JavaScript is disabled on your browser.

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/AggregationRule.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-files/index-1.html)
* [Help](http://docs.google.com/help-doc.html)
* [Prev Class](http://docs.google.com/simplenlg/aggregation/AggregationHelper.html)
* [Next Class](http://docs.google.com/simplenlg/aggregation/Aggregator.html)
* [Frames](http://docs.google.com/index.html?simplenlg/aggregation/AggregationRule.html)
* [No Frames](http://docs.google.com/AggregationRule.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#id.3znysh7)|
* [Method](#id.2et92p0)
* Detail:
* Field |
* [Constr](#id.3dy6vkm)|
* [Method](#id.4d34og8)

simplenlg.aggregation

## Class AggregationRule

* java.lang.Object
  + simplenlg.aggregation.AggregationRule

Direct Known Subclasses: [BackwardConjunctionReductionRule](http://docs.google.com/simplenlg/aggregation/BackwardConjunctionReductionRule.html), [ClauseCoordinationRule](http://docs.google.com/simplenlg/aggregation/ClauseCoordinationRule.html), [ForwardConjunctionReductionRule](http://docs.google.com/simplenlg/aggregation/ForwardConjunctionReductionRule.html)

* public abstract class AggregationRule  
  extends java.lang.Object  
  This class represents an aggregation rule. All such rules need to implement an [apply(NLGElement, NLGElement)](http://docs.google.com/simplenlg/aggregation/AggregationRule.html#apply(simplenlg.framework.NLGElement,%20simplenlg.framework.NLGElement)) which takes an arbitrary number of [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html)s and perform some form of aggregation on them, returning an SPhraseSpec as a result, or null if the operation fails.Author: Albert Gatt, University of Malta and University of Aberdeen

### Constructor SummaryConstructors

|  |
| --- |
| * + Constructor and Description |
| * + [**AggregationRule**](http://docs.google.com/simplenlg/aggregation/AggregationRule.html#AggregationRule())() Creates a new instance of AggregationRule |

### Method SummaryMethods

|  |  |
| --- | --- |
| * + Modifier and Type | * + Method and Description |
| * + java.util.List<[NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html)> | * + [**apply**](http://docs.google.com/simplenlg/aggregation/AggregationRule.html#apply(java.util.List))(java.util.List<[NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html)> phrases) Performs aggregation on an arbitrary number of elements in a list. |
| * + [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) | * + [**apply**](http://docs.google.com/simplenlg/aggregation/AggregationRule.html#apply(simplenlg.framework.NLGElement))([NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) phrase) Perform aggregation on a single phrase. |
| * + abstract [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) | * + [**apply**](http://docs.google.com/simplenlg/aggregation/AggregationRule.html#apply(simplenlg.framework.NLGElement,%20simplenlg.framework.NLGElement))([NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) sentence1, [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) sentence2) Performs aggregation on a pair of sentences. |
| * + [NLGFactory](http://docs.google.com/simplenlg/framework/NLGFactory.html) | * + [**getFactory**](http://docs.google.com/simplenlg/aggregation/AggregationRule.html#getFactory())() |
| * + void | * + [**setFactory**](http://docs.google.com/simplenlg/aggregation/AggregationRule.html#setFactory(simplenlg.framework.NLGFactory))([NLGFactory](http://docs.google.com/simplenlg/framework/NLGFactory.html) factory) Set the factory that the rule should use to create phrases. |

### Methods inherited from class java.lang.Objectequals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Detail

#### AggregationRule public AggregationRule() Creates a new instance of AggregationRule

### Method Detail

#### setFactory public void setFactory([NLGFactory](http://docs.google.com/simplenlg/framework/NLGFactory.html) factory) Set the factory that the rule should use to create phrases.Parameters:factory - the factory

#### getFactory public [NLGFactory](http://docs.google.com/simplenlg/framework/NLGFactory.html) getFactory()Returns:the factory being used by this rule to create phrases

#### apply public java.util.List<[NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html)> apply(java.util.List<[NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html)> phrases) Performs aggregation on an arbitrary number of elements in a list. This method calls {[apply(NLGElement, NLGElement)](http://docs.google.com/simplenlg/aggregation/AggregationRule.html#apply(simplenlg.framework.NLGElement,%20simplenlg.framework.NLGElement)) on all pairs of elements in the list, recursively aggregating whenever it can.Parameters:phrases - the sentences Returns:a list containing the phrases, such that, for any two phrases s1 and s2, if [apply(NLGElement, NLGElement)](http://docs.google.com/simplenlg/aggregation/AggregationRule.html#apply(simplenlg.framework.NLGElement,%20simplenlg.framework.NLGElement)) succeeds on s1 and s2, the list contains the result; otherwise, the list contains s1 and s2.

#### apply public [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) apply([NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) phrase) Perform aggregation on a single phrase. This method only works on a [CoordinatedPhraseElement](http://docs.google.com/simplenlg/framework/CoordinatedPhraseElement.html), in which case it calls [apply(List)](http://docs.google.com/simplenlg/aggregation/AggregationRule.html#apply(java.util.List)) on the children of the coordinated phrase, returning a coordinated phrase whose children are the result.Parameters:phrase - Returns:aggregated result

#### apply public abstract [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) apply([NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) sentence1, [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) sentence2) Performs aggregation on a pair of sentences. This is the only method that extensions of AggregationRule need to implement.Parameters:sentence1 - the first sentencesentence2 - the second sentence Returns:an aggregated sentence, if the method succeeds, null otherwise

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/AggregationRule.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-files/index-1.html)
* [Help](http://docs.google.com/help-doc.html)
* [Prev Class](http://docs.google.com/simplenlg/aggregation/AggregationHelper.html)
* [Next Class](http://docs.google.com/simplenlg/aggregation/Aggregator.html)
* [Frames](http://docs.google.com/index.html?simplenlg/aggregation/AggregationRule.html)
* [No Frames](http://docs.google.com/AggregationRule.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#id.3znysh7)|
* [Method](#id.2et92p0)
* Detail:
* Field |
* [Constr](#id.3dy6vkm)|
* [Method](#id.4d34og8)