

JAVA.LANG.RUNTIME.EXIT METHOD

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

Copyright © tutorialspoint.com

Description

The **java.lang.Runtime.exit***int***status** method terminates the currently running Java virtual machine by initiating its shutdown sequence. This method never returns normally. The argument serves as a status code; by convention, a nonzero status code indicates abnormal termination.

The virtual machine's shutdown sequence consists of two phases. In the first phase all registered shutdown hooks, if any, are started in some unspecified order and allowed to run concurrently until they finish. In the second phase all uninvoked finalizers are run if finalization-on-exit has been enabled. Once this is done the virtual machine halts.

If this method is invoked after the virtual machine has begun its shutdown sequence then if shutdown hooks are being run this method will block indefinitely. If shutdown hooks have already been run and on-exit finalization has been enabled then this method halts the virtual machine with the given status code if the status is nonzero; otherwise, it blocks indefinitely. The **System.exit** method is the conventional and convenient means of invoking this method.

Declaration

Following is the declaration for **java.lang.Runtime.exit** method

```
public void exit(int status)
```

Parameters

- **status** -- Termination status. By convention, a nonzero status code indicates abnormal termination.

Return Value

This method does not return a value.

Exception

- **SecurityException** -- If a security manager is present and its **checkExit** method does not permit exiting with the specified status

Example

The following example shows the usage of **lang.Runtime.exit** method.

```
package com.tutorialspoint;

public class RuntimeDemo {

    public static void main(String[] args) {

        // print when the program starts
        System.out.println("Program starting...");

        // cause the program to exit
        Runtime.getRuntime().exit(0);

        // try to print something
        System.out.println("Program still running...");

    }
}
```

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

Program starting...

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