Huffman

Oleksandr Holobokov

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 3

Class Documentation

3.1 Compare Struct Reference

Struktura porównująca węzły na podstawie częstotliwości.

Public Member Functions

- bool operator() (const Node *a, const Node *b)
  Operator porównania.

3.1.1 Detailed Description

Struktura porównująca węzły na podstawie częstotliwości.

3.1.2 Member Function Documentation

3.1.2.1 operator()()

```
bool Compare::operator() (
            const Node * a,
            const Node * b)  [inline]
```

Operator porównania.

Parameters

a	Wskaźnik na pierwszy węzeł.
b	Wskaźnik na drugi węzeł.

Returns

true, jeśli częstotliwość `a` jest większa niż `b`.

The documentation for this struct was generated from the following file:

- Huffman.cpp

3.2 Node Struct Reference

Struktura reprezentująca węzeł drzewa Huffmana.

Public Member Functions

- **Node** ()
  Konstruktor domyślny.
- Node (char s, int i)
  Konstruktor liścia.
- Node (int i, Node ∗l=nullptr, Node ∗r=nullptr)
  Konstruktor węzła wewnętrznego.

Public Attributes

- char **data**
- int **freq**
- Node ∗ **left**
- Node ∗ **right**

3.2.1 Detailed Description

Struktura reprezentująca węzeł drzewa Huffmana.

Węzeł drzewa Huffmana, przechowuje symbol, częstotliwość i wskaźniki na dzieci.

3.2.2 Constructor & Destructor Documentation

3.2.2.1 Node() [1/2]

```
Node::Node (
            char s,
            int i)  [inline]
```

Konstruktor liścia.

Parameters

s	Symbol (znak).
i	Częstotliwość wystąpień symbolu.

3.2.2.2 Node() [2/2]

```
Node::Node (
            int i,
            Node * l = nullptr,
            Node * r = nullptr)  [inline]
```

Konstruktor węzła wewnętrznego.

Parameters

i	Suma częstotliwości dzieci.
l	Wskaźnik na lewe dziecko.
r	Wskaźnik na prawe dziecko.

The documentation for this struct was generated from the following file:

- Huffman.cpp

Chapter 4

File Documentation

4.1 Huffman.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <map>
#include <bitset>
#include <vector>
#include <queue>
#include "Huffman.h"
```

Classes

- struct Node

  Struktura reprezentująca węzeł drzewa Huffmana.
- struct Compare

  Struktura porównująca węzły na podstawie częstotliwości.

Functions

- void **help** ()

  Wyświetla instrukcję obsługi programu.
- bool getCommands (int argc, char ∗argv[ ], std::string &inputFile, std::string &outputFile, std::string &dictionaryFile, char &mode)

  Funkcja przetwarza argumenty wiersza poleceń i weryfikuje ich poprawność.
- Node ∗ buildHuffmanTree (const std::vector< char > &buffer)

  Buduje drzewo Huffmana na podstawie częstotliwości symboli.
- void assignCodes (Node ∗root, std::string code, std::map< char, std::string > &huffmanCodes)

  Przypisuje kody Huffmana liściom drzewa.
- void deleteTree (Node ∗root)

  Usuwa drzewo Huffmana z pamięci.
- std::string encodeText (std::vector< char > &buffer, std::map< char, std::string > &huffmanCodes)

  Koduje tekst na podstawie kodów Huffmana.
- bool writeCompressedText (std::string compressedText, std::string &outputFile)

Zapisuje skompresowany tekst do pliku w formie binarnej.

- void writeDictionary (std::map< char, std::string > &huffmanCodes, std::string &dictionaryFile)

  Zapisuje słownik kodowania Huffmana do pliku.
- void readFromDictionary (std::map< char, std::string > &huffmanCodes, std::string &dictionaryFile)

  Odczytuje słownik kodowania Huffmana z pliku.
- void recreateText (std::map< char, std::string > &huffmanCodes, std::string &bits, std::string &recreatedText)

  Dekoduje ciąg bitów na oryginalny tekst na podstawie słownika Huffmana.
- bool writeDecompressedText (std::string compressedText, std::string &outputFile)

  Zapisuje zdekompresowany tekst do pliku wyjściowego.
- void compressFile (std::string &inputFile, std::string &outputFile, std::string &dictionaryFile)

  Funkcja kompresuje plik za pomocą kodowania Huffmana.
- void decompressFile (std::string &inputFile, std::string &outputFile, std::string &dictionaryFile)

  Funkcja dekompresuje plik za pomocą kodowania Huffmana.

4.1.1 Detailed Description

@Author Oleksandr Holobokov (oh318935@student.polsl.pl)

Date

29.01.2025

4.1.2 Function Documentation

4.1.2.1 assignCodes()

