| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Field.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/Constructor.html)   [**NEXT CLASS**](http://docs.google.com/java/lang/reflect/GenericArrayType.html) | [**FRAMES**](http://docs.google.com/index.html?java/lang/reflect/Field.html)    [**NO FRAMES**](http://docs.google.com/Field.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 Field

[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.Field**

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

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

A Field provides information about, and dynamic access to, a single field of a class or an interface. The reflected field may be a class (static) field or an instance field.

A Field permits widening conversions to occur during a get or set access operation, but 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.getFields()](http://docs.google.com/java/lang/Class.html#getFields()), [Class.getField(String)](http://docs.google.com/java/lang/Class.html#getField(java.lang.String)), [Class.getDeclaredFields()](http://docs.google.com/java/lang/Class.html#getDeclaredFields()), [Class.getDeclaredField(String)](http://docs.google.com/java/lang/Class.html#getDeclaredField(java.lang.String))

| **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/Field.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Compares this Field against the specified object. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**get**](http://docs.google.com/java/lang/reflect/Field.html#get(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Returns the value of the field represented by this Field, on the specified object. |
| | <T extends [Annotation](http://docs.google.com/java/lang/annotation/Annotation.html)>  T | | --- | | [**getAnnotation**](http://docs.google.com/java/lang/reflect/Field.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. |
| boolean | [**getBoolean**](http://docs.google.com/java/lang/reflect/Field.html#getBoolean(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Gets the value of a static or instance boolean field. |
| byte | [**getByte**](http://docs.google.com/java/lang/reflect/Field.html#getByte(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Gets the value of a static or instance byte field. |
| char | [**getChar**](http://docs.google.com/java/lang/reflect/Field.html#getChar(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Gets the value of a static or instance field of type char or of another primitive type convertible to type char via a widening conversion. |
| [Annotation](http://docs.google.com/java/lang/annotation/Annotation.html)[] | [**getDeclaredAnnotations**](http://docs.google.com/java/lang/reflect/Field.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/Field.html#getDeclaringClass())()            Returns the Class object representing the class or interface that declares the field represented by this Field object. |
| double | [**getDouble**](http://docs.google.com/java/lang/reflect/Field.html#getDouble(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Gets the value of a static or instance field of type double or of another primitive type convertible to type double via a widening conversion. |
| float | [**getFloat**](http://docs.google.com/java/lang/reflect/Field.html#getFloat(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Gets the value of a static or instance field of type float or of another primitive type convertible to type float via a widening conversion. |
| [Type](http://docs.google.com/java/lang/reflect/Type.html) | [**getGenericType**](http://docs.google.com/java/lang/reflect/Field.html#getGenericType())()            Returns a Type object that represents the declared type for the field represented by this Field object. |
| int | [**getInt**](http://docs.google.com/java/lang/reflect/Field.html#getInt(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Gets the value of a static or instance field of type int or of another primitive type convertible to type int via a widening conversion. |
| long | [**getLong**](http://docs.google.com/java/lang/reflect/Field.html#getLong(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Gets the value of a static or instance field of type long or of another primitive type convertible to type long via a widening conversion. |
| int | [**getModifiers**](http://docs.google.com/java/lang/reflect/Field.html#getModifiers())()            Returns the Java language modifiers for the field represented by this Field object, as an integer. |
| [String](http://docs.google.com/java/lang/String.html) | [**getName**](http://docs.google.com/java/lang/reflect/Field.html#getName())()            Returns the name of the field represented by this Field object. |
| short | [**getShort**](http://docs.google.com/java/lang/reflect/Field.html#getShort(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Gets the value of a static or instance field of type short or of another primitive type convertible to type short via a widening conversion. |
| [Class](http://docs.google.com/java/lang/Class.html)<?> | [**getType**](http://docs.google.com/java/lang/reflect/Field.html#getType())()            Returns a Class object that identifies the declared type for the field represented by this Field object. |
| int | [**hashCode**](http://docs.google.com/java/lang/reflect/Field.html#hashCode())()            Returns a hashcode for this Field. |
| boolean | [**isEnumConstant**](http://docs.google.com/java/lang/reflect/Field.html#isEnumConstant())()            Returns true if this field represents an element of an enumerated type; returns false otherwise. |
| boolean | [**isSynthetic**](http://docs.google.com/java/lang/reflect/Field.html#isSynthetic())()            Returns true if this field is a synthetic field; returns false otherwise. |
| void | [**set**](http://docs.google.com/java/lang/reflect/Field.html#set(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) value)            Sets the field represented by this Field object on the specified object argument to the specified new value. |
| void | [**setBoolean**](http://docs.google.com/java/lang/reflect/Field.html#setBoolean(java.lang.Object,%20boolean))([Object](http://docs.google.com/java/lang/Object.html) obj, boolean z)            Sets the value of a field as a boolean on the specified object. |
| void | [**setByte**](http://docs.google.com/java/lang/reflect/Field.html#setByte(java.lang.Object,%20byte))([Object](http://docs.google.com/java/lang/Object.html) obj, byte b)            Sets the value of a field as a byte on the specified object. |
| void | [**setChar**](http://docs.google.com/java/lang/reflect/Field.html#setChar(java.lang.Object,%20char))([Object](http://docs.google.com/java/lang/Object.html) obj, char c)            Sets the value of a field as a char on the specified object. |
| void | [**setDouble**](http://docs.google.com/java/lang/reflect/Field.html#setDouble(java.lang.Object,%20double))([Object](http://docs.google.com/java/lang/Object.html) obj, double d)            Sets the value of a field as a double on the specified object. |
| void | [**setFloat**](http://docs.google.com/java/lang/reflect/Field.html#setFloat(java.lang.Object,%20float))([Object](http://docs.google.com/java/lang/Object.html) obj, float f)            Sets the value of a field as a float on the specified object. |
| void | [**setInt**](http://docs.google.com/java/lang/reflect/Field.html#setInt(java.lang.Object,%20int))([Object](http://docs.google.com/java/lang/Object.html) obj, int i)            Sets the value of a field as an int on the specified object. |
| void | [**setLong**](http://docs.google.com/java/lang/reflect/Field.html#setLong(java.lang.Object,%20long))([Object](http://docs.google.com/java/lang/Object.html) obj, long l)            Sets the value of a field as a long on the specified object. |
| void | [**setShort**](http://docs.google.com/java/lang/reflect/Field.html#setShort(java.lang.Object,%20short))([Object](http://docs.google.com/java/lang/Object.html) obj, short s)            Sets the value of a field as a short on the specified object. |
| [String](http://docs.google.com/java/lang/String.html) | [**toGenericString**](http://docs.google.com/java/lang/reflect/Field.html#toGenericString())()            Returns a string describing this Field, including its generic type. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/java/lang/reflect/Field.html#toString())()            Returns a string describing this Field. |

