Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
(інститут)

Факультет інформаційних технологій
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня

магістра
(назва освітньо-кваліфікаційного рівня)

студента	*Анісімова Марка В'ячеславовича*
	(ПІБ)
академічної групи	*121М-20-1*
	(шифр)
спеціальності	*121 Інженерія програмного забезпечення*
	(код і назва спеціальності)
освітньої програми	*«Інженерія програмного забезпечення»*
	(назва освітньої програми)
на тему:	*Розробка методики розпізнавання відкритих, архівованих та шифрованих текстів на підставі дискретного перетворення Фур'є*

_____ *Анісімов М.В.*

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
розділів кваліфікаційної роботи	*Проф. Алексєєв М.О.*			
спеціальний	*Проф. Алексєєв М.О.*			
економічний	*Проф. Вагонова О.Г.*			
Рецензент				
Нормоконтролер	*Доц. Приходченко С.Д.*			

Дніпро
2022

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

ЗАТВЕРДЖЕНО:
Завідувач кафедри
Програмного забезпечення комп'ютерних систем
(повна назва)

_____ І.М. Удовик
(підпис) (прізвище, ініціали)

« » _____ 2022 Року

ЗАВДАННЯ
на виконання кваліфікаційної роботи магістра

спеціальності *121 Інженерія програмного забезпечення*
(код і назва спеціальності)

студенту *121м-20-1* *Анісімову Марку В'ячеславовичу*
(група) (прізвище та ініціали)

Тема дипломної роботи *Розробка методики розпізнавання відкритих, архівованих та шифрованих текстів на підставі дискретного перетворення Фур'є*

1 ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Наказ ректора НТУ «Дніпровська політехніка» від 10.12.2021 р. № 1036-с

2 МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень – цифрові сигнали систем передачі даних.

Предмет досліджень – програмне забезпечення для класифікації цифрових сигналів.

Мета роботи – розробка методики класифікації цифрових сигналів, які представляють відкриті, архівні і шифровані тексти, за допомогою дискретного перетворення Фур'є, пошук закономірностей, створення спеціалізованого програмного забезпечення для автоматизації розробки.

Вихідні дані для проведення роботи теоретичні та експериментальні дослідження дискретного перетворення Фур'є при обробці цифрових сигналів

3 ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Новизна запропонованих рішень полягає у:

– реалізації ефективного алгоритму класифікації цифрових сигналів, які представляють відкриті, архівовані і шифровані тексти.

Практична цінність результатів роботи результатів роботи полягає в створенні програмного забезпечення, що функціонує на основі розробленої методики аналізу цифрових сигналів, які представляють відкриті, архівні і шифровані тексти, з

використанням дискретного перетворення Фур'є, що істотно спрощує процес аналізу у порівнянні з сучасними системами.

4 ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Результати досліджень мають бути подані у вигляді, що дозволяє безпосереднє використання методики класифікації цифрових сигналів, які представляють відкриті, архівні і шифровані тексти,. Згідно виробничих функцій та професійних задач магістра, які виносяться на кваліфікаційну роботу, повинні бути розроблені програмна документація та відповідні програмні засоби.

5 ЕТАПИ ВИКОНАННЯ РОБІТ

Найменування етапів робіт	**Строки виконання робіт (початок – кінець**
Збір інформації для дослідження предметної області	04.09.2021 – 02.10.2021
Дослідження методів для вирішення поставленого завдання. Розробка методики класифікації цифрових сигналів	03.10.2021 – 04.11.2021
Розробка програмного забезпечення. Експериментальні дослідження	05.11.2021 – 23.12.2021
Економічна частина	24.12.2021 – 30.12.2021

Завдання видав _____ *Алексєєв О.М.*
(підпис) (прізвище, ініціали)

Завдання прийняла до виконання _____ *Анісімов М.В.*
(підпис) (прізвище, ініціали)

Дата видачі завдання: *03.09.2021 р.*

Термін подання дипломного проекту до ЕК _____

РЕФЕРАТ

Пояснювальна записка: 75 с., 3 додатки, 18 джерел.

Об’єкт досліджень: цифрові сигнали систем передачі даних.

Предмет досліджень: програмне забезпечення для класифікації цифрових сигналів.

Мета магістерської роботи: розробка методики класифікації цифрових сигналів, які представляють відкриті, архівні і шифровані тексти, за допомогою дискретного перетворення Фур'є, пошук закономірностей, створення спеціалізованого програмного забезпечення для автоматизації розробки.

Методи дослідження. При рішенні задач, які поставленні, виконано аналіз та наукове узагальнення літературних джерел по вихідним посиланням досліджень, використовувались метод спектрального перетворення Фур'є. У процесі досліджень виконувалось імітаційне моделювання на ПЕОМ.

Новизна отриманих результатів полягає у реалізації ефективного алгоритму класифікації цифрових сигналів, які представляють відкриті, архівні і шифровані тексти.

Практична цінність результатів роботи полягає в створенні програмного забезпечення, що функціонує на основі розробленої методики аналізу цифрових сигналів, які представляють відкриті, архівні і шифровані тексти, з використанням дискретного перетворення Фур'є, що істотно спрощує процес аналізу у порівнянні з сучасними системами.

Область застосування. Розроблене програмне забезпечення може використовуватися при контролі передачі інформації по будь-яких каналах обміну інформацією, наприклад, по інтернет каналах, з використанням мобільного зв'язку тощо.

Значення роботи та висновки. Розроблена методика для класифікації цифрових сигналів, які представляють відкриті, архівні і шифровані тексти та програмне забезпечення, та програмне забезпечення, яке її реалізує, дозволяє суттєво спростити процес аналізу архівних і шифрованих текстів у порівнянні з існуючими сучасними системами.

Прогнози щодо розвитку досліджень. Покращити метод класифікації, додавши процедуру аналізу спектрів Фур'є нейронними мережами.

У розділі «Економіка» проведені розрахунки трудомісткості розробки програмного забезпечення, витрат на створення ПЗ і тривалості його розробки, а також проведені маркетингові дослідження ринку збуту створеного програмного продукту.

Список ключових слів: програмне забезпечення, дискретне перетворення Фур'є, статистичний аналіз, спектральний фрагмент, архівація, шифрування, текстова послідовність.

ABSTRACT

Explanatory note: 75 pp., 3 appendices, 18 sources.

Object of research: digital signals of data transmission systems.

Subject of research: software for classification of digital signals.

The purpose of the master's work: development of methods for classification of digital signals that represent open, archival and encrypted texts, using discrete Fourier transform, search for patterns, the creation of specialized software to automate development.

Research methods. In solving the problems posed, the analysis and scientific generalization of literature sources based on the original references of research were performed, the method of spectral Fourier transform was used. In the process of research simulation simulation on a PC was performed.

The novelty of the obtained results is the implementation of an effective algorithm for the classification of digital signals that represent open, archival and encrypted texts.

The practical value of the results is to create software that operates on the basis of developed methods of analysis of digital signals representing open, archived and encrypted texts, using discrete Fourier transform, which greatly simplifies the analysis process compared to modern systems.

Scope. The developed software can be used to control the transmission of information through any information exchange channels, such as Internet channels, using mobile communications and the like.

The value of the work and conclusions. The developed methodology for classification of digital signals representing open, archived and encrypted texts and software, and the software that implements it, can significantly simplify the process of analysis of archived and encrypted texts compared to existing modern systems.

Forecasts for research development. Improve the classification method by adding a procedure for analyzing Fourier spectra by neural networks.

In the section "Economics" calculations of the complexity of software development, the cost of creating software and the duration of its development, as well as marketing research of the market for the software product.

Keyword list: software, discrete Fourier transform, statistical analysis, spectral fragment, archiving, encryption, text sequence.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ПЗ – програмне забезпечення;

АСОІ – автоматизовані системи обробки інформації;

ПК – персональний комп'ютер;

ДПФ – дискретне перетворення Фур'є;

РКЛ – розкладання Карунена-Лоева.

ЗМІСТ

ВСТУП

Актуальність роботи.

В даний час інформаційні технології та реалізовані з їх застосуванням програмні продукти дозволяють будь-якому власнику персонального комп'ютера (ПК) або аналогічного йому апаратного забезпечення отримати доступ до інформаційної мережі Internet.

Глобальна інформаційна мережа Internet – це сукупність взаємопов'язаних інформаційних вузлів, які забезпечують функціонування мережі як у цілому, і окремих її сегментів. Як вузли мережі можуть бути: підмережі - аналогічні Internet, але мають значно меншу масштабність, мейнфрейми - потужні інформаційно-обчислювальні станції, що забезпечують роботу значної кількості термінальних (віддалених) робочих станцій, функціонування найбільших баз даних і т.п., інформаційні сервери (портали) - вузли, що забезпечують введення інформації в інформаційну мережу, зберігання, обробку та надання її на запит користувачів мережі або інших вузлів мережі.

Однією з найважливіших функцій мережі Internet є надання необхідної інформації кінцевого користувача ПК. Виникає завдання здійснення автоматичного збирання, зберігання та обробки інформації на інформаційних вузлах мережі. В даний час ця задача вирішена з використанням автоматизованих систем обробки інформації (АСОІ), в основі функціонування яких лежать різні алгоритми: масковий аналіз, нейромережевий аналіз, реляційний метод, статистичний аналіз та ін. Практично у всіх перерахованих методів є значні недоліки (неможливість створення універсальної маски, складність побудови і навчання нейромережі, слабкі або відсутні зв'язки з вузлом, що надає інформацію, низький коефіцієнт ймовірності правильного аналізу і т.п.)

Ця робота присвячена розробці методики аналізу отриманої АСОІ інформації, основу якої становить метод дискретного перетворення Фур'є (ДПФ).

Мета магістерської роботи – розробка методики класифікації цифрових сигналів, які представляють відкриті, архівні і шифровані тексти, за допомогою дискретного перетворення Фур'є, пошук закономірностей, створення спеціалізованого програмного забезпечення для автоматизації розробки.

Об’єкт досліджень – цифрові сигнали систем передачі даних.

Предмет досліджень – програмне забезпечення для класифікації цифрових сигналів.

Методи дослідження: При рішенні задач, які поставленні, виконано аналіз та наукове узагальнення літературних джерел по вихідним посиланням досліджень, використовувались метод спектрального перетворення Фур'є. У процесі досліджень виконувалось імітаційне моделювання на ПЕОМ.

Новизна отриманих результатів полягає у реалізації ефективного алгоритму класифікації цифрових сигналів, які представляють відкриті, архівні і шифровані тексти.

Практична цінність результатів роботи полягає в створенні програмного забезпечення, що функціонує на основі розробленої методики аналізу цифрових сигналів, які представляють відкриті, архівні і шифровані тексти, з використанням дискретного перетворення Фур'є, що істотно спрощує процес аналізу у порівнянні з сучасними системами.

Особистий внесок автора полягає в розробці теоретичної частини роботи магістра, в дослідженні і систематизації знання про існуючі методики, розробці програмної реалізації, в оцінці отриманих результатів.

Структура та обсяг дипломної роботи. Робота складається з вступу, чотирьох розділів і висновків. Містить 76 сторінок друкованого тексту, в тому числі 60 сторінок тексту основної частини з 9 рисунками, списку використаних джерел з 27 найменуваннями на 2 сторінках, 3 додатків на 14 сторінках.

РОЗДІЛ 1.
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАВДАННЯ

1.1. Постановка задачі на розробку програми

Розробити алгоритм та програмно реалізувати додаток, що дозволяє автоматизувати процес аналізу текстових послідовностей спектральним методом, отриманих у процесі збору текстової інформації з різних інформаційних джерел. Надати експертну оцінку отриманим результатам. Визначити імовірність правильної ідентифікації спектральних ознак текстової послідовності. Забезпечити можливість введення текстів для обробки програмою як окремих файлів. Кожен файл – це текст, який відноситься до одного з трьох типів: відкритий, архівний або шифрований. Результати роботи програми вивести на екран.

1.2. Стан питання

При створенні сучасних аналітичних систем збору та аналізу інформації виникає завдання визначення типу переданих даних: відкриті, архівовані та шифровані. Таке завдання можна вирішити з використанням статистичних методів.

У роботі пропонується оригінальний підхід до класифікації типів відкритих, архівованих та шифрованих даних із застосуванням спектрального аналізу.

Теоретично в криптоаналізі одним із найпоширеніших критеріїв оцінки якості алгоритму шифрування є перевірка рівномірності розподілу бітів у шифртексті. Існують різні методи оцінки, наприклад: критерій Колмогорова-Смирнова, критерій Пірсона χ^2 тощо.

Суть останнього ось у чому. Нехай $X = (X_0, X_1, ..., X_{N-1})$ незалежні та ідентично розподілені випадкові змінні, що приймають значення на множині

$a = (a_0, a_1, ..., a_{m-1})$ з невідомою ймовірністю розподілу. χ^2 -тест використовується для того, щоб визначити чи узгоджуються результати спостережень змінних з гіпотезою рівності змінних X з значенням з множини $a = (a_0, a_1, ..., a_{m-1})$ з ймовірністю *p(j)*: $P\{X = a_j\} = p(j)$ $0 \le j < m-1$, где *p(j)*- ймовірність розподілу на множині $\{a_0, a_1, ..., a_{m-1}\}$.

Теорема, доведена Пірсоном, є граничною теоремою, яка говорить, що якщо *{p(j): 0 <= j < m}* – імовірнісний розподіл , то функція розподілу еталона

$$\Pr\{\chi^2 \le x\} = \Pr\left\{ \sum_{0 \le j < m} \frac{\left(N_{a_j}(X) - np(j)\right)^2}{np(j)} \le x \right\}, \qquad (1.1)$$

сходиться до інтегралу

$$\int_0^x k_{m-1}(y)dy, \qquad (1.2)$$

де χ^2 – це випадкова змінна.

При використанні χ^2 - тесту висувається дві гіпотези: нуль гіпотеза - $H_0 : \{p(j) : 0 \le j < m\}$ ймовірно є розподілом $X_0, X_1, ..., X_{N-1}$ і альтернативна гіпотеза ймовірно не є розподілом $X_0, X_1, ..., X_{N-1}$.

Обчислення проводяться в такий спосіб. Спочатку, за формулою (3) обчислюється χ^2 для $X_0, X_1, ..., X_{N-1}$. Потім порівнюється χ^2 з $\chi_{p,m-1}$, яке обчислюється за формулою (3), і приймається H_0 якщо $\chi^2 \le \chi_{p,m-1}$, інакше приймається гіпотеза H_1.

Для завдання розпізнавання типу документа даний тест можна застосувати в такий спосіб. Відомо, що алгоритми шифрування сильно згладжують статистичну неоднорідність тексту. На противагу цьому відкритий текст має сильну неоднорідну структуру. Стислі дані мають менш неоднорідну структуру, ніж відкритий текст, але розподіл у них нулів і одиниць не має бути однорідним, так як завданням стиснення даних є зняття надмірності кодування символів, а не

зняття неоднорідності. Користуючись оцінкою χ^2 тесту можна за рівномірністю розподілу віднести аналізований текст до того чи іншого типу.

Альтернативним методом при дослідженні випадкових процесів і, зокрема, при визначенні рівномірності послідовності нулів і одиниць для перерахованих вище типів файлів, можна використовувати розкладання даних послідовностей в ряд Фур'є.

Як відомо, гармонійний аналіз отримав широке поширення при дослідженні невипадкових періодичних функцій. Тобто. Будь-яка функція з періодом Т, що задовольняє умові Діріхле, може бути розкладена в ряд Фур'є:

$$\boldsymbol{f(t) = \sum_{k=0}^{\infty} c_k \, exp(\frac{i2\pi kt}{T})}, \qquad (1.3)$$

Тобто. функцію f(t) можна представити у вигляді ряду (4) гармонійних коливань з частотами $w = \frac{2pk}{T}$ і амплітудами c_k.

Стаціонарний випадковий процес *X(t)* на проміжку [0,T] можна представити також у вигляді нескінченного ряду гармонійних коливань з різними частотами та випадковими амплітудами Y_k:

$$X(t) = \sum_{k=0}^{\infty} Y_k \exp(\frac{i2\pi kt}{T}), \qquad (1.4)$$

Наведені вище вирази справедливі не тільки для безперервних випадкових процесів, але і для дискретних, тобто, коли значення випадкової величини визначені тільки у фіксовані моменти часу. Позначимо таку дискретну величину через X_n. Тоді інтеграл заміниться сумою

$$\boldsymbol{Y_k = \sum_{k=0}^{N} X_k \, exp(\frac{-2\pi nk}{N})}, \qquad (1.5)$$

Розглянемо три типи файлів, представлених у вигляді послідовності нулів і одиниць: відкриті текстові файли, текстові файли заархівовані яким-небудь архіватором, текстові файли закриті за допомогою американського стандарту DES або українського стандарту ДСТУ ГОСТ 28147:2009. Перетворимо таку послідовність на нову, замінивши нульові значення на (-1).

