

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

Волянський Андрій Ігорович

**Програмний модуль кодування файлів на основі
алгоритму Base64/ The software module for file
encoding based on Base64 algorithm**

напрямок підготовки: 6.050102 - Комп'ютерна інженерія
фахове спрямування - Комп'ютерні системи та мережі
Бакалаврська робота

Виконав студент групи КСМ КСМ-
42/1
А.І. Волянський

Науковий керівник:
Івасьєв С.В.

Тернопіль - 2018

РЕЗЮМЕ

Дипломний проект містить 69 сторінок пояснюючої записки, 10 рисунків, 10 таблиць, 2 додатки. Обсяг графічного матеріалу 2 аркуші формату А3.

Метою дипломної роботи є розробка програмного засобу кодування текстової інформації алгоритмом BASE64.

Методи досліджень базуються на теорії алгоритмів (для аналізу розроблених методів та алгоритмів), алгоритмах сегментації (для виділення окремих символів ввхідній стрічці), технологій об'єктно-орієнтованого програмування (для програмної реалізації розробленого програмного модуля).

В дипломній роботі на основі аналізу існуючих алгоритмів структурного аналізу та алгоритмів кодування інформації було спроектовано та реалізовано програмний модуль кодування BASE64. Програмний засіб працює з текстовою інформацією та файлами.

Проведено тестування розробленого програмного додатку на мові програмування Delphi, що підтвердило доцільність використання структурного підходу при проектуванні та реалізації програмних систем кодування інформації.

Розроблений програмний продукт є ефективним засобом з простим інтерфейсом, що дозволяє вирішувати проблему кодування та обробки інформації.

Ключові слова: КС, BASE64, BASE32, ASCII, KOI-8, CP1252, CP1251, UNICODE.

RESUME

Diploma project to host 69 pages of explanatory notes, 10 figures, 10 tables, 2 dodatki. Oblig metropolitan materialu 2 arkushi format A3.

The method of graduate robotics is the deconstruction of the program koduvannya textual information using the BASE64 algorithm.

The methodologies are based on the theory algorithms (algorithms for algorithms), segmentation algorithms (for the sake of symbolizing the symbols in the input), and the technology of object-oriented programming (for the program implementation of the decoupled program module).

In the graduate robot on the fundamentals of the basic algorithms of structural analysis and algorithms in the koduvannya information boole, the real-life program code module BASE64 was designed. The program is written in the textual format of the files.

The testament of the robbed software dodatka on the Delphi program was performed, but it was confirmed that the structural design documentation was not available for the projected implementation of the software systems of the codebook information.

Disintegration of software products is an effective way to forgive an interface, but allow virishuvati problem koduvannya that obrobki informatsi.

Key words: CS, BASE64, BASE32, ASCII, KOI-8, CP1252, CP1251, UNICODE.

ЗМІСТ

Перелік умовних скорочень.....	9
Вступ.....	10
1 Аналіз предметної області.....	11
1.1 Поняття кодування інформації.....	11
1.2 Аналіз існуючих рішень.....	15
1.3 Аналіз поставленого завдання	21
2 Архітектура програмної системи.....	23
2.1 Обґрунтування вибору проектних рішень	23
2.2 Схема кодування Base64.....	26
2.3 Алгоритм кодування Base64.....	28
3 Проектування програмного засобу.....	35
3.1 Розробка графічного інтерфейсу.....	35
3.2 Розробка основних функцій.....	38
3.3 Тестування та верифікація програмного засобу.....	41
4 Техніко-економічний розділ.....	49
4.1 Розрахунок витрат на розробку програмного додатку.....	49
4.2 Визначення експлуатаційних витрат	54
4.3 Визначення економічної ефективності і терміну окупності капітальних вкладень.....	57
Висновки.....	59
Список використаних джерел	60
Додаток А Код програмного засобу.....	64
Додаток Б Довідка про використання	69

ДП.КСМ.07125/14.00.00.000.ПЗ				
Змн.	Лист	№ докум.	Підпис	Дата
Розробив		Волянський А.І.		
Перевір.		Івасьєв С.В.		
Консульт.		Паздрій І.Р.		
Н. Контр.		Гураль І.В.		
Затвердив		Березький О.М.		
ПРОГРАМНИЙ МОДУЛЬ КОДУВАННЯ ФАЙЛІВ НА ОСНОВІ АЛГОРИТМУ BASE64				
		Літ.	Арк.	Акрушів
		8	69	
ТНЕУ.ФКІТ. КСМ-42/1				

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

- КС – Комп'ютерна система.
- BASE64 – Стандарт кодування символів 64 значним кодом.
- BASE32 – Стандарт кодування символів 32 значним кодом.
- ASCII – Американський код обміну інформації.
- KOI-8 – 8 – ми бітна кодова таблиця сумісна з ASCII.
- CP1252 – Набір символів для кодування українських символів в 8 – бітному кодуванні.
- CP1251 – Набір символів для кодування російських символів в 8 – бітному кодуванні.
- Unicode – Стандарт кодування символів, який включає в себе всі мови світу.

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

При проектуванні абстрактних засобів відображення інформації ЗВІ виникає проблема оптимального кодування інформації. Термін "кодування" запозичений з теорії інформації і означає перетворення відомостей у сигнал, зручний для передавання по каналах зв'язку [1]. Щодо діяльності оператора кодування визначає спосіб представлення інформації за допомогою умовних символів. Проблема оптимального кодування - це вирішення питань вибору категорій коду, довжини алфавіту сигналів, компонування кодового знака, можливості компонування сигналів у групи.

Кодування інформації - це процес формування певного уявлення інформації. При кодуванні інформація представляється у вигляді дискретних даних. Декодування є зворотним до кодування процесом. У більш вузькому сенсі під терміном В«кодуванняВ» часто розуміють перехід від однієї форми подання інформації до іншої, більш зручною для зберігання, передачі або обробки. Комп'ютер може обробляти тільки інформацію, представлену в числовій формі. Вся інша інформація (наприклад, звуки, зображення, показання приладів і т. д.) для обробки на комп'ютері повинна бути перетворена в числову форму. За допомогою програм для комп'ютера можна виконати зворотні перетворення отриманої інформації. Отже розробка програмного модуля кодування файлів на основі алгоритму Base64 сактуальною задачею [2].

Метою дипломного проектування є розробка програмного продукту кодування файлів на основі алгоритму Base64 для операційної системи Windows, що забезпечить можливості кодування та декодування файлів та текстової інформації.

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Поняття кодування інформації

Кодування інформації - це процес формування певного представлення інформації. В більш вузькому розумінні під терміном «кодування» часто розуміють перехід від однієї форми представлення інформації до іншої, більш зручної для зберігання, передачі чи обробки.

Код - це набір умовних позначень (або сигналів) для запису (або передачі) деяких наперед відомих певних понять. Категорія коду визначається засобами кодування інформації, серед яких виділяють фігури, літери, цифри, колір, , розмір, орієнтацію, частоту тощо.

Вибір категорій коду залежить від характеру завдання, що вирішується. Категорія коду, що є ефективною для вирішення одних завдань не може бути ефективною для інших. Експериментальні дані свідчать, що для визначення кількісних характеристик максимально ефективною є категорія числа, для інформаційного пошуку - категорія кольору, а для впізнання - умовний знак.

Зазвичай кожен образ під час кодування (іноді говорять - шифрування) представляється окремим знаком. Знак - це елемент кінцевого набору відмінних друг від друга елементів [3].

Сукупність (допустимих) символів, з яких утворюють повідомлення (кодів, що їх кодують) називається алфавіт. Довжина алфавіту сигналів визначається кількістю можливих рівнів (станів) даної категорії коду з урахуванням збереження оптимальних умов розрізнення і впізнання сигналів, а також можливостями оперативної пам'яті людини (таблиця 1.1).

Довжина алфавіту збільшується завдяки використанню багатомірного кодування (наприклад, кольору, форми, розміру, просторової орієнтації). Застосування багатомірних кодів дещо знижує точність і швидкість декодування, але суттєво підвищує швидкість обробки інформації людиною-оператором.

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

Таблиця 1.1 - Довжина алфавіту за різних засобів кодування

Засоби кодування	Довжина алфавіту
Форма символів: - алфавітно-цифрова - абстрактна - асоціативна	необмежена 16 200
Розмір символів	6
Кольоровий алфавіт	11
Орієнтація і довжина лінії	4
Кількість крапок	5
Просторова орієнтація символів	8
Яскравість, частота мерехтіння	4

Комп'ютер може обробляти тільки інформацію, представлену в числовій формі, тому вся інша інформація, наприклад, звуки, зображення, покази приборів, для обробки на комп'ютері повинна бути перетворена в цифрову форму. Наприклад, щоб перевести в цифрову форму музичний звук, можна через невеликі проміжки часу вимірювати інтенсивність звуку на певних частотах, представляючи результати кожного виміру в числовій формі. За допомогою програм для комп'ютера можна виконати перетворення отриманої інформації, наприклад "накладати" на інформацію звуку з різних джерел.

Аналогічним чином на комп'ютері можна обробляти текстову інформацію. При введенні в комп'ютер кожна буква кодується певним числом, а при виведенні на зовнішні пристрої (екран або друк) для сприйняття людиною за числам побудовані зображення букв. Відповідність між набором букв і числами називається кодуванням символів.

Як правило, представлення всіх чисел в комп'ютері відбувається за допомогою нулів і одиниць (а не десяти цифр, як зазвичай прийнято). Іншими словами, комп'ютери зазвичай працюють в двійковій системі, оскільки при цьому пристрої для їх обробки мають значно більш просту схемотехнічну

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

будову. Введення цифр в комп'ютер і виведення їх для читання людиною може здійснюватися в звичайній десятковій формі, а всі необхідні перетворення виконують програми, що працюють на комп'ютері.

З появою комп'ютерів виникла необхідність кодування всіх видів інформації, з якими має справу людина. Але вирішувати задачу кодування інформації людство почало задовго до появи комп'ютерів. Найбільші досягнення людства - письменність і арифметика - це не що інше, як система кодування мовних та цифрових відомостей. Інформація ніколи не з'являється в чистому вигляді, вона завжди якимось чином представлена, закодована. Одна і та сама інформація може бути представлена (закодована) в декількох формах.

Двійкове кодування - один з розповсюджених способів представлення інформації. У обчислювальних пристроях, в роботах і пристроях з цифровим програмним управлінням, як правило, вся інформація, з якою має справу пристрій, кодується у вигляді слова двійкового алфавіту.

Основна операція, вироблена над окремими символами тексту - порівняння символів. При порівнянні символів найважливішими аспектами є унікальність коду для кожного символу та довжини цього коду, а сам вибір принципу кодування практично не має значення.

Для кодування тексту використовуються різні таблиці перекодування. Важливо, щоб при кодування та декодування одного й того самого тексту використовувалася одна і та ж таблиця.

Таблиця перекодування - таблиця, що містить упорядкований деяким чином список символів, що кодуються, відповідно до якого відбувається перетворення символу в його двійковий код і назад.

Найбільш поширені таблиці перекодування: KOI-8, ASCII, CP1251, Unicode. Історично склалося, що в якості довжини коду для кодування символів було вибрано 8 біт або 1 байт. Тому у більшості випадків одному символу тексту, що зберігається в комп'ютері, відповідає один байт пам'яті.

Різні комбінації з 0 і 1 при довжині коду 8 біт можуть бути в кількості 256 варіантів, тому з допомогою однієї таблиці кодування можна закодувати не більше 256 символів. При довжині коду в 2 байта (16 біт) можна кодувати

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

65536 символів.

Збіг в кодування цифрової та текстової інформації полягає в наступному: щоб можна було порівняти дані цього типу, для різних чисел (як і для різних символів) повинен бути різний код. Основна відмінність цифрових даних від символів полягає в тому, що над числами, крім операції порівняння, проводяться різноманітні математичні операції: додавання, множення, знаходження кореня, розрахунок логарифмів і інші математичні операції. Правила виконання цих операцій у математиці детально розроблені для чисел, представлених в позиційній системі числення.

Основною системою числення для представлення чисел у комп'ютері є двійкова позиційна система з числом [5].

Більша частина інформації, яка обробляється за допомогою комп'ютера, це текстова інформація, яка складається з символів: букв, цифр, розділових знаків та ін. Підрахуємо скільки знаків для кодування інформації нам потрібно: 10 цифр, 12 розділових знаків, 15 знаків арифметичних дій, букв руського та латинського алфавіту, всього: 155 символів, що відповідає 8 біт інформації.