| **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 field represented by this Field 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 field represented by this Field object.

**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 field represented by this Field 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)

### isEnumConstant

public boolean **isEnumConstant**()

Returns true if this field represents an element of an enumerated type; returns false otherwise.

**Returns:**true if and only if this field represents an element of an enumerated type.**Since:** 1.5

### isSynthetic

public boolean **isSynthetic**()

Returns true if this field is a synthetic field; 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 field is a synthetic field as defined by the Java Language Specification.**Since:** 1.5

### getType

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

Returns a Class object that identifies the declared type for the field represented by this Field object.

**Returns:**a Class object identifying the declared type of the field represented by this object

### getGenericType

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

Returns a Type object that represents the declared type for the field represented by this Field object.

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

If the type of the underlying field is a type variable or a parameterized type, it is created. Otherwise, it is resolved.

**Returns:**a Type object that represents the declared type for the field represented by this Field object **Throws:** [GenericSignatureFormatError](http://docs.google.com/java/lang/reflect/GenericSignatureFormatError.html) - if the generic field 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 generic type signature of the underlying field refers to a non-existent type declaration [MalformedParameterizedTypeException](http://docs.google.com/java/lang/reflect/MalformedParameterizedTypeException.html) - if the generic signature of the underlying field 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 Field against the specified object. Returns true if the objects are the same. Two Field objects are the same if they were declared by the same class and have the same name and 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 Field. This is computed as the exclusive-or of the hashcodes for the underlying field's declaring class name and its 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 Field. The format is the access modifiers for the field, if any, followed by the field type, followed by a space, followed by the fully-qualified name of the class declaring the field, followed by a period, followed by the name of the field. For example:

public static final int java.lang.Thread.MIN\_PRIORITY  
 private int java.io.FileDescriptor.fd

The 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: static, final, transient, volatile.

**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 Field, including its generic type. The format is the access modifiers for the field, if any, followed by the generic field type, followed by a space, followed by the fully-qualified name of the class declaring the field, followed by a period, followed by the name of the field.