До даних процесів застосовні методи аналізу випадкових процесів на основі розкладання їх у гармонійний ряд із застосуванням ряду методів, що використовуються при обробці звичайних сигналів.

У разі виконання аналізу виду інформації з використанням критерію Пірсона проводиться підрахунок кількості нулів і одиниць на ділянці тексту об'ємом 1024 біт, обчислюється значення χ^2 за формулою (3) і порівнюється з еталонним значенням. Отримані результати дають низький коефіцієнт правильного розпізнавання. Алгоритми шифрування, зокрема DES, який завжди рівномірно розподіляють дані. На деяких тестових даних нерівномірність розподілу можна порівняти з нерівномірністю відкритого тексту. В архівованих даних виявляються перекриття із зашифрованим та відкритим текстами.

Виходячи з наявності в певних місцях піків на кривій, а також їх інтенсивності, можна визначити, чи є файл відкритим, не зашифрованим текстовим файлом. Крім того, по амплітуді піків відкритого тексту можна зробити висновок про статистичний розподіл букв у тексті, його мовної приналежності та ряд інших властивостей.

Цікавим завданням є можливість поділу архівованих і шифрованих текстових файлів, що мають досить рівномірний розподіл нулів та одиниць. Аналіз амплітудного спектра різних типів файлів показує існуючі істотні відмінності в їх спектральних складових.

При використанні статистичного методу із 150 тестових прикладів правильне розпізнавання зазначено у 70% випадків. При використанні спектрального методу – 90%.

1.3. Опис структури математичної моделі, що застосовується

У цій роботі застосовується трирівнева математична модель системи пошуку спектральних складових текстових послідовностей. Для отримання кінцевого результату повинні бути виконані всі три етапи аналізу вихідної текстової смислової послідовності.

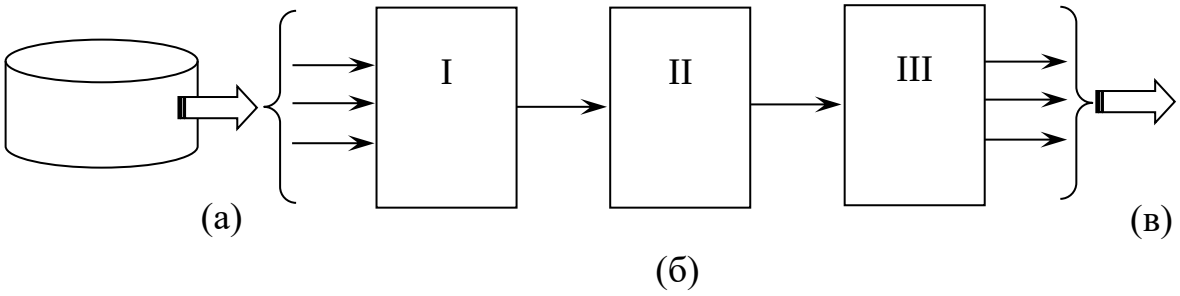

Рис. 1.1. Структурна схема моделі

Опис моделі:

Вихідна текстова послідовність (ТП) завантажується в програму з якогось носія інформації або інформаційного вузла (пункт (а)). Далі ТП розрізається на окремі фрагменти встановленого розміру та один із фрагментів передається на блок підготовчого етапу (пункт (б) етап I).

У процесі підготовчого етапу ТП перетворюється на дискретний сигнал, який надходить у модуль спектрального перетворення (пункт (б) етап II) у вигляді кінцевого масиву дискретних відліків.

У цьому модулі масив дискретних відліків перетворюється методом СПФ (пункт (б) етап III), і результуючий масив дискретних відліків надходить у модуль пошуку та аналізу спектральних ознак.

В останньому модулі (пункт (в)) проводиться остаточний аналіз перетвореної ТП і формується висновок про її тип.

1.3.1. Опис алгоритму функціонування програми

Отримавши вихідну текстову послідовність, програма починає її обробку із створення спеціальних сегментів у ТП. Розмір сегмента грає важливу роль - від нього залежить точність і швидкість виконання аналізу. У процесі дослідження було встановлено оптимальну величину сегмента, що дорівнює 256 відлікам – мікро елементам вихідної ТП (символам).

Для спрощення роботи програми розглядається лише один (перший) сегмент.

Виділивши сегмент послідовності, програма представляє його у вигляді кінцевого масиву дискретних відліків, отриманих шляхом відновлення ASCII - кодів символів сегмента ТП. Результат цього відновлення, а саме масив дискретних відліків, піддається спектральному перетворенню за методом Фур'є, в результаті чого виходить спектральний фрагмент (СФ).

УФ надходить у фінальний блок програми, де проводиться його аналіз, виконується пошук та виділення спектральних ознак вихідної ТП.

За підсумками виділених спектральних ознак робиться висновок про тип вихідної текстової послідовності.

1.3.2. Опис технологій та мов програмування, що використовуються при розробці програми.

Середовищем програмування для розробленого програмного комплексу був обраний продукт фірми Borland Delphi 10.4. (2020 рік)

1.3.3. Опис та обґрунтування вибору організації вхідних та вихідних даних

Вхідними є відкриті, архівні або шифровані текстові послідовності, представлені у вигляді файлів у текстовому форматі (TXT). Для зручності

реалізовано метод динамічної зміни вихідного тексту в інтерактивному режимі у процесі роботи з програмою.

Вихідними даними є:

- графічне відображення спектрального аналізу вихідної ТП як графічних файлів у форматі BMP;

- математична форма результатів аналізу (виведення на екран деяких спектральних ознак вихідної та проаналізованої ТП).

1.4. Подальші напрями досліджень

При подальшому поглибленому розгляді реалізованого методу можна відзначити, що теоретично можливе розпізнавання як типу тексту, а й різні його характеристики:

- для відкритого тексту – мова, статистичний розподіл букв у тексті тощо;

- для архівного тексту - метод архівації, його ефективність;

- для шифрованого тексту – принцип шифрування. При запровадженні деякої бази стандартів планується провести дослідження можливість розпізнавання алгоритму шифрування.

РОЗДІЛ 2. АНАЛІЗ МЕТОДИКИ ШИФРУВАННЯ

2.1. Застосування розкладання Карунена-Лоева (РКЛ) у вирішенні задачі формування ознак

Основна ідея застосування РКЛ при вирішенні завдання формування інформативних ознак для реалізацій випадкових процесів полягає у приписуванні ваг ознакам відповідно до їх відносної важливості, яка розуміється в сенсі отримання меншої помилки апроксимації порівняно з іншими розкладаннями та отримання більшої інформації щодо розрізнення класів. Розкладання дозволяє уникнути знання щільностей розподілу образів, які входять у окремі класи.

Розглянемо процедуру побудови базису, що задовольняє співвідношення Карунена-Лоева. Припустимо, що випадкові вектори $X_1, \ldots, X_M$ представляють спостереження, які стосуються одному з класів $m = \overline{1, M}$. Можна отримати розкладання в узагальнений ряд Фур'є за системою базових функцій

$$X_i = \sum_{j=1}^{N} y_{ij} \psi_i, \qquad (2.1)$$

де X_i- N-мірний вектор;

y_{ij} - Випадкові коефіцієнти;

ψ_i - N-мірний вектор.

Якщо коефіцієнти подати у векторній формі , то (2.1) можна подати в матричному записі

$$Y_i = [y_{i1}, \ldots, y_{iN}]^T, \qquad (2.2)$$

де $\Psi = [\psi_1, \psi_2, \ldots, \psi_N]$.

Кореляційна матриця Rx для аналізованих векторів визначається як

$$R_x = \sum_{P_{E[X^m X^{mT}]m}}^{M\Sigma} \quad m=1 \quad , \qquad (2.3)$$

яка задовольняє дискретному аналогу рівняння Фредгольма 2 роду

$$R_x \psi_j = \lambda_j \psi_j, \qquad (2.4)$$

За заданою кореляційною матрицею R_x можуть бути визначені власні значення λ_j та відповідні їм власні вектори ψ_j. Оскільки базисні вектори є власні вектори дійсної симетричної матриці, всі вони взаємно ортогональні. Якщо вони, крім того, ортонормовані, то

$$\psi_j^T \psi_k = \begin{cases} 1, j = k \\ 0, j \neq k. \end{cases}, \qquad (2.5)$$

Виходячи з останньої умови, коефіцієнти розкладання визначаються таким чином

$$\mathrm{Y_i} = \Psi^{\mathrm{T}} \mathrm{X_i}\,, \qquad (2.6)$$

У завданнях стиснення випадкових процесів коефіцієнти РКЛ розглядаються як ознаки, що представляють вектор, що спостерігається $\mathrm{X_i}$. Ці ознаки мають такі властивості [17]:

- ефективність ознаки визначається відповідним значенням. Якщо ознака виключається з розкладання, середньоквадратична помилка збільшується на λ_j. Тому при зменшенні числа ознак ознаку з найменшим значенням слід виключити насамперед і т.д. Таким чином, ознаки впорядковуються за важливістю відповідно до порядку убування власних чисел ($\lambda_1 > \lambda_2 > ... > \lambda_N > 0$;

оскільки $\mathrm{Y_i} = \Psi^{\mathrm{T}} \mathrm{X}$, то коваріаційна матриця вектора Y визначається наступним чином [177]:

$$R_y = \Psi R_x \Psi^T \qquad (2.7)$$

Матриця Ψ складається з власних векторів R_x і, отже

$$R_y = diag(\lambda_1, \lambda_2, \ldots, \lambda_N), \qquad (2.8)$$

Якщо R_y - діагональна матриця, то неважко дійти висновку, що коефіцієнти перетворення Карунена-Лоева y_{ij} взаємно не кореловані. В окремому випадку, коли випадкові вектори $\{X_i\}$ розподілені нормально, коефіцієнти взаємно незалежні;

- власні вектори коваріаційної матриці R_y дають найменше значення середньоквадратичної помилки $\bar{\varepsilon}^2(K)$ серед усіх ортонормованих векторів.

$$\bar{\varepsilon}^2(K)_{\min} = \sum_{j=k+1}^{N} \lambda_j \qquad (2.9)$$

- розкладання Карунена-Лоева забезпечує максимальну нерівномірність дисперсій ознак. Ентропія розподілу енергії за координатами розкладання Карунена-Лоева мінімальна порівняно з іншими лінійними перетвореннями

$$S(A) = -\sum_{k=1}^{N} \rho_k \, log \, \rho_k \geq -\sum_{k=1}^{N} \lambda_k^2 \, log \, \lambda_k^2 = S_{к-л} \qquad (2.10)$$

Ентропійний показник характеризує ступінь розподілу базисних функцій перетворення Карунена-Лоева за рівнем їх важливості, а дисперсія коефіцієнтів – ступінь уявлення всього процесу даної конкретної координатою. За будь-якого іншого базису ентропія зростає. Одночасно з мінімумом ентропії в оптимальному базисі досягається мінімальна середньоквадратична похибка наближення при заданому числі членів ряду розкладання порівняно з усіма ортогональними перетвореннями.

Для того, щоб розкладання Карунена-Лоева було оптимальним базисом, необхідно виконання умови $E\{Y_i\}=0$ (математичне очікування вектора спектральних коефіцієнтів дорівнює нулю) або рівносильної умови, що математичне очікування вихідних векторів також дорівнює нулю $E\{X_i\}=0$ [17]. В окремих випадках проблему можна вирішити, якщо центрувати вектори, що належать різним класам, щодо відповідних математичних очікувань. Однак при вирішенні завдань розпізнавання (класифікації) практично завжди відсутні відомості про те, до якого класу належить вектор X_i, що класифікується. Винятком є етап навчання при побудові базису Карунена-Лоева. Тому центрування векторів навчальної вибірки перед визначенням оцінки кореляційної матриці не принесе очікуваного ефекту, крім окремого випадку, коли математичні очікування для векторів всіх класів рівні [17]. Природно, це припущення обмежує можливості застосування базису Карунена-Лоева на формування спектральних ознак при класифікації сигналів.

З енергетичної точки зору базис Карунена-Лоева дозволяє забезпечити найбільшу концентрацію енергії випадкового процесу у мінімальній ширині смуги. В рамках спектрально-кореляційної теорії випадкових процесів це означає, що енергетичний спектр деякого класу процесів, що описується матрицею підступів, зосереджений у мінімумі коефіцієнтів.

При використанні перетворення Карунена-Лоева у завдання формування інформативних ознак з метою класифікації або стиснення вихідних даних інформативними вважаються такі ознаки, значення яких найбільше змінюються для різних реалізацій, так як ці ознаки несуть найбільше інформації про відмінність вихідних даних. Тому ознаки, у яких дисперсія коефіцієнтів розкладання (або самі коефіцієнти розкладання) менші за деякий поріг, відкидаються. Таким чином, забезпечується стиснення інформації, і вектор вихідних даних може бути адекватно представлений спектральними ознаками у просторі меншої розмірності (у підпросторі спектральних ознак).

З виразу (2.9) випливає, що ефективність коефіцієнта перетворення Карунена-Лоева y_i для представлення вектора даних X визначається

відповідним власним значенням. Якщо коефіцієнт y_k не враховується, то середньоквадратична помилка збільшується на відповідне власне значення λ_k. Таким чином, при стисканні процесів управління необхідно вибрати множину y_i, відповідне M найбільшим коефіцієнтам та інші y_i відкинути, оскільки їх можна замінити константами B_j, $j = \overline{M+1, N}$. Якщо вихідні дані попередньо центрувати, $(\overline{X}) = 0$ то $y_j = 0$ для $j = \overline{M+1, N}$. Оскільки власні значення є елементами, що стоять на головній діагоналі матриці власних значень R_y, вони відповідають дисперсії коефіцієнтів перетворення. Для решти перетворень матриця містить ненульові позадіагональні елементи. Тому природним критерієм при виборі безлічі коефіцієнтів перетворення є збереження коефіцієнтів з найбільшими дисперсіями [17]. Графічним уявленням дисперсійного критерію є графік дисперсій коефіцієнтів перетворення, де дисперсії розташовані в порядку зменшення та нормовані до сліду матриці R_x або R_y.

Зазначимо, що оптимальність по Карунену-Лоеву розглядається за ансамблем реалізацій випадкового процесу, тому відновлення окремих реалізацій з використанням власних функцій може призвести до значних відмінностей у формі сигналу. Зокрема, в тих завданнях, де при стисненні випадкових процесів критерієм є обмеженість максимальної помилки, розкладання Карунена-Лоева може бути непридатним, оскільки воно має статистичними властивостями, що згладжують в просторі L_2 і не гарантує найвищої збіжності в просторі C.

Оскільки перетворення Карунена-Лоева не можна обчислити за допомогою швидких алгоритмів, доцільно розглянути застосування для вирішення практичних завдань традиційних базисів Фур'є, Уолша і Хаара, які мають високу обчислювальну ефективність через наявність швидких алгоритмів. З іншого боку, окремих типів випадкових процесів ці базиси оптимальні у сенсі Карунена-Лоева.

2.2. Застосування базисів Фур'є, Уолша, Хаара для отримання простору інформативних ознак

Застосування традиційних базисів на формування інформативних ознак при стисканні випадкових процесів пояснюється, переважно, такими причинами:

- функції традиційних базисів мають простий вигляд, який залежить від елементів матриці підступів вихідних данных;

- базиси мають алгоритми швидких перетворень [9], що забезпечує їх обчислювальні переваги перед базисом Карунена-Лоева;

- традицією їх використання у вирішенні завдань стиснення випадкових процесів.

Слід зазначити, що, незважаючи на зовнішню подібність у поведінці традиційних базисів, між функціями цих базисів існує важлива відмінність, що полягає у природі їхньої освіти.

В основі широкого використання перетворення Фур'є на формування спектральних ознак лежить фундаментальна властивість інваріантності тригонометричного базису до циклічного зсуву. Інваріантність процесів однієї групи зрушень є необхідною умовою для зведення динамічного завдання до статичної [15].

Сутність відмінності функцій Уолша і Хаара полягає в тому, що вони, у загальному випадку, не мають цієї властивості. Наслідком зазначеної відмінності для базису Уолша є властивість мультиплікативності, з якої випливає властивість діадного зсуву функцій.