Одиниці виміру інформації.

1 байт = 8 біт

1 Кбайт = 1024 байтів

1 Мбайт = 1024 Кбайт

1 Гбайт = 1024 Мбайт

1 Тбайт = 1024 Гбайт

Суть кодування полягає в тому, що кожному символу прийнята відповідність двійкового коду від 00000000 до 11111111 або відповідний йому десятковий код від 0 до 255 [6].

На даний час для кодування кирилиці використовуються п'ять різних кодових таблиць (KOI-8, CP1251, CP866, Mac, ISO), причому тексти, кодовані за допомогою однієї таблиці, не будуть правильно відображатися в іншому кодуванні.

Основним символом кодування символів є код ASCII - американський стандартний код для інформаційного обміну, американський стандартний код

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

обміну інформацією, який представляє собою таблицю 16-16, де символи кодуються в шістнадцятковій системі числення [7].

Кодування Base64 - це загальний термін для ряду аналогічних схем кодування, які кодують бінарні дані, обробляючи його чисельним способом і перетворюючи його на базу представлення 64. Термін Base64 походить від стандарту кодування MIME. Схеми кодування Base64 часто використовуються, для кодування бінарних даних, які потрібно зберігати та передавати через носії, призначені для обробки текстових даних. Це забезпечує збереження даних без змін при транспортуванні. Base64 часто використовується в ряді додатків, включаючи електронну пошту через MIME, та зберігання складних даних у XML [8].

Base64 буквально означає - позиційна система числення з основою 64. Тут 64 - це найбільша ступінь двійки (2^6), яка може бути представлена з впровадженням друкованих символів ASCII. Ця система широко використовується в електронній пошті для подання бінарних файлів в тексті листа (транспортне кодування). Весь діапазон закодованих символів укладається в англійський алфавіт, цифри і ряд: A-Z, a-z і 0-9, що складає 62 знаки, для яких бракує двох знаків в різних системах використовуються різні символи [9].

Важливим етапом кодування графічного зображення є розбивка його на дискретні елементи (дискретизація).

Основними способами представлення графіки для її зберігання та обробки за допомогою комп'ютера є растрові та векторні зображення [10].

Векторне зображення представляє собою графічний об'єкт, що складається з елементарних геометричних фігур (частіше всього відрізків і дуг). Положення цих елементарних відрізків визначається координатами точок і величини радіусу. Для кожної лінії вказуються двійкові коди типу лінії (суцільна, пунктирна, штрих-пунктирна), товщина та кольори.

Растрове зображення представляє собою сукупність точок (пікселів), отриманих в результаті дискретизації зображення у відповідності з матричним принципом.

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

Матричний принцип кодування графічних зображень полягає в тому, що зображення розбивається на задану кількість строк і стовпців. Потім кожен елемент отриманої сітки кодується за вибраним правилом [11].

Pixel (елемент зображення - елемент рисунка) - мінімальна одиниця зображення, колір та яскравість якого можна задавати незалежно від іншого зображення.

Відповідно до матричного принципу побудовані зображення, виведені на принтер, відображаються на екрані дисплея, отримані за допомогою сканера.

Якість зображення буде більше, чим більше "плотніше" розташовані пікселі, тобто чим більша роздільна здатність пристрою, і точніше кодується колір кожного з них [12].

Для чорно-білого зображення код кожного пікселя заданий одним бітом.

Якщо малюнок кольоровий, то для кожної точки заданий двійковий код її кольорів.

Точки і кольору кодуються в двійковому коді, якщо, наприклад, необхідно використовувати 16-бітним малюнком, то для кодування кожного пікселя буде потрібно 4 біта ($16 = 2^4$), а якщо є можливість використовувати 16 біт (2 байта) для кодування кольорів одного пікселя, то можна передавати тоді $2^{16} = 65536$ різних кольорів. Використання 3 байтів (24 біт) для кодування кольору однієї точки дозволяє відобразити 2^{24} (або близько 17 мільйонів) різних відтінків кольорів - так званих "істинних кольорів" режиму (True Color). Це використовується в даний час, але далеко не граничні можливості сучасних комп'ютерів [13].

Звук - це хвилі, що розповсюджуються в повітрі. По своїй природі звук є безперервним сигналом. Якщо перетворити звук в електричний сигнал (наприклад, за допомогою мікрофона), побачимо, як він плавно змінюється з потоком напруги [14].

Для комп'ютерної обробки аналоговий сигнал треба якоь перетворити в послідовність двійкових чисел, а для цього потрібно його дискретизувати і оцифрувати.

Можна виконати наступний порядок: вимірювати амплітуду сигналу

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

через рівномірні проміжки часу і записувати отримані числове значення в пам'ять комп'ютера.

1.2 Аналіз існуючих рішень

Неодноразово поштові сервіси надсилають листи по електронній пошті які мають приблизно наступний зміст: «ОПНІДЕР ОПЕІЙ-ЙНМТЕПЕМЖХ', ОНЯБ'ИЕММЮ' БШУНДС Б ЯБЕР ОНДЦНРНБКЕММНІ IREX ЙМХЦЙ" пНЯХІЯЙХі хМРЕПМЕР: МЮЙЮМСМ АНКЕЪХ ОЕПЕЛЬ "Дана шифровка означає лише те, що лист пришов в кодуванні відмінному від того, в якому за замовчанням працює поштовий клієнт. Якщо штатними засобами програми не вдається перевести текст у потрібне кодування, необхідно використовувати додаткове програмне забезпечення. Програма, яка працює з величезним кількостями та видами кодування, це продукт під назвою «Штирлиц» [15].

«Штирлиц» розшифровує тексти в кодуванні: Win-1251, KOI-8r, DOS, ISO-8859-5, MAC і ін., латинської транслітерації, форматів Quoted Printable, HTML та Unicode, закодовані (uuencode, ххencode, base64, binhex (Нqx7)) тексти та файли, а також тексти із змішаним перекодуванням, і мають фрагменти у різних кодуваннях.

Даний продукт приводив ці тексти до кодування Windows (1251) для читання та подальшої роботи з ними [16]. За допомогою «Штирліца», даний текст можна розглядати в будь-якому кодування, а також перетворити в будь-яке кодування (win, koi, dos, iso, mac, lat і др.). Також є можливість налаштувати та використовувати різні схеми транслітерації та кодування. Головне вікно програмного засобу наведено на рисунку 1.1.

Сам механізм розпізнавання може запускатися в автоматичному режимі, якщо вставити текст з буфера обміну, а також при завантаженні файлу і зберегти текст у будь-якому з перелічених кодів. «Штирлиц» має можливість стежити за зміною буфера обміну і приведення цього вмісту до кодування Win.

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

Після закінчення скачування розпаковуємо файли і запускаємо файл Shtirlitz.exe.

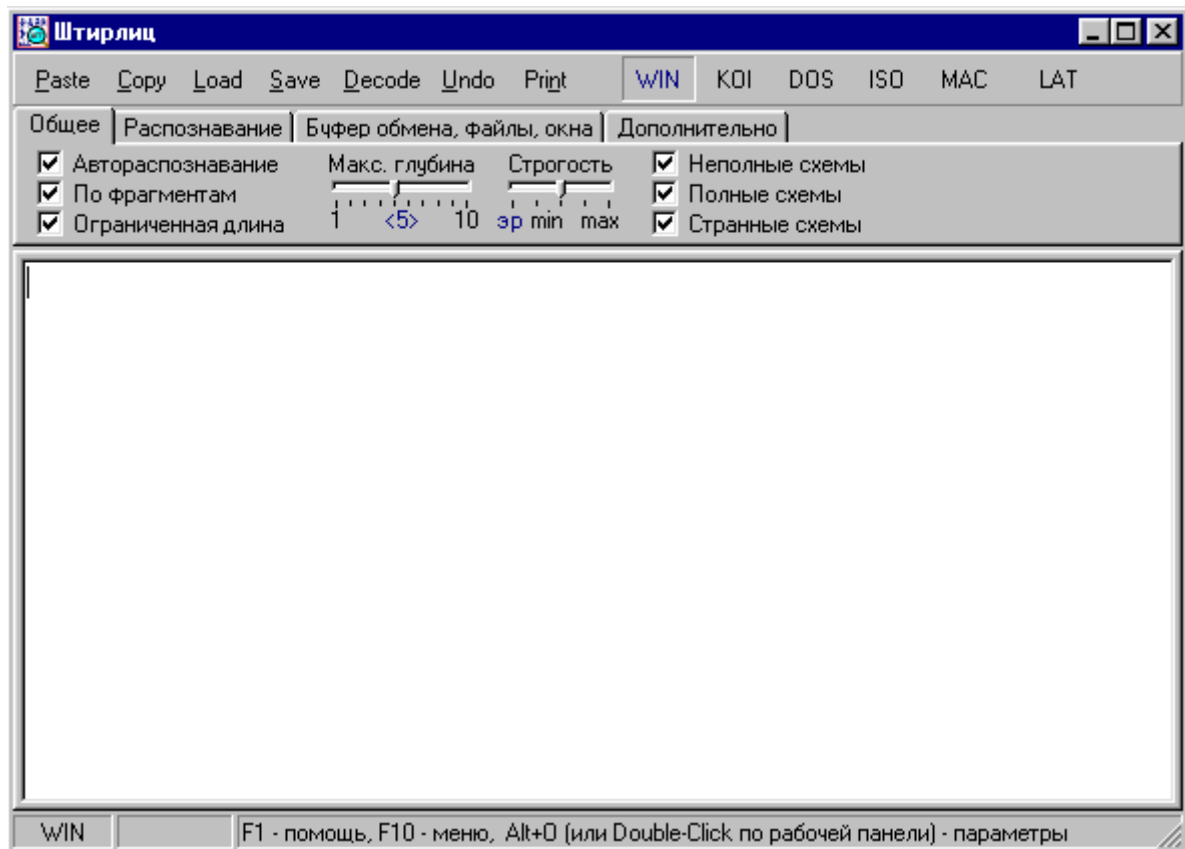


Рисунок 1.1 – Головне вікно додатку

У каталозі з програмою також містяться наступні файли:

- Shtirlitz.exe - програма "Штирлиц", версія 3.0,
- Shtirlitz.hlp - довідковий файл,
- Shtirlitz.cnt - вміст довідкової,
- Shtirlitz.lat - файл із описом схем транслітерації,
- Shtirlitz.ncd - файл з описом кодування,
- Shtirlitz.vcb - файл із словниками,
- uninstall.txt - керівництво по деінсталяціям,
- файли з прикладами.

Для зручності подальшого роботи можна створити на робочому столі ярлик програм.

Використання засобу можливе двома способами. Перший спосіб: можна

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

скопювати нечитабельний текст в обмін буфера (виділивши його, натиснувши правою кнопкою миші та обравши «Копіювати» або CTRL + C), вставити в вікно програми і натиснути «Decode». Іншим способом можна задати в програму розширення файлів, до яких будуть використовуватися можливості «Штирліца». Для цього відкриваємо програму «Провідник» і натискаємо правою кнопкою миші на потрібну файлу. Появиться контекстне меню, де вибираємо «Відкрити за допомогою Штирліца», і у відкритому вікні натискаємо «декодувати» [17].

Якщо потрібно перемістити «Штирліц» в інший каталог (наприклад C: \ Program Files \ SHTIRLITZ), то, при першому запуску з нового каталогу, програма буде перезаписуватися в реєстрі (автоматично).

Розглянемо елементи управління програмою, а саме головне меню, вигляд якого наведений на рисунку 1.2.

