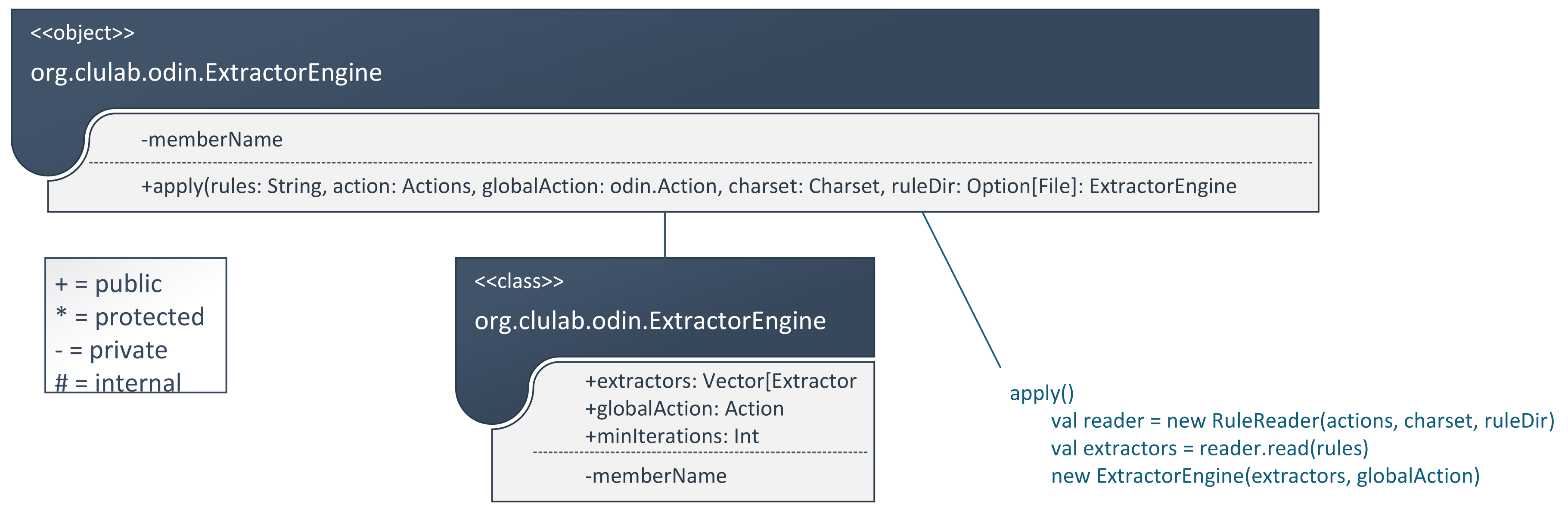
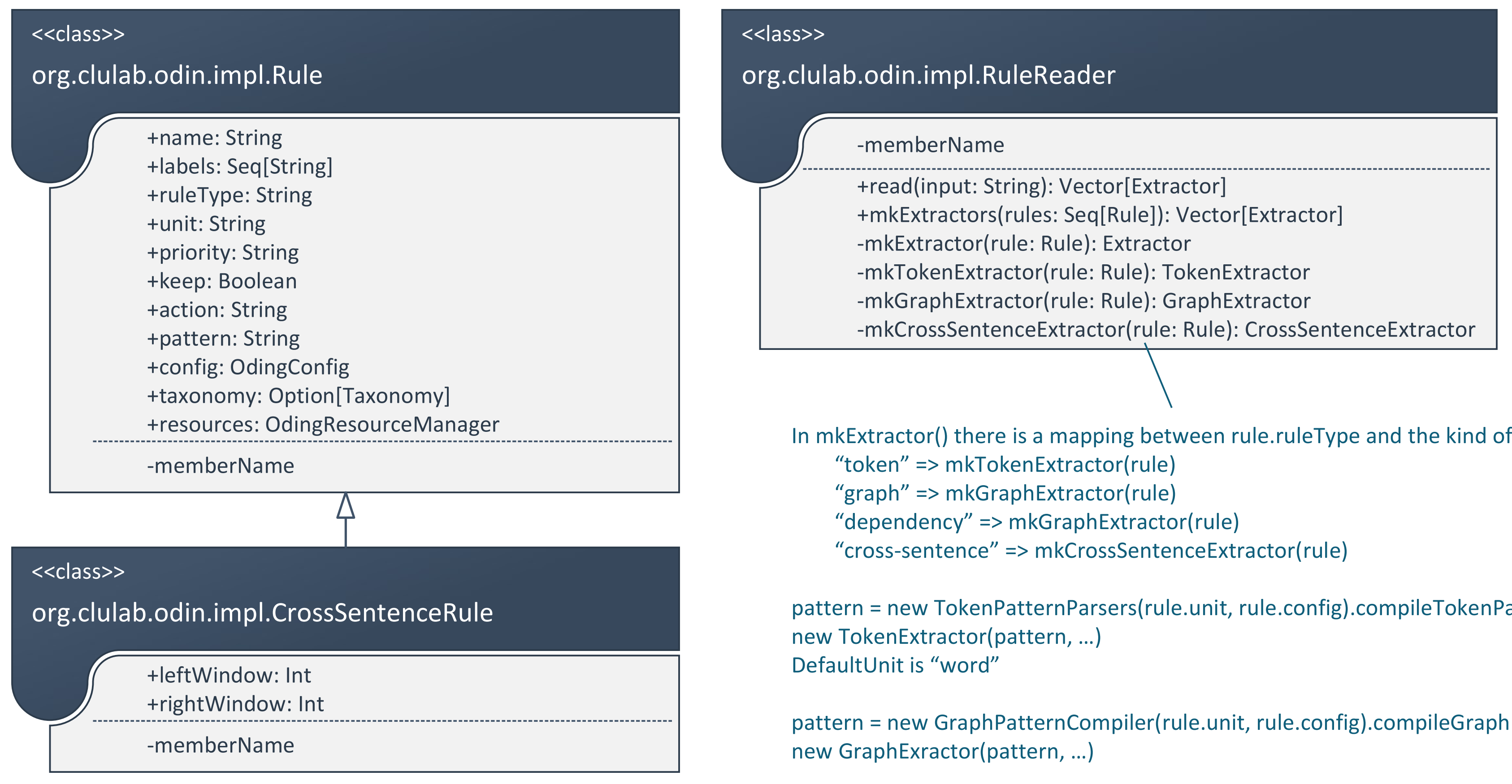

<<object>>
org.clulab.odin.ExtractorEngine
-memberName
+apply(rules: String, action: Actions, globalAction: odin.Action, charset: Charset, ruleDir: Option[File]: ExtractorEngine
<<class>>
org.clulab.odin.ExtractorEngine
+extractors: Vector[Extractor
+globalAction: Action
+minIterations: Int
-memberName
+ = public
* = protected
- = private
# = internal

```
apply()
     val reader = new RuleReader(actions, charset, ruleDir)
     val extractors = reader.read(rules)
     new ExtractorEngine(extractors, globalAction)
```

