| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Types.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/TypeKindVisitor6.html)   NEXT CLASS | [**FRAMES**](http://docs.google.com/index.html?javax/lang/model/util/Types.html)    [**NO FRAMES**](http://docs.google.com/Types.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

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

Interface Types

public interface **Types**

Utility methods for operating on types.

**Compatibility Note:** Methods may be added to this interface in future releases of the platform.

**Since:** 1.6 **See Also:**[ProcessingEnvironment.getTypeUtils()](http://docs.google.com/javax/annotation/processing/ProcessingEnvironment.html#getTypeUtils())

| **Method Summary** | |
| --- | --- |
| [Element](http://docs.google.com/javax/lang/model/element/Element.html) | [**asElement**](http://docs.google.com/javax/lang/model/util/Types.html#asElement(javax.lang.model.type.TypeMirror))([TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t)            Returns the element corresponding to a type. |
| [TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) | [**asMemberOf**](http://docs.google.com/javax/lang/model/util/Types.html#asMemberOf(javax.lang.model.type.DeclaredType,%20javax.lang.model.element.Element))([DeclaredType](http://docs.google.com/javax/lang/model/type/DeclaredType.html) containing, [Element](http://docs.google.com/javax/lang/model/element/Element.html) element)            Returns the type of an element when that element is viewed as a member of, or otherwise directly contained by, a given type. |
| [TypeElement](http://docs.google.com/javax/lang/model/element/TypeElement.html) | [**boxedClass**](http://docs.google.com/javax/lang/model/util/Types.html#boxedClass(javax.lang.model.type.PrimitiveType))([PrimitiveType](http://docs.google.com/javax/lang/model/type/PrimitiveType.html) p)            Returns the class of a boxed value of a given primitive type. |
| [TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) | [**capture**](http://docs.google.com/javax/lang/model/util/Types.html#capture(javax.lang.model.type.TypeMirror))([TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t)            Applies capture conversion to a type. |
| boolean | [**contains**](http://docs.google.com/javax/lang/model/util/Types.html#contains(javax.lang.model.type.TypeMirror,%20javax.lang.model.type.TypeMirror))([TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t1, [TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t2)            Tests whether one type argument *contains* another. |
| [List](http://docs.google.com/java/util/List.html)<? extends [TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html)> | [**directSupertypes**](http://docs.google.com/javax/lang/model/util/Types.html#directSupertypes(javax.lang.model.type.TypeMirror))([TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t)            Returns the direct supertypes of a type. |
| [TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) | [**erasure**](http://docs.google.com/javax/lang/model/util/Types.html#erasure(javax.lang.model.type.TypeMirror))([TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t)            Returns the erasure of a type. |
| [ArrayType](http://docs.google.com/javax/lang/model/type/ArrayType.html) | [**getArrayType**](http://docs.google.com/javax/lang/model/util/Types.html#getArrayType(javax.lang.model.type.TypeMirror))([TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) componentType)            Returns an array type with the specified component type. |
| [DeclaredType](http://docs.google.com/javax/lang/model/type/DeclaredType.html) | [**getDeclaredType**](http://docs.google.com/javax/lang/model/util/Types.html#getDeclaredType(javax.lang.model.type.DeclaredType,%20javax.lang.model.element.TypeElement,%20javax.lang.model.type.TypeMirror...))([DeclaredType](http://docs.google.com/javax/lang/model/type/DeclaredType.html) containing, [TypeElement](http://docs.google.com/javax/lang/model/element/TypeElement.html) typeElem, [TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html)... typeArgs)            Returns the type corresponding to a type element and actual type arguments, given a [containing type](http://docs.google.com/javax/lang/model/type/DeclaredType.html#getEnclosingType()) of which it is a member. |
