

Національний лісотехнічний університет України  
(повне найменування вищого навчального закладу)

Навчально-науковий інститут деревообробних та  
комп'ютерних технологій і дизайну  
(повне найменування інституту, назва факультету (відділення))

Кафедра інформаційних технологій  
(повна назва кафедри (предметної, циклової комісії))

## **Пояснювальна записка**

до дипломної роботи

перший (бакалаврський)

(рівень вищої освіти)

на тему: **Розробка програмного застосунку забезпечення завадостійкості  
з використанням баркероподібної послідовності**

Виконав: студент 5 курсу, групи КНз-51  
спеціальності

122 – “Комп'ютерні науки”

(шифр і назва напряму підготовки, спеціальності)

Рибак Н. В.

(прізвище та ініціали)

Керівник Самотій Т.С.

Процик Ю.С.

(прізвище та ініціали)

Рецензент \_\_\_\_\_

(прізвище та ініціали)

Львів – 2022 р.  
Національний лісотехнічний університет України  
 (повне найменування вищого навчального закладу)

ННІ деревообробних та комп'ютерних технологій і дизайну  
 Кафедра інформаційних технологій  
 Рівень вищої освіти перший (бакалаврський)  
 Спеціальність 122 “Комп'ютерні науки”  
 (шифр і назва)  
**ЗАТВЕРДЖУЮ**

Завідувач кафедри

\_\_\_\_\_ Крошній І. М.  
 “\_\_\_” \_\_\_\_\_ 2022 року

**З А В Д А Н Н Я**  
**НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ**  
**Рибаку Назарію Васильовичу**

(прізвище, ім'я, по батькові)

1. Тема роботи Розробка програмного застосунку забезпечення завадостійкості з використанням баркероподібної послідовності  
 керівник роботи ас. Самотій Тетяна Сергіївна, доц. Процик Юрій Степанович  
 (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
- Затверджені наказом вищого навчального закладу від 13.12. 2021 року № С-618
2. Термін подання студентом роботи 11. 04. 2022 р.
3. Вихідні дані до роботи:
  - провести огляд завадостійких послідовностей
  - дослідити математичну модель завадостійких баркероподібних послідовностей;
  - розробити на мові програмування C++ застосунок з інтуїтивним інтерфейсом;
  - провести тестування роботи розробленого застосунка.
4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)
  - a. Характеристика об'єкту дослідження та постановка задачі
  - b. Огляд літератури
  - c. Системний аналіз
  - d. Розробка та відображення алгоритмів побудови баркероподібних послідовностей
  - e. Розробка програмного рішення
  6. Експериментальна частина
5. Перелік графічного матеріалу (системний аналіз, розробка та відображення алгоритмів роботи побудови баркероподібних послідовностей, розробка програмного рішення, експериментальна частина)  
 Додаток А. Код програми синтезу завадостійких баркероподібних послідовностей
6. Дата видачі завдання 15 грудня 2021 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел	15.12-31.12.2021	виконано
2	Характеристика об'єкту дослідження та постановка задачі	03.01-15.01.2022	виконано
3	Огляд літератури	17.01-22.01.2022	виконано
4	Системний аналіз	24.01-29.01.2022	виконано
5	Розробка та відображення алгоритмів роботи компонентів та системи	31.01-19.02.2022	виконано
6	Розробка програмного рішення	21.02-12.03.2022	виконано
7	Експериментальна частина	14.03-19.03.2022	виконано
8	Оформлення дипломної роботи	21.03-31.03.2022	виконано
9	Підготовка до захисту дипломної роботи, оформлення презентації	01.04-09.04. 2022	виконано

**Студент**

\_\_\_\_\_ (підпис)

**Рибак Н. В.**

(прізвище та ініціали)

**Керівники роботи** \_\_\_\_\_

**Самотій Т.С.**

(підпис)

(прізвище та ініціали)

**Процик Ю.С.**

(підпис)

## РЕФЕРАТ

Дипломна робота містить 71 сторінку пояснювальної записки, 11 рисунків, 3 таблиці, 1 додаток, 39 літературних джерел.

В дипломній роботі спроектовано і здійснена реалізація програмного додатку «Програмний застосунок забезпечення завадостійкості з використанням баркероподібної послідовності» відповідно до технічного завдання.

У першому розділі детально описуються баркероподібні послідовності, їх типи, відомі методи та алгоритми побудови, а також практичне застосування завадостійких баркероподібних кодових послідовностей.

У другому розділі детально представлені методи та алгоритми побудови завадостійких баркероподібних кодових послідовностей на основі ідеальних кільцевих в'язанок для проведення подальших досліджень.

У третьому розділі приведений детальний опис розробленого програмного продукту щодо застосування завадостійких баркероподібних послідовностей на основі ідеальних кільцевих в'язанок для їх практичного використання.

Програмний продукт виконаний з використанням інтегрованого середовища мови C++ під операційною системою Windows 11.

На сьогоднішній день побудова завадостійких баркероподібних послідовностей на основі ідеальних кільцевих в'язанок є актуальною задачею з практичним впровадженням і протестований розроблений програмний продукт є конкурентноздатним.

**Ключові слова:** завадостійка баркероподібна послідовність, ідеальна кільцева в'язанка, код Баркера.

## ABSTRACT

This thesis contains 71 pages of explanatory note, 11 figures, 3 tables, 1 appendix, 39 references.

In the thesis, the software application "Software application for noise immunity using a barker-like sequence" was designed and implemented in accordance with the terms of reference.

The first section describes in detail the barker-like groups, their types, known methods and algorithms for construction, as well as the practical application of noise-tolerant barker-like code responses.

The second section presents methods and algorithms for constructing noise-tolerant barque-like code-based elements based on ideal knitted rings for further research.

The third section provides a detailed description of the developed software product for the use of noise-resistant barkers based on ideal knitted rings for their practical use.

The software product is made using the integrated environment of the C ++ language under the Windows 11 operating system.

Today, the construction of noise-resistant barker-like sequences on the basis of ideal ring bundles is an urgent task with practical implementation and the tested developed software product is competitive.

**Keywords:** noise-tolerant barker-like code sequence, perfect ring bundle, Barker code.

## ТЕХНІЧНЕ ЗАВДАННЯ

В дипломній роботі потрібно вирішити такі завдання:

- дослідження побудови завадостійких баркероподібних послідовностей;
- оптимізація алгоритму побудови завадостійких баркероподібних послідовностей;
- порівняння та аналіз характеристик побудованих завадостійких баркероподібних послідовностей з відомими послідовностями;
- практичне застосування побудованих завадостійких баркероподібних послідовностей.

## ЗМІСТ

### ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ОДИНИЦЬ І ТЕРМІНІВ<sup>9</sup>

#### ВСТУП 10

#### РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ 12

##### 1.1 ХАРАКТЕРИСТИКА ОБ'ЄКТУ ПРОЕКТУВАННЯ ТА ПОСТАНОВКА ЗАДАЧІ 12

##### 1.2. ОГЛЯД ЛІТЕРАТУРИ 15

##### 1.2.1 Завадостійка система на основі кодів Боуза-Чоудхурі-Хоквінгема 15

##### 1.2.2. Завадостійка система на основі ідеальних кільцевих в'язанок 18

#### РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ 20

##### 2.1. СИСТЕМНИЙ АНАЛІЗ 20

##### 2.1.1 Дерево цілей завадостійких послідовностей 20

##### 2.1.2 Дерево проблем вибору завадостійких баркероподібних послідовностей 22

##### 2.2. РОЗРОБКА ТА ВІДОБРАЖЕННЯ АЛГОРИТМІВ ПОБУДОВИ БАРКЕРОПОДІБНИХ ПОСЛІДОВНОСТЕЙ 24

##### 2.2.1. Синтез завадостійких баркероподібних кодів за допомогою ідеальних кільцевих в'язанок 24

##### 2.2.2. Синтез кодових комбінацій на основі ІКВ для баркероподібних послідовностей 26

#### РОЗДІЛ 3. ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ 30

##### 3.1. РОЗРОБКА ПРОГРАМНОГО РІШЕННЯ 30

##### 3.1.1. Програма синтезу завадостійких баркероподібних послідовностей 30

##### 3.1.2. Коригувальна спроможність баркероподібних послідовностей 31

##### 3.1.3. Порівняльна характеристика кодів ІКВ та БЧХ 32

##### 3.2. ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА 34

##### 3.2.1. Практичне застосування завадостійкої баркероподібної системи кодування на основі ІКВ 34

**ВИСНОВКИ 38**

**ПЕРЕЛІК ПОСИЛАНЬ 39**

**ДОДАТОК А. КОД ПРОГРАМИ СИНТЕЗУ ЗАВАДОСТІЙКИХ  
ПОСЛІДОВНОСТЕЙ БАРКЕРА 43**

**ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ОДИНИЦЬ І  
ТЕРМІНІВ**

АКФ	автокореляційна функція
ВКФ	взаємна кореляційна функція
ІКВ	ідеальна кільцева в'язанка
ПАХ	поверхнева акустична хвиля
ПЗ	програмне забезпечення
РЧІД	радіочастотна ідентифікація
ШПС	шумоподібний сигнал
ШСС	широкосмуговий сигнал
ФКМ	фазова кодова маніпуляція
ФМС	фазоманіпулірований сигнал

## ВСТУП

**Актуальність роботи.** Сучасні системи передачі та зберігання інформації не ідеальні. Застосовуючи комп'ютерну техніку, необхідно враховувати можливість появи помилок при передачі і зберіганні даних. Причиною можуть стати як завади (noise), які діють в інформаційних системах, так і дії зловмисників, що направлені на порушення цілісності даних.

**Метою роботи** є дослідження деяких властивостей завадостійких баркероподібних послідовностей, які дозволять збільшити потужність кодових послідовностей, що підвищить кількість знайдених і виправлених помилок.

**Об'єкт проектування** – завадостійкі баркероподібні послідовності.

**Предмет дослідження** – метод швидкої побудови завадостійких баркероподібних послідовностей.

**Задачі роботи:**

- визначення методу швидкої побудови завадостійких баркероподібних послідовностей;
- оптимізація алгоритму швидкої побудови завадостійких баркероподібних послідовностей;
- порівняння та аналіз характеристик отриманих завадостійких баркероподібних послідовностей з відомими послідовностями;
- розгляд деяких практичних прикладів отриманих завадостійких баркероподібних послідовностей.

**Наукова новизна.** Серед відомих існуючих послідовностей Баркера максимальне співвідношення головного пелюстка до інших пелюстків дорівнює 13. Для баркероподібних послідовностей було знайдено співвідношення, що дорівнює 14 та більше. Також запропонований швидкий алгоритм побудови баркероподібних послідовностей.

**Практична цінність.** При найбільшому співвідношенні головного пелюстка до інших пелюстків, що визначає співвідношення сигнал/шум, появляється

можливість розробки алгоритмів, які дозволяють створювати ефективну систему прийому/передачі даних відносно кількості завад.

## РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ

### 1.1 ХАРАКТЕРИСТИКА ОБ'ЄКТУ ПРОЕКТУВАННЯ ТА ПОСТАНОВКА ЗАДАЧІ

Особливість багатопозиційних кодів полягає в тому, що їх взаємна кореляція мінімальна. Багатопозиційні коди створюються на основі коду Баркера і лінійок Голомба.

Найкращим кодом для створення шумоподібної послідовності є код Баркера, але він має велику надлишковість. Багатопозиційні коди на основі лінійок Голомба мають меншу надлишковість.

Корисною особливістю систем з багатопозиційних кодів є їх високі конфіденційність і завадостійкість. У основі техніки багатопозиційних кодів лежить використання в каналі зв'язку для перенесення інформації декількох реалізацій цих кодів, розділення яких на прийомі здійснюється за допомогою селекції їх послідовності.

Перевагою багатопозиційного коду є можливість застосовувати новий вигляд селекції – за допомогою послідовності. Цікавою особливістю систем з шумоподібними кодами є її адаптивні властивості із зменшенням числа завад завадостійкість зростає.

Недоліком є перехід до складнішого носія інформації, що приводить, природно, до відомого ускладнення систем передачі повідомлень.

Властивості цих завадостійких баркероподібних послідовностей:

- у кожному періоді кількість чисел 1 і 0 відрізняється не більш, ніж на одиницю;
- серед груп чисел 1 і 0 в кожному періоді половина має довжину в один символ, четверта частина має довжину в два символи, восьма частина має довжину в чотири символи і так далі.
- кореляційна функція послідовності має єдиний значний пік амплітуди 1 і при всіх зрушеннях рівна  $1/m$  ( $m$  – довжина послідовності).

Кореляція між векторами обчислюється за формулою:

$$p(x, y) = \frac{A - B}{A + B} \quad ((1.1))$$

де  $A$  - число позицій, в яких символи послідовностей  $x$  і  $y$  збігаються;

$B$  - число позицій, в яких символи послідовностей  $x$  і  $y$  різні.

Мінімальна кодова відстань для багатопозиційного коду визначається як:

$$d_{min} = 2(N - R) \quad (1.2)$$

Число помилок, які можна виявити  $t_1$ , і число помилок, що можна виправити

$t_2$  за допомогою шумоподібного коду, визначається мінімальною кодовою

відстанню:

$$t_1 \leq d_{min} - 1 \quad (1.3)$$

$$t_2 \leq \frac{t_1 - 1}{2} \quad ((1.4))$$

Для побудови багатопозиційного коду виберемо ПКВ з наступними параметрами:

$$n = 4, R = 2, n = 4, R = 2, S_n = \frac{4(4-1)}{2} + 1 = 7$$

$$S_n = \frac{4(4-1)}{2} + 1 = 7 : 1\ 1\ 2\ 3\ 1\ 1\ 2\ 3. \quad (1.5)$$

Кодові комбінації за допомогою баркероподібної послідовності формуються за рахунок циклічного зсуву вправо.

Наведемо таблицю кодівих комбінацій монолітного коду.

Таблиця 1.1.

Багатопозиційний код на основі ІКВ (1, 1, 2, 3)

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

Мінімальна кодова відстань даного коду:

$$d_{\min} = 2(4 - 2) = 4$$

Число помилок, які можна виявити:

$$t_1 \leq 4 - 1 = 3$$

Число помилок, які можна виправити:

$$t_2 \leq \frac{3 - 1}{2} = 1$$

Багатопозиційний код на основі ІКВ (1, 1, 2, 3) може закодувати 7 значень, виявити три помилки і виправити одну. Багатопозиційні коди відносяться до множини з украй нерегулярною розгалуженою структурою. Основні поняття теорії поки що знаходяться в процесі становлення та розвитку, але поле їх застосування безперервно розширюється.

Великий інтерес до цих кодів пов'язаний з тим, що їх аналоги, такі як баркероподібні кодові послідовності, лінійки Голомба, числові в'язанки знаходять використання в реальних завданнях, причому в типових, а не в екзотичних ситуаціях.

## 1.2. ОГЛЯД ЛІТЕРАТУРИ

### 1.2.1 Завадостійка система на основі кодів Боуза-Чоудхури-Хоквінгема

Коди Боуза-Чоудхури-Хоквінгема (БЧХ) являють собою великий клас кодів, здатних виправляти кілька помилок і займають помітне місце в теорії і практиці кодування.

Примітивним кодом БЧХ, що виправляє  $t_u$  помилок, називається код довжиною  $n = gm - 1$  над  $GF(gm)$ , для якого елементи  $a^1, a^2, \dots, a^{2t_u}$  є коріннями породжувального багаточлена. Тут примітивний елемент  $GF(g)$  – породжувальний багаточлен, визначається з виразу  $g(x) = \text{НОК} [f_1(x), f_2(x), \dots, f_{2t_u}(x)]$ , де  $f(x)$  – мінімальні багаточлени кореня  $g(x)$ . Число перевірочних елементів коду БЧХ задовольняє співвідношенню  $r = n - k \leq mt_u$ .

Для прикладу визначимо значення породжувального багаточлена для побудови примітивного коду над  $GF(2)$  довжини 31, що виправляє двократні помилки ( $t_u = 2$ ). Виходячи з визначення значення мінімальних многочленів для побудови коду є наступними:

$$\begin{aligned} f_1(x) &= (x - a^1)(x - a^2)(x - a^4)(x - a^8)(x - a^{16}) = x^5 + x^2 + 1 & (2.1) \\ f_2(x) &= (x - a^2)(x - a^4)(x - a^8)(x - a^{16})(x - a^1) = x^5 + x^2 + 1 \\ f_3(x) &= (x - a^3)(x - a^6)(x - a^{12})(x - a^{24})(x - a^{17}) = x^5 + x^4 + x^3 + x^2 + 1 \\ f_4(x) &= (x - a^4)(x - a^8)(x - a^{16})(x - a^1)(x - a^2) = x^5 + x^2 + 1 \end{aligned}$$

Оскільки  $f_1(x) = f_2(x) = f_4(x)$ , то

$$g(x) = f_1(x) \cdot f_3(x) = (x^5 + x^2 + 1) \cdot (x^5 + x^4 + x^3 + x^2 + 1) = x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1 \quad (2.2)$$

На практиці при визначенні значень породжувального багаточлена користуються спеціальною таблицею мінімальних багаточленів, і виразом для породжувального багаточлена  $g(x) = f_1(x), f_2(x), \dots, f_n(x)$ . При цьому робота спрощується та по заданій довжині коду  $n$  і кратності помилок  $t_u$ , що виправляють, визначають:

- з виразу  $n = 2m - 1$  значення параметра  $m$ , що є максимальним ступенем співмножників  $g(x)$ ;
- з виразу  $j = 2m - 1$  максимальний порядок мінімального багаточлена, що входить у число співмножників  $g(x)$ ;
- користуючись таблицею мінімальних багаточленів, визначається вираз для  $g(x)$  залежно від  $m$  і  $j$ . Для цього з колонки, що відповідає параметру  $m$ , вибираються багаточлени з порядками від 1 до  $j$ , які в результаті перемножування дають значення  $g(x)$ .