Те, що функції традиційних базисів не залежать від статистики випадкового процесу, є, з одного боку, їх гідністю, оскільки забезпечує просту програмну та апаратну реалізацію при обчисленні спектральних коефіцієнтів. З іншого боку, фіксований вид ортогональних функцій не дозволяє забезпечити у багатьох випадках високу швидкість збіжності рядів розкладання.

Розглянемо, яких типів інформаційних процесів управління перетворення на традиційних базисах є власними перетвореннями, тобто задовольняють співвідношенню Карунена-Лоева.

Нехай випадковий процес описується позитивно визначеною, симетричною матрицею підступів R розміром $N * N$, яка в спектральному поданні Карунену-Лоеву має діагональний вигляд. При використанні матричного варіанта рівняння Фредгольма 2 роди маємо [17]:

$$R = \Psi \Lambda \Psi^* \, , \Psi^* R \Psi = \Lambda, \qquad (2.11)$$

Підставивши в це рівняння замість оптимального по Карунену-Лоеву базису систему базисних функцій Фур'є F, Уолша W, Хаара H, можна отримати наступні рівняння [170]:

$$\begin{gathered} R = FC_F F^*, C_F = F^* RF, \\ R = WC_W W, C_W = WRW, \\ R = HC_H H^T, C_H = H^T RH, \end{gathered} \qquad (2.12)$$

Де C_F, C_W, C_H - відповідно матриці підступів спектральних коефіцієнтів у базисах Фур'є, Уолша, Хаара.

У випадку ці матриці не забезпечують декореляцію взаємозв'язків у вихідних даних. При цьому через наявність позадіагональних елементів важко вибрати найбільш значущі функції базису. Виняток із розгляду позадіагональних елементів призводить до зростання середньоквадратичної $\bar{\varepsilon}^2$ помилки та ентропії S коефіцієнтів розкладання, що зменшує їхню інформативність.

У [15] розглянуто питання, для яких структур коваріаційних матриць діагоналізується перетворення (2.29) у традиційних базисах Фур'є, Уолша і Хаара, тобто для яких класів випадкових процесів ці базиси оптимальні у сенсі Карунена-Лоева. Показано, що базис Фур'є оптимальний по Карунену-Лоеву для класу періодично стаціонарних випадкових процесів, у яких статистичні характеристики другого порядку описуються матрицею підступів, що мають циклічну структуру (циркулянт): $R_u = \{r_{tu}\} = \{ \ r_{|u-t| \ mod \ N}\} = \{ \ r_{|Nu-t|}\}$.

Елементи рядків у матриці циркулянтного типу розставляються за правилом циклічного зсуву елементів першого рядка, які є відліками кореляційної функції $R = [r_1, r_2, \ldots, r_N]$. Наприклад, для N=4 циркулянтна матриця має вигляд:

$$R_ц = \begin{bmatrix} r_1 & r_2 & r_3 & r_4 \\ r_4 & r_1 & r_2 & r_3 \\ r_3 & r_4 & r_1 & r_2 \\ r_2 & r_3 & r_4 & r_1 \end{bmatrix} \qquad (2.13)$$

Базис Уолша оптимальний для стаціонарних випадкових процесів, що мають діадну матрицю коваріацій $R_д = \{r_{ut}\} = \{r_{|u \oplus t| \, mod \, 2}\}$: Елементи рядків у матриці такого типу розставлені за правилом діадного зсуву. Для випадку *N*=4 матриця має вигляд:

$$R_д = \begin{bmatrix} r_1 & r_2 & r_3 & r_4 \\ r_2 & r_1 & r_4 & r_3 \\ r_3 & r_4 & r_1 & r_2 \\ r_4 & r_3 & r_2 & r_1 \end{bmatrix} \qquad (2.14)$$

Діадна матриця підступів симетрична щодо обох діагоналей і базис її діагоналізуючий є важливо речовим.

Базис Хаара забезпечує діагональне уявлення для коваріаційної матриці наступної структури (при *N*=4):

$$R_x = \begin{bmatrix} r_1 & r_2 & r_3 & r_4 \\ r_2 & r_1 & r_3 & -r_4 \\ r_3 & r_3 & r_1 + r_2 & 0 \\ r_4 & -r_4 & 0 & r_1 - r_2 \end{bmatrix} \qquad (2.15)$$

Застосування базисів Фур'є та Уолша за умови їхньої оптимальності у сенсі Карунена-Лоева відрізняється високою ефективністю завдяки наявності алгоритмів швидкого перетворення. Базис Хаара, який є традиційним, також має алгоритм швидкого перетворення з найбільшою обчислювальною ефективністю. У [17] зроблено спробу визначити клас процесів, що відповідають оптимальності

розкладання по функцій Хаара, проте структура матриці, яка діагоналізується цим базисом для будь-якого N у загальному випадку невідома.

Діагоналізовані в традиційних базисах матриці підступів не повністю враховують особливості описуваної статистики процесу. Вся інформація про випадковий процес міститься в першому рядку матриці підступів, а інші рядки визначаються прийнятим законом перестановки елементів першого рядка. При формуванні зазначених базисів вид координатних функцій визначається структурою коваріаційної матриці і не залежить від її елементів. Тому на практиці часто спостерігається слабка збіжність розкладів у традиційних базисах, зокрема, у базисі Фур'є, який використовується як основний.

Як уже зазначалося у розділі 2.1, інформаційні процеси управління, моделлю яких є періодично корельований випадковий процес, мають періодичні у часі математичне очікування та кореляційну функцію. Цей факт полегшує обґрунтований вибір базису з традиційних базисів, що широко використовуються. Застосування базису Фур'є для стиснення інформаційних процесів на інтервалі, що дорівнює періоду корелювання випадкового процесу, дозволяє оцінити коефіцієнти розкладання періодичного в часі математичного очікування (2.4). Запропонований підхід для стиснення випадкових процесів дозволяє отримати простір ознак невисокої розмірності, проте вимагає попереднього визначення періоду корелювання інформаційного процесу управління. Ця обставина може ускладнити оперативне стиснення процесу, тому згаданий підхід, який розширює можливості застосування традиційного базису Фур'є, не можна вважати універсальним.

Застосування базису Уолша є доцільним у разі потреби стиснення дворівневих керуючих сигналів [4,10]. Найкращі результати в порівнянні з іншими традиційними базисами базис Уолша забезпечує при стисканні дельта-модульованих сигналів [8].

Насамкінець доцільно коротко розглянути обчислювальні витрати при використанні традиційних базисів для обчислення спектральних характеристик. Відомо, що це зазначені базиси мають алгоритми швидких перетворень,

дозволяють значно скоротити кількість обчислень проти дискретним варіантом перетворення Фурье. Проте базис Фур'є вимагає виконання $N \log_2 N$ операцій комплексного множення-складання, а базиси Уолша і Хаара лише операції складання-віднімання. Слід зазначити, що базис Хаара в силу природи своїх функцій має найбільш швидкодіючий алгоритм швидкого перетворення 2(N-1) операцій складання-віднімання замість $N\log_2 N$ операцій складання-віднімання в базисі Уолша. Крім того, застосування базису Фур'є в деяких випадках вимагає використання різних вікон для достовірного оцінювання спектра. При стисканні інформаційних процесів управління чинник тимчасових витрат грає значної ролі, а разі однакової інформативності спектральних ознак у різних базисах стає вирішальним.

2.3. Модифіковане перетворення Уолша

До найпоширеніших завдань спектрального аналізу сигналів ставляться отримання показників, інваріантних до різноманітних спотворень інформації, зокрема до зрушення початку відліку. Для одномірних сигналів – це інваріантність до тимчасового зсуву, для просторових – інваріантність до зміщення чи перенесення даних у просторі. У спектрально-кореляційному аналізі випадкових процесів як характеристики, інваріантні до циклічного зсуву, застосовується автокореляційна функція та енергетичний спектр. Використання нелінійних операцій при їх утворенні (множення, зведення в квадрат) призводить до втрати фазової інформації сигналів, що забезпечує інваріантність до тимчасового зрушення.

Спектральний підхід до отримання інваріантних характеристик має ту перевагу, що забезпечує декореляцію коефіцієнтів розкладання, а також високу швидкість їх обчислення за рахунок використання алгоритмів швидких перетворень (БП).

Властивість інваріантності перетворення Фур'є до тимчасового зсуву (до циклічного зсуву для цифрових сигналів) дозволяє звести динамічне завдання до

статичної та охарактеризувати кожен клас сигналів еталоном у просторі спектральних ознак Фур'є.

Однак при оперативному контролі параметрів ТОУ та при великих розмірах цифрових сигналів отримання енергетичного спектра стає менш ефективним унаслідок суттєвих витрат часу на обчислення коефіцієнтів Фур'є та їх квадратів. Крім операції зведення квадрат для отримання інваріантних характеристик можуть бути використані інші нелінійні залежності. Підхід, заснований на поєднанні лінійних та нелінійних перетворень, значною мірою розширює поле пошуку найбільш вигідних перетворень за умовами конкретного завдання [17,18].

За допомогою методу узагальнених спектральних перетворень в ядерних уявленнях матричних операторів, запропонованого [170], створюється можливість проведення спрямованого пошуку БП, забезпечується інваріантність до циклічного зсуву. При цьому завдання може бути сформульована як задача синтезу нелінійних БП, що забезпечують інваріантність амплітудної характеристики циклічного зсуву відліків вихідного сигналу. Цифрова спектральна обробка одновимірних сигналів формально призводить до необхідності обробки періодичних послідовностей даних розміром 1xN. Постановка задачі полягає в наступному: нехай вихідний сигнал у лінійному просторі може бути:

$$X_t = AY \qquad (2.16)$$

де $A = [A_1, A_2, \ldots, A_N]^T$ □ квадратна матриця спектрального оператора, розміром, що факторизується на матриці однакової структури:

$$A \quad = G_n \cdot G_{n-1} \cdot \ldots \cdot G_1 = \prod_{r=1}^{n} G_r \qquad (2.17)$$

$Y = [y_1, y_2, \ldots, y_N]^{\boldsymbol{T}}$ - Вектор, компоненти якого є дискретним спектром вектора X_t

Використовуючи поєднання нелінійної функції $Q_r(.)$ і лінійного перетворення G_r на кожному r-му етапі обчислення коефіцієнтів загального перетворення Z, потрібно знайти умови, що накладаються на Q(.) і A для забезпечення інваріантності Z, до циклічного зсуву відліків вихідного вектора X_t. Інакше кажучи, це означає, що

$$Z = Q(AX_t) = Q(AX_{(t+\tau)modN}) \qquad (2.18)$$

При цьому необхідно знайти найпростішу функцію Q(.), що забезпечує мінімум обчислювальних витрат при отриманні (2.36).

На рис. 2.3 наведено граф обчислень швидкого перетворення Уолша (БПУ) для $N=2^n$ ($n=3$), що пояснює принцип формування факторизованої матриці G_r [46]:

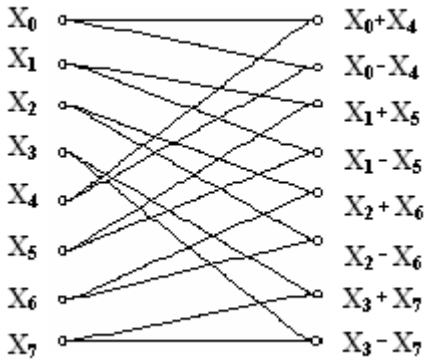

Рис. 2.3 Граф швидкого перетворення Волша

$$\mathbf{G}_r(t,l) = \begin{bmatrix} 1 & * & * & * & 1 & * & * & * \\ 1 & * & * & * & -1 & * & * & * \\ * & 1 & * & * & * & 1 & * & * \\ * & 1 & * & * & * & -1 & * & * \\ * & * & 1 & * & * & * & 1 & * \\ * & * & 1 & * & * & * & -1 & * \\ * & * & * & 1 & * & * & 1 & * \\ * & * & * & 1 & * & * & -1 & * \end{bmatrix} = (-1)^{\,t_r l_r} \qquad (2.19)$$

$$(t_r, l_r) \in 0,1;\ t = (t_n, t_{n-1}, \ldots t_1);\ l = (l_n, l_{n-1}, \ldots l_1) \quad (2.20)$$

Структури факторизованих матриць G_r однакові всім r=1…n, і це дозволяє уніфікувати процедуру (2.36). Ненульові елементи (±1) у мікроблоках визначаються відповідно до ядра Уолша-Адамара

$$W = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad (2.21)$$

і вимагають лише виконання операцій складання-віднімання при реалізації алгоритму БП Волша-Адамара. На кожній з ітерацій поряд з лінійними операціями складання-віднімання також використовуватимемо іншу пару перетворень, нелінійну щодо відліків як вихідного вектора $X_t=X^0_t$, так і проміжних векторів. Порядок чергування нелінійних операцій $Q_0(.)$ та $Q_1(.)$ для кожної з пар відліків (x_t, $x_{t+N/2}$) вектора X_t, (t=0, N-1) виберемо відповідно до виду виконаної раніше лінійної операції. $Q_0(.)$ як нелінійну операцію для сумованих відліків (x_t, $x_{t+N/2}$), а $Q_1(.)$ - у разі відрахованих відліків (x_t, $x_{t+N/2}$). Оскільки порядок чергування $Q_0(.)$ і $Q_1(.)$ кожної r-й ітерації однаковий, то загальному вигляді нелінійна функція може бути представлена формулою [15]:

$$Q_{l_r} = \begin{cases} Q_0, & l_r = 0 \\ Q_1, & l_r = 1 \end{cases}, r = \overline{1,n} \quad (2.22)$$

При використанні спектральних коефіцієнтів як діагностичні ознаки доцільно використовувати такі перетворення, які забезпечують мінімум обчислювальних витрат.

Використання поєднання лінійного перетворення G_i (матриць Гуда) і нелінійної функції Q_i на кожному і етапі процедури БПУ має забезпечити інваріантність вектора спектральних коефіцієнтів Y до циклічного зсуву відліків вектора вихідного сигналу X_t.

Розглянемо умови для забезпечення співвідношення

$$Y = Q(AX_t) = Q(AX_{(t+\tau)\mathrm{mod}N}) \quad (2.23)$$

Вектор Y може бути отриманий із співвідношення

$$\mathbf{Y} = [y_1, y_2, ..., y_N]^T = Q_n[...Q_2[\mathbf{G}_2 Q_1[\mathbf{G}_1 \mathbf{X}_t]]] \quad (2.24)$$

Граф обчислення вектора N=8 представлений на рис.2.4.

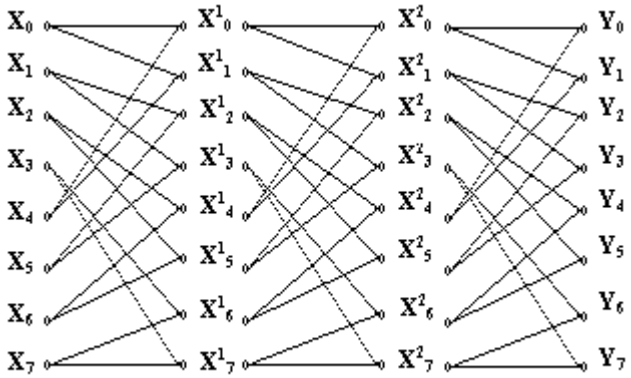

Рис. 2.4 Граф обчислення вектора Y

Покажемо, що циклічне зрушення вихідних даних (компонент вектора X_t.) на довільну величину δ не призводить до утворення таких пар відліків, яких немає при нульовому зсуві δ. Перетворення вихідних та проміжних відліків при $\delta \neq 0$ можна представити так:

- для першої ітерації

$$Z_1(\delta) = Q_1(x_{t+\delta}, x_{t+\delta+N/2}) \quad (2.25)$$

- для кожної наступної i-ої ітерації

$$Z_i(\delta) = Q_i(Z_{i-1}(\delta), Z(\delta + N/2)); \quad (2.26)$$

При $\delta \neq 0$ вектор вихідних відліків набуде вигляду

$$X \quad = [x_{N-\delta}, x_{N-\delta-1}, \ldots, x_0 \ldots]^{T} \qquad (2.27)$$

Як аргументи нелінійних функцій Q_i використовуються пари відліків вихідних даних, які з точністю до перестановки всередині кожної пари збігаються з парами відліків вектора X_t, при $\delta = 0$. В результаті після першої ітерації виходить вектор $Z1(\delta)$, що складається з тих же компонентів, що вектор $Z_1(\delta)$. Очевидно, що компоненти вектора Z1(□) зсунуті на величину □ щодо відповідних компонентів вектора $Z_1(0)$.

