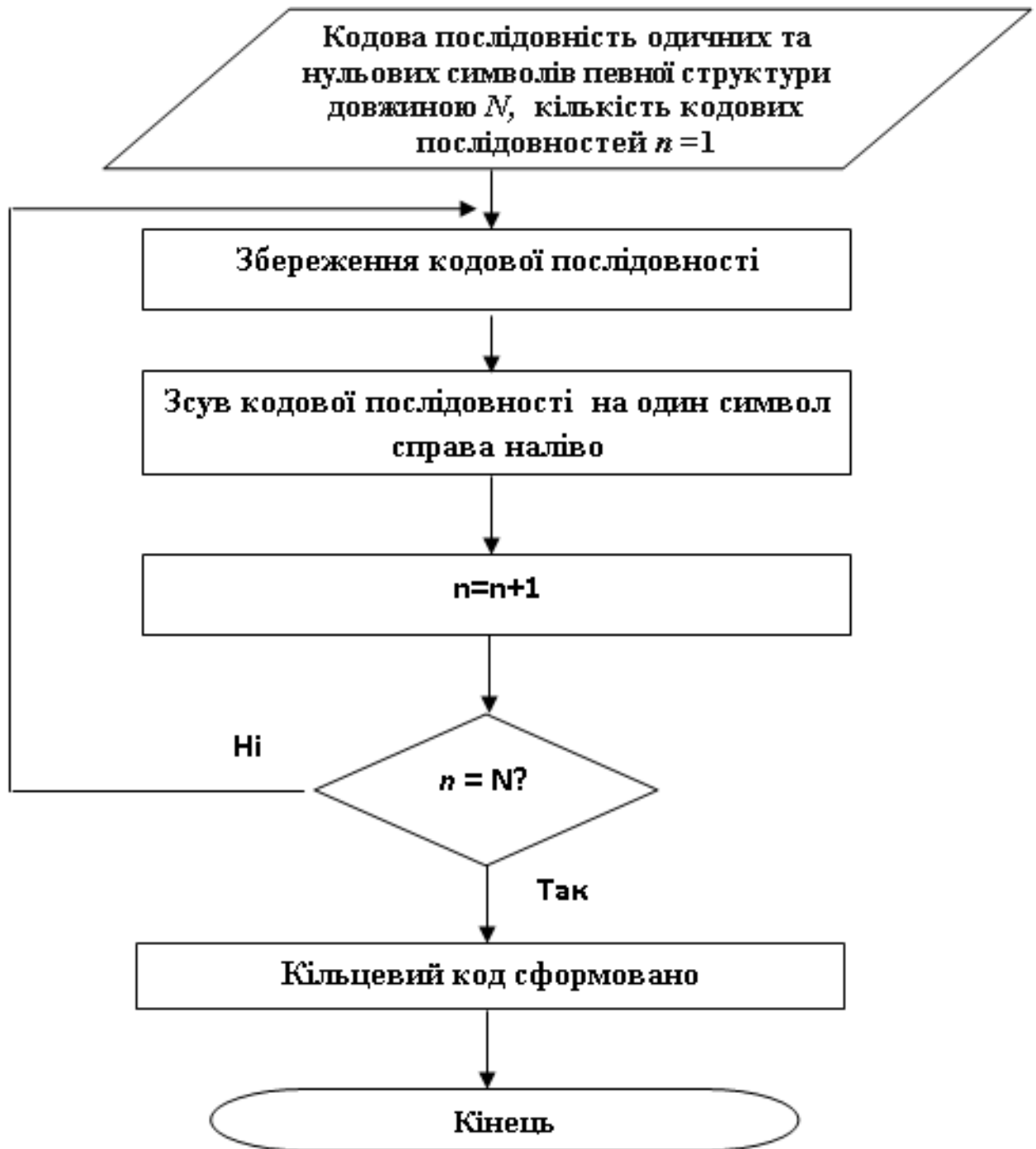


Рисунок 3.1.1 – Алгоритм утворення кільцевого коду

Кожен рядок утворюючої матриці циклічного коду має однакову кількість елементів та однакову структуру поєднань одиничних та нульових символів, проте кількість рядків та стовпців в утворюючій матриці можуть бути довільними.

Кільцевий код, на відміну від циклічного коду, завжди являє собою квадратну

матрицю розміром $N \times N$, кожний рядок якої містить m одиниць i , відповідно, $N - m$ нулів. Перший рядок утворюючої матриці кільцевого коду називають початковим вектором, або початковою послідовністю, а останній рядок є завершальним вектором циклічного зсуву елементів кодової послідовності (Рисунок 3.1.2). При цьому рядки матриці ніби утворюють кільце повного циклу зсуву елементів кодової послідовності.

$$\begin{array}{ccccccc}
 0 & 0 & 1 & 0 & 0 & 0 & 1 & \leftarrow \text{початковий вектор} \\
 1 & 0 & 0 & 1 & 0 & 0 & 0 & \\
 0 & 1 & 0 & 0 & 1 & 0 & 0 & \\
 0 & 0 & 1 & 0 & 0 & 1 & 0 & \leftarrow \text{завершальний вектор}
 \end{array}$$

Рисунок 3.1.2 – Приклад структури кільцевого коду

В загальному випадку кодова послідовність кільцевого коду (3.1.1) набуває такого вигляду:

$$C(x) = k \cdot x^{N-1} \cdots + \cdots k \cdot x^2 + k \cdot x^1 + k \cdot x^0 \quad (3.1.1)$$

де k - коефіцієнт, що набуває значення 1 або 0.

Тоді утворююча матриця кільцевого коду розміром $N \times N$ в загальному виді набуває такого вигляду (Рисунок 3.1.3):

$$C = \begin{bmatrix} k \cdot x_1^{N-1} \dots + \dots k \cdot x_1^2 + k \cdot x_1^1 + k \cdot x_1^0 \\ k \cdot x_2^{N-1} \dots + \dots k \cdot x_2^2 + k \cdot x_2^1 + k \cdot x_2^0 \\ \vdots \\ k \cdot x_N^{N-1} \dots + \dots k \cdot x_N^2 + k \cdot x_N^1 + k \cdot x_N^0 \end{bmatrix}$$

Рисунок 3.1.2 – Матриця структури кільцевого коду

Кожний рядок (кодова послідовність) кільцевого коду характеризується дельта-фактором — розподілом нульових і одиничних символів між двома крайніми одиницями, відокремленими найбільшою для даного початкового вектору кількістю нульових символів.

3.2 Характеристики векторів показників зсуву

Вектором показників зсуву називається послідовність десяткових чисел, утворених шляхом підсумовування кількості одиниць, отриманих в результаті здійснення одного з двійкових перетворень XOR, OR, AND (із запереченням Not або без нього) елементів початкової послідовності (першого рядка) кільцевого коду і решти його рядків. При цьому необхідно зазначити, що вектор показників зсуву є груповим інтегральним показником всього кільцевого коду, а не окремого його рядка.

На Рисунках 3.2.1, 3.2.2, 3.2.3 наведено структури векторів показників зсуву, отриманих за допомогою двійкових перетворень OR, XOR та AND кільцевого коду.

№ рядка кільцевого коду	Структури кодових послідовностей	Номери рядків, між елементами яких здійснено двійкове перетворення	Структури утворюючих векторів	Суми одиничних елементів утворюючих Векторів
1	0 1 0 1 0 1 1	1 - 2	1 1 1 1 1 1 1	7
2	1 0 1 0 1 1 0	1 - 3	0 1 0 1 1 1 1	5
3	0 1 0 1 1 0 1	1 - 4	1 1 1 1 0 1 1	6
4	1 0 1 1 0 1 0	1 - 5	0 1 1 1 1 1 1	6
5	0 1 1 0 1 0 1	1 - 6	1 1 0 1 0 1 1	5
6	1 1 0 1 0 1 0	1 - 7	1 1 1 1 1 1 1	7
7	1 0 1 0 1 0 1			

Рисунок 3.2.1 – Структура вектора показників зсуву, утвореного за допомогою двійкового перетворення OR.