The 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: static, final, transient, volatile.

**Returns:**a string describing this Field, including its generic type**Since:** 1.5

### get

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

Returns the value of the field represented by this Field, on the specified object. The value is automatically wrapped in an object if it has a primitive type.

The underlying field's value is obtained as follows:

If the underlying field is a static field, the obj argument is ignored; it may be null.

Otherwise, the underlying field is an instance field. If the specified obj argument is null, the method throws a NullPointerException. If the specified object is not an instance of the class or interface declaring the underlying field, the method throws an IllegalArgumentException.

If this Field object enforces Java language access control, and the underlying field is inaccessible, the method throws an IllegalAccessException. If the underlying field is static, the class that declared the field is initialized if it has not already been initialized.

Otherwise, the value is retrieved from the underlying instance or static field. If the field has a primitive type, the value is wrapped in an object before being returned, otherwise it is returned as is.

If the field is hidden in the type of obj, the field's value is obtained according to the preceding rules.

**Parameters:**obj - object from which the represented field's value is to be extracted **Returns:**the value of the represented field in object obj; primitive values are wrapped in an appropriate object before being returned **Throws:** [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html) - if the underlying field is inaccessible. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the specified object is not an instance of the class or interface declaring the underlying field (or a subclass or implementor thereof). [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified object is null and the field is an instance field. [ExceptionInInitializerError](http://docs.google.com/java/lang/ExceptionInInitializerError.html) - if the initialization provoked by this method fails.

### getBoolean

public boolean **getBoolean**([Object](http://docs.google.com/java/lang/Object.html) obj)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html),  
 [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html)

Gets the value of a static or instance boolean field.

**Parameters:**obj - the object to extract the boolean value from **Returns:**the value of the boolean field **Throws:** [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html) - if the underlying field is inaccessible. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the specified object is not an instance of the class or interface declaring the underlying field (or a subclass or implementor thereof), or if the field value cannot be converted to the type boolean by a widening conversion. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified object is null and the field is an instance field. [ExceptionInInitializerError](http://docs.google.com/java/lang/ExceptionInInitializerError.html) - if the initialization provoked by this method fails.**See Also:**[get(java.lang.Object)](http://docs.google.com/java/lang/reflect/Field.html#get(java.lang.Object))

### getByte

public byte **getByte**([Object](http://docs.google.com/java/lang/Object.html) obj)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html),  
 [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html)

Gets the value of a static or instance byte field.

**Parameters:**obj - the object to extract the byte value from **Returns:**the value of the byte field **Throws:** [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html) - if the underlying field is inaccessible. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the specified object is not an instance of the class or interface declaring the underlying field (or a subclass or implementor thereof), or if the field value cannot be converted to the type byte by a widening conversion. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified object is null and the field is an instance field. [ExceptionInInitializerError](http://docs.google.com/java/lang/ExceptionInInitializerError.html) - if the initialization provoked by this method fails.**See Also:**[get(java.lang.Object)](http://docs.google.com/java/lang/reflect/Field.html#get(java.lang.Object))

### getChar

public char **getChar**([Object](http://docs.google.com/java/lang/Object.html) obj)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html),  
 [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html)

Gets the value of a static or instance field of type char or of another primitive type convertible to type char via a widening conversion.

**Parameters:**obj - the object to extract the char value from **Returns:**the value of the field converted to type char **Throws:** [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html) - if the underlying field is inaccessible. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the specified object is not an instance of the class or interface declaring the underlying field (or a subclass or implementor thereof), or if the field value cannot be converted to the type char by a widening conversion. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified object is null and the field is an instance field. [ExceptionInInitializerError](http://docs.google.com/java/lang/ExceptionInInitializerError.html) - if the initialization provoked by this method fails.**See Also:**[get(java.lang.Object)](http://docs.google.com/java/lang/reflect/Field.html#get(java.lang.Object))

### getShort

public short **getShort**([Object](http://docs.google.com/java/lang/Object.html) obj)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html),  
 [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html)

Gets the value of a static or instance field of type short or of another primitive type convertible to type short via a widening conversion.

