***# Secondary school of electrotechnical engineering Jecna 30***

***# 2022-2023***

***# Matěj Šturma***

***# C4c***

## **Introduction**

The intention of this program is to compress and decompress given files according to the Huffman Coding Algorithm.

It is written in Python.

### Hierarchy

* /src/ - this folder contains the source code. Huffman.py, which contains the implementation and main.py, which contains the UI.
* /doc/ - this folder contains the documentation.
* /log/ - this folder contains the XML files with the process logs.
* /config/ - this folder contains the config.json file, which you can customize according to your preferences. *More details in the* [*configuration*](#_w3y7ssipgn60) *part.*
* /data/ - this is the default folder for the output.

## **Configuration**

As mentioned before, configuration can be made in the /config/config.json file.

Parameters:

* **path**: the input path.

## **Source Code**

### Classes

#### Node

* This class is used to create a node for the huffman tree
* Methods
* \_\_lt\_\_: used to compare two nodes
* \_\_eq\_\_: used to compare two nodes
* \_\_repr\_\_: used to print the node

#### Huffman

* This class is used to perform huffman encoding and decoding
* Methods
* make\_freq\_dict: used to make a dictionary of frequencies of each character
* make\_heap: used to make a heap of nodes
* merge\_nodes: used to merge nodes in the heap
* make\_codes\_helper: used to make codes for each character
* make\_codes: used to make codes for each character
* get\_encoded\_text: used to get the encoded text
* pad\_encoded\_text: used to pad the encoded text
* get\_byte\_array: used to get the byte array of the encoded text
* compress: used to compress the text
* decompress: used to decompress the text