| [DeclaredType](http://docs.google.com/javax/lang/model/type/DeclaredType.html) | [**getDeclaredType**](http://docs.google.com/javax/lang/model/util/Types.html#getDeclaredType(javax.lang.model.element.TypeElement,%20javax.lang.model.type.TypeMirror...))([TypeElement](http://docs.google.com/javax/lang/model/element/TypeElement.html) typeElem, [TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html)... typeArgs)            Returns the type corresponding to a type element and actual type arguments. |
| [NoType](http://docs.google.com/javax/lang/model/type/NoType.html) | [**getNoType**](http://docs.google.com/javax/lang/model/util/Types.html#getNoType(javax.lang.model.type.TypeKind))([TypeKind](http://docs.google.com/javax/lang/model/type/TypeKind.html) kind)            Returns a pseudo-type used where no actual type is appropriate. |
| [NullType](http://docs.google.com/javax/lang/model/type/NullType.html) | [**getNullType**](http://docs.google.com/javax/lang/model/util/Types.html#getNullType())()            Returns the null type. |
| [PrimitiveType](http://docs.google.com/javax/lang/model/type/PrimitiveType.html) | [**getPrimitiveType**](http://docs.google.com/javax/lang/model/util/Types.html#getPrimitiveType(javax.lang.model.type.TypeKind))([TypeKind](http://docs.google.com/javax/lang/model/type/TypeKind.html) kind)            Returns a primitive type. |
| [WildcardType](http://docs.google.com/javax/lang/model/type/WildcardType.html) | [**getWildcardType**](http://docs.google.com/javax/lang/model/util/Types.html#getWildcardType(javax.lang.model.type.TypeMirror,%20javax.lang.model.type.TypeMirror))([TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) extendsBound, [TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) superBound)            Returns a new wildcard type argument. |
| boolean | [**isAssignable**](http://docs.google.com/javax/lang/model/util/Types.html#isAssignable(javax.lang.model.type.TypeMirror,%20javax.lang.model.type.TypeMirror))([TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t1, [TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t2)            Tests whether one type is assignable to another. |
| boolean | [**isSameType**](http://docs.google.com/javax/lang/model/util/Types.html#isSameType(javax.lang.model.type.TypeMirror,%20javax.lang.model.type.TypeMirror))([TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t1, [TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t2)            Tests whether two TypeMirror objects represent the same type. |
| boolean | [**isSubsignature**](http://docs.google.com/javax/lang/model/util/Types.html#isSubsignature(javax.lang.model.type.ExecutableType,%20javax.lang.model.type.ExecutableType))([ExecutableType](http://docs.google.com/javax/lang/model/type/ExecutableType.html) m1, [ExecutableType](http://docs.google.com/javax/lang/model/type/ExecutableType.html) m2)            Tests whether the signature of one method is a *subsignature* of another. |
| boolean | [**isSubtype**](http://docs.google.com/javax/lang/model/util/Types.html#isSubtype(javax.lang.model.type.TypeMirror,%20javax.lang.model.type.TypeMirror))([TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t1, [TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t2)            Tests whether one type is a subtype of another. |
| [PrimitiveType](http://docs.google.com/javax/lang/model/type/PrimitiveType.html) | [**unboxedType**](http://docs.google.com/javax/lang/model/util/Types.html#unboxedType(javax.lang.model.type.TypeMirror))([TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t)            Returns the type (a primitive type) of unboxed values of a given type. |

