| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/TypeVariable.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) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/lang/model/type/TypeMirror.html)   [**NEXT CLASS**](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) | [**FRAMES**](http://docs.google.com/index.html?javax/lang/model/type/TypeVariable.html)    [**NO FRAMES**](http://docs.google.com/TypeVariable.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#tyjcwt) |

## **javax.lang.model.type**

Interface TypeVariable

**All Superinterfaces:** [ReferenceType](http://docs.google.com/javax/lang/model/type/ReferenceType.html), [TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html)

public interface **TypeVariable**extends [ReferenceType](http://docs.google.com/javax/lang/model/type/ReferenceType.html)

Represents a type variable. A type variable may be explicitly declared by a [type parameter](http://docs.google.com/javax/lang/model/element/TypeParameterElement.html) of a type, method, or constructor. A type variable may also be declared implicitly, as by the capture conversion of a wildcard type argument (see chapter 5 of *The Java Language Specification, Third Edition*).

**Since:** 1.6 **See Also:**[TypeParameterElement](http://docs.google.com/javax/lang/model/element/TypeParameterElement.html)

| **Method Summary** | |
| --- | --- |
| [Element](http://docs.google.com/javax/lang/model/element/Element.html) | [**asElement**](http://docs.google.com/javax/lang/model/type/TypeVariable.html#asElement())()            Returns the element corresponding to this type variable. |
| [TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) | [**getLowerBound**](http://docs.google.com/javax/lang/model/type/TypeVariable.html#getLowerBound())()            Returns the lower bound of this type variable. |
| [TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) | [**getUpperBound**](http://docs.google.com/javax/lang/model/type/TypeVariable.html#getUpperBound())()            Returns the upper bound of this type variable. |

| **Methods inherited from interface javax.lang.model.type.**[**TypeMirror**](http://docs.google.com/javax/lang/model/type/TypeMirror.html) |
| --- |
| [accept](http://docs.google.com/javax/lang/model/type/TypeMirror.html#accept(javax.lang.model.type.TypeVisitor,%20P)), [equals](http://docs.google.com/javax/lang/model/type/TypeMirror.html#equals(java.lang.Object)), [getKind](http://docs.google.com/javax/lang/model/type/TypeMirror.html#getKind()), [hashCode](http://docs.google.com/javax/lang/model/type/TypeMirror.html#hashCode()), [toString](http://docs.google.com/javax/lang/model/type/TypeMirror.html#toString()) |

| **Method Detail** |
| --- |

### asElement

[Element](http://docs.google.com/javax/lang/model/element/Element.html) **asElement**()

Returns the element corresponding to this type variable.

**Returns:**the element corresponding to this type variable

### getUpperBound

[TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) **getUpperBound**()

Returns the upper bound of this type variable.

If this type variable was declared with no explicit upper bounds, the result is java.lang.Object. If it was declared with multiple upper bounds, the result is an intersection type (modeled as a [DeclaredType](http://docs.google.com/javax/lang/model/type/DeclaredType.html)). Individual bounds can be found by examining the result's [supertypes](http://docs.google.com/javax/lang/model/util/Types.html#directSupertypes(javax.lang.model.type.TypeMirror)).

**Returns:**the upper bound of this type variable

### getLowerBound

[TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) **getLowerBound**()

Returns the lower bound of this type variable. While a type parameter cannot include an explicit lower bound declaration, capture conversion can produce a type variable with a non-trivial lower bound. Type variables otherwise have a lower bound of [NullType](http://docs.google.com/javax/lang/model/type/NullType.html).

**Returns:**the lower bound of this type variable

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/TypeVariable.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) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/lang/model/type/TypeMirror.html)   [**NEXT CLASS**](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) | [**FRAMES**](http://docs.google.com/index.html?javax/lang/model/type/TypeVariable.html)    [**NO FRAMES**](http://docs.google.com/TypeVariable.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#tyjcwt) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).