

Структура дерева кодирования

Структура дерева Хаффмана строится на основе **PriorityQueue**. Каждый узел дерева представляет символ и его частоту в исходном тексте. Узлы с наименьшей частотой объединяются вместе, строя более высокоуровневые узлы, пока не будет построено полное дерево.

Кодирование происходит путем присвоения битовых кодов символам в соответствии с их положением в дереве. Декодирование выполняется путем следования по дереву от корня к листьям, используя биты закодированного текста.

HuffmanNode (класс):

Определяет узел дерева Хаффмана.

Содержит поля:

1. **data**: целочисленное значение, представляющее частоту (или сумму частот) символов в поддереве, связанном с данным узлом.
2. **c**: символ, который представляется в листьях дерева. Для внутренних узлов это символ '\0'.
3. **left** и **right**: ссылки на левое и правое поддерева соответственно.

Реализует интерфейс **Comparable**, чтобы узлы можно было сравнивать по их частотам.

HuffmanCoding (класс):

Основной класс, который выполняет кодирование и декодирование с использованием алгоритма Хаффмана.

Использует статическую переменную **huffmanCodes** для хранения кодов Хаффмана для каждого символа.

Метод **encode**:

1. Считывает текст из файла и строит частотный массив символов.
2. Строит дерево Хаффмана на основе частот.
3. Генерирует коды Хаффмана для каждого символа.
4. Записывает дерево в закодированный файл и кодирует текст.

Метод **decode**:

1. Считывает дерево Хаффмана из закодированного файла.
2. Декодирует закодированный текст, используя дерево.

Методы **buildFrequencyArray** и **buildHuffmanTree** используются для построения частотного массива и дерева Хаффмана соответственно.

Методы **generateHuffmanCodes** и **printHuffmanTree** помогают визуализировать коды и структуру дерева.

Методы **writeTree**, **writeEncodedText**, **readTree** и **readEncodedText** используются для записи и чтения дерева и закодированного текста в/из файла.

Основной метод **main**:

1. Задаёт пути к входному файлу (**inputFile**), закодированному файлу (**encodedFile**) и декодированному файлу (**decodedFile**).
2. Выполняет кодирование и декодирование.
3. Выводит информацию о размерах файлов и коэффициенте сжатия.