НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. Ігоря СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

Комп'ютерний практикум №2
з курсу алгоритми кодування двійкових даних

РЕАЛІЗАЦІЯ БІТОВОГО ПОТОКУ

Виконав студент
групи ФІ-42мн
Бєш Радомир Андрійович

Київ — 2026

Зміст

1 Мета

Опанувати методи роботи із потоками бітів та із масивами даних на рівні окремих бітів.

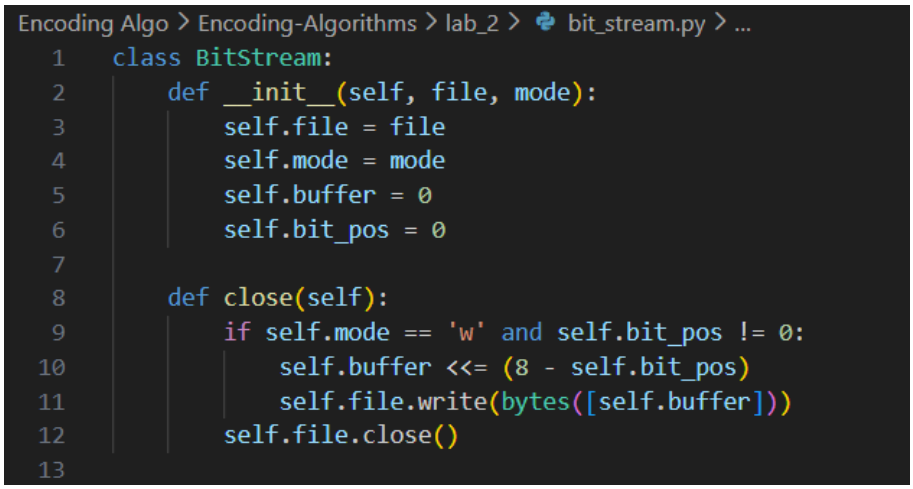
2 Постановка задачі

Реалізувати програмно інтерфейс для роботи із двійковими потоками даних. Реалізувати дві функії:
ReadBitSequence() - функція, яка вичитує із файла послідовність вказаної довжини;
WriteBitSequence() - функція, яка записує у файл послідовність бітів вказаної довжини. **Результати дослідження:**
`https://github.com/Radomir21/Encoding-Algorithms/tree/main/lab_2`.

3 Хід роботи

3.1 Особливості реалізації

1. Створення класу для роботи з бітовим потоком
   Було реалізовано окремий клас *BitStream*.Даний клас зберігає внутрішній байтовий буфер та позицію поточного біта, що дозволяє виконувати побітові операції читання і запису поверх звичайного файлу.



2. Буферизація бітів під час запису
   Запис бітів реалізовано шляхом поступового накопичення бітів у буфері. Кожен новий біт зсуває вміст буфера вліво та додається у молодший розряд. Після заповнення 8 біт формується байт і записується у файл.

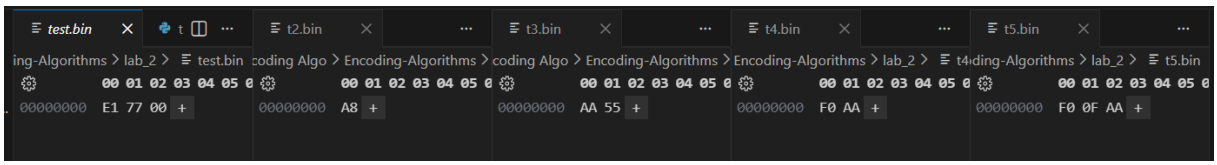
```python
def WriteBitSequence(bs, data, bit_length):
    bit_index = 0

    for _ in range(bit_length):
        byte_index = bit_index // 8
        bit_in_byte = 7 - (bit_index % 8)
        bit = (data[byte_index] >> bit_in_byte) & 1

        bs.buffer = (bs.buffer << 1) | bit
        bs.bit_pos += 1

        if bs.bit_pos == 8:
            bs.file.write(bytes([bs.buffer]))
            bs.buffer = 0
            bs.bit_pos = 0

        bit_index += 1
```

3. Зчитування з файлу
   Якщо у буфері відсутні доступні біти, з файлу зчитується новий байт, з якого по черзі витягуються біти, починаючи зі старшого. Функція *ReadBitSequence()* дозволяє зчитувати послідовності довільної довжини, незалежно від їх вирівнювання відносно байтів у файлі.

```python
def ReadBitSequence(bs, bit_length):
    result = []
    current_byte = 0
    bits_collected = 0

    for _ in range(bit_length):
        if bs.bit_pos == 0:
            byte = bs.file.read(1)
            if not byte:
                raise EOFError("Кінець файлу")
            bs.buffer = byte[0]
            bs.bit_pos = 8

        bit = (bs.buffer >> (bs.bit_pos - 1)) & 1
        bs.bit_pos -= 1

        current_byte = (current_byte << 1) | bit
        bits_collected += 1

        if bits_collected == 8:
            result.append(current_byte)
            current_byte = 0
            bits_collected = 0

    if bits_collected > 0:
        current_byte <<= (8 - bits_collected)
        result.append(current_byte)

    return bytes(result)
```

4. Вирівнювання бітового потоку при завершенні запису
   Вирівнювання бітового потоку до межі байта виконується лише під час закриття файлу. Якщо у буфері залишаються незаписані біти, вони доповнюються нульовими бітами та записуються у файл як завершальний байт. Даний механізм створен в класі *BitStream*.

3.2 Результати



4 Висновки

У даній лабораторній роботі було реалізовано інтерфейс для побітового запису та зчитування даних. Реалізація дозволяє працювати з бітовими послідовностями довільної довжини без прив'язки до меж байта.