Для прикладу визначимо значення породжувального багаточлена для побудови примітивного коду над  $GF(2)$  довжини 31, що забезпечує  $t_u = 2$ . Визначаємо значення  $m$  і  $j$ :

$$\begin{aligned} m &= \log_2(n + 1) = \log_2 32 = 5 \\ j &= 2 \cdot t_u - 1 = 2 \cdot 2 - 1 = 3 \end{aligned} \quad (2.3)$$

Із таблиці мінімальних багаточленів відповідно до  $m = 5$  і  $j = 3$  одержуємо:

$$\begin{aligned} g(x) &= 45 \cdot 75 \cdot 67 = (x^5 + x^2 + 1)(x^5 + x^4 + x^3 + x^2 + 1)(x^5 + x^4 + x^2 + x + 1) = \\ &= x^{15} + x^{11} + x^{10} + x^9 + x^8 + x^7 + x^5 + x^3 + x^2 + x + 1 \end{aligned} \quad (2.4)$$

Аналогічно визначається поліноми для побудови циклічних кодів з іншими параметри. В контексті задачі порівняння обох криптографічних систем спробуємо побудувати многочлен для коду, довжина якого дорівнює 15:

$$t_u = 2, GF(2):$$

$$\begin{aligned} g(x) &= \text{НОК}[f_1(x), f_2(x), f_3(x), f_4(x)] = \text{НОК}[x^4 + x + 1, x^4 + x + 1, x^4 + x^3 + x^2 + x + 1, x^4 + x + 1] = \\ &= (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1) = x^8 + x^7 + x^6 + x^4 + 1 \end{aligned} \quad (2.5)$$

Оскільки ступінь  $g(x) = 8$ , то  $n - k = 8 \Rightarrow k = 7$  і ми отримали породжувальний багаточлен коду БЧХ (15, 7), що виправляє 2 помилки. Відзначимо, що коди БЧХ будуються по заданим  $n$  та  $t_u$ , значення  $k$  не відомо, поки не знайдений  $g(x)$ ;

$$t_u = 3, GF(2):$$

$$g(x) = \text{НОК}[f_1(x), f_2(x), f_3(x), f_4(x), f_5(x), f_6(x)] = (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1)(x^2 + x + 1) = (2.6) \\ = x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1$$

Це породжувальний багаточлен для коду БЧХ (15, 5), що виправляє 3 помилки;

$$t_u = 4, GF(2):$$

$$g(x) = \text{НОК}[f_1(x), f_2(x), f_3(x), f_4(x), f_5(x), f_6(x), f_7(x), f_8(x)] = (x^4 + x + 1) \times (2.7) \\ \times (x^4 + x^3 + x^2 + x + 1)(x^2 + x + 1)(x^4 + x^3 + 1) = x^{14} + x^{13} + x^{12} + x^{11} + x^{10} + x^9 + \\ x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$$

Це породжувальний багаточлен для коду БЧХ (15, 1). Ми получили простий код з повторенням, що виправляє 7 помилок. Всі попередні багаточлени будувалися на основі множини  $GF(2)$ . Для подальшого дослідження можливостей кодів БЧХ спробуймо визначити багаточлени на множині  $GF(4)$ .

$$t_u = 4, GF(4):$$

$$g(x) = x^{11} + x^{10} + 2x^8 + 3x^7 + 3x^6 + x^5 + x + 3 \quad (2.8)$$

Це породжувальний багаточлен для коду БЧХ (15, 4), що виправляє 4 помилки;

$$t_u = 5, GF(4):$$

$$g(x) = x^{12} + 2x^{11} + 3x^{10} + 2x^9 + 2x^8 + x^7 + 3x^6 + 3x^4 + 3x^3 + x^2 + 2 \quad (2.9)$$

Це породжувальний багаточлен для коду БЧХ (15, 3), що виправляє 5 помилки;

$$t_u = 5, GF(4):$$

$$g(x) = x^{12} + 2x^{11} + 3x^{10} + 2x^9 + 2x^8 + x^7 + 3x^6 + 3x^4 + 3x^3 + x^2 + 2 \quad (2.10)$$

Це породжувальний багаточлен для коду БЧХ (15, 1), що виправляє 6 помилок.

Насправді ми одержали простий код з повторенням, що в реальності виправляє. Проаналізувавши отримані багаточлени та параметри коду БЧХ можна дійти до висновку, що найоптимальнішу сукупність завадостійких кодів ми отримаймо на основі багаточлена  $g(x) = x^{11} + x^{10} + 2x^8 + 3x^7 + 3x^6 + x^5 + x + 3$ . Завадостійкий код на його основі здатен виправити 4 помилки в блоці довжиною 15 та кількість інформаційних позицій 4.

Задані вхідні дані:  $n$  і  $t_u$  або  $k$  і  $t_u$  для побудови циклічного коду часто приводять до вибору завищеного значення  $m$  і, як наслідок цього, до

невиправданого збільшення довжини коду. Таке положення знижує ефективність отриманого коду, тому що частина інформаційних розрядів взагалі не використовується, а тому є актуальним вибір оптимальних параметрів з мінімальною ступеню надлишковості.

Коди БЧХ є циклічним, а отже при кодування та декодуванні використовуються будь-які методи кодування та декодування інформації, призначені для циклічних кодів.

### **1.2.2. Завадостійка система на основі ідеальних кільцевих в'язанок**

Перед нами стоїть задача побудувати завадостійку систему, здатну забезпечити захист по двом основним напрямкам:

- від завад – система повинна володіти якомога вищою коректувальною здатністю для успішної передачі величезного масиву даних по мережі, де, як відомо, можуть виникати перешкоди різного характеру;
- від несанкціонованого доступу – алгоритм, на базі якого побудована дана система, зобов'язаний бути достатньо складним та неочевидним, щоб унеможливити декодування масиву даних якимось обхідним шляхом.

Тож перш, ніж приступати до програмної реалізації, потрібно оцінити технічну доцільність реалізації завадостійкої системи на основі ідеальних кільцевих в'язанок (ІКВ), а для необхідно порівняти її з кращими аналогічними системами, що відомі на сьогоднішній день. Більшість експертів такою вважають систему на основі кодів Боуза- Чоудхурі-Хоквінгема (БЧХ). В ході порівняння беруться до уваги такі показники як швидкість процесу кодування-декодування, ступінь надлишковості, коректувальна здатність, алгоритмічна складність тощо.

В'язанки дають змогу забезпечити достатньо високу коректувальну здатність такої системи за рахунок своїх унікальних комбінаторних властивостей, а неподібність процес кодування-декодування в порівнянні з існуючими

криптографічними методами, що використовують властивості багаточленів, робить захист ще сильнішим.

## РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

### 2.1. СИСТЕМНИЙ АНАЛІЗ

#### 2.1.1 Дерево цілей завадостійких послідовностей

Найбільш доцільне використання завадостійкого кодування, коли разом з інформаційними бітами передаються контрольні біти, що використовуються для перевірок наявності помилок та для їх виправлення.

Коди, які дозволяють знаходити та виправляти помилки називають коригуючими кодами і вони бувають наступними (рис. 2.1):

- блочні;
- неперервні;
- згорточні;
- рівномірні;
- нерівномірні;
- подільні;
- неподільні;
- рівноважні;
- систематичні;
- несистематичні;
- баркероподібні коди;
- комбінаторні коди.

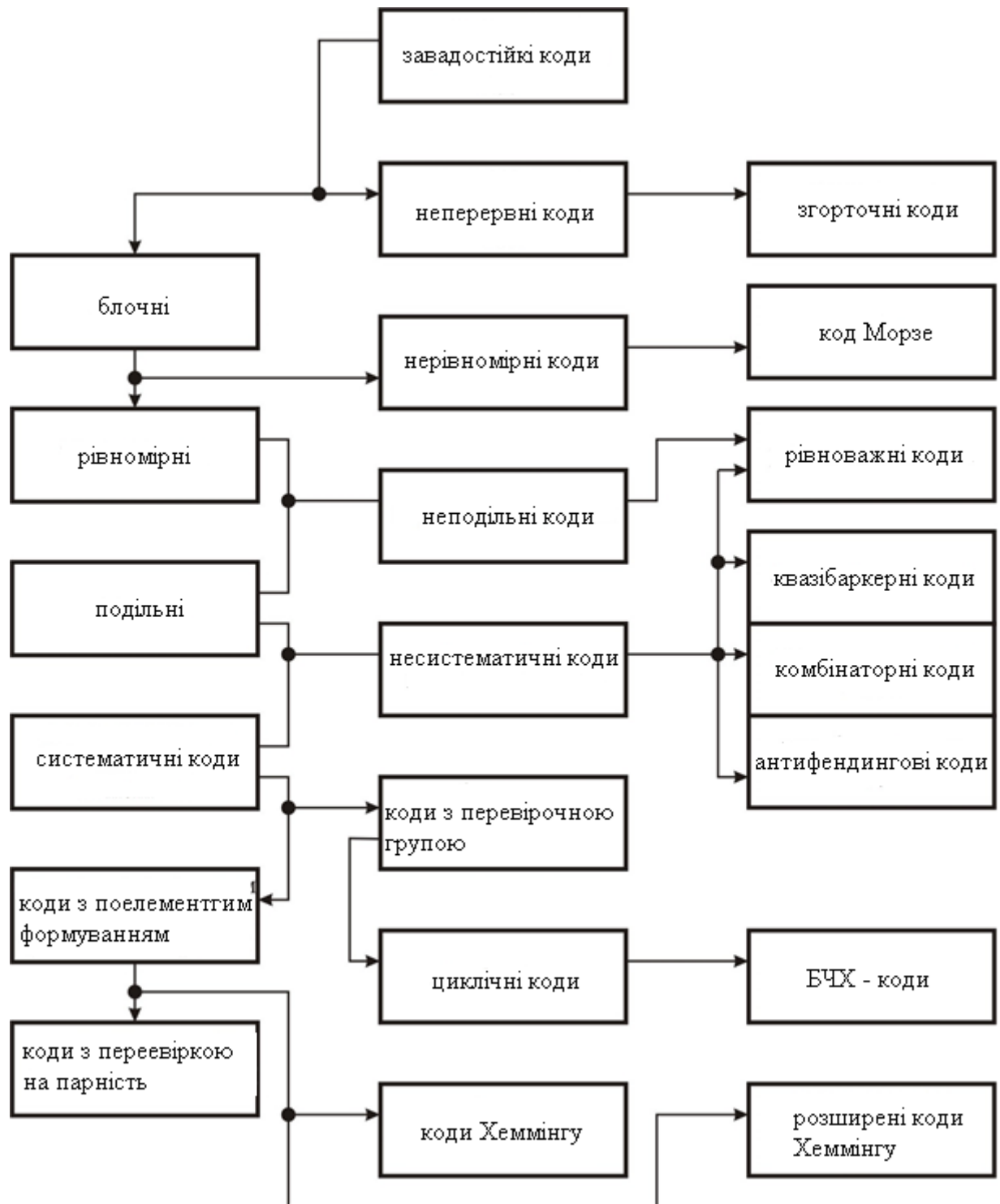


Рис. 2.1. Дерево цілей завадостійких послідовностей

На рис. 2.1 показано дерево цілей побудови завадостійких послідовностей.

### 2.1.2 Дерево проблем вибору завадостійких баркероподібних послідовностей

Сучасні кодові послідовності, що застосовуються для розширення смуги частот і створюють широкопasmовий сигнал, поділяються на два класи: ортогональні (квазіортогональні) та псевдовипадкові з мінімальним рівнем взаємної кореляції (рис. 2.2).

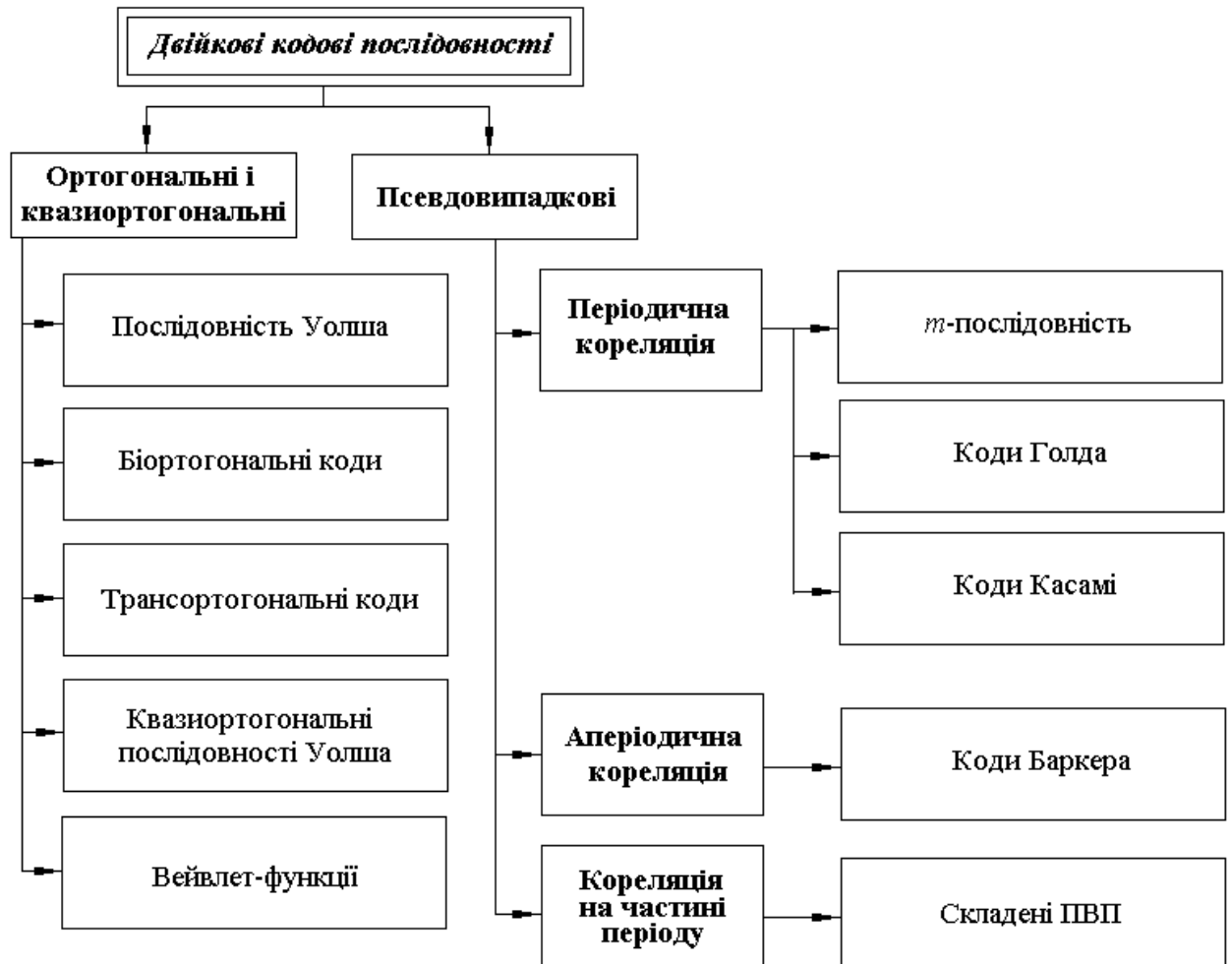


Рис. 2.2. Дерево проблем вибору завадостійких баркероподібних послідовностей

Методи побудови широкопasmових завадостійких баркероподібних послідовностей класифікують за базою сигналу, що визначається формулою (2.1):

$$B = F \cdot T, \quad (2.1)$$

де  $F$  – смуга частот сигналу;

$T$  – час передачі.



Рис. 2.3. Дерево методів синтезу широкопasmових завадостійких баркероподібних послідовностей

Методи синтезу широкопasmових завадостійких баркероподібних послідовностей наведені на рис. 2.3.

## 2.2. РОЗРОБКА ТА ВІДОБРАЖЕННЯ АЛГОРИТМІВ ПОБУДОВИ БАРКЕРОПОДІБНИХ ПОСЛІДОВНОСТЕЙ

### 2.2.1. Синтез завадостійких баркероподібних кодів за допомогою ідеальних кільцевих в'язанок

Синтез завадостійких кодів на основі ідеальних кільцевих в'язанок (ІКВ) став можливим завдяки тому, що ідеальному кільцю з параметрами  $n$  та  $R$  циклічна ВІВ-схема на множині елементів  $\{b_j\} = \{j\}, j = \overline{1, 2, \dots, S_n}$ , яка є одним із основних видів блок-схем.

Під блок-схемою розуміють розміщення елементів множини  $\{b_i\} = \{i\}, i = \overline{1, 2, \dots, v}$  в  $a$  підмножинах  $B_j, j = \overline{1, 2, \dots, a}$ , які називаються блоками, з однаковим числом елементів  $k_j = k, A_j = 1, 2, \dots, a$  в кожному блоці, причому елемент  $b_i$  належить до  $r_i$  різних блоків, а кожна  $P^-$ -та пара різних елементів  $(b_i, b_j), i \neq j, p = 1, 2, \dots, \frac{v(v-1)}{2}$  повторюється в  $\lambda_p$  блоках.

Як говорилося вище, до основного виду блок-схем належать зрівноважені неповні блок-схеми (balanced incomplete block design) або ВІВ-схеми. ВІВ-схеми утворені на основі множини  $v$  різних елементів ( $i \neq j \Rightarrow b_i \neq b_j$ ) з числом елементів  $k < v$  у кожному блоці, причому для розміщення елементів у блоках характерно те, що:

всі елементи одного блоку різні;

кожний елемент з'являється точно в  $r$  різних блоках ( $r_i = r, i = 1, 2, \dots, v$ );

кожна  $P^-$ -та пара різних елементів  $(b_i, b_j), p = 1, 2, \dots, \frac{v(v-1)}{2}$  з'являється точно

в  $\lambda$  різних блоках  $\lambda_p = \lambda$ .