ExtractorEngine class



<<class>>
org.clulab.odin.impl.Rule
+name: String
+labels: Seq[String]
+ruleType: String
+unit: String
+priority: String
+keep: Boolean
+action: String
+pattern: String
+config: OdingConfig
+taxonomy: Option[Taxonomy]
+resources: OdingResourceManager
-memberName
<<class>>
org.clulab.odin.impl.CrossSentenceRule
+leftWindow: Int
+rightWindow: Int
-memberName
<<lass>>
org.clulab.odin.impl.RuleReader
-memberName
+read(input: String): Vector[Extractor]
+mkExtractors(rules: Seq[Rule]): Vector[Extractor]
-mkExtractor(rule: Rule): Extractor
-mkTokenExtractor(rule: Rule): TokenExtractor
-mkGraphExtractor(rule: Rule): GraphExtractor
-mkCrossSentenceExtractor(rule: Rule): CrossSentenceExtractor

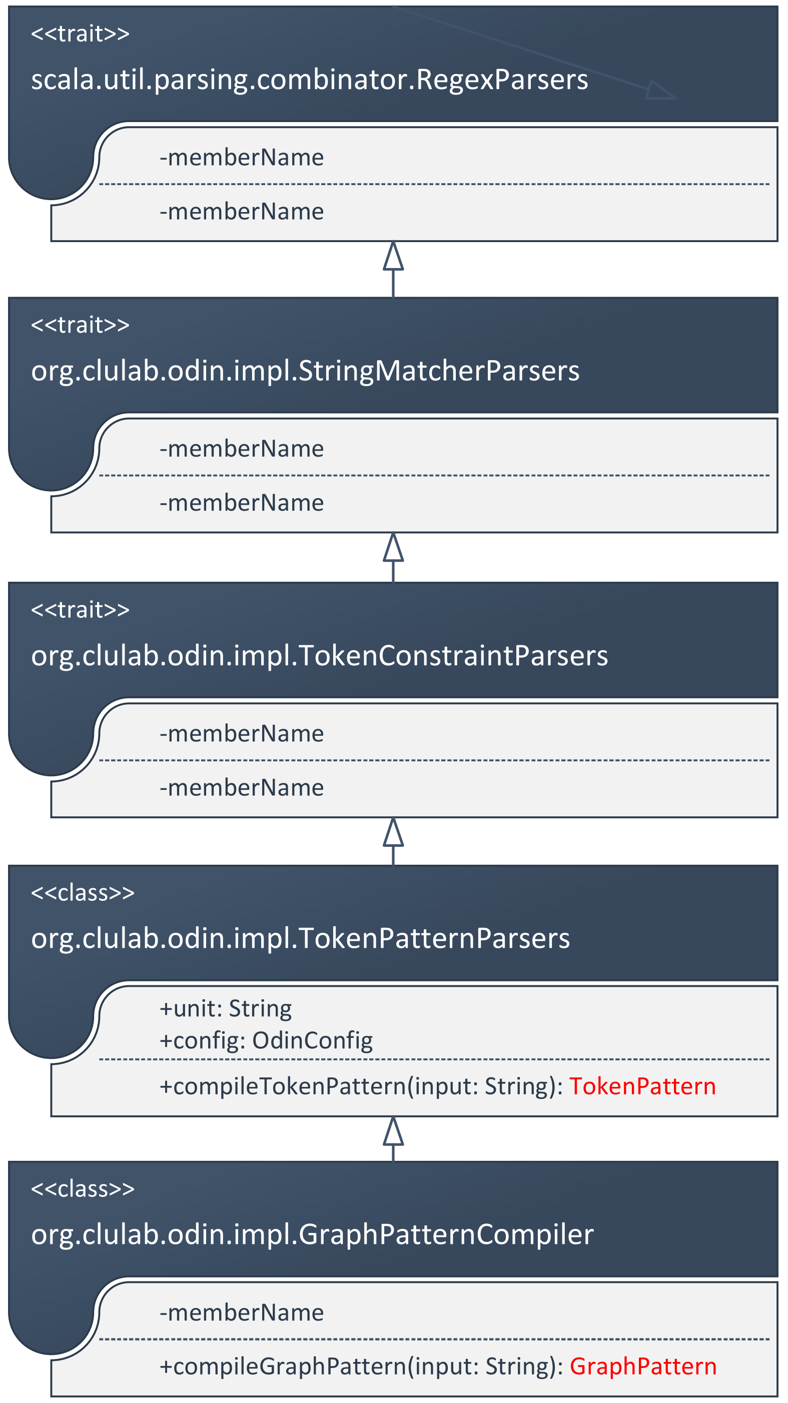
In mkExtractor() there is a mapping between rule.ruleType and the kind of extractor produced:
“token” => mkTokenExtractor(rule)
“graph” => mkGraphExtractor(rule)
“dependency” => mkGraphExtractor(rule)
“cross-sentence” => mkCrossSentenceExtractor(rule)

pattern = new TokenPatternParsers(rule.unit, rule.config).compileTokenPattern(rule.pattern)
new TokenExtractor(pattern, …)
DefaultUnit is “word”

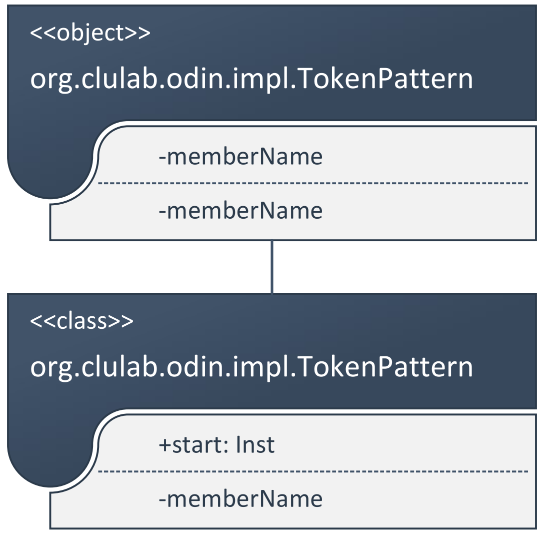
pattern = new GraphPatternCompiler(rule.unit, rule.config).compileGraphPattern(rule.pattern)
new GraphExractor(pattern, …)