На кожній і-ій ітерації в утворенні значень вектора $Z_i(\delta)$ беруть участь 2^i компонент вихідного вектора X_t. Очевидно, що ці компоненти можуть бути різними за $\delta = 0$ і за $\delta \neq 0$. Однак після виконання всіх ітерацій усі вихідні компоненти вектора Xt. беруть участь у формуванні компонент вектора Y. Таким чином, при $\delta \neq 0$:

- для першої ітерації

$$Z_1(N-\delta) = Q_1\left(x_{(N-\delta+t)\, mod\, N}, x_{(N/2-\delta+t)\, mod\, N}\right); \qquad (2.28)$$

- для n-ої ітерації

$$Z_n(\delta) = Q_i\left(Z_{n-1}(N-\delta)\, mod \quad N, \quad Z(N/2-\delta)\, mod \quad N\right). \qquad (2.29)$$

Аналізуючи вирази (2.45) і (2.46) та граф обчислення вектора Y на рис.2.4 неважко зробити висновок, що інваріантність вектора Y до циклічного зсуву відліків вектора X_t може бути досягнуто у разі, коли нелінійна функція забезпечує виконання наступного співвідношення:

$$Q \quad (x, y) = Q \quad (y, x), \qquad (2.30)$$

де x та y – аргументи нелінійної функції.

Як нелінійне перетворення можна вибрати зведення в квадрат значень проміжних векторів, які виходять після кожної ітерації. Однак найбільш простим та швидкодіючим є перетворення наступного виду:

$$Q\ (x,y) = \left|\ x,y\ \right|, \tag{2.31}$$

Враховуючи, що функції Уолша є дворівневими і приймають значення +1 і –1, можна використовувати як нелінійне перетворення операції логічного множення і додавання у разі, якщо відліки вектора вихідних сигналів є також дворівневими.

Для багаторівневих сигналів найбільш простим та швидкодіючим є перетворення наступного виду:

$$Q_0(a,b) = |a+b|,' \quad Q_1(a,b) = |a-b|. \tag{2.32}$$

Результуюче перетворення можна назвати модифікованим перетворенням у базисі Уолша (МПУ), яке вимагає для своєї реалізації $Nlog_2N$ операцій складання-віднімання та $Nlog_2N$ операцій взяття за модулем. Спектральні коефіцієнти при цьому обчислюються в базисі, ядра якого набувають вигляду:

$$W_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad W_2 = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}, \quad W_3 = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \tag{2.33}$$

Результуюче перетворення, яке спочатку розглядалося як поєднання нелінійної операції взяття по модулю та лінійної операції проміжного перетворення за допомогою факторизованої матриці з однакових ядер, набуває вигляду лінійного перетворення, яке може бути представлене у вигляді добутку факторизованих матриць, складених з різних ядер [17,18] .

Властивість інваріантності до циклічного зсуву МПУ дає можливість однозначного порівняння реалізацій стаціонарного випадкового процесу незалежної змінної і дозволяє охарактеризувати кожен клас випадкових процесів відповідним еталонним спектром, отриманим за допомогою МПУ.

2.4. Основні результати та висновки за розділом

Методи аналізу аналогових та цифрових сигналів у базисі тригонометричних функцій Фур'є мають досить глибоку теорію. Але, незважаючи на наявність способів обчислення дискретних спектрів за допомогою алгоритмів швидкого перетворення Фур'є, отримання спектральних характеристик є трудомістким завданням. Оперативний контроль параметрів ТОУ за енергетичним спектром стає менш ефективним при великих розмірностях цифрових сигналів унаслідок суттєвих витрат часу на обчислення коефіцієнтів Фур'є та їх квадратів.

РОЗДІЛ 3.
ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА ПРАКТИЧНЕ ЗАСТОСУВАННЯ

3.1 Загальні відомості. Функціональне призначення

Для функціонування програми потрібна наявність наступного програмного забезпечення:

- Операційна система Windows 7/10;

Функціональне призначення даного ПС полягає у створенні методи пошуку спектральних ознак текстів, яка дозволить правильно розпізнавати та аналізувати текстові послідовності.

3.2 Опис логічної структури

Програмний продукт включає 25 файлів (рис.2):

- 1 файл проекту (DPR);
- 1 виконавчий файл (EXE);
- 5 файлів форм (DFM);
- 5 файлів модулів (PAS);
- 5 скомпільованих Delphi-модулів (DCU);
- 8 ресурсних, конфігураційних та допоміжних файлів (DDP, CFG, DOF, RES).

Крім того, використовується допоміжна зовнішня програма (Butch-er.EXE © AC SOFT) для реалізації різних алгоритмів шифрування.

Реалізація базового методу спектрального перетворення Фур'є має вигляд:

```
procedure Furie(X:Tvi; var Y:Tvi);
var
  i,j:integer;
begin
for i:=0 to N-1 do
 for j:=0 to N-1 do
```

```
   Asin[i,j]:=sin(2*pi*i*j/N);

for i:=0 to N-1 do
  for j:=0 to N-1 do
    Acos[i,j]:=cos(2*pi*i*j/N);

for i:=1 to N-1 do begin
  Ysin[i]:=0;
  Ycos[i]:=0;
                end;

for i:=1 to N-1 do
  for j:=1 to N-1 do begin
     Ysin[i]:=Ysin[i]+Asin[i,j]*X[j];
     Ycos[i]:=Ycos[i]+Acos[i,j]*X[j];
                  end;
for i:=1 to N-1 do
  Y[i]:=round(sqrt(sqr(Ysin[i])+sqr(Ycos[i])));
       end;
```

3.3. Таблиці ідентифікаторів підпрограм та компонентів

Таблиця 3.1

Таблиця ідентифікаторів

Назва	Модуль	Опис
N2Click	archformu	Обробник події вибору опції меню
N3Click	archformu	Обробник події вибору опції меню
Button1Click	archformu	Обробник події натискання на кнопку
Button2Click	archformu	Обробник події натискання на кнопку
Button3Click	archformu	Обробник події натискання на кнопку
N5Click	archformu	Обробник події вибору опції меню
Button4Click	archformu	Обробник події натискання на кнопку
N6Click	archformu	Обробник події вибору опції меню
RadioButton1Click	archformu	Обробник події вибору селективного компонента
RadioButton2Click	archformu	Обробник події вибору селективного компонента
opget	archformu	Функція для виконання архівації
init	archformu	Функція для виконання архівації
putc	archformu	Функція для виконання архівації
clput	archformu	Функція для виконання архівації
out	archformu	Функція для виконання архівації
sift	archformu	Функція для виконання архівації
obh	archformu	Функція для виконання архівації
build	archformu	Функція для виконання архівації
getc	archformu	Функція для виконання архівації
opget1	archformu	Функція для виконання розархівації
init1	archformu	Функція для виконання розархівації
clput1	archformu	Функція для виконання розархівації
sift1	archformu	Функція для виконання розархівації
build1	archformu	Функція для виконання розархівації
getc1	archformu	Функція для виконання розархівації
putc1	archformu	Функція для виконання розархівації
getb1	archformu	Функція для виконання розархівації
XOR1Click	codeformu	Обробник події вибору опції меню
N3Click	codeformu	Обробник події вибору опції меню
N4Click	codeformu	Обробник події вибору опції меню
FormShow	codeformu	Оброблювач події появи форми
N5Click	codeformu	Обробник події вибору опції меню
Memo1Change	codeformu	Обробник події зміни вмісту текстового компонента
Button1Click	codeformu	Обробник події натискання на кнопку

XOR2Click	codeformu	Обробник події вибору опції меню
XORCoding	codeformu	Функція для виконання кодування
N2Click	mainu	Обробник події вибору опції меню
N3Click	mainu	Обробник події вибору опції меню
N4Click	mainu	Обробник події вибору опції меню
N6Click	mainu	Обробник події вибору опції меню
N7Click	mainu	Обробник події вибору опції меню
N10Click	mainu	Обробник події вибору опції меню
N11Click	mainu	Обробник події вибору опції меню
N12Click	mainu	Обробник події вибору опції меню
Button1Click	mainu	Обробник події натискання на кнопку
Button2Click	mainu	Обробник події натискання на кнопку
ReadStrFromTextfile	mainu	Функція зчитування рядка заданої довжини текстового файлу
Furie	mainu	Функція спектрального перетворення
Stat	mainu	Функція аналізу перетвореного масиву дискретних відліків
Result	mainu	Функція виведення результатів аналізу

Таблиця 3.2

Таблиця компонентів, використаних у програмі

Назва	Класс-батько	Форма	Призначення
Label1	TLabel	aboutform	Візуалізація текстової інформації
Label2	TLabel	aboutform	Візуалізація текстової інформації
Label3	TLabel	aboutform	Візуалізація текстової інформації
Label4	TLabel	aboutform	Візуалізація текстової інформації
Label5	TLabel	aboutform	Візуалізація текстової інформації
Label6	TLabel	aboutform	Візуалізація текстової інформації
Label7	TLabel	aboutform	Візуалізація текстової інформації
Image1	TImage	aboutform	Візуалізація графічної інформації
MainMenu1	TMainMenu	archform	Основне меню
N1	TMenuItem	archform	Підміню
N2	TMenuItem	archform	Підміню
N3	TMenuItem	archform	Підміню
N4	TMenuItem	archform	Підміню
N5	TMenuItem	archform	Підміню
N5	TMenuItem	archform	Підміню
Edit1	TEdit	archform	Компонент уведення інформації
Edit2	TEdit	archform	Компонент уведення інформації
Label1	TLabel	archform	Візуалізація текстової інформації

Label2	TLabel	archform	Візуалізація текстової інформації
Button1	TButton	archform	Компонент активації дії
Button2	TButton	archform	Компонент активації дії
Button3	TButton	archform	Компонент активації дії
Button4	TButton	archform	Компонент активації дії
OpenDialog1	TOpenDialog	archform	Діалог відкриття файлу
OpenDialog2	TOpenDialog	archform	Діалог відкриття файлу
SaveDialog1	TSaveDialog	archform	Діалог збереження файлу
SaveDialog2	TSaveDialog	archform	Діалог збереження файлу
RadioButton1	TRadioButton	archform	Компонент вибору
RadioButton2	TRadioButton	archform	Компонент вибору
Image1	TImage	authorform	Візуалізація графічної інформації
MainMenu1	TMainMenu	codeform	Основне меню
N1	TMenuItem	codeform	Підміню
N2	TMenuItem	codeform	Підміню
N3	TMenuItem	codeform	Підміню
N4	TMenuItem	codeform	Підміню
N5	TMenuItem	codeform	Підміню
N5	TMenuItem	codeform	Підміню
XOR1	TMenuItem	codeform	Підміню
Label1	TLabel	codeform	Візуалізація текстової інформації
Label2	TLabel	codeform	Візуалізація текстової інформації
Label3	TLabel	codeform	Візуалізація текстової інформації
Label4	TLabel	codeform	Візуалізація текстової інформації
Label5	TLabel	codeform	Візуалізація текстової інформації
OpenDialog1	TOpenDialog	codeform	Діалог відкриття файлу
Memo1	TMemo	codeform	Компонент відображення тексту
Memo2	TMemo	codeform	Компонент відображення тексту
Button1	TButton	codeform	Компонент активації дії
Edit1	TEdit	codeform	Компонент уведення інформації
MainMenu1	TMainMenu	mainform	Основне меню
N1	TMenuItem	mainform	Підміню
N2	TMenuItem	mainform	Підміню
N3	TMenuItem	mainform codeform	Підміню

N4	TMenuItem	mainform codeform	Підміню
N5	TMenuItem	mainform codeform	Підміню
N6	TMenuItem	mainform codeform	Підміню
N7	TMenuItem	mainform	Підміню
N8	TMenuItem	mainform codeform	Підміню
N9	TMenuItem	mainform codeform	Підміню
N10	TMenuItem	mainform codeform	Підміню
N11	TMenuItem	mainform codeform	Підміню
N12	TMenuItem	mainform codeform	Підміню
Label1	TLabel	mainform	Візуалізація текстової інформації
Label2	TLabel	mainform	Візуалізація текстової інформації
Label3	TLabel	mainform	Візуалізація текстової інформації
Label4	TLabel	mainform	Візуалізація текстової інформації
Label5	TLabel	mainform	Візуалізація текстової інформації
Label6	TLabel	mainform	Візуалізація текстової інформації
Label7	TLabel	mainform	Візуалізація текстової інформації
Label8	TLabel	mainform	Візуалізація текстової інформації
Label9	TLabel	mainform	Візуалізація текстової інформації
Label10	TLabel	mainform	Візуалізація текстової інформації
Label11	TLabel	mainform	Візуалізація текстової інформації
Label12	TLabel	mainform	Візуалізація текстової інформації
Label13	TLabel	mainform	Візуалізація текстової інформації
Label14	TLabel	mainform	Візуалізація текстової інформації
Label15	TLabel	mainform	Візуалізація текстової інформації
Label16	TLabel	mainform	Візуалізація текстової інформації
Label17	TLabel	mainform	Візуалізація текстової інформації
Label18	TLabel	mainform	Візуалізація текстової інформації
Button1	TButton	mainform	Компонент активації дії
Button2	TButton	mainform	Компонент активації дії
OpenDialog1	TOpenDialog	mainform	Діалог відкриття файлу
Chart1	TChart	mainform	Візуалізація графічної інформації
Chart2	TChart	mainform	Візуалізація графічної інформації
Series1	TAreaSeries	mainform	Візуалізація графічної інформації
Series2	TAreaSeries	mainform	Візуалізація графічної інформації

3.4 Виклик та завантаження

Виклик програми здійснюється шляхом запуску виконуваного файлу mainp.exe. Для подальшої роботи з програмою необхідно завантажити вихідний текст.

3.5 Вхідні дані

Вхідними є відкриті, архівні або шифровані текстові послідовності, представлені у вигляді файлів у текстовому форматі (TXT). Для зручності реалізовано метод динамічної зміни вихідного тексту в інтерактивному режимі у процесі роботи з програмою.

3.6 Вихідні дані

Вихідними даними є:

- графічне відображення спектрального аналізу вихідної ТП як графічних файлів у форматі BMP;
- математична форма результатів аналізу (виведення на екран деяких спектральних ознак вихідної та проаналізованої ТП).

3.7 Опис інтерфейсу користувача

Розроблене ПС є SDI (Single Document Interface) Windows додаток.

При запуску програми користувач бачить вікно, наведене на рис. 3.1

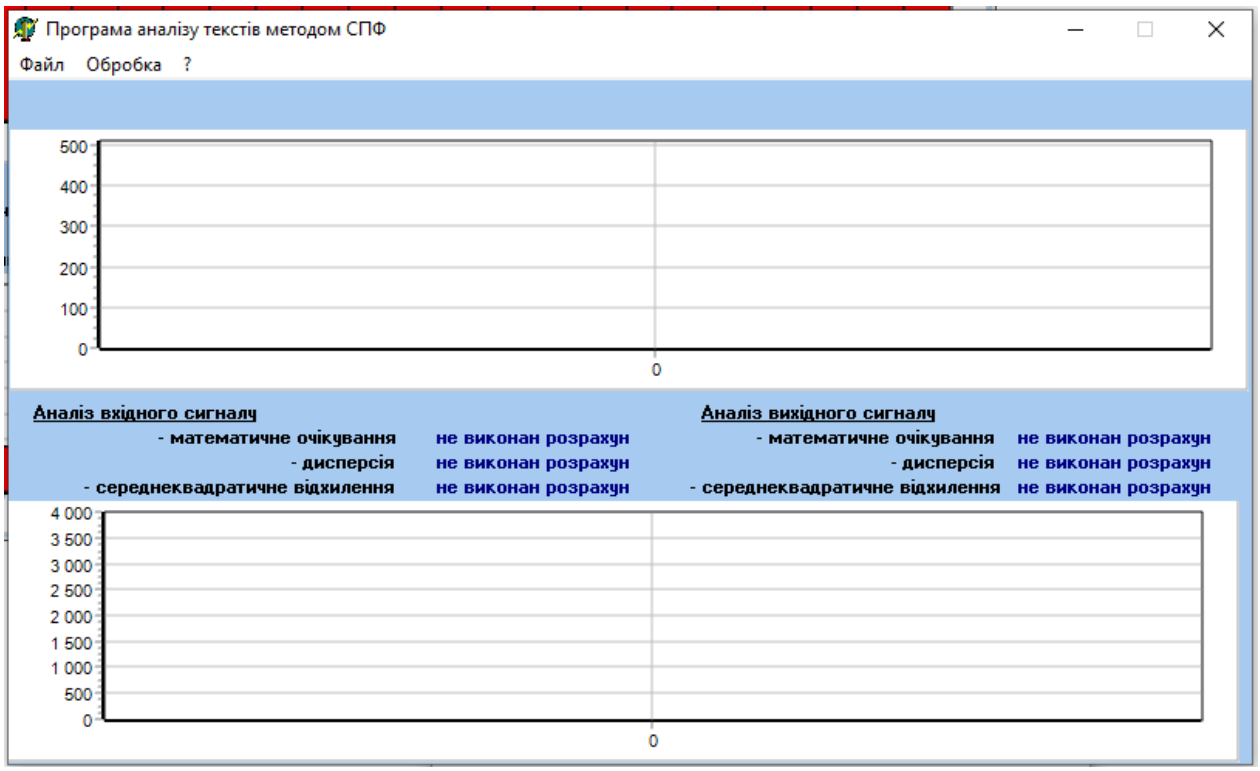

Рис. 3.1 Головне вікно програми

Для початку роботи з програмою необхідно завантажити вихідний текстовий файл за допомогою пункту меню "Файл\Відкрити файл". З'явиться вікно (рис.4), у якому необхідно вибрати досліджуваний текстовий файл і натиснути кнопку діалогу «Відкрити».

