| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Method.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/Member.html)   [**NEXT CLASS**](http://docs.google.com/java/lang/reflect/Modifier.html) | [**FRAMES**](http://docs.google.com/index.html?java/lang/reflect/Method.html)    [**NO FRAMES**](http://docs.google.com/Method.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#tyjcwt) | DETAIL: FIELD | CONSTR | [METHOD](#4d34og8) |

## **java.lang.reflect**

Class Method

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.lang.reflect.AccessibleObject](http://docs.google.com/java/lang/reflect/AccessibleObject.html)  
 **java.lang.reflect.Method**

**All Implemented Interfaces:** [AnnotatedElement](http://docs.google.com/java/lang/reflect/AnnotatedElement.html), [GenericDeclaration](http://docs.google.com/java/lang/reflect/GenericDeclaration.html), [Member](http://docs.google.com/java/lang/reflect/Member.html)

public final class **Method**extends [AccessibleObject](http://docs.google.com/java/lang/reflect/AccessibleObject.html)implements [GenericDeclaration](http://docs.google.com/java/lang/reflect/GenericDeclaration.html), [Member](http://docs.google.com/java/lang/reflect/Member.html)

A Method provides information about, and access to, a single method on a class or interface. The reflected method may be a class method or an instance method (including an abstract method).

A Method permits widening conversions to occur when matching the actual parameters to invoke with the underlying method's formal parameters, but it throws an IllegalArgumentException if a narrowing conversion would occur.

**See Also:**[Member](http://docs.google.com/java/lang/reflect/Member.html), [Class](http://docs.google.com/java/lang/Class.html), [Class.getMethods()](http://docs.google.com/java/lang/Class.html#getMethods()), [Class.getMethod(String, Class[])](http://docs.google.com/java/lang/Class.html#getMethod(java.lang.String,%20java.lang.Class...)), [Class.getDeclaredMethods()](http://docs.google.com/java/lang/Class.html#getDeclaredMethods()), [Class.getDeclaredMethod(String, Class[])](http://docs.google.com/java/lang/Class.html#getDeclaredMethod(java.lang.String,%20java.lang.Class...))

| **Field Summary** | |
| --- | --- |

| **Fields inherited from interface java.lang.reflect.**[**Member**](http://docs.google.com/java/lang/reflect/Member.html) |
| --- |
| [DECLARED](http://docs.google.com/java/lang/reflect/Member.html#DECLARED), [PUBLIC](http://docs.google.com/java/lang/reflect/Member.html#PUBLIC) |