newCrossSentenceExtractor

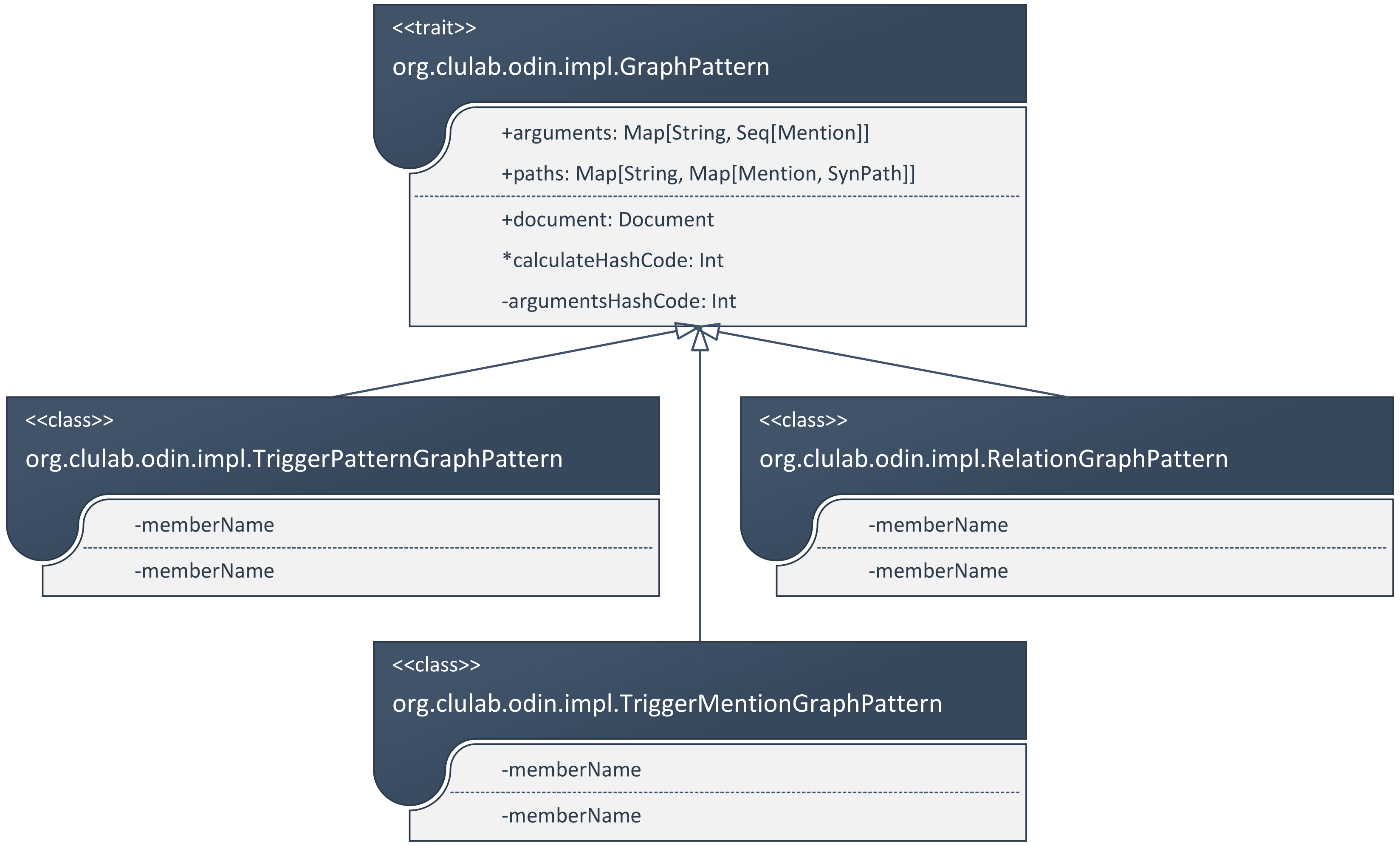
Rule class


<<trait>>
scala.util.parsing.combinator.RegexParsers
-memberName
-memberName
<<trait>>
org.clulab.odin.impl.StringMatcherParsers
-memberName
-memberName
<<trait>>
org.clulab.odin.impl.TokenConstraintParsers
-memberName
-memberName
<<class>>
org.clulab.odin.impl.TokenPatternParsers
+unit: String
+config: OdinConfig
+compileTokenPattern(input: String): TokenPattern
<<class>>
org.clulab.odin.impl.GraphPatternCompiler
-memberName
+compileGraphPattern(input: String): GraphPattern

Parsers/Compiler class

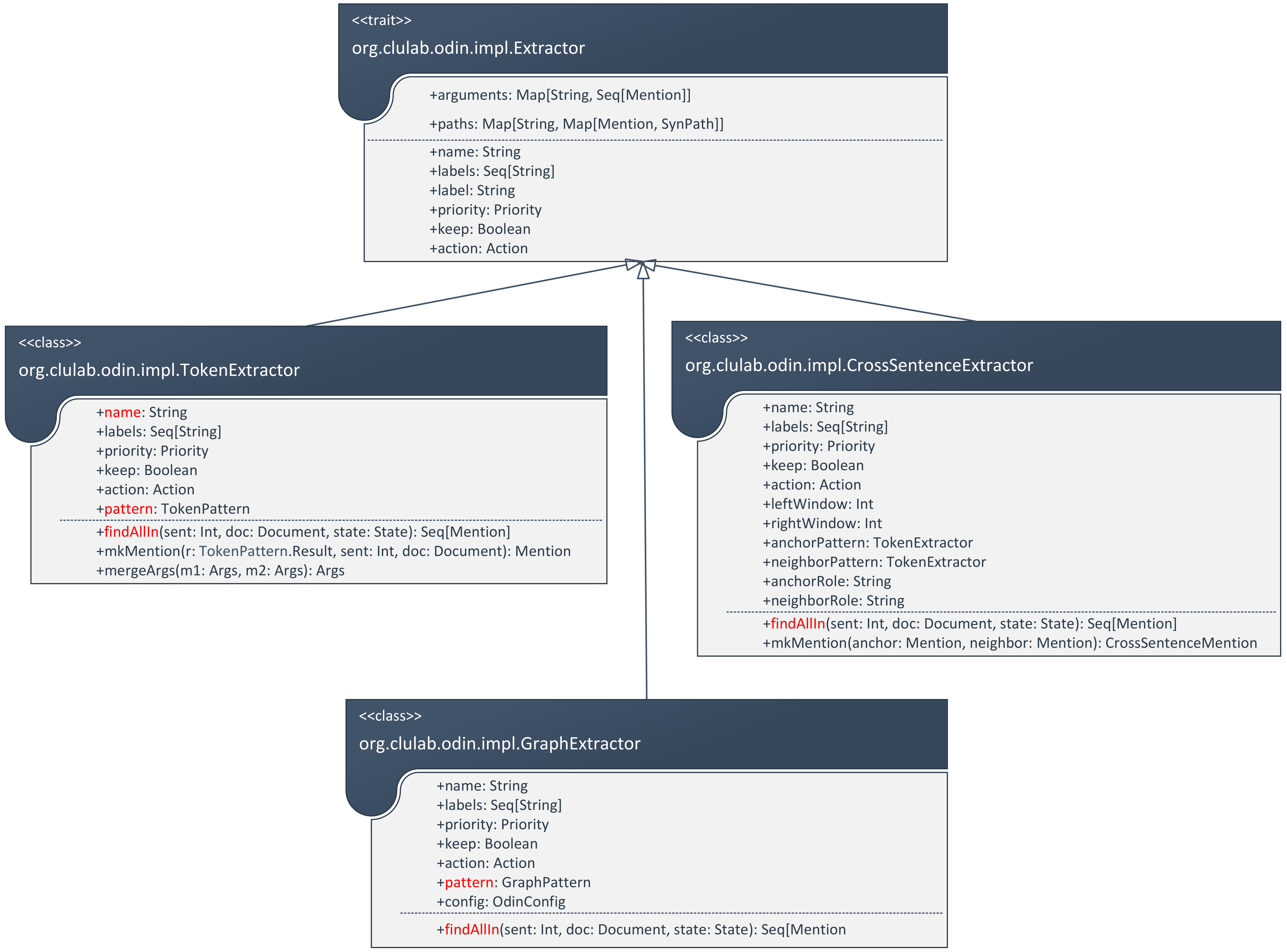

<<object>>
org.clulab.odin.impl.TokenPattern
-memberName
-memberName
<<class>>
org.clulab.odin.impl.TokenPattern
+start: Inst
-memberName

