| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/AbstractTypeVisitor6.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/util/AbstractElementVisitor6.html)   [**NEXT CLASS**](http://docs.google.com/javax/lang/model/util/ElementFilter.html) | [**FRAMES**](http://docs.google.com/index.html?javax/lang/model/util/AbstractTypeVisitor6.html)    [**NO FRAMES**](http://docs.google.com/AbstractTypeVisitor6.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#2s8eyo1) |

## **javax.lang.model.util**

Class AbstractTypeVisitor6<R,P>

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.lang.model.util.AbstractTypeVisitor6<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 Implemented Interfaces:** [TypeVisitor](http://docs.google.com/javax/lang/model/type/TypeVisitor.html)<R,P> **Direct Known Subclasses:** [SimpleTypeVisitor6](http://docs.google.com/javax/lang/model/util/SimpleTypeVisitor6.html)

public abstract class **AbstractTypeVisitor6<R,P>**extends [Object](http://docs.google.com/java/lang/Object.html)implements [TypeVisitor](http://docs.google.com/javax/lang/model/type/TypeVisitor.html)<R,P>

A skeletal visitor of types with default behavior appropriate for the version 6 language level.

**WARNING:** The TypeVisitor interface implemented by this class may have methods added to it in the future to accommodate new, currently unknown, language structures added to future versions of the Java™ programming language. Therefore, methods whose names begin with "visit" may be added to this class in the future; to avoid incompatibilities, classes which extend this class should not declare any instance methods with names beginning with "visit".

When such a new visit method is added, the default implementation in this class will be to call the [visitUnknown](http://docs.google.com/javax/lang/model/util/AbstractTypeVisitor6.html#visitUnknown(javax.lang.model.type.TypeMirror,%20P)) method. A new abstract type visitor class will also be introduced to correspond to the new language level; this visitor will have different default behavior for the visit method in question. When the new visitor is introduced, all or portions of this visitor may be deprecated.

**Since:** 1.6

| **Constructor Summary** | |
| --- | --- |
| protected | [**AbstractTypeVisitor6**](http://docs.google.com/javax/lang/model/util/AbstractTypeVisitor6.html#AbstractTypeVisitor6())()            Constructor for concrete subclasses to call. |

| **Method Summary** | |
| --- | --- |
| [R](http://docs.google.com/javax/lang/model/util/AbstractTypeVisitor6.html) | [**visit**](http://docs.google.com/javax/lang/model/util/AbstractTypeVisitor6.html#visit(javax.lang.model.type.TypeMirror))([TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t)            Visits any type mirror as if by passing itself to that type mirror's [accept](http://docs.google.com/javax/lang/model/type/TypeMirror.html#accept(javax.lang.model.type.TypeVisitor,%20P)) method and passing null for the additional parameter. |
| [R](http://docs.google.com/javax/lang/model/util/AbstractTypeVisitor6.html) | [**visit**](http://docs.google.com/javax/lang/model/util/AbstractTypeVisitor6.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/util/AbstractTypeVisitor6.html) p)            Visits any type mirror as if by passing itself to that type mirror's [accept](http://docs.google.com/javax/lang/model/type/TypeMirror.html#accept(javax.lang.model.type.TypeVisitor,%20P)) method. |
| [R](http://docs.google.com/javax/lang/model/util/AbstractTypeVisitor6.html) | [**visitUnknown**](http://docs.google.com/javax/lang/model/util/AbstractTypeVisitor6.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/util/AbstractTypeVisitor6.html) p)            Visits an unknown kind of type. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Methods inherited from interface javax.lang.model.type.**[**TypeVisitor**](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)), [visitDeclared](http://docs.google.com/javax/lang/model/type/TypeVisitor.html#visitDeclared(javax.lang.model.type.DeclaredType,%20P)), [visitError](http://docs.google.com/javax/lang/model/type/TypeVisitor.html#visitError(javax.lang.model.type.ErrorType,%20P)), [visitExecutable](http://docs.google.com/javax/lang/model/type/TypeVisitor.html#visitExecutable(javax.lang.model.type.ExecutableType,%20P)), [visitNoType](http://docs.google.com/javax/lang/model/type/TypeVisitor.html#visitNoType(javax.lang.model.type.NoType,%20P)), [visitNull](http://docs.google.com/javax/lang/model/type/TypeVisitor.html#visitNull(javax.lang.model.type.NullType,%20P)), [visitPrimitive](http://docs.google.com/javax/lang/model/type/TypeVisitor.html#visitPrimitive(javax.lang.model.type.PrimitiveType,%20P)), [visitTypeVariable](http://docs.google.com/javax/lang/model/type/TypeVisitor.html#visitTypeVariable(javax.lang.model.type.TypeVariable,%20P)), [visitWildcard](http://docs.google.com/javax/lang/model/type/TypeVisitor.html#visitWildcard(javax.lang.model.type.WildcardType,%20P)) |