| **Method Summary** | |
| --- | --- |
| boolean | [**equals**](http://docs.google.com/java/lang/reflect/Method.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Compares this Method against the specified object. |
| | <T extends [Annotation](http://docs.google.com/java/lang/annotation/Annotation.html)>  T | | --- | | [**getAnnotation**](http://docs.google.com/java/lang/reflect/Method.html#getAnnotation(java.lang.Class))([Class](http://docs.google.com/java/lang/Class.html)<T> annotationClass)            Returns this element's annotation for the specified type if such an annotation is present, else null. |
| [Annotation](http://docs.google.com/java/lang/annotation/Annotation.html)[] | [**getDeclaredAnnotations**](http://docs.google.com/java/lang/reflect/Method.html#getDeclaredAnnotations())()            Returns all annotations that are directly present on this element. |
| [Class](http://docs.google.com/java/lang/Class.html)<?> | [**getDeclaringClass**](http://docs.google.com/java/lang/reflect/Method.html#getDeclaringClass())()            Returns the Class object representing the class or interface that declares the method represented by this Method object. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getDefaultValue**](http://docs.google.com/java/lang/reflect/Method.html#getDefaultValue())()            Returns the default value for the annotation member represented by this Method instance. |
| [Class](http://docs.google.com/java/lang/Class.html)<?>[] | [**getExceptionTypes**](http://docs.google.com/java/lang/reflect/Method.html#getExceptionTypes())()            Returns an array of Class objects that represent the types of the exceptions declared to be thrown by the underlying method represented by this Method object. |
| [Type](http://docs.google.com/java/lang/reflect/Type.html)[] | [**getGenericExceptionTypes**](http://docs.google.com/java/lang/reflect/Method.html#getGenericExceptionTypes())()            Returns an array of Type objects that represent the exceptions declared to be thrown by this Method object. |
| [Type](http://docs.google.com/java/lang/reflect/Type.html)[] | [**getGenericParameterTypes**](http://docs.google.com/java/lang/reflect/Method.html#getGenericParameterTypes())()            Returns an array of Type objects that represent the formal parameter types, in declaration order, of the method represented by this Method object. |
| [Type](http://docs.google.com/java/lang/reflect/Type.html) | [**getGenericReturnType**](http://docs.google.com/java/lang/reflect/Method.html#getGenericReturnType())()            Returns a Type object that represents the formal return type of the method represented by this Method object. |
| int | [**getModifiers**](http://docs.google.com/java/lang/reflect/Method.html#getModifiers())()            Returns the Java language modifiers for the method represented by this Method object, as an integer. |
| [String](http://docs.google.com/java/lang/String.html) | [**getName**](http://docs.google.com/java/lang/reflect/Method.html#getName())()            Returns the name of the method represented by this Method object, as a String. |
| [Annotation](http://docs.google.com/java/lang/annotation/Annotation.html)[][] | [**getParameterAnnotations**](http://docs.google.com/java/lang/reflect/Method.html#getParameterAnnotations())()            Returns an array of arrays that represent the annotations on the formal parameters, in declaration order, of the method represented by this Method object. |
| [Class](http://docs.google.com/java/lang/Class.html)<?>[] | [**getParameterTypes**](http://docs.google.com/java/lang/reflect/Method.html#getParameterTypes())()            Returns an array of Class objects that represent the formal parameter types, in declaration order, of the method represented by this Method object. |
| [Class](http://docs.google.com/java/lang/Class.html)<?> | [**getReturnType**](http://docs.google.com/java/lang/reflect/Method.html#getReturnType())()            Returns a Class object that represents the formal return type of the method represented by this Method object. |
| [TypeVariable](http://docs.google.com/java/lang/reflect/TypeVariable.html)<[Method](http://docs.google.com/java/lang/reflect/Method.html)>[] | [**getTypeParameters**](http://docs.google.com/java/lang/reflect/Method.html#getTypeParameters())()            Returns an array of TypeVariable objects that represent the type variables declared by the generic declaration represented by this GenericDeclaration object, in declaration order. |
| int | [**hashCode**](http://docs.google.com/java/lang/reflect/Method.html#hashCode())()            Returns a hashcode for this Method. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**invoke**](http://docs.google.com/java/lang/reflect/Method.html#invoke(java.lang.Object,%20java.lang.Object...))([Object](http://docs.google.com/java/lang/Object.html) obj, [Object](http://docs.google.com/java/lang/Object.html)... args)            Invokes the underlying method represented by this Method object, on the specified object with the specified parameters. |
| boolean | [**isBridge**](http://docs.google.com/java/lang/reflect/Method.html#isBridge())()            Returns true if this method is a bridge method; returns false otherwise. |
| boolean | [**isSynthetic**](http://docs.google.com/java/lang/reflect/Method.html#isSynthetic())()            Returns true if this method is a synthetic method; returns false otherwise. |
| boolean | [**isVarArgs**](http://docs.google.com/java/lang/reflect/Method.html#isVarArgs())()            Returns true if this method was declared to take a variable number of arguments; returns false otherwise. |
| [String](http://docs.google.com/java/lang/String.html) | [**toGenericString**](http://docs.google.com/java/lang/reflect/Method.html#toGenericString())()            Returns a string describing this Method, including type parameters. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/java/lang/reflect/Method.html#toString())()            Returns a string describing this Method. |

| **Methods inherited from class java.lang.reflect.**[**AccessibleObject**](http://docs.google.com/java/lang/reflect/AccessibleObject.html) |
| --- |
| [getAnnotations](http://docs.google.com/java/lang/reflect/AccessibleObject.html#getAnnotations()), [isAccessible](http://docs.google.com/java/lang/reflect/AccessibleObject.html#isAccessible()), [isAnnotationPresent](http://docs.google.com/java/lang/reflect/AccessibleObject.html#isAnnotationPresent(java.lang.Class)), [setAccessible](http://docs.google.com/java/lang/reflect/AccessibleObject.html#setAccessible(java.lang.reflect.AccessibleObject%5B%5D,%20boolean)), [setAccessible](http://docs.google.com/java/lang/reflect/AccessibleObject.html#setAccessible(boolean)) |

| **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()), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [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)) |

