| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/TypeVisitor.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/TypeVariable.html)   [**NEXT CLASS**](http://docs.google.com/javax/lang/model/type/UnknownTypeException.html) | [**FRAMES**](http://docs.google.com/index.html?javax/lang/model/type/TypeVisitor.html)    [**NO FRAMES**](http://docs.google.com/TypeVisitor.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

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

Interface TypeVisitor<R,P>

**Type Parameters:**R - the return type of this visitor's methods. Use [Void](http://docs.google.com/java/lang/Void.html) for visitors that do not need to return results.P - the type of the additional parameter to this visitor's methods. Use Void for visitors that do not need an additional parameter. **All Known Implementing Classes:** [AbstractTypeVisitor6](http://docs.google.com/javax/lang/model/util/AbstractTypeVisitor6.html), [SimpleTypeVisitor6](http://docs.google.com/javax/lang/model/util/SimpleTypeVisitor6.html), [TypeKindVisitor6](http://docs.google.com/javax/lang/model/util/TypeKindVisitor6.html)

public interface **TypeVisitor<R,P>**

A visitor of types, in the style of the visitor design pattern. Classes implementing this interface are used to operate on a type when the kind of type is unknown at compile time. When a visitor is passed to a type's [accept](http://docs.google.com/javax/lang/model/type/TypeMirror.html#accept(javax.lang.model.type.TypeVisitor,%20P)) method, the visit*XYZ* method most applicable to that type is invoked.

Classes implementing this interface may or may not throw a NullPointerException if the additional parameter p is null; see documentation of the implementing class for details.

**WARNING:** It is possible that methods will be added to this interface to accommodate new, currently unknown, language structures added to future versions of the Java™ programming language. Therefore, visitor classes directly implementing this interface may be source incompatible with future versions of the platform. To avoid this source incompatibility, visitor implementations are encouraged to instead extend the appropriate abstract visitor class that implements this interface. However, an API should generally use this visitor interface as the type for parameters, return type, etc. rather than one of the abstract classes.

**Since:** 1.6

| **Method Summary** | |
| --- | --- |
| [R](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) | [**visit**](http://docs.google.com/javax/lang/model/type/TypeVisitor.html#visit(javax.lang.model.type.TypeMirror))([TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t)            A convenience method equivalent to v.visit(t, null). |
| [R](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) | [**visit**](http://docs.google.com/javax/lang/model/type/TypeVisitor.html#visit(javax.lang.model.type.TypeMirror,%20P))([TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t, [P](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) p)            Visits a type. |
| [R](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) | [**visitArray**](http://docs.google.com/javax/lang/model/type/TypeVisitor.html#visitArray(javax.lang.model.type.ArrayType,%20P))([ArrayType](http://docs.google.com/javax/lang/model/type/ArrayType.html) t, [P](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) p)            Visits an array type. |
| [R](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) | [**visitDeclared**](http://docs.google.com/javax/lang/model/type/TypeVisitor.html#visitDeclared(javax.lang.model.type.DeclaredType,%20P))([DeclaredType](http://docs.google.com/javax/lang/model/type/DeclaredType.html) t, [P](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) p)            Visits a declared type. |
| [R](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) | [**visitError**](http://docs.google.com/javax/lang/model/type/TypeVisitor.html#visitError(javax.lang.model.type.ErrorType,%20P))([ErrorType](http://docs.google.com/javax/lang/model/type/ErrorType.html) t, [P](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) p)            Visits an error type. |
| [R](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) | [**visitExecutable**](http://docs.google.com/javax/lang/model/type/TypeVisitor.html#visitExecutable(javax.lang.model.type.ExecutableType,%20P))([ExecutableType](http://docs.google.com/javax/lang/model/type/ExecutableType.html) t, [P](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) p)            Visits an executable type. |
| [R](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) | [**visitNoType**](http://docs.google.com/javax/lang/model/type/TypeVisitor.html#visitNoType(javax.lang.model.type.NoType,%20P))([NoType](http://docs.google.com/javax/lang/model/type/NoType.html) t, [P](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) p)            Visits a [NoType](http://docs.google.com/javax/lang/model/type/NoType.html) instance. |
| [R](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) | [**visitNull**](http://docs.google.com/javax/lang/model/type/TypeVisitor.html#visitNull(javax.lang.model.type.NullType,%20P))([NullType](http://docs.google.com/javax/lang/model/type/NullType.html) t, [P](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) p)            Visits the null type. |
| [R](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) | [**visitPrimitive**](http://docs.google.com/javax/lang/model/type/TypeVisitor.html#visitPrimitive(javax.lang.model.type.PrimitiveType,%20P))([PrimitiveType](http://docs.google.com/javax/lang/model/type/PrimitiveType.html) t, [P](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) p)            Visits a primitive type. |
| [R](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) | [**visitTypeVariable**](http://docs.google.com/javax/lang/model/type/TypeVisitor.html#visitTypeVariable(javax.lang.model.type.TypeVariable,%20P))([TypeVariable](http://docs.google.com/javax/lang/model/type/TypeVariable.html) t, [P](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) p)            Visits a type variable. |
| [R](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) | [**visitUnknown**](http://docs.google.com/javax/lang/model/type/TypeVisitor.html#visitUnknown(javax.lang.model.type.TypeMirror,%20P))([TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t, [P](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) p)            Visits an unknown kind of type. |
| [R](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) | [**visitWildcard**](http://docs.google.com/javax/lang/model/type/TypeVisitor.html#visitWildcard(javax.lang.model.type.WildcardType,%20P))([WildcardType](http://docs.google.com/javax/lang/model/type/WildcardType.html) t, [P](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) p)            Visits a wildcard type. |