TokenPattern class

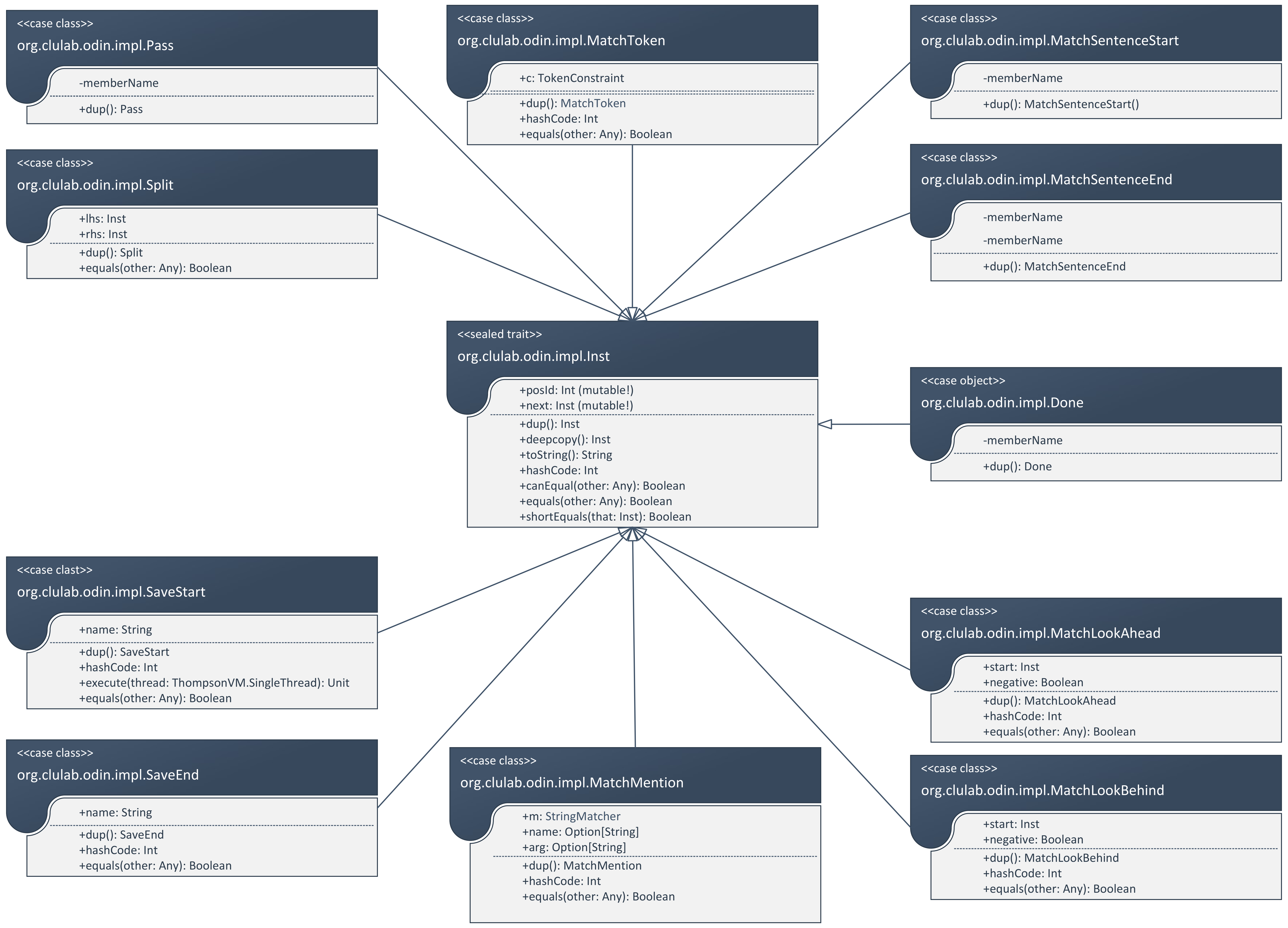


<<trait>>
org.clulab.odin.impl.GraphPattern
+arguments: Map[String, Seq[Mention]]
+paths: Map[String, Map[Mention, SynPath]]
+document: Document
*calculateHashCode: Int
-argumentsHashCode: Int
<<class>>
org.clulab.odin.impl.TriggerPatternGraphPattern
-memberName
-memberName
<<class>>
org.clulab.odin.impl.RelationGraphPattern
-memberName
-memberName
<<class>>
org.clulab.odin.impl.TriggerMentionGraphPattern
-memberName
-memberName