| **Constructor Detail** |
| --- |

### AbstractTypeVisitor6

protected **AbstractTypeVisitor6**()

Constructor for concrete subclasses to call.

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

### visit

public final [R](http://docs.google.com/javax/lang/model/util/AbstractTypeVisitor6.html) **visit**([TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t,  
 [P](http://docs.google.com/javax/lang/model/util/AbstractTypeVisitor6.html) p)

Visits any type mirror as if by passing itself to that type mirror's [accept](http://docs.google.com/javax/lang/model/type/TypeMirror.html#accept(javax.lang.model.type.TypeVisitor,%20P)) method. The invocation v.visit(t, p) is equivalent to t.accept(v, p).

**Specified by:**[visit](http://docs.google.com/javax/lang/model/type/TypeVisitor.html#visit(javax.lang.model.type.TypeMirror,%20P)) in interface [TypeVisitor](http://docs.google.com/javax/lang/model/type/TypeVisitor.html)<[R](http://docs.google.com/javax/lang/model/util/AbstractTypeVisitor6.html),[P](http://docs.google.com/javax/lang/model/util/AbstractTypeVisitor6.html)> **Parameters:**t - the type to visitp - a visitor-specified parameter **Returns:**a visitor-specified result

### visit

public final [R](http://docs.google.com/javax/lang/model/util/AbstractTypeVisitor6.html) **visit**([TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t)

Visits any type mirror as if by passing itself to that type mirror's [accept](http://docs.google.com/javax/lang/model/type/TypeMirror.html#accept(javax.lang.model.type.TypeVisitor,%20P)) method and passing null for the additional parameter. The invocation v.visit(t) is equivalent to t.accept(v, null).

**Specified by:**[visit](http://docs.google.com/javax/lang/model/type/TypeVisitor.html#visit(javax.lang.model.type.TypeMirror)) in interface [TypeVisitor](http://docs.google.com/javax/lang/model/type/TypeVisitor.html)<[R](http://docs.google.com/javax/lang/model/util/AbstractTypeVisitor6.html),[P](http://docs.google.com/javax/lang/model/util/AbstractTypeVisitor6.html)> **Parameters:**t - the type to visit **Returns:**a visitor-specified result

### visitUnknown

public [R](http://docs.google.com/javax/lang/model/util/AbstractTypeVisitor6.html) **visitUnknown**([TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t,  
 [P](http://docs.google.com/javax/lang/model/util/AbstractTypeVisitor6.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.

The default implementation of this method in AbstractTypeVisitor6 will always throw UnknownTypeException. This behavior is not required of a subclass.

**Specified by:**[visitUnknown](http://docs.google.com/javax/lang/model/type/TypeVisitor.html#visitUnknown(javax.lang.model.type.TypeMirror,%20P)) in interface [TypeVisitor](http://docs.google.com/javax/lang/model/type/TypeVisitor.html)<[R](http://docs.google.com/javax/lang/model/util/AbstractTypeVisitor6.html),[P](http://docs.google.com/javax/lang/model/util/AbstractTypeVisitor6.html)> **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/AbstractTypeVisitor6.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/util/AbstractElementVisitor6.html)   [**NEXT CLASS**](http://docs.google.com/javax/lang/model/util/ElementFilter.html) | [**FRAMES**](http://docs.google.com/index.html?javax/lang/model/util/AbstractTypeVisitor6.html)    [**NO FRAMES**](http://docs.google.com/AbstractTypeVisitor6.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#2s8eyo1) |

[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).