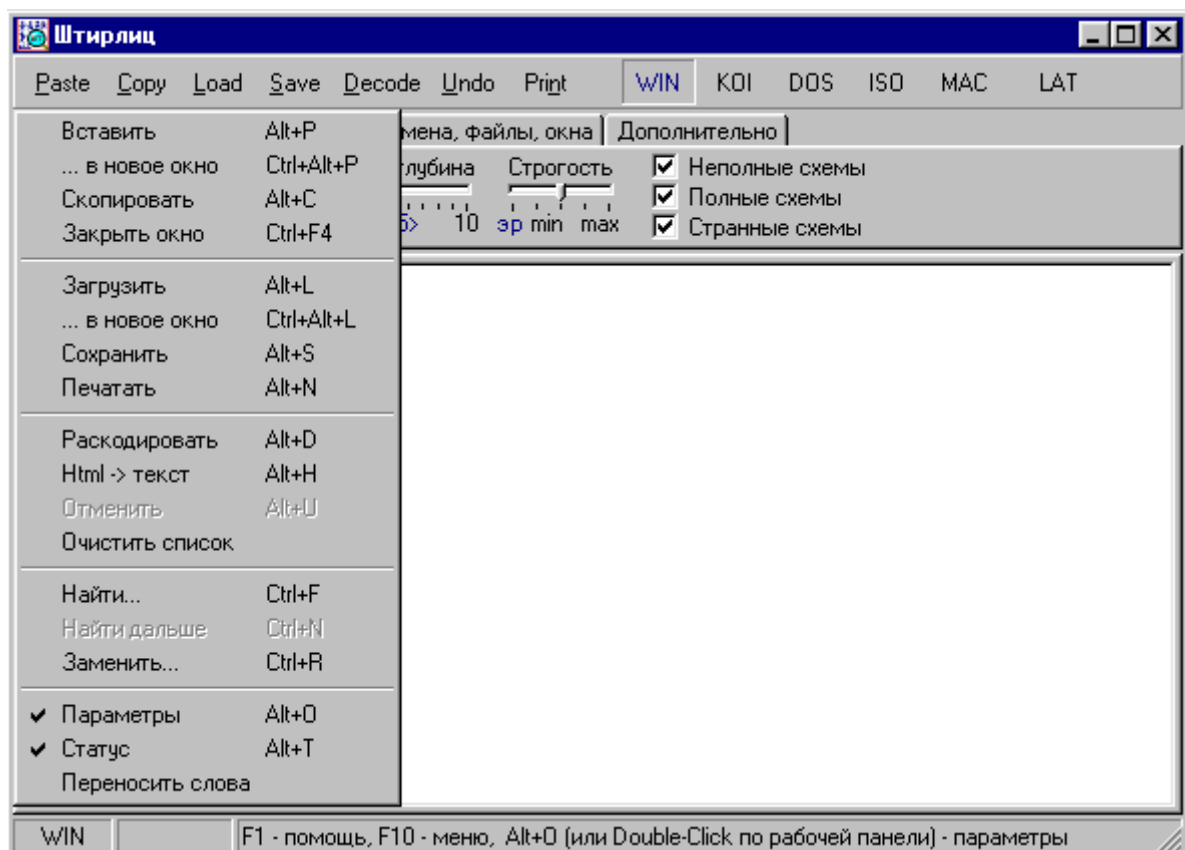


Рисунок 1.2 – Головне меню програмного засобу

Автозагрузка (Autodetect) - програма автоматично буде проводити декодування тексту відразу після вставки з обміну або після завантаження з

									ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата						19

файлу. Якщо в цьому пункті не вирізано з буферу обміну, то завантажений текст буде показано у вихідному кодуванні (поточне кодування показане утопленою кнопкою вліво в панелі статусу).

По фрагментам (пошук фрагментів) - при встановленому прапорці «Штирлиц» буде виділяти фрагменти тексту, які мають різне кодування. Буде виконаний пошук фрагментів, які містять більше двох букв кожного. Якщо потрібно розпізнати фрагменти менш двох букв, то потрібно вручну їх виділити і натиснути декод. При проведенні експрес-розпізнавання, фрагменти писатися не будуть і не буде показано прапорець [17].

Обмежена довжина - "Штирлиц" буде перевіряти кодування за першим словом тексту (умовно робить висновок, що і решту тексту написано в тому ж кодуванні). Ця функція значно прискорить роботу «Штирлица», і якщо її видалити, то буде перевірятися весь текст. Як описувалось вище при встановленому по фрагментам прапорці (фрагменти пошуку), цей пункт ігнорується автоматично [18].

Максимальна глибина вкладень (максимальна глибина) - встановлюється максимальна кількість вкладень. Якщо текст, для прикладу, взятий з електронної пошти, тоді буде достатньо глибини 2-4. Однак, коли програма знайде придатне перетворення, то сама завершує процес розпізнавання (можна зупинити даний процес).

Суворість - ця функція встановлює деякі якісні параметри (в текстах частіше зустрічаються опечатки та скорочення), але з іншого боку включає режим експрес розпізнавання, який хоч і працює швидше, але також має і свої особливості. Оптимальне положення цього пункту буде "посередині". Чем вище буде стояти суворість - тим більше комбінацій кодування до заданої глибини "Штирлиц" буде вимушений переглянути [19].

Схеми - підключає для авторозпізнавання різні набори схем. Неповні переводять лише символи, які відповідають кирилиці. Повні переводять символи, визначені в початковій та кінцевій кодовій сторінці. Істотно ніхто не може знати, в якому місці текст піддається якому-небудь перетворенню.

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

Режим експресс розпізнавання, коли встановлено в крайньому лівому положенні регулятора строгості прапорець [20].

Результат досягається швидше проте має ряд недоліків:

- текст не перевіряється на фрагменти з різними кодуваннями;
- обмежується довжина перевіряемого тексту.
- відсутня можливість переглянути краще з перевірених кодувань.
- у випадку невдачі програма не показує версії щодо послідовних кодувань, де хоч щось можна зрозуміти в тексті.

Вставити та копіювати – стандартні операції проте є декілька особливостей:

- вставляючи текст (paste), новий текст повністю замінює попередній;
- якщо встановлено прапорець «відкриття в новому вікні» або натиснуто CTRL, то вміст буфера з'явиться в новому вікні.

Закрити вікно - закриє вікно редактора, і якщо він буде один, то очистить його.

Зберегти файл - з'являється, коли виконується розкодування файлів у форматі uu-encode, xx-encode, base64, binhex.

Розкодувати (Decode), Відмінити (Undo) - знаходить кодування відокремлених фрагментів і перекодує їх у Win-1251. Якщо нічого не виділено, то перетворюється весь текст.

«Отменить» - відновлює текст до останньої операції.

Html-> Текст - виконує перетворення HTML-файлу в текстовий [21].

Очистити список (очистити список об'єктів) - очищає список останніх виконаних 10 перетворень, при яких «Штирліц» розшифровує текст. Кожен раз, коли програма розширює текст, перед перевіркою двосторонніх перетворень, перевіряє останні 10 успішних, що скорочує час операції по розшифруванні текстів, які мають одне кодування. Інколи попереднє перетворення майже правильно розшифровує текст і видає його як готовий результат. Щоб такого не відбулося, ця функція допомагає цьому уникнути [22].

Знайти, знайти далі, замінити - пошук по взірцю та заміна фрагментів.

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

Параметри, панель стану - показують або ховають панель параметрів і статусу в нижній частині екрана.

Перенестислова (Word Wrap) - вмикає або вимикає м'який перенос слів на нову стрічку, якщо вона перевищує ширину вікна.

1.3 Постановка завдання

Система кодування алгоритмом Base64 – програма призначена для виконання перекодування з різних кодових систем та таблиць символів в Base64 та навпаки. Необхідна простота інтерфейсу, авторські алгоритми роботи, додаткові функції. Можливість приводити тексти до кодування Windows (1251) для читання та подальшої роботи з ними. За допомогою програмного засобу, дані та текст можна розглядати в будь-якому кодуванні, а також перетворити в будь-яке кодування (win, koі, dos, iso, mac, lat). Також має бути можливість налаштовувати та використовувати різні схеми транслітерації та кодування.

В результаті аналізу технічного завдання було виділено наступні цілі проектування:

- розробка зручного графічного інтерфейсу;
- забезпечення можливості ручного кодування та перекодування в BASE64;
- можлива робота з файлами та фреймами;
- робота з буфером обміну;
- висока швидкодія.

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

2 ДОСЛІДЖЕННЯ КОДУВАННЯ BASE64

2.1 Кодування Base64

В електронних листах та інтернет мережі досить часто використовується кодування Base64, рідше використовується Base32 і ще рідше ZBase32, для кодування листів електронної пошти.

Часто виникає необхідність використовувати кодовані дані в адресі http-посилання. Стандартної реалізації в .net алгоритму кодування Base32 немає (на відміну від base64) тому були розроблені спеціальні процедури та модулі для подальшого їх використання [29].

Base64 – має ряд переваг та недоліків. Метод кодування дозволяє кодувати інформацію, представлену набором байтів, використовуючи всього 64 символи: AZ, az, 0-9, /, +. В кінці кодованої послідовності може бути декілька спецсимволів (зазвичай "=").

Серед переваг методу кодування є наступні спефікації:

- метод дозволяє представити послідовність будь-яких байтів в друкованих символах;
- у порівнянні з іншими Base-кодуваннями дає результат, який становить лише 133. (3)% від довжини вихідних даних.

Серед недоліків кодування цим алгоритмом є:

- чутливий до регістру алгоритм кодування.

Base32 - використовує лише 32 символу: AZ (або az), 2-7. Може містити в кінці кодованої послідовності кілька спецсимволов (по аналогії з base64).

До переваг Base32 можна віднести:

- послідовність будь-яких байтів переводить в друковані символи;
- регістронезалежне кодування;
- не використовуються цифри, занадто схожі на літери (наприклад, 0 схожий на O, 1 на l).

Серед недоліків кодування Base32:

- кодовані дані становлять 160% від вихідних.

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

З цього випливає, що якщо уявити вихідний набір байтів як єдиний масив бітів і розбити його на частини по 5 бітів, то отримаємо набір координат символів з алфавіту.

Алгоритми Base32 і Base64 аналогічні ZBase32, тільки використовуються різні алфавіти (за складом у випадку з Base32, за складом і розміром в випадку Base64) та кількості біт (6 біт для Base64) [30].

Перед тим, як почати розбиття вихідних даних на групи по 5 біт, необхідно підготувати місце куди буде записуватися результат. Щоб спростити написання коду замість статичних масивів, необхідно використати StringBuilder.

```
var encodedResult = new StringBuilder((int)Math.Ceiling(data.Length * 8.0 / 5.0));
```

При ініціалізації відразу необхідно задати розмір результуючого рядка (щоб не витратити ресурси на розширення в процесі роботи алгоритму).

Необхідно проаналізувати вихідний масив байтів і розділити його на 5-ти бітові послідовності. Для зручності програмний продукт буде працювати з групою по 5 байтів, так як це 40 бітне - число, кратне довжині "блоків послідовності".

Вихідні дані можуть мати розмір не кратний 5, тому необхідно врахувати можливість недостачі.

```
for (var i = 0; i < data.Length; i += 5) { var byteCount = Math.Min(5, data.Length - i);
```

Так як програмний засіб працює з групою з 5 байтів, додатку буде потрібен буфер, де буде формуватися суцільний набір бітів (всього 40 біт). Для виконання завдання необхідно оголосити змінну типу ulong (64 біти доступні для виконання процедури) і помістити туди поточну партію байтів.

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

```
ulong buffer = 0; for (var j = 0; j < byteCount; ++j) { buffer = (buffer << 8) | data [i + j]; }
```

Заключний етап - це "розподіл" того, що вийшло, на групи по 5 біт і формування результату.

```
var bitCount = byteCount * 8; while (bitCount > 0) { var index = bitCount >= 5 ? (int)(buffer >> (bitCount - 5)) & 0x1f : (int)(buffer & (ulong)(0x1f >> (5 - bitCount))) << (5 - bitCount); encodedResult.Append(EncodingTable [index]); bitCount -= 5; }
```

Процес декодування відбувається аналогічно процесу кодування, тільки в зворотному порядку.

2.2 Схема кодування Base64

Спочатку для передачі електронної пошти в Інтернет використовувався тільки текст (RFC822). Потім, з розвитком комп'ютерних комплектуючих, потрібна була можливість передачі нетекстової інформації: аудіо, відео, графічних файлів, файлів додатків і т.д. Однак поштові сервери які розуміли тільки текст, так і залишилися розуміти тільки його. Тому виникла потреба якимось чином перетворити двійковий файл в текстовий. Спосіб такого перетворення вже мав місце - це UUE кодування. Але з'явився ще один - base64. Цей спосіб використовується в специфікації MIME (RFC2045-2049) [31].

Це стандарт опису заголовків e-mail повідомлень. Використовуючи цей стандарт, в одному листі можна відправити відразу кілька різних вкладень. Наприклад, можна вкласти в вкладення: листи, як архівований файл, оформити

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

повідомлення як просто текст, а також помістити HTML-сторінку і відправити одержувачу.

Поштова програма-одержувач, з встроєним аналізатором MIME, абсолютно вільно з файлу електронної пошти (який насправді є "звичайним" текстовим файлом) витягне архів, відобразить повідомлення і пропарсить теги HTML. Деякі поштові програми, наприклад Outlook Express, дозволять запуснути вкладені в HTML-сторінку скрипти.