Після цього програма автоматично проведе простий статистичний аналіз вихідної послідовності та виведе результати на основну форму (рис.5).

Далі для проведення спектрального перетворення вихідної текстової послідовності необхідно вибрати пункт меню «Обробка \ Аналіз текстового файлу» (рис. 3.2).

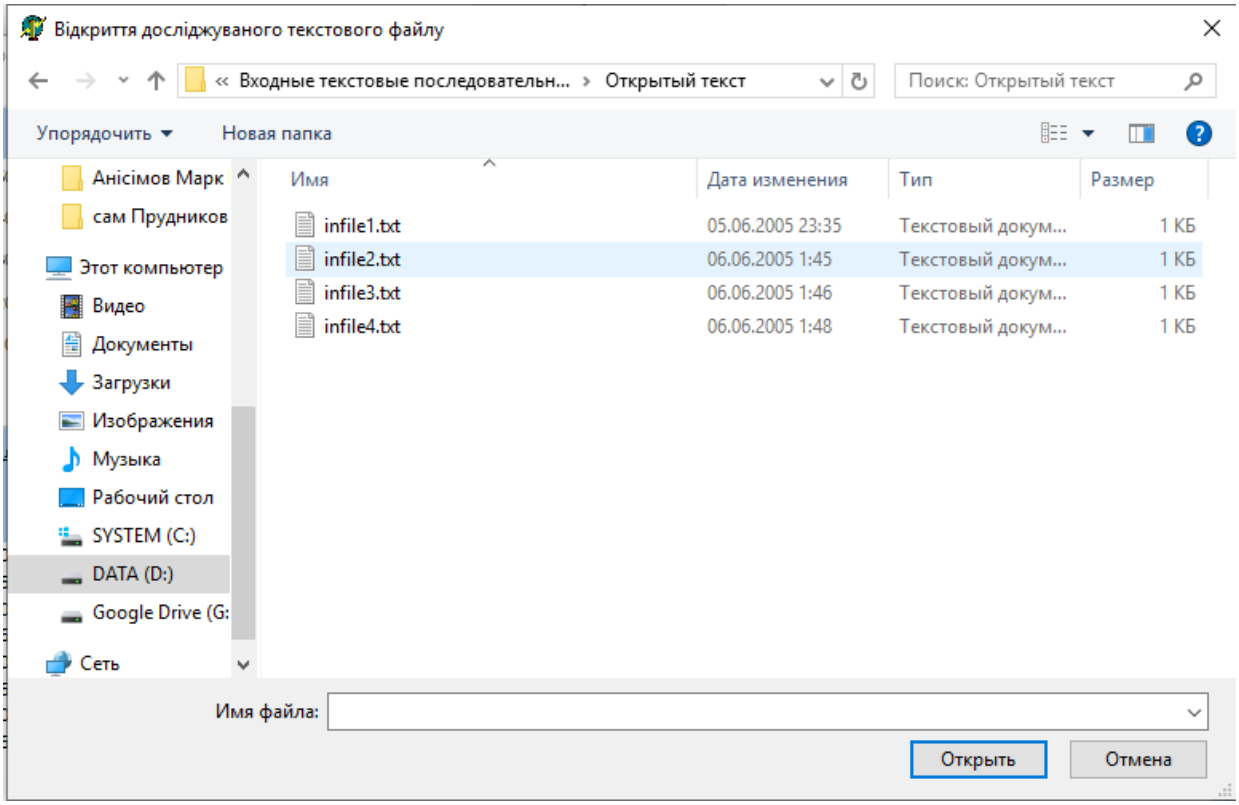

Рис.3.2 Діалог відкриття вихідного текстового файлу.

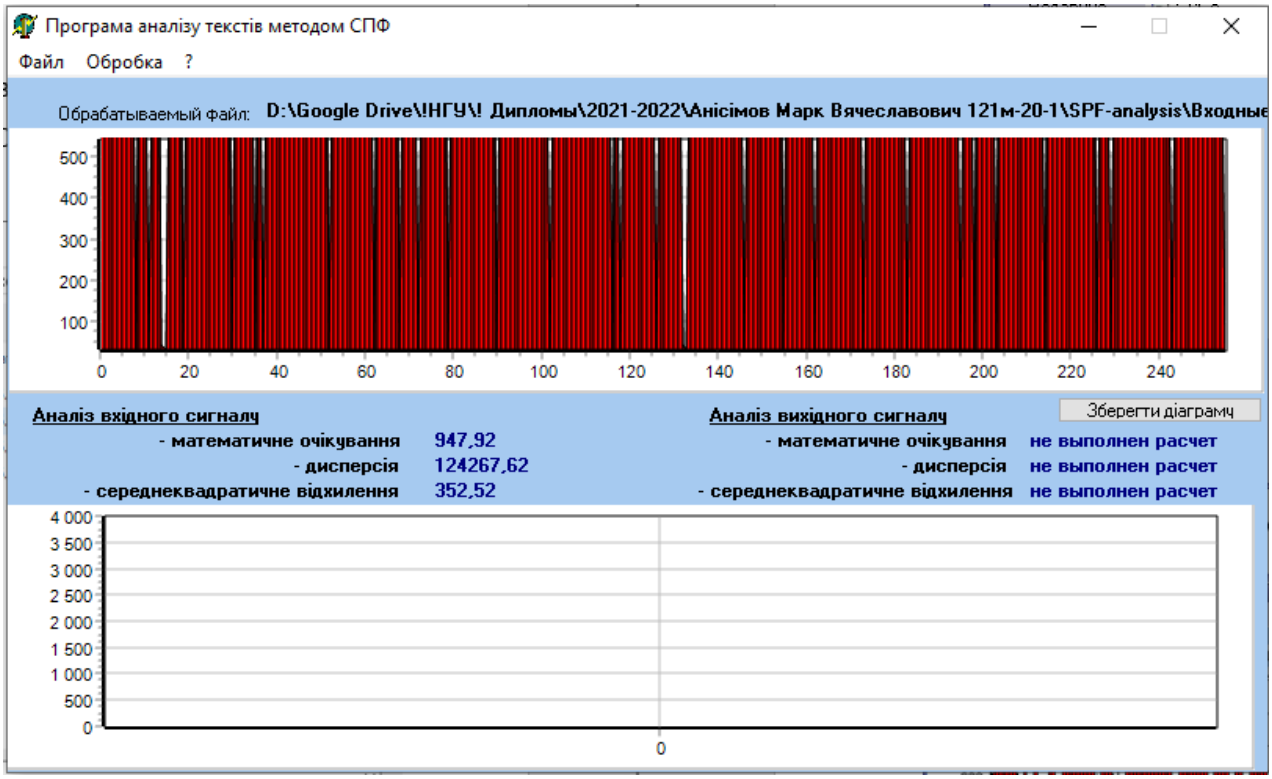

Рис.3.3 Відображення результатів простого статистичного аналізу вихідної текстової послідовності

При цьому на головній формі автоматично будуть відображені результати проведеного спектрального аналізу і показаний кінцевий результат - висновок за типом вихідного тексту (відкритий, архівний шифрований млм) (рис. 3.4).

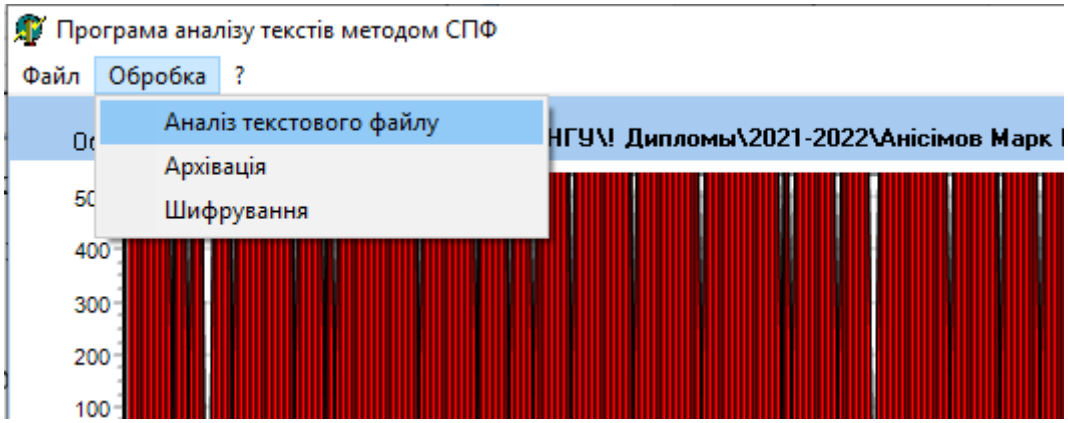

Рис. 3.4 Виконання спектрального аналізу

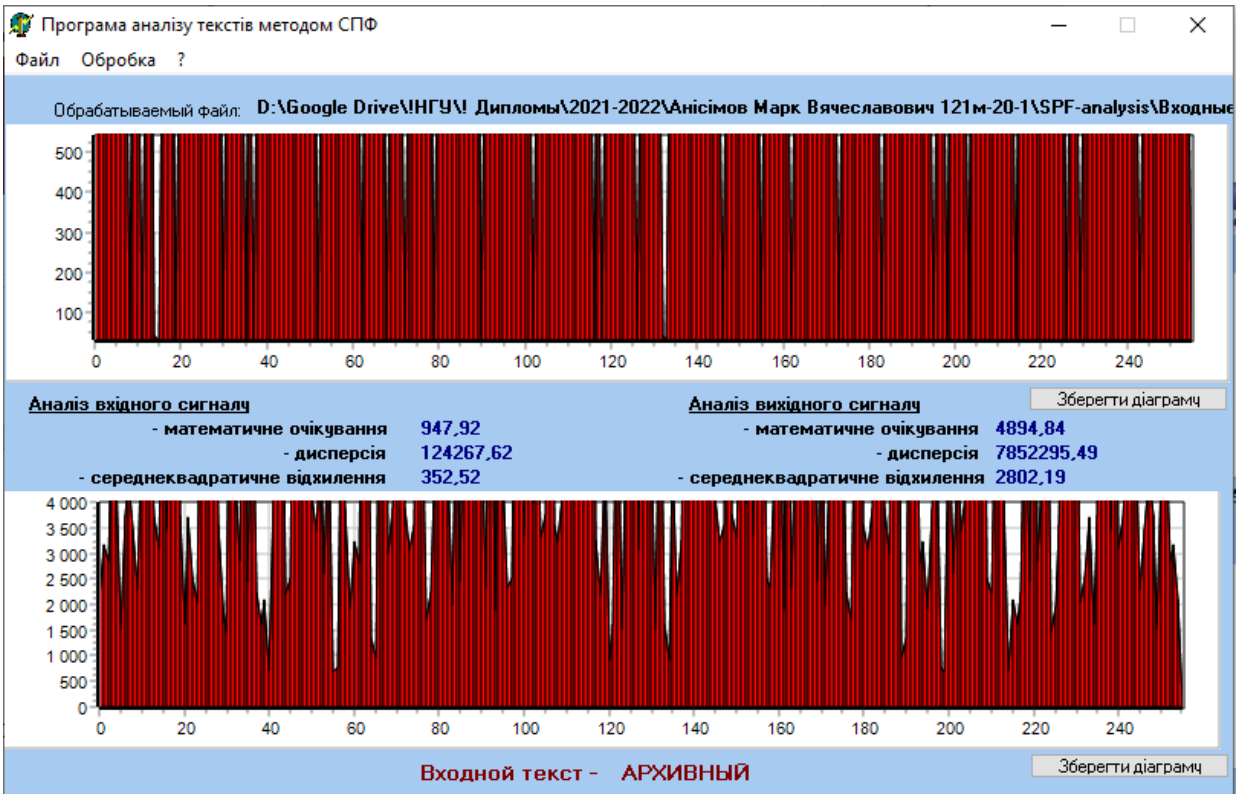

Рис. 3.5 Основна форма з результатами спектрального аналізу.

При необхідності можна зберегти графічне відображення результатів спектрального перетворення, натиснувши на кнопку під відповідною діаграмою (рис. 3.6)

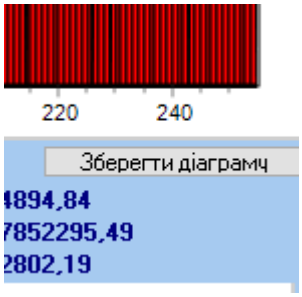

Рис. 3.6 Збереження діаграм результатів аналізу.

Крім основних розглянутих функцій, у програмі реалізовано додаткові:

- архівація за методом Хаффмана («Обробка Архівація»);

- Шифрування з використанням XOR-алгоритму («Обробка \ Шифрування»).

Для виконання стиснення текстового файлу необхідно вибрати відповідний пункт меню і на формі, що з'явилася, вказати файл для архівації та ім'я архіву (рис. 3.7).

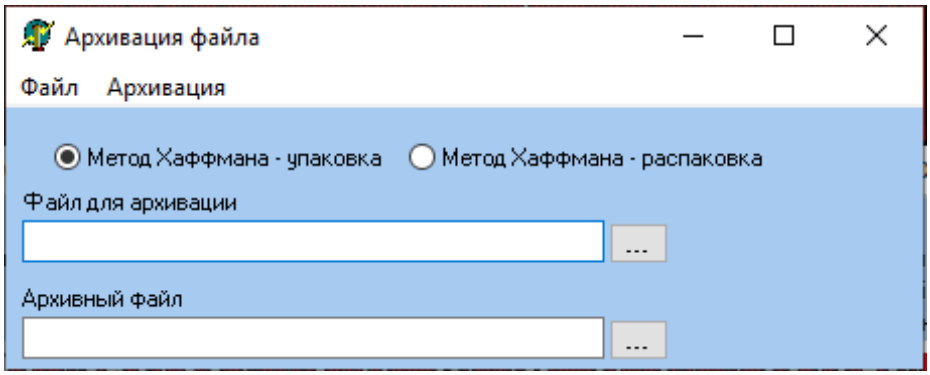

Рис. 3.7 Архівація файлів.

Для виконання шифрування текстового файлу за допомогою XOR-алгоритму можна скористатися відповідним пунктом меню. На формі, що з'явилася, користувачеві пропонується або ввести шифрований текст вручну, або завантажити його з файлу. Після завантаження в меню необхідно ввести на

формі ключ шифрування та вибрати потрібний пункт (шифрування або розшифрування). У нижній частині форми (рис. 3.8) відобразиться зашифрований (розшифрований) текст. Крім того, результуючий текст зберігається в поточну директорію під ім'ям XORCoding.TXT.

Для реалізації інших алгоритмів шифрування було використано зовнішню програму Butcher.EXE фірми AC Soft ®. Його вид і модуль шифрування представлені на рис. 3.8.

