1. Структура дерева кодирования Хаффмана

Дерево Хаффмана — это бинарное дерево, где каждый лист представляет символ из исходных данных, а каждая ветвь соответствует бинарному значению (0 или 1).

- **Листовые узлы (TreeLeaf):** содержат символ и его частоту.
- **Внутренние узлы (TreeNode):** не содержат символов, но хранят суммарную частоту всех символов в поддереве.

Каждое ребро дерева помечено либо 0, либо 1. Проход от корня до листа создаёт уникальный бинарный код для каждого символа. Более частые символы имеют короткие коды, а редкие — длинные.

2. Структура классов TreeNode и TreeLeaf

Класс TreeNode

Этот класс является базовым для всех узлов дерева.

Поля:

- **frequencyValue:** частота появления символов, содержащихся в поддереве.
- **leftChild:** ссылка на левое поддерево (соответствует 0).
- **rightChild:** ссылка на правое поддерево (соответствует 1).

Методы:

- **TreeNode(int frequency):** конструктор для инициализации узла с заданной частотой.
- **populateEncodingMap(String code, Map<Byte, String> encodingMap):** рекурсивно проходит по дереву и добавляет коды в карту кодирования.

Класс TreeLeaf

Этот класс наследуется от TreeNode и представляет лист дерева.

Дополнительные поля:

- **symbol:** символ, который этот лист представляет.

Методы:

- **TreeLeaf(byte symbol, int frequency):** конструктор для инициализации символа и его частоты.
- **populateEncodingMap(String code, Map<Byte, String> encodingMap):** добавляет текущий символ и его код в карту кодирования.

3. Принцип и алгоритмы кодирования дерева

3.1. Построение дерева

1. **Частотная таблица:** подсчитывается количество вхождений каждого символа в исходные данные.
2. **Очередь с приоритетом:** создаётся очередь, где приоритет определяется частотой символа (узлы с меньшей частотой имеют больший приоритет).
3. **Объединение узлов:**
    - Извлекаются два узла с минимальной частотой.
    - Создаётся новый узел с суммой их частот.
    - Этот узел добавляется обратно в очередь.
4. **Повторение:** Процесс продолжается, пока в очереди не останется один узел — корень дерева.

3.2. Генерация кодов

Коды генерируются путём рекурсивного прохода по дереву:

- Добавляем 0, переходя в левое поддерево.
- Добавляем 1, переходя в правое поддерево.
- Для листа сохраняется полученный путь как бинарный код символа.

4. Формат хранения данных

Данные хранятся в виде 3 блоков в сжатом файле:

Блок 1: количество уникальных символов

Блок 2: таблица кодов (вида 0:A, 1:B)

Блок 3: сжатые данные