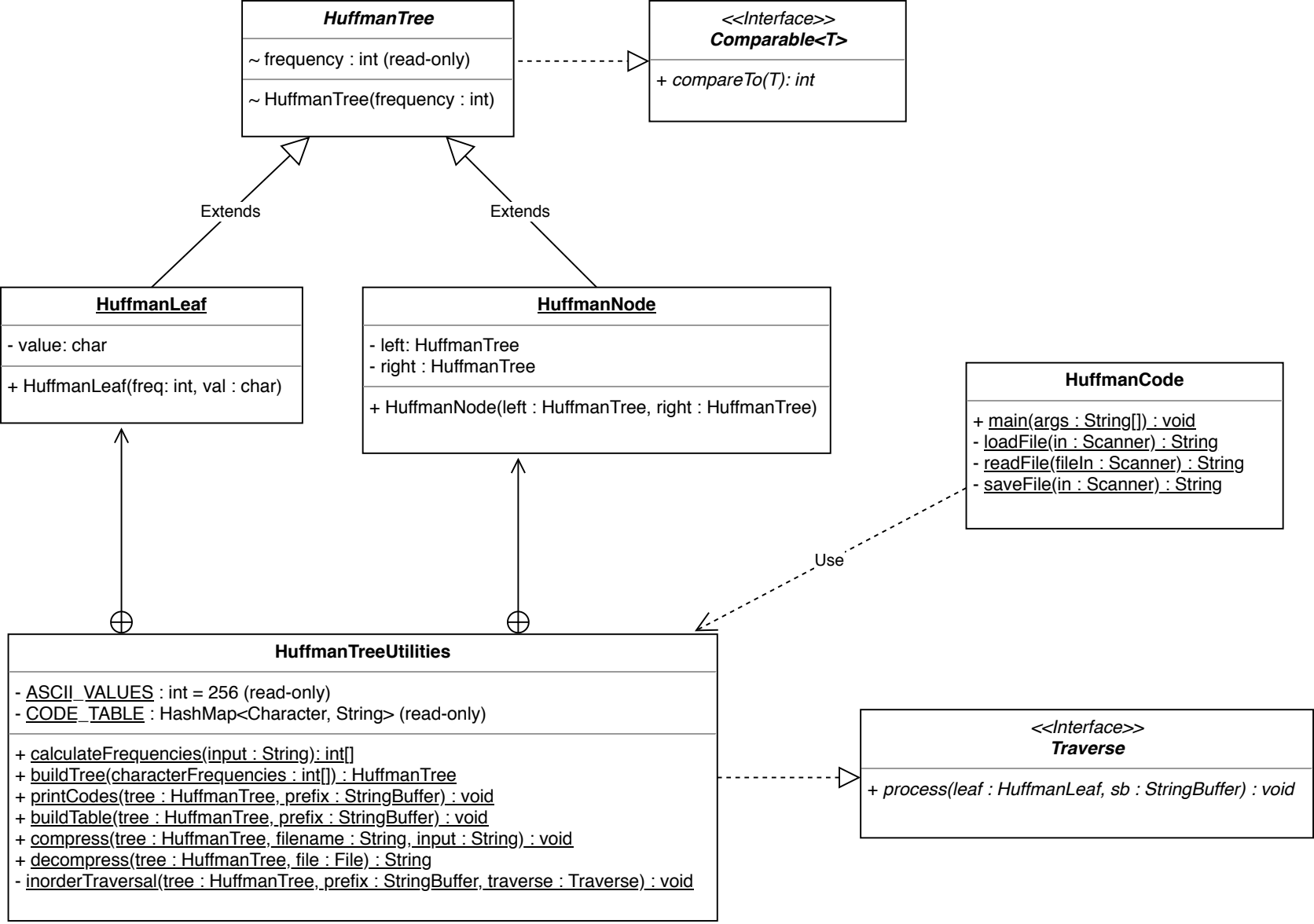
HuffmanTree
~ frequency : int (read-only)
~ HuffmanTree(frequency : int)
<<Interface>>
Comparable<T>
+ compareTo(T): int
Extends
Extends
HuffmanLeaf
- value: char
+ HuffmanLeaf(freq: int, val : char)
HuffmanNode
- left: HuffmanTree
- right : HuffmanTree
+ HuffmanNode(left : HuffmanTree, right : HuffmanTree)
HuffmanCode
+ main(args : String[]) : void
- loadFile(in : Scanner) : String
- readFile(fileIn : Scanner) : String
- saveFile(in : Scanner) : String
Use
HuffmanTreeUtilities
- ASCII_VALUES : int = 256 (read-only)
- CODE_TABLE : HashMap<Character, String> (read-only)
+ calculateFrequencies(input : String): int[]
+ buildTree(characterFrequencies : int[]) : HuffmanTree
+ printCodes(tree : HuffmanTree, prefix : StringBuffer) : void
+ buildTable(tree : HuffmanTree, prefix : StringBuffer) : void
+ compress(tree : HuffmanTree, filename : String, input : String) : void
+ decompress(tree : HuffmanTree, file : File) : String
- inorderTraversal(tree : HuffmanTree, prefix : StringBuffer, traverse : Traverse) : void
<<Interface>>
Traverse
+ process(leaf : HuffmanLeaf, sb : StringBuffer) : void