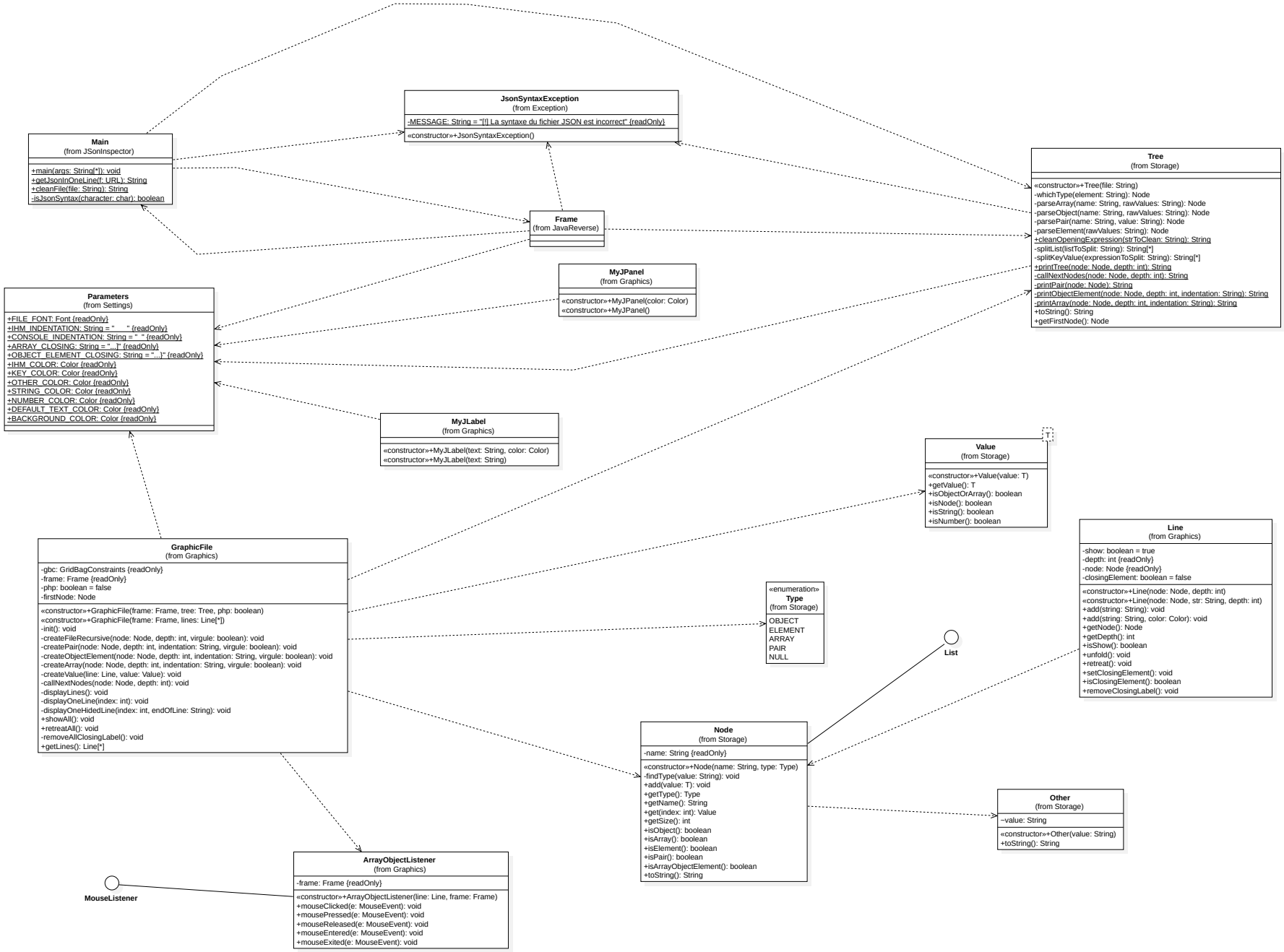
Model::Main


JsonSyntaxException
(from Exception)
-MESSAGE: String = "[!] La syntaxe du fichier JSON est incorrect" {readOnly}
«constructor»+JsonSyntaxException()
Main
(from JSonInspector)
+main(args: String[*]): void
+getJsonInOneLine(f: URL): String
+cleanFile(file: String): String
-isJsonSyntax(character: char): boolean
Tree
(from Storage)
«constructor»+Tree(file: String)
-whichType(element: String): Node
-parseArray(name: String, rawValues: String): Node
-parseObject(name: String, rawValues: String): Node
-parsePair(name: String, value: String): Node
-parseElement(rawValues: String): Node
+cleanOpeningExpression(strToClean: String): String
-splitList(listToSplit: String): String[*]
-splitKeyValue(expressionToSplit: String): String[*]
+printTree(node: Node, depth: int): String
-callNextNodes(node: Node, depth: int): String
-printPair(node: Node): String
-printObjectElement(node: Node, depth: int, indentation: String): String
-printArray(node: Node, depth: int, indentation: String): String
+toString(): String
+getFirstNode(): Node
Frame
(from JavaReverse)
MyJPanel
(from Graphics)
«constructor»+MyJPanel(color: Color)
«constructor»+MyJPanel()
Parameters
(from Settings)
+FILE_FONT: Font {readOnly}
+IHM_INDENTATION: String = " " {readOnly}
+CONSOLE_INDENTATION: String = " " {readOnly}
+ARRAY_CLOSING: String = "...]" {readOnly}
+OBJECT_ELEMENT_CLOSING: String = "...}" {readOnly}
+IHM_COLOR: Color {readOnly}
+KEY_COLOR: Color {readOnly}
+OTHER_COLOR: Color {readOnly}
+STRING_COLOR: Color {readOnly}
+NUMBER_COLOR: Color {readOnly}
+DEFAULT_TEXT_COLOR: Color {readOnly}
+BACKGROUND_COLOR: Color {readOnly}
MyJLabel
(from Graphics)
«constructor»+MyJLabel(text: String, color: Color)
«constructor»+MyJLabel(text: String)
T
Value
(from Storage)
«constructor»+Value(value: T)
+getValue(): T
+isObjectOrArray(): boolean
+isNode(): boolean
+isString(): boolean
+isNumber(): boolean
Line
(from Graphics)
-show: boolean = true
-depth: int {readOnly}
-node: Node {readOnly}
-closingElement: boolean = false