В один байт можна вкласти 256 цифр, від 0 до 255. Однак, якщо замість восьми байт використовувати тільки шість, то обсяг вкладеної інформації зменшується до 64 цифр, від 0 до 63. Тепер головне: будь-яку цифру 6-ти бітового байта можна представити в вигляді друкованого символу. 64 символи це не так багато, us-ascii символів цілком вистачить. Нижче представлений 64-х символний base64 "алфавіт":

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

0123456789 + / ,

де код символу A - 0, а код символу / - 63.

Далі беруться три послідовних байти по вісім біт (всього 24 біта), і побітно діляться на чотири 6-ти бітних байта (всього 24 біта) [32].

Досить нестандартний підхід: "шестибітний байт". Насправді біт вісім, проте використовуються тільки 6 молодших біт, два старших біти ігноруються.

Схематично такий "поділ три до чотирьох" можна представити, як показано на рисунку 2.2.

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

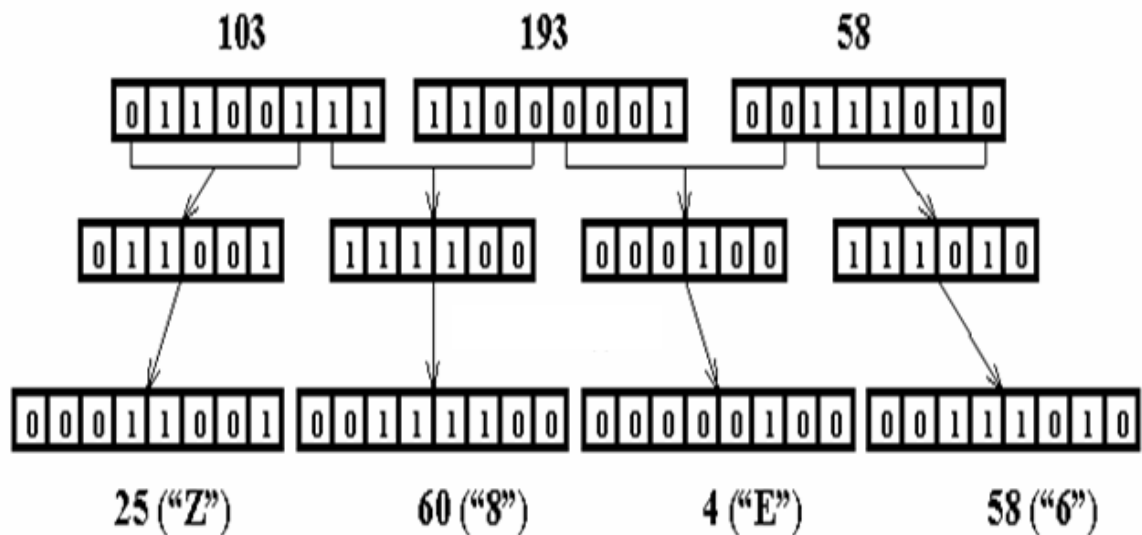


Рисунок 2.2 – Схема розподілу послідовності байтів на 4 групи

У наведеному прикладі три числа 103, 193 і 58 були закодовані в base64 формат. В результаті отриманий 4-х символний стрінг Z8E6.

Грунтуючись на цьому принципі, можна закодувати будь-яку двійкову інформацію в текст, причому не дуже сильно збільшуючи її обсяг (на 30%). Потім інформація через поштовий сервер потрапить до потрібного адресата, поштовик якого декодує текст в двійковий файл.

У випадку коли замість трьох байтів є тільки один або два в кінець чотирьохбайтного символного стрінга додається символ = (дорівнює). Якщо не вистачає (до трьох) одного байта, то додається один символ "дорівнює": Z8E = . Якщо не вистачає двох байт, то додаються два символи "дорівнює": Z8 ==.

Символ "дорівнює" треба аналізувати лише один раз - при читанні кінця файлу.

2.3 Алгоритм кодування base64

Формат Base64 має обмеження - загальна довжина рядка, що складається з 4-х символних стрінгів становить 72 символи (за винятком самого

FABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAF
FFFABRRRQAUUUUAFFFFABRRRQAUUUU
AFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFF
FABRRRQAUUUUAFFFFABRRRQA
UUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUaf / Z ----- ----
73121211AFBA8E3--

Content, boundary і все інше, що йде після рядка Mime-Version: 1.0 - це стандарт MIME. Букви, написані в koi8-r, читаються у вигляді хаотичного набору символів. Тому що кодування цього документа встановлено в windows-1251. Якщо у браузері встановити кодування koi8-r (в Internet Explorer 5.0 в головному меню вибрати "Вид" - "Вид кодування" - "Кирилиця (KOI8-R)"), то ці букви можна буде прочитати.

В кінці тексту листа знаходиться дуже великий блок ще одного набору символів. У наведеному тексті він дещо скорочений. Цей блок і є двійковий файл закодований base64 в текст. Трохи вище "великого блоку" видно, що для кодування застосовувався base64, видно також назви файлу. Те, що вище - реальна назва файлу і його тип: 'ex1.jpg', тип файлу image / jpeg, нижче - назва цього файлу у вікні вкладення листів.

Існує найоптимальніший і швидкий алгоритм кодування і декодування base64.

Розглядаючи ідеологію base64, перше, що необхідно - це влаштувати невеликий бітовий конвеєр. Тобто (для випадку кодування) взяти 8-ий (старший) біт вихідного байта і помістити його в початок 6-ти бітового байта.

Потім на місце 8-го біта вихідного байта помістити 7-й біт, а в 6-ти бітовий байт перший біт (молодший) перемістити на місце 2-го біта. Після такого переміщення звільняється перший (молодший) біт 6-ти бітного числа. У нього і необхідно помістити старший (колишній сьомий) біт вихідного байта. Потім ще раз пересунемо біти в обох байтах.

Схема алгоритму наведена на рисунку 2.3.

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

0 0 0 0 0 0

1 0 1 1 1 0 1 1

Початковий стан

0 0 0 0 0 0 1

1 0 1 1 1 0 1 1

Крок 1. Встановлення в молодший біт 6-ти бітового байту значення старшого біта 8 - ми бітового байту

0 0 0 0 1 0
shl

0 1 1 1 0 1 1 0
shl

Крок 2. Побітовий зсув всліво

.. повторюємо крок 1 і 2 шість раз ..

Крок 3-12

1 0 1 1 1 0

1 1 0 0 0 0 0 0

Кінцевий стан

Рисунок 2.3 – Схема кодування Base64

Розглянемо крок 1. Як відомо стандартні операнди мови Delphi AND і OR можуть виступати в двох якостях: бітова арифметика і логіка. У першому кроці використовуємо оператор AND як бітову операцію.

Для перевірки установки старшого біта вихідного 8-ми бітного байта накладемо на нього так звану "маску". Тобто необхідно застосувати до нього побітову операцію AND з числом 128 (10000000), як це показано на рисунку 2.4.

Змн.	Арк.	№ докум.	Підпис	Дата

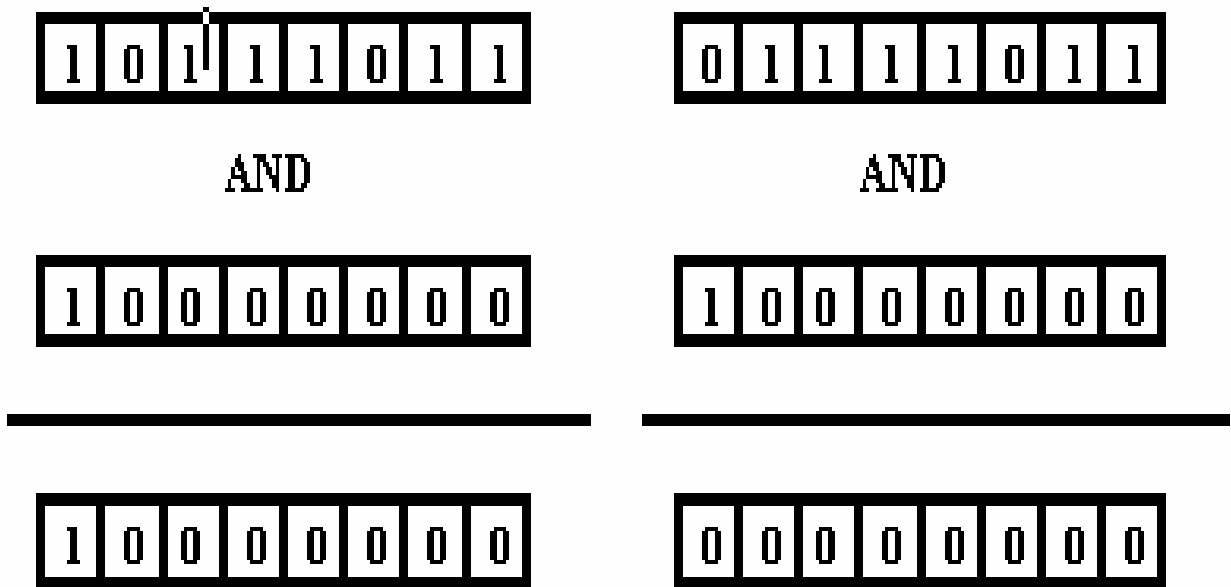


Рисунок 2.4 – Накладання маски

Як видно з наведеної схеми, перевірити установку старшого біта зовсім нескладно. Якщо результатом операції виходить число 128, значить біт встановлений, а якщо в результаті отримуємо 0, значить старший біт не встановлений. Далі в залежності від отриманого результату можна застосувати до 6-ти битового байту бітову операцію OR з числом 1, яка в будь-якому випадку застосування встановлює перший (молодший) байт в 1, як показано на рисунку 2.5.

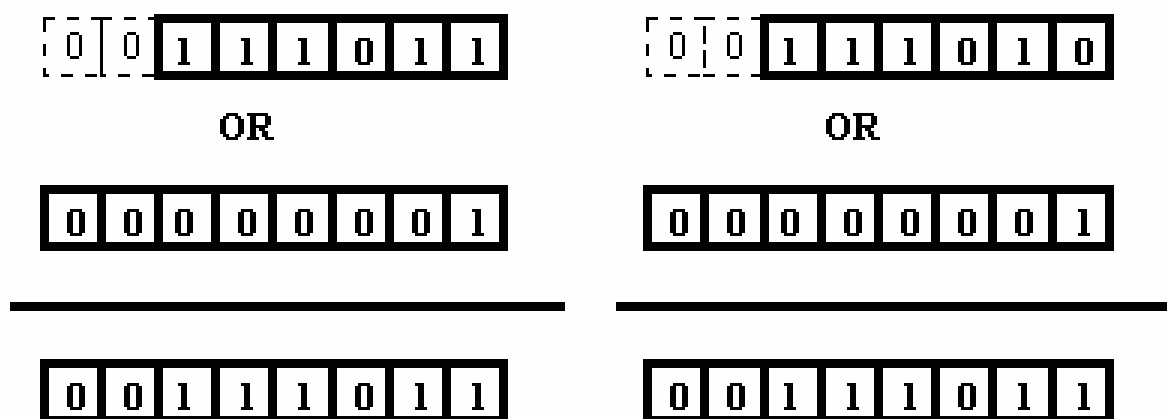


Рисунок 2.5 – Встановлення молодшого біту

Необхідно враховувати те, що перед застосуванням побітової операції OR з числом 1 молодший біт 6-ти бітового байта завжди 0. У випадку для перших двох кроків тому, що біти були обтулені при ініціалізації, для подальших кроків, тому що застосовується операція SHR.

Реалізація першого кроку в delphi:

```
If (Source AND 128) = 128 Then
```

```
Destination: = Destination OR 1;
```

де Source - 8-ми бітний байт джерело, а Destination - 6-ти бітний байт приймач.

Виконаємо крок 2. Змістимо перший біт 6-ти бітного байта на другу позицію, одночасно обнуляючи його перший біт.

```
Destination: = Destination SHR 1;
```

Змістимо сьомий біт 8-ми бітного байта на 8-е (старше) місце.

```
Source: = Source SHR 1;
```

Операція X SHR Y, зміщує в байті X біти на Y-місце вліво і автоматично встановлює Y-молодші байти в 0, отже значення числа змінюється. Тому в реальному кодї необхідно працювати не з самим байтом-джерелом, а з його тимчасової копією.