GraphPattern class

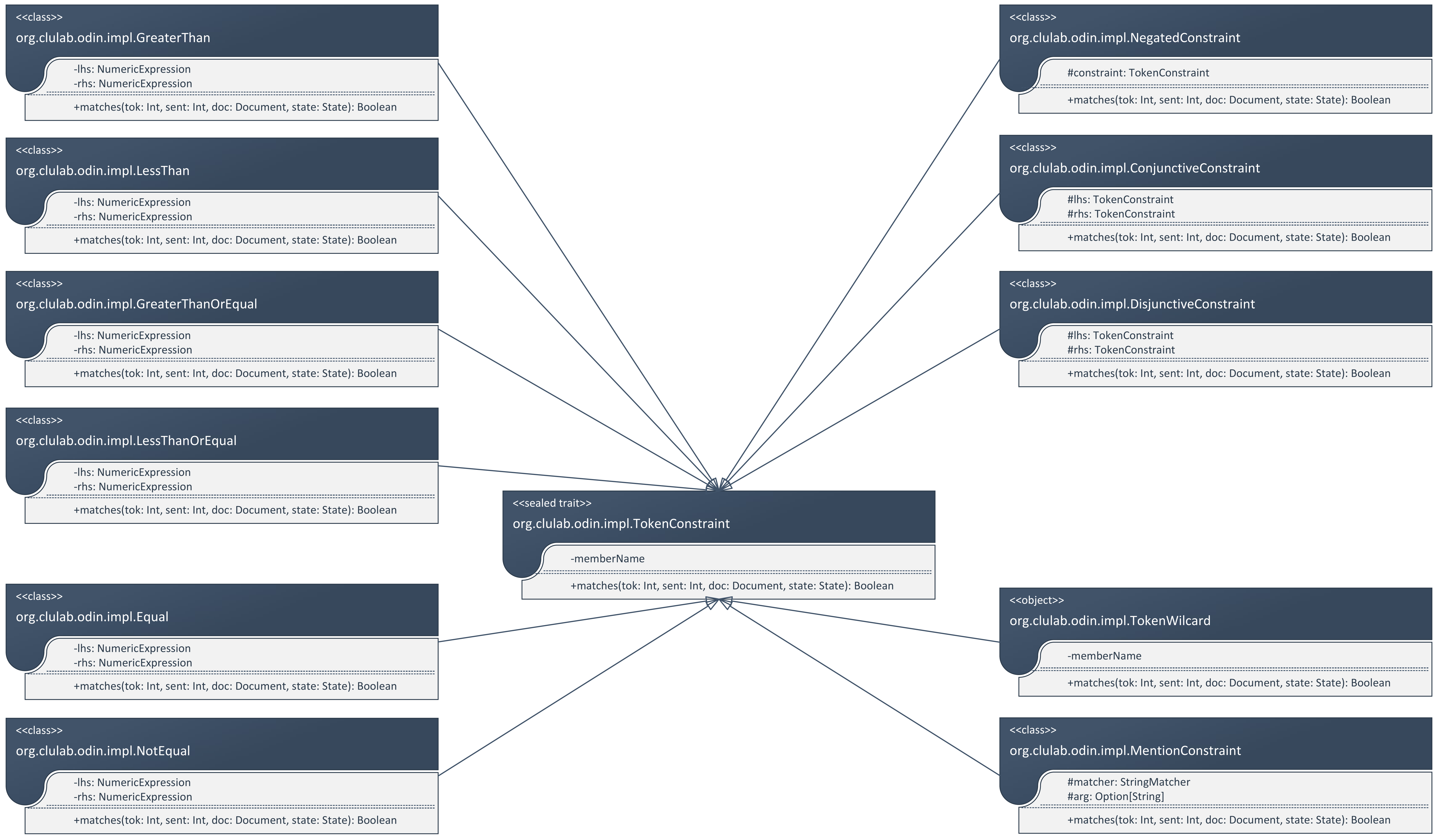

<<trait>>
org.clulab.odin.impl.Extractor
+arguments: Map[String, Seq[Mention]]
+paths: Map[String, Map[Mention, SynPath]]
+name: String
+labels: Seq[String]
+label: String
+priority: Priority
+keep: Boolean
+action: Action
<<class>>
org.clulab.odin.impl.TokenExtractor
+name: String
+labels: Seq[String]
+priority: Priority
+keep: Boolean
+action: Action
+pattern: TokenPattern
+findAllIn(sent: Int, doc: Document, state: State): Seq[Mention]
+mkMention(r: TokenPattern.Result, sent: Int, doc: Document): Mention
+mergeArgs(m1: Args, m2: Args): Args
<<class>>
org.clulab.odin.impl.CrossSentenceExtractor
+name: String
+labels: Seq[String]
+priority: Priority
+keep: Boolean
+action: Action
+leftWindow: Int
+rightWindow: Int
+anchorPattern: TokenExtractor
+neighborPattern: TokenExtractor
+anchorRole: String
+neighborRole: String
+findAllIn(sent: Int, doc: Document, state: State): Seq[Mention]
+mkMention(anchor: Mention, neighbor: Mention): CrossSentenceMention
<<class>>
org.clulab.odin.impl.GraphExtractor
+name: String
+labels: Seq[String]
+priority: Priority
+keep: Boolean
+action: Action
+pattern: GraphPattern
+config: OdinConfig
+findAllIn(sent: Int, doc: Document, state: State): Seq[Mention

