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/NLGFactory.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/framework/NLGElement.html)
* [Next Class](http://docs.google.com/simplenlg/framework/NLGModule.html)
* [Frames](http://docs.google.com/index.html?simplenlg/framework/NLGFactory.html)
* [No Frames](http://docs.google.com/NLGFactory.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.2s8eyo1)

simplenlg.framework

## Class NLGFactory

* java.lang.Object
  + simplenlg.framework.NLGFactory
* public class NLGFactory  
  extends java.lang.Object  
  This class contains methods for creating syntactic phrases. These methods should be used instead of directly invoking new on SPhraseSpec, etc. The phrase factory should be linked to s lexicon if possible (although it will work without one)Version: 4.0 Author: D. Westwater, University of Aberdeen.

### Constructor SummaryConstructors

|  |
| --- |
| * + Constructor and Description |
| * + [**NLGFactory**](http://docs.google.com/simplenlg/framework/NLGFactory.html#NLGFactory())() Creates a new phrase factory with no associated lexicon. |
| * + [**NLGFactory**](http://docs.google.com/simplenlg/framework/NLGFactory.html#NLGFactory(simplenlg.lexicon.Lexicon))([Lexicon](http://docs.google.com/simplenlg/lexicon/Lexicon.html) newLexicon) Creates a new phrase factory with the associated lexicon. |

