**Черкаський національний університет імені Богдана Хмельницького**

**Кафедра Інформаційних технологій**

**КУРСОВА РОБОТА**

***з дисципліни***

***«Програмування та алгоритмічні мови»***

***НА ТЕМУ «Програмний додаток розрахунку з персоналом. Нарахування та виплата відпускних»***

Студентки 2 курсу, групи КН-22

напряму підготовки «Інформаційні

технології»

спеціальності «Комп’ютерні науки»

Бабченко В.О. \_

**(прізвище та ініціали)**

Керівник викладач Авраменко А.С.

**(посада, вчене звання, прізвище та ініціали)**

Національна шкала: \_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_ Оцінка: ECIS\_\_\_

Члени комісії:

\_\_\_\_\_\_\_\_ Гребенович Ю.Є.\_

**(підпис) (прізвище та ініціали)**

\_\_\_\_\_\_\_\_ Любченко К. М.\_

**(підпис) (прізвище та ініціали)**

\_\_\_\_\_\_\_\_ Царик Т. Ю.\_\_\_\_

**(підпис) (прізвище та ініціали)**

**м. Черкаси – 2023 рік**

**МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ ЧЕРКАСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ім. Б. ХМЕЛЬНИЦЬКОГО**

**Кафедра інформаційних технологій**

ЗАТВЕРДЖУЮ

Завідувач кафедри,

Веретельник В. В.\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис)

«\_\_\_» \_\_\_\_\_\_\_\_ 2023 р.

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ**

**на курсову роботу**

студентки Бабченко Вікторії Олегівни\_\_\_\_\_\_\_\_\_\_ групи КН-22\_ другого курсу

**ТЕМА** Програмний додаток розрахунку з персоналом. Нарахування та виплата відпускних

**Вихідні дані до курсової роботи:** Розроблений програмний продукт повинен: \_

1. Реалізовувати графічний інтерфейс (GUI) за допомогою вікна Windows Form з написами для зручності користування, текстовими полями, списком, вибором дат та кнопкою для взаємодії користувача. Також при виборі працівника із списку повинна відображатися відповідна інформація про нього.

2. Вичитувати дані про працівників (імена, посади, місячні заробітні плати) з Excel файлу та на основі цих даних обраховувати відпускні. Сума виплати розраховується як добуток середньоденної заробітної плати на кількість відпусткових днів, які вираховуються на основі вибраних користувачем дат початку та закінчення відпустки.

3. Містити перевірки та повідомлення користувачеві для випадків, коли вибрані дати недійсні або коли кількість відпусткових днів перевищує певне обмеження.

Зміст **Пояснювальної записки** до курсової роботи:

Індивідуальне завдання

Вступ

1. Огляд…

2. Розробка схеми алгоритму…

3. Розробка програми…

Висновки

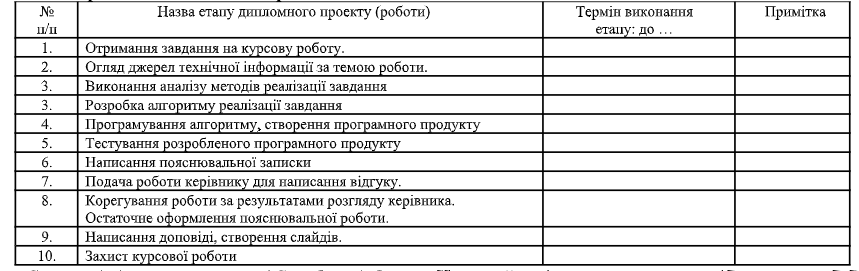
Список літератури

Додатки (за необхідності)

Перелік наочного матеріалу:

­­ *Програмний продукт…, пояснювальна записка, презентація результатів роботи*.\_\_\_

**Календарний план виконання роботи:**



Студент(ка) \_\_\_\_\_ / Бабченко В.О. Науковий керівник \_\_\_\_\_ / Авраменко А.С.

(підпис) (Прізвище, ініціали) (підпис) (Прізвище, ініціали)

Завдання видане « » « » 2023 р.