Між параметрами  $v, a, k, r, \lambda$  ВІВ-схеми існують залежності, які виражаються відомими співвідношеннями:

$$ak = vr \quad (2.2)$$

$$r(k-1) = \lambda(v-1) \quad (2.3)$$

ВІВ-схема називається симетричною, якщо:

$$v = a \quad (2.4)$$

Отже, згідно з (1):

$$k = r \quad (2.5)$$

Для симетричної ВІВ-схеми згідно з (2.3) і (2.5) справедливе співвідношення:

$$k(k-1) = \lambda(v-1) \quad (2.6)$$

У частковому випадку, коли  $\lambda = 1$ , симетрична ВІВ-схема називається скінченною проєктивною площиною  $(k-1)$ -го порядку.

Симетрична ВІВ-схема називається циклічною, коли для неї існує циклічний автоморфізм  $\alpha$ , який полягає в тому, що зміна індекс  $j \rightarrow j+1(\text{mod } v)$  при всіх  $k$  елементах  $j$ -го блоку ВІВ-схеми приводить до утворення множини елементів  $(j+1(\text{mod } v))$ -го блоку цієї ж ВІВ-схеми, так що множина елементів окремого блоку повністю визначає всю ВІВ-схему.

Тепер послідовності чисел  $(k_1, k_2, \dots, k_n)$ , яка утворює ІКВ, поставимо у відповідність суму ІКВ послідовностей:

$$S_n = \frac{n(n-1)}{R} + 1 \quad (2.7)$$

$$B^{(1)} = (b_1^{(1)}, b_2^{(1)}, \dots, b_n^{(1)}) \quad (2.8)$$

$$B^{(2)} = (b_1^{(2)}, b_2^{(2)}, \dots, b_n^{(2)})$$

$\square$

$$B^{(S_n)} = (b_1^{(S_n)}, b_2^{(S_n)}, \dots, b_n^{(S_n)})$$

елементи яких визначаються за формулою:

$$x_j - 1 \equiv \sum_{i=1}^j k_i \pmod{S_n}, j = 1, 2, \dots, n \quad (2.9)$$

Для побудови циклічного коду на основі ІКВ виділимо рядок із  $S_n$  пронумерованих у зростаючому порядку кліток одновимірного масиву та заповнимо інформаційними «одиницями» клітки, номери яких збігаються з числами, визначеними з ІКВ за формулою (2.9). У клітки, що залишилися

незаповненими, занесемо «нулі». Утворена послідовність одиниць і нулів є  $S_n$  розрядною кодовою комбінацією, циклічним зсувом якої можна одержати й всі останні дозволені комбінації для даного циклічного коду.

### 2.2.2. Синтез кодових комбінацій на основі ІКВ для баркероподібних послідовностей

Спираючись на вищесказане, давайте побудуємо таблицю кодових комбінацій на основі ІКВ (1, 1, 1, 2, 2, 1, 3, 4) з параметрами  $n = 8$  та  $R = 4$  (і відповідно  $S_n = 15$ ). Дана ІКВ вибрана не випадково, та про це трохи згодом. Згідно формули (2.8) ми отримуємо:

$$x_1 = \sum_{i=1}^1 k_i (\text{mod } S_n) + 1 = 2 \quad (2.10)$$

$$x_2 = \sum_{i=1}^2 k_i (\text{mod } S_n) + 1 = 3$$

□

$$x_8 = \sum_{i=1}^8 k_i (\text{mod } S_n) + 1 = 1$$

Проте дана таблиця кодових комбінацій не буде повної без «нульової» комбінації. Таким чином, ми отримуємо  $S_n + 1 = 16$  кодових комбінацій. Повна таблиця кодових комбінацій баркероподібних послідовностей на базі ІКВ (1, 1, 1, 2, 2, 1, 3, 4) відображена в табл. 2.1.

Циклічний ІКВ-код є блочним кодом, тобто кількість бітів в одному блоці є величиною сталою. Тож давайте з'ясуємо скільки бітів ми можемо кодувати в одному блоці:

$$n = \log_2(S_n + 1) = \log_2 16 = 4. \quad (2.11)$$

Таблиця 2.1.

## Кодові комбінації баркероподібних послідовностей

Число	Код
0000	0000000000000000
0001	0001001101011111
0010	001001101011110
0010	010011010111100
0011	100110101111000
0100	001101011110001
0101	011010111100010
0110	110101111000100
0111	101011110001001
1000	010111100010011
1001	101111000100110
1011	011110001001101
1100	111100010011010
1101	111000100110101
1110	110001001101011
1111	100010011010111

Після побудови кодових комбінацій давайте з'ясуємо можливості даного коду, а саме визначимо мінімальну кодову відстань, коректувальну та виправну здатність.

Легко виявити, що кожна з  $S_n (S_n - 1)/2$  різних пар кодових комбінацій містить точно  $R$  із  $n$  одиничних символів в однойменних розрядах, що впливає з властивостей ІКВ.

Решта  $n - R$  символів однієї і стільки ж іншої кодової комбінації відрізняються від символів, що містяться в однойменних розрядах. Тому мінімальна кодова відстань для даного коду визначається як:

$$d_{\min} = 2(n - R) \quad (2.12)$$

Число помилок, які можна виявити  $t_d$ , та число помилок, що можна виправити  $t_c$  за допомогою коректувального коду, зв'язані з мінімальною кодовою відстанню залежностями, і відповідно становлять:

$$t_d \leq d_{\min} - 1 \quad (2.13)$$

$$t_c \leq \frac{d_{\min} - 1}{2} \quad (2.14)$$

У розглянутому прикладі значення  $n = 8$  та  $R = 4$  вибиралися не випадково і цьому є пояснення. Тож давайте розглянемо питання про можливість встановлення співвідношення між  $n$  і  $R$ , при якому розглянутий код набуває додаткових переваг. Як ми можемо побачити з формул (4.12), (4.13) та (4.11), коректувальна здатність коду збільшується зі збільшення мінімальної кодової відстані і, по суті, дана задача зводиться до забезпечення найбільшої мінімальної кодової відстані  $d_{\min}$  при якомога меншому значенні  $S_n$ .

Спробуймо виразити вищесказане математично. Коректувальна спроможність коду зростає зі збільшенням різниці  $\sigma = n - R$ . Після підстановки  $R = \sigma - n$  в (2.7) ми отримуємо наступне рівняння відносно  $\sigma$ :

$$\sigma = n - \frac{n \times (n - 1)}{S_n - 1} \quad (2.15)$$

Після нескладних математичних маніпуляцій ми отримуємо:

$$S_n = 2n \quad (2.16)$$

Після взаємної підстановки і розв'язання рівняння в цілих числах знаходимо співвідношення між  $n$  і  $R$ , при яких код набуває здатності виявляти та виправляти максимально можливу кількість помилок, а саме:

$$R = \begin{cases} \frac{n}{2}, & \text{для парних значень } n \\ \frac{n-1}{2}, & \text{для непарних значень } n \end{cases} \quad (2.17)$$

Знайдене нами співвідношення має надзвичайно важливе значення для вирішення задачі побудови оптимального циклічного ІКВ-коду, яка, по суті зводиться до вибору ІКВ з потрібними параметрами.

## РОЗДІЛ 3. ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

### 3.1. РОЗРОБКА ПРОГРАМНОГО РІШЕННЯ

#### 3.1.1. Програма синтезу завадостійких баркероподібних послідовностей

Меню програмного продукту на мові C++ побудови завадостійкої квазібаркерної кодової послідовності наведено на рис. 3.1.

Для запуску програми синтезу завадостійких баркероподібних послідовностей необхідно задати:

- параметри ІКВ;
- назву файлу, який буде закодований за допомогою завадостійкої баркероподібної кодової послідовності.

Далі можна запускати процес синтезу завадостійкої баркероподібної кодової послідовності.

Далі необхідно натиснути кнопку <Кодування>.

Для імітації завад натиснути кнопку <Генерування помилок>.

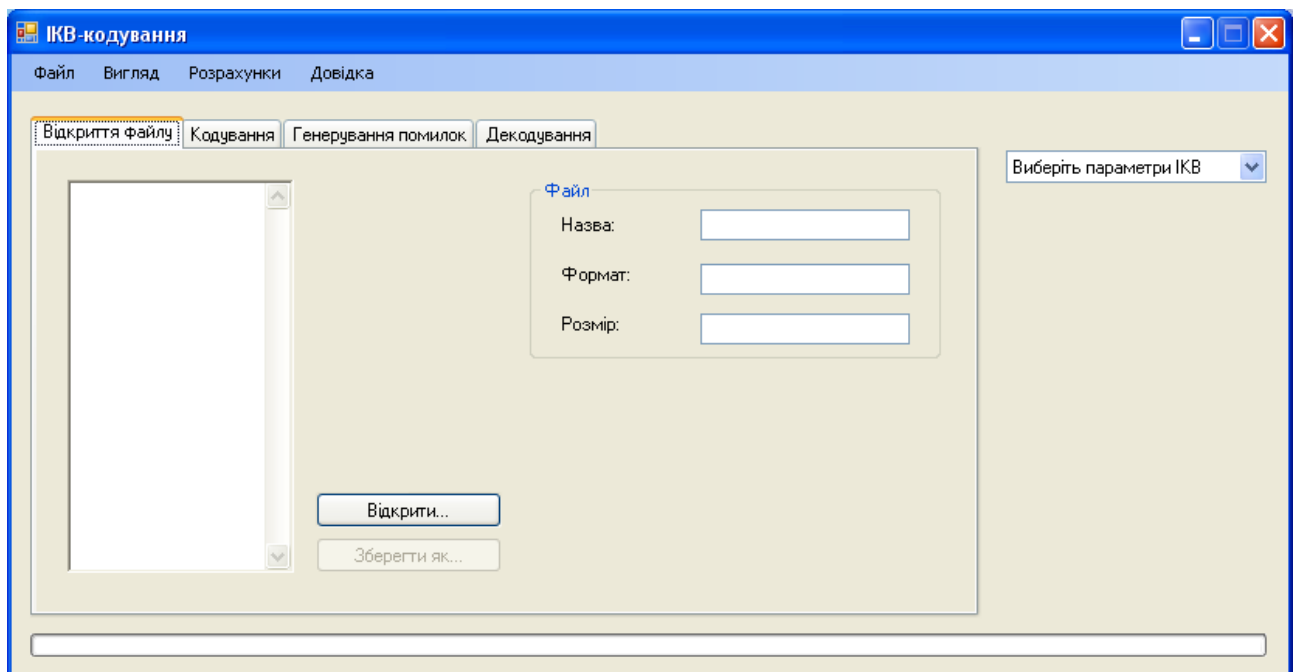


Рис. 3.1. Вікно програми побудови завадостійкої баркероподібної послідовності

Для декодування завадостійких баркероподібних послідовностей необхідно натиснути кнопку <Декодування>. Результат виправлених завадостійких

баркероподібних послідовностей буде представлений у вікні та збережений у вибраній файл.

### 3.1.2. Кориговальна спроможність баркероподібних послідовностей

Для ілюстрації розглянемо приклад розрахунку кориговальної спроможності завадостійких кодів, побудованих на основі ІКВ з параметрами:

$$1) n = 6, R = 1;$$

$$2) n = 15, R = (n - 1) / 2; \quad (3.1)$$

$$3) n = 16, R = n / 2.$$

Довжина дозволених кодових комбінацій для всіх трьох кодів однакова:  $S_n = 31$ . Максимальне число помилок, які можна виявляти або виправляти за допомогою першого з вказаних кодів, дорівнює відповідно 9 та 4, в той час як кожен із інших двох кодів дає змогу виявляти до 15 і виправляти до 7 помилок. В принципі, будь-яка ІКВ може стати основою для синтезу завадостійкого коду. Однак слід відзначити, що з допомогою ІКВ, параметри яких зв'язані даним відношенням, можна будувати найбільш ефективні завадостійкі коди.

А також продовжуючи описувати використання бази ІКВ (1, 1, 1, 2, 2, 1, 3, 4), визначаємо мінімальну кодову відстань та коректувальну здатність коду:

$$d_{\min} = 2 \cdot (8 - 4) = 8 \quad (3.2)$$

$$t_d = 8 - 1 = 7$$

$$t_c = \frac{8 - 1}{2} = 3$$

### 3.1.3. Порівняльна характеристика кодів ІКВ та БЧХ

Завадостійка баркероподібна система на основі ІКВ, об'єкт дослідження проекту, здатна закодувати 4 розряди масиву даних в послідовність довжиною 15 та при цьому виправляти до 3 позицій послідовності (або 20%), тоді як завадостійка система на основі кодів БЧХ при цій же довжині послідовності та кількості інформаційних розрядів дозволяє редагувати до 4 позицій (або

26,67% ). Для більшої наочності порівняльні характеристики цих кодових послідовностей відобразимо в табл. 3.1.

Таблиця 3.1.

## Порівняльна характеристика кодових послідовностей

Показник	Числове значення (бали)	
	ІКВ	БЧХ
Надлишковість коду	11	11
Коректувальна здатність	3	4
Потужність коду	16	16
Швидкість кодування даних, МБ/с	2,015	1,936
Швидкість декодування даних, МБ/с	0,608	1,316
Ресурсозатратність системи, МБ	80	120
Захищеність, (1-10)	9	4

Другим параметром по значущості, який визначає доцільність реалізації тої чи іншої розробки, є швидкість процесу кодування та декодування. Експериментально було встановлено, що обидві системи видають приблизно однакові результати під час кодування масиву даних, проте процес декодування даних за допомогою ІКВ виконується більш, ніж двічі швидше за аналог. Ми живемо в час, коли графічні дані можуть займати десятки мегабайт або навіть гігабайт, швидкість виконання операцій є важливим, а іноді й визначальним, показником.

Ще додає «плюсів» ІКВ параметр, що описує ресурсозатратність. Під час експериментального кодування та декодування 10-мегабайтного файлу БЧХ займає 120МБ пам'яті комп'ютера, тоді як ІКВ – лише 80МБ .

Наступний параметр, на жаль, немає чіткого числового вираження і для визначення доводиться користатися суб'єктивними оцінками – мова йде про захищеність даних. ІКВ базується на добре вивчених властивості в'язанок, що

мають складну структуру для відтворення з боку зловмисників. До того ж ми можемо використовувати різні ІКВ, результати побудови сортувати на різний смак. БЧХ базоване на полях Галуа, хоча й теоретичного складніша, що збільшує кількість операцій, проте прогнозованіша, адже число варіантів многочленів є обмеженим та вони всі вже відомі.

## 3.2. ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА

### 3.2.1. Практичне застосування завадостійкої баркероподібної системи кодування на основі ІКВ

- 1) вікно програми при запуску програми (рис. 3.2):

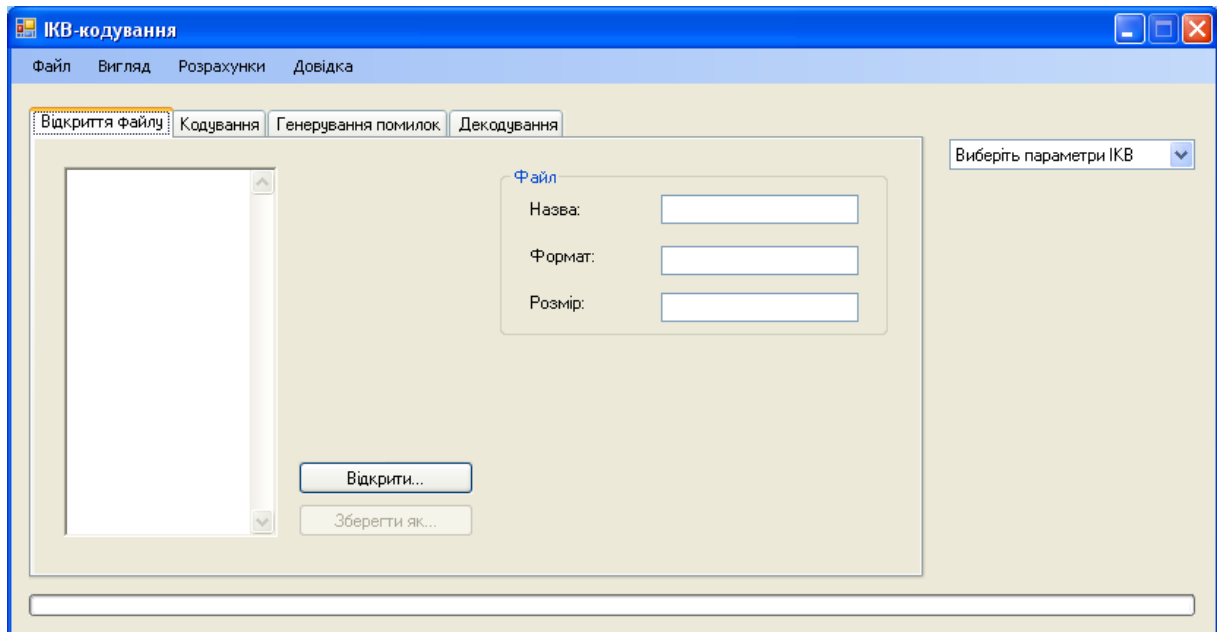


Рис. 3.2. Вікно програми побудови завадостійкої баркероподібної послідовності

- 2) відкриття файлу для роботи (рис. 3.3):