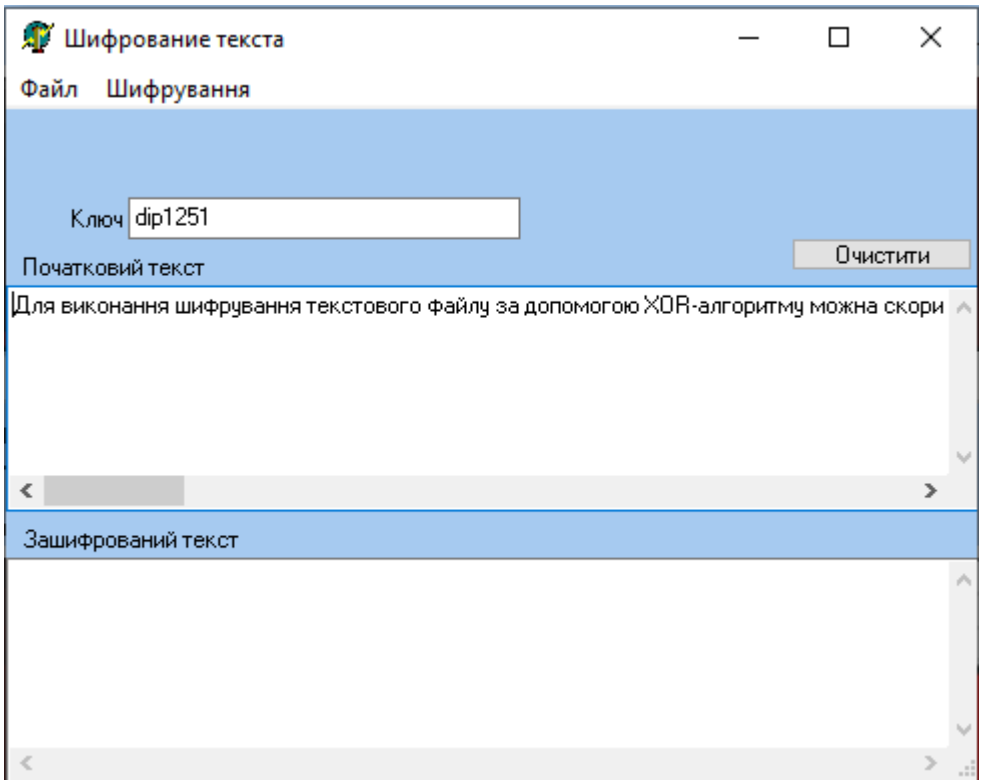

Рис. 3.8 Шифрування файлів за XOR-алгоритмом.

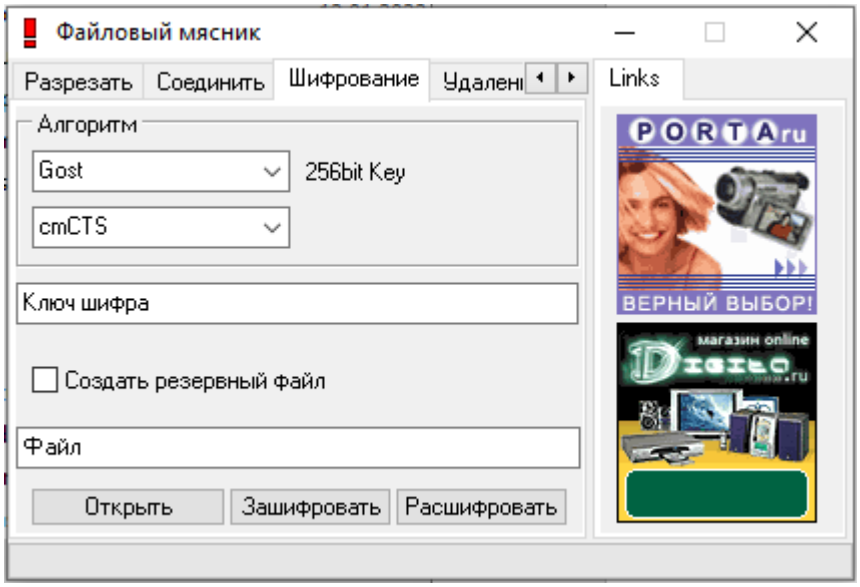

Рис. 3.8 Зовнішня програма для шифрування файлів

3.8 Аналіз результатів дослідження

У процесі досліджень було проаналізовано велику кількість смислових текстів різних типів: відкритих, архівних (архіватор Winrar, алгоритм архівації Хаффмана) та шифрованих (алгоритм шифрування DES single 8 byte, DES triple 8 byte, XOR-алгоритм, RC5 RSA-алгоритм).

По кожному з алгоритмів були проведені статистичні підрахунки (математичне очікування, дисперсія, середньоквадратичне відхилення) щодо кожного з перетворених спектрів вихідних текстів.

Отримані результати наведено у табл.3-9

Таблиця 3.3

Аналіз відкритого тексту

Тип тексту	Маточікування (MX)	Дисперсія (DX)	Середньоквадратичне відхилення (SqDevX)
російська+англійська	200 (1048)	5675 (354301)	75 (595)
англійська	91 (435)	911 (44102)	30 (210)
російська	209 (948)	4720 (307844)	69 (555)
українська+англійська	190 (1156)	6439 (312698)	80 (559)

У дужках вказано значення перетвореної текстової послідовності, а перше значення – результати аналізу вихідної послідовності.

Таблиця 3.4

Аналіз архівного тексту (метод Хаффмана).

Тип тексту	Маточікування (MX)	Дисперсія (DX)	Середньоквадратичне відхилення (SqDevX)
російська+англійська	148 (1330)	8852 (498350)	94 (706)
англійська	133 (1213)	7200 (371108)	85 (609)
російська	152 (1319)	8671 (480247)	93 (693)
українська+англійська	141 (1238)	7636 (421385)	88 (649)

Таблиця 3.5

Аналіз архівного тексту (архів тор WinRar).

Тип тексту	Маточікування (MX)	Дисперсія (DX)	Середньоквадратичне відхилення (SqDevX)
російська+англійська	116 (1034)	5809 (418825)	76 (647)
англійська	113 (1089)	6261 (417075)	79 (646)
російська	112 (1077)	6028 (384002)	78 (620)
українська+англійська	112 (1133)	6482 (375915)	81 (613)

Таблиця 3.6.

Аналіз шифрованого тексту (XOR-алгоритм).

Тип тексту	Маточікування (MX)	Дисперсія (DX)	Середньоквадратичне відхилення (SqDevX)
російська+англійська	159 (834)	3766 (269286)	61 (519)
англійська	162 (820)	3402 (287356)	60 (618)
російська	165 (766)	3307 (260413)	58 (510)
українська+англійська	161 (840)	3982 (301254)	64 (556)

Таблиця 3.7

Аналіз шифрованого тексту (DES single 8 byte).

Тип тексту	Маточікування (MX)	Дисперсія (DX)	Середньоквадратичне відхилення (SqDevX)
російська+англійська	123 (1046)	5311 (264618)	73 (514)
англійська	135 (965)	5114 (377146)	72 (614)
російська	124 (1053)	5708 (351592)	76 (593)
українська+англійська	126 (1073)	5738 (317360)	76 (563)

Таблиця 3.8

Аналіз зашифрованого тексту (DES triple 8 byte).

Тип тексту	Маточікування (MX)	Дисперсія (DX)	Середньоквадратичне відхилення (SqDevX)
російська+англійська	120 (1007)	5255 (330262)	73 (575)
англійська	130 (1068)	5704 (318986)	76 (565)
російська	131 (1068)	5703 (319176)	76 (565)
українська+англійська	121 (1061)	5499 (282763)	74 (532)

Таблиця 3.9

Аналіз зашифрованого тексту (RC5 RSA).

Тип тексту	Маточікування (MX)	Дисперсія (DX)	Середньоквадратичне відхилення (SqDevX)
російська+англійська	129 (972)	4854 (297265)	70 (545)
англійська	127 (985)	4908 (286064)	70 (535)
російська	131 (1047)	5545 (323536)	75 (569)
українська+англійська	130 (1064)	5687 (323530)	75 (569)

Підбивши підсумки по наведених таблицях, можна з упевненістю сказати, що кожному типу вихідного смислового тексту відповідає певне значення статистичних оцінок, причому воно коливається в незначних межах для типу тексту (відкритий, шифрований або архівний) і мови написання.

У проведених дослідженнях були використані такі поєднання мов:

- російська;
- англійська;
- російська + англійська;
- український;

Середньостатистичні значення проведених досліджень наведено в табл. 3.10.

Таблиця 3.10

Середньостатистичні значення за вихідними та перетвореними текстовими послідовностями (для кожного алгоритму).

Вихідна ТП	Перетворена ТП	Вихідна ТП	Перетворена ТП	Вихідна ТП	Перетворена ТП
200,00	1048,00	5675,00	354301,00	75,00	595,00
91,00	435,00	911,00	44102,00	30,00	210,00
209,00	948,00	4720,00	307844,00	69,00	555,00
190,00	1156,00	6439,00	312698,00	80,00	559,00
172,50	**896,75**	**4436,25**	**254736,25**	**63,50**	**479,75**
148,00	1330,00	8852,00	498350,00	94,00	706,00
133,00	1213,00	7200,00	371108,00	85,00	609,00
152,00	1319,00	8671,00	480247,00	93,00	693,00
141,00	1238,00	7636,00	421385,00	88,00	649,00
143,50	**1275,00**	**8089,75**	**442772,50**	**90,00**	**664,25**
116,00	1034,00	5809,00	418825,00	76,00	647,00
113,00	1089,00	6261,00	417075,00	79,00	646,00
112,00	1077,00	6028,00	384002,00	78,00	620,00
112,00	1133,00	6482,00	375915,00	81,00	613,00
113,25	**1083,25**	**6145,00**	**398954,25**	**78,50**	**631,50**
159,00	834,00	3766,00	269286,00	61,00	519,00
162,00	820,00	3402,00	287356,00	60,00	618,00
165,00	766,00	3307,00	260413,00	58,00	510,00
161,00	840,00	3982,00	301254,00	64,00	556,00
161,75	**815,00**	**3614,25**	**279577,25**	**60,75**	**550,75**
123,00	1046,00	5311,00	264618,00	73,00	514,00
135,00	965,00	5114,00	377146,00	72,00	614,00
124,00	1053,00	5708,00	351592,00	76,00	593,00
126,00	1073,00	5738,00	317360,00	76,00	563,00
127,00	**1034,25**	**5467,75**	**327679,00**	**74,25**	**571,00**
120,00	1007,00	5255,00	330262,00	73,00	575,00
130,00	1068,00	5704,00	318986,00	76,00	565,00
131,00	1068,00	5703,00	319176,00	76,00	565,00
121,00	1061,00	5499,00	282763,00	74,00	532,00
125,50	**1051,00**	**5540,25**	**312796,75**	**74,75**	**559,25**
129,00	972,00	5854,00	297265,00	70,00	545,00
127,00	985,00	4908,00	286064,00	70,00	535,00
131,00	1047,00	5545,00	323536,00	75,00	569,00
130,00	1064,00	5687,00	323530,00	75,00	569,00
129,25	**1017,00**	**5498,50**	**307598,75**	**72,50**	**554,50**

Жирним шрифтом вказані середні значення за чотирма попередніми значеннями – для кожної мови та алгоритму стиснення чи шифрування.

РОЗДІЛ 4.
ЕКОНОМІЧНА ЧАСТИНА

4.1. Визначення трудомісткості та вартості розробки програмного продукту

Початкові дані:

1. передбачуване число операторів програми – 12000;
2. коефіцієнт складності програми – 1,7;
3. коефіцієнт корекції програми в ході її розробки – 0,07;
4. годинна заробітна плата програміста– 90 грн/год;
5. коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі – 1,2;
6. коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності – 1,2;
7. вартість машино-години ЕОМ –15 грн/год.

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{отл} + t_д \text{, людино-годин,} \quad (4.1)$$

де t_o- витрати праці на підготовку й опис поставленої задачі (приймається 50 людино-годин);

t_u - витрати праці на дослідження алгоритму рішення задачі;

t_a- витрати праці на розробку блок-схеми алгоритму;

t_n-витрати праці на програмування по готовій блок-схемі;

$t_{отл}$-витрати праці на налагодження програми на ЕОМ;

$t_д$ - витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у програмногму забезпеченні, яке розробляється.

Умовне число операторів:

$$Q = q \cdot C \cdot (1 + p), \quad (4.2)$$

де q - передбачуване число операторів (12000);

C - коефіцієнт складності програми (1,7);

p - коефіцієнт корекції програми в ході її розробки (0,07).

Звідси умовне число операторів в програмі:

$$Q = 1{,}7 \cdot 12000 \cdot (1 + 0{,}07) = 22000$$

Витрати праці на вивчення опису задачі t_u визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q \cdot B}{(75..85) \cdot k}, \text{ людино-годин, } (4.3)$$

де B - коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі;

k - коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності. При стажі роботи від 3 до 5 років він складає 1,2.

Приймемо збільшення витрат праці внаслідок недостатнього опису завдання не більше 50% (B = 1,2). З урахуванням коефіцієнта кваліфікації k = 1,2, отримуємо витрати праці на вивчення опису завдання:

$$t_u = \frac{(22000 \cdot 1{,}2)}{(75 \cdot 1{,}2)} = 294 \text{ людино-годин}$$

Витрати праці на розробку алгоритму рішення задачі визначаються за формулою:

$$t_a = \frac{Q}{(20\ldots25)\cdot k}\ \text{, людино-годин,}\quad (4.4)$$

де Q – умовне число операторів програми;

k – коефіцієнт кваліфікації програміста.

Підставивши відповідні значення в формулу (4.4), людино-годин:

$$t_a = \frac{22000}{(20\cdot1{,}2)} = 917 \text{ людино-годин.}$$

Витрати на складання програми по готовій схемі:

$$t_n = \frac{Q}{(20\ldots25)\cdot k}\ \text{, людино-годин,}\quad (4.5)$$

$$t_n = \frac{(22000\cdot1{,}2)}{(20\cdot1{,}2)} = 1100 \text{ людино-годин.}$$

Витрати праці на налагодження програми на ЕОМ:

- за умови автономного налагодження одного завдання:

$$t_{отл} = \frac{Q}{(4..5)\cdot k}\ \text{, людино-годин,}\quad (4.6)$$

$$t_{отл} = \frac{22000}{(5\cdot1{,}2)} = 3665 \text{ людино-годин.}$$

- за умови комплексного налагодження завдання:

$$t^{k}_{отл} = 1{,}5 \cdot t_{отл}$$, людино-годин, (4.7)

$t^{k}_{отл}$= 1,5 · 3665 = 5497 людино-годин.

Витрати праці на підготовку документації визначаються за формулою:

$$t_{д} = t_{др} + t_{до}$$, людино-годин, (4.8)

де $t_{др}$-трудомісткість підготовки матеріалів і рукопису:

$$t_{др} = \frac{Q}{(15..20) \cdot k}$$, людино-годин, (4.9)

$t_{до}$ - трудомісткість редагування, печатки й оформлення документації:

$$t_{до} = 0{,}75 \cdot t_{др}$$, людино-годин, (4.10)

Підставляючи відповідні значення, отримаємо:

$$t_{др} = \frac{22000}{(18 \cdot 1{,}2)} = 1018 \text{ людино-годин.}$$

$t_{до}$ = 0,75 · 1018 = 763 людино-годин.

$t_{д}$= 1018 + 763 = 1781 людино-годин.

Повертаючись до формули (3.1), отримаємо повну оцінку трудомісткості розробки програмного забезпечення:

t = 50 + 294+ 917 + 1100 + 3665 + 1781 = 7807 людино-годин.

4.2. Витрати на створення програмного забезпечення

Витрати на створення ПЗ $К_{ПО}$ включають витрати на заробітну плату виконавця програми $З_{ЗП}$ і витрат машинного часу, необхідного на налагодження програми на ЕОМ:

$$К_{ПО} = З_{ЗП} + З_{МВ}, \text{ грн. (4.11)}$$

Заробітна плата виконавців визначається за формулою:

$$З_{ЗП} = t \cdot С_{ПР}, \text{ грн, (4.12)}$$

де: t - загальна трудомісткість, людино-годин;

$С_{ПР}$ - середня годинна заробітна плата програміста, грн/година

З урахуванням того, що середня годинна зарплата програміста становить 90 грн / год, отримуємо:

$$З_{ЗП} = 7807 \cdot 90 = 702{,}630 \text{ грн.}$$

Вартість машинного часу, необхідного для налагодження програми на ЕОМ, визначається за формулою:

$$З_{мв} = t_{отл} \cdot С_{мч}, \text{ грн, (4.13)}$$

де $t_{отл}$ - трудомісткість налагодження програми на ЕОМ, год;

$С_{мч}$ - вартість машино-години ЕОМ, грн/год (15 грн/год).