**Зміст**

**Вступ ........................................................................................................................**1

**Розділ 1. Огляд та аналіз алгоритмів стиснення..............................................**2

**1.1 Описання встановленого завдання............................................................**2

**1.2 Методи стиснення інформації.....................................................................**2

**1.2.1 Алгоритми групи KWE ........................................................................**3

**1.2.2 Алгоритм Lossless JPEG .......................................................................**4

**1.2.3 Алгоритм Гаффмана..............................................................................**4

**1.3 Висновок до першого розділу ....................................................................**5

**Розділ 2. Розробка програми ..............................................................................**6

**2.1 Блок-схема ..................................................................................................**6

**2.2 Огляд складових компонентів...................................................................**7

**2.3 Розгляд методів програми .........................................................................**8

**2.4 Висновок до другого розділу ...................................................................**12

**Розділ 3. Реалізація архіватора файлів ...........................................................**13

**3.1 Переваги алгоритму Гаффмана .............................................................**13

**3.2 Реалізація методів кодування та декодування ....................................**14

**3.3 Висновок до третього розділу ................................................................**21

**Загальний висновок ...........................................................................................**22

**Використані джерела ........................................................................................**23

**Вступ**

Завдання полягає у розробці програмного додатку розрахунку з персоналом, нарахуванні та виплаті відпускних. Курсова робота написана на мові програмування С#. Графічний інтерфейс реалізовано за допомогою засобів Windows Forms.

У сучасному світі інформаційних технологій важливість автоматизації бізнес-процесів та оптимізації управління персоналом є надзвичайно актуальним. Зокрема, розробка програмних продуктів, спрямованих на ефективний розрахунок та виплату відпускних, є важливою для підприємств, які прагнуть не лише полегшити та прискорити рутинні процеси, але й уникнути можливих помилок та забезпечити високий рівень точності обчислень.

**Актуальність додатку:** з урахуванням сучасних тенденцій у розвитку бізнесу та його інформаційного забезпечення, стає очевидним, що відпускні — важлива частина корпоративного управління та HR-процесів. Інтеграція програмного продукту для розрахунку відпусткових у щоденну діяльність підприємства забезпечить не лише ефективність у внутрішніх операціях, але й сприятиме підвищенню задоволеності працівників та виробничої продуктивності.

**Тема:** програмний додаток розрахунку з персоналом. Нарахування та виплата відпускних.

**Мета курсової роботи:** розробка програмного додатку для розрахунку та виплати відпускних працівникам підприємства.

**Задача курсової роботи**:

1. аналіз та вивчення процесів розрахунку відпускних
2. розробка графічного інтерфейсу для взаємодії з користувачем
3. вибір та вивчення засобів для роботи з даними у форматі Excel
4. імплементація алгоритмів розрахунку та виплати відпускних
5. забезпечення користувацької зручності та точності введених даних.

**Об’єкт:** процеси розрахунку та виплати відпусткових на підприємстві.

**Предмет роботи:** програмний продукт, призначений для автоматизації процесів розрахунку та виплати відпускних та забезпечення ефективного управління відпустками працівників.

Відпускні — це період відпочинку, на який працівник має право у робочому році. Зазвичай виражається у кількості днів, які працівник може взяти для відпочинку без втрати заробітної плати.

Середньоденна зарплата визначає, скільки грошей працівник отримує в середньому за кожен робочий день. Це обчислюється як місячна зарплата, поділена на кількість робочих днів у місяці. Цей показник є ключовим у визначенні вартості праці протягом коротших періодів, таких як відпустка.

Сума виплати відпусткових — це грошова сума, яку працівник отримує під час відпустки. Розраховується як добуток середньоденної зарплати на кількість відпусткових днів. Цей показник є ключовим для розрахунку витрат підприємства на відпустки своїх працівників.

З урахуванням швидкого розвитку технологій та конкурентного бізнес-середовища, ефективне управління персоналом є важливою складовою успіху будь-якого підприємства. Автоматизація процесів розрахунку та виплати відпусткових сприяє швидкості та точності цих операцій, полегшуючи роботу відділу кадрів та зменшуючи ймовірність помилок. Систематичне та точне нарахування відпусткових важливо для створення сприятливого робочого середовища. Професіонально налаштована система виплати відпусткових сприяє задоволенню працівників та позитивному відношенню до компанії. Автоматизація розрахунків дозволяє уникнути людських помилок та забезпечити точність обчислень. Це, в свою чергу, зменшує ризик неправильних виплат та можливих претензій з боку працівників. Крім того, це оптимізує внутрішні процеси та зменшує витрати на ведення обліку.

Сучасні тенденції в управлінні персоналом вимагають від підприємств ефективних та технологічних рішень. Розробка програмного продукту для розрахунку відпусткових відповідає вимогам сучасного бізнесу та сприяє адаптації до високих стандартів корпоративного управління. Тож, розробка та впровадження програмного продукту для розрахунку та виплати відпусткових є дійсно важливим та актуальним питанням для підприємств.

