JAVA.LANG.CLASS CLASS

http://www.tutorialspoint.com/java/lang/java_lang_class.htm

Introduction

The **java.lang.Class** class instance represent classes and interfaces in a running Java application.It has no public constructor.

Class declaration

Following is the declaration for **java.lang.Class** class:

```
public final class Class<T>
  extends Object
    implements Serializable, GenericDeclaration, Type, AnnotatedElement
```

Class methods

S.N.	Method & Description
1	<U> Class<? extends U> asSubclass*Class < U > clazz* This method casts this Class object to represent a subclass of the class represented by the specified class object.
2	T cast*Objectobj* This method casts an object to the class or interface represented by this Class object.
3	boolean desiredAssertionStatus This method returns the assertion status that would be assigned to this class if it were to be initialized at the time this method is invoked.
4	static Class<?> forName*StringclassName* This method returns the Class object associated with the class or interface with the given string name.
5	static Class<?> forName*Stringname, booleaninitialize, ClassLoaderloader* This method returns the Class object associated with the class or interface with the given string name, using the given class loader.
6	<A extends Annotation> A getAnnotation*Class < A > annotationClass* This method returns this element's annotation for the specified type if such an annotation is present, else null.
7	Annotation[] getAnnotations

This method returns all annotations present on this element.

8

String getCanonicalName

This method returns the canonical name of the underlying class as defined by the Java Language Specification.

9

Class<?>[] getClasses

This method returns an array containing Class objects representing all the public classes and interfaces that are members of the class represented by this Class object.

10

ClassLoader getClassLoader

This method returns the class loader for the class.

11

Class<?> getComponentType

This method returns the Class representing the component type of an array.

12

Constructor<T> getConstructor*Class < ? > . . . parameterTypes*

This method returns a Constructor object that reflects the specified public constructor of the class represented by this Class object.

13

Constructor<?>[] getConstructors

This method returns an array containing Constructor objects reflecting all the public constructors of the class represented by this Class object.

14

Annotation[] getDeclaredAnnotations

This method returns all annotations that are directly present on this element.

15

Class<?>[] getDeclaredClasses

This method returns an array of Class objects reflecting all the classes and interfaces declared as members of the class represented by this Class object.

16

Constructor<T> getDeclaredConstructor*Class < ? > . . . parameterTypes*

This method returns a Constructor object that reflects the specified constructor of the class or interface represented by this Class object.

17

Constructor<?>[] getDeclaredConstructors

This method returns an array of Constructor objects reflecting all the constructors declared by the class represented by this Class object.

18	Field getDeclaredField*Stringname* This method returns a Field object that reflects the specified declared field of the class or interface represented by this Class object.
19	Field[] getDeclaredFields This method returns an array of Field objects reflecting all the fields declared by the class or interface represented by this Class object.
20	Method getDeclaredMethod*Stringname, Class < ? > . . . parameterTypes* This method returns a Method object that reflects the specified declared method of the class or interface represented by this Class object.
21	Method[] getDeclaredMethods This method returns an array of Method objects reflecting all the methods declared by the class or interface represented by this Class object.
22	Class<?> getDeclaringClass If the class or interface represented by this Class object is a member of another class, returns the Class object representing the class in which it was declared.
23	Class<?> getEnclosingClass This method returns the immediately enclosing class of the underlying class.
24	Constructor<?> getEnclosingConstructor If this Class object represents a local or anonymous class within a constructor, returns a Constructor object representing the immediately enclosing constructor of the underlying class.
25	Method getEnclosingMethod If this Class object represents a local or anonymous class within a method, returns a Method object representing the immediately enclosing method of the underlying class.
26	T[] getEnumConstants This method returns the elements of this enum class or null if this Class object does not represent an enum type.
27	Field getField*Stringname* This method returns a Field object that reflects the specified public member field of the class or interface represented by this Class object.

28

Field[] getFields

This method returns an array containing Field objects reflecting all the accessible public fields of the class or interface represented by this Class object.

29

Type[] getGenericInterfaces

This method returns the Types representing the interfaces directly implemented by the class or interface represented by this object.

30

Type getGenericSuperclass

This method returns the Type representing the direct superclass of the entity *class, interface, primitivetypeorvoid* represented by this Class.

31

Class<?>[] getInterfaces

This method determines the interfaces implemented by the class or interface represented by this object.

32

Method getMethod*Stringname, Class < ? > . . . parameterTypes*

This method returns a Method object that reflects the specified public member method of the class or interface represented by this Class object.

33

Method[] getMethods

This method returns an array containing Method objects reflecting all the public member methods of the class or interface represented by this Class object, including those declared by the class or interface and those inherited from superclasses and superinterfaces.

34

int getModifiers

This method returns the Java language modifiers for this class or interface, encoded in an integer.

35

String getName

This method returns the name of the entity *class, interface, arrayclass, primitivetype, orvoid* represented by this Class object, as a String.

36

Package getPackage

This method gets the package for this class.

37

ProtectionDomain getProtectionDomain

This method returns the ProtectionDomain of this class.

38

URL getResource*Stringname*

This method finds a resource with a given name.

39

InputStream getResourceAsStream*Stringname*

This method finds a resource with a given name.

40

Object[] getSigners

This method gets the signers of this class.

41

String getSimpleName

This method returns the simple name of the underlying class as given in the source code.

42

Class<? super T> getSuperclass

This method returns the Class representing the superclass of the entity *class, interface, primitivetypeorvoid* represented by this Class.

43

TypeVariable<Class<T>>[]getTypeParameters

This method returns an array of TypeVariable objects that represent the type variables declared by the generic declaration represented by this GenericDeclaration object, in declaration order.

44

boolean isAnnotation

This method returns true if this Class object represents an annotation type.

45

boolean isAnnotationPresent*Class < ?extendsAnnotation > annotationClass*

This method returns true if an annotation for the specified type is present on this element, else false.

46

boolean isAnonymousClass

This method returns true if and only if the underlying class is an anonymous class.

47

boolean isArray

This method determines if this Class object represents an array class.

48

boolean isAssignableFrom*Class < ? > cls*

This method determines if the class or interface represented by this Class object is either the same as, or is a superclass or superinterface of, the class or interface represented by the specified Class parameter.

49

boolean isEnum

This method returns true if and only if this class was declared as an enum in the source code.

50

boolean isInstance$Objectobj$

This method determines if the specified Object is assignment-compatible with the object represented by this Class.

51

boolean isInterface

This method Determines if the specified Class object represents an interface type.

52

boolean isLocalClass

This method returns true if and only if the underlying class is a local class.

53

boolean isMemberClass

This method returns true if and only if the underlying class is a member class.

54

boolean isPrimitive

This method determines if the specified Class object represents a primitive type.

55

boolean isSynthetic

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

56

T newInstance

This method creates a new instance of the class represented by this Class object.

57

String toString

This method converts the object to a string.

Methods inherited

This class inherits methods from the following classes:

- java.lang.Object

Loading [MathJax]/jax/output/HTML-CSS/jax.js