Виконаємо подальші кроки. Повторюємо кроки 1 і 2 шість разів для кожного біта 6-ти бітного байта. Одночасно необхідно стежити за тим, що б програмний засіб після опрацювання восьмого біта байта-джерела, слід перейти до нового байту. Якщо оброблені всі три байта-джерела, слід виконати вихід з функції кодування.

Декодування по суті нічим не відрізняється від кодування. Необхідно виконати ті ж самі дії, тільки джерело і приймач міняються місцями. А бітовий конвеєр повинен продовжувати працювати в ту ж сторону - справа - на ліво:

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

необхідно виконати перевірку першого біта 6-ти бітного байта (маску в цьому випадку треба накладати з числом 32 (0010000) - не вісім бітів, а шість) і в залежності від результату встановити (або не встановити) молодший байт 8-ми бітного байта. Потім виконати побітовий зсув вліво.

При кодуванні треба зчитувати по 58 байт, а не по 3. Зчитування по 58 байт значно збільшить "швидкість" розробленого програмного засобу.

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

3 РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ

3.1 Розробка графічного інтерфейсу

При розробці програмного продукту було використано наступні модулі:

- Windows;
- Messages;
- SysUtils;
- Classes;
- Graphics;
- Controls;
- Forms;
- Dialogs;
- StdCtrls;
- Menus;
- Unit3.

Для побудови графічного інтерфейсу використано компоненти наступних класів:

- TButton;
- TOpenDialog;
- TSaveDialog;
- TMainMenu;
- TMenuItem;
- TGroupBox;
- TMemo.

Загальні принципи яких потрібно дотримуватись при проектуванні графічного інтерфейсу складаються з декількох правил.

Інтерфейс повинен бути інтуїтивно зрозумілим. Таким, щоб користувачеві не потрібно пояснювати як ним користуватися. Алгоритм кодування Base64 наведено в ДП.КСМ.07125/14.00.00.000 С1.

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

Для спрощення процесу вивчення необхідна довідка - графічна підказка, що пояснює значення того чи іншого елементу. Повне керівництво повинно бути частиною інтерфейсу, доступною в будь-який момент.

Завжди необхідно повертати користувача в те місце, де він закінчив роботу в минулий раз.

Найчастіше, користувачі в інтерфейсі спочатку шукають сутність (іменник), а потім дію (дієслово) до неї. Необхідно користуватись правилом «іменник -> дієслово». Наприклад, шрифт -> змінити. Алгоритм декодування Base 64 приведений на ДП.КСМ.07125/14.00.00.001 С1.

Чим швидше людина побачить результат - тим краще. Приклад - «живий» пошук, коли варіанти, в процесі набору пошукового запиту система пропонує варіанти вибору пошукового запиту. Основний принцип: програма повинна взаємодіяти з користувачем на основі найменшої значущої одиниці введення.

Розроблений графічний інтерфейс приведено на рисунку 3.1.

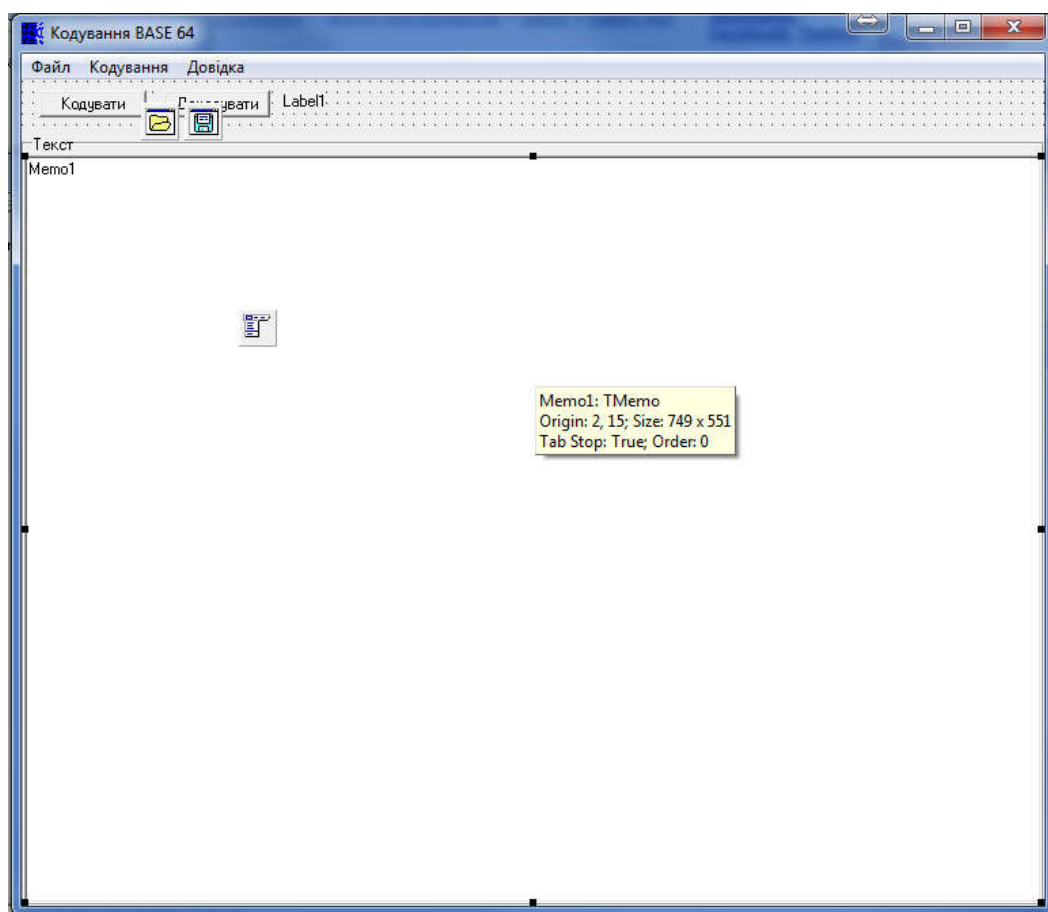


Рисунок 3.1 – Головне вікно додатку на етапі проектування

									Арк.
									36
Змн.	Арк.	№ докум.	Підпис	Дата					

Слід з обережністю надавати користувачеві можливість, по встановленню особистих налаштувань.

Для користувача розроблено просте інтуїтивно зрозуміле головне меню, яке відображене на рисунку 3.2.

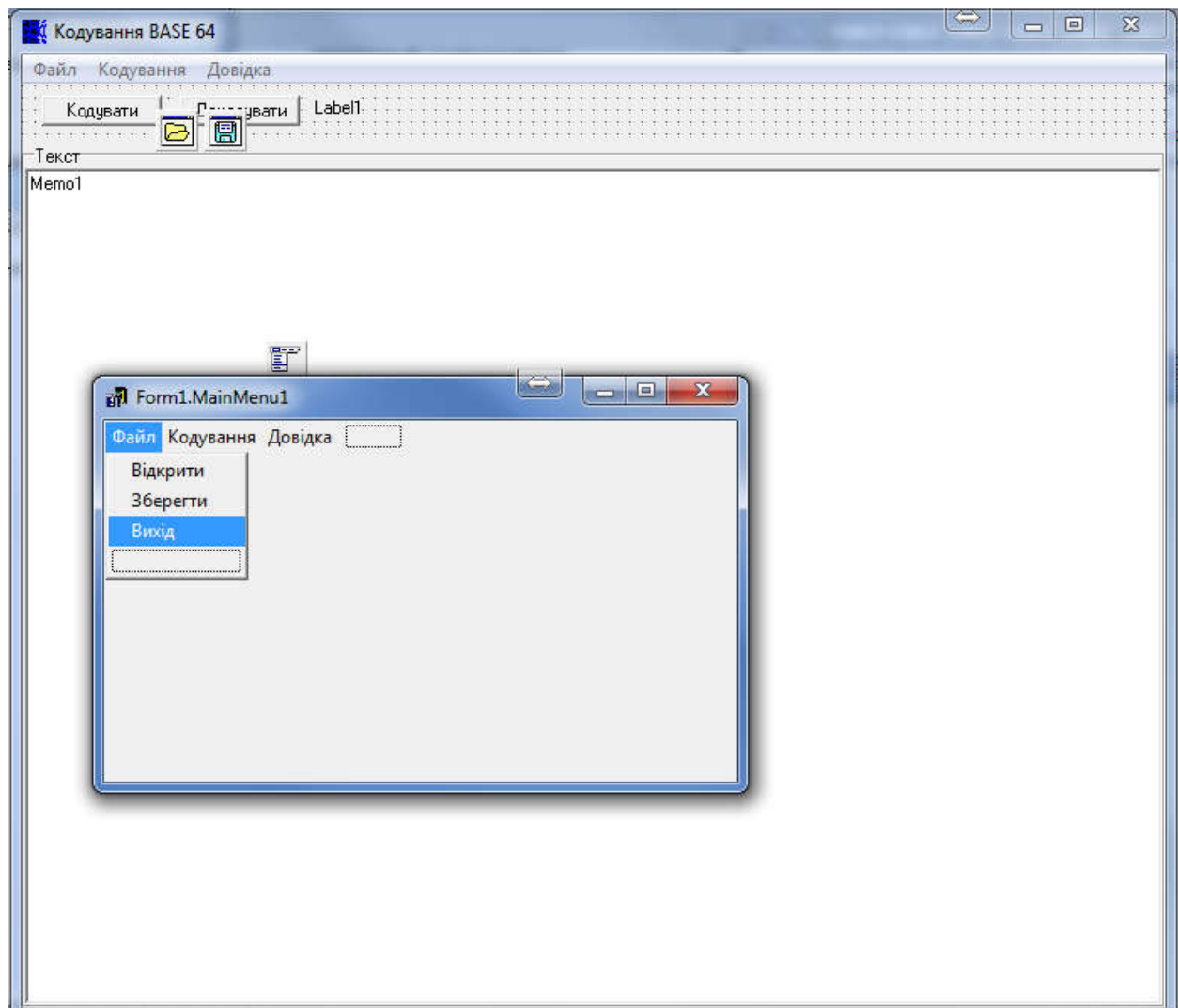


Рисунок 3.2 – Головне меню додатку. Етап проектування.

Чим більше користувач працює з якимось конкретним завданням, тим більше він на ньому концентрується і тим менше перестає помічати підказки та повідомлення, виведені програмою.

Чим більш критичним є завдання, тим менша ймовірність того, що користувач помітить попередження щодо тих чи інших потенційно небезпечних дій.

3.2 Розробка основних функцій

При проектуванні основних функцій було розроблено дві процедури, які дозволяють виконувати кодування та декодування, як файлів так і текстової інформації. Для кодування розроблено наступну процедуру:

Оголошення процедури декодування виглядає так

```
procedure TForm1.Button1Click(Sender: TObject);
const
  Base64MaxLength=72;
var
  hFile: integer;
  base64String: string;
  base64File: textfile;
  Base64: TBase64;
  Buf: array [0..2] of Byte;
```

Блок виконання команд, складається з відкриття файлу, розбиття бітової послідовності на групи та власне процесу декодування.

Процедура відкриття файлу має наступний код:

```
begin
  OpenFileDialog1.Title:='Оберіть файл для кодування base64 ';
  if not OpenFileDialog1.Execute then Exit;
  Label1.Caption:='Виконується кодування...';
  Application.ProcessMessages;
  base64String:='';
  hFile:=FileOpen(OpenFileDialog1.FileName,fmOpenReadWrite);
  AssignFile(base64File,OpenFileDialog1.FileName+'.b64');
  Rewrite(base64File);
```

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

```
FillChar(Buf,SizeOf(Buf),#0);
```

Читання коду та саме кодування алгоритмом base64 виглядає так:

```
repeat  
  Base64.ByteCount:=FileRead(hFile,Buf,SizeOf(Buf));  
  Move(Buf,Base64.ByteArr,SizeOf(Buf));  
  base64String:=base64String+CodeBase64(Base64);  
  if Length(base64String)=Base64MaxLength  
  then  
    begin  
      Writeln(base64File,base64String);  
      base64String:="";  
    end;
```

Після виконаного кодування дані збереженні в змінну base64String зберігаються до файлу як наведено в коді.

```
until Base64.ByteCount<3;  
Writeln(base64File,base64String);  
CloseFile(base64File);  
FileClose(hFile);  
Label1.Caption:='Кодування виконано!';  
ShowMessage('Файл '+ExtractFileName(OpenDialog1.FileName)+'  
збережено в кодуванні base64!'+#13#10+  
'файл '+ExtractFileName(OpenDialog1.FileName)+'.b64');  
Memo1.Lines.LoadFromFile(OpenDialog1.FileName);  
end;
```

Для декодування було розроблено процедуру, яка максимально ефективно виконує процес перетворення тексту з BASE64 в кодування sr1252.