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

### visit

[R](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) **visit**([TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t,  
 [P](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) p)

Visits a type.

**Parameters:**t - the type to visitp - a visitor-specified parameter **Returns:**a visitor-specified result

### visit

[R](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) **visit**([TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t)

A convenience method equivalent to v.visit(t, null).

**Parameters:**t - the element to visit **Returns:**a visitor-specified result

### visitPrimitive

[R](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) **visitPrimitive**([PrimitiveType](http://docs.google.com/javax/lang/model/type/PrimitiveType.html) t,  
 [P](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) p)

Visits a primitive type.

**Parameters:**t - the type to visitp - a visitor-specified parameter **Returns:**a visitor-specified result

### visitNull

[R](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) **visitNull**([NullType](http://docs.google.com/javax/lang/model/type/NullType.html) t,  
 [P](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) p)

Visits the null type.

**Parameters:**t - the type to visitp - a visitor-specified parameter **Returns:**a visitor-specified result

### visitArray

[R](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) **visitArray**([ArrayType](http://docs.google.com/javax/lang/model/type/ArrayType.html) t,  
 [P](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) p)

Visits an array type.

**Parameters:**t - the type to visitp - a visitor-specified parameter **Returns:**a visitor-specified result

### visitDeclared

[R](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) **visitDeclared**([DeclaredType](http://docs.google.com/javax/lang/model/type/DeclaredType.html) t,  
 [P](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) p)

Visits a declared type.

**Parameters:**t - the type to visitp - a visitor-specified parameter **Returns:**a visitor-specified result

### visitError

[R](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) **visitError**([ErrorType](http://docs.google.com/javax/lang/model/type/ErrorType.html) t,  
 [P](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) p)

Visits an error type.

**Parameters:**t - the type to visitp - a visitor-specified parameter **Returns:**a visitor-specified result

### visitTypeVariable

[R](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) **visitTypeVariable**([TypeVariable](http://docs.google.com/javax/lang/model/type/TypeVariable.html) t,  
 [P](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) p)

Visits a type variable.

**Parameters:**t - the type to visitp - a visitor-specified parameter **Returns:**a visitor-specified result

### visitWildcard

[R](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) **visitWildcard**([WildcardType](http://docs.google.com/javax/lang/model/type/WildcardType.html) t,  
 [P](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) p)

Visits a wildcard type.

**Parameters:**t - the type to visitp - a visitor-specified parameter **Returns:**a visitor-specified result

### visitExecutable

[R](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) **visitExecutable**([ExecutableType](http://docs.google.com/javax/lang/model/type/ExecutableType.html) t,  
 [P](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) p)

Visits an executable type.

**Parameters:**t - the type to visitp - a visitor-specified parameter **Returns:**a visitor-specified result

### visitNoType

[R](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) **visitNoType**([NoType](http://docs.google.com/javax/lang/model/type/NoType.html) t,  
 [P](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) p)

Visits a [NoType](http://docs.google.com/javax/lang/model/type/NoType.html) instance.

**Parameters:**t - the type to visitp - a visitor-specified parameter **Returns:**a visitor-specified result

### visitUnknown

[R](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) **visitUnknown**([TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t,  
 [P](http://docs.google.com/javax/lang/model/type/TypeVisitor.html) p)

Visits an unknown kind of type. This can occur if the language evolves and new kinds of types are added to the TypeMirror hierarchy.

**Parameters:**t - the type to visitp - a visitor-specified parameter **Returns:**a visitor-specified result **Throws:** [UnknownTypeException](http://docs.google.com/javax/lang/model/type/UnknownTypeException.html) - a visitor implementation may optionally throw this exception

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/TypeVisitor.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/TypeVariable.html)   [**NEXT CLASS**](http://docs.google.com/javax/lang/model/type/UnknownTypeException.html) | [**FRAMES**](http://docs.google.com/index.html?javax/lang/model/type/TypeVisitor.html)    [**NO FRAMES**](http://docs.google.com/TypeVisitor.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).