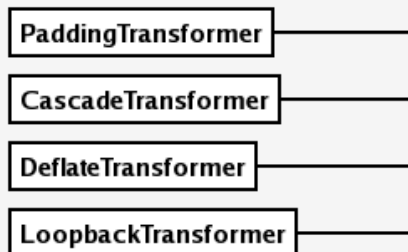


Operation
<u>+PRE_PROCESSING: Operation = new Operation(1)</u>
<u>+POST_PROCESSING: Operation = new Operation(2)</u>

Direction
<u>+FORWARD: Direction = new Direction(1)</u>
<u>+REVERSED: Direction = new Direction(2)</u>



Assembly
<u>+DIRECTION: String = "gnu.crypto.assembly.assembly.direction"</u>
-wired: Direction
-head: Transformer
<u>+<<create>> Assembly()</u>
<u>+addPreTransformer(t:Transformer): void</u>
<u>+addPostTransformer(t:Transformer): void</u>
<u>+init(in attributes:Map): void</u>
<u>+reset(): void</u>
<u>+update(b:byte): byte[]</u>
<u>+update(in:byte[]): byte[]</u>
<u>+update(in:byte[],offset:int,length:int): byte[]</u>
<u>+lastUpdate(): byte[]</u>
<u>+lastUpdate(b:byte): byte[]</u>
<u>+lastUpdate(in:byte[]): byte[]</u>
<u>+lastUpdate(in:byte[],offset:int,length:int): byte[]</u>

Transformer
<u>+DIRECTION: String = "gnu.crypto.assembly.transformer.direction"</u>
#tail: Transformer
#wired: Direction
#mode: Operation
<u>#<<create>> Transformer()</u>
<u>+getCascadeTransformer(cascade:Cascade): Transformer</u>
<u>+getPaddingTransformer(padding:IPad): Transformer</u>
<u>+getDeflateTransformer(): Transformer</u>
<u>+setMode(mode:Operation): void</u>
<u>+isPreprocessing(): boolean</u>
<u>+isPostProcessing(): boolean</u>
<u>+init(in attributes:Map): void</u>
<u>+currentBlockSize(): int</u>
<u>+reset(): void</u>
<u>+update(b:byte): byte[]</u>
<u>+update(in:byte[]): byte[]</u>
<u>+update(in:byte[],offset:int,length:int): byte[]</u>
<u>+lastUpdate(): byte[]</u>
<u>+lastUpdate(b:byte): byte[]</u>
<u>+lastUpdate(in:byte[]): byte[]</u>
<u>+lastUpdate(in:byte[],offset:int,length:int): byte[]</u>
<u>initDelegate(attributes:Map): void</u>
<u>delegateBlockSize(): int</u>
<u>resetDelegate(): void</u>
<u>updateDelegate(in:byte[],off:int,len:int): byte[]</u>
<u>lastUpdateDelegate(): byte[]</u>

TransformerException
<u>+<<create>> TransformerException()</u>
<u>+<<create>> TransformerException(detail:String)</u>
<u>+<<create>> TransformerException(cause:Throwable)</u>
<u>+<<create>> TransformerException(detail:String,cause:Throwable)</u>
<u>+getCause(): Throwable</u>
<u>+printStackTrace(): void</u>
<u>+printStackTrace(ps:PrintStream): void</u>
<u>+toString(): String</u>