```
void assignCodes (
            Node * root,
            std::string code,
            std::map< char, std::string > & huffmanCodes)
```

Przypisuje kody Huffmana liściom drzewa.

Parameters

root	Korzeń drzewa Huffmana.
code	Aktualny kod binarny.
huffmanCodes	Mapa przechowująca przypisane kody.

4.1.2.2 buildHuffmanTree()

```
Node * buildHuffmanTree (
            const std::vector< char > & buffer)
```

Buduje drzewo Huffmana na podstawie częstotliwości symboli.

Parameters

buffer	Bufor zawierający znaki z pliku wejściowego.

Returns

Wskaźnik na korzeń drzewa Huffmana.

4.1.2.3 compressFile()

```
void compressFile (
            std::string & inputFile,
            std::string & outputFile,
            std::string & dictionaryFile)
```

Funkcja kompresuje plik za pomocą kodowania Huffmana.

Funkcja odczytuje zawartość pliku wejściowego, buduje drzewo Huffmana na podstawie częstotliwości znaków, przypisuje kody binarne i zapisuje skompresowany tekst oraz słownik kodowania do odpowiednich plików.

Parameters

inputFile	Ścieżka do pliku wejściowego.
outputFile	Ścieżka do pliku wyjściowego.
dictionaryFile	Ścieżka do pliku ze słownikiem kodowania.

4.1.2.4 decompressFile()

```
void decompressFile (
            std::string & inputFile,
            std::string & outputFile,
            std::string & dictionaryFile)
```

Funkcja dekompresuje plik za pomocą kodowania Huffmana.

Funkcja odczytuje plik wejściowy oraz słownik kodowania, a następnie odtwarza oryginalną zawartość pliku na podstawie kodów Huffmana.

Parameters

inputFile	Ścieżka do pliku wejściowego (skompresowanego).
outputFile	Ścieżka do pliku wyjściowego (zdekompresowanego).
dictionaryFile	Ścieżka do pliku ze słownikiem kodowania.

4.1.2.5 deleteTree()

```
void deleteTree (
            Node * root)
```

Usuwa drzewo Huffmana z pamięci.

Parameters

root	Korzeń drzewa Huffmana.

4.1.2.6 encodeText()

```
std::string encodeText (
            std::vector< char > & buffer,
            std::map< char, std::string > & huffmanCodes)
```

Koduje tekst na podstawie kodów Huffmana.

Parameters

buffer	Bufor zawierający znaki z pliku wejściowego.
huffmanCodes	Mapa przechowująca przypisane kody.

Returns

Zakodowany tekst w formacie binarnym.

4.1.2.7 getCommands()

```
bool getCommands (
            int argc,
            char * argv[],
            std::string & inputFile,
            std::string & outputFile,
            std::string & dictionaryFile,
            char & mode)
```

Funkcja przetwarza argumenty wiersza poleceń i weryfikuje ich poprawność.

Funkcja analizuje argumenty wiersza poleceń, przypisuje wartości do odpowiednich parametrów, sprawdza ich poprawność i ustawia domyślny tryb działania (jeśli nie podano trybu). W przypadku błędnych danych lub braku wymaganych argumentów, funkcja zwraca `false`.

Parameters

argc	Liczba argumentów wiersza poleceń.
argv	Tablica argumentów wiersza poleceń.
inputFile	Ścieżka do pliku wejściowego (parametr wyjściowy).
outputFile	Ścieżka do pliku wyjściowego (parametr wyjściowy).
dictionaryFile	Ścieżka do pliku ze słownikiem (parametr wyjściowy).
mode	Tryb działania programu ('k' dla kompresji, 'd' dla dekompresji) (parametr wyjściowy).

Returns

`true` jeśli wszystkie argumenty są poprawne, `false` w przypadku błędu.

4.1.2.8 readFromDictionary()

```
void readFromDictionary (
            std::map< char, std::string > & huffmanCodes,
            std::string & dictionaryFile)
```

Odczytuje słownik kodowania Huffmana z pliku.

Funkcja odczytuje zawartość pliku tekstowego `dictionaryFile`, w którym każda linia zawiera symbol (lub specjalny znak, jak spacja czy nowa linia) oraz odpowiadający mu kod Huffmana. Mapuje symbole do ich kodów w przekazanej mapie `huffmanCodes`.

Format pliku:

- [symbol][kod][
  ]
- Pusta linia reprezentuje znak nowej linii (`\n`), a kolejna linia zawiera odpowiadający mu kod.

Parameters

huffmanCodes	Mapa, która będzie zawierać odczytane symbole i ich kody Huffmana.
dictionaryFile	Ścieżka do pliku zawierającego słownik kodowania Huffmana.

4.1.2.9 recreateText()