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

### getDeclaringClass

public [Class](http://docs.google.com/java/lang/Class.html)<?> **getDeclaringClass**()

Returns the Class object representing the class or interface that declares the method represented by this Method object.

**Specified by:**[getDeclaringClass](http://docs.google.com/java/lang/reflect/Member.html#getDeclaringClass()) in interface [Member](http://docs.google.com/java/lang/reflect/Member.html) **Returns:**an object representing the declaring class of the underlying member

### getName

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

Returns the name of the method represented by this Method object, as a String.

**Specified by:**[getName](http://docs.google.com/java/lang/reflect/Member.html#getName()) in interface [Member](http://docs.google.com/java/lang/reflect/Member.html) **Returns:**the simple name of the underlying member

### getModifiers

public int **getModifiers**()

Returns the Java language modifiers for the method represented by this Method object, as an integer. The Modifier class should be used to decode the modifiers.

**Specified by:**[getModifiers](http://docs.google.com/java/lang/reflect/Member.html#getModifiers()) in interface [Member](http://docs.google.com/java/lang/reflect/Member.html) **Returns:**the Java language modifiers for the underlying member**See Also:**[Modifier](http://docs.google.com/java/lang/reflect/Modifier.html)

### getTypeParameters

public [TypeVariable](http://docs.google.com/java/lang/reflect/TypeVariable.html)<[Method](http://docs.google.com/java/lang/reflect/Method.html)>[] **getTypeParameters**()

Returns an array of TypeVariable objects that represent the type variables declared by the generic declaration represented by this GenericDeclaration object, in declaration order. Returns an array of length 0 if the underlying generic declaration declares no type variables.

**Specified by:**[getTypeParameters](http://docs.google.com/java/lang/reflect/GenericDeclaration.html#getTypeParameters()) in interface [GenericDeclaration](http://docs.google.com/java/lang/reflect/GenericDeclaration.html) **Returns:**an array of TypeVariable objects that represent the type variables declared by this generic declaration **Throws:** [GenericSignatureFormatError](http://docs.google.com/java/lang/reflect/GenericSignatureFormatError.html) - if the generic signature of this generic declaration does not conform to the format specified in the Java Virtual Machine Specification, 3rd edition**Since:** 1.5

### getReturnType

public [Class](http://docs.google.com/java/lang/Class.html)<?> **getReturnType**()

Returns a Class object that represents the formal return type of the method represented by this Method object.

**Returns:**the return type for the method this object represents

### getGenericReturnType

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

Returns a Type object that represents the formal return type of the method represented by this Method object.

If the return type is a parameterized type, the Type object returned must accurately reflect the actual type parameters used in the source code.

If the return type is a type variable or a parameterized type, it is created. Otherwise, it is resolved.

**Returns:**a Type object that represents the formal return type of the underlying method **Throws:** [GenericSignatureFormatError](http://docs.google.com/java/lang/reflect/GenericSignatureFormatError.html) - if the generic method signature does not conform to the format specified in the Java Virtual Machine Specification, 3rd edition [TypeNotPresentException](http://docs.google.com/java/lang/TypeNotPresentException.html) - if the underlying method's return type refers to a non-existent type declaration [MalformedParameterizedTypeException](http://docs.google.com/java/lang/reflect/MalformedParameterizedTypeException.html) - if the underlying method's return typed refers to a parameterized type that cannot be instantiated for any reason**Since:** 1.5

### getParameterTypes

public [Class](http://docs.google.com/java/lang/Class.html)<?>[] **getParameterTypes**()

Returns an array of Class objects that represent the formal parameter types, in declaration order, of the method represented by this Method object. Returns an array of length 0 if the underlying method takes no parameters.

