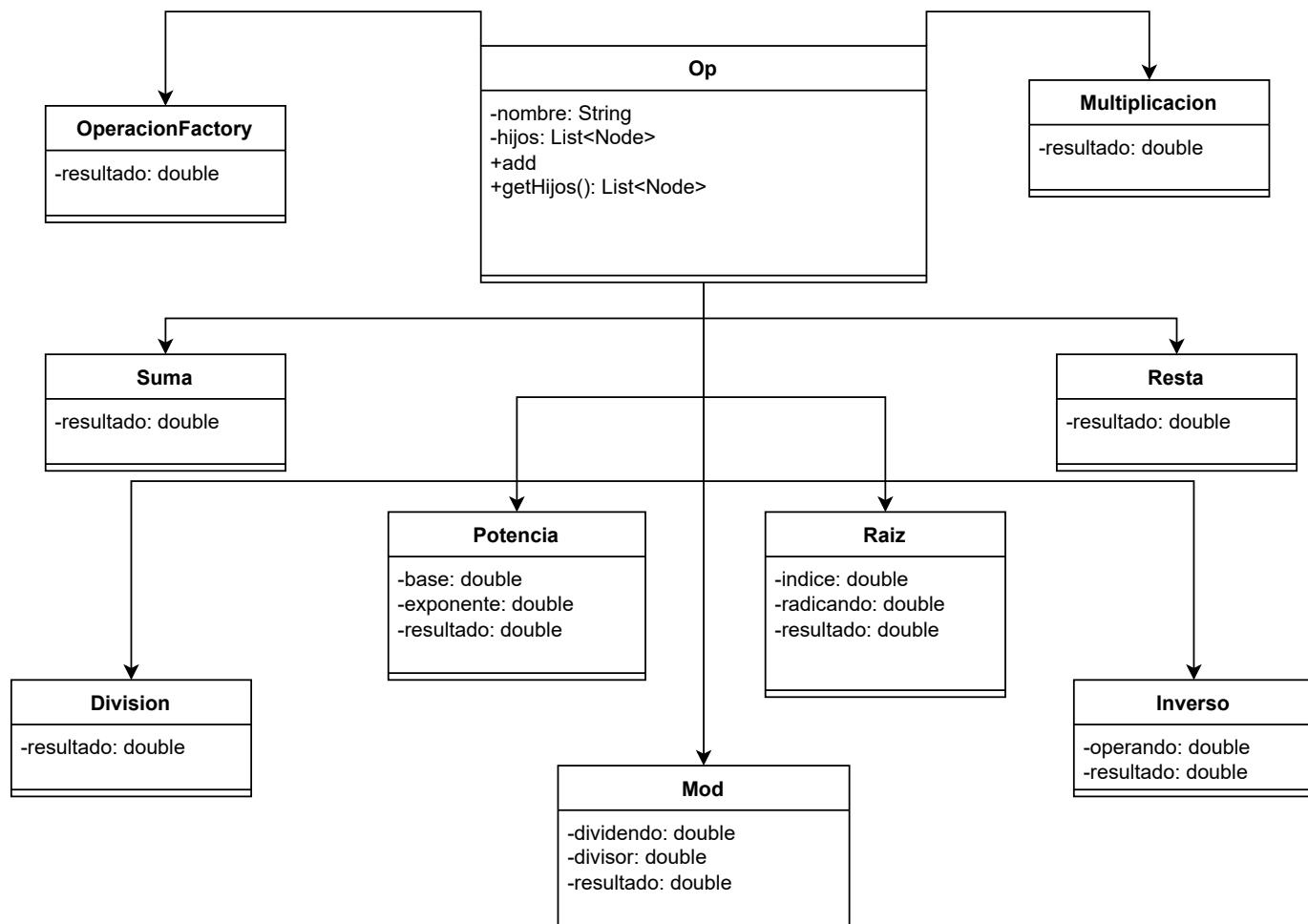


Lexer
<ul style="list-style-type: none"> <li>- input: String</li> <li>- idx: int</li> <li>- line: int</li> <li>- col: int</li> </ul>
<ul style="list-style-type: none"> <li>+ Lexer(String entrada)</li> <li>+ analizar(): List&lt;Token&gt;</li> <li>- siguienteCaracter(): char</li> <li>- retroceder()</li> <li>- agregarToken(String tipo, String valor)</li> <li>+ obtenerTokens(): List&lt;Token&gt;</li> <li>+ obtenerErrores(): List&lt;String&gt;</li> </ul>