Extractor class

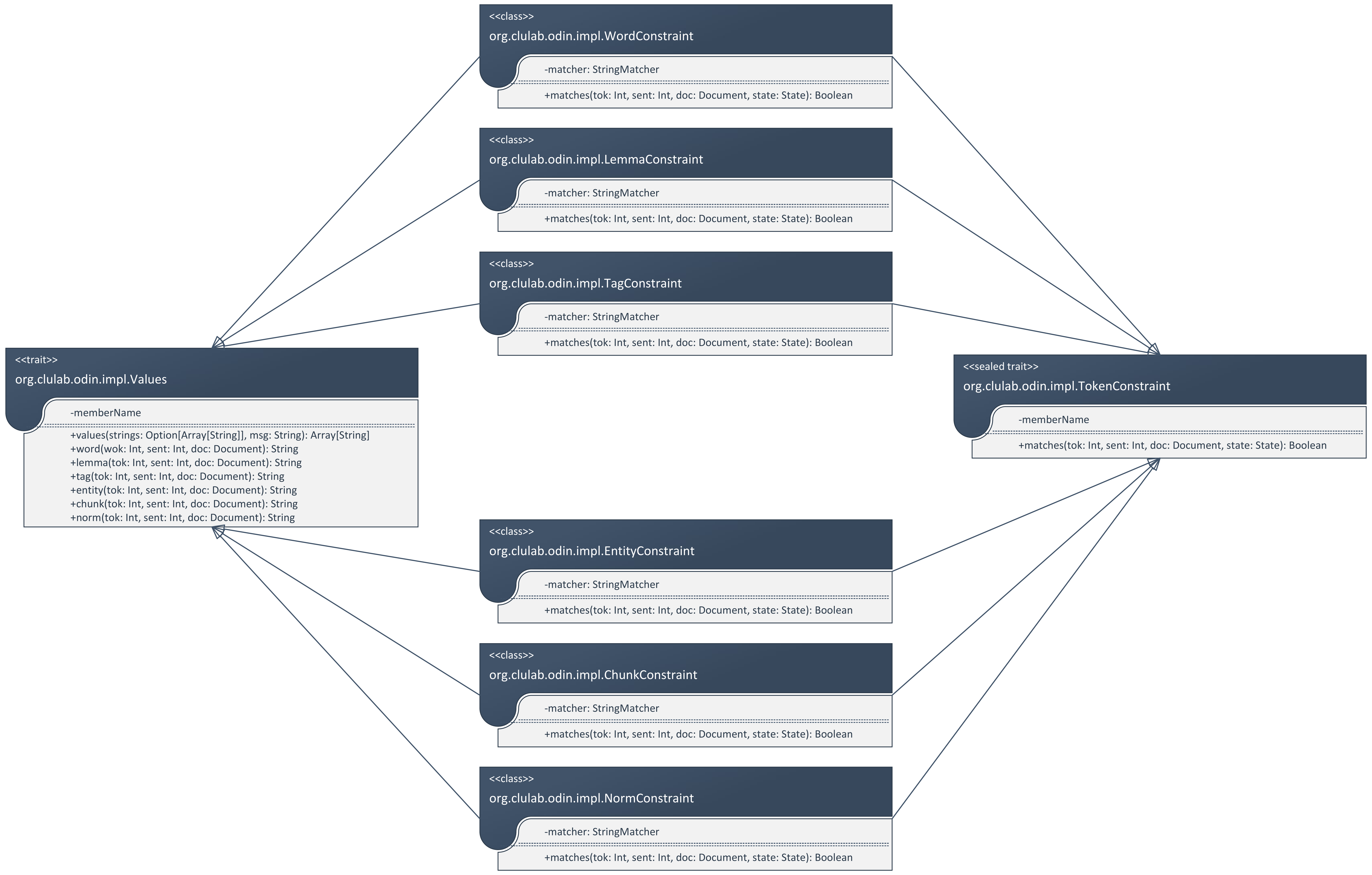


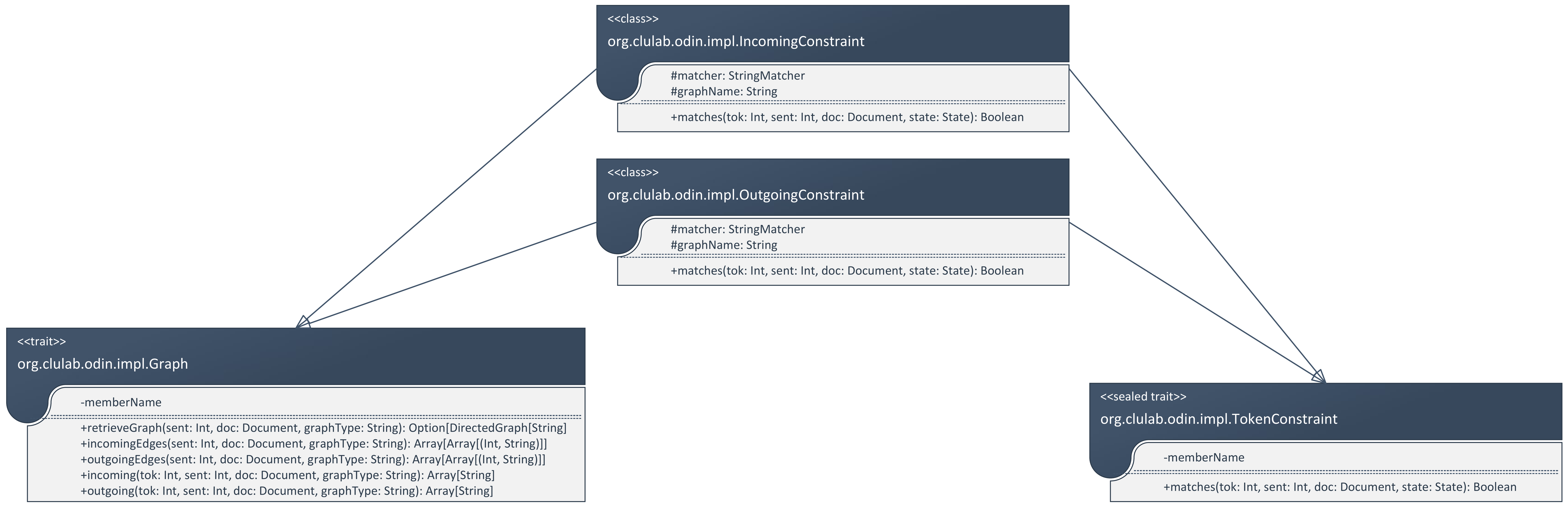
<<case class>>
org.clulab.odin.impl.Pass
-memberName
+dup(): Pass
<<case class>>
org.clulab.odin.impl.MatchToken
+c: TokenConstraint
+dup(): MatchToken
+hashCode: Int
+equals(other: Any): Boolean
<<case class>>
org.clulab.odin.impl.MatchSentenceStart
-memberName
+dup(): MatchSentenceStart()
<<case class>>
org.clulab.odin.impl.Split
+lhs: Inst
+rhs: Inst
+dup(): Split
+equals(other: Any): Boolean
<<case class>>
org.clulab.odin.impl.MatchSentenceEnd
-memberName
-memberName
+dup(): MatchSentenceEnd
<<sealed trait>>
org.clulab.odin.impl.Inst
+posId: Int (mutable!)
+next: Inst (mutable!)
+dup(): Inst
+deepcopy(): Inst
+toString(): String
+hashCode: Int
+canEqual(other: Any): Boolean
+equals(other: Any): Boolean
+shortEquals(that: Inst): Boolean
<<case object>>
org.clulab.odin.impl.Done
-memberName
+dup(): Done
<<case clast>>
org.clulab.odin.impl.SaveStart
+name: String
+dup(): SaveStart
+hashCode: Int
+execute(thread: ThompsonVM.SingleThread): Unit
+equals(other: Any): Boolean
<<case class>>
org.clulab.odin.impl.MatchLookAhead
+start: Inst
+negative: Boolean
+dup(): MatchLookAhead
+hashCode: Int
+equals(other: Any): Boolean
<<case class>>
org.clulab.odin.impl.SaveEnd
+name: String
+dup(): SaveEnd
+hashCode: Int
+equals(other: Any): Boolean
<<case class>>
org.clulab.odin.impl.MatchMention
+m: StringMatcher
+name: Option[String]
+arg: Option[String]
+dup(): MatchMention
+hashCode: Int
+equals(other: Any): Boolean
<<case class>>
org.clulab.odin.impl.MatchLookBehind
+start: Inst
+negative: Boolean
+dup(): MatchLookBehind
+hashCode: Int
+equals(other: Any): Boolean

