

main

```
+ main(in arg: char** in arg: int)
+ mainPage()
```

properties

```
+ create_properties(in arg: char** in arg: int) PROPERTIES*
+ delete_properties(in prop: PROPERTIES*)
+ properties_get(in prop: PROPERTIES*) MODE
+ properties_get_input_stream(in prop: PROPERTIES*) FILE*
+ properties_get_output_stream(in prop: PROPERTIES*) FILE*
```

+PROPERTIES

compressor

```
+ compress(in prop: PROPERTIES*)
+ decompress(in prop: PROPERTIES*)
```

content\_coder

```
+ encode_content(in input_stream: FILE*, in output_stream: FILE*, in codeTab: CODETAB*)
+ decode_content(in input_stream: FILE*, in output_stream: FILE*, in codeTab: CODETAB*)
```

codeTab

```
+ create_codeTab(in htree: HTREE*, CODETAB*)
+ read_codeTab(in input_stream: FILE*, CODETAB*)
+ write_codeTab(in output_stream: FILE*, in codeTab: CODETAB*)
+ delete_codeTab(in codeTab: CODETAB*)
+ codeTab_get_char(in codeTab: CODETAB*) short
+ codeTab_get_char(in code: short, in codeTab: CODETAB*) char
```

+CODETAB

htree

```
+ create_htree(in freqTab: FREQTAB*, HTREE*)
+ delete_htree(in htree: HTREE*)
+ htree_get_element(in htree: HTREE*) CODETAB_ELEMENT*
+ htree_is_empty(in htree: HTREE*) boolean
+ htree_prep_codeTab_element(in htree: HTREE*)
```

+HTREE

freqTab

```
+ create_frequency_table(in input_stream: FILE*, FREQTAB*)
+ delete_frequency_table(in freqTab: FREQTAB*)
+ frequency_table_get_element(in freqTab: FREQTAB*) FREQTAB_ELEMENT*
+ frequency_table_is_empty(in freqTab: FREQTAB*) boolean
```

+FREQTAB

freqTab\_element

```
+ create_freqTab_element(in frequency: long, in character: char) FREQTAB_ELEMENT*
+ delete_freqTab_element(in element: FREQTAB_ELEMENT*)
+ freqTab_element_get_char(in element: FREQTAB_ELEMENT*) char
+ freqTab_element_get_char(in element: FREQTAB_ELEMENT*) long
```

+FREQTAB\_ELEMENT

pqueue

```
+ create_pqueue() PQQUEUE*
+ delete_pqueue(in pq: PQQUEUE*)
+ pqqueue_insert(in pq: PQQUEUE*, in htree_element: HTREE_ELEMENT*)
+ pqqueue_get_min(in pq: PQQUEUE*) HTREE_ELEMENT*
```

+PQQUEUE

htree\_element

```
+ create_htree_node(in left_child: HTREE_ELEMENT*, in right_child: HTREE_ELEMENT*)
+ create_htree_leaf(in frequency: long, in character: char) HTREE_ELEMENT*
+ delete_htree_element(in element: HTREE_ELEMENT*)
+ merge_htree_elements(in left_child: HTREE_ELEMENT*, in right_child: HTREE_ELEMENT*) HTREE_ELEMENT*
+ htree_element_is_leaf(in htree_element: HTREE_ELEMENT*) boolean
+ htree_element_print(in htree_element: HTREE_ELEMENT*)
+ htree_node_get_left_child(in node: HTREE_ELEMENT*) HTREE_ELEMENT*
+ htree_node_get_right_child(in node: HTREE_ELEMENT*) HTREE_ELEMENT*
+ htree_node_has_left_child(in node: HTREE_ELEMENT*) boolean
+ htree_node_has_right_child(in node: HTREE_ELEMENT*) boolean
+ htree_node_get_weight(in node: HTREE_ELEMENT*) long
+ htree_leaf_get_char(in leaf: HTREE_ELEMENT*) char
+ htree_leaf_get_frequency(in leaf: HTREE_ELEMENT*) long
```

+HTREE\_ELEMENT  
+HTREE\_NODE  
+HTREE\_NODE