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

### asElement

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

Returns the element corresponding to a type. The type may be a DeclaredType or TypeVariable. Returns null if the type is not one with a corresponding element.

**Returns:**the element corresponding to the given type

### isSameType

boolean **isSameType**([TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t1,  
 [TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t2)

Tests whether two TypeMirror objects represent the same type.

Caveat: if either of the arguments to this method represents a wildcard, this method will return false. As a consequence, a wildcard is not the same type as itself. This might be surprising at first, but makes sense once you consider that an example like this must be rejected by the compiler:

List<?> list = new ArrayList<Object>();  
 list.add(list.get(0));

**Parameters:**t1 - the first typet2 - the second type **Returns:**true if and only if the two types are the same

### isSubtype

boolean **isSubtype**([TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t1,  
 [TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t2)

Tests whether one type is a subtype of another. Any type is considered to be a subtype of itself.

**Parameters:**t1 - the first typet2 - the second type **Returns:**true if and only if the first type is a subtype of the second **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if given an executable or package type**See**  [**The Java Language Specification, Third Edition**](http://java.sun.com/docs/books/jls/)**:** 4.10 Subtyping

### isAssignable

boolean **isAssignable**([TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t1,  
 [TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t2)

Tests whether one type is assignable to another.

**Parameters:**t1 - the first typet2 - the second type **Returns:**true if and only if the first type is assignable to the second **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if given an executable or package type**See**  [**The Java Language Specification, Third Edition**](http://java.sun.com/docs/books/jls/)**:** 5.2 Assignment Conversion

### contains

boolean **contains**([TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t1,  
 [TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t2)

Tests whether one type argument *contains* another.

**Parameters:**t1 - the first typet2 - the second type **Returns:**true if and only if the first type contains the second **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if given an executable or package type**See**  [**The Java Language Specification, Third Edition**](http://java.sun.com/docs/books/jls/)**:** 4.5.1.1 Type Argument Containment and Equivalence

### isSubsignature

boolean **isSubsignature**([ExecutableType](http://docs.google.com/javax/lang/model/type/ExecutableType.html) m1,  
 [ExecutableType](http://docs.google.com/javax/lang/model/type/ExecutableType.html) m2)

Tests whether the signature of one method is a *subsignature* of another.

**Parameters:**m1 - the first methodm2 - the second method **Returns:**true if and only if the first signature is a subsignature of the second**See**  [**The Java Language Specification, Third Edition**](http://java.sun.com/docs/books/jls/)**:** 8.4.2 Method Signature

### directSupertypes

[List](http://docs.google.com/java/util/List.html)<? extends [TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html)> **directSupertypes**([TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) t)

Returns the direct supertypes of a type. The interface types, if any, will appear last in the list.

**Parameters:**t - the type being examined **Returns:**the direct supertypes, or an empty list if none **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if given an executable or package type

### erasure

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

Returns the erasure of a type.

**Parameters:**t - the type to be erased **Returns:**the erasure of the given type **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if given a package type**See**  [**The Java Language Specification, Third Edition**](http://java.sun.com/docs/books/jls/)**:** 4.6 Type Erasure

### boxedClass

[TypeElement](http://docs.google.com/javax/lang/model/element/TypeElement.html) **boxedClass**([PrimitiveType](http://docs.google.com/javax/lang/model/type/PrimitiveType.html) p)

Returns the class of a boxed value of a given primitive type. That is, *boxing conversion* is applied.

**Parameters:**p - the primitive type to be converted **Returns:**the class of a boxed value of type p**See**  [**The Java Language Specification, Third Edition**](http://java.sun.com/docs/books/jls/)**:** 5.1.7 Boxing Conversion

### unboxedType

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

Returns the type (a primitive type) of unboxed values of a given type. That is, *unboxing conversion* is applied.

**Parameters:**t - the type to be unboxed **Returns:**the type of an unboxed value of type t **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the given type has no unboxing conversion**See**  [**The Java Language Specification, Third Edition**](http://java.sun.com/docs/books/jls/)**:** 5.1.8 Unboxing Conversion

### capture

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

Applies capture conversion to a type.

**Parameters:**t - the type to be converted **Returns:**the result of applying capture conversion **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if given an executable or package type**See**  [**The Java Language Specification, Third Edition**](http://java.sun.com/docs/books/jls/)**:** 5.1.10 Capture Conversion

### getPrimitiveType

[PrimitiveType](http://docs.google.com/javax/lang/model/type/PrimitiveType.html) **getPrimitiveType**([TypeKind](http://docs.google.com/javax/lang/model/type/TypeKind.html) kind)

Returns a primitive type.

**Parameters:**kind - the kind of primitive type to return **Returns:**a primitive type **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if kind is not a primitive kind

### getNullType

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

Returns the null type. This is the type of null.

**Returns:**the null type

### getNoType

[NoType](http://docs.google.com/javax/lang/model/type/NoType.html) **getNoType**([TypeKind](http://docs.google.com/javax/lang/model/type/TypeKind.html) kind)

Returns a pseudo-type used where no actual type is appropriate. The kind of type to return may be either [VOID](http://docs.google.com/javax/lang/model/type/TypeKind.html#VOID) or [NONE](http://docs.google.com/javax/lang/model/type/TypeKind.html#NONE). For packages, use [Elements.getPackageElement(CharSequence)](http://docs.google.com/javax/lang/model/util/Elements.html#getPackageElement(java.lang.CharSequence)).asType() instead.

**Parameters:**kind - the kind of type to return **Returns:**a pseudo-type of kind VOID or NONE **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if kind is not valid

### getArrayType

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

Returns an array type with the specified component type.