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

Оголошення процедури для декодування файлу містить змінні, які будуть використані при використанні в коді.

```
procedure TForm1.Button2Click(Sender: TObject);  
var  
    base64File: textfile;  
    BufStr: string;  
    base64String: string;  
    Base64: TBase64;  
    hFile: integer;  
begin
```

Для відкриття файлу та створення нового пустого файлу в якому буде розміщена декодована інформація написано наступний код.

```
    OpenFileDialog1.Title:='Оберіть файл';  
    SaveDialog1.Title:='Виберіть файл для зберігання';  
    if not OpenFileDialog1.Execute then Exit;  
    if not SaveDialog1.Execute then Exit;  
    Label1.Caption:='Виконується декодування...';  
    Application.ProcessMessages;  
    AssignFile(base64File,OpenDialog1.FileName);  
    Reset(base64File);  
    hFile:=FileCreate(SaveDialog1.FileName);
```

Читання з файлу та процедура декодування інформації записує в файл декодовану інформацію використовуючи буфер BufStr.

```
    while not EOF(base64File) do  
    begin  
        Readln(base64File,BufStr);
```

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

```

while Length(BufStr)>0 do
begin
base64String:=Copy(BufStr,1,4);
Delete(BufStr,1,4);
Base64:=DecodeBase64(base64String);
FileWrite(hFile,Base64.ByteArr,Base64.ByteCount);
end;
end;

```

Наступний код виконує закривання файлу та виводить результат виконання операцій кодування в Label1 та виводиться Message.

```

Label1.Caption:=Виконується кодування!';
FileClose(hFile);
CloseFile(base64File);
ShowMessage('Кодування виконано!');
Memo1.Lines.LoadFromFile(SaveDialog1.FileName);
end;

```

Повний код програмного засобу наведено в додатку А та містить, додаткові функції для роботи з MIME та емейл файлами.

3.3 Тестування та верифікація програмного засобу

Тестування програмного забезпечення - це процес, що використовується для виміру якості розроблюваного програмного забезпечення.

Тестування - це процес технічного дослідження, який виконується на вимогу замовників, і призначений для вияву інформації про якість продукту відносно контексту, в якому він має використовуватись.

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

До цього процесу входить виконання програми з метою знайдення помилок. Певне уявлення про справжні причини появи помилок у роботі програми дає таке процентне співвідношення джерел збоїв: вхідні дані та апаратура складають 2%, системне програмне забезпечення – 3%, помилки користувачів – 5%, розробка системи – 15% та безпосередньо програмування 75%. Можна виділити два типи програмних помилок:

– синтаксичні помилки - виникають через порушення правил мови програмування. Такі помилки зазвичай виявляються під час компіляції. Можуть бути виключені порівняно легко. Навіть якщо не переглядати текст програми можна бути впевненим, що компілятор на стадії трансляції знайде помилки і видасть відповідні попередження. Фактично пошук помилок здійснює компілятор, а їхнє виправлення - програміст;

– семантичні (логічні) помилки - ті, що призводять до некоректних обчислень або помилок під час виконання (run-time error). Семантичні помилки усуваються зазвичай за допомогою виконання програми з ретельно підібраними перевірочними даними, для яких відома правильна відповідь.

На сьогоднішній день не існує таких методів тестування, які б могли виявити повністю всі помилки, тому всі існуючі методи діють в рамках перевірки розроблюваного програмного забезпечення. Цей процес називається верифікація. Є багато рівнів, методів та технік тестування, але найбільш доцільні для даної системи буде модульне, інтеграційне та системне тестування.

Системне тестування – це тестування функціональних та нефункціональних вимог.

Щоб перевірити, чи програму взагалі варто починати тестувати, було проведено Smoke тестування, протягом якого були перевірені основні функції програми, а саме:

- кодування файлів алгоритмом BASE64;
- декодування файлів алгоритмом BASE64;
- зберігання файлів;
- відкриття бінарних файлів та робота з ними.

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

Для кращого розуміння було розроблено test-case даного етапу перевірки, зображений на таблиці 3.1

Таблиця 3.1 - Smoke Тестування

ID	Name	Precondition	Steps	Expected result	Actual result	Status	Environment
SMOKE	Smoke testing	Install and open the application	1 Fill the Start position field 2 Fill the End position field 3 Click “Кодувати чисел” button 4 Click “Декодувати” button 5 Chose Destination of saving 6 Click “OK”	File with saving results is saved in Destination folder.	File with saving results is saved in Destination folder.	PASS	Windows XP, 7, 8.1, 10

Наступним етапом перевірки стала перевірка всіх функціональних вимог до додатку на позитивні та випадки. Для цього було розроблено Test suite, зображений в таблиці 3.2.

Таблиця 3.2 - Тестування функціональних значень

ID	Name	Precondition	Steps	Expected result	Actual result	Status	Environment
1	2	3	4	5	6	7	8
FUNC1	Searching of Primes, by filling fields	Install and open the application	1 Fill the Start position field 2 Fill the End position field 3 Click “Відкрити” button	List with primes from choosed position is showed	List with primes from choosed position is showed	PASS	Windows XP, 7, 8.1, 10
FUNC2	Searching of Primes with help of slider	Open the application	1 Fill the Start position field with help of slider 2 Fill the End position field with help of slider 3 Click “Зберегти” button	List with primes from choosed position is showed	List with primes from choosed position is showed	PASS	Windows XP, 7, 8.1, 10

1	2	3	4	5	6	7	8
FUNC3	Saving of Primes	FUNC1 Test case is done	1 Click “Зберегти” button. 2 Chose Destinaion of saving 3 Click “Кодувати”	File with saving results is saved in Destinati on folder.	File with saving results is saved in Destin ation folder.	PASS	Windows XP, 7, 8.1, 10
FUNC4	Saving of Primes, that are chosen with help of slider	FUNC2 Test case is done	1 Click “Зберегти” button. 2 Chose Destinaion of saving 3 Click “Декодувати”	File with saving results is saved in Destinati on folder.	File with saving results is saved in Destin ation folder.	PASS	Windows XP, 7, 8.1, 10

Модульне тестування – це таке тестування, в якому перевіряється кожен модуль системи, незалежно від програмного продукту, який розробляється. Його ціллю є показати, що кожен модуль може працювати окремо.

Інтеграційне тестування – це такий вид тестування коли всі модулі об'єднуються і тоді тестуються разом.

Програму було перевірено на декількох версіях операційної системи Windows, а саме:

- Windows XP.
- Windows 7.
- Windows 8.1.
- Windows 10.

У всіх випадках спостерігалась однакова поведінка.

Наступним кроком було вирішено перевірити додаток на негативні сценарії.

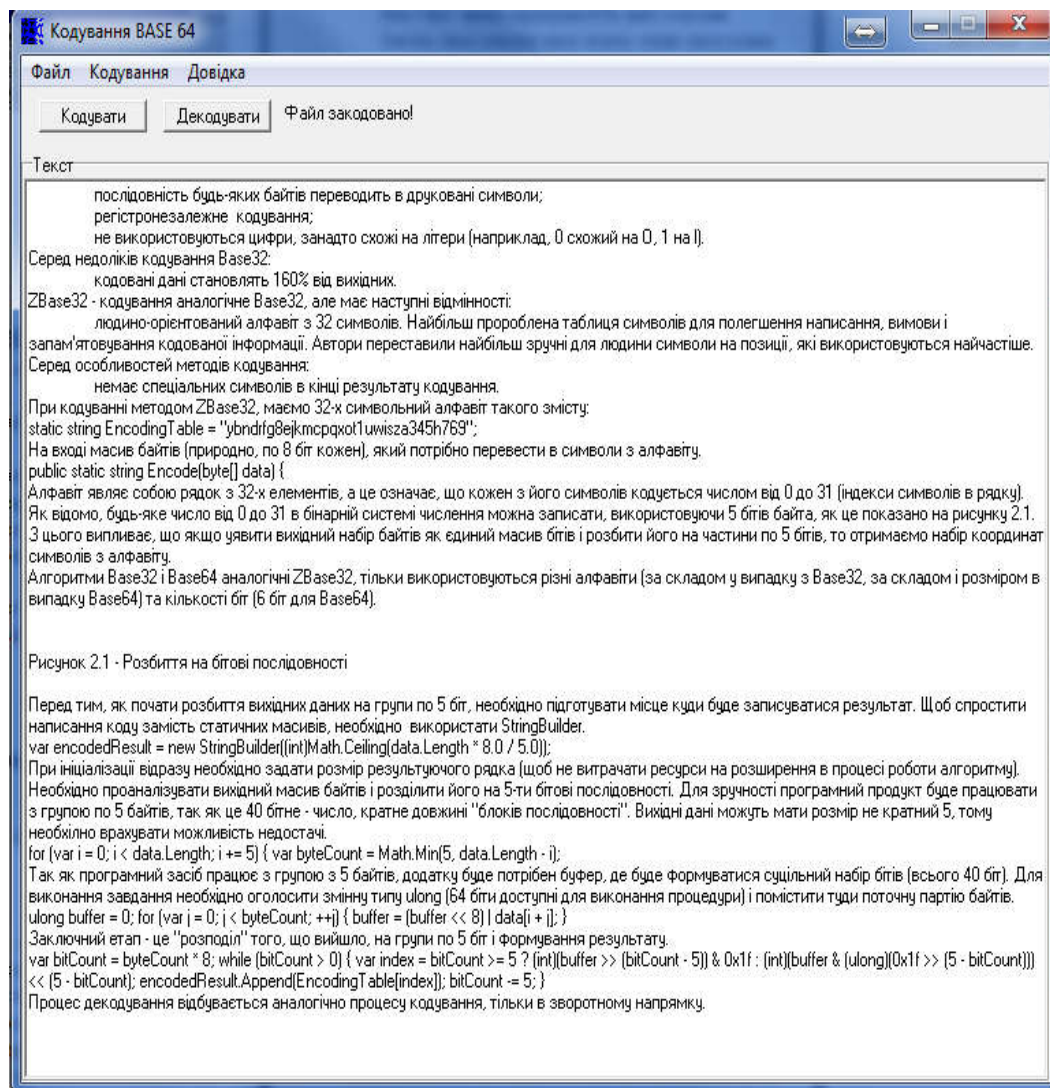


Рисунок 3.3 – Головне вікно додатку на етапі тестування

На рисунку 3.4 наведено результати перекодування тексту алгоритмом BASE64. Довідка про використання програмного продукту наведена в додатку Б.

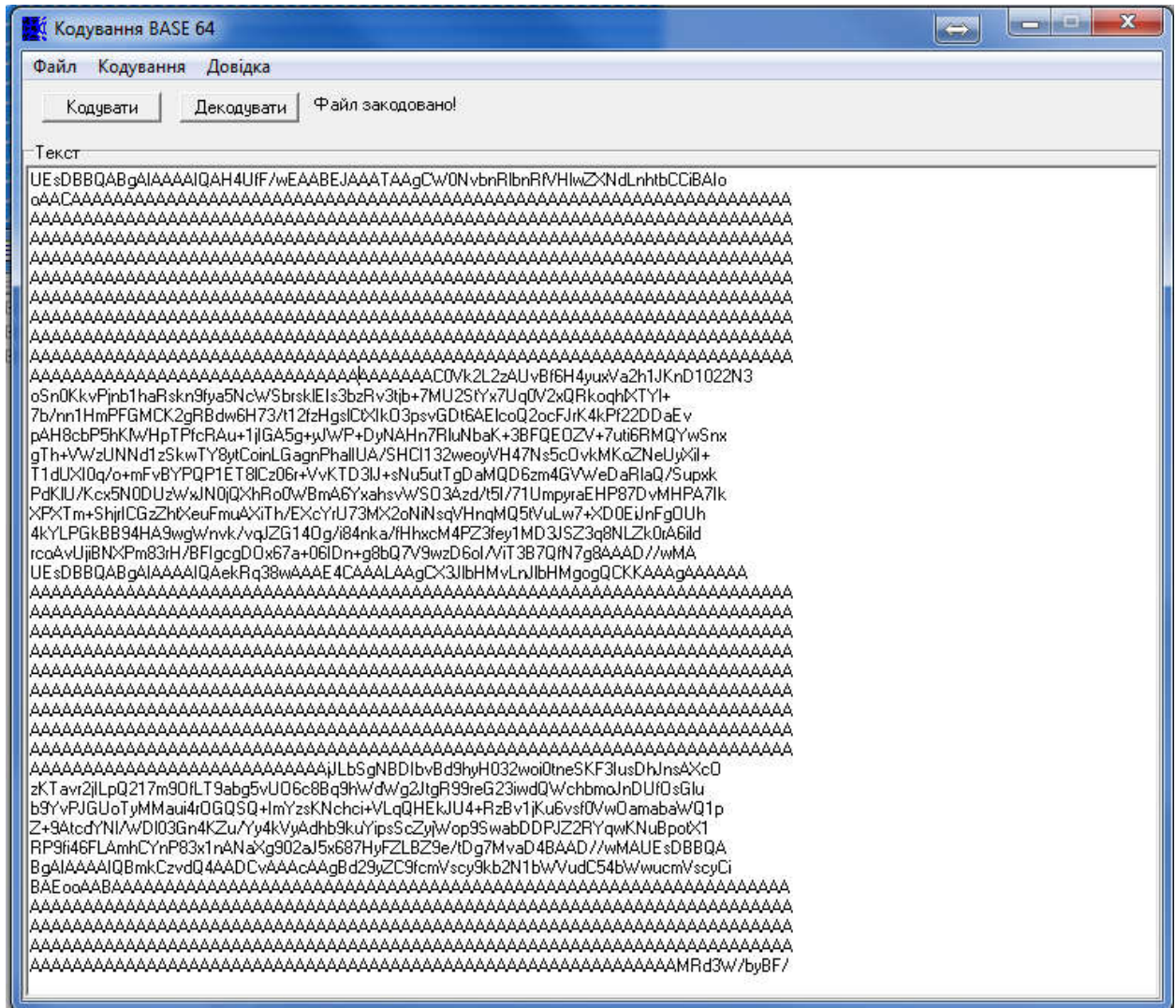


Рисунок 3.4 – Головне вікно додатку. Результат кодування BASE64.