### Method SummaryMethods

|  |  |
| --- | --- |
| * + Modifier and Type | * + Method and Description |
| * + [AdjPhraseSpec](http://docs.google.com/simplenlg/phrasespec/AdjPhraseSpec.html) | * + [**createAdjectivePhrase**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createAdjectivePhrase())() Creates a blank adjective phrase with no base adjective set. |
| * + [AdjPhraseSpec](http://docs.google.com/simplenlg/phrasespec/AdjPhraseSpec.html) | * + [**createAdjectivePhrase**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createAdjectivePhrase(java.lang.Object))(java.lang.Object adjective) Creates an adjective phrase wrapping the given adjective. |
| * + [AdvPhraseSpec](http://docs.google.com/simplenlg/phrasespec/AdvPhraseSpec.html) | * + [**createAdverbPhrase**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createAdverbPhrase())() Creates a blank adverb phrase that has no adverb. |
| * + [AdvPhraseSpec](http://docs.google.com/simplenlg/phrasespec/AdvPhraseSpec.html) | * + [**createAdverbPhrase**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createAdverbPhrase(java.lang.String))(java.lang.String adverb) Creates an adverb phrase wrapping the given adverb. |
| * + [SPhraseSpec](http://docs.google.com/simplenlg/phrasespec/SPhraseSpec.html) | * + [**createClause**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createClause())() Creates a blank clause with no subject, verb or objects. |
| * + [SPhraseSpec](http://docs.google.com/simplenlg/phrasespec/SPhraseSpec.html) | * + [**createClause**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createClause(java.lang.Object,%20java.lang.Object))(java.lang.Object subject, java.lang.Object verb) Creates a clause with the given subject and verb but no objects. |
| * + [SPhraseSpec](http://docs.google.com/simplenlg/phrasespec/SPhraseSpec.html) | * + [**createClause**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createClause(java.lang.Object,%20java.lang.Object,%20java.lang.Object))(java.lang.Object subject, java.lang.Object verb, java.lang.Object directObject) Creates a clause with the given subject, verb or verb phrase and direct object but no indirect object. |
| * + [CoordinatedPhraseElement](http://docs.google.com/simplenlg/framework/CoordinatedPhraseElement.html) | * + [**createCoordinatedPhrase**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createCoordinatedPhrase())() Creates a new (empty) coordinated phrase |
| * + [CoordinatedPhraseElement](http://docs.google.com/simplenlg/framework/CoordinatedPhraseElement.html) | * + [**createCoordinatedPhrase**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createCoordinatedPhrase(java.lang.Object,%20java.lang.Object))(java.lang.Object coord1, java.lang.Object coord2) Creates a new coordinated phrase with two elements (initially) |
| * + [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) | * + [**createDocument**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createDocument())() Creates a new document element with no title. |
| * + [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) | * + [**createDocument**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createDocument(java.lang.String))(java.lang.String title) Creates a new document element with the given title. |
| * + [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) | * + [**createDocument**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createDocument(java.lang.String,%20java.util.List))(java.lang.String title, java.util.List<[DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html)> components) Creates a new document element with the given title and adds all of the given components in the list |
| * + [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) | * + [**createDocument**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createDocument(java.lang.String,%20simplenlg.framework.NLGElement))(java.lang.String title, [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) component) Creates a new document element with the given title and adds the given component. |
| * + [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) | * + [**createInflectedWord**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createInflectedWord(java.lang.Object,%20simplenlg.framework.LexicalCategory))(java.lang.Object word, [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) category) Create an inflected word element. |
| * + [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) | * + [**createList**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createList())() Creates a new list element with no components. |
| * + [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) | * + [**createList**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createList(java.util.List))(java.util.List<[DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html)> textComponents) Creates a new list element and adds all of the given components in the list |
| * + [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) | * + [**createList**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createList(simplenlg.framework.NLGElement))([NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) component) Creates a new section element with the given title and adds the given component. |
| * + [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) | * + [**createListItem**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createListItem())() Creates a list item for adding to a list element. |
| * + [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) | * + [**createListItem**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createListItem(simplenlg.framework.NLGElement))([NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) component) Creates a list item for adding to a list element. |
| * + [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) | * + [**createNLGElement**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createNLGElement(java.lang.Object))(java.lang.Object element) create an NLGElement from the element, no default lexical category |
| * + [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) | * + [**createNLGElement**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createNLGElement(java.lang.Object,%20simplenlg.framework.LexicalCategory))(java.lang.Object element, [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) category) this method creates an NLGElement from an object If object is null, return null If the object is already an NLGElement, it is returned unchanged Exception: if it is an InflectedWordElement, return underlying WordElement If it is a String which matches a lexicon entry or pronoun, the relevant WordElement is returned If it is a different String, a wordElement is created if the string is a single word Otherwise a StringElement is returned Otherwise throw an exception |
| * + [NPPhraseSpec](http://docs.google.com/simplenlg/phrasespec/NPPhraseSpec.html) | * + [**createNounPhrase**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createNounPhrase())() Creates a blank noun phrase with no subject or specifier. |