```
void recreateText (
            std::map< char, std::string > & huffmanCodes,
            std::string & bits,
            std::string & recreatedText)
```

Dekoduje ciąg bitów na oryginalny tekst na podstawie słownika Huffmana.

Funkcja wykorzystuje mapę `huffmanCodes`, aby dekodować ciąg `bits` (ciąg znaków '0' i '1') na oryginalny tekst, który jest zapisywany w `recreatedText`. Dla każdego fragmentu bitów, funkcja dopasowuje kod do odpowiedniego symbolu w słowniku Huffmana.

Parameters

huffmanCodes	Mapa zawierająca symbole i odpowiadające im kody Huffmana.
bits	Skompresowany ciąg bitów w postaci ciągu znaków '0' i '1'.
recreatedText	Wyjściowy odtworzony tekst, wypełniany przez funkcję.

4.1.2.10 writeCompressedText()

```
bool writeCompressedText (
            std::string compressedText,
            std::string & outputFile)
```

Zapisuje skompresowany tekst do pliku w formie binarnej.

Funkcja przetwarza skompresowany tekst (ciąg znaków '0' i '1') na bajty, a następnie zapisuje te bajty w pliku binarnym. Każde 8 bitów (1 bajt) z ciągu wejściowego jest przekształcane na wartość typu `unsigned char`.

Parameters

compressedText	Skompresowany tekst w postaci ciągu '0' i '1'.
outputFile	Nazwa pliku wyjściowego, w którym dane mają zostać zapisane.

Returns

true, jeśli zapis zakończył się sukcesem, false w przypadku błędu (np. brak dostępu do pliku).

4.1.2.11 writeDecompressedText()

```
bool writeDecompressedText (
            std::string compressedText,
            std::string & outputFile)
```

Zapisuje zdekompresowany tekst do pliku wyjściowego.

Funkcja zapisuje ciąg znaków zdekompresowanego tekstu do pliku wyjściowego. Jeśli plik nie może zostać otwarty, wyświetlany jest komunikat o błędzie, a funkcja zwraca wartość `false`.

Parameters

compressedText	Zdekompresowany tekst, który ma zostać zapisany do pliku.
outputFile	Nazwa pliku wyjściowego, w którym tekst zostanie zapisany.

Returns

true, jeśli zapis zakończył się sukcesem; false w przypadku błędu.

4.1.2.12 writeDictionary()

```
void writeDictionary (
            std::map< char, std::string > & huffmanCodes,
            std::string & dictionaryFile)
```

Zapisuje słownik kodowania Huffmana do pliku.

Funkcja zapisuje zawartość mapy `huffmanCodes` do pliku tekstowego, gdzie każdy klucz (symbol) jest zapisany wraz z odpowiadającym mu kodem Huffmana w formacie: [symbol][kod][
]

W przypadku symboli specjalnych, takich jak spacja lub znak nowej linii, zapisywane są odpowiednie znaki w pliku (np. spacja pozostaje spacją, a nowa linia jest zapisana w osobnym wierszu).

Parameters

huffmanCodes	Mapa przechowująca symbole i ich odpowiadające kody Huffmana.
dictionaryFile	Ścieżka do pliku, w którym słownik ma zostać zapisany.

4.2 Huffman.h File Reference

```
#include <iostream>
```

Functions

- bool getCommands (int argc, char *argv[], std::string &inputFile, std::string &outputFile, std::string &dictionaryFile, char &mode)

  Funkcja przetwarza argumenty wiersza poleceń i weryfikuje ich poprawność.
- void compressFile (std::string &inputFile, std::string &outputFile, std::string &dictionaryFile)

  Funkcja kompresuje plik za pomocą kodowania Huffmana.
- void decompressFile (std::string &inputFile, std::string &outputFile, std::string &dictionaryFile)

  Funkcja dekompresuje plik za pomocą kodowania Huffmana.

4.2.1 Detailed Description

@Author Oleksandr Holobokov (oh318935@student.polsl.pl)

Date

29.01.2025

4.2.2 Function Documentation

4.2.2.1 compressFile()

```
void compressFile (
            std::string & inputFile,
            std::string & outputFile,
            std::string & dictionaryFile)
```

Funkcja kompresuje plik za pomocą kodowania Huffmana.

Funkcja odczytuje zawartość pliku wejściowego, buduje drzewo Huffmana na podstawie częstotliwości znaków, przypisuje kody binarne i zapisuje skompresowany tekst oraz słownik kodowania do odpowiednich plików.

Parameters

inputFile	Ścieżka do pliku wejściowego.
outputFile	Ścieżka do pliku wyjściowego.
dictionaryFile	Ścieżka do pliku ze słownikiem kodowania.

4.2.2.2 decompressFile()