Для цього використано такі техніки тестування, як пошук граничних значень та тестування, за допомогою класів еквівалентності. Результати тестування представлені в таблицях

Для виконання тестування внесемо тестовий приклад, який дозволить перевірити результати виконання основних функцій програмного засобу. Оберемо якийсь текст для перевірки результатів кодування.

Для тестового прикладу було обрано текст який є частиною тексту дипломного проекту. Проведемо перетворення згідно кодування BASE64.

						ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			47

Після вибору відповідного пункту меню або кнопки користувачеві буде запропоновано обрати файл в який необхідно помістити результати кодування.

Головне меню додатку складається з таких пунктів:

- файл ;
- кодування;
- довідка.

При виконанні тестування було використано наступне апаратне забезпечення: 15.6" (1920x1080) Full HD, матовий / Intel Core i5-7200U (2.5 - 3.1 ГГц) / RAM 8 ГБ / SSD 256 ГБ / Intel HD Graphics 620 / без ОД / LAN / Wi-Fi / Bluetooth / веб-камера.

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

4 ТЕХНІКО-ЕКОНОМІЧНИЙ РОЗДІЛ

Метою техніко – економічного розділу дипломної роботи є здійснення економічних розрахунків, спрямованих на визначення економічної ефективності програмного модуля розпізнавання та візуалізації математичних виразів довільної складності та прийняття рішення про його подальший розвиток і впровадження або ж недоцільність проведення відповідної розробки. Для проведення даного дослідження необхідно провести ряд розрахунків.

4.1 Розрахунок витрат на розробку програмного додатку

Витрати на розробку і впровадження програмного модуля кодування BASE64 (K) включають:

$$K = K_1 + K_2,$$

де K_1 - витрати на розробку апаратного та програмного забезпечення грн.;

K_2 - витрати на відлагодження і дослідну експлуатацію програми рішення задачі на комп'ютері, грн.

Витрати на розробку апаратних та програмних засобів включають:

- витрати на оплату праці розробників ($B_{оп}$);
- витрати на відрахування у спеціальні державні фонди ($B_{ф}$);
- витрати на матеріали та комплектуючі ($П_в$);
- накладні витрати (H);
- інші витрати ($I_в$);
- витрати на використання комп'ютерної техніки ($B_{КТ}$).

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

Розрахунок витрат на оплату праці.

Витрати на оплату праці включають заробітну плату (ЗП) всіх категорій працівників, безпосередньо зайнятих на всіх етапах проектування. Розмір ЗП обчислюється на основі трудоемності відповідних робіт у людино-днях та середньої ЗП відповідних категорій працівників.

У розробці проектного рішення задіяні наступні спеціалісти - розробники, а саме: керівник проекту; студент-дипломант; консультант техніко-економічного розділу (таблиця 4.1).

Таблиця 4.1 - Вихідні дані для розрахунку витрат на оплату праці

№п/п	Посада виконавців	Місячний оклад, грн.
1	Керівник ДП, викладач	5286
2	Консультант техніко-економічного розділу, доцент	6026
3	Студент	1100

Витрати на оплату праці розробників проекту визначаються за наступною формулою (4.1):

$$B_{OP} = \sum_{i=1}^N \sum_{j=1}^M n_{ij} \cdot t_{ij} \cdot C_{ij} , \quad (4.1)$$

де n_{ij} – чисельність розробників i -ої спеціальності j -го тарифного розряду, осіб;

t_{ij} – затрачений час на розробку проекту співробітником i -ої спеціальності j -го тарифного розряду, год;

C_{ij} – годинна ставка працівника i -ої спеціальності j -го тарифного розряду, грн.,

Середньо годинна ставка працівника може бути розрахована за такою формулою (4.2):

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

$$C_{ij} = \frac{C_{ij}^0(1+h)}{PЧ_i}, \quad (4.2)$$

де C_{ij} – основна місячна заробітна плата розробника i -ої спеціальності j -го тарифного розряду, грн.;

h – коефіцієнт, що визначає розмір додаткової заробітної плати (при умові наявності доплат);

$PЧ_i$ - місячний фонд робочого часу працівника i -ої спеціальності j -го тарифного розряду, год. (приймаємо 168 год.).

Коефіцієнт h , який визначає розмір додаткової заробітної плати, для керівника та консультанта техніко-економічного розділу дорівнює 0,47.

Результати розрахунку записують до таблиці 4.2.

Таблиця 4.2 - Розрахунок витрат на оплату праці

№ п/п	Посада виконавців	Час розробки, год	Погодинна заробітна плата, грн/год.	Витрати на розробку, грн
1	Керівник ДП, старший викладач	16	46,25	740
2	Консультант техніко-економічного розділу, доцент	2	88,6	177,2
3	Студент	144	6,55	943,2
Разом				1860,4

Відрахування на соціальні заходи. Величну відрахувань у спеціальні державні фонди визначають у відсотковому співвідношенні від суми основної та додаткової заробітних плат. Згідно діючого нормативного законодавства сума відрахувань у спеціальні державні фонди складає 20,5% від суми заробітної плати:

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

$$B_{\Phi} = \frac{20,5}{100} \cdot 1860,4 = 381,38 \text{ грн.}$$

Розрахунок витрат на матеріали та комплектуючі.

Загальна сума витрат на матеріальні ресурси (B_M) визначається за формулою (4.3):

$$B_M = \sum_{i=1}^n K_i \cdot C_i, \quad (4.3)$$

де K_i - витрата i -го типу матеріалу, натуральні одиниці вимірювання;

C_i - ціна за одиницю i -го типу матеріалу, грн.;

i - тип матеріального ресурсу;

n - кількість типів матеріальних ресурсів.

Таблиця 4.3 - Зведені розрахунки матеріальних витрат

№ п/п	Найменування матеріальних ресурсів	Од. виміру	Факт. витрачено матеріалів	Ціна за одиницю, грн.	Сума, грн	Транспортні витрати (10% від суми)	Загальна сума, грн	
	Допоміжна література	шт	1	600	600	60	660	
	Папір (формат А4)	уп	2	80	160	16	176	
	Ручка кулькова	шт	2	10	20	2	22	
	Олівець простий	шт	2	10	20	2	22	
	Диски CD-R	шт	2	15	30	3	33	
	Зошит, 96 арк	шт	1	50	50	5	55	
	Тонер для принтера	уп	1	90	90	9	99	
	Канцелярські маркери (червоний, зелений)	шт	2	20	40	4	44	
	Р а з о м							1111,00

Витрати на використання комп'ютерної техніки.

Витрати на використання комп'ютерної техніки (V_{KT}) включають витрати на амортизацію комп'ютерної техніки, витрати на користування програмним забезпеченням, витрати на електроенергію, що споживається комп'ютером. За даними обчислювального центру ТНЕУ для комп'ютера типу IBM PC/ATX вартість години роботи становить 6 грн. Середній щоденний час роботи на комп'ютері – 2 години. Розрахунок витрат на використання комп'ютерної техніки приведений в таблиці 4.4.

Таблиця 4.4- Розрахунок витрат на використання комп'ютерної техніки

№ п/п	Назва етапів робіт, при виконанні яких використовується комп'ютер	Час використання комп'ютера, год.	Витрати на використання комп'ютера грн.
1	Проведення досліджень та оформлення їх результатів	60	360
2	Оформлення техніко-економічного розділу	8	48
3	Оформлення ДП	12	72
Разом		80	480

Накладні витрати.

Накладні витрати проектних організацій включають три групи видатків: витрати на управління, загальногосподарські витрати, невиробничі витрати. Вони розраховуються за встановленими відсотками до витрат на оплату праці. Середньостатистичний відсоток накладних витрат приймемо 150% від заробітної плати:

$$H = 1,5 \cdot 1860,4 = 2790,6 \text{ (грн).}$$

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

Інші витрати.

Інші витрати є витратами, які не враховані в попередніх статтях. Вони становлять 10% від заробітної плати:

$$I_B = 1860,4 \cdot 0,1 = 186,04 \text{ (грн).}$$

Витрати на розробку програмного забезпечення складають:

$$K_1 = B_{OP} + B_{\Phi} + B_M + H + I_B + B_{KT},$$

$$K_1 = 1860,4 + 381,38 + 1111,00 + 2790,6 + 186,04 + 480,00 = 6809,42 \text{ (грн).}$$

Витрати на відлагодження і дослідну експлуатацію програмного продукту визначаємо за формулою (4.4):

$$K_2 = S_{m.g.} \cdot t_{vid} \quad (4.4)$$

де $S_{m.g.}$ - вартість однієї машино-години роботи ПК, грн./год;

t_{vid} - комп'ютерний час, витрачений на відлагодження і дослідну експлуатацію створеного програмного продукту, год.

Загальна кількість днів роботи на комп'ютері дорівнює 30 днів. Середній щоденний час роботи на комп'ютері – 2 години. Вартість години роботи комп'ютера дорівнює 6 грн., тому $K_2 = 6 \cdot 60 = 360$ грн.

4.2 Визначення експлуатаційних витрат

Для оцінки економічної ефективності розроблювальної системи моніторингу слід порівняти її з аналогом, тобто існуючим програмним забезпеченням ідентичного функціонального призначення.

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

Експлуатаційні одноразові витрати по програмному забезпеченню і аналогу включають вартість підготовки даних і вартість роботи комп'ютера (за час дії програми):

$$E_{\Pi} = E_{1\Pi} + E_{2\Pi},$$

де E_{Π} - одноразові експлуатаційні витрати на ПЗ (аналог), грн.;

$E_{1\Pi}$ - вартість підготовки даних для експлуатації ПЗ (аналог), грн.;

$E_{2\Pi}$ - вартість роботи комп'ютера для виконання проектного рішення (аналог), грн.

Річні експлуатаційні витрати $B_{E\Pi}$ визначаються за формулою:

$$B_{E\Pi} = E_{\Pi} * N_{\Pi},$$

де N_{Π} - періодичність експлуатації ПЗ (аналог), раз/рік.

Вартість підготовки даних для роботи на комп'ютері визначається за формулою:

$$E_{1\Pi} = \sum_{i=1}^n n_i t_i c_i,$$

де i - категорії працівників, які приймають участь у підготовці даних ($i=1,2,\dots,n$);

n_i - кількість працівників i -ої категорії, осіб.;

t_i - трудомісткість роботи співробітників i -ої категорії по підготовці даних, год.;

c_i - середнього годинна ставка працівника i -ої категорії з врахуванням додаткової заробітної плати, що знаходиться із співвідношення:

$$c_i = \frac{c_i^0(1+b)}{m},$$

де c_i^0 - основна місячна заробітна плата працівника i -ої категорії, грн.;

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

b - коефіцієнт, який враховує додаткову заробітну плату (прийmemo 0,57);

m - кількість робочих годин у місяці, год.