№ рядка кільцевого коду	Структури кодових послідовностей	Номери рядків, між елементами яких здійснено двійкове перетворення	Структури утворюючих векторів	Суми одиничних елементів утворюючих векторів
1	0 1 0 1 0 1 1	1 - 2	1 1 1 1 1 0 1	6
2	1 0 1 0 1 1 0	1 - 3	0 0 0 0 1 1 0	2
3	0 1 0 1 1 0 1	1 - 4	1 1 1 0 0 0 1	4
4	1 0 1 1 0 1 0	1 - 5	0 0 1 1 1 1 0	4
5	0 1 1 0 1 0 1	1 - 6	1 0 0 0 0 0 1	2
6	1 1 0 1 0 1 0	1 - 7	1 1 1 1 1 1 0	6
7	1 0 1 0 1 0 1			

Рисунок 3.2.2 – Структура вектора показників зсуву, утвореного за допомогою двійкового перетворення XOR

№ рядка кільцевого коду	Структури кодових послідовностей	Номери рядків, між елементами яких здійснено двійкове перетворення	Структури утворюючих векторів	Суми одиничних елементів утворюючих векторів
1	0 1 0 1 0 1 1	1 - 2	0 0 0 0 0 1 0	1
2	1 0 1 0 1 1 0	1 - 3	0 1 0 1 0 0 1	3
3	0 1 0 1 1 0 1	1 - 4	0 0 0 1 0 1 0	2
4	1 0 1 1 0 1 0	1 - 5	0 1 0 0 0 0 1	2
5	0 1 1 0 1 0 1	1 - 6	0 1 0 1 0 1 0	3
6	1 1 0 1 0 1 0	1 - 7	0 0 0 0 0 0 1	1
7	1 0 1 0 1 0 1			

Рисунок 3.2.3 – Структура вектора показників зсуву, утвореного за допомогою двійкового перетворення AND

Властивості векторів показників зсуву:

- елементи будь-якого ВПЗ розміщені симетрично відносно його центру;
- сума на однакових місцях значень елементів XOR- і AND-ВПЗ дає результат, що дорівнює значенню елемента OR-ВПЗ на тому самому місці вектори показників зсуву кільцевого коду можуть бути отримані як за рядками, так і за стовпцями матриці кільцевого коду;
- вектори показників зсуву для розподілу одиниць по довжині початкового вектора, якщо цей розподіл повернено на 180° , лишаються незмінними;
- структура XOR- вектора показників зсуву лишається незмінною, якщо значення символів кодової послідовності кільцевого коду змінюються на протилежні;
- сума десяткових значень елементів ВПЗ, утворених шляхом OR-перетворення, складається з суми десяткових значень елементів ВПЗ, утворених шляхом XOR-перетворення та з суми десяткових значень елементів ВПЗ, утворених шляхом AND-перетворення. При цьому аналіз структури векторів показників зсуву та їх сумарних значень дозволяє зазначити, що незалежно від кількості елементів N та кількості одиничних символів m в кодовій послідовності існує функціональна залежність між

сумою десяткових значень елементів вектору показників та кількістю нульових та одиничних символів, яка виражається такими формулами:

- для ВПЗ, утворених шляхом двійкового XOR- перетворення (3.2.1):

$$S_{\text{ВПЗ}(XOR)} = (N - m) \cdot 2m \quad (3.2.1)$$

При цьому для визначення кількості одиничних та нульових символів в кодовій послідовності можна застосувати формулу обчислення дискримінанту та коренів квадратного рівняння (3.2.2):

$$x_{1,2} = \frac{N \pm \sqrt{N^2 - 4 \frac{S_{\text{ВПЗ}(XOR)}}{2}}}{2} \quad (3.2.2)$$

- для ВПЗ, утворених шляхом двійкового AND- перетворення (3.2.3):

$$S_{\text{ВПЗ}(AND)} = (m - 1) \cdot m \quad (3.2.3)$$

При цьому для визначення кількості одиничних та нульових символів в кодовій послідовності можна застосувати наступні прості формули:

- для ВПЗ, утворених шляхом двійкового OR- перетворення (3.2.4):

$$S_{\text{ВПЗ}(OR)} = (N - m) \cdot 2m + (m - 1) \cdot m = N \cdot 2m - 2m^2 + m^2 - m = N \cdot 2m - m^2 - m = m \cdot (2N - m - 1) \quad (3.2.4)$$

Вектори показників зсуву мають певні корисні властивості стосовно ідентифікації кільцевих кодів. Зокрема, вони дозволяють ідентифікувати кільцевий код за його довжиною та кількістю одиничних символів в кодовій послідовності, а також дозволяють визначити структуру більшості кільцевих кодів. Проте, за векторами показників зсуву неможливо ідентифікувати: прямі та інверсні кільцеві

коди, кільцеві коди з симетричними одна до одної структурами базової конфігурації кодових послідовностей та окрему кодову послідовність в межах кільцевого коду. Ці недоліки можна подолати шляхом попереднього спотворення кільцевого коду, яке реалізується двома методами:

- за допомогою заміни елементів в вихідному векторі кільцевого коду;
- за допомогою зміщення елементів вихідного вектору кільцевого коду на певну кількість елементів від 1 до $N-1$ елементів вправо або вліво.

3.3 Формування векторів показників зсуву кільцевих кодів з попереднім спотворенням

Назвемо неспотворений кільцевий код класичним кільцевим кодом, кодову послідовність класичного кільцевого коду – вихідним вектором кільцевого коду, спотворений кільцевий код – похідним кільцевим кодом, спотворену кодову послідовність – похідним вектором кільцевого коду .

Формування векторів показників зсуву шляхом заміни елементів вихідного вектору кільцевого коду здійснюється за наступним алгоритмом:

- Формується вихідний вектор кільцевого коду з певною довжиною, певною кількістю одиничних символів та певною структурою розподілу одиничних символів та нульових символів по довжині.
- Формується класичний кільцевий код шляхом циклічного зміщення елементів чергової кодової послідовності на один символ вправо або вліво.
- Формується похідний спотворений вектор кодової послідовності шляхом заміни будь-якого нульового символу вихідного вектору кодової послідовності на одиничний символ.
- Для кожної кодової послідовності сформованого похідного кільцевого коду будуються вектори показників зсуву, що складаються з десяткових

елементів, отриманих шляхом підсумовування кількості одиничних символів.

- Одиничні символи утворюються в результаті здійснення математичних операцій XOR, OR та AND над попарними елементами певної кодової послідовності кільцевого коду та кожної наступної кодової послідовності в межах кільцевого коду.

Послідовність формування векторів показників зсуву шляхом зміщення елементів вихідного вектору кільцевого коду аналогічна послідовності формування векторів показників зсуву шляхом заміни елементів вихідного вектору кільцевого коду. Нижче (Рисунок 3.3.1) наведено структуру класичного кільцевого коду довжиною 7 та кількістю одиниць 3. На Рисунку 3.3.2 наведено структуру похідного кільцевого коду, отриманого з класичного кільцевого коду шляхом зміщення елементів вихідного вектора на два елементи вправо. На Рисунку 3.3.3 наведено структури векторів показників зсуву кодових послідовностей класичного та похідного кільцевих кодів.

Найменування об'єкту	Структура об'єкту
Вихідний вектор кільцевого коду	0 0 1 0 0 0 1
Класичний кільцевий код, отриманий шляхом циклічного зміщення елементів кодових послідовностей довжиною 7 та кількістю одиничних символів 2 на один символ вправо	0 0 1 0 0 0 1
	1 0 0 1 0 0 0
	0 1 0 0 1 0 0
	0 0 1 0 0 1 0
	0 0 0 1 0 0 1
	1 0 0 0 1 0 0
	0 1 0 0 0 1 0

Рисунок 3.3.1 – Структура класичного кільцевого коду

Найменування об'єкту	Структура об'єкту
Похідний вектор кільцевого коду, отриманий шляхом заміни на одиничний символ другого справа нульового символу	0 0 1 0 0 1 1
Похідний кільцевий код, отриманий шляхом заміни вихідного вектора класичного кільцевого коду похідним вектором	0 0 1 0 0 1 1 1 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 1 1 0 0 0 1 0 0 0 1 0 0 0 1 0

Рисунок 3.3.2 – Структура похідного кільцевого коду, отриманого з класичного кільцевого коду з попереднім спотворенням.

Структура векторів показників зсуву класичного кільцевого коду з застосуванням математичних операцій XOR, OR та AND			Структура векторів показників зсуву похідного кільцевого коду з застосуванням математичних операцій XOR, OR та AND		
XOR	AND	OR	XOR	AND	OR
4 4 2 2 4 4	0 0 1 1 0 0	4 4 3 3 4 4	5 5 1 3 5	0 0 2 1 0 0	5 5 3 4 5 5
4 4 2 2 4 4	0 0 1 1 0 0	4 4 3 3 4 4	5	0 0 1 1 0 0	4 4 3 3 4 5
4 4 2 2 4 4	0 0 1 1 0 0	4 4 3 3 4 4	4 4 2 2 4 5	0 0 1 2 0 0	4 4 3 2 5 4
4 4 2 2 4 4	0 0 1 1 0 0	4 4 3 3 4 4	4 4 2 0 5 4	0 0 0 2 0 0	4 4 4 3 4 4
4 4 2 2 4 4	0 0 1 1 0 0	4 4 3 3 4 4	4 4 4 1 4 4	0 0 1 1 0 0	4 4 4 3 4
4 4 2 2 4 4	0 0 1 1	4 4 3 3 4 4	4 4 3 2 4 4	1 0 1 1 0 0	4
	0 0		2 5 2 2 4 4		3 5 3 3 4
					4

Рисунок 3.3.3 – Структура векторів показників зсуву для кодів довжиною 7 та кількістю одиниць 2

Множина кільцевих кодів (КЦК) будується за принципом блокових циклічних кодів. Стандартний КЦК являє собою квадратну двійкову матрицю розміром $N \times N$, кожний рядок якої містить m одиниць i , відповідно $N - m$ нулів.