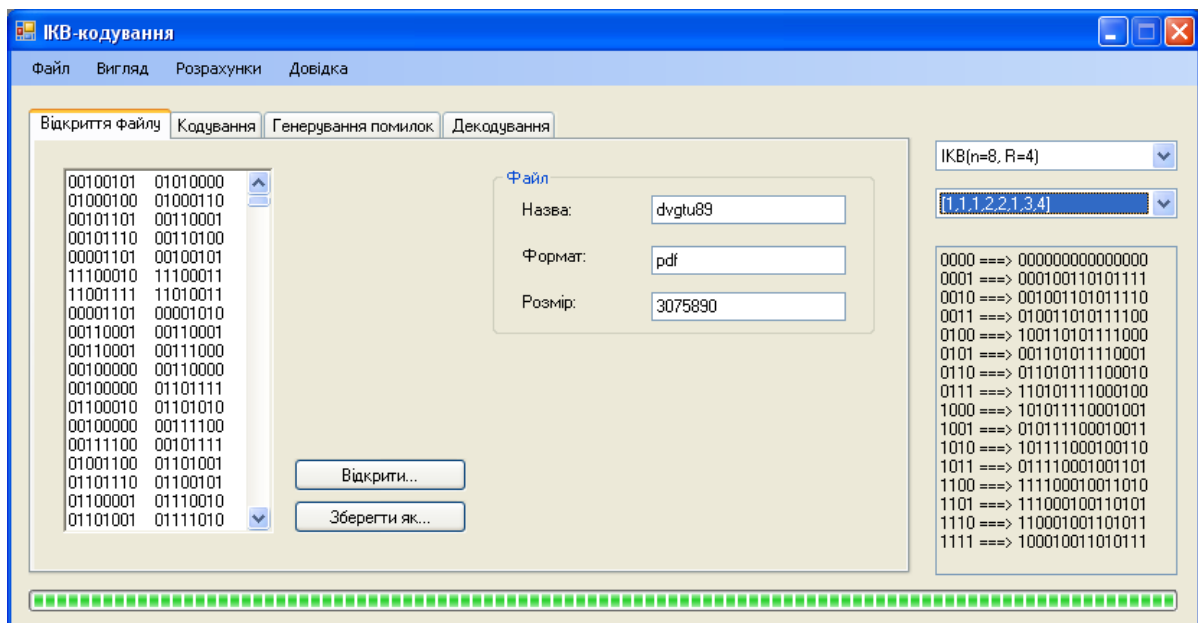


Рис. 3.3. Відкриття файлу для роботи

3) кодування файлу згідно визначених правил (рис. 3.4):

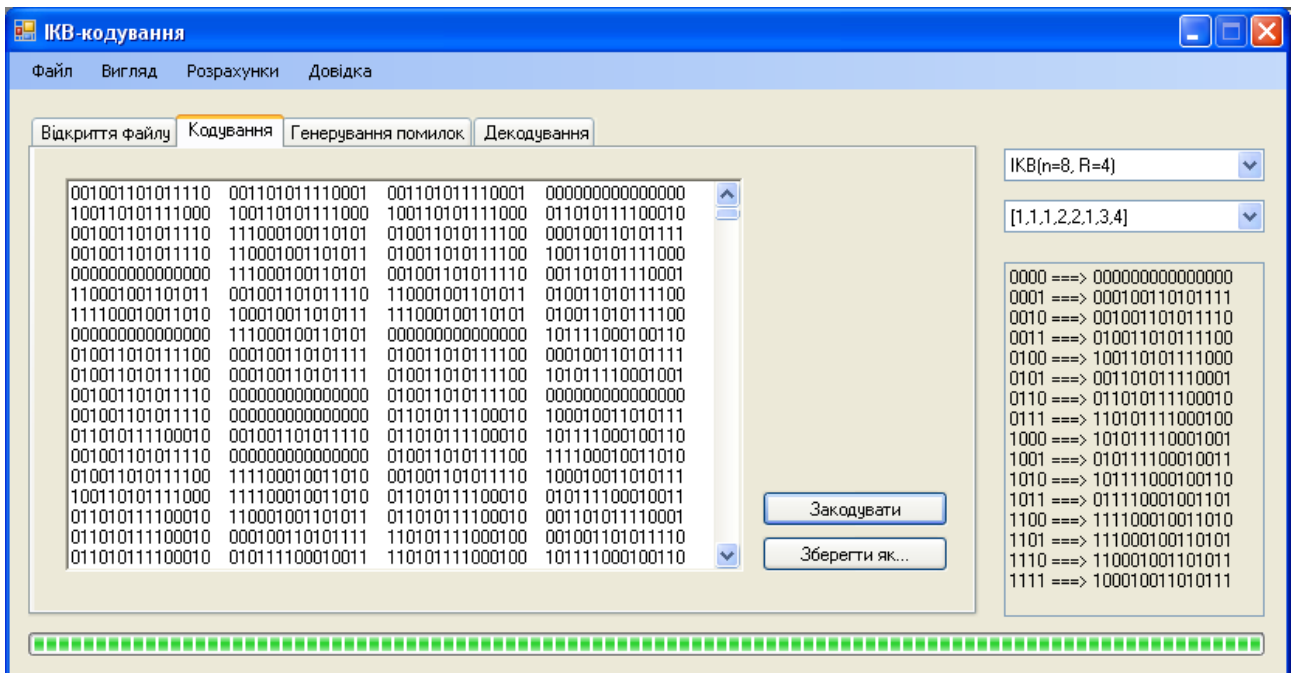


Рис. 3.4. Кодування файлу згідно визначених правил

4) штучне генерування помилок для демонстрації роботи програми (рис. 3.5):

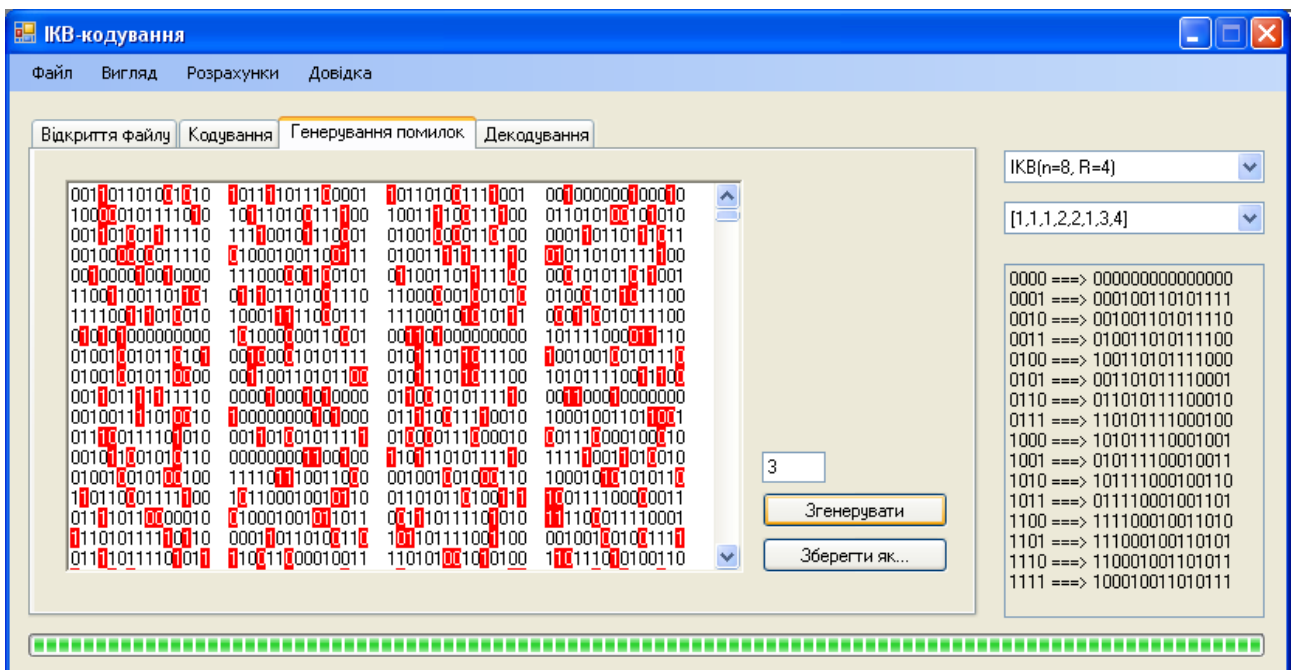


Рис. 3.5. Штучне генерування помилок для демонстрації роботи програми

5) декодування файлу для перегляду (рис. 3.6):

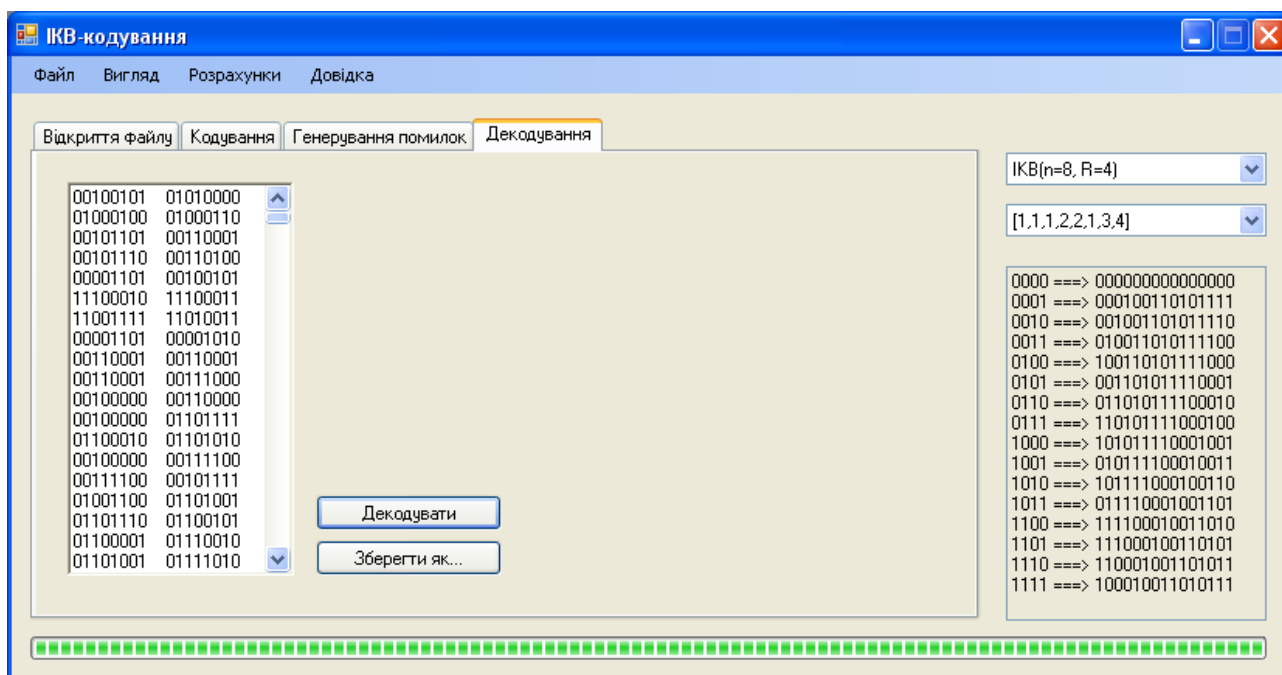


Рис. 3.6. Декодування файлу для перегляду

б) розрахунок параметрів коду в залежності від вибраного КВ (рис. 3.7):

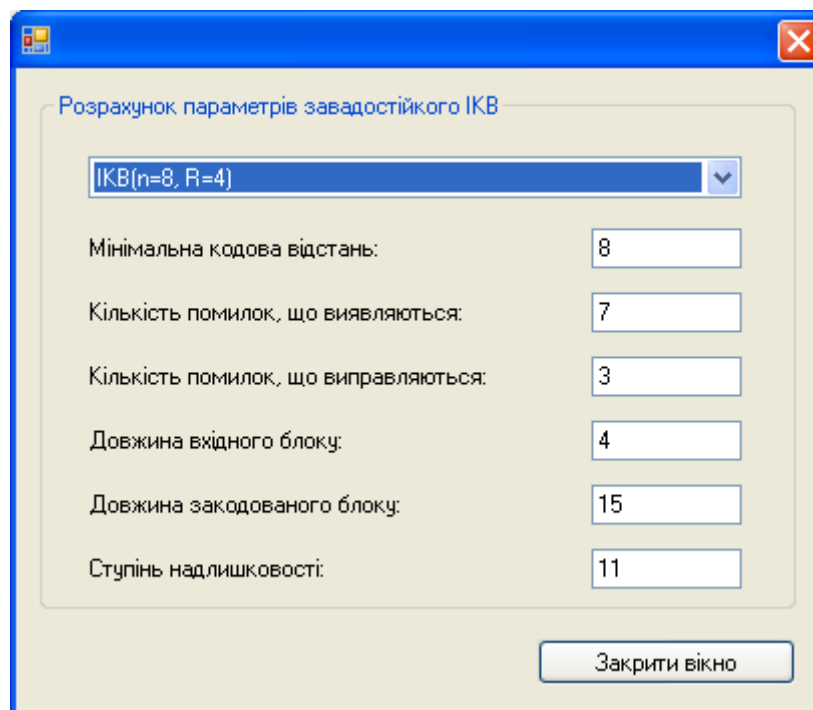


Рис. 3.7. Розрахунок параметрів коду в залежності від вибраного КВ

7) розрахунок розміру файлів в залежності від вибраного ІКВ (рис. 3.8):

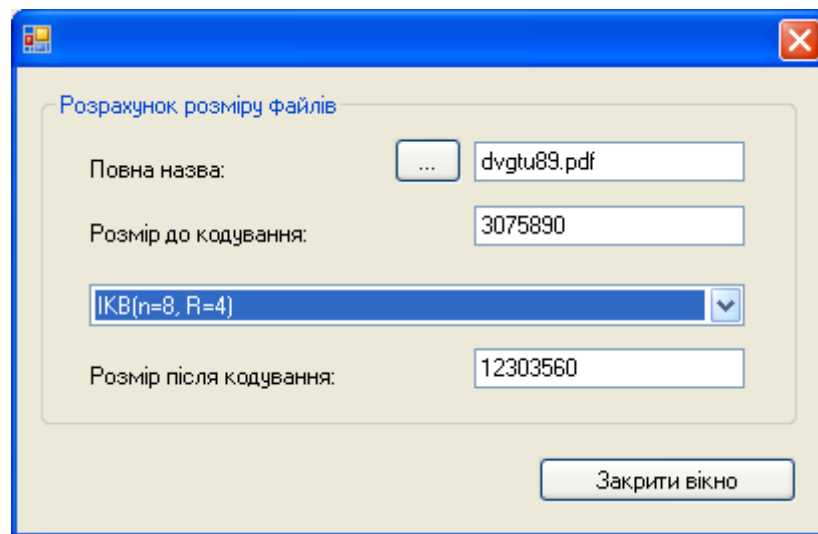


Рис. 3.8. Розрахунок розміру файлів в залежності від вибраного ІКВ

Основним завданням нашої роботи є захист даних і відповідно завдання завадостійкості (для неї потрібно високу коригувальну здатність). Спираючись на вищесказане, обирається використання завадостійкої баркероподібної послідовності на основі ІКВ як рішення з добрими показниками коригувальної здатності та набагато кращими показниками по захисту.

## ВИСНОВКИ

1. Наведений метод синтезу завадостійких баркероподібних послідовностей, що дозволило за короткий час побудувати ці послідовності.
2. Досліджена кількість знаходження та виправлення помилок для завадостійких баркероподібних послідовностей.
3. Створений програмний продукт синтезу завадостійких баркероподібних послідовностей має практичне значення та є конкурентноздатним.

**ПЕРЕЛІК ПОСИЛАНЬ**

1. А.с. 1073717 СССР. Способ сжатия фазоманипулированных сигналов и устройство для его осуществления / Н.П. Лавлинский. – 15.10.1983.
2. Арслан Х., Чен Чж. Н., Бендетто М. Сверхширокополосная беспроводная связь // М.: Техносфера, 2012 – 640 с.: ил.
3. Бабков В.Ю., Вознюк М.А., Никитин А.Н., Сиверс М.А. Системы связи с кодовым разделением каналов. СПб: Санкт-Петербургский государственный университет телекоммуникаций им. Бонч-Бруевича, 1999 – 121 с.
4. Банкет В.Л. Композитные коды Баркера / В. Л. Банкет, М. С. Токарь // Цифрові технології. – 2007. –№2. – С.8-17.
5. Берлин А. Н. Коммутация в системах и сетях связи. – М. : Эко-Трендз, 2006. – 344 с.
6. Вакман Д.Е. Регулярный метод синтеза Ф. М. сигналов / Вакман Д.Е. – Изд-во «Советское радио», 1967. – 96 с.
7. Варакин Л. Е. Системы связи с шумоподобными сигналами / Л. Е. Варакин. – М.: Радио и связь, 1985. – 384с.
8. Васин В.А., Калмыков В.В. Радиосистемы передачи информации: учеб. пособие для вузов. – М. : Горячая линия-Телеком, 2005. – 472 с.
9. Герасимов И. В., Сафьянников Н. М., Якимовский Д. О. Сложнофункциональные блоки смешанных систем на кристалле: автоматизация функционального проектирования: Монография / под ред. И. В. Герасимова. – СПб.: Изд-во «ЭЛМОР», 2012. – 237 с.
10. Голдсмит А. Беспроводные коммуникации // М.: Техносфера, 2011 – 904 с.: ил.
11. Гольдштейн Б.С., Соколов Н.А., Яновский Г.Г. Сети связи: Учебник для ВУЗов. СПб.:БХВ-Петербург, 2010 – 400 с.
12. Гордиенко В.Н., Крухмалев В.В., Моченов А.Д., Шарафутдинов Р.М. Оптические телекоммуникационные системы. Учебник для вузов /. Под ред. В.Н. Гордиенко. – М.: Горячая линия – Телеком, 2011 – 368 с.