**Returns:**the parameter types for the method this object represents

### getGenericParameterTypes

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

Returns an array of Type objects that represent the formal parameter types, in declaration order, of the method represented by this Method object. Returns an array of length 0 if the underlying method takes no parameters.

If a formal parameter type is a parameterized type, the Type object returned for it must accurately reflect the actual type parameters used in the source code.

If a formal parameter type is a type variable or a parameterized type, it is created. Otherwise, it is resolved.

**Returns:**an array of Types that represent the formal parameter types of the underlying method, in declaration order **Throws:** [GenericSignatureFormatError](http://docs.google.com/java/lang/reflect/GenericSignatureFormatError.html) - if the generic method signature does not conform to the format specified in the Java Virtual Machine Specification, 3rd edition [TypeNotPresentException](http://docs.google.com/java/lang/TypeNotPresentException.html) - if any of the parameter types of the underlying method refers to a non-existent type declaration [MalformedParameterizedTypeException](http://docs.google.com/java/lang/reflect/MalformedParameterizedTypeException.html) - if any of the underlying method's parameter types refer to a parameterized type that cannot be instantiated for any reason**Since:** 1.5

### getExceptionTypes

public [Class](http://docs.google.com/java/lang/Class.html)<?>[] **getExceptionTypes**()

Returns an array of Class objects that represent the types of the exceptions declared to be thrown by the underlying method represented by this Method object. Returns an array of length 0 if the method declares no exceptions in its throws clause.

**Returns:**the exception types declared as being thrown by the method this object represents

### getGenericExceptionTypes

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

Returns an array of Type objects that represent the exceptions declared to be thrown by this Method object. Returns an array of length 0 if the underlying method declares no exceptions in its throws clause.

If an exception type is a parameterized type, the Type object returned for it must accurately reflect the actual type parameters used in the source code.

If an exception type is a type variable or a parameterized type, it is created. Otherwise, it is resolved.

**Returns:**an array of Types that represent the exception types thrown by the underlying method **Throws:** [GenericSignatureFormatError](http://docs.google.com/java/lang/reflect/GenericSignatureFormatError.html) - if the generic method signature does not conform to the format specified in the Java Virtual Machine Specification, 3rd edition [TypeNotPresentException](http://docs.google.com/java/lang/TypeNotPresentException.html) - if the underlying method's throws clause refers to a non-existent type declaration [MalformedParameterizedTypeException](http://docs.google.com/java/lang/reflect/MalformedParameterizedTypeException.html) - if the underlying method's throws clause refers to a parameterized type that cannot be instantiated for any reason**Since:** 1.5

### equals

public boolean **equals**([Object](http://docs.google.com/java/lang/Object.html) obj)

Compares this Method against the specified object. Returns true if the objects are the same. Two Methods are the same if they were declared by the same class and have the same name and formal parameter types and return type.

**Overrides:**[equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)) in class [Object](http://docs.google.com/java/lang/Object.html) **Parameters:**obj - the reference object with which to compare. **Returns:**true if this object is the same as the obj argument; false otherwise.**See Also:**[Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### hashCode

public int **hashCode**()

Returns a hashcode for this Method. The hashcode is computed as the exclusive-or of the hashcodes for the underlying method's declaring class name and the method's name.

**Overrides:**[hashCode](http://docs.google.com/java/lang/Object.html#hashCode()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a hash code value for this object.**See Also:**[Object.equals(java.lang.Object)](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### toString

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

Returns a string describing this Method. The string is formatted as the method access modifiers, if any, followed by the method return type, followed by a space, followed by the class declaring the method, followed by a period, followed by the method name, followed by a parenthesized, comma-separated list of the method's formal parameter types. If the method throws checked exceptions, the parameter list is followed by a space, followed by the word throws followed by a comma-separated list of the thrown exception types. For example:

public boolean java.lang.Object.equals(java.lang.Object)

The access modifiers are placed in canonical order as specified by "The Java Language Specification". This is public, protected or private first, and then other modifiers in the following order: abstract, static, final, synchronized, native.