**Розділ 1. Огляд алгоритмів та методів реалізації програмного додатку нарахування та виплати відпускних.**

**1.1 Описання встановленого завдання**

Поставлені задачі:

1. Реалізація механізму зчитування даних про працівників (імена, посади, місячні заробітні плати) з Excel файлу.
2. Реалізація графічного інтерфейсу з елементами керування для вибору працівника, введення дат відпустки та відображення інформації про обраного працівника та результатів розрахунку.
3. Алгоритм розрахунку кількості робочих днів між обраними датами відпустки.
4. Розрахунок середньоденної заробітної плати та відпусткових виплат на основі обраних дат та інформації про працівника.
5. Реалізація перевірок коректності введених дат відпустки. Відображення повідомлень у випадку помилок та невірно введених даних.
6. Проектування схеми алгоритмів.
7. Реалізація програмного продукту.
8. Тестування готового програмного продукту.

Цільова мета проекту полягає в розробці програмного забезпечення, яке надає підтримку для розрахунку та виплати відпусток працівникам підприємства. Проект розглядається як інструмент для автоматизації управління відпустками, забезпечуючи зручний та точний розрахунок фінансових аспектів відпусткової політики підприємства. Реалізація програми має забезпечити зручність користування, високу точність обчислень та можливість легкої адаптації до конкретних потреб користувача.

**1.2 Огляд існуючих рішень**

Перед початком розробки важливо вивчити існуючі рішення та врахувати їхні переваги та недоліки. У сфері управління персоналом і відпусток існує декілька програм, проте багато з них може бути складними для використання, не містити необхідного функціоналу або бути недостатньо гнучкими для адаптації під конкретні потреби підприємства.

**1.3 Вибір алгоритмів та методів**

**1.3.1 Мова програмування**

Для реалізації програмного продукту обрана мова програмування C#, оскільки вона є високорівневою, об'єктно-орієнтованою та добре підходить для створення Windows-застосунків. C# також володіє розширеними можливостями для роботи зі зчитуванням та обробкою даних з Excel файлів.

**1.3.2 Графічний інтерфейс**

Для забезпечення зручності користувача обрано технологію Windows Forms. Це дозволить створити інтуїтивно зрозумілий та легкий у використанні інтерфейс, який надасть можливість користувачам швидко та ефективно взаємодіяти з програмою.

**1.3.3 Алгоритм розрахунку відпускних**

Алгоритм обчислення відпускних є простим та ефективним, базуючись на розрахунку середньоденної заробітної плати та кількості відпусткових днів. Кількість відпускних днів вираховується між обраними користувачем датами початку та закінчення відпустки. Середньоденна зарплата обчислюється на основі зчитаних даних про заробітну плату обраного працівника з Excel файлу. На основі цих даних обчислюється сума відпускних.

**1.4 Розробка графічного інтерфейсу**

Розробка графічного інтерфейсу буде виконана з використанням засобів Windows Forms. Головна форма буде містити випадаючий список зі списком працівників, текстові поля для виведення інформації про працівника та розраховані відпусткові, а також елементи для вибору та введення дат відпустки

**1.5 Висновки до першого розділу**

Огляд алгоритмів та методів реалізації дозволив визначити ключові елементи для успішної розробки програмного проекту. Вибір мови програмування, графічного інтерфейсу та алгоритмів розрахунку відпусток зроблено з урахуванням вимог ефективності, зручності використання та гнучкості налаштувань.

**Розділ 2. Опис Алгоритмів та Структура Програми**

**2.1 Опис Алгоритмів**

**2.1.1 Вичитування Даних**

Алгоритм вичитування даних із Excel файлу використовує бібліотеку ExcelDataReader. Програма відкриває файл, створює читача та зчитує дані у форматі DataSet. Далі, дані зберігаються в DataTable, який використовується для відображення списку працівників у вікні.

**2.1.2 Обчислення Відпускних**

Основний алгоритм для обчислення відпускних включає наступні кроки:

1. Вибір працівника зі списку та отримання його місячної заробітної плати.
2. Обчислення середньоденної заробітної плати на основі місячної зарплати з урахування обмежень (не може перевищувати 3301.58 грн).
3. Вибір дат початку та закінчення відпустки та обчислення кількості відпусткових днів.
4. Перевірка на коректність введених дат та кількості відпусткових днів.
5. Розрахунок загальної суми відпусткової виплати.

**2.1.3 Аналіз Дат та Відпусткових Днів**

Алгоритм аналізу дат та відпусткових днів перевіряє, чи дата початку не є пізнішою або співпадає з датою кінця, а також чи кількість відпусткових днів не перевищує обмеження в 24 дні.

