Day 47

Priority of a Thread (Thread Priority)

Each thread has a priority. Priorities are represented by a number between 1 and 10. In most cases, the thread scheduler schedules the threads according to their priority (known as preemptive scheduling). But it is not guaranteed because it depends on JVM specification that which scheduling it chooses. Note that not only JVM a Java programmer can also assign the priorities of a thread explicitly in a Java program.

Setter & Getter Method of Thread Priority

Let's discuss the setter and getter method of the thread priority.

**public final int getPriority():** The java.lang.Thread.getPriority() method returns the priority of the given thread.

**public final void setPriority(int newPriority):** The java.lang.Thread.setPriority() method updates or assign the priority of the thread to newPriority. The method throws IllegalArgumentException if the value newPriority goes out of the range, which is 1 (minimum) to 10 (maximum).

3 constants defined in Thread class:

1. public static int MIN\_PRIORITY
2. public static int NORM\_PRIORITY
3. public static int MAX\_PRIORITY

Default priority of a thread is 5 (NORM\_PRIORITY). The value of MIN\_PRIORITY is 1 and the value of MAX\_PRIORITY is 10.

Example of priority of a Thread:

**FileName:** ThreadPriorityExample.java

1. // Importing the required classes
2. **import** java.lang.\*;
4. **public** **class** ThreadPriorityExample **extends** Thread
5. {
7. // Method 1
8. // Whenever the start() method is called by a thread
9. // the run() method is invoked
10. **public** **void** run()
11. {
12. // the print statement
13. System.out.println("Inside the run() method");
14. }
16. // the main method
17. **public** **static** **void** main(String argvs[])
18. {
19. // Creating threads with the help of ThreadPriorityExample class
20. ThreadPriorityExample th1 = **new** ThreadPriorityExample();
21. ThreadPriorityExample th2 = **new** ThreadPriorityExample();
22. ThreadPriorityExample th3 = **new** ThreadPriorityExample();
24. // We did not mention the priority of the thread.
25. // Therefore, the priorities of the thread is 5, the default value
27. // 1st Thread
28. // Displaying the priority of the thread
29. // using the getPriority() method
30. System.out.println("Priority of the thread th1 is : " + th1.getPriority());
32. // 2nd Thread
33. // Display the priority of the thread
34. System.out.println("Priority of the thread th2 is : " + th2.getPriority());
36. // 3rd Thread
37. // // Display the priority of the thread
38. System.out.println("Priority of the thread th2 is : " + th2.getPriority());
40. // Setting priorities of above threads by
41. // passing integer arguments
42. th1.setPriority(6);
43. th2.setPriority(3);
44. th3.setPriority(9);
46. // 6
47. System.out.println("Priority of the thread th1 is : " + th1.getPriority());
49. // 3
50. System.out.println("Priority of the thread th2 is : " + th2.getPriority());
52. // 9
53. System.out.println("Priority of the thread th3 is : " + th3.getPriority());
55. // Main thread
57. // Displaying name of the currently executing thread
58. System.out.println("Currently Executing The Thread : " + Thread.currentThread().getName());
60. System.out.println("Priority of the main thread is : " + Thread.currentThread().getPriority());
62. // Priority of the main thread is 10 now
63. Thread.currentThread().setPriority(10);
65. System.out.println("Priority of the main thread is : " + Thread.currentThread().getPriority());
66. }
67. }

**Output:**

Priority of the thread th1 is : 5

Priority of the thread th2 is : 5

Priority of the thread th2 is : 5

Priority of the thread th1 is : 6

Priority of the thread th2 is : 3

Priority of the thread th3 is : 9

Currently Executing The Thread : main

Priority of the main thread is : 5

Priority of the main thread is : 10

We know that a thread with high priority will get preference over lower priority threads when it comes to the execution of threads. However, there can be other scenarios where two threads can have the same priority. All of the processing, in order to look after the threads, is done by the Java thread scheduler. Refer to the following example to comprehend what will happen if two threads have the same priority.

**FileName:** ThreadPriorityExample1.java

1. // importing the java.lang package
2. **import** java.lang.\*;
4. **public** **class** ThreadPriorityExample1 **extends** Thread
5. {
7. // Method 1
8. // Whenever the start() method is called by a thread
9. // the run() method is invoked
10. **public** **void** run()
11. {
12. // the print statement
13. System.out.println("Inside the run() method");
14. }

17. // the main method
18. **public** **static** **void** main(String argvs[])
19. {
21. // Now, priority of the main thread is set to 7
22. Thread.currentThread().setPriority(7);
24. // the current thread is retrieved
25. // using the currentThread() method
27. // displaying the main thread priority
28. // using the getPriority() method of the Thread class
29. System.out.println("Priority of the main thread is : " + Thread.currentThread().getPriority());
31. // creating a thread by creating an object of the class ThreadPriorityExample1
32. ThreadPriorityExample1 th1 = **new** ThreadPriorityExample1();
34. // th1 thread is the child of the main thread
35. // therefore, the th1 thread also gets the priority 7
37. // Displaying the priority of the current thread
38. System.out.println("Priority of the thread th1 is : " + th1.getPriority());
39. }
40. }

**Output:**

Priority of the main thread is : 7

Priority of the thread th1 is : 7

**Explanation:** If there are two threads that have the same priority, then one can not predict which thread will get the chance to execute first. The execution then is dependent on the thread scheduler's algorithm (First Come First Serve, Round-Robin, etc.)

Example of IllegalArgumentException

We know that if the value of the parameter *newPriority* of the method getPriority() goes out of the range (1 to 10), then we get the IllegalArgumentException. Let's observe the same with the help of an example.

**FileName:** IllegalArgumentException.java

1. // importing the java.lang package
2. **import** java.lang.\*;
4. **public** **class** IllegalArgumentException **extends** Thread
5. {
7. // the main method
8. **public** **static** **void** main(String argvs[])
9. {
11. // Now, priority of the main thread is set to 17, which is greater than 10
12. Thread.currentThread().setPriority(17);
14. // The current thread is retrieved
15. // using the currentThread() method
17. // displaying the main thread priority
18. // using the getPriority() method of the Thread class
19. System.out.println("Priority of the main thread is : " + Thread.currentThread().getPriority());
21. }
22. }

When we execute the above program, we get the following exception:

Exception in thread "main" java.lang.IllegalArgumentException

at java.base/java.lang.Thread.setPriority(Thread.java:1141)

at IllegalArgumentException.main(IllegalArgumentException.java:12)