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/WordElement.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/StringElement.html)
* Next Class
* [Frames](http://docs.google.com/index.html?simplenlg/framework/WordElement.html)
* [No Frames](http://docs.google.com/WordElement.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#id.3znysh7)|
* [Method](#id.2et92p0)
* Detail:
* Field |
* [Constr](#id.1t3h5sf)|
* [Method](#id.26in1rg)

simplenlg.framework

## Class WordElement

* java.lang.Object
  + [simplenlg.framework.NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html)
    - simplenlg.framework.WordElement
* public class WordElement  
  extends [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html)  
  This is the class for a lexical entry (ie, a word). Words are stored in a [Lexicon](http://docs.google.com/simplenlg/lexicon/Lexicon.html), and usually the developer retrieves a WordElement via a lookup method in the lexicon Words always have a base form, and usually have a [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html). They may also have a Lexicon ID. Words also have features (which are retrieved from the Lexicon), these are held in the standard NLGElement feature mapVersion: 4.0 Author: E. Reiter, University of Aberdeen.

### Constructor SummaryConstructors

|  |
| --- |
| * + Constructor and Description |
| * + [**WordElement**](http://docs.google.com/simplenlg/framework/WordElement.html#WordElement())() empty constructor |
| * + [**WordElement**](http://docs.google.com/simplenlg/framework/WordElement.html#WordElement(java.lang.String))(java.lang.String baseForm) create a WordElement with the specified baseForm (no category or ID specified) |
| * + [**WordElement**](http://docs.google.com/simplenlg/framework/WordElement.html#WordElement(java.lang.String,%20simplenlg.framework.LexicalCategory))(java.lang.String baseForm, [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) category) create a WordElement with the specified baseForm and category |
| * + [**WordElement**](http://docs.google.com/simplenlg/framework/WordElement.html#WordElement(java.lang.String,%20simplenlg.framework.LexicalCategory,%20java.lang.String))(java.lang.String baseForm, [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) category, java.lang.String id) create a WordElement with the specified baseForm, category, ID |

### Method SummaryMethods

|  |  |
| --- | --- |
| * + Modifier and Type | * + Method and Description |
| * + void | * + [**addInflectionalVariant**](http://docs.google.com/simplenlg/framework/WordElement.html#addInflectionalVariant(simplenlg.features.Inflection))([Inflection](http://docs.google.com/simplenlg/features/Inflection.html) infl) Specify that this word has an inflectional variant (e.g. |
| * + void | * + [**addInflectionalVariant**](http://docs.google.com/simplenlg/framework/WordElement.html#addInflectionalVariant(simplenlg.features.Inflection,%20java.lang.String,%20java.lang.String))([Inflection](http://docs.google.com/simplenlg/features/Inflection.html) infl, java.lang.String lexicalFeature, java.lang.String form) Add an inflectional variant to this word element. |
| * + boolean | * + [**equals**](http://docs.google.com/simplenlg/framework/WordElement.html#equals(java.lang.Object))(java.lang.Object o) Check if this WordElement is equal to an object. |
| * + java.lang.String | * + [**getBaseForm**](http://docs.google.com/simplenlg/framework/WordElement.html#getBaseForm())() |
| * + java.util.List<[NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html)> | * + [**getChildren**](http://docs.google.com/simplenlg/framework/WordElement.html#getChildren())() This method returns an empty List as words do not have child elements. |
| * + java.lang.Object | * + [**getDefaultInflectionalVariant**](http://docs.google.com/simplenlg/framework/WordElement.html#getDefaultInflectionalVariant())() |
| * + java.lang.String | * + [**getDefaultSpellingVariant**](http://docs.google.com/simplenlg/framework/WordElement.html#getDefaultSpellingVariant())() Convenience method, equivalent to getFeatureAsString(LexicalFeature.DEFAULT\_SPELL). |
| * + java.lang.String | * + [**getId**](http://docs.google.com/simplenlg/framework/WordElement.html#getId())() |
| * + boolean | * + [**hasInflectionalVariant**](http://docs.google.com/simplenlg/framework/WordElement.html#hasInflectionalVariant(simplenlg.features.Inflection))([Inflection](http://docs.google.com/simplenlg/features/Inflection.html) infl) Check whether this word has a particular inflectional variant |
| * + java.lang.String | * + [**printTree**](http://docs.google.com/simplenlg/framework/WordElement.html#printTree(java.lang.String))(java.lang.String indent) |
| * + void | * + [**setBaseForm**](http://docs.google.com/simplenlg/framework/WordElement.html#setBaseForm(java.lang.String))(java.lang.String baseForm) |
| * + void | * + [**setDefaultInflectionalVariant**](http://docs.google.com/simplenlg/framework/WordElement.html#setDefaultInflectionalVariant(simplenlg.features.Inflection))([Inflection](http://docs.google.com/simplenlg/features/Inflection.html) variant) Set the default inflectional variant of a word. |
| * + void | * + [**setDefaultSpellingVariant**](http://docs.google.com/simplenlg/framework/WordElement.html#setDefaultSpellingVariant(java.lang.String))(java.lang.String variant) Convenience method to set the default spelling variant of a word. |
| * + void | * + [**setId**](http://docs.google.com/simplenlg/framework/WordElement.html#setId(java.lang.String))(java.lang.String id) |
| * + java.lang.String | * + [**toString**](http://docs.google.com/simplenlg/framework/WordElement.html#toString())() |
| * + java.lang.String | * + [**toXML**](http://docs.google.com/simplenlg/framework/WordElement.html#toXML())() |