**2.2 Блок-Схеми основних функцій**

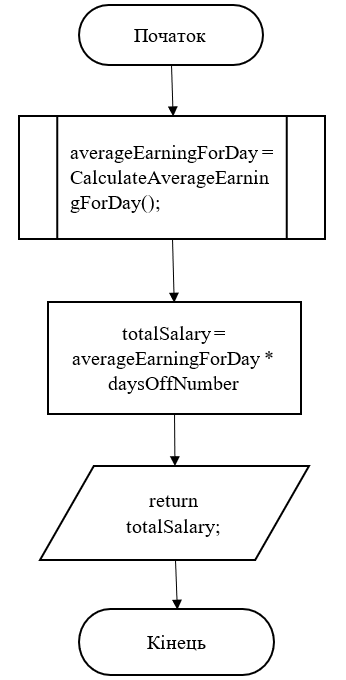
****

Рис. 2.1. Блок-схема функції обчислення відпускних

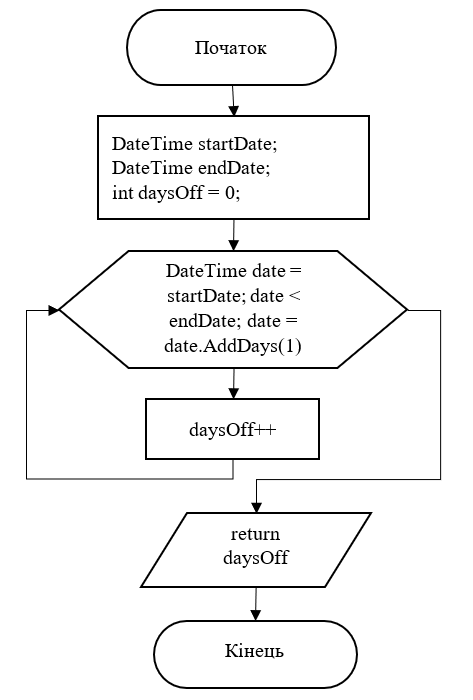


Рис. 2.2. Блок-схема функції getDaysOffNumber(DateTime startDate, DateTime endDate)

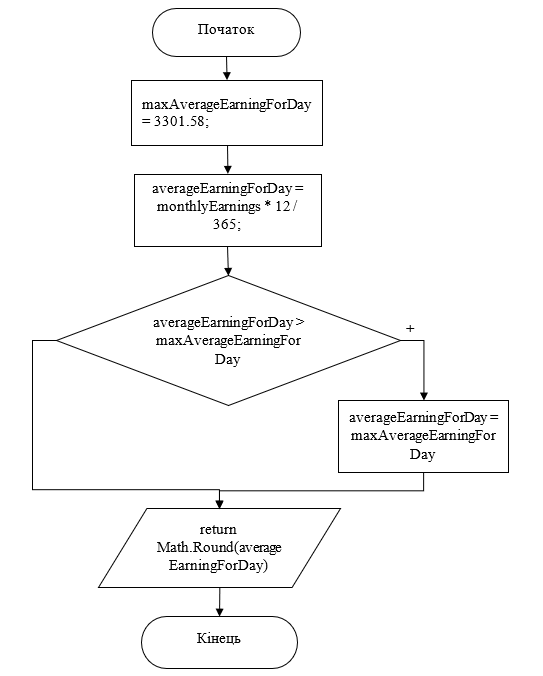


Рис. 2.3. Блок-схема функції CalculateAverageEarningForDay()

**2.3 Узагальнена Структурна Схема Програмного Продукту**

Main Program

|

|---- LoadEmployeeList()

| |

| |---- Відкриття Excel файлу

| |

| |---- Зчитування даних за допомогою ExcelDataReader

| |

| |---- Заповнення DataTable інформацією про робітника

| |

| |---- Показ списку робітників у графічному інтерфейсі

|

|---- CalculateTotalSalary()

| |

| |---- CalculateAverageEarningForDay()

| | |

| | |---- Обрахування середньоденної зарплати: місячна зарплата \* 12 / 365

| | |

| | |---- Перевірення валідністі середньоденної зарплати

| |

| |---- getDaysOffNumber()

| |

| |---- Проходження циклом між датами та визначення кількості відпускних днів

|

|---- UpdateDaysOffNumber()

| |

| |---- Перевірення дати початку та завершення

| |

| |---- getDaysOffNumber()

|

|---- listBox1\_SelectedIndexChanged()