| * + [NPPhraseSpec](http://docs.google.com/simplenlg/phrasespec/NPPhraseSpec.html) | * + [**createNounPhrase**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createNounPhrase(java.lang.Object))(java.lang.Object noun) Creates a noun phrase with the given subject but no specifier. |
| * + [NPPhraseSpec](http://docs.google.com/simplenlg/phrasespec/NPPhraseSpec.html) | * + [**createNounPhrase**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createNounPhrase(java.lang.Object,%20java.lang.Object))(java.lang.Object specifier, java.lang.Object noun) Creates a noun phrase with the given specifier and subject. |
| * + [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) | * + [**createParagraph**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createParagraph())() Creates a new paragraph element with no components. |
| * + [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) | * + [**createParagraph**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createParagraph(java.util.List))(java.util.List<[DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html)> components) Creates a new paragraph element and adds all of the given components in the list |
| * + [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) | * + [**createParagraph**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createParagraph(simplenlg.framework.NLGElement))([NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) component) Creates a new paragraph element and adds the given component |
| * + [PPPhraseSpec](http://docs.google.com/simplenlg/phrasespec/PPPhraseSpec.html) | * + [**createPrepositionPhrase**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createPrepositionPhrase())() Creates a blank preposition phrase with no preposition or complements. |
| * + [PPPhraseSpec](http://docs.google.com/simplenlg/phrasespec/PPPhraseSpec.html) | * + [**createPrepositionPhrase**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createPrepositionPhrase(java.lang.Object))(java.lang.Object preposition) Creates a preposition phrase with the given preposition. |
| * + [PPPhraseSpec](http://docs.google.com/simplenlg/phrasespec/PPPhraseSpec.html) | * + [**createPrepositionPhrase**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createPrepositionPhrase(java.lang.Object,%20java.lang.Object))(java.lang.Object preposition, java.lang.Object complement) Creates a preposition phrase with the given preposition and complement. |
| * + [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) | * + [**createSection**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createSection())() Creates a new section element. |
| * + [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) | * + [**createSection**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createSection(java.lang.String))(java.lang.String title) Creates a new section element with the given title. |
| * + [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) | * + [**createSection**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createSection(java.lang.String,%20java.util.List))(java.lang.String title, java.util.List<[DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html)> components) Creates a new section element with the given title and adds all of the given components in the list |
| * + [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) | * + [**createSection**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createSection(java.lang.String,%20simplenlg.framework.NLGElement))(java.lang.String title, [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) component) Creates a new section element with the given title and adds the given component. |
| * + [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) | * + [**createSentence**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createSentence())() Creates a new sentence element with no components. |
| * + [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) | * + [**createSentence**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createSentence(java.util.List))(java.util.List<[NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html)> components) Creates a new sentence element and adds all of the given components. |
| * + [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) | * + [**createSentence**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createSentence(simplenlg.framework.NLGElement))([NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) components) Creates a new sentence element |
| * + [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) | * + [**createSentence**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createSentence(java.lang.Object,%20java.lang.Object))(java.lang.Object subject, java.lang.Object verb) Creates a sentence with the given subject and verb. |
| * + [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) | * + [**createSentence**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createSentence(java.lang.Object,%20java.lang.Object,%20java.lang.Object))(java.lang.Object subject, java.lang.Object verb, java.lang.Object complement) Creates a sentence with the given subject, verb and direct object. |
| * + [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) | * + [**createSentence**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createSentence(java.lang.String))(java.lang.String cannedSentence) Creates a new sentence with the given canned text. |
| * + [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) | * + [**createStringElement**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createStringElement())() Creates a blank canned text phrase with no text. |
| * + [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) | * + [**createStringElement**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createStringElement(java.lang.String))(java.lang.String text) Creates a canned text phrase with the given text. |
| * + [VPPhraseSpec](http://docs.google.com/simplenlg/phrasespec/VPPhraseSpec.html) | * + [**createVerbPhrase**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createVerbPhrase())() Creates a blank verb phrase with no main verb. |
| * + [VPPhraseSpec](http://docs.google.com/simplenlg/phrasespec/VPPhraseSpec.html) | * + [**createVerbPhrase**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createVerbPhrase(java.lang.Object))(java.lang.Object verb) Creates a verb phrase wrapping the main verb given. |
| * + [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) | * + [**createWord**](http://docs.google.com/simplenlg/framework/NLGFactory.html#createWord(java.lang.Object,%20simplenlg.framework.LexicalCategory))(java.lang.Object word, [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) category) Creates a new element representing a word. |
| * + void | * + [**setLexicon**](http://docs.google.com/simplenlg/framework/NLGFactory.html#setLexicon(simplenlg.lexicon.Lexicon))([Lexicon](http://docs.google.com/simplenlg/lexicon/Lexicon.html) newLexicon) Sets the lexicon to be used by this factory. |