Перший рядок КЦК називають його початковим вектором або початковою послідовністю. Кожний наступний рядок КЦК містить ті самі елементи, що й попередній, але розміщення їх змінюється внаслідок послідовного колового зсуву на один розряд праворуч або ліворуч. Будь-який рядок КЦК може виступати кодовим словом при передаванні дискретної інформації.

Приклад стандартного КЦК з параметрами $N = 13$ і $m = 5$ наведено на рисунку 3.3.4.

Структура векторів показників зсуву класичного кільцевого коду з застосуванням математичних операцій XOR, OR та AND			Структура векторів показників зсуву похідного кільцевого коду з застосуванням математичних операцій XOR, OR та AND		
XOR	AND	OR	XOR	AND	OR
4 6 2 2 6 4	1 0 2 2 0 1	5 6 4 4 6 5	4 0 4 6 2	1 3 1 0 2 2	5 3 5 6 4 4
4 6 2 2 6 4	1 0 2 2 0 1	5 6 4 4 6 5	2	1 0 2 2 0 1	5 6 4 4 6 5
4 6 2 2 6 4	1 0 2 2 0 1	5 6 4 4 6 5	4 6 2 2 6 4	1 0 2 2 3 1	5 6 4 4 3 5
4 6 2 2 6 4	1 0 2 2 0 1	5 6 4 4 6 5	4 6 2 2 0 4	1 0 2 1 0 1	5 6 4 5 6 5
4 6 2 2 6 4	1 0 2 2 0 1	5 6 4 4 6 5	4 6 2 4 6 4	1 0 0 2 0 1	5 6 6 4 6 5
4 6 2 2 6 4	1 0 2 2 0 1	5 6 4 4 6 5	4 6 6 2 6 4	1 2 2 2 0 1	5 4 4 4 6 5
			4 2 2 2 6 4		

Рисунок 3.3.4 – Структура векторів показників зсуву для кодів довжиною 7 та кількістю одиниць 3.

Головна характеристика будь-якого КЦК – це його вектор показників зсуву, який дістаємо за допомогою одного з двійкових перетворень XOR, OR, AND (із запереченням Not або без нього) початкової послідовності (першого рядка) КЦК і

решти його рядків. Отже, з однієї «заготівки» кільцевого коду загалом може утворитись шість різновидів вектора показників зсуву, що дуже важливо для ідентифікації і виправлення помилок окремих рядків (кодових слів). При цьому необхідно зазначити, що вектор показників зсуву є груповим інтегральним показником всього кільцевого коду, а не окремого його рядка.

3.4 Основні властивості стандартного кільцевого коду

Характерні ознаки ВПЗ стандартних кільцевих кодів наступні:

- Наявність центральної симетрії елементів векторів показників зсуву.
- Сума на однакових місцях значень елементів *XOR* і *AND*-ВПЗ дає результат, що дорівнює значенню елемента *OR*-ВПЗ на тому ж самому місці.
- Вектори показників зсуву кільцевого коду можуть бути отримані як за рядками, так і за стовпцями матриці КЦК.
- Сума елементів ВПЗ для всіх КЦК з однаковими t є постійною.
- Вектори показників зсуву для розподілу одиниць за довжиною початкового вектора, якщо цей розподіл повернено на 180 градусів, залишаються незмінними.
- Вигляд *XOR*-вектору показників зсуву залишається незмінним, якщо значення символів матриці КЦК змінюються на протилежні. *AND* і *OR*- вектори показників зсуву такої властивості не мають. Приклад такого кільцевого коду представлено на рисунку 3.4.1.
- Єдиний функціональний зв'язок між розподілом одиниць в початковому векторі a , отже, і у всьому кільцевому коді і видом його вектора показників зсуву відсутній, за винятком КЦК особливого вигляду.

Вихідний вектор кільцевого коду	0 1 0 0 0 1 0 0 1 0 1
Вихідний вектор кільцевого коду після зсуву на 2 розряди	0 1 0 1 0 0 0 1 0 0 1
XOR-вектор показників зсуву	6 4 6 6 4 6 4 8 0 8
AND-вектор показників зсуву	1 2 1 1 2 1 2 0 4 0
OR-вектор показників зсуву	7 6 7 7 6 7 6 8 4 8
ВРЗ класичного кільцевого коду	8 4 6 4 6 6 4 6 4 8

Рисунок 3.4.1 – Кільцевий код з параметрами $N=11$, $m=4$ після зсуву вихідного вектора на 2 розряди вправо

Також є інші особливі властивості стандартних КЦК. Головним недоліком стандартного кільцевого коду є мала придатність ВРЗ для ідентифікації його окремих рядків. Для усунення цього недоліку застосовуються особливі кільцеві коди з передспотворенням.

Об'єктами передспотворення спеціальних КЦК можуть бути:

- Початковий вектор кільцевого коду, в якому один або декілька символів змінюють своє значення на протилежне.
- Будь-який рядок кільцевого коду, спотворення якого здійснюються за тим же принципом, як і для початкового вектора.
- Початковий вектор і будь-який рядок кільцевого коду, що спотворюються одночасно.
- Початковий вектор і будь-який рядок кільцевого коду, що попередньо зміщуються на один або декілька розрядів вліво або вправо. При цьому інші рядки кільцевого коду передспотворенню не піддаються.

3.5 Сутність поняття сімейства кільцевого коду

Кожний рядок (кодова послідовність) кільцевого коду характеризується дельта-фактором — розподілом нульових і одиничних символів між двома крайніми одиницями, відокремленими найбільшою для даного початкового вектора кількістю нульових символів. Кільцеві коди, що мають дельта-фактор певного типу, утворюють сімейство кільцевих кодів. Проаналізуємо властивості векторів показників зсуву для сімейств кільцевих кодів на основі дельта-факторів типів 0011100 (одиничні символи в кодовій послідовності розміщено підряд) та 1010100 (одиничні та нульові символи чергуються між собою). Побудуємо математичну модель формування сімейства кільцевих кодів типу 0011100. На Рисунку 3.5.1 наведено характеристики сімейства кільцевих кодів на основі дельта-фактору 0011100 довжиною N (7, 8 та 9) та кількістю одиничних символів від 1 до $N - 1$. На Рисунку 3.5.2 наведено характеристики сімейства кільцевих кодів на основі дельта-фактору 1010100 довжиною N (від 7 до 12) і кількістю одиничних символів від 1 до $N/2$ для парної кількості елементів та кількістю одиничних символів від 1 до $(N+1)/2$ для непарної кількості елементів.

Довжина коду	Кількість одиничних символів	Структура кільцевого коду	Вектор показників зсуву (ВПЗ)	Кількість кодових послідовностей з певним вектором показників зсуву
Структура дельта фактора - усі одиничні символи розміщено підряд				
	1	0000001	222222	7
8	2	00000011	2444442	8
	3	00000111	2466642	8
	4	00001111	2468642	8
	5	00011111	2466642	8
	6	00111111	2444442	8
	7	01111111	2222222	8
	1	000000001	22222222	9
9	2	000000011	24444442	9
	3	000000111	24666642	9
	4	000001111	24688642	9
	5	000011111	24688642	9
	6	000111111	24666642	9
	7	001111111	24444442	9
	8	011111111	22222222	9

Рисунок 3.5.1 – Характеристики сімейства кільцевих кодів на основі дельта-фактору типу 0011100

Довжина коду	Кількість одиничних символів	Структура кільцевого коду	Вектор показників зсуву (ВПЗ)	Кількість кодових послідовностей з певним вектором показників зсуву
Структура дельта фактора - одиничні та нульові символи чергуються між собою				
7	2	0000101	424424	7
	3	0010101	624426	7
	4	1010101	624426	7
8	2	00000101	4244424	8
	3	00010101	6262626	8
	4	01010101	8080808	8
9	2	000000101	42444424	9
	3	000010101	62644626	9
	4	001010101	82644628	9
	5	101010101	82644628	9
10	2	0000000101	424444424	10
	3	0000010101	626464626	10
	4	0001010101	82828828	10
	5	0101010101	10 0 10 0 10 0 10 0 10	10
11	2	00000000101	4244444424	11
	3	00000010101	6264664626	11
	4	00001010101	8284664828	11
	5	00101010101	102846648210	11
	6	10101010101	102846648210	11
12	2	000000000101	42444444424	12
	3	000000010101	62646664626	12
	4	000001010101	82848484828	12
	5	000101010101	10 2 10 2 10 2 10 2 10 2 10	12
	6	010101010101	12 0 12 0 12 0 12 0 12	12

Рисунок 3.5.2 – Характеристики сімейства кільцевих кодів на основі дельта-фактору типу 1010100

3.6 Формування сімейства кільцевого коду типу 011100

Аналіз десяткових значень елементів кодових послідовностей певної довжини та певної кількості одиничних символів дозволяє зазначити, що загальна сукупність елементів кодових послідовностей складається з двох сукупностей, які назовемо $S1$ та $S2$. Між десятковими значеннями суміжних елементів сукупності $S1$ кодових послідовностей спостерігається лінійна залежність, яка полягає в тому, що десяткове значення наступного елемента в два рази більше десяткового значення попереднього елемента кодової послідовності і кількість таких елементів становить $N - (m-1)$, де N -