```
void decompressFile (
            std::string & inputFile,
            std::string & outputFile,
            std::string & dictionaryFile)
```

Funkcja dekompresuje plik za pomocą kodowania Huffmana.

Funkcja odczytuje plik wejściowy oraz słownik kodowania, a następnie odtwarza oryginalną zawartość pliku na podstawie kodów Huffmana.

Parameters

inputFile	Ścieżka do pliku wejściowego (skompresowanego).
outputFile	Ścieżka do pliku wyjściowego (zdekompresowanego).
dictionaryFile	Ścieżka do pliku ze słownikiem kodowania.

4.2.2.3 getCommands()

```
bool getCommands (
            int argc,
            char * argv[],
            std::string & inputFile,
            std::string & outputFile,
            std::string & dictionaryFile,
            char & mode)
```

Funkcja przetwarza argumenty wiersza poleceń i weryfikuje ich poprawność.

Funkcja analizuje argumenty wiersza poleceń, przypisuje wartości do odpowiednich parametrów, sprawdza ich poprawność i ustawia domyślny tryb działania (jeśli nie podano trybu). W przypadku błędnych danych lub braku wymaganych argumentów, funkcja zwraca `false`.

Parameters

argc	Liczba argumentów wiersza poleceń.
argv	Tablica argumentów wiersza poleceń.
inputFile	Ścieżka do pliku wejściowego (parametr wyjściowy).
outputFile	Ścieżka do pliku wyjściowego (parametr wyjściowy).
dictionaryFile	Ścieżka do pliku ze słownikiem (parametr wyjściowy).
mode	Tryb działania programu ('k' dla kompresji, 'd' dla dekompresji) (parametr wyjściowy).

Returns

`true` jeśli wszystkie argumenty są poprawne, `false` w przypadku błędu.

4.3 Huffman.h

Go to the documentation of this file.

```
00001
00006
00007 #include <iostream>
00008
00009 #ifndef HUFFMAN_H
00010 #define HUFFMAN_H
00011
00028 bool getCommands(int argc, char* argv[], std::string& inputFile, std::string& outputFile, std::string&
      dictionaryFile, char& mode);
00029
00041 void compressFile(std::string &inputFile, std::string& outputFile, std::string &dictionaryFile);
00042
00053 void decompressFile(std::string &inputFile, std::string& outputFile, std::string &dictionaryFile);
00054
00055 #endif
```

4.4 main.cpp File Reference

```
#include <iostream>
#include "Huffman.h"
```

Functions

- int main (int argc, char ∗argv[ ])

  Główna funkcja obsługująca kompresję i dekompresję plików za pomocą kodowania Huffmana.

4.4.1 Detailed Description

@Author Oleksandr Holobokov (oh318935@student.polsl.pl)

Date

29.01.2025

4.4.2 Function Documentation

4.4.2.1 main()

```
int main (
            int argc,
            char * argv[ ])
```

Główna funkcja obsługująca kompresję i dekompresję plików za pomocą kodowania Huffmana.

Parameters

argc	Liczba argumentów wiersza poleceń.
argv	Tablica argumentów wiersza poleceń.

Returns

int Zwraca 0 w przypadku pomyślnego wykonania, 1 w przypadku błędnych argumentów wiersza poleceń.

Parsuje argumenty wiersza poleceń.

Funkcja ustawia wartości zmiennych inputFile, outputFile, dictionaryFile oraz mode na podstawie dostarczonych argumentów wiersza poleceń.

Parameters

argc	Liczba argumentów wiersza poleceń.
argv	Tablica argumentów wiersza poleceń.
inputFile	Nazwa pliku wejściowego do przetworzenia.
outputFile	Nazwa pliku wyjściowego, w którym zostaną zapisane wyniki.
dictionaryFile	Nazwa pliku słownika używanego w kodowaniu Huffmana.
mode	Znak określający tryb działania ('k' dla kompresji, 'd' dla dekompresji).

Returns

bool Zwraca true, jeśli argumenty wiersza poleceń są poprawne, w przeciwnym razie false.

Kompresuje plik wejściowy za pomocą kodowania Huffmana.

Parameters

inputFile	Nazwa pliku wejściowego do skompresowania.
outputFile	Nazwa pliku wyjściowego, w którym zostaną zapisane dane skompresowane.
dictionaryFile	Nazwa pliku słownika, w którym zostaną zapisane kody Huffmana.

Dekompresuje plik wejściowy za pomocą kodowania Huffmana.

Parameters

inputFile	Nazwa pliku wejściowego do zdekompresowania.
outputFile	Nazwa pliku wyjściowego, w którym zostaną zapisane dane zdekompresowane.
dictionaryFile	Nazwa pliku słownika zawierającego kody Huffmana.

Index