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

### Constructor Detail

#### NLGFactory public NLGFactory() Creates a new phrase factory with no associated lexicon.

#### NLGFactory public NLGFactory([Lexicon](http://docs.google.com/simplenlg/lexicon/Lexicon.html) newLexicon) Creates a new phrase factory with the associated lexicon.Parameters:newLexicon - the Lexicon to be used with this factory.

### Method Detail

#### setLexicon public void setLexicon([Lexicon](http://docs.google.com/simplenlg/lexicon/Lexicon.html) newLexicon) Sets the lexicon to be used by this factory. Passing a parameter of null will remove any existing lexicon from the factory.Parameters:newLexicon - the new Lexicon to be used.

#### createWord public [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) createWord(java.lang.Object word, [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) category) Creates a new element representing a word. If the word passed is already an NLGElement then that is returned unchanged. If a String is passed as the word then the factory will look up the Lexicon if one exists and use the details found to create a new WordElement.Parameters:word - the base word for the new element. This can be a NLGElement, which is returned unchanged, or a String, which is used to construct a new WordElement.category - the LexicalCategory for the word. Returns:an NLGElement representing the word.

#### createInflectedWord public [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) createInflectedWord(java.lang.Object word, [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) category) Create an inflected word element. InflectedWordElement represents a word that already specifies the morphological and other features that it should exhibit in a realisation. While normally, phrases are constructed using WordElements, and features are set on phrases, it is sometimes desirable to set features directly on words (for example, when one wants to elide a specific word, but not its parent phrase). If the object passed is already a WordElement, then a new InflectedWordElement is returned which wraps this WordElement . If the object is a String, then the WordElement representing this String is looked up, and a new InflectedWordElement wrapping this is returned. If no such WordElement is found, the element returned is an InflectedWordElement with the supplied string as baseform and no base WordElement . If an NLGElement is passed, this is returned unchanged.Parameters:word - the wordcategory - the category Returns:an InflectedWordElement, or the original supplied object if it is an NLGElement.

#### createPrepositionPhrase public [PPPhraseSpec](http://docs.google.com/simplenlg/phrasespec/PPPhraseSpec.html) createPrepositionPhrase() Creates a blank preposition phrase with no preposition or complements.Returns:a PPPhraseSpec representing this phrase.

#### createPrepositionPhrase public [PPPhraseSpec](http://docs.google.com/simplenlg/phrasespec/PPPhraseSpec.html) createPrepositionPhrase(java.lang.Object preposition) Creates a preposition phrase with the given preposition.Parameters:preposition - the preposition to be used. Returns:a PPPhraseSpec representing this phrase.

#### createPrepositionPhrase public [PPPhraseSpec](http://docs.google.com/simplenlg/phrasespec/PPPhraseSpec.html) createPrepositionPhrase(java.lang.Object preposition, java.lang.Object complement) Creates a preposition phrase with the given preposition and complement. An NLGElement representing the preposition is added as the head feature of this phrase while the complement is added as a normal phrase complement.Parameters:preposition - the preposition to be used.complement - the complement of the phrase. Returns:a PPPhraseSpec representing this phrase.

#### createNLGElement public [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) createNLGElement(java.lang.Object element, [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) category) this method creates an NLGElement from an object If object is null, return null If the object is already an NLGElement, it is returned unchanged Exception: if it is an InflectedWordElement, return underlying WordElement If it is a String which matches a lexicon entry or pronoun, the relevant WordElement is returned If it is a different String, a wordElement is created if the string is a single word Otherwise a StringElement is returned Otherwise throw an exceptionParameters:element - - object to look upcategory - - default lexical category of object Returns:NLGelement

#### createNLGElement public [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) createNLGElement(java.lang.Object element) create an NLGElement from the element, no default lexical categoryParameters:element - Returns:NLGelement

#### createNounPhrase public [NPPhraseSpec](http://docs.google.com/simplenlg/phrasespec/NPPhraseSpec.html) createNounPhrase() Creates a blank noun phrase with no subject or specifier.Returns:a NPPhraseSpec representing this phrase.

#### createNounPhrase public [NPPhraseSpec](http://docs.google.com/simplenlg/phrasespec/NPPhraseSpec.html) createNounPhrase(java.lang.Object noun) Creates a noun phrase with the given subject but no specifier.Parameters:noun - the subject of the phrase. Returns:a NPPhraseSpec representing this phrase.

#### createNounPhrase public [NPPhraseSpec](http://docs.google.com/simplenlg/phrasespec/NPPhraseSpec.html) createNounPhrase(java.lang.Object specifier, java.lang.Object noun) Creates a noun phrase with the given specifier and subject.Parameters:specifier - the specifier or determiner for the noun phrase.noun - the subject of the phrase. Returns:a NPPhraseSpec representing this phrase.

#### createAdjectivePhrase public [AdjPhraseSpec](http://docs.google.com/simplenlg/phrasespec/AdjPhraseSpec.html) createAdjectivePhrase() Creates a blank adjective phrase with no base adjective set.Returns:a AdjPhraseSpec representing this phrase.

#### createAdjectivePhrase public [AdjPhraseSpec](http://docs.google.com/simplenlg/phrasespec/AdjPhraseSpec.html) createAdjectivePhrase(java.lang.Object adjective) Creates an adjective phrase wrapping the given adjective.Parameters:adjective - the main adjective for this phrase. Returns:a AdjPhraseSpec representing this phrase.

