**Module** java.base
**Package** java.lang.reflect

# Interface TypeVariable<D extends GenericDeclaration>

**Type Parameters:**

D - the type of generic declaration that declared the underlying type variable.

**All Superinterfaces:**

AnnotatedElement, Type

---

```
public interface TypeVariable<D extends GenericDeclaration>
extends Type, AnnotatedElement
```

TypeVariable is the common superinterface for type variables of kinds. A type variable is created the first time it is needed by a reflective method, as specified in this package. If a type variable t is referenced by a type (i.e, class, interface or annotation type) T, and T is declared by the nth enclosing class of T (see JLS 8.1.2), then the creation of t requires the resolution (see JVMS 5) of the ith enclosing class of T, for i = 0 to n, inclusive. Creating a type variable must not cause the creation of its bounds. Repeated creation of a type variable has no effect.

Multiple objects may be instantiated at run-time to represent a given type variable. Even though a type variable is created only once, this does not imply any requirement to cache instances representing the type variable. However, all instances representing a type variable must be equal() to each other. As a consequence, users of type variables must not rely on the identity of instances of classes implementing this interface.

**Since:**

1.5

## *Method Summary*

All Methods | Instance Methods | Abstract Methods

| Modifier and Type | Method | Description |
|---|---|---|
| AnnotatedType[] | getAnnotatedBounds() | Returns an array of AnnotatedType objects that represent the use of types to denote the upper bounds of the type parameter represented by this TypeVariable. |

| | | |
|---|---|---|
| | getGenericDeclaration() | Returns the GenericDeclaration object representing the generic declaration declared this type variable. |
| String | getName() | Returns the name of this type variable, as it occurs in the source code. |

### Methods declared in interface java.lang.reflect.AnnotatedElement

getAnnotation, getAnnotations, getAnnotationsByType, getDeclaredAnnotation, getDeclaredAnnotations, getDeclaredAnnotationsByType, isAnnotationPresent

### Methods declared in interface java.lang.reflect.Type

getTypeName

## *Method Detail*

### getBounds

```
Type[] getBounds()
```

Returns an array of Type objects representing the upper bound(s) of this type variable. If no upper bound is explicitly declared, the upper bound is Object.

For each upper bound B:

- if B is a parameterized type or a type variable, it is created, (see ParameterizedType for the details of the creation process for parameterized types).
- Otherwise, B is resolved.

**Returns:**

an array of Types representing the upper bound(s) of this type variable

**Throws:**

TypeNotPresentException - if any of the bounds refers to a non-existent type declaration

MalformedParameterizedTypeException - if any of the bounds refer to a parameterized type that cannot be instantiated for any reason

Returns the `GenericDeclaration` object representing the generic declaration declared this type variable.

**Returns:**

the generic declaration declared for this type variable.

**Since:**

1.5

### getName

`String getName()`

Returns the name of this type variable, as it occurs in the source code.

**Returns:**

the name of this type variable, as it appears in the source code

### getAnnotatedBounds

`AnnotatedType[] getAnnotatedBounds()`

Returns an array of AnnotatedType objects that represent the use of types to denote the upper bounds of the type parameter represented by this TypeVariable. The order of the objects in the array corresponds to the order of the bounds in the declaration of the type parameter. Note that if no upper bound is explicitly declared, the upper bound is unannotated `Object`.

**Returns:**

an array of objects representing the upper bound(s) of the type variable

**Since:**

1.8