| |

| |---- Отримання інформацію про вибраного працівника

| |

| |---- Відображення інформації про робітника в GUI

|

|---- startDateTimePicker1\_ValueChanged()

| |

| |---- Очищення TextBox4

| |

| |---- UpdateDaysOffNumber()

|

|---- endDateTimePicker2\_ValueChanged()

| |

| |---- Очищення TextBox4

| |

| |---- UpdateDaysOffNumber()

|

|---- button1\_Click()

|

|---- getDaysOffNumber()

|

|---- CalculateTotalSalary()

|

|---- Відображення загальної зарплати в TextBox4

**2.4 Висновки**

Розділ 2 надає детальний опис алгоритмів, що використовуються для реалізації функціоналу програмного додатку. Блок-схеми основного та допоміжного алгоритмів допомагають візуалізувати послідовність операцій.

Структурна схема продукту показує логічну структуру програми та взаємодію між її основними компонентами. Усі ці елементи спрямовані на виконання головної задачі — розрахунок відпускних виплат для працівників на основі введених та отриманих даних з файлу.

**Розділ 3. Реалізація архіватора файлів**

**3.1 Переваги алгоритму Гаффмана**

Трохи повернемось до історії створення методу Гаффмана. Як нам вже відомо, алгоритм вигадав Девід А. Гаффман у 1951 році. Йому та його однокурсникам дали вибір – або курсова робота, або випускний екзамен. Професор Роберт М. Фано задав студентам курсову роботу з проблемою пошуку найбільш ефективного двійкового коду. Гаффман спочатку хотів відмовитись, адже ніяк не міг довести найефективніший з якихось кодів, але майже в останню мить йому прийшла ідея. Вона заклечалась у наступному - використання двійкового дерева з сортуванням по частоті. Й після цього було доведено, що даний метод є найбільш ефективним.

При цьому, Девід Гаффман перевершив свого професора Фано, який працював над розробкою кодування з Клодом Шенноном. Його побудова дерева знизу вгору гарантувала оптимальність, на відміну від кодування Шеннона-Фано.

**3.2 Реалізація методів кодування та декодування**

Після того, як ми створили в класі Huffman метод CompressFile й у ньому викликали метод CompressBytes, нам тепер потрібно його створити. Він буде містити в собі наступне: частотний словник, дерево, масив з кодами (якому ми передаємо посилання на елемент – корінь), заголовок та безпосередньо саме стиснення (Рис. 3.1).

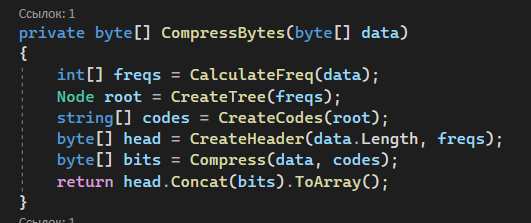


Рис. 3.1. Метод CompressBytes

Тепер ми створюємо метод CalculateFreq, який буде виконувати функцію словника й перебирати байти в масиві та рахувати їх (Рис. 3.2). Також у ньому буде ще один метод, який нормалізує частотну таблицю (щоб максимальне значення було 255).

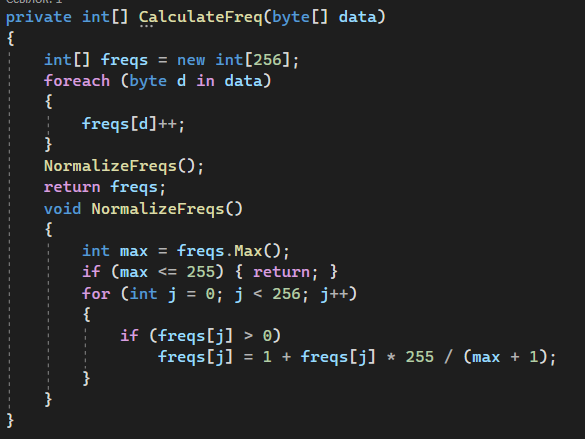


Рис. 3.2. Метод CalculateFreq

Наступний крок – це створення дерева. Для цього ми створюємо метод CreateTree, у якому записуємо екземпляр класу PriorityQueue й потім у циклі for перебираємо всі елементи масиву. Потім у циклі while зображуємо процес створення нових вузлів (Рис. 3.3).