**Parameters:**obj - the object to extract the short value from **Returns:**the value of the field converted to type short **Throws:** [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html) - if the underlying field is inaccessible. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the specified object is not an instance of the class or interface declaring the underlying field (or a subclass or implementor thereof), or if the field value cannot be converted to the type short by a widening conversion. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified object is null and the field is an instance field. [ExceptionInInitializerError](http://docs.google.com/java/lang/ExceptionInInitializerError.html) - if the initialization provoked by this method fails.**See Also:**[get(java.lang.Object)](http://docs.google.com/java/lang/reflect/Field.html#get(java.lang.Object))

### getInt

public int **getInt**([Object](http://docs.google.com/java/lang/Object.html) obj)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html),  
 [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html)

Gets the value of a static or instance field of type int or of another primitive type convertible to type int via a widening conversion.

**Parameters:**obj - the object to extract the int value from **Returns:**the value of the field converted to type int **Throws:** [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html) - if the underlying field is inaccessible. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the specified object is not an instance of the class or interface declaring the underlying field (or a subclass or implementor thereof), or if the field value cannot be converted to the type int by a widening conversion. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified object is null and the field is an instance field. [ExceptionInInitializerError](http://docs.google.com/java/lang/ExceptionInInitializerError.html) - if the initialization provoked by this method fails.**See Also:**[get(java.lang.Object)](http://docs.google.com/java/lang/reflect/Field.html#get(java.lang.Object))

### getLong

public long **getLong**([Object](http://docs.google.com/java/lang/Object.html) obj)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html),  
 [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html)

Gets the value of a static or instance field of type long or of another primitive type convertible to type long via a widening conversion.

**Parameters:**obj - the object to extract the long value from **Returns:**the value of the field converted to type long **Throws:** [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html) - if the underlying field is inaccessible. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the specified object is not an instance of the class or interface declaring the underlying field (or a subclass or implementor thereof), or if the field value cannot be converted to the type long by a widening conversion. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified object is null and the field is an instance field. [ExceptionInInitializerError](http://docs.google.com/java/lang/ExceptionInInitializerError.html) - if the initialization provoked by this method fails.**See Also:**[get(java.lang.Object)](http://docs.google.com/java/lang/reflect/Field.html#get(java.lang.Object))

### getFloat

public float **getFloat**([Object](http://docs.google.com/java/lang/Object.html) obj)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html),  
 [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html)

Gets the value of a static or instance field of type float or of another primitive type convertible to type float via a widening conversion.

**Parameters:**obj - the object to extract the float value from **Returns:**the value of the field converted to type float **Throws:** [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html) - if the underlying field is inaccessible. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the specified object is not an instance of the class or interface declaring the underlying field (or a subclass or implementor thereof), or if the field value cannot be converted to the type float by a widening conversion. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified object is null and the field is an instance field. [ExceptionInInitializerError](http://docs.google.com/java/lang/ExceptionInInitializerError.html) - if the initialization provoked by this method fails.**See Also:**[get(java.lang.Object)](http://docs.google.com/java/lang/reflect/Field.html#get(java.lang.Object))

### getDouble

public double **getDouble**([Object](http://docs.google.com/java/lang/Object.html) obj)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html),  
 [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html)

Gets the value of a static or instance field of type double or of another primitive type convertible to type double via a widening conversion.

**Parameters:**obj - the object to extract the double value from **Returns:**the value of the field converted to type double **Throws:** [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html) - if the underlying field is inaccessible. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the specified object is not an instance of the class or interface declaring the underlying field (or a subclass or implementor thereof), or if the field value cannot be converted to the type double by a widening conversion. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified object is null and the field is an instance field. [ExceptionInInitializerError](http://docs.google.com/java/lang/ExceptionInInitializerError.html) - if the initialization provoked by this method fails.**See Also:**[get(java.lang.Object)](http://docs.google.com/java/lang/reflect/Field.html#get(java.lang.Object))

### set

public void **set**([Object](http://docs.google.com/java/lang/Object.html) obj,  
 [Object](http://docs.google.com/java/lang/Object.html) value)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html),  
 [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html)

Sets the field represented by this Field object on the specified object argument to the specified new value. The new value is automatically unwrapped if the underlying field has a primitive type.

The operation proceeds as follows:

If the underlying field is static, the obj argument is ignored; it may be null.

Otherwise the underlying field is an instance field. If the specified object argument is null, the method throws a NullPointerException. If the specified object argument is not an instance of the class or interface declaring the underlying field, the method throws an IllegalArgumentException.

If this Field object enforces Java language access control, and the underlying field is inaccessible, the method throws an IllegalAccessException.