Inst class

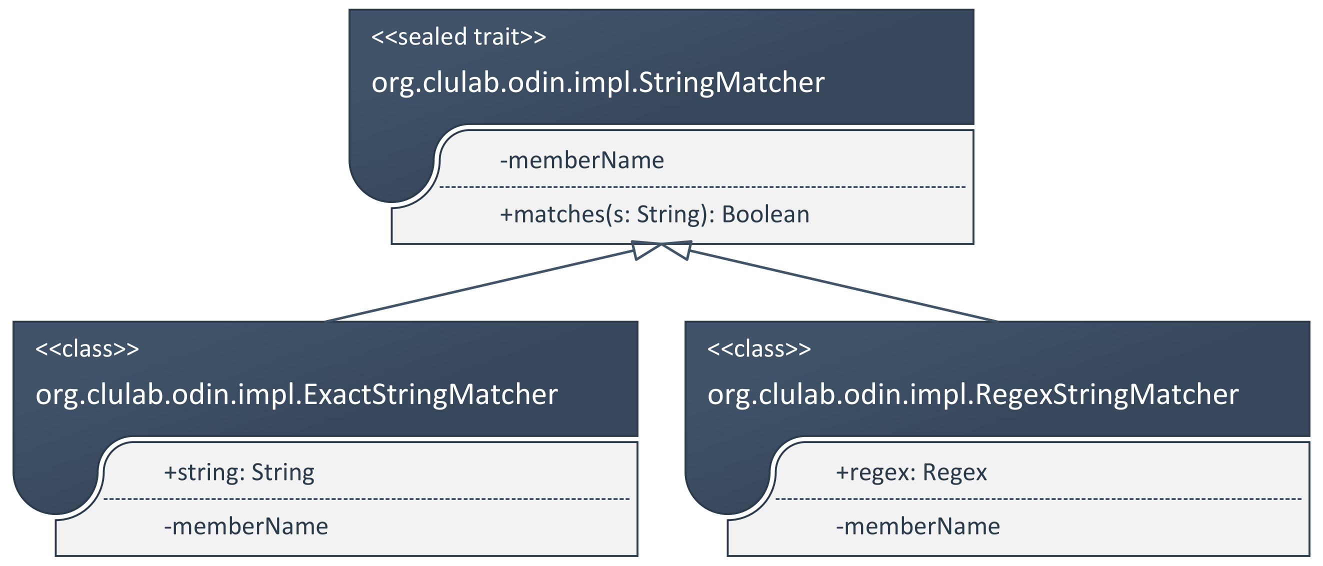

<<class>>
org.clulab.odin.impl.GreaterThan
-lhs: NumericExpression
-rhs: NumericExpression
+matches(tok: Int, sent: Int, doc: Document, state: State): Boolean
<<class>>
org.clulab.odin.impl.LessThan
-lhs: NumericExpression
-rhs: NumericExpression
+matches(tok: Int, sent: Int, doc: Document, state: State): Boolean
<<class>>
org.clulab.odin.impl.GreaterThanOrEqual
-lhs: NumericExpression
-rhs: NumericExpression
+matches(tok: Int, sent: Int, doc: Document, state: State): Boolean
<<class>>
org.clulab.odin.impl.LessThanOrEqual
-lhs: NumericExpression
-rhs: NumericExpression
+matches(tok: Int, sent: Int, doc: Document, state: State): Boolean
<<class>>
org.clulab.odin.impl.Equal
-lhs: NumericExpression
-rhs: NumericExpression
+matches(tok: Int, sent: Int, doc: Document, state: State): Boolean
<<class>>
org.clulab.odin.impl.NotEqual
-lhs: NumericExpression
-rhs: NumericExpression
+matches(tok: Int, sent: Int, doc: Document, state: State): Boolean
<<sealed trait>>
org.clulab.odin.impl.TokenConstraint
-memberName
+matches(tok: Int, sent: Int, doc: Document, state: State): Boolean
<<class>>
org.clulab.odin.impl.NegatedConstraint
#constraint: TokenConstraint
+matches(tok: Int, sent: Int, doc: Document, state: State): Boolean
<<class>>
org.clulab.odin.impl.ConjunctiveConstraint
#lhs: TokenConstraint
#rhs: TokenConstraint
+matches(tok: Int, sent: Int, doc: Document, state: State): Boolean
<<class>>
org.clulab.odin.impl.DisjunctiveConstraint
#lhs: TokenConstraint
#rhs: TokenConstraint
+matches(tok: Int, sent: Int, doc: Document, state: State): Boolean
<<object>>
org.clulab.odin.impl.TokenWilcard
-memberName
+matches(tok: Int, sent: Int, doc: Document, state: State): Boolean
<<class>>
org.clulab.odin.impl.MentionConstraint
#matcher: StringMatcher
#arg: Option[String]
+matches(tok: Int, sent: Int, doc: Document, state: State): Boolean

TokenConstraint

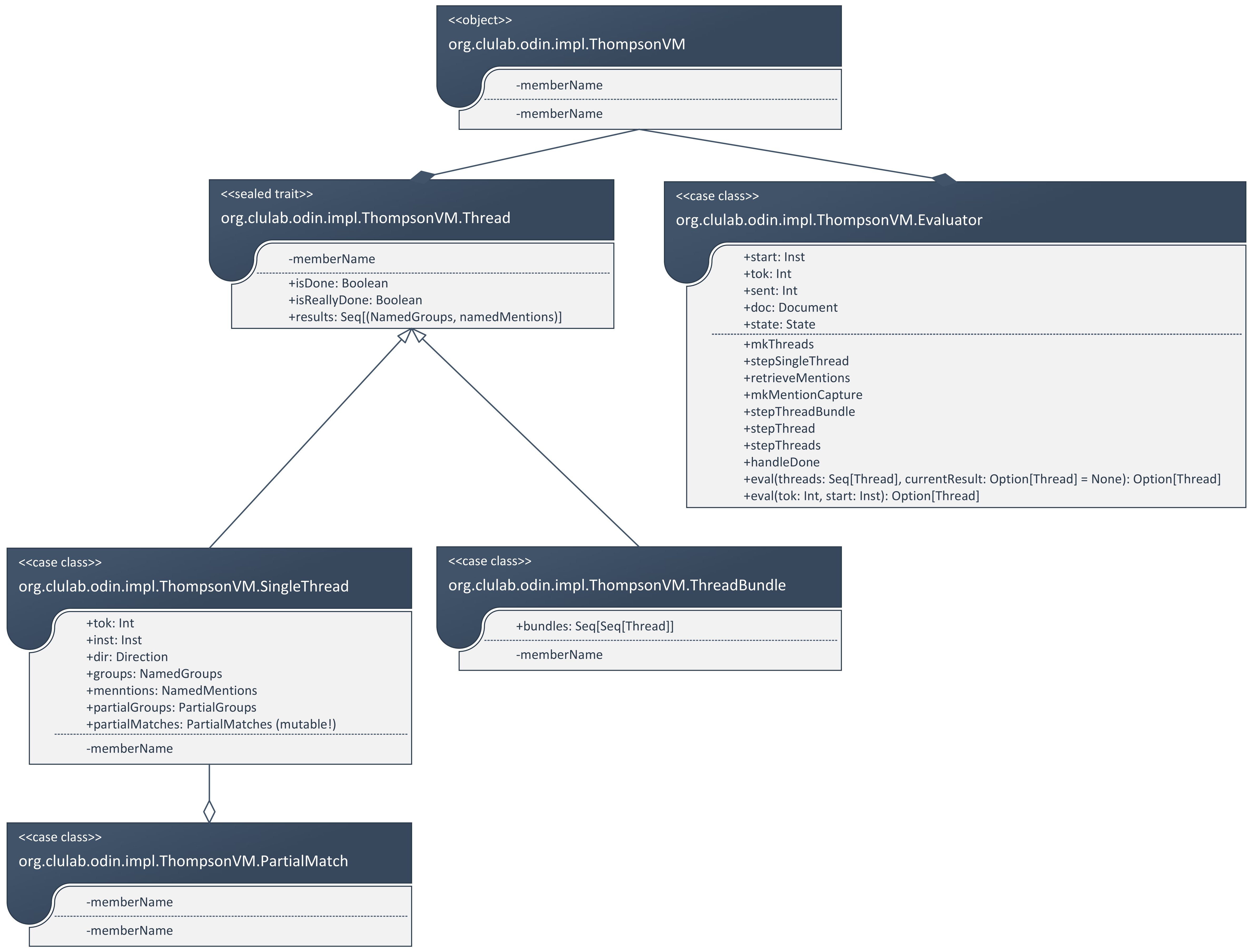

<<class>>
org.clulab.odin.impl.WordConstraint
-matcher: StringMatcher
+matches(tok: Int, sent: Int, doc: Document, state: State): Boolean
<<class>>
org.clulab.odin.impl.LemmaConstraint
-matcher: StringMatcher
+matches(tok: Int, sent: Int, doc: Document, state: State): Boolean
<<class>>
org.clulab.odin.impl.TagConstraint
-matcher: StringMatcher
+matches(tok: Int, sent: Int, doc: Document, state: State): Boolean
<<trait>>
org.clulab.odin.impl.Values
-memberName
+values(strings: Option[Array[String]], msg: String): Array[String]
+word(wok: Int, sent: Int, doc: Document): String
+lemma(tok: Int, sent: Int, doc: Document): String
+tag(tok: Int, sent: Int, doc: Document): String
+entity(tok: Int, sent: Int, doc: Document): String
+chunk(tok: Int, sent: Int, doc: Document): String
+norm(tok: Int, sent: Int, doc: Document): String
<<sealed trait>>
org.clulab.odin.impl.TokenConstraint
-memberName
+matches(tok: Int, sent: Int, doc: Document, state: State): Boolean
<<class>>
org.clulab.odin.impl.EntityConstraint
-matcher: StringMatcher
+matches(tok: Int, sent: Int, doc: Document, state: State): Boolean
<<class>>
org.clulab.odin.impl.ChunkConstraint
-matcher: StringMatcher
+matches(tok: Int, sent: Int, doc: Document, state: State): Boolean
<<class>>
org.clulab.odin.impl.NormConstraint
-matcher: StringMatcher
+matches(tok: Int, sent: Int, doc: Document, state: State): Boolean