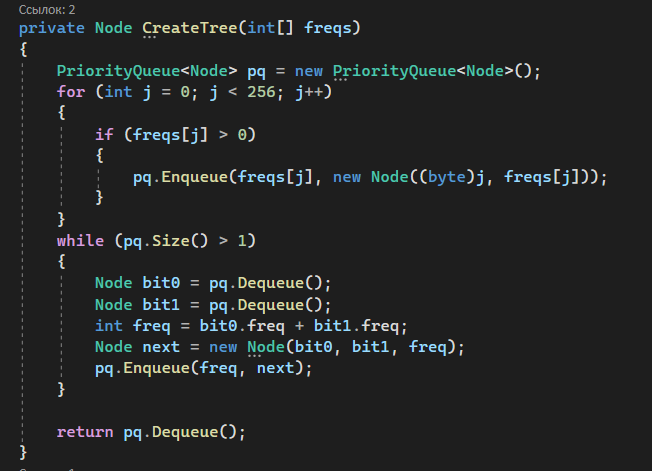


Рис. 3.3. Метод CreateTree (дерево)

Далі ми створюємо метод CreateCodes, у якому буде вкладена функція Next. Вона приймає вузол й накопичує код, але якщо нульовий біт дорівнює null, то записується код, якщо ні – рекурсивно йдемо по нульовому і одиничному бітах (Рис. 3.4).

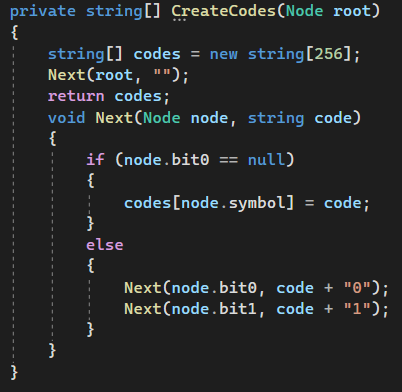


Рис. 3.4 Метод CreateCodes

Також нам потрібно створити метод CreateHeader, який буде виконувати функції таблиці (Рис. 3.5).

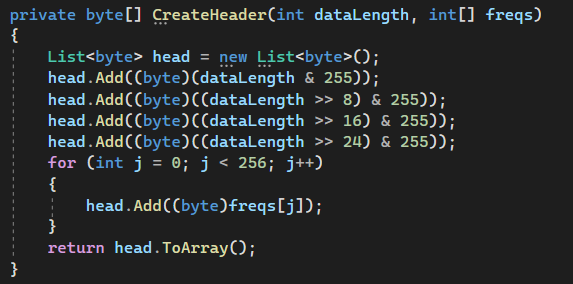


Рис. 3.5. Метод CreateHeader

Для завершення кодування ми створюємо метод Compress, який стискає файл з вхідними даними і кодами. Він містить список байтів, змінні, які накопичують суми чергового байта. Також він має два foreach, перший спочатку перебирає всі символи, другий потім по символу дивиться код. Якщо після цього ще щось залишилось – його ми додаємо у список, потім все це конвертуємо у масив і повертаємо (Рис. 3.6).

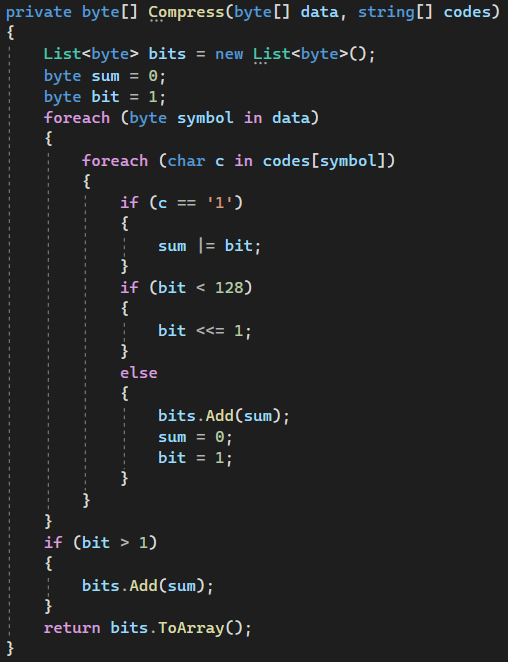


Рис. 3.6. Метод Compress

Тепер розглянемо декодування. Його принцип дуже схожий на метод CompressFile, але замість файлу, ми вже будемо зчитувати архів й його розтискати. Спершу створюємо метод DecompressFile (Рис. 3.7).

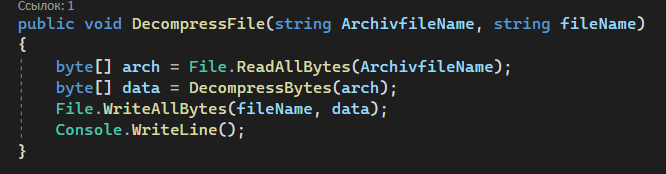


Рис. 3.7. Метод DecompressFile

Створюємо метод DecompressBytes, який буде виконувати функцію декомпресії (Рис. 3.8).

