

JAVA.LANG.CLASS.GETDECLAREDCONSTRUCTOR METHOD

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

Copyright © tutorialspoint.com

Description

The **java.lang.Class.getDeclaredConstructor** method returns a Constructor object that reflects the specified constructor of the class or interface represented by this Class object. The **parameterTypes** parameter is an array of Class objects that identify the constructor's formal parameter types, in declared order.

Declaration

Following is the declaration for **java.lang.Class.getDeclaredConstructor** method

```
public Constructor<T> getDeclaredConstructor(Class<?>... parameterTypes) throws  
NoSuchMethodException, SecurityException
```

Parameters

- **parameterTypes** -- This is the parameter array.

Return Value

This method returns the Constructor object for the constructor with the specified parameter list.

Exception

- **NoSuchMethodException** -- If a matching method is not found.
- **SecurityException** -- If a security manager, s, is present.

Example

The following example shows the usage of java.lang.Class.getDeclaredConstructor method.

```
package com.tutorialspoint;  
  
import java.lang.reflect.*;  
  
public class ClassDemo {  
  
    public static void main(String[] args) {  
  
        try {  
            ClassDemo cls = new ClassDemo();  
            Class c = cls.getClass();  
  
            // constructor with arguments as Double and Long  
            Class[] cArg = new Class[2];  
            cArg[0] = Double.class;  
            cArg[1] = Long.class;  
            Constructor ct = c.getDeclaredConstructor(cArg);  
            System.out.println("Constructor = " + ct.toString());  
        }  
  
        catch (NoSuchMethodException e) {  
            System.out.println(e.toString());  
        }  
  
        catch (SecurityException e) {  
            System.out.println(e.toString());  
        }  
    }  
}
```

```
private ClassDemo() {  
    System.out.println("no argument constructor");  
}  
  
public ClassDemo(Double d, Long l) {  
    this.d = d;  
    this.l = l;  
}  
  
Double d = new Double(3.9d);  
Long l = new Long(7687);  
}
```

Let us compile and run the above program, this will produce the following result:

```
no argument constructor  
Constructor = public ClassDemo(java.lang.Double, java.lang.Long)  
Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js
```