13. Гришенцев А. Ю., Коробейников А. Г. Методы и модели цифровой обработки изображений – Монография. СПб: Изд-во Политехн. ун-та, 2014. – 190 с.: ил.
14. Гришенцев А. Ю., Коробейников А. Г. Понижение размерности пространства при корреляции и свертке цифровых сигналов // Изв. вузов. Приборостроение. 2016. Т. 59, № 3. С. 211—218.
15. Гришенцев А.Ю., Коробейников А. Г., Величко Е. Н., Непомнящая Э. К., Розов С. В. Синтез бинарных матриц для формирования сигналов широкополосной связи // Радиотехника, 2015, №9, С.: 51-58.
16. Дворкович В. П., Дворкович А. В. Оконные функции для гармонического анализа сигналов // М.: Техносфера, 2014. – 112 с.: ил.
17. Дятлов А. П., Кульбикаян Б. Х. Корреляционная обработка широкополосных сигналов в автоматизированных комплексах радиомониторинга // М.: Горячая линия–Телеком, 2014. – 332 с.: ил.
18. Жаринов О. О., Жаринов И. О. Синтез и оптимизация вычислительных алгоритмов обработки сигнала на основе корреляционно-экстремального метода в электрокардиографии высокого разрешения // Известия высших учебных заведений. Приборостроение. 2014. Т. 57. № 1. С. 29-38.
19. Зарецкий С.В. Метод подавления боковых лепестков сигнала в виде бинарной фазоманипулированной последовательности / С.В. Зарецкий, Е.А. Сельменев // Электронный научный журнал «Исследовано в России». – Режим доступа : <http://zhurnal.apc.relam.ru/articles/2008/078.pdf>. p. 846 -852.
20. Ипатов В. Широкополосные системы и кодовое разделение сигналов. Принципы и приложения. М.: Техносфера, 2007. – 488 с.
21. Кудряшов Б. Д. Основы теории кодирования: учеб. пособие // СПб.: БХВ-
22. Кулаков В.Г., Гаранин М.В. Информационная безопасность телекоммуникационных систем. (Технические аспекты) – М.: Радио и связь, 2004. – 304 с.
23. Лавлинский Н.П. Исследования свойств АКФ неквантованных по длительности непериодических бинарных ФМн сигналов / Н.П. Лавлинский //

Автоматизированные системы сбора и переработки гидрофизической информации. – Севастополь, 1985. – 19 с.

24. Лавлинский Н.П. О потенциальной помехоустойчивости широкополосных сигналов / Н.П. Лавлинский // Отчет по хоздоговорной теме «Посейдон». – Севастополь, 1984. – 125 с.

25. Лавлинский Н.П. Повышение безопасности связи путем применения широкополосных фазоманипулированных сигналов / Н.П. Лавлинский, Н.И. Галлини // Системи обробки інформації. – 2010. – Вид. 3 (84) – С. 42-46.

26. Лавлинский Н.П. Синтез «баркеровских» сигналов по автокорреляционной функции для современных систем связи и локации / Н.П. Лавлинский, Е.А. Мусатова // Тезисы докладов V Всеукраинской научно-практической конференции. – Симферополь, 2010. – С. 54-55.

27. Лавлинский Н.П. Синтез одного класса сигналов по автокорреляционной функции» в отчете: Адаптивные обучаемые и самообучающиеся системы / Н.П. Лавлинский // Координационный план АН СССР на 1976 – 1980 гг. шифр 1.12.1 № гос. Регистрации 78017018 за 1980 г. – Томск : СФТИ, 1980. – С.147-188.

28. Ломовицкий В.В., Михайлов А.И. Основы построения систем и сетей передачи информации: учеб. пособие для вузов. – М. : Горячая линия - Телеком, 2005. – 382 с.

29. Макаров С. Б., Певцов Н. В., Попов Е. А., Сиверс М. А. Телекоммуникационные технологии. Введение в технологии GSM : учеб. пособие для вузов. – 2-е изд., испр. – М.: Академия, 2008 – 256 с.

30. Маковеева М.М., Шинаков Ю.С. Системы связи с подвижными объектами. Учебное пособие. М.: Радио и связь, 202 – 440 с.

31. Малахов В.П. Моделювання багатоканальної системи збирання інформації / В.П. Малахов, М.І. Шпинковська // Зб. наук. праць УДМТУ. - Миколаїв, 2002.-№ 5. – с.118-123.

32. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. СПб: Питер, 2011 – 944 с.

33. Оппенгейм А., Шафер Р. Цифровая обработка сигналов. 2-е издание, испр. М.: Техносфера, 2009. – 856 с. Петербург, 2016 – 400 с.: ил.
34. Ратынский М.В. Основы сотовой связи./ Под ред. Д. Б. Зимина. – М.: Радио и связь – 248 с.
35. Режим доступа : <http://chem.net/sprav/sprav111.php> Barker, R. H. (1953). Group Synchronizing of Binary Digital Sequences. Communication Theory. – London: Butterworth. – P. 273-287.
36. Родимов А. П., Поповский В. В. Статистическая теория поляризационновременной обработки сигналов и помех. – М.: Радио и связь, 1984. 272 с.: ил.
37. Седжевик Р. Алгоритмы на C++.: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2011. – 1056 с.
38. Скляр Б. Цифровая связь. Теоретические основы и практическое применение / Скляр Б. ; пер. с англ. – М.: Издательский дом «Вильямс», 2003. – 1104 с.
39. Хрусталева Д. А. Новейшее руководство по сотовой связи. – М.: СОЛОН-Пресс, 2004. – 176 с.

# ДОДАТОК А. КОД ПРОГРАМИ СИНТЕЗУ ЗАВАДОСТІЙКИХ ПОСЛІДОВНОСТЕЙ БАРКЕРА

## /\*Form1.h\*/

```

#pragma once
#include <iostream>

#include "HGlobalVariableFunction.h"
#include "TIdealRingBundle.h"
#include "TAllowedCombination.h"
#include "TCodingDecoding.h"
#include "TGenerationError.h"
#include "Form2.h"
#include "Form3.h"

namespace bundleCodingDecoding {

using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;
using namespace System::Runtime::InteropServices;

/// <summary>
/// Summary for Form1
/// </summary>
public ref class Form1 : public System::Windows::Forms::Form
{
public:
    Form1(void)
    {
        InitializeComponent();
        //
        //TODO: Add the constructor code here
        //
    }

protected:
    /// <summary>
    /// Clean up any resources being used.
    /// </summary>
    ~Form1()
    {
        if (components)
        {
            delete components;
        }
    }

private: System::Windows::Forms::MenuStrip^ menuStrip1;

```

```

protected:
private: System::Windows::Forms::ToolStripMenuItem^ файлToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^ відкритиToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^ зберегтиToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^ вихіднийToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^ закодованийToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^ закодованийЗПомилкамиToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^ декодованийToolStripMenuItem;
private: System::Windows::Forms::ToolStripSeparator^ toolStripSeparator1;
private: System::Windows::Forms::ToolStripMenuItem^ закритиToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^ виглядToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^ очиститиВсіПоляToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^ розрахункиToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^ параметриКодуToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^ розміриФайлівToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^ проПрограмуToolStripMenuItem;
private: System::Windows::Forms::TabControl^ tabControl1;
private: System::Windows::Forms::TabPage^ tabPage1;
private: System::Windows::Forms::TabPage^ tabPage2;
private: System::Windows::Forms::TabPage^ tabPage3;
private: System::Windows::Forms::TabPage^ tabPage4;
private: System::Windows::Forms::RichTextBox^ richTextBox1;
private: System::Windows::Forms::GroupBox^ groupBox1;
private: System::Windows::Forms::TextBox^ textBox2;
private: System::Windows::Forms::TextBox^ textBox3;
private: System::Windows::Forms::TextBox^ textBox1;
private: System::Windows::Forms::Label^ label3;
private: System::Windows::Forms::Label^ label2;
private: System::Windows::Forms::Label^ label1;
private: System::Windows::Forms::Button^ button2;
private: System::Windows::Forms::Button^ button1;
private: System::Windows::Forms::ComboBox^ comboBox1;
private: System::Windows::Forms::ComboBox^ comboBox2;
private: System::Windows::Forms::TextBox^ textBox4;
private: System::Windows::Forms::OpenFileDialog^ openFileDialog1;
private: System::Windows::Forms::RichTextBox^ richTextBox2;
private: System::Windows::Forms::Button^ button4;
private: System::Windows::Forms::Button^ button3;

private: System::Windows::Forms::Button^ button6;
private: System::Windows::Forms::TextBox^ textBox5;
private: System::Windows::Forms::Button^ button5;
private: System::Windows::Forms::RichTextBox^ richTextBox3;
private: System::Windows::Forms::ToolStripMenuItem^ проПрограмуToolStripMenuItem1;
private: System::Windows::Forms::RichTextBox^ richTextBox4;
private: System::Windows::Forms::ProgressBar^ progressBar1;
private: System::Windows::Forms::Button^ button8;
private: System::Windows::Forms::Button^ button7;

private:

```

```

/// <summary>
/// Required designer variable.
/// </summary>
System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
void InitializeComponent(void)
{
    this->menuStrip1 = (gcnew System::Windows::Forms::MenuStrip());
    this->файлToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());
    this->відкритиToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());
    this->зберегтиToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());
    this->вихіднийToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());
    this->закодованийToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());
    this->закодованийЗПомилкамиToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());
    this->декодованийToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());
    this->toolStripSeparator1 = (gcnew System::Windows::Forms::ToolStripSeparator());
    this->закритиToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());
    this->виглядToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());
    this->очиститиВсіПоляToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());
    this->розрахункиToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());
    this->параметриКодуToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());
    this->розміриФайлівToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());
    this->проПрограмуToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());
    this->проПрограмуToolStripMenuItem1 = (gcnew System::Windows::Forms::ToolStripMenuItem());
    this->tabControl1 = (gcnew System::Windows::Forms::TabControl());
    this->tabPage1 = (gcnew System::Windows::Forms::TabPage());
    this->groupBox1 = (gcnew System::Windows::Forms::GroupBox());
    this->textBox2 = (gcnew System::Windows::Forms::TextBox());
    this->textBox3 = (gcnew System::Windows::Forms::TextBox());
    this->textBox1 = (gcnew System::Windows::Forms::TextBox());
    this->label3 = (gcnew System::Windows::Forms::Label());
    this->label2 = (gcnew System::Windows::Forms::Label());
    this->label1 = (gcnew System::Windows::Forms::Label());
    this->button2 = (gcnew System::Windows::Forms::Button());
    this->button1 = (gcnew System::Windows::Forms::Button());
    this->richTextBox1 = (gcnew System::Windows::Forms::RichTextBox());
    this->tabPage2 = (gcnew System::Windows::Forms::TabPage());
    this->button4 = (gcnew System::Windows::Forms::Button());
    this->button3 = (gcnew System::Windows::Forms::Button());
    this->richTextBox2 = (gcnew System::Windows::Forms::RichTextBox());
    this->tabPage3 = (gcnew System::Windows::Forms::TabPage());
    this->button6 = (gcnew System::Windows::Forms::Button());
    this->textBox5 = (gcnew System::Windows::Forms::TextBox());
    this->button5 = (gcnew System::Windows::Forms::Button());
    this->richTextBox3 = (gcnew System::Windows::Forms::RichTextBox());
    this->tabPage4 = (gcnew System::Windows::Forms::TabPage());
    this->button8 = (gcnew System::Windows::Forms::Button());
}

```

```

this->button7 = (gcnew System::Windows::Forms::Button());
this->richTextBox4 = (gcnew System::Windows::Forms::RichTextBox());
this->comboBox1 = (gcnew System::Windows::Forms::ComboBox());
this->comboBox2 = (gcnew System::Windows::Forms::ComboBox());
this->textBox4 = (gcnew System::Windows::Forms::TextBox());
this->openFileDialog1 = (gcnew System::Windows::Forms::OpenFileDialog());
this->progressBar1 = (gcnew System::Windows::Forms::ProgressBar());
this->menuStrip1->SuspendLayout();
this->tabControl1->SuspendLayout();
this->tabPage1->SuspendLayout();
this->groupBox1->SuspendLayout();
this->tabPage2->SuspendLayout();
this->tabPage3->SuspendLayout();
this->tabPage4->SuspendLayout();
this->SuspendLayout();
//
// menuStrip1
//
this->menuStrip1->Items->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(4) { this-
>файлToolStripMenuItem,
        this->виглядToolStripMenuItem, this->розрахункиToolStripMenuItem, this->проПорогамуToolStripMenuItem});
this->menuStrip1->Location = System::Drawing::Point(0, 0);
this->menuStrip1->Name = L"menuStrip1";
this->menuStrip1->Size = System::Drawing::Size(823, 24);
this->menuStrip1->TabIndex = 0;
this->menuStrip1->Text = L"menuStrip1";
//
// файлToolStripMenuItem
//
this->файлToolStripMenuItem->DropDownItems->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(4) { this-
>відкритиToolStripMenuItem,
        this->зберегтиToolStripMenuItem, this->toolStripSeparator1, this->закритиToolStripMenuItem});
this->файлToolStripMenuItem->Name = L"файлToolStripMenuItem";
this->файлToolStripMenuItem->Size = System::Drawing::Size(45, 20);
this->файлToolStripMenuItem->Text = L"Файл";
//
// відкритиToolStripMenuItem
//
this->відкритиToolStripMenuItem->Name = L"відкритиToolStripMenuItem";
this->відкритиToolStripMenuItem->Size = System::Drawing::Size(142, 22);
this->відкритиToolStripMenuItem->Text = L"Відкрити...";
this->відкритиToolStripMenuItem->Click += gcnew System::EventHandler(this, &Form1::відкритиToolStripMenuItem_Click);
//
// зберегтиToolStripMenuItem
//
this->зберегтиToolStripMenuItem->DropDownItems->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(4)
{ this->вихіднийToolStripMenuItem,
        this->закодованийToolStripMenuItem, this->закодованийЗПомилкамиToolStripMenuItem, this-
>декодованийToolStripMenuItem});
this->зберегтиToolStripMenuItem->Name = L"зберегтиToolStripMenuItem";
this->зберегтиToolStripMenuItem->Size = System::Drawing::Size(142, 22);
this->зберегтиToolStripMenuItem->Text = L"Зберегти";

```

```

//
// вихіднийToolStripMenuItem
//
this->вихіднийToolStripMenuItem->Name = L"вихіднийToolStripMenuItem";
this->вихіднийToolStripMenuItem->Size = System::Drawing::Size(228, 22);
this->вихіднийToolStripMenuItem->Text = L"вихідний...";
//
// закодованийToolStripMenuItem
//
this->закодованийToolStripMenuItem->Name = L"закодованийToolStripMenuItem";
this->закодованийToolStripMenuItem->Size = System::Drawing::Size(228, 22);
this->закодованийToolStripMenuItem->Text = L"закодований...";
//
// закодованийЗПомилкамиToolStripMenuItem
//
this->закодованийЗПомилкамиToolStripMenuItem->Name = L"закодованийЗПомилкамиToolStripMenuItem";
this->закодованийЗПомилкамиToolStripMenuItem->Size = System::Drawing::Size(228, 22);
this->закодованийЗПомилкамиToolStripMenuItem->Text = L"закодований з помилками...";
//
// декодованийToolStripMenuItem
//
this->декодованийToolStripMenuItem->Name = L"декодованийToolStripMenuItem";
this->декодованийToolStripMenuItem->Size = System::Drawing::Size(228, 22);
this->декодованийToolStripMenuItem->Text = L"декодований...";
//
// toolStripSeparator1
//
this->toolStripSeparator1->Name = L"toolStripSeparator1";
this->toolStripSeparator1->Size = System::Drawing::Size(139, 6);
//
// закритиToolStripMenuItem
//
this->закритиToolStripMenuItem->Name = L"закритиToolStripMenuItem";
this->закритиToolStripMenuItem->Size = System::Drawing::Size(142, 22);
this->закритиToolStripMenuItem->Text = L"Закрити";
this->закритиToolStripMenuItem->Click += gcnew System::EventHandler(this, &Form1::закритиToolStripMenuItem_Click);
//
// виглядToolStripMenuItem
//
this->виглядToolStripMenuItem->DropDownItems->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(1)
{this->очиститиВсіПоляToolStripMenuItem});
this->виглядToolStripMenuItem->Name = L"виглядToolStripMenuItem";
this->виглядToolStripMenuItem->Size = System::Drawing::Size(55, 20);
this->виглядToolStripMenuItem->Text = L"Вигляд";
//
// очиститиВсіПоляToolStripMenuItem
//
this->очиститиВсіПоляToolStripMenuItem->Name = L"очиститиВсіПоляToolStripMenuItem";
this->очиститиВсіПоляToolStripMenuItem->Size = System::Drawing::Size(177, 22);
this->очиститиВсіПоляToolStripMenuItem->Text = L"Очистити всі поля";
this->очиститиВсіПоляToolStripMenuItem->Click += gcnew System::EventHandler(this,
&Form1::очиститиВсіПоляToolStripMenuItem_Click);

```

```

//
// розрахункиToolStripMenuItem
//
this->розрахункиToolStripMenuItem->DropDownItems->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^
>(2) { this->параметриКодуToolStripMenuItem,
    this->розміриФайлівToolStripMenuItem});
this->розрахункиToolStripMenuItem->Name = L"розрахункиToolStripMenuItem";
this->розрахункиToolStripMenuItem->Size = System::Drawing::Size(78, 20);
this->розрахункиToolStripMenuItem->Text = L"Розрахунки";
//
// параметриКодуToolStripMenuItem
//
this->параметриКодуToolStripMenuItem->Name = L"параметриКодуToolStripMenuItem";
this->параметриКодуToolStripMenuItem->Size = System::Drawing::Size(180, 22);
this->параметриКодуToolStripMenuItem->Text = L"Параметри коду...";
this->параметриКодуToolStripMenuItem->Click += gcnew System::EventHandler(this,
&Form1::параметриКодуToolStripMenuItem_Click);
//
// розміриФайлівToolStripMenuItem
//
this->розміриФайлівToolStripMenuItem->Name = L"розміриФайлівToolStripMenuItem";
this->розміриФайлівToolStripMenuItem->Size = System::Drawing::Size(180, 22);
this->розміриФайлівToolStripMenuItem->Text = L"Розміри файлів...";
this->розміриФайлівToolStripMenuItem->Click += gcnew System::EventHandler(this,
&Form1::розміриФайлівToolStripMenuItem_Click);
//
// проПрограмуToolStripMenuItem
//
this->проПрограмуToolStripMenuItem->DropDownItems->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^
>(1) { this->проПрограмуToolStripMenuItem1 });
this->проПрограмуToolStripMenuItem->Name = L"проПрограмуToolStripMenuItem";
this->проПрограмуToolStripMenuItem->Size = System::Drawing::Size(60, 20);
this->проПрограмуToolStripMenuItem->Text = L"Довідка";
//
// проПрограмуToolStripMenuItem1
//
this->проПрограмуToolStripMenuItem1->Name = L"проПрограмуToolStripMenuItem1";
this->проПрограмуToolStripMenuItem1->Size = System::Drawing::Size(166, 22);
this->проПрограмуToolStripMenuItem1->Text = L"Про програму...";
//
// tabControl1
//
this->tabControl1->Controls->Add(this->tabPage1);
this->tabControl1->Controls->Add(this->tabPage2);
this->tabControl1->Controls->Add(this->tabPage3);
this->tabControl1->Controls->Add(this->tabPage4);
this->tabControl1->Location = System::Drawing::Point(12, 40);
this->tabControl1->Name = L"tabControl1";
this->tabControl1->SelectedIndex = 0;
this->tabControl1->Size = System::Drawing::Size(613, 323);
this->tabControl1->TabIndex = 1;
//