кількість елементів кодової послідовності, а m - кількість одиничних символів в кодовій послідовності. Кількість елементів сукупності S_2 кодової послідовності становить $m-1$ і різниця між десятковими значеннями суміжних елементів цієї сукупності послідовно зменшується в два рази. Більш ретельний аналіз десяткових значень сімейства кільцевих кодів типу 0011100 дозволив побудувати математичну модель формування кодових послідовностей певної довжини та певної кількості одиничних символів, яка описується таким математичним виразом (3.6.1):

$$C_k(N, m) \rightarrow S_1 \cup S_2 \quad (3.6.1)$$

В результаті аналізу властивостей сімейства кільцевих кодів типу 0011100 виявлено такі закономірності:

- максимальна кількість одиничних символів в кільцевих кодах певної довжини дорівнює $N-1$;
- кількість кільцевих кодів однакової довжини дорівнює $N-1$;
- кількість різновидів векторів показників зсуву для кільцевих кодів однакової довжини, що мають непарну кількість елементів, дорівнює $(N-1)/2$;
- кількість різновидів векторів показників зсуву для кільцевих кодів однакової довжини, що мають парну кількість елементів, дорівнює $N/2$;
- сімейство кільцевих кодів однакової довжини має симетричну послідовність розташування векторів показників зсуву, що пояснюється однаковою структурою векторів показників зсуву для прямих та інверсних кільцевих кодів;
- максимальне значення компоненту вектору показників зсуву дорівнює:
 - для прямих кільцевих кодів - числу одиничних символів кодової послідовності кільцевого коду, помноженому на два;
 - для інверсних кільцевих кодів - числу нульових символів кодової

послідовності кільцевого коду, помноженому на два.

3.7 Формування сімейства кільцевого коду типу 010101

Проаналізуємо динаміку змінювання десяткових значень кодових послідовностей сімейства кільцевих кодів типу 010101 в залежності від кількості символів в кодовій послідовності $N=4,5, \dots,7$ при кількості одиничних символів $m = 2$ та від кількості символів $N = 6,7, \dots,10$ при $m = 3$, а також динаміку змінювання десяткових значень кодових послідовностей в залежності від кількості одиничних символів $m=2,3,\dots,5$ при $N=10$, які можна простежити на Рисунках 3.7.1, 3.7.2, 3.7.3.

Система числення	Довжи на коду N	Структура кодових послідовностей кільцевих кодів в двійковій та десяткових значеннях						
Двійкова	4	0101	1010	0101	1010			
Десяткова		5	10	5	10			
Двійкова	5	00101	01010	10100	01001	10010		
Десяткова		5	10	20	9	18		
Двійкова	6	000101	001010	010100	101000	010001	100010	
Десяткова		5	10	20	40	17	34	
Двійкова	7	0000101	0001010	0010100	0101000	1010000	0100001	1000010
Десяткова		5	10	20	40	80	33	66

Рисунок 3.7.1 – Характеристики кільцевих кодів при $m=2$ та $N=4,5,\dots,7$

Система числення	Структура кодових послідовностей кільцевих кодів в двійковій та десяткових значеннях				
	Довжина коду N				
	6	7	8	9	10
Двійкова	010101	0010101	00010101	000010101	0000010101
Десяткова	21	21	21	21	21
Двійкова	101010	0101010	00101010	000101010	0000101010
Десяткова	42	42	42	42	42
Двійкова	010101	1010100	01010100	001010100	0001010100
Десяткова	21	84	84	84	84
Двійкова	101010	0101001	10101000	010101000	0010101000
Десяткова	42	41	168	168	168
Двійкова	010101	1010010	01010001	101010000	0101010000
Десяткова	21	82	81	336	336
Двійкова	101010	0100101	10100010	010100001	1010100000
Десяткова	42	37	162	161	672
Двійкова		1001010	01000101	101000010	0101000001
Десяткова		74	69	322	321
Двійкова			10001010	010000101	1010000010
Десяткова			138	133	642
Двійкова				100001010	0100000101
Десяткова				266	261
Двійкова					1000001010
Десяткова					522

Рисунок 3.7.2 – Характеристики кільцевих кодів при $m=3$ та $N=6,7,\dots,10$

Система числення	Структура кодових послідовностей кільцевих кодів в двійковій та десяткових значеннях			
	Кількість одиничних символів m			
	2	3	4	5
Двійкова	000000101	000010101	0001010101	0101010101
Десяткова	5	21	85	341
Двійкова	000001010	0000101010	0010101010	1010101010
Десяткова	10	42	170	682
Двійкова	000010100	0001010100	0101010100	0101010101
Десяткова	20	84	340	341
Двійкова	000101000	0010101000	1010101000	1010101010
Десяткова	40	168	720	682
Двійкова	0001010000	0101010000	0101010001	0101010101
Десяткова	80	336	337	341
Двійкова	0010100000	1010100000	1010100010	1010101010
Десяткова	160	672	674	682
Двійкова	0101000000	0101000001	0101000101	0101010101
Десяткова	320	321	325	341
Двійкова	1010000000	1010000010	1010001010	1010101010
Десяткова	640	642	650	682
Двійкова	0100000001	0100000101	0100010101	0101010101
Десяткова	257	261	277	341
Двійкова	1000000010	1000001010	1000101010	1010101010
Десяткова	514	522	554	682

Рисунок 3.7.3 – Характеристики кільцевих кодів при $N=10$ та $m=2,3,\dots,5$

Аналіз сформованих множин десяткових значень кодових послідовностей кільцевих кодів дозволяє встановити такі закономірності:

- найменше десяткове значення в кожному стовпці (верхній рядок десяткової системи числення) можна обчислити за формулою (3.7.1):

$$n_1 = \sum_{m-1} 22k \quad (3.7.1)$$

- кожна сукупність десяткових значень кодових послідовностей складається з двох множин S_1 і S_2 .

Між значеннями сусідніх елементів множини S_1 простежується лінійна залежність, значення наступного елемента вдвічі більше, ніж значення попереднього, причому, кількість таких елементів дорівнює $N-2(m-1)$. Кількість елементів множини S_2 дорівнює $2(m-1)$, яка в свою чергу ділиться на групи по два елементи в кожній. Різниця між десятковим значенням першого елемента множини S_2 та десятковим значенням передостаннього елемента множини S_1 обчислюється за формулою (3.7.2):

$$r_1 = 2N - 2m - 1 \quad (3.7.2)$$

Другий елемент множини S_2 подвоюється. Різниця між десятковим значенням третього елемента множини S_2 та десятковим значенням першого елемента множини S_2 дорівнює $r_2 = 22 * r_1$.

Четвертий елемент множини S_2 подвоюється. Різниця між десятковим значенням п'ятого елемента множини S_2 та десятковим значенням третього елемента множини S_2 дорівнює $r_3 = 22 * r_2$. Порядок розрахунку наступних елементів множини S_2 здійснюється аналогічно.

Поглиблений аналіз сукупності десяткових значень сімейства кільцевих кодів типу 010101 дозволив побудувати математичну модель формування кодових послідовностей даної довжини N при даній кількості m одиничних символів.

В результаті аналізу властивостей сімейства кільцевих кодів типу 1010100 виявлено такі закономірності:

- максимальна кількість одиничних символів в кільцевих кодах певної довжини, що мають парну кількість елементів, дорівнює $N/2$;

- максимальна кількість одиничних символів в кільцевих кодах певної довжини, що мають непарну кількість елементів, дорівнює $(N+1)/2$;
- кількість кільцевих кодів однакової довжини, що мають парну кількість елементів, дорівнює $(N-1)/2$;
- кількість кільцевих кодів однакової довжини, що мають непарну кількість елементів, дорівнює $(N-2)/2$;
- кількість різновидів векторів показників зсуву для кільцевих кодів однакової довжини, що мають непарну кількість елементів, дорівнює $(N-3)/2$;
- кількість різновидів векторів показників зсуву для кільцевих кодів однакової довжини, що мають парну кількість елементів, дорівнює $(N-2)/2$;
- максимальне значення компоненту вектору показників зсуву дорівнює:
 - для прямих кільцевих кодів - числу одиничних символів кодової послідовності кільцевого коду, помноженому на два;
 - для інверсних кільцевих кодів - числу нульових символів кодової послідовності кільцевого коду, помноженому на два.

Висновки

В цьому розділі було продемонстровано математичні моделі утворення сімейств кільцевих кодів для побудови ефективного каналу передачі мережі. Проведено аналіз використання кільцевих кодів. Їх особливостями є: ефективність при виявленні та виправленні помилок, схеми кодувальних і декодувальних пристроїв надзвичайно прості та будуються на основі звичайних регістрів зсуву, виграш використання становить 2,7 рази порівняно з обсягом переданої інформації.

4 ЗАСОБИ РОЗРОБКИ

В даному розділі міститься обґрунтування архітектурного рішення при розробці системи кодування інформації. Також розділ обґрунтовує використання інтегрованого середовища, архітектурного шаблону та мови програмування.