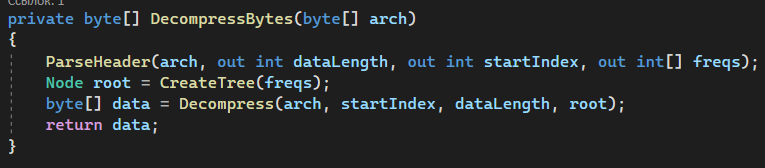


Рис.3.8. Метод DecompressBytes

Тепер нам потрібно створити метод ParseHeader, який зчитує наш заголовок (Рис. 3.9).

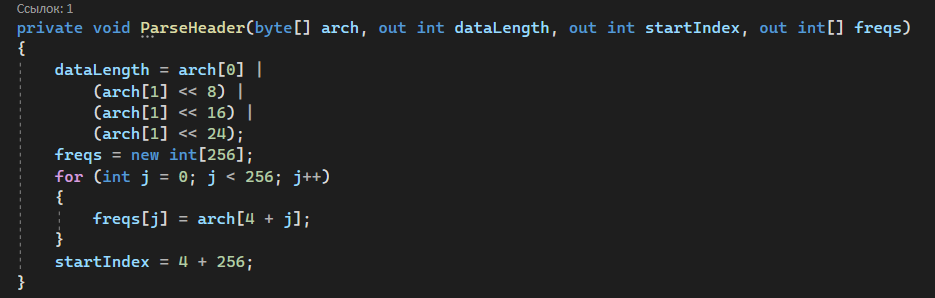


Рис. 3.9. Метод ParseHeader

І останнє, що нам залишається – це написати метод Decompress, який і повертає наш закодований файл у первинний вигляд (Рис. 3.10).

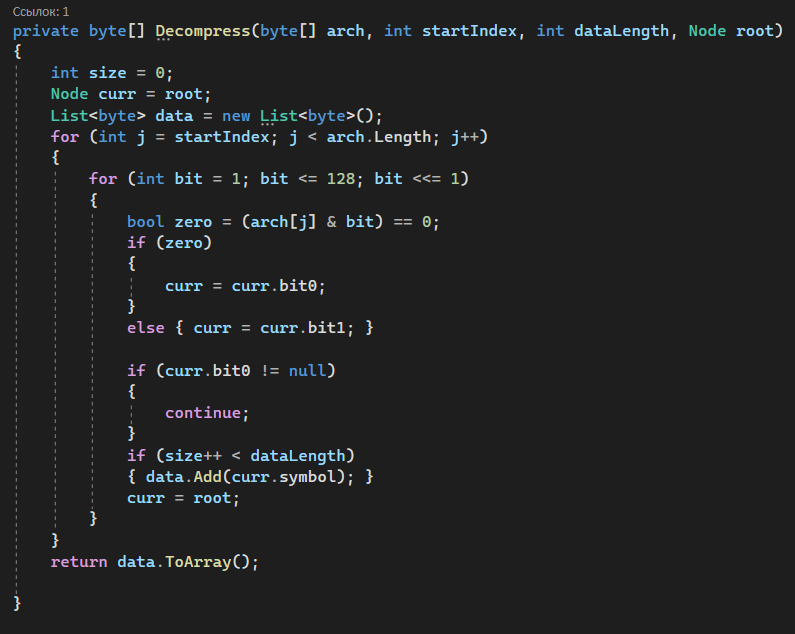


Рис. 3.10. Метод Decompress

У методі Main створюємо екземпляр класу Huffman й за допомогою нього викликаємо методи стиснення та розтискання (Рис. 3.11).

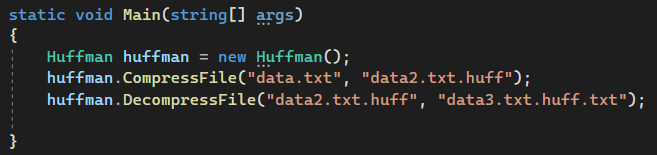


Рис. 3.11. Метод Main

Архіватор на основі коду Гаффмана готовий. Тепер можна відобразити його роботу вже з текстовим файлом. У нас є текстовий файл з розміром 1696 байт (Рис. 3.12).