```

```

// tabPage1
//
this->tabPage1->BackColor = System::Drawing::SystemColors::Control;
this->tabPage1->Controls->Add(this->groupBox1);
this->tabPage1->Controls->Add(this->button2);
this->tabPage1->Controls->Add(this->button1);
this->tabPage1->Controls->Add(this->richTextBox1);
this->tabPage1->Location = System::Drawing::Point(4, 22);
this->tabPage1->Name = L"tabPage1";
this->tabPage1->Padding = System::Windows::Forms::Padding(3);
this->tabPage1->Size = System::Drawing::Size(605, 297);
this->tabPage1->TabIndex = 0;
this->tabPage1->Text = L"Відкриття файлу";
//
// groupBox1
//
this->groupBox1->Controls->Add(this->textBox2);
this->groupBox1->Controls->Add(this->textBox3);
this->groupBox1->Controls->Add(this->textBox1);
this->groupBox1->Controls->Add(this->label3);
this->groupBox1->Controls->Add(this->label2);
this->groupBox1->Controls->Add(this->label1);
this->groupBox1->Location = System::Drawing::Point(318, 19);
this->groupBox1->Name = L"groupBox1";
this->groupBox1->Size = System::Drawing::Size(265, 115);
this->groupBox1->TabIndex = 3;
this->groupBox1->TabStop = false;
this->groupBox1->Text = L"Файл";
//
// textBox2
//
this->textBox2->Location = System::Drawing::Point(110, 54);
this->textBox2->Name = L"textBox2";
this->textBox2->Size = System::Drawing::Size(135, 20);
this->textBox2->TabIndex = 4;
//
// textBox3
//
this->textBox3->Location = System::Drawing::Point(110, 86);
this->textBox3->Name = L"textBox3";
this->textBox3->Size = System::Drawing::Size(135, 20);
this->textBox3->TabIndex = 5;
//
// textBox1
//
this->textBox1->Location = System::Drawing::Point(110, 19);
this->textBox1->Name = L"textBox1";
this->textBox1->Size = System::Drawing::Size(135, 20);
this->textBox1->TabIndex = 3;
//
// label3
//

```

```

this->label3->AutoSize = true;
this->label3->Location = System::Drawing::Point(17, 86);
this->label3->Name = L"label3";
this->label3->Size = System::Drawing::Size(45, 13);
this->label3->TabIndex = 2;
this->label3->Text = L"Розмір:";
//
// label2
//
this->label2->AutoSize = true;
this->label2->Location = System::Drawing::Point(17, 54);
this->label2->Name = L"label2";
this->label2->Size = System::Drawing::Size(52, 13);
this->label2->TabIndex = 1;
this->label2->Text = L"Формат:";
//
// label1
//
this->label1->AutoSize = true;
this->label1->Location = System::Drawing::Point(17, 22);
this->label1->Name = L"label1";
this->label1->Size = System::Drawing::Size(42, 13);
this->label1->TabIndex = 0;
this->label1->Text = L"Назва:";
//
// button2
//
this->button2->Enabled = false;
this->button2->Location = System::Drawing::Point(180, 249);
this->button2->Name = L"button2";
this->button2->Size = System::Drawing::Size(120, 23);
this->button2->TabIndex = 2;
this->button2->Text = L"Зберегти як...";
this->button2->UseVisualStyleBackColor = true;
//
// button1
//
this->button1->Location = System::Drawing::Point(180, 220);
this->button1->Name = L"button1";
this->button1->Size = System::Drawing::Size(120, 23);
this->button1->TabIndex = 1;
this->button1->Text = L"Відкрити...";
this->button1->UseVisualStyleBackColor = true;
this->button1->Click += gnew System::EventHandler(this, &Form1::button1_Click);
//
// richTextBox1
//
this->richTextBox1->BackColor = System::Drawing::SystemColors::Window;
this->richTextBox1->Location = System::Drawing::Point(20, 19);
this->richTextBox1->Name = L"richTextBox1";
this->richTextBox1->ReadOnly = true;
this->richTextBox1->ScrollBars = System::Windows::Forms::RichTextBoxScrollBars::ForcedVertical;

```

```

this->richTextBox1->Size = System::Drawing::Size(146, 253);
this->richTextBox1->TabIndex = 0;
this->richTextBox1->Text = L"";
//
// tabPage2
//
this->tabPage2->BackColor = System::Drawing::SystemColors::Control;
this->tabPage2->Controls->Add(this->button4);
this->tabPage2->Controls->Add(this->button3);
this->tabPage2->Controls->Add(this->richTextBox2);
this->tabPage2->Location = System::Drawing::Point(4, 22);
this->tabPage2->Name = L"tabPage2";
this->tabPage2->Padding = System::Windows::Forms::Padding(3);
this->tabPage2->Size = System::Drawing::Size(605, 297);
this->tabPage2->TabIndex = 1;
this->tabPage2->Text = L"Кодування";
//
// button4
//
this->button4->Enabled = false;
this->button4->Location = System::Drawing::Point(469, 249);
this->button4->Name = L"button4";
this->button4->Size = System::Drawing::Size(120, 23);
this->button4->TabIndex = 2;
this->button4->Text = L"Зберегти як...";
this->button4->UseVisualStyleBackColor = true;
//
// button3
//
this->button3->Enabled = false;
this->button3->Location = System::Drawing::Point(469, 220);
this->button3->Name = L"button3";
this->button3->Size = System::Drawing::Size(120, 23);
this->button3->TabIndex = 1;
this->button3->Text = L"Закодувати";
this->button3->UseVisualStyleBackColor = true;
this->button3->Click += genew System::EventHandler(this, &Form1::button3_Click);
//
// richTextBox2
//
this->richTextBox2->BackColor = System::Drawing::SystemColors::Window;
this->richTextBox2->Location = System::Drawing::Point(20, 19);
this->richTextBox2->Name = L"richTextBox2";
this->richTextBox2->ReadOnly = true;
this->richTextBox2->ScrollBars = System::Windows::Forms::RichTextBoxScrollBars::ForcedVertical;
this->richTextBox2->Size = System::Drawing::Size(437, 253);
this->richTextBox2->TabIndex = 0;
this->richTextBox2->Text = L"";
//
// tabPage3
//
this->tabPage3->BackColor = System::Drawing::SystemColors::Control;

```

```

this->tabPage3->Controls->Add(this->button6);
this->tabPage3->Controls->Add(this->textBox5);
this->tabPage3->Controls->Add(this->button5);
this->tabPage3->Controls->Add(this->richTextBox3);
this->tabPage3->Location = System::Drawing::Point(4, 22);
this->tabPage3->Name = L"tabPage3";
this->tabPage3->Size = System::Drawing::Size(605, 297);
this->tabPage3->TabIndex = 2;
this->tabPage3->Text = L"Генерування помилок";
//
// button6
//
this->button6->Enabled = false;
this->button6->Location = System::Drawing::Point(469, 249);
this->button6->Name = L"button6";
this->button6->Size = System::Drawing::Size(120, 23);
this->button6->TabIndex = 6;
this->button6->Text = L"Зберегти як...";
this->button6->UseVisualStyleBackColor = true;
//
// textBox5
//
this->textBox5->Location = System::Drawing::Point(469, 194);
this->textBox5->Name = L"textBox5";
this->textBox5->Size = System::Drawing::Size(41, 20);
this->textBox5->TabIndex = 5;
//
// button5
//
this->button5->Enabled = false;
this->button5->Location = System::Drawing::Point(469, 220);
this->button5->Name = L"button5";
this->button5->Size = System::Drawing::Size(120, 23);
this->button5->TabIndex = 2;
this->button5->Text = L"Згенерувати";
this->button5->UseVisualStyleBackColor = true;
this->button5->Click += genew System::EventHandler(this, &Form1::button5_Click);
//
// richTextBox3
//
this->richTextBox3->BackColor = System::Drawing::SystemColors::Window;
this->richTextBox3->Location = System::Drawing::Point(20, 19);
this->richTextBox3->Name = L"richTextBox3";
this->richTextBox3->ReadOnly = true;
this->richTextBox3->ScrollBars = System::Windows::Forms::RichTextBoxScrollBars::ForcedVertical;
this->richTextBox3->Size = System::Drawing::Size(437, 253);
this->richTextBox3->TabIndex = 1;
this->richTextBox3->Text = L"";
//
// tabPage4
//
this->tabPage4->BackColor = System::Drawing::SystemColors::Control;

```

```

this->tabPage4->Controls->Add(this->button8);
this->tabPage4->Controls->Add(this->button7);
this->tabPage4->Controls->Add(this->richTextBox4);
this->tabPage4->Location = System::Drawing::Point(4, 22);
this->tabPage4->Name = L"tabPage4";
this->tabPage4->Size = System::Drawing::Size(605, 297);
this->tabPage4->TabIndex = 3;
this->tabPage4->Text = L"Декодування";
//
// button8
//
this->button8->Enabled = false;
this->button8->Location = System::Drawing::Point(180, 249);
this->button8->Name = L"button8";
this->button8->Size = System::Drawing::Size(120, 23);
this->button8->TabIndex = 2;
this->button8->Text = L"Зберегти як...";
this->button8->UseVisualStyleBackColor = true;
//
// button7
//
this->button7->Enabled = false;
this->button7->Location = System::Drawing::Point(180, 220);
this->button7->Name = L"button7";
this->button7->Size = System::Drawing::Size(120, 23);
this->button7->TabIndex = 1;
this->button7->Text = L"Декодувати";
this->button7->UseVisualStyleBackColor = true;
this->button7->Click += gcnew System::EventHandler(this, &Form1::button7_Click);
//
// richTextBox4
//
this->richTextBox4->BackColor = System::Drawing::SystemColors::Window;
this->richTextBox4->Location = System::Drawing::Point(20, 19);
this->richTextBox4->Name = L"richTextBox4";
this->richTextBox4->ReadOnly = true;
this->richTextBox4->ScrollBars = System::Windows::Forms::RichTextBoxScrollBars::ForcedVertical;
this->richTextBox4->Size = System::Drawing::Size(146, 253);
this->richTextBox4->TabIndex = 0;
this->richTextBox4->Text = L"";
//
// comboBox1
//
this->comboBox1->DropDownStyle = System::Windows::Forms::ComboBoxStyle::DropDownList;
this->comboBox1->FormattingEnabled = true;
this->comboBox1->Items->AddRange(gcnew cli::array< System::Object^ >(2) {L"Виберіть параметри ІКВ", L"ІКВ(н=8, R=4)"});
this->comboBox1->Location = System::Drawing::Point(641, 62);
this->comboBox1->Name = L"comboBox1";
this->comboBox1->Size = System::Drawing::Size(168, 21);
this->comboBox1->TabIndex = 2;
this->comboBox1->SelectedIndexChanged += gcnew System::EventHandler(this, &Form1::comboBox1_SelectedIndexChanged);
//

```

```

// comboBox2
//
this->comboBox2->DropDownStyle = System::Windows::Forms::ComboBoxStyle::DropDownList;
this->comboBox2->FormattingEnabled = true;
this->comboBox2->Items->AddRange(gcnew cli::array< System::Object^ >(2) {L"Виберіть ряд ІКВ", L"[1,1,1,2,2,1,3,4]"});
this->comboBox2->Location = System::Drawing::Point(641, 95);
this->comboBox2->Name = L"comboBox2";
this->comboBox2->Size = System::Drawing::Size(168, 21);
this->comboBox2->TabIndex = 3;
this->comboBox2->Visible = false;
this->comboBox2->SelectedIndexChanged += gcnew System::EventHandler(this, &Form1::comboBox2_SelectedIndexChanged);
//
// textBox4
//
this->textBox4->Location = System::Drawing::Point(641, 135);
this->textBox4->Multiline = true;
this->textBox4->Name = L"textBox4";
this->textBox4->ReadOnly = true;
this->textBox4->Size = System::Drawing::Size(168, 228);
this->textBox4->TabIndex = 4;
this->textBox4->Visible = false;
//
// openFileDialog1
//
this->openFileDialog1->FileName = L"openFileDialog1";
//
// progressBar1
//
this->progressBar1->Location = System::Drawing::Point(12, 373);
this->progressBar1->Name = L"progressBar1 ";
this->progressBar1->Size = System::Drawing::Size(797, 15);
this->progressBar1->Step = 1;
this->progressBar1->TabIndex = 5;
//
// Form1
//
this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
this->ClientSize = System::Drawing::Size(823, 400);
this->Controls->Add(this->progressBar1);
this->Controls->Add(this->textBox4);
this->Controls->Add(this->comboBox2);
this->Controls->Add(this->comboBox1);
this->Controls->Add(this->tabControl1);
this->Controls->Add(this->menuStrip1);
this->FormBorderStyle = System::Windows::Forms::FormBorderStyle::FixedSingle;
this->MainMenuStrip = this->menuStrip1;
this->MaximizeBox = false;
this->Name = L"Form1 ";
this->Text = L"ІКВ-кодвання";
this->Load += gcnew System::EventHandler(this, &Form1::Form1_Load);
this->menuStrip1->ResumeLayout(false);

```

```

this->menuStrip1->PerformLayout();
this->tabControl1->ResumeLayout(false);
this->tabPage1->ResumeLayout(false);
this->groupBox1->ResumeLayout(false);
this->groupBox1->PerformLayout();
this->tabPage2->ResumeLayout(false);
this->tabPage3->ResumeLayout(false);
this->tabPage3->PerformLayout();
this->tabPage4->ResumeLayout(false);
this->ResumeLayout(false);
this->PerformLayout();

}
#pragma endregion

void setCurrentSymbolColor(unsigned long ind, unsigned char* errPos, unsigned size){

    for(unsigned i=0; i<size; i++){
        richTextBox3->HideSelection=true;
        richTextBox3->SelectionStart=ind+14-errPos[i];
        richTextBox3->SelectionLength=1;

        richTextBox3->SelectionColor=System::Drawing::Color::White;
        richTextBox3->SelectionBackColor=System::Drawing::Color::Red;
    }
}

private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e) {
    comboBox1->SelectedIndex=0; comboBox1->SelectedIndex=0;
}

private: System::Void comboBox1_SelectedIndexChanged(System::Object^ sender, System::EventArgs^ e) {
    int ind=comboBox1->SelectedIndex;
    if(!ind) comboBox2->Visible=false;
    else{ comboBox2->SelectedIndex=0;
        comboBox2->Visible=true;
    }
}

private: System::Void comboBox2_SelectedIndexChanged(System::Object^ sender, System::EventArgs^ e) {
    int ind=comboBox2->SelectedIndex;
    if(!ind){ textBox4->Visible=false;
        textBox4->Clear();
    }
    else{ this->textBox4->Visible=true;

        int serie[]={1,1,1,2,2,1,3,4};
        TAllowedCombination *allowedComb=new TAllowedCombination(
            new TIdealRingBundle(8,4,serie) );
    }
}

```

```

char *strBinary1=new char[5], *strBinary2=new char[16], *strOutput=new char[30];
for(unsigned short i=0; i<allowedComb->size; i++){
    _itoa_s(i,strBinary1,5,2);
    _itoa_s(allowedComb->code_comb[i],strBinary2,16,2);
    sprintf_s(strOutput,30,"%04s ==> %015s",strBinary1,strBinary2);
    textBox4->AppendText(gcnew String(strOutput));
    textBox4->AppendText(Environment::NewLine);
}
}
}

private: System::Void відкритиToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) {
    this->openFileDialog1->FileName="Без назви";
    this->openFileDialog1->DefaultExt="jpg";
    this->openFileDialog1->Filter="JPEG files (*.jpg)|*.jpg|
                                        \"PNG files (*.png)|*.png|
                                        \"BMP files (*.bmp)|*.bmp|
                                        \"GIF files (*.gif)|*.gif|
                                        \"All files (*.*)|*.*;

    if(this->openFileDialog1->ShowDialog()==System::Windows::Forms::DialogResult::OK){
        filename=(char*)(void*)Marshal::StringToHGlobalAnsi
            (this->openFileDialog1->FileName);

        richTextBox1->Clear();
        FILE *fp=fopen(filename, "rb"); unsigned char chTmp;
        char *strBinary=new char[9], *strOutput=new char[9];
        fsize=getFileSize(fp); int sVisible=fsize<1000?fsize:1000;
        int min=progressBar1->Minimum, max=progressBar1->Maximum;
        progressBar1->Value=0;

        for(int i=0; i<sVisible; i++){
            fread(&chTmp, sizeof(unsigned char), 1, fp);
            _itoa_s(chTmp,strBinary,9,2); sprintf_s(strOutput,9,"%08s", strBinary);
            richTextBox1->AppendText(gcnew String(strOutput));
            if(i&1==0) this->richTextBox1->AppendText(" ");
            else richTextBox1->AppendText(Environment::NewLine);
            progressBar1->Value=(max-min)*(i+1)/sVisible;
        }
        fclose(fp);

        char *locFilename=new char[400], *pCh1, *pCh2; strcpy(locFilename, filename);
        pCh1=strrchr(locFilename, '\\')+1; pCh2=strrchr(pCh1, '.'); *pCh2=0;
        textBox1->Text=gcnew String(pCh1); textBox2->Text=gcnew String(pCh2+1);
        textBox3->Text=fsize.ToString(); button2->Enabled=true; button3->Enabled=true;
    }
}

private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {

```

```

        відкритиToolStripMenuItem->PerformClick();
    }

private: System::Void button3_Click(System::Object^ sender, System::EventArgs^ e) {
    int serie[]={1,1,1,2,2,1,3,4};
    TCodingDecoding *obj=new TCodingDecoding(
        new TAllowedCombination(new TIdealRingBundle(8,4,serie)) );
    richTextBox2->Clear();
    FILE *fp1, *fp2; fp1=fopen(filename,"rb"); fp2=fopen("encoded.bin","wb");
    unsigned char chTmp; unsigned short *shTmp=new unsigned short[2];
    int min=progressBar1->Minimum, max=progressBar1->Maximum;
    progressBar1->Value=0;

    for(unsigned long i=0; i<fsize; i++){
        fread(&chTmp, sizeof(unsigned char), 1, fp1);
        obj->doCoding(chTmp, shTmp);
        fwrite(shTmp, sizeof(unsigned short), 2, fp2);
        progressBar1->Value=(max-min)*(i+1)/(2*fsize);
    }
    fclose(fp1); fclose(fp2);

    unsigned sVisible=fsize<2000?fsize:2000;
    char *strBinary=new char[16], *strOutput=new char[16];
    fp1=fopen("encoded.bin","rb");

    for(unsigned long i=0; i<sVisible; i++){
        fread(shTmp, sizeof(unsigned short), 1, fp1);
        _itoa_s(*shTmp,strBinary,16,2); sprintf_s(strOutput,16,"%015s", strBinary);
        richTextBox2->AppendText(gcnew String(strOutput));
        if((i&3)!=3) richTextBox2->AppendText(" ");
        else richTextBox2->AppendText(Environment::NewLine);
        progressBar1->Value=(max-min)*(i+1)/(2*sVisible)+(max-min)/2;
    }
    fclose(fp1); this->button4->Enabled=true; this->button5->Enabled=true;
}

private: System::Void button5_Click(System::Object^ sender, System::EventArgs^ e) {
    int len=Convert::ToInt32(textBox5->Text);
    TGenerationError *err=new TGenerationError(len);

    FILE *fp1=fopen("encoded.bin","rb"), *fp2=fopen("encodedErr.bin","wb"),
        *fp3=fopen("temp.bin","wb"); richTextBox3->Clear();

    unsigned char *chTmp=new unsigned char[len];
    int min=progressBar1->Minimum, max=progressBar1->Maximum;
    progressBar1->Value=0; unsigned short shTmp;

    for(unsigned long i=0; i<2*fsize; i++){
        fread(&shTmp, sizeof(unsigned short), 1, fp1);
        shTmp^=err->getError(); err->getErrorArray(chTmp);
    }
}

```

```

        fwrite(&shTmp, sizeof(unsigned short), 1, fp2);
        fwrite(chTmp, sizeof(unsigned char), len, fp3);
        progressBar1->Value=(max-min)*(i+1)/(4*fsize);
    }
fclose(fp1); fclose(fp2); fclose(fp3);

unsigned sVisible=fsize<2000?fsize:2000;
char *strBinary=new char[16], *strOutput=new char[16];
fp1=fopen("encodedErr.bin","rb"); fp2=fopen("temp.bin","rb");

for(unsigned long i=0, ind=-16; i<sVisible; i++){
    fread(&shTmp, sizeof(unsigned short),1,fp1);
    fread(chTmp, sizeof(unsigned char),len,fp2);
    _itoa_s(shTmp,strBinary,16,2); sprintf_s(strOutput,16,"%015s", strBinary);
    richTextBox3->AppendText(gcnew String(strOutput));

    if((i&3)!=3) this->richTextBox3->AppendText(" ");
    else this->richTextBox3->AppendText(Environment::NewLine);
    if(!(i%4)) {ind+=16; setCurrentSymbolColor(ind,chTmp,len);}
    else {ind+=19; setCurrentSymbolColor(ind,chTmp,len);}
    progressBar1->Value=(max-min)*(i+1)/(2*sVisible)+(max-min)/2;
}
fclose(fp1); fclose(fp2); this->button6->Enabled=true; this->button7->Enabled=true;
}

private: System::Void button7_Click(System::Object^ sender, System::EventArgs^ e) {
    int serie[]={1,1,1,2,2,1,3,4};
    TCodingDecoding *obj=new TCodingDecoding(
        new TAllowedCombination(new TIdealRingBundle(8,4,serie)) );
    richTextBox4->Clear();

    char *filenameNew=new char[400], *pCh1, *pCh2; strcpy(filenameNew, filename);
    pCh1=strchr(filenameNew,'\')+1; *pCh1=0; pCh2=strchr(pCh1+1,');
    strcat(filenameNew,"DECODED"); strcat(filenameNew,pCh2);
    FILE *fp1=fopen("encodedErr.bin","rb"), *fp2=fopen(filenameNew,"wb");
    unsigned char chTmp; unsigned short *shTmp=new unsigned short[2];
    int min=progressBar1->Minimum, max=progressBar1->Maximum;
    progressBar1->Value=0;

    for(unsigned long i=0; i<fsize; i++){
        fread(shTmp, sizeof(unsigned short), 2, fp1);
        chTmp=obj->doDecoding(shTmp);
        fwrite(&chTmp, sizeof(unsigned char), 1, fp2);
        progressBar1->Value=(max-min)*(i+1)/(2*fsize);
    }
    fclose(fp1); fclose(fp2);

    fp1=fopen(filenameNew,"rb"); unsigned sVisible=fsize<2000?fsize:2000;
    char *strBinary=new char[9], *strOutput=new char[9];

```

```

        for(unsigned long i=0; i<sVisible; i++){
            fread(&chTmp, sizeof(unsigned char), 1, fp1);
            _itoa_s(chTmp, strBinary, 9, 2); sprintf_s(strOutput, 9, "%08s", strBinary);
            richTextBox4->AppendText(gcnew String(strOutput));
            if((i&1)==0) this->richTextBox4->AppendText("  ");
            else richTextBox4->AppendText(Environment::NewLine);
            progressBar1->Value=(max-min)*(i+1)/(2*sVisible)+(max-min)/2;
        }
        fclose(fp1); this->button8->Enabled=true;
    }

private: System::Void закритиToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) {
    Form1::Close();
}

private: System::Void очиститиВсіПоляToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) {
    richTextBox1->Clear(); richTextBox2->Clear(); richTextBox3->Clear(); richTextBox4->Clear();
    button2->Enabled=false; button3->Enabled=false; button4->Enabled=false; button5->Enabled=false;
    button6->Enabled=false; button7->Enabled=false; button8->Enabled=false;
    textBox1->Clear(); textBox2->Clear(); textBox3->Clear(); textBox5->Clear();
}

private: System::Void параметриКодуToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) {
    Form2 ^form2 = gcnew Form2();
    form2->ShowDialog();
}

private: System::Void розміриФайлівToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) {
    Form3 ^form3 = gcnew Form3();
    form3->ShowDialog();
}
};
}

```

## **/\*Form2.h\*/**

```

#pragma once
#include <iostream>

namespace bundleCodingDecoding {

using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;

/// <summary>
/// Summary for Form2
/// </summary>

```

```

public ref class Form2 : public System::Windows::Forms::Form
{
public:
Form2(void)
{
    InitializeComponent();
    //
    //TODO: Add the constructor code here
    //
}

protected:
/// <summary>
/// Clean up any resources being used.
/// </summary>
~Form2()
{
    if (components)
    {
        delete components;
    }
}
private: System::Windows::Forms::GroupBox^ groupBox1;
private: System::Windows::Forms::Label^ label6;
private: System::Windows::Forms::Label^ label5;
private: System::Windows::Forms::Label^ label4;
private: System::Windows::Forms::Label^ label3;
private: System::Windows::Forms::Label^ label2;
private: System::Windows::Forms::Label^ label1;
private: System::Windows::Forms::ComboBox^ comboBox1;
private: System::Windows::Forms::TextBox^ textBox6;
private: System::Windows::Forms::TextBox^ textBox5;
private: System::Windows::Forms::TextBox^ textBox4;
private: System::Windows::Forms::TextBox^ textBox3;
private: System::Windows::Forms::TextBox^ textBox2;
private: System::Windows::Forms::TextBox^ textBox1;
private: System::Windows::Forms::Button^ button1;
protected:

private:
/// <summary>
/// Required designer variable.
/// </summary>
System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
void InitializeComponent(void)
{
    this->groupBox1 = (gcnew System::Windows::Forms::GroupBox());
    this->textBox6 = (gcnew System::Windows::Forms::TextBox());
    this->textBox5 = (gcnew System::Windows::Forms::TextBox());
    this->textBox4 = (gcnew System::Windows::Forms::TextBox());
    this->textBox3 = (gcnew System::Windows::Forms::TextBox());
    this->textBox2 = (gcnew System::Windows::Forms::TextBox());
    this->textBox1 = (gcnew System::Windows::Forms::TextBox());
    this->comboBox1 = (gcnew System::Windows::Forms::ComboBox());
    this->label6 = (gcnew System::Windows::Forms::Label());
    this->label5 = (gcnew System::Windows::Forms::Label());
    this->label4 = (gcnew System::Windows::Forms::Label());
    this->label3 = (gcnew System::Windows::Forms::Label());
    this->label2 = (gcnew System::Windows::Forms::Label());
    this->label1 = (gcnew System::Windows::Forms::Label());
    this->button1 = (gcnew System::Windows::Forms::Button());
    this->groupBox1->SuspendLayout();
    this->SuspendLayout();
    //
    // groupBox1
    //
    this->groupBox1->Controls->Add(this->textBox6);
    this->groupBox1->Controls->Add(this->textBox5);
    this->groupBox1->Controls->Add(this->textBox4);
}

```

```

this->groupBox1->Controls->Add(this->textBox3);
this->groupBox1->Controls->Add(this->textBox2);
this->groupBox1->Controls->Add(this->textBox1);
this->groupBox1->Controls->Add(this->comboBox1);
this->groupBox1->Controls->Add(this->label6);
this->groupBox1->Controls->Add(this->label5);
this->groupBox1->Controls->Add(this->label4);
this->groupBox1->Controls->Add(this->label3);
this->groupBox1->Controls->Add(this->label2);
this->groupBox1->Controls->Add(this->label1);
this->groupBox1->Location = System::Drawing::Point(12, 12);
this->groupBox1->Name = L"groupBox1";
this->groupBox1->Size = System::Drawing::Size(377, 258);
this->groupBox1->TabIndex = 9;
this->groupBox1->TabStop = false;
this->groupBox1->Text = L"Розрахунок параметрів завадостійкого ІКБ";
//
// textBox6
//
this->textBox6->Location = System::Drawing::Point(275, 228);
this->textBox6->Name = L"textBox6";
this->textBox6->Size = System::Drawing::Size(75, 20);
this->textBox6->TabIndex = 7;
//
// textBox5
//
this->textBox5->Location = System::Drawing::Point(275, 196);
this->textBox5->Name = L"textBox5";
this->textBox5->Size = System::Drawing::Size(75, 20);
this->textBox5->TabIndex = 6;
//
// textBox4
//
this->textBox4->Location = System::Drawing::Point(275, 164);
this->textBox4->Name = L"textBox4";
this->textBox4->Size = System::Drawing::Size(75, 20);
this->textBox4->TabIndex = 5;
//
// textBox3
//
this->textBox3->Location = System::Drawing::Point(275, 132);
this->textBox3->Name = L"textBox3";
this->textBox3->Size = System::Drawing::Size(75, 20);
this->textBox3->TabIndex = 4;
//
// textBox2
//
this->textBox2->Location = System::Drawing::Point(275, 100);
this->textBox2->Name = L"textBox2";
this->textBox2->Size = System::Drawing::Size(75, 20);
this->textBox2->TabIndex = 3;
//
// textBox1
//
this->textBox1->Location = System::Drawing::Point(275, 68);
this->textBox1->Name = L"textBox1";
this->textBox1->Size = System::Drawing::Size(75, 20);
this->textBox1->TabIndex = 2;
//
// comboBox1
//
this->comboBox1->DisplayMember = L"0";
this->comboBox1->DropDownStyle = System::Windows::Forms::ComboBoxStyle::DropDownList;
this->comboBox1->FormattingEnabled = true;
this->comboBox1->Items->AddRange(gcnew cli::array< System::Object^ >(8) {L"Виберіть
параметри ІКБ", L"ІКБ(n=3, R=1)", L"ІКБ(n=4, R=1)",
L"ІКБ(n=4, R=2)", L"ІКБ(n=6, R=1)", L"ІКБ(n=8, R=4)", L"ІКБ(n=15, R=7)",
L"ІКБ(n=32, R=16)"});
this->comboBox1->Location = System::Drawing::Point(24, 32);
this->comboBox1->Name = L"comboBox1";
this->comboBox1->Size = System::Drawing::Size(326, 21);
this->comboBox1->TabIndex = 0;
this->comboBox1->ValueMember = L"1";
this->comboBox1->SelectedIndexChanged += gcnew System::EventHandler(this,
&Form2::comboBox1_SelectedIndexChanged);

```

```

//
// label6
//
this->label6->AutoSize = true;
this->label6->Location = System::Drawing::Point(21, 231);
this->label6->Name = L"label6";
this->label6->Size = System::Drawing::Size(125, 13);
this->label6->TabIndex = 14;
this->label6->Text = L"Ступінь надлишковості:";
//
// label5
//
this->label5->AutoSize = true;
this->label5->Location = System::Drawing::Point(21, 199);
this->label5->Name = L"label5";
this->label5->Size = System::Drawing::Size(163, 13);
this->label5->TabIndex = 13;
this->label5->Text = L"Довжина закодованого блоку:";
//
// label4
//
this->label4->AutoSize = true;
this->label4->Location = System::Drawing::Point(21, 167);
this->label4->Name = L"label4";
this->label4->Size = System::Drawing::Size(134, 13);
this->label4->TabIndex = 12;
this->label4->Text = L"Довжина вхідного блоку:";
//
// label3
//
this->label3->AutoSize = true;
this->label3->Location = System::Drawing::Point(21, 135);
this->label3->Name = L"label3";
this->label3->Size = System::Drawing::Size(206, 13);
this->label3->TabIndex = 11;
this->label3->Text = L"Кількість помилок, що виправляються:";
//
// label2
//
this->label2->AutoSize = true;
this->label2->Location = System::Drawing::Point(21, 103);
this->label2->Name = L"label2";
this->label2->Size = System::Drawing::Size(194, 13);
this->label2->TabIndex = 10;
this->label2->Text = L"Кількість помилок, що виявляються:";
//
// label1
//
this->label1->AutoSize = true;
this->label1->Location = System::Drawing::Point(21, 71);
this->label1->Name = L"label1";
this->label1->Size = System::Drawing::Size(152, 13);
this->label1->TabIndex = 9;
this->label1->Text = L"Мінімальна кодова відстань:";
//
// button1
//
this->button1->Location = System::Drawing::Point(260, 285);
this->button1->Name = L"button1";
this->button1->Size = System::Drawing::Size(129, 23);
this->button1->TabIndex = 8;
this->button1->Text = L"Закрити вікно";
this->button1->UseVisualStyleBackColor = true;
this->button1->Click += gcnew System::EventHandler(this, &Form2::button1_Click);
//
// Form2
//
this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
this->ClientSize = System::Drawing::Size(403, 322);
this->Controls->Add(this->button1);
this->Controls->Add(this->groupBox1);
this->FormBorderStyle = System::Windows::Forms::FormBorderStyle::FixedSingle;
this->MaximizeBox = false;
this->MinimizeBox = false;

```

```

        this->Name = L"Form2";
        this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;
        this->Load += gcnew System::EventHandler(this, &Form2::Form2_Load);
        this->groupBox1->ResumeLayout(false);
        this->groupBox1->PerformLayout();
        this->ResumeLayout(false);
    }
#pragma endregion

private: System::Void Form2_Load(System::Object^ sender, System::EventArgs^ e) {
    comboBox1->SelectedIndex=0;
}

private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
    Form2::Close();
}

private: System::Void comboBox1_SelectedIndexChanged(System::Object^ sender, System::EventArgs^
e) {
    int arr_param[][2]={{-1,-1}, {3,1}, {4,1}, {4,2},
        {6,1}, {8,4}, {15,7}, {32,16}};
    int ind=comboBox1->SelectedIndex,
        rank, multy, ringSum, dmin;
    if(ind){
        rank=arr_param[ind][0]; multy=arr_param[ind][1];
        dmin=2*(rank-multy); ringSum=rank*(rank-1)/multy+1;
        textBox1->Text=dmin.ToString();
        textBox2->Text=(dmin-1).ToString();
        textBox3->Text=((dmin-1)/2).ToString();
        textBox4->Text=((int)(log((double)ringSum+1)/log(2.0))).ToString();
        textBox5->Text=ringSum.ToString();
        textBox6->Text=(ringSum-(int)(log((double)ringSum+1)/log(2.0))).ToString();
    }
    else{
        textBox1->Clear(); textBox2->Clear(); textBox3->Clear();
        textBox4->Clear(); textBox5->Clear(); textBox6->Clear();
    }
}
};
}