4.1 MVC

Модель-вид-контролер (Model-view-controller, англ. MVC) - архітектурний шаблон, що використовується при проектуванні та розробці програмного забезпечення. [7]

Цей шаблон ділить систему на три взаємопов'язані частини: модель даних, вид (інтерфейс користувача) та модуль управління. Він використовується для відділення даних (моделі) від користувальницького інтерфейсу (зовнішній вигляд), щоб зміни в інтерфейсі користувача мали мінімальний вплив на дані, а зміни в моделі даних можуть бути внесені без змін у інтерфейс.

Метою цього шаблону є гнучка конструкція програмного забезпечення, яка повинна сприяти подальшим змінам або вдосконаленням програми, а також надає можливість повторного використання окремих компонентів програми. Крім того, використання цього шаблону у великих системах допомагає організувати їх структуру та робить їх більш зрозумілими, зменшуючи складність.

У архітектурному шаблоні MVC програма розділена на три окремі, але взаємопов'язані частини з розподілом функцій між компонентами. Модель відповідає за зберігання даних та надання інтерфейсу з ними. Дисплей несе відповідальність за представлення цих даних користувачеві. Контролер керує компонентами, приймає сигнали у вигляді відповіді на дії користувача (зміна положення курсору миші, натискання кнопки, введення даних у текстове поле) та передача даних у модель (Рисунок 4.1.1).

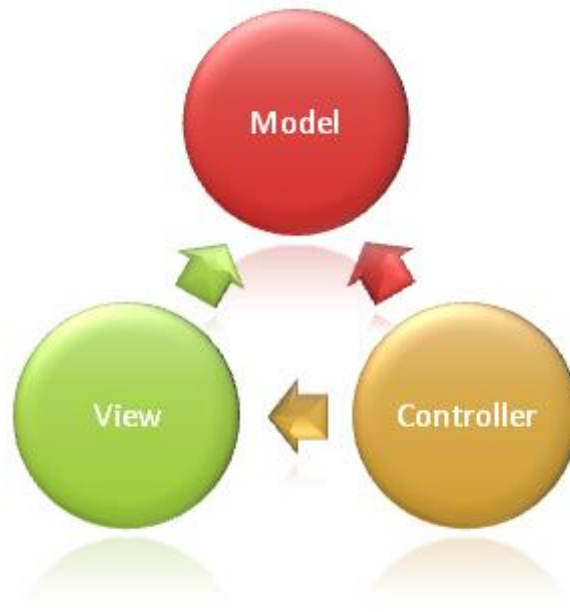


Рисунок 4.1.1 – Шаблон MVC

Модель є центральною частиною шаблону MVC і відображає поведінку застосунків, незалежних від додатків. Ця модель стосується прямого управління даними, логікою та правилами застосування.

Переглядом може бути будь-яке представлення інформації, отриманої на виході, наприклад, графік або діаграма. Тут може бути декілька типів (уявлень) однієї і тієї ж інформації одночасно, наприклад гістограма для управління компанією та таблиця для обліку.

Блок управління приймає вхідні дані та перетворює їх у команди для моделі чи дисплея.

Модель інкапсулює ядро даних та основну функціональність їх обробки та не залежить від процесу введення чи виведення даних.

Перегляд може мати декілька взаємопов'язаних областей, таких як різні таблиці та поля форми, які відображають дані.

Функція контролера полягає у відслідковуванні певних подій, що відбуваються в результаті дій користувача. Драйвер дозволяє організувати свій код, згрупувавши відповідні дії в окремий клас. Наприклад, типовий проект MVC може мати контролер користувача, який включає набір методів, пов'язаних з керуванням

обліковими записами користувачів, такі як реєстрація, авторизація, зміна профілю та зміна пароля.

Зареєстровані події переводяться на різні запити, що надсилаються компонентам моделі або об'єктам, відповідальним за показ даних. Відокремлення моделі від типу даних дозволяє використовувати різні компоненти для відображення інформації незалежно. Таким чином, якщо користувач вносить зміни в модель даних через контролер, інформація, що надається одним або кількома візуальними компонентами, автоматично регулюється відповідно до внесених змін.

Концепція MVC вперше була використана при розробці мови програмування Smalltalk як моделі для інтерфейсу користувача. Обсяг концепції також включає реалізацію рамки перегляду документів у бібліотеці MFC для Visual C ++. У сучасних технологіях концепція MVC являє собою схему JSP Model 1/2 для динамічної обробки веб-контенту на основі JSP.

4.2 Середовище розробки Sublime Text 3

Sublime Text - інтегроване середовище розробки, текстовий редактор на різних платформах. Підтримує додатки, розроблені за допомогою мови програмування Python. [8]

Sublime Text - це не безкоштовне програмне забезпечення з відкритим кодом, але деякі його додатки поширюються за безкоштовною ліцензією, розробляються та підтримуються спільнотою розробників.

Редактор містить різні візуальні теми з можливістю завантаження додаткових тем. Натиснувши на яку навігацію, користувачі побачать увесь код у правій частині екрана у вигляді міні-карти.

Sublime Text дозволяє користувачеві збирати програми та запускати їх, не перемикаючись на командний рядок. Користувач також може налаштувати свою систему компіляції та дозволити програмі автоматично збирати кожен раз, коли код зберігається.

Вінтажний режим забезпечує команди ві для використання в ST3. Вкладки, схожі на Chrome, спрощують навігацію та редагування кількох файлів.

Фрагменти коду підвищують продуктивність, дозволяючи створювати загальний код за допомогою одного ключового слова.

Особливості:

- Інтерфейс

Редактор містить різні візуальні теми з можливістю завантаження додаткових тем (Рисунок 4.2.1).

Натиснувши на яку навігацію, користувачі побачать увесь код у правій частині екрана у вигляді міні-карти.

Існує кілька режимів екрану. Одна з них містить від 1 до 4 панелей, за допомогою яких можна відображати до чотирьох файлів одночасно. Повні режими відображають лише один файл без інших меню поблизу.

- Вибір стовпців та кілька редагувань

Виберіть цілі стовпці або розмістіть на тексті кілька посилань для миттєвого редагування. Показчики поведуться так, ніби кожен був єдиним у тексті. Такі команди, як: перейти до символу, перейти до рядка, виділити текст, перейти до слова або частини слова (CamelCase, розділеної дефісом або підкресленням), перейти до початку чи кінця рядка тощо. , без використання макросів або регулярних виразів.

- Автозаповнення

Коли користувач вводить код, Sublime Text пропонує різні варіанти завершення запису залежно від мови, що використовується. Редактор також автоматично припиняє створені користувачем змінні.

- Підсвічування синтаксису та висока контрастність

Підвищений текст на темному тлі використовується для збільшення контрастності тексту. Основні елементи синтаксису виділені різними кольорами, які краще поєднуються з темним фоном, ніж зі світлим фоном.

- Підтримка систем кріплення

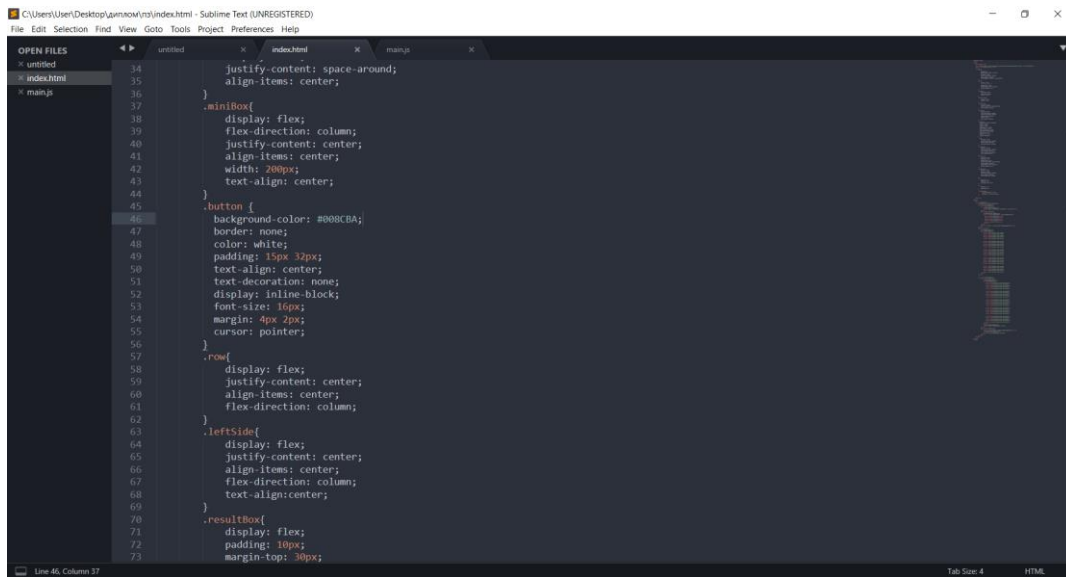


Рисунок 4.2.1 – Інтерфейс Sublime Text 3

Sublime Text дозволяє користувачеві збирати та запускати програми, не потребуючи переходу на командний рядок. Користувач також може налаштувати свою систему компіляції та дозволити програмі автоматично збирати кожен раз, коли код зберігається.