If the underlying field is final, the method throws an IllegalAccessException unless setAccessible(true) has succeeded for this field and this field is non-static. Setting a final field in this way is meaningful only during deserialization or reconstruction of instances of classes with blank final fields, before they are made available for access by other parts of a program. Use in any other context may have unpredictable effects, including cases in which other parts of a program continue to use the original value of this field.

If the underlying field is of a primitive type, an unwrapping conversion is attempted to convert the new value to a value of a primitive type. If this attempt fails, the method throws an IllegalArgumentException.

If, after possible unwrapping, the new value cannot be converted to the type of the underlying field by an identity or widening conversion, the method throws an IllegalArgumentException.

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

The field is set to the possibly unwrapped and widened new value.

If the field is hidden in the type of obj, the field's value is set according to the preceding rules.

**Parameters:**obj - the object whose field should be modifiedvalue - the new value for the field of obj being modified **Throws:** [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html) - if the underlying field is inaccessible. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the specified object is not an instance of the class or interface declaring the underlying field (or a subclass or implementor thereof), or if an unwrapping conversion fails. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified object is null and the field is an instance field. [ExceptionInInitializerError](http://docs.google.com/java/lang/ExceptionInInitializerError.html) - if the initialization provoked by this method fails.

### setBoolean

public void **setBoolean**([Object](http://docs.google.com/java/lang/Object.html) obj,  
 boolean z)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html),  
 [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html)

Sets the value of a field as a boolean on the specified object. This method is equivalent to set(obj, zObj), where zObj is a Boolean object and zObj.booleanValue() == z.

**Parameters:**obj - the object whose field should be modifiedz - the new value for the field of obj being modified **Throws:** [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html) - if the underlying field is inaccessible. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the specified object is not an instance of the class or interface declaring the underlying field (or a subclass or implementor thereof), or if an unwrapping conversion fails. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified object is null and the field is an instance field. [ExceptionInInitializerError](http://docs.google.com/java/lang/ExceptionInInitializerError.html) - if the initialization provoked by this method fails.**See Also:**[set(java.lang.Object, java.lang.Object)](http://docs.google.com/java/lang/reflect/Field.html#set(java.lang.Object,%20java.lang.Object))

### setByte

public void **setByte**([Object](http://docs.google.com/java/lang/Object.html) obj,  
 byte b)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html),  
 [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html)

Sets the value of a field as a byte on the specified object. This method is equivalent to set(obj, bObj), where bObj is a Byte object and bObj.byteValue() == b.

**Parameters:**obj - the object whose field should be modifiedb - the new value for the field of obj being modified **Throws:** [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html) - if the underlying field is inaccessible. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the specified object is not an instance of the class or interface declaring the underlying field (or a subclass or implementor thereof), or if an unwrapping conversion fails. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified object is null and the field is an instance field. [ExceptionInInitializerError](http://docs.google.com/java/lang/ExceptionInInitializerError.html) - if the initialization provoked by this method fails.**See Also:**[set(java.lang.Object, java.lang.Object)](http://docs.google.com/java/lang/reflect/Field.html#set(java.lang.Object,%20java.lang.Object))

### setChar

public void **setChar**([Object](http://docs.google.com/java/lang/Object.html) obj,  
 char c)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html),  
 [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html)

Sets the value of a field as a char on the specified object. This method is equivalent to set(obj, cObj), where cObj is a Character object and cObj.charValue() == c.

**Parameters:**obj - the object whose field should be modifiedc - the new value for the field of obj being modified **Throws:** [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html) - if the underlying field is inaccessible. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the specified object is not an instance of the class or interface declaring the underlying field (or a subclass or implementor thereof), or if an unwrapping conversion fails. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified object is null and the field is an instance field. [ExceptionInInitializerError](http://docs.google.com/java/lang/ExceptionInInitializerError.html) - if the initialization provoked by this method fails.**See Also:**[set(java.lang.Object, java.lang.Object)](http://docs.google.com/java/lang/reflect/Field.html#set(java.lang.Object,%20java.lang.Object))

### setShort

public void **setShort**([Object](http://docs.google.com/java/lang/Object.html) obj,  
 short s)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html),  
 [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html)

Sets the value of a field as a short on the specified object. This method is equivalent to set(obj, sObj), where sObj is a Short object and sObj.shortValue() == s.