**Parameters:**componentType - the component type **Returns:**an array type with the specified component type. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the component type is not valid for an array

### getWildcardType

[WildcardType](http://docs.google.com/javax/lang/model/type/WildcardType.html) **getWildcardType**([TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) extendsBound,  
 [TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) superBound)

Returns a new wildcard type argument. Either of the wildcard's bounds may be specified, or neither, but not both.

**Parameters:**extendsBound - the extends (upper) bound, or null if nonesuperBound - the super (lower) bound, or null if none **Returns:**a new wildcard **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if bounds are not valid

### getDeclaredType

[DeclaredType](http://docs.google.com/javax/lang/model/type/DeclaredType.html) **getDeclaredType**([TypeElement](http://docs.google.com/javax/lang/model/element/TypeElement.html) typeElem,  
 [TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html)... typeArgs)

Returns the type corresponding to a type element and actual type arguments. Given the type element for Set and the type mirror for String, for example, this method may be used to get the parameterized type Set<String>.

The number of type arguments must either equal the number of the type element's formal type parameters, or must be zero. If zero, and if the type element is generic, then the type element's raw type is returned.

If a parameterized type is being returned, its type element must not be contained within a generic outer class. The parameterized type Outer<String>.Inner<Number>, for example, may be constructed by first using this method to get the type Outer<String>, and then invoking [getDeclaredType(DeclaredType, TypeElement, TypeMirror...)](http://docs.google.com/javax/lang/model/util/Types.html#getDeclaredType(javax.lang.model.type.DeclaredType,%20javax.lang.model.element.TypeElement,%20javax.lang.model.type.TypeMirror...)).

**Parameters:**typeElem - the type elementtypeArgs - the actual type arguments **Returns:**the type corresponding to the type element and actual type arguments **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if too many or too few type arguments are given, or if an inappropriate type argument or type element is provided

### getDeclaredType

[DeclaredType](http://docs.google.com/javax/lang/model/type/DeclaredType.html) **getDeclaredType**([DeclaredType](http://docs.google.com/javax/lang/model/type/DeclaredType.html) containing,  
 [TypeElement](http://docs.google.com/javax/lang/model/element/TypeElement.html) typeElem,  
 [TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html)... typeArgs)

Returns the type corresponding to a type element and actual type arguments, given a [containing type](http://docs.google.com/javax/lang/model/type/DeclaredType.html#getEnclosingType()) of which it is a member. The parameterized type Outer<String>.Inner<Number>, for example, may be constructed by first using [getDeclaredType(TypeElement, TypeMirror...)](http://docs.google.com/javax/lang/model/util/Types.html#getDeclaredType(javax.lang.model.element.TypeElement,%20javax.lang.model.type.TypeMirror...)) to get the type Outer<String>, and then invoking this method.

If the containing type is a parameterized type, the number of type arguments must equal the number of typeElem's formal type parameters. If it is not parameterized or if it is null, this method is equivalent to getDeclaredType(typeElem, typeArgs).

**Parameters:**containing - the containing type, or null if nonetypeElem - the type elementtypeArgs - the actual type arguments **Returns:**the type corresponding to the type element and actual type arguments, contained within the given type **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if too many or too few type arguments are given, or if an inappropriate type argument, type element, or containing type is provided

### asMemberOf

[TypeMirror](http://docs.google.com/javax/lang/model/type/TypeMirror.html) **asMemberOf**([DeclaredType](http://docs.google.com/javax/lang/model/type/DeclaredType.html) containing,  
 [Element](http://docs.google.com/javax/lang/model/element/Element.html) element)

Returns the type of an element when that element is viewed as a member of, or otherwise directly contained by, a given type. For example, when viewed as a member of the parameterized type Set<String>, the Set.add method is an ExecutableType whose parameter is of type String.

**Parameters:**containing - the containing typeelement - the element **Returns:**the type of the element as viewed from the containing type **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the element is not a valid one for the given type

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Types.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/TypeKindVisitor6.html)   NEXT CLASS | [**FRAMES**](http://docs.google.com/index.html?javax/lang/model/util/Types.html)    [**NO FRAMES**](http://docs.google.com/Types.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).