- Порожні пробіли (фрагменти)

Збережіть фрагменти коду, що часто використовується, ключові слова для їх запуску.

- Перейдіть до файлів

Навігаційний інструмент, який дозволяє користувачам переміщатися між файлами та всередині них за допомогою нечіткого пошуку.

- Додатково реалізована функція автоматичного збереження, яка допомагає користувачам не втрачати роботу.

- Налаштовані комбінації клавіш та інструмент навігації дозволяють призначити власні комбінації клавіш меню та панелям інструментів (лише перша версія, лише друга та третя палітри команд).

- Пошук телефону використовується для пошуку документа.

- Засіб перевірки синтаксису працює аналогічно і перевіряє правильність під час введення.

- Можна автоматизувати макроси та повторити останні дії.
- Змінюйте команди, включаючи відступи, переформатування абзаців та рядки злиття.

4.3 Мова програмування Javascript

JavaScript — динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки. [9]

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу (Рисунок 4.3.1).

Мова JavaScript використовується для:

- написання сценаріїв веб-сторінок для надання їм інтерактивності;
- створення односторінкових веб-застосунків (React, AngularJS, Vue.js);
- програмування на стороні сервера (Node.js);
- стаціонарних застосунків (Electron, NW.js);
- мобільних застосунків (React Native, Cordova);
- сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite чи Apache JMeter);
- всередині PDF-документів тощо.

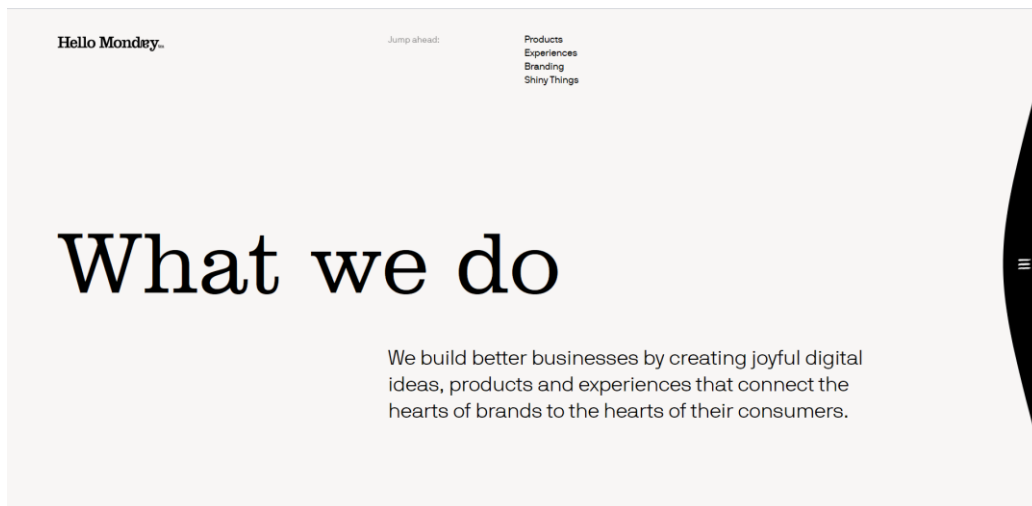


Рисунок 4.3.1 – Приклад веб-додатку мовою Javascript

Незважаючи на схожість назв, мови Java та JavaScript є двома різними мовами, що мають відмінну семантику, хоча й мають схожі риси в стандартних бібліотеках та правилах іменування. Синтаксис обох мов отриманий «у спадок» від мови C, але семантика та дизайн JavaScript є результатом впливу мов Self та Scheme (Рисунок 4.3.2).

```

window.onload = ()=>{
  document.querySelector("#btn1").addEventListener("click",function(){
    let arr = [];
    let data = document.querySelector('.inputData').value;
    arr = Array.from(data);
    console.log(arr);

    // 1st row
    for(let i = 0; i<5; i++){
      document.querySelector(`.el0${i}`).value = arr[i];
    }
    // 2
    document.querySelector('.el10').value = arr[4];
    for(let i = 0; i<4; i++){
      document.querySelector(`.el1${i+1}`).value = arr[i];
    }
  });
}

```

Рисунок 4.3.2 – Фрагмент синтаксису Javascript

JavaScript має низку властивостей об'єктно-орієнтованої мови, але завдяки концепції прототипів підтримка об'єктів в ній відрізняється від традиційних мов ООП. Крім того, JavaScript має ряд властивостей, притаманних функціональним

мовам, — функції як об'єкти першого класу, об'єкти як списки, каррінг, анонімні функції, замикання (closures) — що додає мові додаткову гнучкість.

JavaScript має C-подібний синтаксис, але в порівнянні з мовою C має такі корінні відмінності:

- об'єкти, з можливістю інтроспекції і динамічної зміни типу через механізм прототипів
- функції як об'єкти першого класу
- обробка винятків
- автоматичне приведення типів
- автоматичне збирання сміття
- анонімні функції

Оскільки JavaScript є інтерпретатором, без строгої типізації, і може виконуватися в різних середовищах, кожне зі своїми власними особливостями сумісності, програміст має бути уважним, і повинен перевіряти, що його код виконується як очікується в широкому переліку можливих конфігурацій.

4.4 HTML

HTML (Hypertext Markup Language — Мова гіпертекстової розмітки) — це мова опису структури сторінок документів, яка дозволяє звичайний текст формувати в абзаци, заголовки, списки та інші структури, створювати посилання на інші сторінки. Це текстова мова, в якій інструкції з форматування, що називаються тегами, вбудовані в розділи документа, які містять конкретну інформацію. Теги повідомляють браузерам, як формувати і представляти інформацію на екрані. [10]

Мова HTML дозволяє визначити структуру електронного документа з поліграфічним рівнем оформлення. Результуючий документ може містити різноманітні елементи: ілюстрації, аудіо і відео фрагменти. Мова HTML включає розвинені засоби для визначення кількох рівнів заголовків, шрифтових виділень, різних груп об'єктів та багато інших можливостей.

За основу моделі розмітки документів у HTML прийнята тегова модель. Тегова модель описує документ як сукупність контейнерів, кожен з яких починається і закінчується тегами. Тобто документ HTML є не чим іншим, як звичайним ASCII-файлом з доданими до нього керуючими HTML-кодами (тегами)

Теги HTML-документів в основному є простими і зрозумілими для використання, оскільки вони створені за допомогою загальноживаних слів англійської мови, зрозумілих скорочень і позначень.

Для створення HTML-документа можна застосувати редактор ASCII (зокрема, Блокнот системи Windows). Такі редактори дозволяють вводити HTML-теги, не додаючи до створеного нічого додатково. Створення документа у такому редакторі дозволяє паралельно переглядати результат у програмі-браузері (Рисунок 4.4.1).

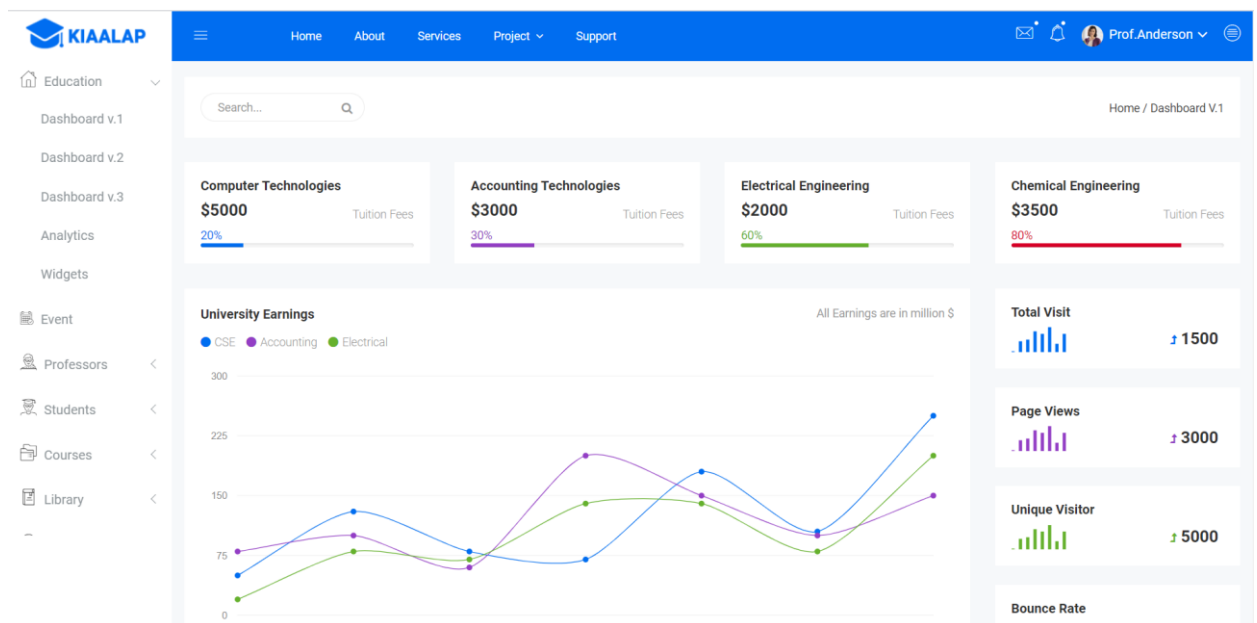


Рисунок 4.4.1 – Приклад веб-додатку мовою HTML.

