

Описание структуры программы

1. Структура программы

Программа реализует кодирование и декодирование файлов методом Хаффмана с управлением через командную строку.

Основные компоненты:

1. Класс **HuffmanNode**:

- Описывает узел дерева Хаффмана.
- Содержит поля: `frequency`, `data`, `left`, `right`.
- Реализует интерфейс `Comparable<HuffmanNode>` для сравнения узлов по частоте.

2. Класс **HuffmanCoding**:

- Основной класс, реализующий алгоритм Хаффмана.
- Содержит статические поля: `huffmanCodes` (таблица кодов) и `root` (корень дерева Хаффмана).

3. Методы в **HuffmanCoding**:

- **main**: Управляет программой через командную строку.
- **readFile**: Читает данные из файла.
- **writeFile**: Записывает данные в файл.
- **encode**: Кодировать данные с использованием алгоритма Хаффмана.
- **decode**: Декодирует данные с использованием дерева Хаффмана.
- **buildFrequencyMap**: Строит карту частот символов.
- **buildHuffmanTree**: Строит дерево Хаффмана на основе карты частот.
- **generateCodes**: Генерирует коды Хаффмана для каждого символа.
- **saveCodesToFile**: Сохраняет таблицу кодов в файл.
- **loadCodesFromFile**: Загружает таблицу кодов из файла.
- **rebuildHuffmanTree**: Восстанавливает дерево Хаффмана из таблицы кодов.

Краткое описание:

- Программа кодирует и декодирует данные с использованием алгоритма Хаффмана.
- Таблица кодов сохраняется в файл для последующего декодирования.
- Дерево Хаффмана восстанавливается из таблицы кодов при декодировании.

2. Описание структуры закодированного файла

Формат хранения данных:

1. Кодирование:

- Программа создает два файла:
 - encoded.txt: Закодированные данные.
 - huffman_codes.txt: Словарь (таблица кодов).

2. Декодирование:

- Программа использует encoded.txt и huffman_codes.txt для восстановления исходных данных.

Итоговая структура

1. Словарь:

- Сериализованный объект Map<Character, String>.
- Хранится в файле huffman_codes.txt.

2. Закодированные данные:

- Последовательность битов.
- Хранится в файле encoded.txt.