<<class>>
org.clulab.odin.impl.IncomingConstraint
#matcher: StringMatcher
#graphName: String
+matches(tok: Int, sent: Int, doc: Document, state: State): Boolean
<<class>>
org.clulab.odin.impl.OutgoingConstraint
#matcher: StringMatcher
#graphName: String
+matches(tok: Int, sent: Int, doc: Document, state: State): Boolean
<<trait>>
org.clulab.odin.impl.Graph
-memberName
+retrieveGraph(sent: Int, doc: Document, graphType: String): Option[DirectedGraph[String]
+incomingEdges(sent: Int, doc: Document, graphType: String): Array[Array[(Int, String)]]
+outgoingEdges(sent: Int, doc: Document, graphType: String): Array[Array[(Int, String)]]
+incoming(tok: Int, sent: Int, doc: Document, graphType: String): Array[String]
+outgoing(tok: Int, sent: Int, doc: Document, graphType: String): Array[String]
<<sealed trait>>
org.clulab.odin.impl.TokenConstraint
-memberName
+matches(tok: Int, sent: Int, doc: Document, state: State): Boolean

Token&GraphConstraint

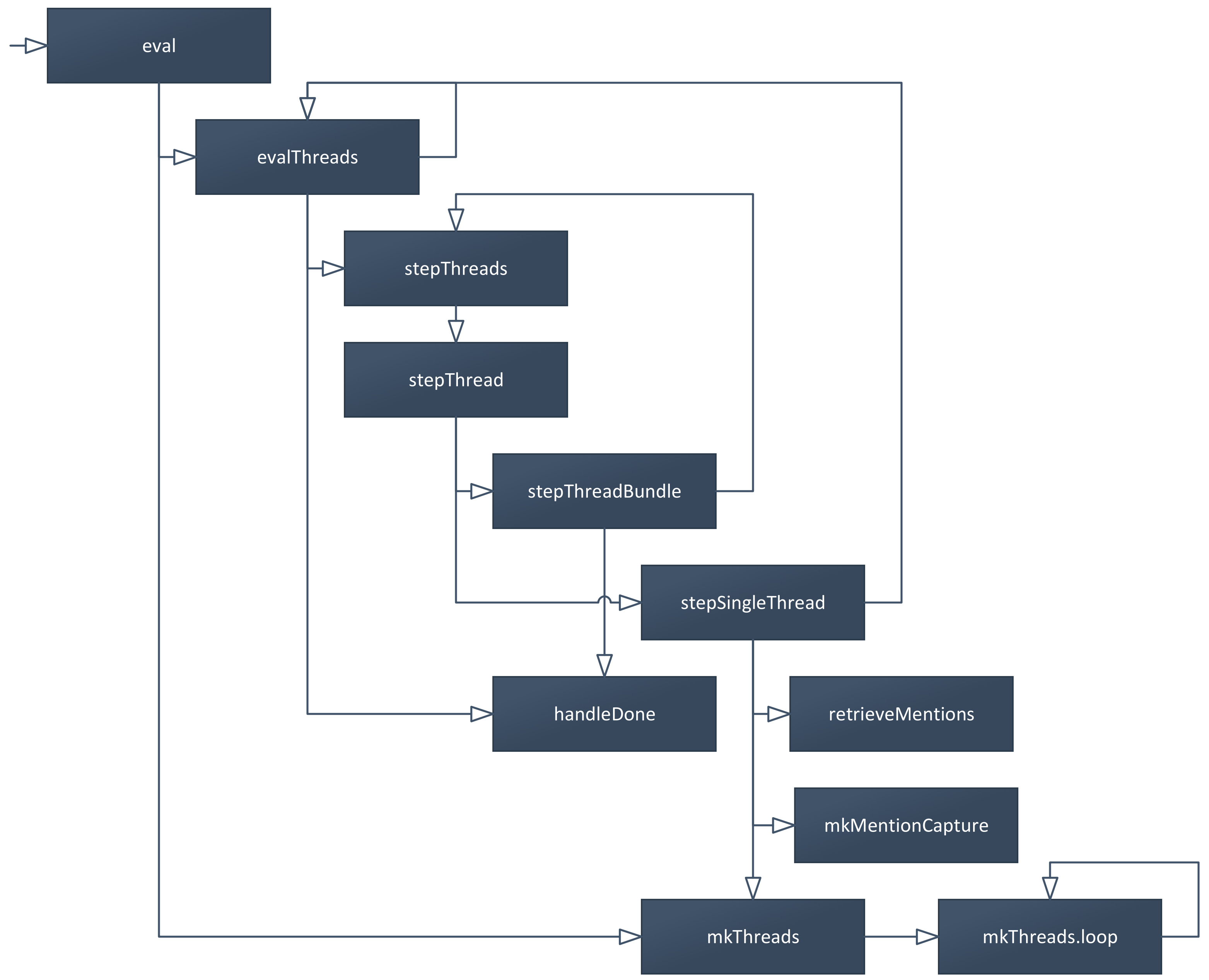

<<sealed trait>>
org.clulab.odin.impl.StringMatcher
-memberName
+matches(s: String): Boolean
<<class>>
org.clulab.odin.impl.ExactStringMatcher
+string: String
-memberName
<<class>>
org.clulab.odin.impl.RegexStringMatcher
+regex: Regex
-memberName

StringMatcher class



<<object>>
org.clulab.odin.impl.ThompsonVM
-memberName
-memberName
<<sealed trait>>
org.clulab.odin.impl.ThompsonVM.Thread
-memberName
+isDone: Boolean
+isReallyDone: Boolean
+results: Seq[(NamedGroups, namedMentions)]
<<case class>>
org.clulab.odin.impl.ThompsonVM.Evaluator
+start: Inst
+tok: Int
+sent: Int
+doc: Document
+state: State
+mkThreads
+stepSingleThread
+retrieveMentions
+mkMentionCapture
+stepThreadBundle
+stepThread
+stepThreads
+handleDone
+eval(threads: Seq[Thread], currentResult: Option[Thread] = None): Option[Thread]
+eval(tok: Int, start: Inst): Option[Thread]
<<case class>>
org.clulab.odin.impl.ThompsonVM.SingleThread
+tok: Int
+inst: Inst
+dir: Direction
+groups: NamedGroups
+menntions: NamedMentions
+partialGroups: PartialGroups
+partialMatches: PartialMatches (mutable!)
-memberName
<<case class>>
org.clulab.odin.impl.ThompsonVM.ThreadBundle
+bundles: Seq[Seq[Thread]]
-memberName
<<case class>>
org.clulab.odin.impl.ThompsonVM.PartialMatch
-memberName
-memberName

ThompsonVM class



Evaluator methods

Debugger class