Num
<ul style="list-style-type: none"> <li>-value: double</li> <li>+evaluate(): double</li> <li>+render(): String</li> </ul> <p>+ Num(double valor)</p> <ul style="list-style-type: none"> <li>+ getValor(): double</li> <li>+ setValor(double nuevoValor)</li> <li>+ getTipoDatos(): String</li> <li>+ toString(): String // Devuelve el número como cadena.</li> </ul>

Token
<ul style="list-style-type: none"> <li>+ type: TokenType</li> <li>+ lexeme: String</li> <li>+ line: int</li> <li>+ columna: int</li> </ul> <p>+ Token(String tipo, String valor, int fila, int columna)</p> <ul style="list-style-type: none"> <li>+ getTipo(): String</li> <li>+ getValor(): String</li> <li>+ getFila(): int</li> <li>+ getColumna(): int</li> <li>+ toString(): String</li> </ul>

Num
<ul style="list-style-type: none"> <li>- tokens: list&lt;Token&gt;</li> <li>- pos: int</li> </ul> <p>+ parsePrograma(): list&lt;Node&gt;</p> <ul style="list-style-type: none"> <li>- parseElemento(): node</li> </ul> <p>+ Parser(List&lt;Token&gt; tokens)</p> <ul style="list-style-type: none"> <li>+ parsear(): Node</li> <li>- expresion(): Node</li> <li>- termino(): Node</li> <li>- factor(): Node</li> <li>- numero(): Node</li> </ul> <p>+ obtenerErrores(): List&lt;String&gt;</p>



<b>ArbolGrafico</b>
<ul style="list-style-type: none"> <li>-rutaDot: String</li> <li>-rutaimagen: String</li> <li>-contadorNodos: int</li> </ul>
+ generarDot(nodoRaiz: Node, rutaSalida: Path)

<b>Analizar</b>
<ul style="list-style-type: none"> <li>-parserAdapter: ParserAdapter</li> <li>-parser: Parser</li> <li>-tokens: List&lt;Token&gt;</li> <li>-arboles: List&lt;Node&gt;</li> </ul>
+ parsearArboles(tokens: List<lexer.Token>): List<ast.Node>

<b>OperacionResultado</b>
<ul style="list-style-type: none"> <li>- tipoOperacion: String</li> <li>- operandos: List&lt;Double&gt;</li> <li>- resultado: Double</li> <li>- mensaje: String</li> </ul>
+ getResumen(): String

<b>Main</b>
<ul style="list-style-type: none"> <li>-rutaEntrada: String</li> <li>-rutaSalida: String</li> <li>-lexer: Lexe</li> <li>-analizador: Analizar</li> <li>-htmlReport: HtmlReport</li> <li>-arbolGrafico: ArbolGrafico</li> </ul>
+ main(args: String[])

<b>InfixPretty</b>
-expresion: String
+ convertir(node: Node): String + limpiarExpresion(): String

<b>HtmlReport</b>
<ul style="list-style-type: none"> <li>-rutaSalida: String</li> <li>-nombreArchivo: String</li> <li>-errores: List&lt;String&gt;</li> <li>-resultados: List&lt;OperacionResultado&gt;</li> <li>-fechaGeneracion: LocalDateTime</li> </ul>
<ul style="list-style-type: none"> <li>+ HtmlReport(String rutaDestino)</li> <li>+ generarReport(List&lt;OperacionResultado&gt; resultados)</li> <li>+ agregarFila(String descripcion, String valor)</li> <li>+ guardarArchivo()</li> </ul>

<b>AppFrame</b>
<ul style="list-style-type: none"> <li>-txtEntrada: JTextArea</li> <li>-btnAnalizar: JButton</li> <li>-btnReporte: JButton</li> <li>-lblEstado: JLabel</li> <li>-analizador: Analizar</li> <li>-htmlReport: HtmlReport</li> <li>-arbolGrafico: ArbolGrafico</li> </ul>
+ inicializarComponentes() + ejecutarAnalisis() + mostrarResultados()

<b>GuiMain</b>
-ventanaPrincipal: AppFrame
+ main(args: String[])