### Methods inherited from class simplenlg.framework.[**NLGElement**](http://docs.google.com/simplenlg/framework/NLGElement.html)[clearAllFeatures](http://docs.google.com/simplenlg/framework/NLGElement.html#clearAllFeatures()), [equals](http://docs.google.com/simplenlg/framework/NLGElement.html#equals(java.lang.String)), [getAllFeatureNames](http://docs.google.com/simplenlg/framework/NLGElement.html#getAllFeatureNames()), [getAllFeatures](http://docs.google.com/simplenlg/framework/NLGElement.html#getAllFeatures()), [getCategory](http://docs.google.com/simplenlg/framework/NLGElement.html#getCategory()), [getFactory](http://docs.google.com/simplenlg/framework/NLGElement.html#getFactory()), [getFeature](http://docs.google.com/simplenlg/framework/NLGElement.html#getFeature(java.lang.String)), [getFeatureAsBoolean](http://docs.google.com/simplenlg/framework/NLGElement.html#getFeatureAsBoolean(java.lang.String)), [getFeatureAsDouble](http://docs.google.com/simplenlg/framework/NLGElement.html#getFeatureAsDouble(java.lang.String)), [getFeatureAsElement](http://docs.google.com/simplenlg/framework/NLGElement.html#getFeatureAsElement(java.lang.String)), [getFeatureAsElementList](http://docs.google.com/simplenlg/framework/NLGElement.html#getFeatureAsElementList(java.lang.String)), [getFeatureAsFloat](http://docs.google.com/simplenlg/framework/NLGElement.html#getFeatureAsFloat(java.lang.String)), [getFeatureAsInteger](http://docs.google.com/simplenlg/framework/NLGElement.html#getFeatureAsInteger(java.lang.String)), [getFeatureAsList](http://docs.google.com/simplenlg/framework/NLGElement.html#getFeatureAsList(java.lang.String)), [getFeatureAsLong](http://docs.google.com/simplenlg/framework/NLGElement.html#getFeatureAsLong(java.lang.String)), [getFeatureAsString](http://docs.google.com/simplenlg/framework/NLGElement.html#getFeatureAsString(java.lang.String)), [getFeatureAsStringList](http://docs.google.com/simplenlg/framework/NLGElement.html#getFeatureAsStringList(java.lang.String)), [getParent](http://docs.google.com/simplenlg/framework/NLGElement.html#getParent()), [getRealisation](http://docs.google.com/simplenlg/framework/NLGElement.html#getRealisation()), [getTense](http://docs.google.com/simplenlg/framework/NLGElement.html#getTense()), [hasFeature](http://docs.google.com/simplenlg/framework/NLGElement.html#hasFeature(java.lang.String)), [isA](http://docs.google.com/simplenlg/framework/NLGElement.html#isA(simplenlg.framework.ElementCategory)), [isNegated](http://docs.google.com/simplenlg/framework/NLGElement.html#isNegated()), [isPlural](http://docs.google.com/simplenlg/framework/NLGElement.html#isPlural()), [removeFeature](http://docs.google.com/simplenlg/framework/NLGElement.html#removeFeature(java.lang.String)), [setCategory](http://docs.google.com/simplenlg/framework/NLGElement.html#setCategory(simplenlg.framework.ElementCategory)), [setFactory](http://docs.google.com/simplenlg/framework/NLGElement.html#setFactory(simplenlg.framework.NLGFactory)), [setFeature](http://docs.google.com/simplenlg/framework/NLGElement.html#setFeature(java.lang.String,%20boolean)), [setFeature](http://docs.google.com/simplenlg/framework/NLGElement.html#setFeature(java.lang.String,%20double)), [setFeature](http://docs.google.com/simplenlg/framework/NLGElement.html#setFeature(java.lang.String,%20float)), [setFeature](http://docs.google.com/simplenlg/framework/NLGElement.html#setFeature(java.lang.String,%20int)), [setFeature](http://docs.google.com/simplenlg/framework/NLGElement.html#setFeature(java.lang.String,%20long)), [setFeature](http://docs.google.com/simplenlg/framework/NLGElement.html#setFeature(java.lang.String,%20java.lang.Object)), [setNegated](http://docs.google.com/simplenlg/framework/NLGElement.html#setNegated(boolean)), [setParent](http://docs.google.com/simplenlg/framework/NLGElement.html#setParent(simplenlg.framework.NLGElement)), [setPlural](http://docs.google.com/simplenlg/framework/NLGElement.html#setPlural(boolean)), [setRealisation](http://docs.google.com/simplenlg/framework/NLGElement.html#setRealisation(java.lang.String)), [setTense](http://docs.google.com/simplenlg/framework/NLGElement.html#setTense(simplenlg.features.Tense))