Для роботи з даними як для проектного рішення так і аналогу потрібен один працівник, основна місячна заробітна плата якого складає: $c = 3723$ грн.

Тоді:

$$c_1 = \frac{3723(1 + 0,57)}{22 * 8} = 33,21 \text{ грн/год}$$

Трудомісткість підготовки даних для проектного рішення складає 1 год., для аналога 1,5 год.

Таблиця 4.5- Розрахунок витрат на підготовку даних та реалізацію проектного рішення на комп'ютері

№	Час роботи співробітників, год.	Середньогодинна заробітна плата, грн./год.	Витрати, грн.
Проектне рішення			
1	1	33,21	33,21
Аналог			
1	1,5	33,21	66,42

Витрати на експлуатацію комп'ютера визначається за формулою:

$$E_{2П} = t * S_{МГ}$$

де t - витрати машинного часу для реалізації рішення (аналогу), год.;

$S_{МГ}$ - вартість однієї години роботи комп'ютера, грн./год.

$$E_{2П} = 1 * 6 = 6 \text{ грн.}; E_{2А} = 1,5 * 6 = 9 \text{ грн.}$$

$$E_{П} = 33,21 + 6 = 39,21 \text{ грн.}; E_{А} = 66,42 + 9 = 75,42 \text{ грн.}$$

$$B_{ЕП} = 39,21 * 252 = 9880,92 \text{ грн.}; B_{ЕА} = 75,42 * 252 = 19005,84 \text{ грн.}$$

Обчислення накладних витрат.

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства (фірми) та створення необхідних умов праці.

В залежності від організаційно-правової форми діяльності господарюючого суб'єкта, накладні витрати можуть становити 60–100 % від суми основної та додаткової заробітної плати працівників.

$$H_B = 0,7 * B_{OP}, \quad (4.7)$$

де H_B – накладні витрати.

$$H_B = 0,7 * 5845,11 = 4091,58 \text{ грн.}$$

Складання кошторису витрат та визначення собівартості.

Результати проведених розрахунків зведемо у таблицю 4.6.

Таблиця 4.6 - Кошторис витрат

№ п/п	Найменування витрат	Сума витрат, грн.
1	Витрати на оплату праці	1860,4
2	Відрахування у спеціальні державні фонди	381,38
3	Витрати на матеріали та комплектуючі	1111,00
4	Накладні витрати на розробку	2790,6
5	Інші витрати	186,04
6	Витрати на відлагодження і дослідну експлуатацію програмного продукту	360
7	Накладні витрати експлуатацію	4091,58
8	Річні експлуатаційні витрати	19005,84
Разом		29786,09

Розрахунок ціни проекту.

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

Договірна ціна (C_D) для проектних рішень розраховується за формулою (4.8):

$$C_D = B_{КС} \cdot \left(1 + \frac{p}{100}\right), \quad (4.8)$$

де $B_{КС}$ – кошторисна вартість, грн.;

p - середній рівень рентабельності, % (приймаємо 26% за погодженням з керівником).

$$C_D = 29786,09 \cdot (1 + 0,26) = 37530,47 \text{ грн.}$$

4.3 Визначення економічної ефективності і терміну окупності капітальних вкладень

Економічна ефективність (E_ϕ) полягає у відношенні результату виробництва до затрачених ресурсів:

$$E_\phi = \frac{П}{B_{КС}}, \quad (4.9)$$

де $П$ – прибуток, грн.;

$B_{КС}$ – кошторисна вартість, грн..

$$E_\phi = 7812,27 \text{ грн.} / 29786,09 \text{ грн.} = 0,25.$$

Поряд із економічною ефективністю розраховують термін окупності капітальних вкладень (Tr):

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		58

$$T_P = \frac{1}{E_P} . \quad (4.10)$$

Тобто: $T_P = 1/0,25 = 4р.$

Прийнятним вважається термін окупності близький до 7 років.

Розраховані економічні показники проекту занесемо до таблиці 4.7.

Таблиця 4.7 - Економічні показники розробки

№ п/п	Показник	Значення
1.	Собівартість, грн.	29786,09
2.	Плановий прибуток, грн.	7744,38в
3.	Ціна, грн.	37530,47
4.	Економічна ефективність	0,25
5.	Термін окупності, рік	4

Враховуючи основні економічні показники з таблиці 4.7, можна зробити висновок, що при економічній ефективності 0,25 та терміні окупності – 4 роки проводити роботи по впровадженню даного програмного модуля є доцільним та економічно вигідним.

ВИСНОВКИ

На основі аналізу сучасних методів проектування та розробки програмного забезпечення можна зробити наступні висновки:

1. Проведено аналіз підходів до кодування інформації та алгоритмів кодування, що дозволило виділити основні особливості методів кодування інформації.

2. Обґрунтовано вибір проектних рішень, що дозволило провести вибір найбільш оптимального середовища розробки та забезпечило ефективне використання розроблених алгоритмів та методів.

3. Досліджено алгоритм та схему кодування для BASE64, що дало можливість модифікувати та розробити алгоритм роботи програми.

4. Розроблено алгоритм кодування та декодування інформації BASE64 на основі структурного підходу та системи спеціалізованих позначень, що дозволило реалізувати програмний засіб.

5. Спроектовано зручний графічний інтерфейс та структуру програмного забезпечення, на основі модульного підходу, що дозволило провести попереднє дослідження запропонованого рішення.

6. Реалізовано програмний модуль кодування алгоритмом BASE64 та проведено тестування розробленого програмного забезпечення для кодування тексту та формул.

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		60

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Баженова И.Ю. Язык программирования Delphi// И.Ю. Баженова/- АО "Диалог-МИФИ", 1997. – 366с.
2. Бартлетт Н. Программирование на Delphi Путеводитель // Н. Бартлетт А. Лесли, С. Симкин /.- The Cogniolis Group,Inc.,1996, Издательство НИПФ "ДиаСофт Лтд.",1996. – 116с.
3. Вебер Дж. Технология С++в подлиннике // Дж. Вебер /- QUE Corporation, 1996, "ВНУ-Санкт-Петербург",1997. – 256с.
4. Волш А. И. Основы программирования на С++для World Wide Web // А. И. Волш /- IDG Books Worldwide,Inc.,1996, - Издательство "Диалектика",1996. – 458с.
5. Марков А. С. «Базы данных. Введение в теорию и методологию // А. С. Марков, К.Ю. Лисовский / - Финансы и статистика»-2006,-Р. 24-35.
6. Абрамов С. А. Задачи по программированию // С. А. Абрамов, Г. Г. Гнездилова, Е. Н. Капустина, М. И. Селюн/ — М.: Наука, 1988. – 256с.
7. Березин Б.И., Начальный курс Delphi // Б.И.Березин, С.Б.Березин / — М.: ДИАЛОГ-МИФИ, 1996. – 331с.
8. Бондарев В.М. Основы программирования // В.М. Бондарев, В.И. Рублинецкий, Е.Г. Качко / — Харьков: Фолио, Ростов н/Д: Феникс, 1997. - 446с.
9. Вирт Н. Алгоритмы и структуры данных. // Н. Вирт /— М.: Мир, 1989. - 345с.
- 10.Гладков В. П. Задачи по информатике на вступительном экзамене в вуз и их решения: Учебное пособие // В. П. Гладков / — Пермь: Перм. техн. ун-т, 1994. – 516с.
- 11.Грогоно П. Программирование на языке Delphi // П. Грогоно / —М.: Мир, 1982. – 216с.
- 12.Дагене В.А. 100 задач по программированию // В.А.Дагене, Г.К. Григас, К.Ф. Аугутис /— М.: Просвещение, 1993. – 106с.

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

13. Джамса К. Библиотека программиста Java // К.Джамса /- Jamsa Press, 1996, ООО "Попурри", 1996. – 656с.

14. Марков А. С. «Базы данных. Введение в теорию и методологию // А. С. Марков, К.Ю. Лисовский / - Финансы и статистика»-2006,-Р. 24-35.

15. Заварыкин В.М. Основы информатики и вычислительной техники // В.М. Заварыкин, В.Г. Житомирский, М.П. Лапчик / — М.: Просвещение, 1989. – 556с.

16. Касаткин В. Н. Информация. Алгоритмы. ЭВМ // В. Н. Касаткин / — М.: Просвещение, 1991. – 219с.

17. Кен А. Язык программирования Delphi // А. Кен, Дж. Гослинг /.- Addison-Wesley Longman, U.S.A., 1996, Издательство "Питер-Пресс", 1997. - 378с.

18. Керниган Б. Язык программирования Delphi // Б. Керниган, Д. Ритчи / Пер. с англ. — М.: Финансы и статистика, 1992. – 391с.

19. Ляхович В.Ф. Руководство к решению задач по основам информатики и вычислительной техники // В.Ф. Ляхович/ — М.: Высшая школа, 1994. – 127с.

20. Мейнджер Дж. Delphi Основы программирования // Дж. Мейнджер /- McGraw-Hill, Inc., 1996, Издательская группа BNV, Киев, 1997. – 346с.

21. Миков А. И. Информатика. Введение в компьютерные науки // А. И. Миков / — Пермь: Изд-во ПГУ, 1998. – 442с.

22. Могилев А. В. Информатика: Учеб. пособие для студ. пед. вузов // А. В. Могилев, Е. К. Хеннера./ — М.: Изд. центр «Академия», 1999. – 629с.

23. Нотон П. JAVA:Справ.руководство// П.Нотон, А.Тихонова.- М.:БИНОМ:Восточ.Кн.Компания,1996:Восточ.Кн.Компания. - 447с.

24. Нотон П. Полный справочник по Java // П. Нотон, Г. Шилдт//.- McGraw-Hill, 1997, Издательство "Диалектика", 1997. – 556с.

25. Ренеган Э.Дж. 1001 адрес WEB для программистов :Новейший путеводитель программиста по ресурсам World Wide Web:Пер.// Э.Дж. Ренеган /с англ..-Минск:Попурри,1997.-512с.ил.

26. Родли Дж. Создание Java-апплетов // Дж. Родли / - The Coriolis Group, Inc., 1996, Издательство НИПФ "ДиаСофт Лтд.", 1996. – 466с.

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

27. Секреты программирования для Internet на Java // М.Томас, П. Пратик, А. Хадсон, Д. Болл/ - Ventana Press, Ventana Communications Group, U.S.A., 1996, Издательство "Питер Пресс", 1997. – 396с.

28. Семакина И. Г. Информатика. Задачник-практикум: В 2 т. // И. Г. Семакина, Е.К.Хеннера/ — М.: Лаборатория Базовых Знаний, 1999. – 476с.

29. Сокольский М.В. Все об Intranet и Internet // М.В. Сокольский / - М.:Элиот, 1998. - 254с.ил.

30. Тассел Д. Стил, разработка, эффективность, отладка и испытание программ // Д. Тассел / — М.: Мир, 1981. – 56с.

31. Тюрин Ю.Н. Анализ данных на компьютере./ Ю.Н. Тюрин, А.А. Макаров, В.Э.Фигурнова. - М.: ИНФРА-М, Финансы и статистика, 1995. - 384с.

32. Флэнэген Д. Java in a Nutshell / Д. Флэнэген /- O'Reilly & Associates, Inc., 1997, Издательская группа BHV, Киев, 1998. – 473с.

33. Чен М.С. Программирование на C++:1001 совет:Наиболее полное руководство по Java и Visual J++ :Пер.с англ.// М.С.Чен, С.В. Грифис, Э.Ф. Изи./ -Минск:Попурри,1997.- 640с.ил.+ Прил.(1диск.).

34. Эферган М. C++: справочник // М. Эферган /.- QUE Corporation, 1997, Издательство "Питер Ком", 1998. – 256с.

35. Методичні рекомендації до виконання дипломного проекту з освітньо-кваліфікаційного рівня “Бакалавр” напряму підготовки 6.050102 «Комп’ютерна інженерія» фахового спрямування «Комп’ютерні системи та мережі» / О.М. Березький, Л.О.Дубчак, Р.Б. Трембач, Г.М. Мельник, Ю.М. Батько, С.В. Івасьєв / Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2014.–65 с.

36. Методичні вказівки до написання техніко-економічного розділу дипломних проектів освітньо-кваліфікаційного рівня «бакалавр» напряму підготовки 6.050102 комп’ютерна інженерія/ І.Р. Паздрій – Тернопіль: ТНЕУ, 2014. – 37 с.

					ДП.КСМ.07125/14.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63