**Parameters:**obj - the object whose field should be modifieds - the new value for the field of obj being modified **Throws:** [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html) - if the underlying field is inaccessible. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the specified object is not an instance of the class or interface declaring the underlying field (or a subclass or implementor thereof), or if an unwrapping conversion fails. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified object is null and the field is an instance field. [ExceptionInInitializerError](http://docs.google.com/java/lang/ExceptionInInitializerError.html) - if the initialization provoked by this method fails.**See Also:**[set(java.lang.Object, java.lang.Object)](http://docs.google.com/java/lang/reflect/Field.html#set(java.lang.Object,%20java.lang.Object))

### setInt

public void **setInt**([Object](http://docs.google.com/java/lang/Object.html) obj,  
 int i)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html),  
 [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html)

Sets the value of a field as an int on the specified object. This method is equivalent to set(obj, iObj), where iObj is a Integer object and iObj.intValue() == i.

**Parameters:**obj - the object whose field should be modifiedi - the new value for the field of obj being modified **Throws:** [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html) - if the underlying field is inaccessible. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the specified object is not an instance of the class or interface declaring the underlying field (or a subclass or implementor thereof), or if an unwrapping conversion fails. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified object is null and the field is an instance field. [ExceptionInInitializerError](http://docs.google.com/java/lang/ExceptionInInitializerError.html) - if the initialization provoked by this method fails.**See Also:**[set(java.lang.Object, java.lang.Object)](http://docs.google.com/java/lang/reflect/Field.html#set(java.lang.Object,%20java.lang.Object))

### setLong

public void **setLong**([Object](http://docs.google.com/java/lang/Object.html) obj,  
 long l)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html),  
 [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html)

Sets the value of a field as a long on the specified object. This method is equivalent to set(obj, lObj), where lObj is a Long object and lObj.longValue() == l.

**Parameters:**obj - the object whose field should be modifiedl - the new value for the field of obj being modified **Throws:** [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html) - if the underlying field is inaccessible. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the specified object is not an instance of the class or interface declaring the underlying field (or a subclass or implementor thereof), or if an unwrapping conversion fails. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified object is null and the field is an instance field. [ExceptionInInitializerError](http://docs.google.com/java/lang/ExceptionInInitializerError.html) - if the initialization provoked by this method fails.**See Also:**[set(java.lang.Object, java.lang.Object)](http://docs.google.com/java/lang/reflect/Field.html#set(java.lang.Object,%20java.lang.Object))

### setFloat

public void **setFloat**([Object](http://docs.google.com/java/lang/Object.html) obj,  
 float f)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html),  
 [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html)

Sets the value of a field as a float on the specified object. This method is equivalent to set(obj, fObj), where fObj is a Float object and fObj.floatValue() == f.

**Parameters:**obj - the object whose field should be modifiedf - the new value for the field of obj being modified **Throws:** [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html) - if the underlying field is inaccessible. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the specified object is not an instance of the class or interface declaring the underlying field (or a subclass or implementor thereof), or if an unwrapping conversion fails. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified object is null and the field is an instance field. [ExceptionInInitializerError](http://docs.google.com/java/lang/ExceptionInInitializerError.html) - if the initialization provoked by this method fails.**See Also:**[set(java.lang.Object, java.lang.Object)](http://docs.google.com/java/lang/reflect/Field.html#set(java.lang.Object,%20java.lang.Object))

### setDouble

public void **setDouble**([Object](http://docs.google.com/java/lang/Object.html) obj,  
 double d)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html),  
 [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html)

Sets the value of a field as a double on the specified object. This method is equivalent to set(obj, dObj), where dObj is a Double object and dObj.doubleValue() == d.

**Parameters:**obj - the object whose field should be modifiedd - the new value for the field of obj being modified **Throws:** [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html) - if the underlying field is inaccessible. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the specified object is not an instance of the class or interface declaring the underlying field (or a subclass or implementor thereof), or if an unwrapping conversion fails. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified object is null and the field is an instance field. [ExceptionInInitializerError](http://docs.google.com/java/lang/ExceptionInInitializerError.html) - if the initialization provoked by this method fails.**See Also:**[set(java.lang.Object, java.lang.Object)](http://docs.google.com/java/lang/reflect/Field.html#set(java.lang.Object,%20java.lang.Object))

### 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

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Field.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/Constructor.html)   [**NEXT CLASS**](http://docs.google.com/java/lang/reflect/GenericArrayType.html) | [**FRAMES**](http://docs.google.com/index.html?java/lang/reflect/Field.html)    [**NO FRAMES**](http://docs.google.com/Field.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).