### Methods inherited from class java.lang.ObjectgetClass, hashCode, notify, notifyAll, wait, wait, wait

### Constructor Detail

#### WordElement public WordElement() empty constructor

#### WordElement public WordElement(java.lang.String baseForm) create a WordElement with the specified baseForm (no category or ID specified)Parameters:baseForm - - base form of WordElement

#### WordElement public WordElement(java.lang.String baseForm, [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) category) create a WordElement with the specified baseForm and categoryParameters:baseForm - - base form of WordElementcategory - - category of WordElement

#### WordElement public WordElement(java.lang.String baseForm, [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) category, java.lang.String id) create a WordElement with the specified baseForm, category, IDParameters:baseForm - - base form of WordElementcategory - - category of WordElementid - - ID of word in lexicon

### Method Detail

#### getBaseForm public java.lang.String getBaseForm()Returns:the baseForm

#### getId public java.lang.String getId()Returns:the id

#### setBaseForm public void setBaseForm(java.lang.String baseForm)Parameters:baseForm - the baseForm to set

#### setId public void setId(java.lang.String id)Parameters:id - the id to set

#### setDefaultInflectionalVariant public void setDefaultInflectionalVariant([Inflection](http://docs.google.com/simplenlg/features/Inflection.html) variant) Set the default inflectional variant of a word. This is mostly relevant if the word has more than one possible inflectional variant (for example, it can be inflected in both a regular and irregular way). If the default inflectional variant is set, the inflectional forms of the word may change as a result. This depends on whether inflectional forms have been specifically associated with this variant, via [addInflectionalVariant(Inflection, String, String)](http://docs.google.com/simplenlg/framework/WordElement.html#addInflectionalVariant(simplenlg.features.Inflection,%20java.lang.String,%20java.lang.String)). The NIHDBLexicon associates different inflectional variants with words, if they are so specified, and adds the correct forms.Parameters:variant - The variant