#### createVerbPhrase public [VPPhraseSpec](http://docs.google.com/simplenlg/phrasespec/VPPhraseSpec.html) createVerbPhrase() Creates a blank verb phrase with no main verb.Returns:a VPPhraseSpec representing this phrase.

#### createVerbPhrase public [VPPhraseSpec](http://docs.google.com/simplenlg/phrasespec/VPPhraseSpec.html) createVerbPhrase(java.lang.Object verb) Creates a verb phrase wrapping the main verb given. If a String is passed in then some parsing is done to see if the verb contains a particle, for example *fall down*. The first word is taken to be the verb while all other words are assumed to form the particle.Parameters:verb - the verb to be wrapped. Returns:a VPPhraseSpec representing this phrase.

#### createAdverbPhrase public [AdvPhraseSpec](http://docs.google.com/simplenlg/phrasespec/AdvPhraseSpec.html) createAdverbPhrase() Creates a blank adverb phrase that has no adverb.Returns:a AdvPhraseSpec representing this phrase.

#### createAdverbPhrase public [AdvPhraseSpec](http://docs.google.com/simplenlg/phrasespec/AdvPhraseSpec.html) createAdverbPhrase(java.lang.String adverb) Creates an adverb phrase wrapping the given adverb.Parameters:adverb - the adverb for this phrase. Returns:a AdvPhraseSpec representing this phrase.

#### createClause public [SPhraseSpec](http://docs.google.com/simplenlg/phrasespec/SPhraseSpec.html) createClause() Creates a blank clause with no subject, verb or objects.Returns:a SPhraseSpec representing this phrase.

#### createClause public [SPhraseSpec](http://docs.google.com/simplenlg/phrasespec/SPhraseSpec.html) createClause(java.lang.Object subject, java.lang.Object verb) Creates a clause with the given subject and verb but no objects.Parameters:subject - the subject for the clause as a NLGElement or String. This forms a noun phrase.verb - the verb for the clause as a NLGElement or String. This forms a verb phrase. Returns:a SPhraseSpec representing this phrase.

#### createClause public [SPhraseSpec](http://docs.google.com/simplenlg/phrasespec/SPhraseSpec.html) createClause(java.lang.Object subject, java.lang.Object verb, java.lang.Object directObject) Creates a clause with the given subject, verb or verb phrase and direct object but no indirect object.Parameters:subject - the subject for the clause as a NLGElement or String. This forms a noun phrase.verb - the verb for the clause as a NLGElement or String. This forms a verb phrase.directObject - the direct object for the clause as a NLGElement or String. This forms a complement for the clause. Returns:a SPhraseSpec representing this phrase.

#### createStringElement public [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) createStringElement() Creates a blank canned text phrase with no text.Returns:a PhraseElement representing this phrase.

#### createStringElement public [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) createStringElement(java.lang.String text) Creates a canned text phrase with the given text.Parameters:text - the canned text. Returns:a PhraseElement representing this phrase.

#### createCoordinatedPhrase public [CoordinatedPhraseElement](http://docs.google.com/simplenlg/framework/CoordinatedPhraseElement.html) createCoordinatedPhrase() Creates a new (empty) coordinated phraseReturns:empty CoordinatedPhraseElement

#### createCoordinatedPhrase public [CoordinatedPhraseElement](http://docs.google.com/simplenlg/framework/CoordinatedPhraseElement.html) createCoordinatedPhrase(java.lang.Object coord1, java.lang.Object coord2) Creates a new coordinated phrase with two elements (initially)Parameters:coord1 - - first phrase to be coordinatedcoord2 - = second phrase to be coordinated Returns:CoordinatedPhraseElement for the two given elements

#### createDocument public [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) createDocument() Creates a new document element with no title.Returns:a DocumentElement

#### createDocument public [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) createDocument(java.lang.String title) Creates a new document element with the given title.Parameters:title - the title for this element. Returns:a DocumentElement.

