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

## **java.lang.reflect**

Interface ParameterizedType

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

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

ParameterizedType represents a parameterized type such as Collection<String>.

A parameterized type is created the first time it is needed by a reflective method, as specified in this package. When a parameterized type p is created, the generic type declaration that p instantiates is resolved, and all type arguments of p are created recursively. See [TypeVariable](http://docs.google.com/java/lang/reflect/TypeVariable.html) for details on the creation process for type variables. Repeated creation of a parameterized type has no effect.

Instances of classes that implement this interface must implement an equals() method that equates any two instances that share the same generic type declaration and have equal type parameters.

**Since:** 1.5

| **Method Summary** | |
| --- | --- |
| [Type](http://docs.google.com/java/lang/reflect/Type.html)[] | [**getActualTypeArguments**](http://docs.google.com/java/lang/reflect/ParameterizedType.html#getActualTypeArguments())()            Returns an array of Type objects representing the actual type arguments to this type. |
| [Type](http://docs.google.com/java/lang/reflect/Type.html) | [**getOwnerType**](http://docs.google.com/java/lang/reflect/ParameterizedType.html#getOwnerType())()            Returns a Type object representing the type that this type is a member of. |
| [Type](http://docs.google.com/java/lang/reflect/Type.html) | [**getRawType**](http://docs.google.com/java/lang/reflect/ParameterizedType.html#getRawType())()            Returns the Type object representing the class or interface that declared this type. |

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

### getActualTypeArguments

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

Returns an array of Type objects representing the actual type arguments to this type.

Note that in some cases, the returned array be empty. This can occur if this type represents a non-parameterized type nested within a parameterized type.

**Returns:**an array of Type objects representing the actual type arguments to this type **Throws:** TypeNotPresentException - if any of the actual type arguments refers to a non-existent type declaration MalformedParameterizedTypeException - if any of the actual type parameters refer to a parameterized type that cannot be instantiated for any reason**Since:** 1.5

### getRawType

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

Returns the Type object representing the class or interface that declared this type.

**Returns:**the Type object representing the class or interface that declared this type**Since:** 1.5

### getOwnerType

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

Returns a Type object representing the type that this type is a member of. For example, if this type is O<T>.I<S>, return a representation of O<T>.

If this type is a top-level type, null is returned.

**Returns:**a Type object representing the type that this type is a member of. If this type is a top-level type, null is returned **Throws:** TypeNotPresentException - if the owner type refers to a non-existent type declaration MalformedParameterizedTypeException - if the owner type refers to a parameterized type that cannot be instantiated for any reason**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/ParameterizedType.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/Modifier.html)   [**NEXT CLASS**](http://docs.google.com/java/lang/reflect/Proxy.html) | [**FRAMES**](http://docs.google.com/index.html?java/lang/reflect/ParameterizedType.html)    [**NO FRAMES**](http://docs.google.com/ParameterizedType.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).