Висновки

Для створення системи кодування, яка формує інформацію в кільцевий код було обрано архітектурний шаблон MVC та встановлено інтегроване середовище розробки Sublime Text 3 з використанням HTML та Javascript.

5 МЕТОДИКА РОБОТИ КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ

Цей розділ містить детальну інформацію щодо використаних технологій реалізації поставленої задачі створення системи кодування, яка формує з інформації кільцеві коди. Для реалізації даної задачі були використані знання, що були отримані під час аналізу поставленої проблеми.

Система кодування інформації шляхом формування кільцевого коду була створена у файлового вигляді. Користувач може використовувати дану СКІ просто отримавши *.html — файл веб-додатку.

Для запуску даної системи кодування користувачеві знадобиться мати встановлений будь-який браузер.

Для відкриття створеної системи кодування інформації у браузері достатньо вибрати відповідний файл з розширенням .html (Рисунок 5.1).



Рисунок 5.1 — Вид ярлику файлу з розширенням .html

Після чого користувач зможе побачити веб-сторінку з інтерфейсом системи (Рисунок 5.2).

Система кодування

Введіть вхідні дані

Кількість помилок

Закодувати

Кодування

Декодування

Вихідні дані

Декодувати

Кількість помилок

Рисунок 5.2 — Інтерфейс системи кодування

Скориставшись клавіатурою користувач може ввести інформацію, яку треба закодувати. Натиснувши кнопку «Закодувати» користувач отримає інформацію у виді кільцевого коду, тобто матриці $N*N$ (Рисунок 5.3).

Система кодування

Введіть вхідні дані

Кількість помилок

Кодування

0	1	1	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Рисунок 5.3 — Результат роботи кнопки «Закодувати»

Також користувач має можливість штучно додати помилку, яку система знайде і виправить (Рисунок 5.4). В полі «Декодування» буде виведено матрицю з помилкою.

Система кодування

Введіть вхідні дані

Кількість помилок

Кодування

0	1	1	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Декодування

0	0	1	1	1
1	0	0	1	1
1	1	0	0	1
1	1	1	0	0
0	1	1	1	0

Вихідні дані

Кількість помилок

Рисунок 5.4 — Додавання штучної помилки

Натиснувши кнопку «Декодувати» користувач отримає правильну і достовірну інформацію з виправленою помилкою і виведенням кількості помилок в окремому полі під назвою «Кількість помилок:» (Рисунок 5.5).

Система кодування

Введіть вхідні дані

Кількість помилок

Закодувати

Кодування

0	1	1	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Декодування

0	0	1	1	1
1	0	0	1	1
1	1	0	0	1
1	1	1	0	0
0	1	1	1	0

Вихідні дані

Декодувати

Кількість помилок

Рисунок 5.5 — Результат декодування отриманого образу інформації

Висновки

В цьому розділі продемонстрована робота з системою. Інтерфейс зроблений максимально зручно для того, щоб кожен користувач, не маючи гарних навиків користуватися ПК, міг застосовувати цю систему кодування.

Данна система дає можливість користувачу кодувати інформацію, декодувати та проводити пошук помилки в отриманій інформації.

ВИСНОВОК

У дипломній роботі згідно до поставленої мети, на основі теоретичних досліджень розв'язано проблему безпеки передачі даних, а саме розроблено систему кодування інформації шляхом формування її в кільцевий код. Проаналізовано та зроблено висновок, що кільцевий код є ефективним з точки зору безпеки та надійності передачі інформації, оскільки в канал зв'язку передається не інформація користувача, а її образ, тобто, вектор показників зсуву. Кільцевий код не є надлишковим, а дозволяє передавати інформацію в 2,7 разів менше, в порівнянні з об'ємом корисної інформації. Також продемонстровано математичні моделі утворення сімейств кільцевих кодів для побудови ефективного каналу передачі мережі. На тему бакалаврської роботи опубліковані тези для участі в конференції (Жицький А., Новоселов С., Андрушкевич М., Отрох С. «Сучасні проблеми наукового забезпечення енергетики»).

СПИСОК ДЖЕРЕЛ

1. Захист інформації в системах обміну даними. МОН УКРАЇНИ. [Електронний ресурс] — Режим доступу: <https://www.mil.gov.ua/ukbs/zahist-informaczii-v-sistemah-obminu-danimi.html>.
2. Надійність комп'ютерних мереж – Вікі ЦДПУ [Електронний ресурс]. – Режим доступу: <https://wiki.cuspu.edu.ua/index.php>.
3. Кодування інформації — Інформаційні системи і технології [Електронний ресурс] — Режим доступу: https://pidru4niki.com/12281128/bankivska_sprava/koduvannya_informatsiyi.
4. Існує три основні способи кодування інформації [Електронний ресурс]. – Режим доступу: <https://sites.google.com/site/navcalni/teoreticni-osnovi-informatiki>.
5. Компоненти системи обробки даних — Інформаційні системи і технології [Електронний ресурс] — Режим доступу: https://pidru4niki.com/12281128/bankivska_sprava/koduvannya_informatsiyi.
6. Отрох С. І., МЕТОДОЛОГІЯ ПІДВИЩЕННЯ ЯКОСТІ ФУНКЦІОНУВАННЯ МЕРЕЖІ МАЙБУТНЬОГО [Електронний ресурс] – 228 с. - Режим доступу: http://www.dut.edu.ua/uploads/p_1436_22936602.pdf.
7. Model-View-Controller [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org>.
8. Sublime Text for Web Developers [Електронний ресурс] — Режим доступу: <https://mherman.org/blog/sublime-text-for-web-developers>.
9. Javascript — [Електронний ресурс] — Режим доступу: <https://uk.wikipedia.org/wiki/JavaScript>.
10. Основні поняття HTML та структура — [Електронний ресурс] — Режим доступу: <https://sites.google.com/site/vivcaemowebdizajndistancijno/html/lekcija-3-osnovni-ponatta-movi-html-ta-struktura-dokumentiv>.

ДОДАТОК 1

Система кодування інформації

Специфікація

УКР.НТУУ"КПІ ім. Ігоря Сікорського"_ТЕФ_АПЕПС_ТМ61124_19Б

Аркушів 2

Київ – 2020

Позначення	Найменування	Примітки
Документація		
УКР.НТУУ«КПІ ім. Ігоря Сікорського»_ТЕФ_АПЕ ПС_ТМ61124_19Б	<u>Записка.docx</u>	Пояснювальна записка
Компоненти		
УКР.НТУУ«КПІ ім. Ігоря Сікорського»_ТЕФ_АПЕ ПС_ТМ61124_19Б 12-1	<u>Main.js</u>	Додаток кодування
УКР.НТУУ«КПІ ім. Ігоря Сікорського»_ТЕФ_АПЕ ПС_ТМ61 124_19Б 12-2	<u>index.html</u>	Веб-додаток системи кодування

ДОДАТОК 2

Система кодування інформації

Текст програми

УКР.НТУУ «КПІ ім. Ігоря Сікорського»_ТЕФ_АПЕПС_ТМ61124_19Б 12-2

Аркушів 7

Київ – 2020

Веб-интерфейс

```
<!DOCTYPE html>
<html>
<head>
  <title>1</title>
  <link href="https://fonts.googleapis.com/css2?family=Roboto&display=swap"
rel="stylesheet">
  <script src="main.js"></script>
  <style>
    body{
      margin: 0;
      background-color: #e4f8ff;
      display: flex;
      justify-content: center;
      align-items: center;
      font-family: 'Roboto', sans-serif;
    }
    .box{
      width: 900px;
      /*height: 750px;*/
      margin-top: 30px;
      background-color: #a9e9ff;
      border-radius: 5px;
    }
    .matrix{
      display: flex;
      flex-wrap: wrap;
      width: 140px;
    }
    .matrix_item{
      height: 20px;
      width: 20px;
    }
    .inputPart{
      display: flex;
      justify-content: space-around;
      align-items: center;
    }
    .miniBox{
      display: flex;
      flex-direction: column;
      justify-content: center;
```

```
        align-items: center;
        width: 200px;
        text-align: center;
    }
.button {
    background-color: #008CBA;
    border: none;
    color: white;
    padding: 15px 32px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 16px;
    margin: 4px 2px;
    cursor: pointer;
}
.row{
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
}
.leftSide{
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
    text-align:center;
}
.resultBox{
    display: flex;
    padding: 10px;
    margin-top: 30px;
    justify-content: space-around;
    align-items: center;
    background-color: #7bddff;
    border-radius: 5px;
}
.rightSide{
    width: 300px;
    display: flex;
    justify-content: center;
    align-items: center;
```

```

        flex-direction: column;
    }
    h1{
        margin: 5px;
        padding: 5px;
        padding-left: 30px;
    }
    h2{
        margin: 5px;
        padding: 5px;
    }
    .customInput{
        border-radius: 0.2rem;
        border: 2px solid #8a8cff;
    }
</style>
</head>
<body>
    <div class="box">
        <h1>Введите исходные данные</h1>
        <div class="inputPart">
            <div class="miniBox">
                <h2>Исходный вектор</h2>
                <input class="customInput inputData" maxlength='5 '>
            </div>
            <div class="miniBox">
                <h2>Количество ошибок</h2>
                <select class='select0432' style="width:100px">
                    <option value="0">0</option>
                    <option value="1">1</option>
                    <option value="2">2</option>
                    <option value="3">3</option>
                    <option value="4">4</option>
                </select>
            </div>
            <button id="btn1" class="button">Закодировать</button>
        </div>
    <div class="row">
        <h2>Кодирование</h2>
        <div class="matrix">
            <input class="matrix_item el00">
            <input class="matrix_item el01">