Підставивши в формулу (4.13) відповідні значення, визначимо вартість необхідного для налагодження машинного часу:

$$З_{мв} = 3665 \cdot 15 = 54{,}975 \text{ грн.}$$

Звідси витрати на створення програмного продукту:

$$K_{ПО} = 702{,}630 + 54{,}975 = 757{,}605 \text{ грн.}$$

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k \cdot F_p}, \text{ міс.}$$

де B_k- число виконавців (дорівнює 1);

F_p - місячний фонд робочого часу (при 40 годинному робочому тижні F_p=176 годин).

Звідси витрати на створення програмного продукту:

$$T = \frac{7807}{(1 \cdot 176)} \approx 9 \text{ місяців.}$$

4.3. Маркетингові дослідження ринку збуту розробленого програмного продукту

У сучасному світі розвиток інновацій у секторі інформаційних технологій щодня набирає обертів та розширює свої потреби у вигляді програмних продуктів. Сфера застосування - мобільний зв'язок, інтернет канали, будь які канали обміну інформації, на даний час це є галузі, що стрімко розвиваються, і потребують постійних іновацій

Використання даного програмного продукту адаптоване під будь-який тип користувача, навіть найменш досвідченого, що значно розширює не тільки ринок збуту продукту, а також і споживацький ринок. Це досягнуто завдяки

впровадженню простого та зрозумілого у користуванні інтуїтивного інтерфейсу користувача.

Розроблене програмне забезпечення відноситься до складної програмної продукції, яка потребує спеціального налагодження та використовування прямого маркетингу між виробником і споживачем. Продукт розроблений за замовленням споживача, що передбачає можливість у разі потреби його доробки та обслуговування, випуску нових версій.

4.4. Оцінка економічної ефективності впровадження розробленого програмного забезпечення

Основною перевагою у впровадженні розробленого програмного продукту з точки зору економічної ефективності в тому, що запропонована методика спрощує процес аналізу архівних і шифрованих текстів у порівнянні з існуючими сучасними системами, і як слідство знижує рівень споживання електричної енергії прискорює час для обробки і передачи даних.

Цінність результатів роботи полягає в створенні автоматизованої системи, що функціонує на основі розробленої методики аналізу відкритих, архівних і шифрованих текстів з використанням спектрального перетворення Фур'є, що істотно спрощує процес аналізу у порівнянні з сучасними системами.

Економічна оцінка ефективності пропонованого впровадження оцінена за системою показників, які використовуються у міжнародній і вітчизняній практиці

При оцінці економічної ефективності використані наступні показники:

• чиста поточна вартість (NPU);

• строк окупності капітальних вкладень;

• індекс прибутковості;

• коефіцієнт ефективності інвестицій.

При ухваленні рішення стосовно доцільності впровадження проекту необхідно враховувати значення всіх показників, тому що кожен показник несе

свій обсяг інформації, і тільки всі вони в сукупності можуть дати повне уявлення про реальну ефективність.

Числовий розрахунок вищезгаданих показників наведений у таблиці 4.4.

Таблиця 4.4.

Розрахунок чистих грошових надходжень від розробки ПЗ

Показники, грн	За роками						Усього за 5 років	Середнє за 5 років
	0	1	2	3	4	5		
1. Інвестиції на ПЗ	5500	-	-	-	-	-	5500	1100
2. Витрати до впровадження ПЗ	-	10700	5700	5700	10700	5700	38500	7700
- на придбання обладнання	-	5000	-	-	5000	-	10000	2000
- на щорічну перевірку на погрішність	-	1000	1000	1000	1000	1000	5000	1000
- на оплату праці бригади КВП	-	3000	3000	3000	3000	3000	12000	3000
- на щорічну перевірку держстандарту	-	500	500	500	500	500	2500	500
- на електроенергію	-	1200	1200	1200	1200	1200	6000	1200
3. Витрати після впровадження ПО	-	16240	1240	1240	1240	1240	21200	4240
- на придбання комплексу сенсорів	-	15000	-	-	-	-	15000	3000
- на щорічну перевірку	-	200	200	200	200	200	1000	200
- на оплату праці провайдера	-	800	800	800	800	800	4000	800
- на електроенергію	-	240	240	240	240	240	1200	240
4. Економія	-	-5540	4460	4460	9460	4460	17300	3460
5. Амортизація	-	1980	1320	-	-	-	3300	660
6. Чисті грошові надходження	-	-3560	3140	4460	9460	4460	19280	3856
7. Коефіцієнт дисконтування	-	0,909	0,826	0,751	0,683	0,621	-	-
Дисконтові грошові надходження	-	-3236	2594	3350	6622	2809	12139	2428

Коефіцієнти економічної ефективності

Чиста поточна вартість доходів:

NPU = 12139 - 3300 = 8839 > 0

Строк окупності:

T=3300/2428 = 1,3 років

Індекс прибутковості:

ИД=12139/3300 = 3,68

Показник економічної ефективності (NPU - чиста поточна вартість доходів за роки реалізації впровадження (3-5 років) складе 8839 грн тобто відповідає умовам ефективності, тому що NPU > 0.

Середній строк окупності капіталовкладень складе 1,3 року.

Індекс прибутковості за 5 років складе 3,63, тобто ИД > 1, що означає, проект варто прийняти до розробки.

Таким чином, показник ефективності свідчить про те, що дане впровадження є економічно вигідним.

Висновок: Основною перевагою у впровадженні розробленого програмного продукту з точки зору економічної ефективності в тому, що запропонована методика спрощує процес аналізу архівних і шифрованих текстів у порівнянні з існуючими сучасними системами, і як слідство знижує рівень споживання електричної енергії прискорює час для обробки і передачи даних.

Цінність результатів роботи полягає в створенні автоматизованої системи, що функціонує на основі розробленої методики аналізу відкритих, архівних і шифрованих текстів з використанням спектрального перетворення Фур'є, що істотно спрощує процес аналізу у порівнянні з сучасними системами. Вартість даного програмного забезпечення становить 757,605 грн. з можливими додатковими витратами при розробці проекту при бажанні замовника випустити оновлену версію додатку через деякий час. Очікуваний час розробки - 9 місяців. Цей термін пов'язаний зі значною кількістю операторів і включає в себе час для дослідження та розробки алгоритму розв'язання задачі, розробки дизайну, створення додатків та підготовку документації.

ВИСНОВКИ

У ході дослідження було проведено аналіз існуючих робіт, присвячених аналізу текстів з використанням нейромережевих методів, масочних методів, статистичних методів. Аналіз показав значну перспективність використання перетворень Фур'є при розпізнаванні типів смислових текстових послідовностей.

Було досліджено метод швидкого та спектрального перетворення Фур'є. За результатами експериментальних даних була виведена оптимальна довжина символьного масиву, необхідного для розпізнавання типу тексту розглянутим методом.

На підставі результатів проведених досліджень була розроблена методика аналізу відкритих, архівних і шифрованих текстів із застосування технології спектрального перетворення Фур'є.

Було проаналізовано можливість правильного розпізнавання типу тексту за цим методом. Вона склала близько 90% (при нейромережевому та статистичному методі – 70%, при застосуванні маскового методу – 45%)

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Xia, Feng; Tian, Yu-Chu; Li, Yanjun; Sun, Youxian (2017-10-09). "Wireless Sensor/Actuator Network Design for Mobile Control Applications". Sensors (Basel, Switzerland). 7 (10): 2157–2173. doi:10.3390/s7102157. ISSN 1424-8220. PMC 3864515. PMID 28903220.

2. Дьяконов В.П. MatLab. Обработка сигналов и изображений. Специальный справочник. – СПб.: Питер, 2002. – 608 с.

3. Новиков Л.В., Доклады 1-й международной конференции «Цифровая обработка сигналов и ее применение». Том 2.

4. Alan G. Konheim, Cryptography a primer, John Wiley& Sons, New York, 2011.

5. Н.Ахмед, К.Р.Рао Ортогональные преобразования при обработке цифровых сигналов. М.:, Связь, 2010.

6. Уосермен Ф. Нейрокомпьютерная техника: Теория и практика. М.:, Мир, 1992.

7. Престон. Сравнение аналоговых и цифровых методов распознавания образов. ТИИЭР, т.60, №10, 141-160, 1972.

8. Дж. Гудмен. Введение в Фурье-преобразование. «Мир», М. 2011.

9. Лендэрис и Стенли. Метод дискретизации дифракцидных картин для автоматического распознавания образов. ТИИЭР, т.58, №2, 22-40, 2010.

10. К.М. Богданов, Ю.Г.Козлов, К.А. Яновский. Исследование двумерных спектров объектов и их изображений. Тезисы докладов Второй Всесоюзной школы-семинара «Статистические свойства микроструктур».

11. Дьяконов В.П. Вейвлеты. От теории к практике. – М.: СОЛОН-Р, 2012. – 448 с.

12. Бортник Г. Г. Методи та засоби підвищення ефективності оцінювання фазового дрижання сигналів у телекомунікаційних системах : моногр. / Г. Г. Бортник, М. В. Васильківський, В. М. Кичак. — Вінниця : ВНТУ, 2015. — 140 с. — ISBN 978-966-641-621-9..

13. http://alglib.sources.ru/fft/fourier.php. Спектральное преобразование Фурье и его свойства. Бочканов Сергей, Быстрицкий Владимир.

14. http://mathc.chat.ru/a1/articl01.htm. Ряды и преобразование Фурье. Александр Красногорский.

15. http://www.computer-museum.ru/histussr/dsp.htm. В. М. Сазанов, Н. С. Парфенов. Цифровая обработка сигналов: прошлое и настоящее. Часть 1.

16. Специализированный процессор для выполнения быстрого преобразования Фурье и обработки сигналов СПФ СМ. Рекламные материалы. М.: ИНЭУМ, 1984.

17. Корнеев В. В., Киселев А. В. Современные микропроцессоры. М.: НОЛИДЖ, 2018. 240 с.

18. Цифровые процессоры обработки сигналов. Справочник. Остапенко А. Г., Лавлинский С. И., Сушков А. В. и др. Под ред. А. Г. Остапенко. М.: Радио и связь, 2014. 264 с.

19. Марпл-мл. С. Л. Цифровой спектральный анализ и его приложения : пер. с англ. / С. Л. Марпл-мл. — М. : Мир, 1990. — 584 с. — ISBN 5-03-001191-9.

20. Рабинер Л. Теория и применение цифровой обработки сигналов : пер. с англ. / Л. Рабинер, Б. Гоулд. — М. : Мир, 1978. — 848 с.

21. Оппенгейм А. Цифровая обработка сигналов : пер. с англ. / А. Оппенгейм, Р. Шафер. — М. : Техносфера, 2006. — 856 с. — ISBN 5-94836-077-6.

22. Бортник Г. Г. Цифровий метод спектрального оцінювання випадкових сигналів / Г. Г. Бортник, М. В. Васильківський, О. В. Стальченко // Вісник Вінницького політехнічного інституту. — 2014. — № 2. — С. 108—114.

23. Бортник Г. Г. Методи та засоби обробки високочастотних сигналів : моногр. / Г. Г. Бортник, В. М. Кичак. — Вінниця : УНІВЕРСУМ-Вінниця, 1998. — 132 с. — ISBN 966-7199-23-1.

ДОДАТОК А

КОД ПРОГРАМИ

```
unit archformu;

interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, StdCtrls;

type
  Tarchform = class(TForm)
    MainMenu1: TMainMenu;
    N1: TMenuItem;
    N2: TMenuItem;
    N3: TMenuItem;
    N4: TMenuItem;
    N5: TMenuItem;
    Edit1: TEdit;
    Label1: TLabel;
    Edit2: TEdit;
    Label2: TLabel;
    Button1: TButton;
    Button2: TButton;
    OpenDialog1: TOpenDialog;
    SaveDialog1: TSaveDialog;
    Button3: TButton;
    N6: TMenuItem;
    Button4: TButton;
    RadioButton1: TRadioButton;
    RadioButton2: TRadioButton;
    OpenDialog2: TOpenDialog;
    SaveDialog2: TSaveDialog;
    procedure N2Click(Sender: TObject);
    procedure N3Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure N5Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure N6Click(Sender: TObject);
    procedure RadioButton1Click(Sender: TObject);
    procedure RadioButton2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  archform: Tarchform;

implementation
uses mainu;

{$R *.dfm}
{$R-,I-,S-,G+}

const sb=256;
const sb1=512;

type obr=record
      vl:longint;
      len:byte
     end;
var a:array[0..511]of longint;
    b:array[byte]of word;
    c:array[0..1,byte]of word;
    st:array[byte]of byte;
    obc:byte;
    f,g:file;
    obb,size:longint;
    brp,brm,bwp:integer;
    bufr,bufw:array[0..sb-1]of byte;
    reof:boolean;
    o:array[byte]of obr;
    co,gb:byte;
    pp:word;
    bufr1,bufw1:array[0..sb1-1]of byte;
```

```
// функции архивации

procedure opget(var f:file;s:string);
begin
 assignfile(f,s);
 reset(f,1);
 blockread(f,bufr,sb,brm)
end;

procedure init;
begin
 brp:=0;
 brm:=0;
 bwp:=0;
 obb:=0;
 obc:=0;
 reof:=false
end;

procedure putc(var f:file;b:byte);
begin
 if bwp=sb then begin blockwrite(f,bufw,sb); bwp:=0 end;
 bufw[bwp]:=b;
 inc(bwp)
end;

procedure clput(var f:file);
begin
 putc(f,obb shr 24);
 blockwrite(f,bufw,bwp);
 close(f)
end;

procedure out(var out:file;ch:byte);
var glk:byte;
begin
 obb:=obb or o[ch].vl shl (32-o[ch].len-obc);
 inc(obc,o[ch].len);
 while obc>=8 do begin
  glk:=obb shr 24;
  putc(out,glk);
  obb:=obb shl 8;
  dec(obc,8)
 end
end;

procedure sift(l,r:word);
var i,j,x:word;
begin
 i:=l; j:=l+l+1; x:=b[l];
 if (j<r)and(a[b[j]]>a[b[j+1]])then inc(j);
 while (j<=r)and(a[x]>=a[b[j]])do begin
  b[i]:=b[j]; i:=j; j:=j+j+1;
  if (j<r)and(a[b[j]]>a[b[j+1]])then inc(j)
 end;
 b[i]:=x
end;

procedure obh(i:word;p:byte);
var j:integer;
begin
 if i<256 then begin
  o[i].len:=p;
  o[i].vl:=0;
  for j:=0 to p-1 do o[i].vl:=o[i].vl shl 1+st[j]
 end
 else begin
  st[p]:=0;
  obh(c[0,i-256],p+1);
  st[p]:=1;
  obh(c[1,i-256],p+1)
 end
end;

procedure build;
var i,p:word;
begin
 for i:=0 to 255 do b[i]:=i;
 for i:=127 downto 0 do sift(i,255);
```

```
 p:=0;
 for i:=255 downto 1 do begin
  c[0,p]:=b[0];
  b[0]:=b[i];
  sift(0,i-1);
  c[1,p]:=b[0];
  b[0]:=p+256;
  a[p+256]:=a[c[0,p]]+a[c[1,p]];
  sift(0,i-1);
  inc(p)
 end;
 obh(p+255,0)
end;

procedure getc(var f:file);
var ch:byte;
begin
 ch:=bufr[brp];
 out(g,ch);
 inc(a[ch]);
 inc(brp);
 if brp=brm then begin
  if eof(f) then reof:=true else build;
  blockread(f,bufr,sb,brm); brp:=0
 end
end;

// функции разархивации

procedure opget1(var f:file;s:string);
begin
 assign(f,s);
 reset(f,1);
 blockread(f,size,4);
 blockread(f,bufr1,sb,brm)
end;

procedure init1;
begin
 brp:=0;
 brm:=0;
 bwp:=0;
 co:=0;
 gb:=0;
 pp:=510;
 reof:=false
end;

procedure clput1(var f:file);
begin
 blockwrite(f,bufw1,bwp);
 close(f)
end;
procedure sift1(l,r:word);
var i,j,x:word;
begin
 i:=l; j:=l+l+1; x:=b[l];
 if (j<r)and(a[b[j]]>a[b[j+1]])then inc(j);
 while (j<=r)and(a[x]>=a[b[j]])do begin
  b[i]:=b[j]; i:=j; j:=j+j+1;
  if (j<r)and(a[b[j]]>a[b[j+1]])then inc(j)
 end;
 b[i]:=x
end;

procedure build1;
var i,p:word;
begin
 for i:=0 to 255 do b[i]:=i;
 for i:=127 downto 0 do sift1(i,255);
 p:=0;
 for i:=255 downto 1 do begin
  c[0,p]:=b[0];
  b[0]:=b[i];
  sift(0,i-1);
  c[1,p]:=b[0];
  b[0]:=p+256;
  a[p+256]:=a[c[0,p]]+a[c[1,p]];
  sift1(0,i-1);
  inc(p)
```

```
 end;
end;

function getc1(var f:file):byte;
begin
 getc1:=bufr1[brp];
 inc(brp);
 if brp=brm then begin
  blockread(f,bufr1,sb1,brm); brp:=0
 end
end;

procedure putc1(var f:file;b:byte);
begin
 inc(a[b]);
 if bwp=sb-1 then build1;
 if bwp=sb then begin
  blockwrite(f,bufw1,sb);
  bwp:=0
 end;
 dec(size);
 if size=0 then reof:=true;
 bufw1[bwp]:=b;
 inc(bwp)
end;

procedure getb1(var f:file);
var bb:byte;
begin
 if co=0 then begin gb:=getc1(f); co:=8 end;
 dec(co);
 bb:=gb shr co and 1;
 pp:=c[bb,pp-256];
 if pp<256 then begin putc1(g,pp); pp:=510 end
end;

procedure Tarchform.N2Click(Sender: TObject);
begin
edit1.Clear;
edit2.Clear;
label1.Caption:='';
label2.Caption:='';
Button3.Visible:=false;
Button4.Visible:=false;
Hide;
CloseFile(f);
CloseFile(g);
end;

procedure Tarchform.N3Click(Sender: TObject);
begin
Application.Terminate;
end;

procedure Tarchform.Button1Click(Sender: TObject);
begin
if (RadioButton1.Checked=true)then begin
OpenDialog1.Execute;
Edit1.Text:=OpenDialog1.FileName;
end;
if (RadioButton2.Checked=true)then begin
OpenDialog2.Execute;
Edit1.Text:=OpenDialog2.FileName;
end;

if (edit1.Text<>'')and(RadioButton1.Checked=true) then Button3.Visible:=true;
if (edit1.Text<>'')and(RadioButton2.Checked=true) then Button4.Visible:=true;
end;

procedure Tarchform.Button2Click(Sender: TObject);
begin
if (RadioButton1.Checked=true)then begin
SaveDialog1.Execute;
Edit2.Text:=SaveDialog1.FileName;
end;
if (RadioButton2.Checked=true)then begin
SaveDialog2.Execute;
Edit2.Text:=SaveDialog2.FileName;
end;
end;
```

```
procedure Tarchform.Button3Click(Sender: TObject);
var j:integer;
begin
 init;
 opget(f,OpenDialog1.FileName);
 size:=filesize(f);
 assignfile(g,SaveDialog1.FileName);
 rewrite(g,1);
 blockwrite(g,size,4);
 for j:=0 to 255 do a[j]:=1;
 build;
 while not reof do getc(f);
 clput(g);
 messagedlg('Архивация выполнена успешно',mtInformation,[mbOk],0);
end;

procedure Tarchform.N5Click(Sender: TObject);
begin
RadioButton1.Checked:=true;
end;

procedure Tarchform.Button4Click(Sender: TObject);
var j:integer;
begin
 init1;
 opget1(f,OpenDialog2.FileName);
 assignfile(g,SaveDialog2.FileName);
 rewrite(g,1);
 for j:=0 to 255 do a[j]:=1;
 build1;
 while not reof do getb1(f);
 clput1(g);
  messagedlg('Распаковка выполнена успешно',mtInformation,[mbOk],0);
CloseFile(f);
CloseFile(g);
end;

procedure Tarchform.N6Click(Sender: TObject);
begin
RadioButton2.Checked:=true;
end;

procedure Tarchform.RadioButton1Click(Sender: TObject);
begin
edit1.Clear;
edit2.Clear;
Button3.Visible:=false;
Button4.Visible:=false;
Label1.Caption:='Файл для архивации';
Label2.Caption:='Архивный файл';
end;

procedure Tarchform.RadioButton2Click(Sender: TObject);
begin
edit1.Clear;
edit2.Clear;
Button3.Visible:=false;
Button4.Visible:=false;
Label1.Caption:='Архивный файл';
Label2.Caption:='Распакованный файл';
end;

end.
```

