

JAVA.LANG.CLASS.GETDECLAREDCLASSES METHOD

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

Copyright © tutorialspoint.com

Description

The **java.lang.Class.getDeclaredClasses** method returns an array of Class objects including public, protected, default *package* access, and private classes and interfaces declared by the class, but excludes inherited classes and interfaces.

This method returns an array of length 0 if the class declares no classes or interfaces as members, or if this Class object represents a primitive type, an array class, or void.

Declaration

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

```
public Class<?>[] getDeclaredClasses() throws SecurityException
```

Parameters

- NA

Return Value

This method returns the array of Class objects representing all the declared members of this class.

Exception

- **SecurityException** -- If a security manager, s, is present.

Example

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

```
package com.tutorialspoint;

import java.lang.*;

public class ClassDemo {

    public static void main(String[] args) {

        try {
            // returns the Class object associated with this class
            Class cls = Class.forName("ClassDemo");

            /* returns the array of Class objects representing all the declared
            members of this class */
            Class[] classes = cls.getDeclaredClasses();
            for (int i = 0; i < classes.length; i++) {
                System.out.println("Class = " + classes[i].getName());
            }
        }
        catch (ClassNotFoundException e) {
            System.out.println(e.toString());
        }
    }

    public class InnerClass1 {
        public InnerClass1() {
            System.out.println("Inner Class1");
        }
    }
}
```

```
public class InnerClass2 {  
    public InnerClass2() {  
        System.out.println("Inner Class2");  
    }  
}  
  
private class InnerPrivateClass {  
    public InnerPrivateClass() {  
        System.out.println("Inner Private Class");  
    }  
}
```

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

```
Class = ClassDemo$InnerPrivateClass  
Class = ClassDemo$InnerClass2  
Class = ClassDemo$InnerClass1
```

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