**Overrides:**[toString](http://docs.google.com/java/lang/Object.html#toString()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a string representation of the object.

### toGenericString

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

Returns a string describing this Method, including type parameters. The string is formatted as the method access modifiers, if any, followed by an angle-bracketed comma-separated list of the method's type parameters, if any, followed by the method's generic return type, followed by a space, followed by the class declaring the method, followed by a period, followed by the method name, followed by a parenthesized, comma-separated list of the method's generic formal parameter types. A space is used to separate access modifiers from one another and from the type parameters or return type. If there are no type parameters, the type parameter list is elided; if the type parameter list is present, a space separates the list from the class name. If the method is declared to throw exceptions, the parameter list is followed by a space, followed by the word throws followed by a comma-separated list of the generic thrown exception types. If there are no type parameters, the type parameter list is elided.

The access modifiers are placed in canonical order as specified by "The Java Language Specification". This is public, protected or private first, and then other modifiers in the following order: abstract, static, final, synchronized native.

**Returns:**a string describing this Method, include type parameters**Since:** 1.5

### invoke

public [Object](http://docs.google.com/java/lang/Object.html) **invoke**([Object](http://docs.google.com/java/lang/Object.html) obj,  
 [Object](http://docs.google.com/java/lang/Object.html)... args)  
 throws [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html),  
 [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html),  
 [InvocationTargetException](http://docs.google.com/java/lang/reflect/InvocationTargetException.html)

Invokes the underlying method represented by this Method object, on the specified object with the specified parameters. Individual parameters are automatically unwrapped to match primitive formal parameters, and both primitive and reference parameters are subject to method invocation conversions as necessary.

If the underlying method is static, then the specified obj argument is ignored. It may be null.

If the number of formal parameters required by the underlying method is 0, the supplied args array may be of length 0 or null.

If the underlying method is an instance method, it is invoked using dynamic method lookup as documented in The Java Language Specification, Second Edition, section 15.12.4.4; in particular, overriding based on the runtime type of the target object will occur.

If the underlying method is static, the class that declared the method is initialized if it has not already been initialized.

If the method completes normally, the value it returns is returned to the caller of invoke; if the value has a primitive type, it is first appropriately wrapped in an object. However, if the value has the type of an array of a primitive type, the elements of the array are *not* wrapped in objects; in other words, an array of primitive type is returned. If the underlying method return type is void, the invocation returns null.

**Parameters:**obj - the object the underlying method is invoked fromargs - the arguments used for the method call **Returns:**the result of dispatching the method represented by this object on obj with parameters args **Throws:** [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html) - if this Method object enforces Java language access control and the underlying method is inaccessible. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the method is an instance method and the specified object argument is not an instance of the class or interface declaring the underlying method (or of a subclass or implementor thereof); if the number of actual and formal parameters differ; if an unwrapping conversion for primitive arguments fails; or if, after possible unwrapping, a parameter value cannot be converted to the corresponding formal parameter type by a method invocation conversion. [InvocationTargetException](http://docs.google.com/java/lang/reflect/InvocationTargetException.html) - if the underlying method throws an exception. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified object is null and the method is an instance method. [ExceptionInInitializerError](http://docs.google.com/java/lang/ExceptionInInitializerError.html) - if the initialization provoked by this method fails.

### isBridge

public boolean **isBridge**()

Returns true if this method is a bridge method; returns false otherwise.

**Returns:**true if and only if this method is a bridge method as defined by the Java Language Specification.**Since:** 1.5

### isVarArgs

public boolean **isVarArgs**()

Returns true if this method was declared to take a variable number of arguments; returns false otherwise.

**Returns:**true if an only if this method was declared to take a variable number of arguments.**Since:** 1.5

### isSynthetic

public boolean **isSynthetic**()

Returns true if this method is a synthetic method; returns false otherwise.