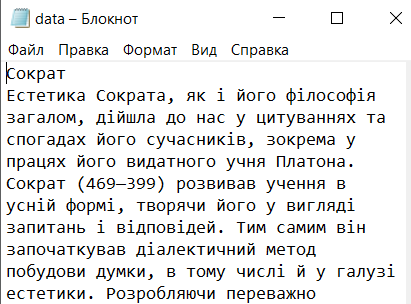


Рис. 3.12. Вхідний файл data

Після запуску програми створилось ще два файли. Один – закодований файл з розміром 1132 байти (Рис. 3.13), другий – декодований файл з розміром 1696 байт (має вигляд вхідного файлу) (Рис. 3.14).

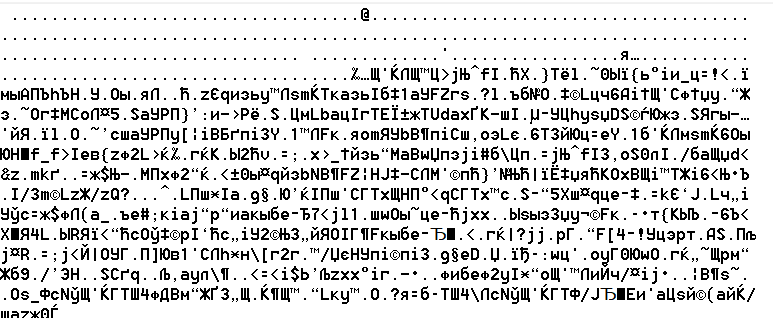


Рис. 3.13. Закодований файл data.txt.huff (двійковий вигляд)

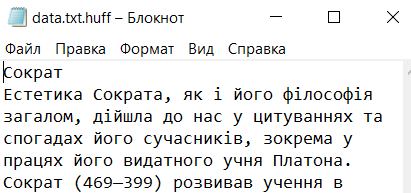


Рис. 3.14. Декодований файл data.txt.huff

Загальний вигляд розмірності файлів (Рис. 3.15).

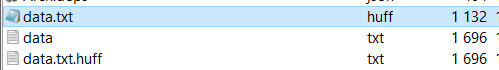


Рис. 3.15. Розміри файлів

**3.3 Висновок до третього розділу**

У даному розділі було більш детально розглянуто перевагу даного методу кодування файлів.

Потім було розписано про методи, які потрібні для стиснення файлу, у якій послідовності вони працюють. Також описано методи, які використовуються для декодування файлів.

Також, було показано як саме відбувається стиснення та розтискання файлу, що дійсно закодований файл має менший розмір, й зазначено те, що він має знаходитись безпосередньо у папці, що зберігає файл програмного продукту, тобто файл з кодом програми.

**Загальний висновок**

Під час виконання даної курсової роботи я дізналась велику кількість нової інформації про принципи роботи архіваторів, як саме відбувається процес стиснення інформації.

В документації було проаналізовано та сформульовано основні задачі даної роботи, її мету. Розроблено план виконання роботи.

Також було розглянуто ще декілька алгоритмів кодування інформації для того, щоб мати уявлення чим саме код Гаффмана відрізняється від інших. У нашому випадку від тих, які були наведені у документації.

Також було розглянуто поетапне створення коду програмного продукту. Описано з чого він складається, з яких методів, для чого ці методи, як вони працюють. Даний архіватор складається з трьох основних класів – Node, PriorityQueue і Huffman.

У другому розділі було створено дві блок-схеми, які показують процес кодування та декодування, відповідно до коду програми. Також розписано про складові коду. А саме те, що спочатку ми створюємо клас Node, який зберігає в собі інформацію про вузли, символи, частоту та біти. Потім клас PriorityQueue, який безпосередньо взаємодіє з чергою (списком). І основний клас Huffman у якому й відбуваються наші дві основні функції - кодування та декодування файлу.

У третьому розділі ми ще раз згадали, про перевагу саме методу Гаффмана й більш детально описали методи, які використовувалися для стиснення та розстикання інформації в файлі.

Отже, документація до курсової роботи містить у собі алгоритм дій поетапного створення архіватора на основі коду Гаффмана, його актуальність й реалізація даної програми.

**Використані джерела**

1. Ю.Є. Гребенович, Б.О. Онищенко, О.О. Супруненко – «МЕТОДИЧНІ ВКАЗІВКИ ДО ВИКОНАННЯ ТА ОФОРМЛЕННЯ КУРСОВОЇ РОБОТИ», 2015 - С.32.

2. <https://habr.com/ru/post/144200/>.

3. <https://www.techiedelight.com/ru/huffman-coding/>.

4. <http://algolist.ru/compress/standard/huffman.php>.