4.3. Файл authorformu.pas

```
unit authorformu;

interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, jpeg, ExtCtrls, Menus;

type
  Tauthorform = class(TForm)
    Image1: TImage;
  private
    { Private declarations }
  public
    { Public declarations }
```

```
  end;

var
  authorform: Tauthorform;

implementation

uses mainu;

{$R *.dfm}

end.
```

4.4. Файл codeformu.pas

```
unit codeformu;

interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Menus;

type
  Tcodeform = class(TForm)
    Memo1: TMemo;
    Edit1: TEdit;
    Memo2: TMemo;
    Label1: TLabel;
    MainMenu1: TMainMenu;
    N1: TMenuItem;
    N2: TMenuItem;
    XOR1: TMenuItem;
    N3: TMenuItem;
    N4: TMenuItem;
    Label2: TLabel;
    Label3: TLabel;
    N5: TMenuItem;
    OpenDialog1: TOpenDialog;
    Label4: TLabel;
    Label5: TLabel;
    Button1: TButton;
    XOR2: TMenuItem;
    procedure XOR1Click(Sender: TObject);
    procedure N3Click(Sender: TObject);
    procedure N4Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure N5Click(Sender: TObject);
    procedure Memo1Change(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure XOR2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  codeform: Tcodeform;

implementation

uses mainu;

var
    codefile:textfile;
{$R *.dfm}

procedure XORCoding; // Xor-шифрование текста
var key,longkey,result:string;
    i:integer;
    toto:char;
begin
longkey:='';
result:='';
  key:=codeform.Edit1.Text;
  for i := 0 to (length(codeform.Memo1.Lines.Text) div length(key)) do
      longkey := longkey + key;
  for i := 1 to length(codeform.Memo1.Lines.Text) do begin
      toto := chr((ord(codeform.Memo1.Lines.Text[i]) XOR ord(longkey[i])));
      result := result + toto;
  end;
```

```
codeform.Memo2.Lines.Text:=result;
codeform.Memo2.Lines.SaveToFile('XORcode.txt');
end;

procedure Tcodeform.XOR1Click(Sender: TObject);
begin
if (label5.Caption='') then
  if (Memo1.Lines.Text='') then begin
messagedlg('Вы не выбрали файл для шифрования или не ввели текст',mtError,[mbOk],0);
exit;
                          end
                            else XORCoding
                       else XORCoding;
end;

procedure Tcodeform.N3Click(Sender: TObject);
begin
Memo1.Clear;
Memo2.Clear;
Label5.Caption:='';
Label5.Visible:=false;
Edit1.Text:='dip1251';
Hide;
end;

procedure Tcodeform.N4Click(Sender: TObject);
begin
Application.Terminate;
end;

procedure Tcodeform.FormShow(Sender: TObject);
begin
Edit1.SetFocus;
end;

procedure Tcodeform.N5Click(Sender: TObject);
begin
Memo1.Clear;
Memo2.Clear;
Label5.Caption:='';
Label5.Visible:=false;
Edit1.Text:='dip1251';
OpenDialog1.Execute;
if OpenDialog1.FileName='' then exit;
MainMenu1.Items.Items[MainMenu1.Items.IndexOf(N2)].Enabled:=true;
Label4.Visible:=true;
Label5.Visible:=true;
Label5.Caption:=OpenDialog1.FileName;
assignfile(codefile,OpenDialog1.FileName);
Memo1.Lines.LoadFromFile(OpenDialog1.FileName);
end;

procedure Tcodeform.Memo1Change(Sender: TObject);
begin
MainMenu1.Items.Items[MainMenu1.Items.IndexOf(N2)].Enabled:=true;
end;

procedure Tcodeform.Button1Click(Sender: TObject);
begin
Memo1.Lines.Text:='';
Memo2.lines.Text:='';
Label5.Caption:='';
label5.Visible:=false;
edit1.Text:='';
end;

procedure Tcodeform.XOR2Click(Sender: TObject);
begin
if (label5.Caption='') then
  if (Memo1.Lines.Text='') then begin
messagedlg('Вы не выбрали файл для расшифрования или не ввели текст',mtError,[mbOk],0);
exit;
                          end
                            else XORCoding
                       else XORCoding;
end;

end.
```

4.5. Файл mainu.pas

```
unit mainu;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, TeEngine, Series, ExtCtrls, TeeProcs, Chart, StdCtrls, Math,
  Menus;

type
  Tmainform = class(TForm)
    Chart1: TChart;
    OpenDialog1: TOpenDialog;
    Label1: TLabel;
    Series1: TAreaSeries;
    Chart2: TChart;
    Series2: TAreaSeries;
    MainMenu1: TMainMenu;
    N1: TMenuItem;
    N2: TMenuItem;
    N3: TMenuItem;
    N5: TMenuItem;
    N6: TMenuItem;
    N7: TMenuItem;
    N8: TMenuItem;
    N9: TMenuItem;
    N10: TMenuItem;
    N11: TMenuItem;
    N12: TMenuItem;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    Label11: TLabel;
    Label12: TLabel;
    Label13: TLabel;
    Label14: TLabel;
    Label15: TLabel;
    Label16: TLabel;
    N4: TMenuItem;
    Label17: TLabel;
    Label18: TLabel;
    Button1: TButton;
    Button2: TButton;
    procedure N2Click(Sender: TObject);
    procedure N3Click(Sender: TObject);
    procedure N6Click(Sender: TObject);
    procedure N12Click(Sender: TObject);
    procedure N7Click(Sender: TObject);
    procedure N11Click(Sender: TObject);
    procedure N10Click(Sender: TObject);
    procedure N4Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  mainform: Tmainform;

implementation

uses codeformu, archformu, authorformu, aboutformu;
{$R *.dfm}
const
     N=256;
type
     Tvi=array[1..N] of integer;
     Tve=array[1..N] of extended;
     Tm=array[1..N,1..N] of extended;
var
     X:Tvi; // исходная последовательность
     Y:Tvi; // результирующая последовательность
     infile:textfile; // рассматриваемый текстовый файл
```

```
     Asin,Acos:Tm; // матрицы коэффициентов преобразования
     Ysin,Ycos:Tve; // компоненты результирующей последовательности
     MX,MY:extended; // матожидание входной (X) и выходной (Y) последовательности
     DX,DY:extended; // дисперсия (X) и (Y)
     SqDevX,SqDevY:extended;// среднеквадратическое отклонение (X) и (Y)
     flag:byte; // признак результата (0,1,2 - откр., арх., зашифр. соотв.)

procedure ReadStrFromTextfile(StrLength:integer;var Str:string);
var i,count:integer;
    bufstr:string;
begin
count:=0;
Str:='';
Reset(infile);
repeat
Readln(infile,bufstr);
for i:=1 to length(bufstr) do begin
    count:=count+1;
    if count>=StrLength then break;
    Str:=Str+bufstr[i];
                                end;
until count=strLength;
Close(infile);
end;

procedure Furie(X:Tvi; var Y:Tvi);
     // функция спектрального преобразования Фурье
var
   i,j:integer;
begin
for i:=0 to N-1 do
  for j:=0 to N-1 do
    Asin[i,j]:=sin(2*pi*i*j/N);

for i:=0 to N-1 do
  for j:=0 to N-1 do
    Acos[i,j]:=cos(2*pi*i*j/N);

for i:=1 to N-1 do begin
  Ysin[i]:=0;
  Ycos[i]:=0;
                 end;

for i:=1 to N-1 do
  for j:=1 to N-1 do begin
     Ysin[i]:=Ysin[i]+Asin[i,j]*X[j];
     Ycos[i]:=Ycos[i]+Acos[i,j]*X[j];
                      end;
for i:=1 to N-1 do
  Y[i]:=round(sqrt(sqr(Ysin[i])+sqr(Ycos[i])));
end;

procedure Stat(x:Tvi; var MX,DX,SqDevX:extended);
              // функция анализа последовательности
var
   sumX:extended; // сумма элементов последовательности
   sumDevX:extended; // сумма квадратов отклонений
   i:integer;
begin
  sumX:=0;
  sumDevX:=0;
  for i:=1 to N do
     sumX:=sumX+x[i];
  MX:=sumX/N;
  for i:=1 to N do
     sumDevX:=sumDevX+sqr(x[i]-MX);
  DX:=sumDevX/N;
  SqDevX:=sqrt(DX);
end;

procedure Tmainform.N2Click(Sender: TObject); // Очистка текущего проекта
begin
Label1.Caption:='';
Chart1.Series[0].Clear;
Chart2.Series[0].Clear;
Label1.Visible:=false;
Label2.Visible:=false;
Height:=491;
Label18.Caption:='';
button1.Visible:=false;
```

```
button2.Visible:=false;
end;

procedure Tmainform.N3Click(Sender: TObject); // Открытие исследуемого текстового файла
var
    i:integer;
    str:string;
begin
chart1.Series[0].Clear;
chart2.Series[0].Clear;
Label7.Caption:=' не выполнен расчет ';
Label8.Caption:=' не выполнен расчет ';
Label9.Caption:=' не выполнен расчет ';
button1.Visible:=false;
button2.Visible:=false;
Label14.Caption:=' не выполнен расчет ';
Label15.Caption:=' не выполнен расчет ';
Label16.Caption:=' не выполнен расчет ';
Height:=491;
Label18.Caption:='';

OpenDialog1.Execute;
if OpenDialog1.FileName='' then exit;
Label1.Caption:='  '+OpenDialog1.FileName+'  ';
Label1.Visible:=true;
Label2.Visible:=true;
button1.Visible:=true;
assignfile(infile,OpenDialog1.FileName);
ReadStrFromTextfile(N,str); // Чтение строки из файла
for i:=1 to N do  begin
    X[i]:=Ord(str[i]);
chart1.Series[0].Add(X[i]);
                      end;
Stat(X,MX,DX,SqDevX);
Label7.Caption:=FloatToStr(RoundTo(MX,-2));
Label8.Caption:=FloatToStr(RoundTo(DX,-2));
Label9.Caption:=FloatToStr(RoundTo(SqDevX,-2));
end;

procedure Tmainform.N6Click(Sender: TObject);
var
   i:integer;
begin
chart2.Series[0].Clear;
Furie(X,Y);
for i:=1 to N do chart2.Series[0].Add(Y[i]);
Stat(Y,MY,DY,SqDevY);
Label14.Caption:=FloatToStr(RoundTo(MY,-2));
Label15.Caption:=FloatToStr(RoundTo(DY,-2));
Label16.Caption:=FloatToStr(RoundTo(SqDevY,-2));
Height:=519;
Result(MX,MY,DX,DY,SqDevX,SqDevY,flag);
if flag=0 then Label18.Caption:=' ОТКРЫТЫЙ ';
if flag=1 then Label18.Caption:=' АРХИВНЫЙ ';
if flag=2 then Label18.Caption:=' ЗАШИФРОВАННЫЙ ';
button2.Visible:=true;
end;

procedure Tmainform.N12Click(Sender: TObject);
begin
codeform.Show;
end;

procedure Tmainform.N7Click(Sender: TObject);
begin
archform.Show;
end;

procedure Tmainform.N11Click(Sender: TObject);
begin
authorform.Show;
end;

procedure Tmainform.N10Click(Sender: TObject);
begin
aboutform.Show;
end;

procedure Tmainform.N4Click(Sender: TObject);
begin
```

```
Application.Terminate;
end;

procedure Tmainform.Button1Click(Sender: TObject);
begin
chart1.SaveToBitmapFile('infile.bmp');
end;

procedure Tmainform.Button2Click(Sender: TObject);
begin
chart2.SaveToBitmapFile('outfile.bmp');
end;
end.
```

ДОДАТОК Б

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ДНІПРОВСЬКА ПОЛІТЕХНІКА»

Факультет інформаційних технологій
Кафедра програмного забезпечення комп'ютерних систем

ВІДГУК

Керівника
економічної
частини Касьяненко Л.В., к.е.н., доцента каф. ПЕП та ПУ
(прізвище, ім'я, по батькові, вчене звання, посада, місце роботи)

На кваліфікаційну роботу

студента _____
(прізвище, ім'я, по батькові)

курсу ІІ групи _____
спеціальності _____
на тему _____

_____

«___»_____2022 р. _____
(підпис)

ДОДАТОК В

ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ

Ім’я файла	**Опис**
Пояснювальні документи	
Диплом_ Анісімов.doc	Пояснювальна записка роботи. Документ Word.
Диплом_ Анісімов.pdf	Пояснювальна записка роботи в форматі PDF
Програма	
SPF-analysis.rar	Архів. Містить коди програми і откомпільовану програму
Презентація	
Презентація_ Анісімов.ppt	Презентація роботи