```

### **/\*Form3.h\*/**

```

#pragma once
#include <iostream>

namespace bundleCodingDecoding {

using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;
using namespace System::Runtime::InteropServices;

/// <summary>
/// Summary for Form3
/// </summary>
public ref class Form3 : public System::Windows::Forms::Form
{
public:
    Form3(void)
    {
        InitializeComponent();
        //
        //TODO: Add the constructor code here
        //
    }

protected:

```

```

/// <summary>
/// Clean up any resources being used.
/// </summary>
~Form3()
{
    if (components)
    {
        delete components;
    }
}
private: System::Windows::Forms::GroupBox^ groupBox1;

private: System::Windows::Forms::Label^ label1;
private: System::Windows::Forms::TextBox^ textBox1;
private: System::Windows::Forms::Label^ label2;
private: System::Windows::Forms::TextBox^ textBox2;
private: System::Windows::Forms::Button^ button1;
private: System::Windows::Forms::Button^ button2;
private: System::Windows::Forms::OpenFileDialog^ openFileDialog1;
private: System::Windows::Forms::ComboBox^ comboBox1;
private: System::Windows::Forms::Label^ label3;
private: System::Windows::Forms::TextBox^ textBox3;

protected:

private:
/// <summary>
/// Required designer variable.
/// </summary>
System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
void InitializeComponent(void)
{
    this->groupBox1 = (gcnew System::Windows::Forms::GroupBox());
    this->textBox3 = (gcnew System::Windows::Forms::TextBox());
    this->label3 = (gcnew System::Windows::Forms::Label());
    this->comboBox1 = (gcnew System::Windows::Forms::ComboBox());
    this->button1 = (gcnew System::Windows::Forms::Button());
    this->textBox2 = (gcnew System::Windows::Forms::TextBox());
    this->label2 = (gcnew System::Windows::Forms::Label());
    this->textBox1 = (gcnew System::Windows::Forms::TextBox());
    this->label1 = (gcnew System::Windows::Forms::Label());
    this->button2 = (gcnew System::Windows::Forms::Button());
    this->openFileDialog1 = (gcnew System::Windows::Forms::OpenFileDialog());
    this->groupBox1->SuspendLayout();
    this->SuspendLayout();
    //
    // groupBox1
    //
    this->groupBox1->Controls->Add(this->textBox3);
    this->groupBox1->Controls->Add(this->label3);
    this->groupBox1->Controls->Add(this->comboBox1);
    this->groupBox1->Controls->Add(this->button1);
    this->groupBox1->Controls->Add(this->textBox2);
    this->groupBox1->Controls->Add(this->label2);
    this->groupBox1->Controls->Add(this->textBox1);
    this->groupBox1->Controls->Add(this->label1);
    this->groupBox1->Location = System::Drawing::Point(12, 12);
    this->groupBox1->Name = L"groupBox1";
    this->groupBox1->Size = System::Drawing::Size(377, 166);
    this->groupBox1->TabIndex = 0;
    this->groupBox1->TabStop = false;
    this->groupBox1->Text = L"Розрахунок розміру файлів";
    //
    // textBox3
    //
    this->textBox3->BackColor = System::Drawing::SystemColors::Window;
    this->textBox3->Location = System::Drawing::Point(216, 128);
    this->textBox3->Name = L"textBox3";
}

```

```

this->textBox3->ReadOnly = true;
this->textBox3->Size = System::Drawing::Size(135, 20);
this->textBox3->TabIndex = 9;
//
// label3
//
this->label3->AutoSize = true;
this->label3->Location = System::Drawing::Point(21, 135);
this->label3->Name = L"label3";
this->label3->Size = System::Drawing::Size(130, 13);
this->label3->TabIndex = 8;
this->label3->Text = L"Розмір після кодування:";
//
// comboBox1
//
this->comboBox1->DropDownStyle = System::Windows::Forms::ComboBoxStyle::DropDownList;
this->comboBox1->FormattingEnabled = true;
this->comboBox1->Items->AddRange(gcnew cli::array< System::Object^ >(8) {L"Виберіть
параметри ІКВ", L"ІКВ(n=3, R=1)", L"ІКВ(n=4, R=1)",
L"ІКВ(n=4, R=2)", L"ІКВ(n=6, R=1)", L"ІКВ(n=8, R=4)", L"ІКВ(n=15, R=7)",
L"ІКВ(n=32, R=16)"});
this->comboBox1->Location = System::Drawing::Point(24, 96);
this->comboBox1->Name = L"comboBox1";
this->comboBox1->Size = System::Drawing::Size(327, 21);
this->comboBox1->TabIndex = 7;
this->comboBox1->SelectedIndexChanged += gcnew System::EventHandler(this,
&Form3::comboBox1_SelectedIndexChanged);
//
// button1
//
this->button1->Location = System::Drawing::Point(176, 23);
this->button1->Name = L"button1";
this->button1->Size = System::Drawing::Size(34, 23);
this->button1->TabIndex = 3;
this->button1->Text = L"...";
this->button1->UseVisualStyleBackColor = true;
this->button1->Click += gcnew System::EventHandler(this, &Form3::button1_Click);
//
// textBox2
//
this->textBox2->BackColor = System::Drawing::SystemColors::Window;
this->textBox2->Location = System::Drawing::Point(216, 57);
this->textBox2->Name = L"textBox2";
this->textBox2->ReadOnly = true;
this->textBox2->Size = System::Drawing::Size(135, 20);
this->textBox2->TabIndex = 5;
//
// label2
//
this->label2->AutoSize = true;
this->label2->Location = System::Drawing::Point(21, 64);
this->label2->Name = L"label2";
this->label2->Size = System::Drawing::Size(116, 13);
this->label2->TabIndex = 4;
this->label2->Text = L"Розмір до кодування:";
//
// textBox1
//
this->textBox1->BackColor = System::Drawing::SystemColors::Window;
this->textBox1->Location = System::Drawing::Point(216, 25);
this->textBox1->Name = L"textBox1";
this->textBox1->ReadOnly = true;
this->textBox1->Size = System::Drawing::Size(135, 20);
this->textBox1->TabIndex = 6;
//
// label1
//
this->label1->AutoSize = true;
this->label1->Location = System::Drawing::Point(21, 32);
this->label1->Name = L"label1";
this->label1->Size = System::Drawing::Size(75, 13);
this->label1->TabIndex = 2;
this->label1->Text = L"Повна назва:";
//
// button2

```

```

//
this->button2->Location = System::Drawing::Point(260, 194);
this->button2->Name = L"button2";
this->button2->Size = System::Drawing::Size(129, 23);
this->button2->TabIndex = 1;
this->button2->Text = L"Закрити вікно";
this->button2->UseVisualStyleBackColor = true;
this->button2->Click += gcnew System::EventHandler(this, &Form3::button2_Click);
//
// openFileDialog1
//
this->openFileDialog1->FileName = L"openFileDialog1";
//
// Form3
//
this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
this->ClientSize = System::Drawing::Size(403, 232);
this->Controls->Add(this->button2);
this->Controls->Add(this->groupBox1);
this->FormBorderStyle = System::Windows::Forms::FormBorderStyle::FixedSingle;
this->MaximizeBox = false;
this->MinimizeBox = false;
this->Name = L"Form3";
this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;
this->Load += gcnew System::EventHandler(this, &Form3::Form3_Load);
this->groupBox1->ResumeLayout(false);
this->groupBox1->PerformLayout();
this->ResumeLayout(false);

}
#pragma endregion

char *filenameLoc; unsigned long fsize, fsizeEnc;
private: System::Void Form3_Load(System::Object^ sender, System::EventArgs^ e) {
    this->comboBox1->SelectedIndex=0;
    this->comboBox1->Enabled=false;
}

private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e) {
    Form3::Close();
}

private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
    this->openFileDialog1->FileName="Без назви";
    this->openFileDialog1->Filter="JPEG files (*.jpg)|*.jpg|"
        "PNG files
    (*.png)|*.png|"
        "BMP files
    (*.bmp)|*.bmp|"
        "GIF files
    (*.gif)|*.gif|"
        "All files (*.*)|*.*";

    if(this->openFileDialog1->ShowDialog()==System::Windows::Forms::DialogResult::OK){
        filenameLoc=(char*)(void*)Marshal::StringToHGlobalAnsi
            (this->openFileDialog1->FileName);
        char *chTmp=strrchr(filenameLoc,'\\')+1;
        FILE *fp=fopen(filenameLoc, "rb");fseek(fp, 0, SEEK_END);
        fsize = ftell(fp); fseek(fp, 0, SEEK_SET);
        this->comboBox1->Enabled=true;
        this->comboBox1->SelectedIndex=0;
        this->textBox3->Clear();
        this->textBox1->Text=gcnew String(chTmp);
        this->textBox2->Text=fsize.ToString();
    }
}

private: System::Void comboBox1_SelectedIndexChanged(System::Object^ sender, System::EventArgs^
e) {
    int arr_param[][2]={{-1,-1}, {3,1}, {4,1}, {4,2},

```

```

        {6,1}, {8,4}, {15,7}, {32,16}};
int ind=comboBox1->SelectedIndex, ringSum;
if(ind){
    ringSum=arr_param[ind][0]*(arr_param[ind][0]-1)/arr_param[ind][1]+1;

fsizeEnc=int(ceil(ringSum/8.0)*ceil(8*fsize/floor(log((double)ringSum+1)/log(2.0))));
    textBox3->Text=fsizeEnc.ToString();
}
else this->textBox3->Clear();
}
};
}

```

## /\*TIdealRingBundle.h\*/

```

#pragma once

class TIdealRingBundle
{
int rank, multy, ring_sum;
unsigned **bundle, *serie;

public:
TIdealRingBundle(void){}
TIdealRingBundle(int,int,int[]);
TIdealRingBundle(TIdealRingBundle&){}
~TIdealRingBundle(void) {delete[] bundle; delete[] serie;}

void print(void);
bool isReally(void);

void getParametr(int*);
void getBaseBundle(int*);
};

```

## /\*TIdealRingBundle.cpp\*/

```

#include "StdAfx.h"
#include <iostream>
#include "TIdealRingBundle.h"
using namespace std;

TIdealRingBundle::TIdealRingBundle(int _rank,int _multy,int _serie[]){
rank=_rank; multy=_multy; serie=new unsigned [rank];
ring_sum=rank*(rank-1)/multy+1;
for(int i=0; i<rank; i++) *(serie+i)=*(_serie+i);

bundle=new unsigned* [rank];
for(int i=0; i<rank; i++){
*(bundle+i)=new unsigned [rank];
for (int j=0; j<rank; j++)
    if (i!=j) *(bundle[i]+j)=-1;
    else *(bundle[i]+j)=*(serie+j);
}

for(int i=0; i<rank; i++)
for(int j=0; j<rank; j++){
    if (i<j) *(bundle[i]+j)=*(bundle[i]+j-1)+*(bundle[j]+j);
    if (i==j+1) *(bundle[i]+j)=*(bundle[0]+rank-1);
    if (i>j+1) *(bundle[i]+j)=*(bundle[i-1]+j)-*(bundle[i-1]+i-1);
}
}

void TIdealRingBundle::print(void){
int tab=4, indent=2;
for(int i=0,k=0; i<rank; i++,k=0){
for(int j=0; j<rank; j++)
    printf("%*d", tab, *(bundle[i]+j) );
}
}

```

```

while(k<indent) {cout<<endl; k++;}
}
}

void TIdealRingBundle::getParametr(int *temp){
*temp=rank; *(temp+1)=multy;
}

void TIdealRingBundle::getBaseBundle(int *temp){
for(int i=0; i<rank; i++){
*(temp+i)=(serie+i);
}
}

bool TIdealRingBundle::isReally(void){
int* temp_arr=new int[ring_sum];
for (int i=0; i<ring_sum; i++) *(temp_arr+i)=0;

for(int i=0, _i; i<rank; i++){
for(int j=0; j<rank; j++){
_i=(bundle[i]+j); temp_arr[_i-1]++;
}
}

for(int i=0; i<ring_sum-1; i++){
if (*(temp_arr+i)!=multy) return false;
if (*(temp_arr+ring_sum-1)!=rank) return false;
}

return true;
}

```

## **/\*TAllowedCombination.h\*/**

```

#pragma once
#include "TIdealRingBundle.h"

class TAllowedCombination
{
int rank, multy, ring_sum;
unsigned *serie;

public:
unsigned short *code_comb;
int size;

TAllowedCombination(void) {}
TAllowedCombination(TIdealRingBundle*);
TAllowedCombination(TAllowedCombination&) {}
~TAllowedCombination(void) {delete[] serie; delete[] code_comb;}

void print(void);

private:
void getAllowedCombination(void);
int getSumFirstElems(int);
};

```

## **/\*TAllowedCombination.cpp\*/**

```

#include "StdAfx.h"
#include <iostream>
#include "TAllowedCombination.h"
#include "TIdealRingBundle.h"
using namespace std;

TAllowedCombination::TAllowedCombination(TIdealRingBundle *_bd1){
int *param=new int[2]; _bd1->getParametr(param);

```

```

int *bundle=new int[*param]; _bd1->getBaseBundle(bundle);

rank=*param; multy=(param+1); ring_sum=rank*(rank-1)/multy+1;
serie=new unsigned [rank];
for(int i=0; i<rank; i++) *(serie+i)=(bundle+i);
size=(int)pow(2.0,log((double)ring_sum+1)/log(2.0));
code_comb=new unsigned short[size];

getAllowedCombination();
}

void TAllowedCombination::print(void){
char *str_bin=new char[ring_sum+1];
for(int i=0; i<size; i++){
_itoa_s(i,str_bin,ring_sum+1,2); printf("%04s: ", str_bin);
_itoa_s(*(code_comb+i),str_bin,ring_sum+1,2);
printf("%015s => %d\n", str_bin, *(code_comb+i));
}
}

//private methods
void TAllowedCombination::getAllowedCombination(void){
unsigned **true_pos=new unsigned*[ring_sum];
for(int i=0; i<ring_sum; i++){
*(true_pos+i)=new unsigned [rank];
for (int j=0; j<rank; j++)
*(true_pos[i]+j)=(i+getSumFirstElems(j+1))%ring_sum+1;
}

for(int i=0; i<size; i++) *(code_comb+i)=0x00;
for(int i=1; i<size; i++){
for(int j=0,temp; j<rank; j++){
temp=(true_pos[i-1]+j)-1;
*(code_comb+i)|=1<<temp;
}
}
}

int TAllowedCombination::getSumFirstElems(int k){
int i,sum; i=sum=0;
while(i<k) {sum+=serie[i]; i++;}
return sum;
}

```

## **/\*TCodingDecoding.h\*/**

```

#pragma once
#include "TAllowedCombination.h"
#include "TCodingDecoding.h"

class TCodingDecoding
{
TAllowedCombination *allowedComb;

public:
TCodingDecoding(){}
TCodingDecoding(TAllowedCombination*);
TCodingDecoding(TCodingDecoding&){}
~TCodingDecoding() {delete[] allowedComb;}

void doCoding(unsigned char, unsigned short*);
unsigned char doDecoding(unsigned short*);

private:
unsigned short getNumbTrueBits(unsigned short);
int getMinElemIndex(unsigned short*,int);
};

```

**/\*TCodingDecoding.cpp\*/**

```

#include "StdAfx.h"
#include <iostream>
#include "TCodingDecoding.h"
#include "TAllowedCombination.h"

TCodingDecoding::TCodingDecoding(TAllowedCombination *_comb16){
allowedComb=_comb16;
}

void TCodingDecoding::doCoding(unsigned char contSource, unsigned short *contEncoded){
int ind=(contSource&0xf0)>>4; *contEncoded=allowedComb->code_comb[ind];
ind=contSource&0x0f; *(contEncoded+1)=allowedComb->code_comb[ind];
}

unsigned char TCodingDecoding::doDecoding(unsigned short *contEncoded){
unsigned short *check_arr=new unsigned short [allowedComb->size];
unsigned short xor_rezult, value; unsigned char contDecoded;

for(int i=0; i<2; i++){
for(int j=0; j<allowedComb->size; j++){
xor_rezult=contEncoded[i]^allowedComb->code_comb[j];
*(check_arr+j)=getNumbTrueBits(xor_rezult);
}
value=getMinElemIndex(check_arr,allowedComb->size);
if ((i&1)==0) contDecoded=value<<4&0xF0;
else contDecoded|=value;
}

return contDecoded;
}

//private methods
unsigned short TCodingDecoding::getNumbTrueBits(unsigned short num){
int k=0;
for(int i=0; i<8*sizeof(num); i++)
if (num&1<<i) k++;

return k;
}

int TCodingDecoding::getMinElemIndex(unsigned short *arr,int size){
unsigned short minVal=*arr; int minInd=0;
for(int i=1; i<size; i++)
if(*(arr+i)<minVal){
minVal=*(arr+i);
minInd=i;
}
return minInd;
}

```

**/\*TGenerationError.h\*/**

```

#pragma once

class TGenerationError
{
unsigned short rezError;
unsigned char *errArray;
int length;

public:
TGenerationError(void){}
TGenerationError(int);
TGenerationError(TGenerationError&){}
~TGenerationError(){}

unsigned short getError(void);
void getErrorArray(unsigned char*);

private:
void setZero(void);

```

```
};
```

## **/\*TGenerationError.cpp\*/**

```
#include "StdAfx.h"
#include <iostream>
#include "TGenerationError.h"

TGenerationError::TGenerationError(int _length){
length=_length;
errArray=new unsigned char[length];
}

unsigned short TGenerationError::getError(void){
bool b; setZero();
for(int i=0; i<length; i++){
do{
    *(errArray+i)=rand()%15;
    b=false;
    for(int j=0; j<i; j++){
        if(errArray[j]==errArray[i]){
            b=true; break;
        }
    }
}while(b);
rezError|=1<<errArray[i];
}

return rezError;
}

void TGenerationError::getErrorArray(unsigned char* temp){
for(int i=0; i<length; i++)
*(temp+i)=*(errArray+i);
}

//private methods
void TGenerationError::setZero(void){
rezError=0x00;
for(int i=0; i<length; i++)
*(errArray+i)=0x00;
```