**Specified by:**[isSynthetic](http://docs.google.com/java/lang/reflect/Member.html#isSynthetic()) in interface [Member](http://docs.google.com/java/lang/reflect/Member.html) **Returns:**true if and only if this method is a synthetic method as defined by the Java Language Specification.**Since:** 1.5

### getAnnotation

public <T extends [Annotation](http://docs.google.com/java/lang/annotation/Annotation.html)> T **getAnnotation**([Class](http://docs.google.com/java/lang/Class.html)<T> annotationClass)

**Description copied from interface:** [**AnnotatedElement**](http://docs.google.com/java/lang/reflect/AnnotatedElement.html#getAnnotation(java.lang.Class)) Returns this element's annotation for the specified type if such an annotation is present, else null.

**Specified by:**[getAnnotation](http://docs.google.com/java/lang/reflect/AnnotatedElement.html#getAnnotation(java.lang.Class)) in interface [AnnotatedElement](http://docs.google.com/java/lang/reflect/AnnotatedElement.html)**Overrides:**[getAnnotation](http://docs.google.com/java/lang/reflect/AccessibleObject.html#getAnnotation(java.lang.Class)) in class [AccessibleObject](http://docs.google.com/java/lang/reflect/AccessibleObject.html) **Parameters:**annotationClass - the Class object corresponding to the annotation type **Returns:**this element's annotation for the specified annotation type if present on this element, else null **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the given annotation class is null**Since:** 1.5

### getDeclaredAnnotations

public [Annotation](http://docs.google.com/java/lang/annotation/Annotation.html)[] **getDeclaredAnnotations**()

**Description copied from interface:** [**AnnotatedElement**](http://docs.google.com/java/lang/reflect/AnnotatedElement.html#getDeclaredAnnotations()) Returns all annotations that are directly present on this element. Unlike the other methods in this interface, this method ignores inherited annotations. (Returns an array of length zero if no annotations are directly present on this element.) The caller of this method is free to modify the returned array; it will have no effect on the arrays returned to other callers.

**Specified by:**[getDeclaredAnnotations](http://docs.google.com/java/lang/reflect/AnnotatedElement.html#getDeclaredAnnotations()) in interface [AnnotatedElement](http://docs.google.com/java/lang/reflect/AnnotatedElement.html)**Overrides:**[getDeclaredAnnotations](http://docs.google.com/java/lang/reflect/AccessibleObject.html#getDeclaredAnnotations()) in class [AccessibleObject](http://docs.google.com/java/lang/reflect/AccessibleObject.html) **Returns:**All annotations directly present on this element**Since:** 1.5

### getDefaultValue

public [Object](http://docs.google.com/java/lang/Object.html) **getDefaultValue**()

Returns the default value for the annotation member represented by this Method instance. If the member is of a primitive type, an instance of the corresponding wrapper type is returned. Returns null if no default is associated with the member, or if the method instance does not represent a declared member of an annotation type.

**Returns:**the default value for the annotation member represented by this Method instance. **Throws:** [TypeNotPresentException](http://docs.google.com/java/lang/TypeNotPresentException.html) - if the annotation is of type [Class](http://docs.google.com/java/lang/Class.html) and no definition can be found for the default class value.**Since:** 1.5

### getParameterAnnotations

public [Annotation](http://docs.google.com/java/lang/annotation/Annotation.html)[][] **getParameterAnnotations**()

Returns an array of arrays that represent the annotations on the formal parameters, in declaration order, of the method represented by this Method object. (Returns an array of length zero if the underlying method is parameterless. If the method has one or more parameters, a nested array of length zero is returned for each parameter with no annotations.) The annotation objects contained in the returned arrays are serializable. The caller of this method is free to modify the returned arrays; it will have no effect on the arrays returned to other callers.

**Returns:**an array of arrays that represent the annotations on the formal parameters, in declaration order, of the method represented by this Method object**Since:** 1.5

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Method.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/Member.html)   [**NEXT CLASS**](http://docs.google.com/java/lang/reflect/Modifier.html) | [**FRAMES**](http://docs.google.com/index.html?java/lang/reflect/Method.html)    [**NO FRAMES**](http://docs.google.com/Method.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#tyjcwt) | DETAIL: FIELD | CONSTR | [METHOD](#4d34og8) |

[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).