#### getDefaultInflectionalVariant public java.lang.Object getDefaultInflectionalVariant()Returns:the default inflectional variant

#### setDefaultSpellingVariant public void setDefaultSpellingVariant(java.lang.String variant) Convenience method to set the default spelling variant of a word. Equivalent to setFeature(LexicalFeature.DEFAULT\_SPELL, variant). By default, the spelling variant used is the base form. If otherwise set, this forces the realiser to always use the spelling variant specified.Parameters:variant - The spelling variant

#### getDefaultSpellingVariant public java.lang.String getDefaultSpellingVariant() Convenience method, equivalent to getFeatureAsString(LexicalFeature.DEFAULT\_SPELL). If this feature is not set, the baseform is returned.Returns:the default inflectional variant

#### addInflectionalVariant public void addInflectionalVariant([Inflection](http://docs.google.com/simplenlg/features/Inflection.html) infl, java.lang.String lexicalFeature, java.lang.String form) Add an inflectional variant to this word element. This method is intended for use by a Lexicon. The idea is that words which have more than one inflectional variant (for example, a regular and an irregular form of the past tense), can have a default variant (for example, the regular), but also store information about the other variants. This comes in useful in case the default inflectional variant is reset to a new one. In that case, the stored forms for the new variant are used to inflect the word. **An example:** The verb *lie* has both a regular form (*lies, lied, lying*) and an irregular form (*lay, lain,* etc). Assume that the Lexicon provides this information and treats this as variant information of the same word (as does the NIHDBLexicon, for example). Typically, the default inflectional variant is the Inflection.REGULAR. This means that morphology proceeds to inflect the verb as *lies, lying* and so on. If the default inflectional variant is reset to Inflection.IRREGULAR, the stored irregular forms will be used instead.Parameters:infl - the Inflection pattern with which this form is associatedlexicalFeature - the actual inflectional feature being set, for example LexicalFeature.PRESENT\_3Sform - the actual inflected word form

#### addInflectionalVariant public void addInflectionalVariant([Inflection](http://docs.google.com/simplenlg/features/Inflection.html) infl) Specify that this word has an inflectional variant (e.g. irregular)Parameters:infl - the variant

#### hasInflectionalVariant public boolean hasInflectionalVariant([Inflection](http://docs.google.com/simplenlg/features/Inflection.html) infl) Check whether this word has a particular inflectional variantParameters:infl - the variant Returns:true if this word has the variant

#### toString public java.lang.String toString()**Overrides:** [toString](http://docs.google.com/simplenlg/framework/NLGElement.html#toString()) in class [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html)

#### toXML public java.lang.String toXML()

#### getChildren public java.util.List<[NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html)> getChildren() This method returns an empty List as words do not have child elements.**Specified by:** [getChildren](http://docs.google.com/simplenlg/framework/NLGElement.html#getChildren()) in class [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) Returns:a List of NLGElements representing the children of this element.

#### printTree public java.lang.String printTree(java.lang.String indent)**Overrides:** [printTree](http://docs.google.com/simplenlg/framework/NLGElement.html#printTree(java.lang.String)) in class [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html)

#### equals public boolean equals(java.lang.Object o) Check if this WordElement is equal to an object.**Overrides:** [equals](http://docs.google.com/simplenlg/framework/NLGElement.html#equals(java.lang.Object)) in class [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) Parameters:o - the object Returns:true iff the object is a word element with the same id and the same baseform and the same features.

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/WordElement.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/StringElement.html)
* Next Class
* [Frames](http://docs.google.com/index.html?simplenlg/framework/WordElement.html)
* [No Frames](http://docs.google.com/WordElement.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#id.3znysh7)|
* [Method](#id.2et92p0)
* Detail:
* Field |
* [Constr](#id.1t3h5sf)|
* [Method](#id.26in1rg)