#### createDocument public [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) createDocument(java.lang.String title, java.util.List<[DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html)> components) Creates a new document element with the given title and adds all of the given components in the listParameters:title - the title of this element.components - a List of NLGElements that form the components of this element. Returns:a DocumentElement

#### createDocument public [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) createDocument(java.lang.String title, [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) component) Creates a new document element with the given title and adds the given component.Parameters:title - the title for this element.component - an NLGElement that becomes the first component of this document element. Returns:a DocumentElement

#### createList public [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) createList() Creates a new list element with no components.Returns:a DocumentElement representing the list.

#### createList public [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) createList(java.util.List<[DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html)> textComponents) Creates a new list element and adds all of the given components in the listParameters:textComponents - a List of NLGElements that form the components of this element. Returns:a DocumentElement representing the list.

#### createList public [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) createList([NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) component) Creates a new section element with the given title and adds the given component.Parameters:component - an NLGElement that becomes the first component of this document element. Returns:a DocumentElement representing the section.

#### createListItem public [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) createListItem() Creates a list item for adding to a list element.Returns:a DocumentElement representing the list item.

#### createListItem public [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) createListItem([NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) component) Creates a list item for adding to a list element. The list item has the given component.Returns:a DocumentElement representing the list item.

#### createParagraph public [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) createParagraph() Creates a new paragraph element with no components.Returns:a DocumentElement representing this paragraph

#### createParagraph public [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) createParagraph(java.util.List<[DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html)> components) Creates a new paragraph element and adds all of the given components in the listParameters:components - a List of NLGElements that form the components of this element. Returns:a DocumentElement representing this paragraph

#### createParagraph public [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) createParagraph([NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) component) Creates a new paragraph element and adds the given componentParameters:component - an NLGElement that becomes the first component of this document element. Returns:a DocumentElement representing this paragraph

#### createSection public [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) createSection() Creates a new section element.Returns:a DocumentElement representing the section.

#### createSection public [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) createSection(java.lang.String title) Creates a new section element with the given title.Parameters:title - the title of the section. Returns:a DocumentElement representing the section.

#### createSection public [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) createSection(java.lang.String title, java.util.List<[DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html)> components) Creates a new section element with the given title and adds all of the given components in the listParameters:title - the title of this element.components - a List of NLGElements that form the components of this element. Returns:a DocumentElement representing the section.

#### createSection public [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) createSection(java.lang.String title, [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) component) Creates a new section element with the given title and adds the given component.Parameters:title - the title for this element.component - an NLGElement that becomes the first component of this document element. Returns:a DocumentElement representing the section.

#### createSentence public [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) createSentence() Creates a new sentence element with no components.Returns:a DocumentElement representing this sentence

#### createSentence public [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) createSentence(java.util.List<[NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html)> components) Creates a new sentence element and adds all of the given components.Parameters:components - a List of NLGElements that form the components of this element. Returns:a DocumentElement representing this sentence

#### createSentence public [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) createSentence([NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) components) Creates a new sentence elementParameters:components - an NLGElement that becomes the first component of this document element. Returns:a DocumentElement representing this sentence

#### createSentence public [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) createSentence(java.lang.Object subject, java.lang.Object verb) Creates a sentence with the given subject and verb. The phrase factory is used to construct a clause that then forms the components of the sentence.Parameters:subject - the subject of the sentence.verb - the verb of the sentence. Returns:a DocumentElement representing this sentence

#### createSentence public [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) createSentence(java.lang.Object subject, java.lang.Object verb, java.lang.Object complement) Creates a sentence with the given subject, verb and direct object. The phrase factory is used to construct a clause that then forms the components of the sentence.Parameters:subject - the subject of the sentence.verb - the verb of the sentence.complement - the object of the sentence. Returns:a DocumentElement representing this sentence

#### createSentence public [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) createSentence(java.lang.String cannedSentence) Creates a new sentence with the given canned text. The canned text is used to form a canned phrase (from the phrase factory) which is then added as the component to sentence element.Parameters:cannedSentence - the canned text as a String. Returns:a DocumentElement representing this sentence

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/NLGFactory.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/framework/NLGElement.html)
* [Next Class](http://docs.google.com/simplenlg/framework/NLGModule.html)
* [Frames](http://docs.google.com/index.html?simplenlg/framework/NLGFactory.html)
* [No Frames](http://docs.google.com/NLGFactory.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.2s8eyo1)