| | [**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/java/lang/reflect/Type.html)   [**NEXT CLASS**](http://docs.google.com/java/lang/reflect/UndeclaredThrowableException.html) | [**FRAMES**](http://docs.google.com/index.html?java/lang/reflect/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](#2et92p0) |

## **java.lang.reflect**

Interface TypeVariable<D extends [**GenericDeclaration**](http://docs.google.com/java/lang/reflect/GenericDeclaration.html)>

**Type Parameters:**D - the type of generic declaration that declared the underlying type variable. **All Superinterfaces:** [Type](http://docs.google.com/java/lang/reflect/Type.html)

public interface **TypeVariable<D extends** [**GenericDeclaration**](http://docs.google.com/java/lang/reflect/GenericDeclaration.html)**>**extends [Type](http://docs.google.com/java/lang/reflect/Type.html)

TypeVariable is the common superinterface for type variables of kinds. A type variable is created the first time it is needed by a reflective method, as specified in this package. If a type variable t is referenced by a type (i.e, class, interface or annotation type) T, and T is declared by the nth enclosing class of T (see JLS 8.1.2), then the creation of t requires the resolution (see JVMS 5) of the ith enclosing class of T, for i = 0 to n, inclusive. Creating a type variable must not cause the creation of its bounds. Repeated creation of a type variable has no effect.

Multiple objects may be instantiated at run-time to represent a given type variable. Even though a type variable is created only once, this does not imply any requirement to cache instances representing the type variable. However, all instances representing a type variable must be equal() to each other. As a consequence, users of type variables must not rely on the identity of instances of classes implementing this interface.

**Since:** 1.5

| **Method Summary** | |
| --- | --- |
| [Type](http://docs.google.com/java/lang/reflect/Type.html)[] | [**getBounds**](http://docs.google.com/java/lang/reflect/TypeVariable.html#getBounds())()            Returns an array of Type objects representing the upper bound(s) of this type variable. |
| [D](http://docs.google.com/java/lang/reflect/TypeVariable.html) | [**getGenericDeclaration**](http://docs.google.com/java/lang/reflect/TypeVariable.html#getGenericDeclaration())()            Returns the GenericDeclaration object representing the generic declaration declared this type variable. |
| [String](http://docs.google.com/java/lang/String.html) | [**getName**](http://docs.google.com/java/lang/reflect/TypeVariable.html#getName())()            Returns the name of this type variable, as it occurs in the source code. |

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

### getBounds

[Type](http://docs.google.com/java/lang/reflect/Type.html)[] **getBounds**()

Returns an array of Type objects representing the upper bound(s) of this type variable. Note that if no upper bound is explicitly declared, the upper bound is Object.

For each upper bound B:

* if B is a parameterized type or a type variable, it is created, (see [ParameterizedType](http://docs.google.com/java/lang/reflect/ParameterizedType.html) for the details of the creation process for parameterized types).
* Otherwise, B is resolved.

**Returns:**an array of Types representing the upper bound(s) of this type variable **Throws:** [TypeNotPresentException](http://docs.google.com/java/lang/TypeNotPresentException.html) - if any of the bounds refers to a non-existent type declaration [MalformedParameterizedTypeException](http://docs.google.com/java/lang/reflect/MalformedParameterizedTypeException.html) - if any of the bounds refer to a parameterized type that cannot be instantiated for any reason

### getGenericDeclaration

[D](http://docs.google.com/java/lang/reflect/TypeVariable.html) **getGenericDeclaration**()

Returns the GenericDeclaration object representing the generic declaration declared this type variable.

**Returns:**the generic declaration declared for this type variable.**Since:** 1.5

### getName

[String](http://docs.google.com/java/lang/String.html) **getName**()

Returns the name of this type variable, as it occurs in the source code.

**Returns:**the name of this type variable, as it appears in the source code

| | [**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/java/lang/reflect/Type.html)   [**NEXT CLASS**](http://docs.google.com/java/lang/reflect/UndeclaredThrowableException.html) | [**FRAMES**](http://docs.google.com/index.html?java/lang/reflect/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](#2et92p0) |

[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).