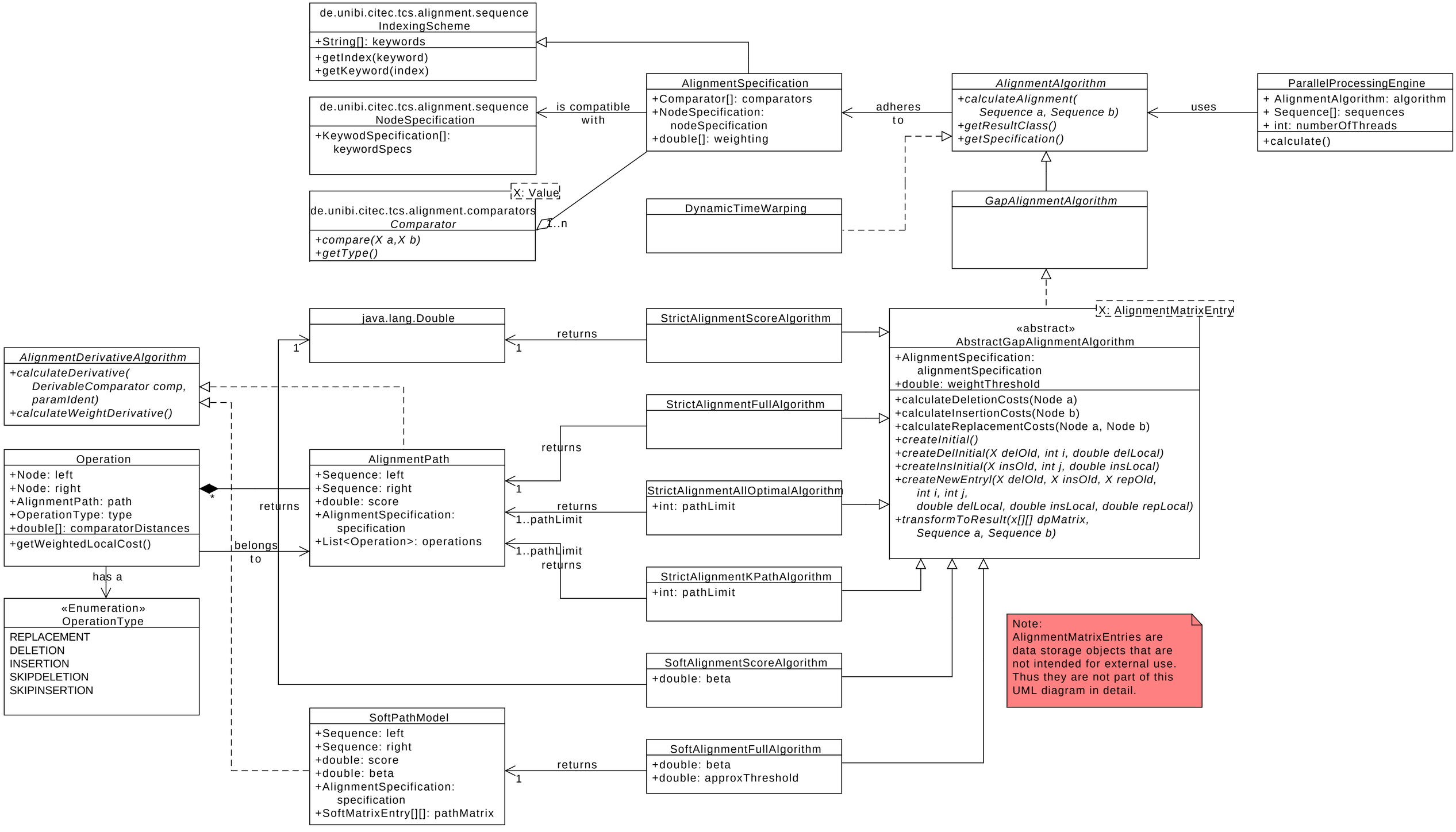
de.unibi.citec.tcs.alignment.sequence
IndexingScheme
+String[]: keywords
+getIndex(keyword)
+getKeyword(index)
de.unibi.citec.tcs.alignment.sequence
NodeSpecification
+KeywodSpecification[]:
keywordSpecs
X: Value
de.unibi.citec.tcs.alignment.comparators
Comparator
+compare(X a,X b)
+getType()
1..n
is compatible
with
AlignmentSpecification
+Comparator[]: comparators
+NodeSpecification:
nodeSpecification
+double[]: weighting
adheres
to
AlignmentAlgorithm
+calculateAlignment(
Sequence a, Sequence b)
+getResultClass()
+getSpecification()
uses
ParallelProcessingEngine
+ AlignmentAlgorithm: algorithm
+ Sequence[]: sequences
+ int: numberOfThreads
+calculate()
DynamicTimeWarping
GapAlignmentAlgorithm
java.lang.Double
returns
1
1
StrictAlignmentScoreAlgorithm
X: AlignmentMatrixEntry
«abstract»
AbstractGapAlignmentAlgorithm
+AlignmentSpecification:
alignmentSpecification
+double: weightThreshold
+calculateDeletionCosts(Node a)
+calculateInsertionCosts(Node b)
+calculateReplacementCosts(Node a, Node b)
+createInitial()
+createDelInitial(X delOld, int i, double delLocal)
+createInsInitial(X insOld, int j, double insLocal)
+createNewEntryI(X delOld, X insOld, X repOld,
int i, int j,
double delLocal, double insLocal, double repLocal)
+transformToResult(x[][] dpMatrix,
Sequence a, Sequence b)
AlignmentDerivativeAlgorithm
+calculateDerivative(
DerivableComparator comp,
paramIdent)
+calculateWeightDerivative()
StrictAlignmentFullAlgorithm
returns
Operation
+Node: left
+Node: right
+AlignmentPath: path
+OperationType: type
+double[]: comparatorDistances
+getWeightedLocalCost()
*
returns
belongs
to
AlignmentPath
+Sequence: left
+Sequence: right
+double: score
+AlignmentSpecification:
specification
+List<Operation>: operations
1
returns
1..pathLimit
1..pathLimit
returns
StrictAlignmentAllOptimalAlgorithm
+int: pathLimit
has a
«Enumeration»
OperationType
REPLACEMENT
DELETION
INSERTION
SKIPDELETION
SKIPINSERTION
StrictAlignmentKPathAlgorithm
+int: pathLimit
Note:
AlignmentMatrixEntries are
data storage objects that are
not intended for external use.
Thus they are not part of this
UML diagram in detail.
SoftAlignmentScoreAlgorithm
+double: beta
SoftPathModel
+Sequence: left
+Sequence: right
+double: score
+double: beta
+AlignmentSpecification:
specification
+SoftMatrixEntry[][]: pathMatrix
returns
1
SoftAlignmentFullAlgorithm
+double: beta
+double: approxThreshold