«constructor»+Line(node: Node, depth: int)
«constructor»+Line(node: Node, str: String, depth: int)
+add(string: String): void
+add(string: String, color: Color): void
+getNode(): Node
+getDepth(): int
+isShow(): boolean
+unfold(): void
+retreat(): void
+setClosingElement(): void
+isClosingElement(): boolean
+removeClosingLabel(): void
GraphicFile
(from Graphics)
-gbc: GridBagConstraints {readOnly}
-frame: Frame {readOnly}
-php: boolean = false
-firstNode: Node
«constructor»+GraphicFile(frame: Frame, tree: Tree, php: boolean)
«constructor»+GraphicFile(frame: Frame, lines: Line[*])
-init(): void
-createFileRecursive(node: Node, depth: int, virgule: boolean): void
-createPair(node: Node, depth: int, indentation: String, virgule: boolean): void
-createObjectElement(node: Node, depth: int, indentation: String, virgule: boolean): void
-createArray(node: Node, depth: int, indentation: String, virgule: boolean): void
-createValue(line: Line, value: Value): void
-callNextNodes(node: Node, depth: int): void
-displayLines(): void
-displayOneLine(index: int): void
-displayOneHidedLine(index: int, endOfLine: String): void
+showAll(): void
+retreatAll(): void
-removeAllClosingLabel(): void
+getLines(): Line[*]
«enumeration»
Type
(from Storage)
OBJECT
ELEMENT
ARRAY
PAIR
NULL
List
Node
(from Storage)
-name: String {readOnly}
«constructor»+Node(name: String, type: Type)
-findType(value: String): void
+add(value: T): void
+getType(): Type
+getName(): String
+get(index: int): Value
+getSize(): int
+isObject(): boolean
+isArray(): boolean
+isElement(): boolean
+isPair(): boolean
+isArrayObjectElement(): boolean
+toString(): String
Other
(from Storage)
−value: String
«constructor»+Other(value: String)
+toString(): String
ArrayObjectListener
(from Graphics)
-frame: Frame {readOnly}
«constructor»+ArrayObjectListener(line: Line, frame: Frame)
+mouseClicked(e: MouseEvent): void
+mousePressed(e: MouseEvent): void
+mouseReleased(e: MouseEvent): void
+mouseEntered(e: MouseEvent): void
+mouseExited(e: MouseEvent): void
MouseListener