```

```
<input class="matrix_item el02">
<input class="matrix_item el03">
<input class="matrix_item el04">

<input class="matrix_item el10">
<input class="matrix_item el11">
<input class="matrix_item el12">
<input class="matrix_item el13">
<input class="matrix_item el14">

<input class="matrix_item el20">
<input class="matrix_item el21">
<input class="matrix_item el22">
<input class="matrix_item el23">
<input class="matrix_item el24">

<input class="matrix_item el30">
<input class="matrix_item el31">
<input class="matrix_item el32">
<input class="matrix_item el33">
<input class="matrix_item el34">

<input class="matrix_item el40">
<input class="matrix_item el41">
<input class="matrix_item el42">
<input class="matrix_item el43">
<input class="matrix_item el44">
</div>
</div>

<div class="resultBox">
  <div class="leftSide">
    <h2>Декодирование</h2>
    <div class="matrix">
      <input class="matrix_item secondArr">
      <input class="matrix_item secondArr">
      <input class="matrix_item secondArr">
      <input class="matrix_item secondArr">
      <input class="matrix_item secondArr">

      <input class="matrix_item secondArr">
      <input class="matrix_item secondArr">
      <input class="matrix_item secondArr">
```

```

<input class="matrix_item secondArr">
<input class="matrix_item secondArr">

<input class="matrix_item secondArr">
<input class="matrix_item secondArr">
<input class="matrix_item secondArr">
<input class="matrix_item secondArr">
<input class="matrix_item secondArr">

<input class="matrix_item secondArr">
<input class="matrix_item secondArr">
<input class="matrix_item secondArr">
<input class="matrix_item secondArr">
<input class="matrix_item secondArr">

<input class="matrix_item secondArr">
<input class="matrix_item secondArr">
<input class="matrix_item secondArr">
<input class="matrix_item secondArr">
<input class="matrix_item secondArr">
</div>
<h2>Конечный вектор</h2>
<input class="customInput resVec">
</div>
<div class="rightSide">
  <button class="button getRes">Разкодировать</button>
  <h2>Ошибок обнаружено:</h2>
  <input class="customInput error">
</div>
</div>
</div>
</body>
</html>

```

Кодування

```

window.onload = ()=>{
  document.querySelector("#btn1").addEventListener("click",function(){
    let arr = [];
    let data = document.querySelector('.inputData').value;
    arr = Array.from(data);
    console.log(arr);
  });
}

```

```

// 1st row
for(let i = 0; i<5; i++){
    document.querySelector(`.el0${i}`).value = arr[i];
}
// 2
document.querySelector('.el10').value = arr[4];
for(let i = 0; i<4; i++){
    document.querySelector(`.el1${i+1}`).value = arr[i];
}
// 3
document.querySelector('.el20').value = arr[3];
document.querySelector('.el21').value = arr[4];
document.querySelector('.el22').value = arr[0];
document.querySelector('.el23').value = arr[1];
document.querySelector('.el24').value = arr[2];

// 4
document.querySelector('.el30').value = arr[2];
document.querySelector('.el31').value = arr[3];
document.querySelector('.el32').value = arr[4];
document.querySelector('.el33').value = arr[0];
document.querySelector('.el34').value = arr[1];

// 5
document.querySelector('.el40').value = arr[1];
document.querySelector('.el41').value = arr[2];
document.querySelector('.el42').value = arr[3];
document.querySelector('.el43').value = arr[4];
document.querySelector('.el44').value = arr[0];
let addArray = []
// push to arr everything from 'кодирование'
for(let i=0;i<25;i++){
    let docs = document.querySelectorAll('.matrix_item');
    addArray.push(docs[i].value);
}

// edit value from 2 pos
let dumbArray =
[addArray[1],addArray[7],addArray[13],addArray[19],addArray[20]];
for(let i =0;i<5;i++){
    if(dumbArray[i] == '0'){
        dumbArray[i] = '1'
    }else if(dumbArray[i] == '1'){

```

```

        dumbArray[i] = '0'
    }
}
// push changes to main arr for 2 grid
addArray[1] = dumbArray[0];
addArray[7] = dumbArray[1];
addArray[13] = dumbArray[2];
addArray[19] = dumbArray[3];
addArray[20] = dumbArray[4];
let secGrid = document.querySelectorAll('.secondArr');

for(let i = 0; i<25;i++){
    secGrid[i].value = addArray[i];
}
secGrid[1].style.backgroundColor = '#fffdc2';
secGrid[7].style.backgroundColor = '#fffdc2';
secGrid[13].style.backgroundColor = '#fffdc2';
secGrid[19].style.backgroundColor = '#fffdc2';
secGrid[20].style.backgroundColor = '#fffdc2';
console.log('addarray', addArray);

    document.querySelector('.getRes').addEventListener('click', function(){
        document.querySelector('.error').value =
document.querySelector('.select0432').value == "0" ? '1':
document.querySelector('.select0432').value;
        document.querySelector('.resVec').value =
document.querySelector('.inputData').value;
    })
})
}

```

ДОДАТОК 3

Система кодування інформації

Опис програмного модулю

УКР.НТУУ «КПІ ім. Ігоря Сікорського»_ТЕФ_АПЕПС_ТМ61124_19Б 13-1

Аркушів 8

Київ – 2020

АНОТАЦІЯ

Веб-додаток системи надає можливість зручного користування програмою та кодування інформації..

Розроблене програмне забезпечення дозволяє вводити вхідні данні, кодувати інформацію, штучно додавати помилку, декодувати, позбавляти помилки кільцевий код та виводити вихідні дані.

Для створення системи кодування, яка формує інформацію в кільцевий код було обрано архітектурний шаблон MVC та встановлено інтегроване середовище розробки Sublime Text 3 з використанням HTML та Javascript .

ЗМІСТ

1. Загальні відомості	4
2. Функціональне призначення	5
3. Опис логічної структури.....	6
4. Використовувані технічні засоби	7
5. Вхідні і вихідні дані	8

ЗАГАЛЬНІ ВІДОМОСТІ

Відповідно до назви дипломної роботи, розроблений продукт - це система кодування, яка кодує інформацію в кільцевий код. Для спрощення взаємодії було також розроблено зручний веб-додаток системи.

Користувач повинен мати будь-який браузер, який працює з системою.

Для використання програми просто запусіть файл за допомогою веб-браузера, не потрібно ніяких втручань, що вимагають знань з програмування.

ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ

Розроблений програмний засіб покликаний вирішити задачу ручного введення даних для кодування. Це було реалізовано за допомогою кількох функцій системи, а саме:

- Зчитування введеної інформації ;
- Формування інформації в кільцевий код;
- Представлення закодованої інформації у виді матриці;
- виправлення та знаходження помилки;
- Декодувати інформацію;
- Вивід правильної інформації після декодування.

-6-

ОПИС ЛОГІЧНОЇ СТРУКТУРИ

Загальний принцип роботи додатку такий:

- 1) користувач запускає веб-додаток;
- 2) користувач вводить інформацію, яку хоче закодувати;
- 3) після натиску миші виводиться закодована інформація у вигляді кільцевого коду;
- 4) система запам'ятовує структуру переданої інформації;
- 5) користувач штучно додає помилку в кільцевий код;
- 6) після натиску кнопки «Декодувати» система знаходить помилку і виправляє її;
- 7) система виводить правильну інформацію і кількість помилок.

ВИКОРИСТОВУВАНІ ТЕХНІЧНІ ЗАСОБИ

Для користування системою кодування інформації користувачеві потрібен встановлений будь-який браузер з налагодженим Інтернетом. Додаток розроблений саме для частини цього програмного комплексу — системи кодування.

Користувач не обтяжений встановленням окремих програмних засобів та інструментів, які потребують спеціальних навичок.

-8-

ВХІДНІ І ВИХІДНІ ДАНІ

Вхідними даними є:

— інформація, яку треба закодувати.

Вихідними даними є:

— правильна інформація ;

— кількість помилок.