```
class HuffmanCodedNode

- byte data
- String code

+ HuffmanCodedNode(byte data, String code)

+ GET byte Data
+ GET String Code
+ String ToString()
```

```
class Node

- Object data;
- List<Node> par;
- List<Node> ch;

+ Node(Object data, Node parent)
+ Node(Object data)
+ Node()

+ bool IsLeaf()
+ GET List<Node> Children
+ void AddChild(Node child)
+ GET;SET Object Data
+ String ToString()
```

data

```
class Tree

- Node root
- int height

+ Tree(Node root, int height)
+ Tree(Node root)
+ Tree()

+ GET int Height
+ GET Node Root
+ String ToString()
```

root

data *

```
class HuffmanNode

+ int freq
+ byte data

+ HuffmanNode(byte data, int freq)
+ HuffmanNode(byte data)
+ HuffmanNode()

+ GET;SET int Frequence
+ GET byte Data
+ operators ==, !=, <, >, >=, <=
+ string ToString()

+ static void Swap(ref HuffmanNode A, ref HuffmanNode B)
+ static int Occurred(HuffmanNode[] data, HuffmanNode el)
```

```
class Huffman

- byte[] UncompressedData
- byte[] CompressedData
- int UncompressedLength
- HuffmanNode[] huffmanNodes

+ Huffman(byte[] data)

+ void Encoding()

- void CreateHuffmanNodes()
- static HuffmanNode[] SimpleSort(HuffmanNode[] nodes)
- static Tree HuffmanTree(HuffmanNode[] nodes)
- static Tree EncodingTree(Tree huffmanTree)
```

huffmanNodes

```
static class Utils

+ static String GetInfo()
+ static String[] ReadStringArray(String path)
+ static byte[] ReadByteArray(String path)
+ static byte[] StringArrayToByteArray(String[] text)
+ static BitArray ByteArrayToBitArray(byte[] bytes)
+ static int Contains(byte[] data, byte element)

+ static void PrintValues(IEnumerable myList, int myWidth)
+ static void PrintBits(BitArray myList, int myWidth)
```

```
class LZCodedNode

- int index
- byte mismatch

+ LZCodedNode(int index, byte mismatch)
```

```
class LempelZiv

+ static void Encode(byte[] data)
```